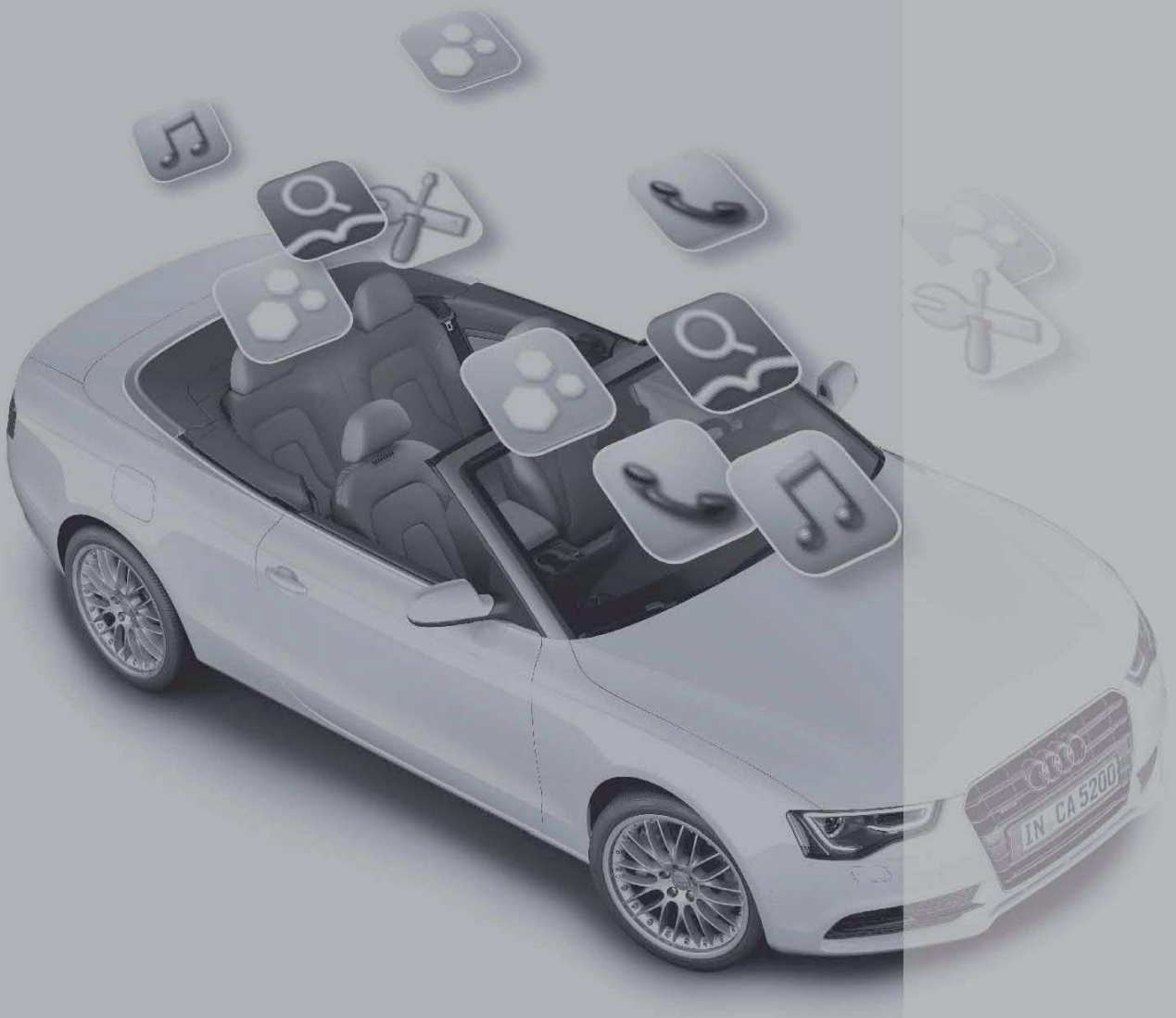
Audi
Dissertationsreihe

Interaktive Gestaltung von Automotive Services durch softwaregestützten Einsatz domänenspezifischer Modellierung

Maximilian Pühler



Cuvillier Verlag Göttingen

INI.TUM
INGOLSTADT INSTITUTE OF THE TU MÜNCHEN

Audi-Dissertationsreihe, Band 53

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Technische Universität München
Lehrstuhl für Wirtschaftsinformatik (I 17)
Univ.-Prof. Dr. Helmut Krcmar

Interaktive Gestaltung von Automotive Services durch softwaregestützten Einsatz domänenspezifischer Modellierung

Maximilian Pühler

Vollständiger Abdruck der von der Fakultät für
Informatik der Technischen Universität München
zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr. rer.nat.)
genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr. Dr. h.c. Manfred Broy
Prüfer der Dissertation:	1. Univ.-Prof. Dr. Helmut Krcmar
	2. Univ.-Prof. Gudrun J. Klinker, Ph.D.

Die Dissertation wurde am 29.06.2011 bei der Technischen Universität München
eingereicht und durch die Fakultät für Informatik
am 17.11.2011 angenommen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.
1. Aufl. - Göttingen : Cuvillier, 2011
Zugl.: (TU) München, Univ., Diss., 2011

978-3-86955-969-8

© CUVILLIER VERLAG, Göttingen 2011
Nonnenstieg 8, 37075 Göttingen
Telefon: 0551-54724-0
Telefax: 0551-54724-21
www.cuvillier.de

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Verlages ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem Weg (Fotokopie, Mikrokopie) zu vervielfältigen.
1. Auflage, 2011
Gedruckt auf säurefreiem Papier

978-3-86955-969-8

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Für Mirona

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Vorwort

Die vorliegende Dissertation entstand in den Jahren 2006 bis 2011 während meiner Zeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Wirtschaftsinformatik der Technischen Universität München von Prof. Dr. Helmut Krcmar.

Viele Menschen haben mich während meiner Dissertationszeit unterstützt, denen ich an dieser Stelle meinen Dank aussprechen möchte.

Besonders danken möchte ich meinem Doktorvater Herrn Prof. Dr. Helmut Krcmar für die wissenschaftliche Betreuung der Arbeit. Er hat mit wertvoller Anleitung zum Gelingen der Dissertation beigetragen und mir während des Entstehungsprozesses den notwendigen Freiraum eingeräumt. Frau Prof. Gudrun Klinker danke ich sehr herzlich für die freundliche Übernahme des Zweitgutachtens und Herrn Prof. Dr. Dr. h.c. Manfred Broy für den Vorsitz bei der mündlichen Prüfung.

Ermöglicht wurde diese Arbeit durch die großzügige inhaltliche und finanzielle Unterstützung der Audi AG. Im Rahmen der Ingolstädter Institute der Technischen Universität München (INI.TUM) wurde mir ein sehr enger und für die Arbeit wichtiger Kontakt und Zugang zum Fachbereich „IT im Fahrzeug“ (I/FP-343) der Audi IT ermöglicht. Hier möchte ich vor allem Stefan Sellschopp, Hubert Fischer, Stefan Bauer, Michael Faulbacher und Dr. Uwe Koser für die großzügige Unterstützung, aber auch für das Einräumen der notwendigen Freiräume, die für ein Gelingen dieser Arbeit notwendig waren, danken.

Eine sehr große Unterstützung waren auch meine Kollegen am Lehrstuhl für Wirtschaftsinformatik. Sie haben mir einerseits durch ihr eigenes Beispiel gezeigt, das man es schaffen kann, und mich so immer wieder aufs Neue motiviert. Andererseits waren sie aber auch meine Leidensgenossen, die meine Probleme und Schwierigkeiten beim Erstellen dieser Arbeit verstanden und mir mit Rat und Tat zur Seite standen. Auch wenn ich hier nicht alle anführen kann, so möchte ich doch wenigsten diejenigen, mit denen ich am Engsten zusammengearbeitet habe, nennen. Ich danke Dr. Michael Schermann, Dr. Holger Hoffmann, Sergej Trushin, Benjamin Schwering, Tobias Schlachtbauer, Armin Sharafi, Dr. Petra Wolf, Andreas Schwertzig, Dr. Stefanie Leimeister, Prof. Dr. Jan-Marco Leimeister, Jens Fähling, Dr. Holger Jehle, Stefan Hörmann, Marina Berkovich, Sebastian Esch, … und den vielen Weiteren, die mich auf meinem Weg begleitet haben.

Einen besonderen Dank möchte ich ebenfalls an Cathleen Stefan und Andrea Trost – Sekretärinnen und zugleich gute Seelen des Lehrstuhls – richten. Sie hatten immer ein offenes Ohr und standen mit guten Ratschlägen zur Seite. Ebenso gilt mein besonderer Dank Kurt Obermeier und Mathias Karl, die mich in allen praktischen und technische Belangen stets unterstützt haben. Schließlich danke ich allen Studenten die mir als studentische Hilfskräfte, in ihren Abschlussarbeiten sowie im Automotive Services Lab tatkräftig unter die Arme gegriffen haben.

Unverzichtbar für das Gelingen der Dissertation war die Unterstützung aus meinem privaten Umfeld. Großer Dank gebührt meiner Familie, insbesondere meinen Eltern. Sie haben mir in meiner gesamten Ausbildungszeit den notwendigen Beistand gegeben, mich uneingeschränkt unterstützt und gefördert. Auch danke ich meinem Freundeskreis für den Rückhalt, die Motivation und die notwendige Ablenkung.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Der größte Dank gilt jedoch meiner Freundin Mirona Marisch. Sie hat mit mir während dieser intensiven Zeit alle Höhen und Tiefen durchlebt und mir stets die notwendige Kraft gegeben auch in schwierigen Phasen durchzuhalten. Zudem hat sie mich immer wieder daran erinnert, dass es ein Leben neben der Dissertation gibt. Dafür bin ich Dir unendlich dankbar!

Maximilian Pühler

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Zusammenfassung

Automotive Services haben in den letzten Jahren eine beträchtliche Bedeutung in der Automobilindustrie gewonnen. Eine stetig zunehmende Verfügbarkeit von Systemressourcen, wie beispielsweise Rechenleistung, Speicherkapazitäten, oder hochauflösenden Bildschirmen erlaubt zunehmend komplexe Dienste in modernen Fahrzeugen. Zusätzlich beschleunigt ein kontinuierlicher Ausbau mobiler Breitbandanbindungen, wie zum Beispiel UMTS (3G) oder LTE (4G), diesen Trend. Zwei prominente Beispiele dieser Entwicklung sind ConnectedDrive von BMW oder das Audi Multi-Media-Interface.

Getrieben durch den enormen Erfolg von Smartphones, wie beispielsweise dem Apple iPhone, hat sich eine ubiquitäre Verfügbarkeit von mobilem Internet im Alltag etabliert. Dies hat natürlich auch die Erwartungen an die digitalen Möglichkeiten in modernen Fahrzeugen maßgeblich beeinflusst. So erwarten Kunden die gleiche Funktionalität, Flexibilität und Kommunikation auch in ihren Fahrzeugen. Aufgrund der speziellen Nutzungssituation ist allerdings eine einfache Übertragung der existierenden Technologien nur schwer möglich. So sind Automotive Services komplexe Leistungsbündel aus Software, Dienstleistung und einem Produkt (dem Fahrzeug), die in diesem Nutzungskontext zum Einsatz kommen.

Jedoch benötigt man für die frühe Anwendung von Prototyping auf den Prozess der Ideenfindung und Ideenentwicklung einen flexiblen Ansatz, um zwischen Erhebung und Validierung iterieren zu können. In die Praxis übertragen bedeutet dies, dass man eine Kommunikation und Kooperation der technischen und nichttechnischen Stakeholder sicherstellen muss. Wie bereits diskutiert, liefern die nichttechnischen Stakeholder in diesem Prozess das Wissen und die Erfahrung aus der Anwendungsdomäne, also das Verständnis darüber, welchen Nutzen ein Service stiften kann, die technische Stakeholder hingegen tragen ihre Erfahrungen aus der Umsetzungsdomänen, das heißt Fragen der Umsetzbarkeit und des effizienten Betriebs, bei.

Die vorliegende Arbeit stellt einen Ansatz zur modellgetriebenen Gestaltung, Definition und prototypischen Umsetzung von Automotive Services vor. Dadurch wird in einem Prozess des Design Thinking ebendiese Kommunikation und Kooperation zwischen technischen und nichttechnischen Stakeholder bei der Gestaltung von Automotive Services unterstützt. Nichttechnische Stakeholder werden in die Lage versetzt, ihre Vorstellungen und ihr Verständnis von Automotive Services zu explizieren ohne dabei ein Vorwissen über die technische Machbarkeit oder Fragen der Umsetzbarkeit zu benötigen. Dieses explizierte Wissen wiederum versetzt den technischen Stakeholder in die Lage, Aspekte der Umsetzung zu adressieren ohne dabei ein Verständnis über deren inhaltliche Gestaltung besitzen zu müssen. Zusätzlich werden wiederverwendbarer Teile von Services, sowohl in ihrer technischen als auch in ihrer nichttechnischen Ausgestaltung, für eine einfache Wiederverwendung verfügbar gemacht. Auf diese Weise können neben der effizienten Unterstützung des interaktiven Gestaltungsprozesses auch die Fragen der Kosten- und Zeitfaktoren bei der Ideenfindung und Gestaltung betrachtet werden.

Der Mehrwert dieser Arbeit liegt somit in der Gestaltung einer domänenspezifischen Modellierungssprache für Automotive Services sowie in der Bereitstellung von entsprechenden Unterstützungswerkzeugen für deren Einsatz. Ein grafisches Modellierungswerkzeug hilft hierbei den nichttechnischen Stakeholdern bei der Explikation ihre Vorstellungen und Ideen. Zusätzlich versetzt eine Laufzeitumgebung alle Beteiligten in

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

die Lage, modellierte Services sowohl an ihrem Rechner zu simulieren als auch diese im Fahrzeug zu evaluieren. Schließlich liefert eine graphische Repräsentation eine gemeinsame Kommunikationsplattform, sowohl über den gesamten Service als auch über einzelne funktionale Komponenten.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Inhaltsverzeichnis

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Abbildungsverzeichnis

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Tabellenverzeichnis

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Abkürzungsverzeichnis

ACM ..... *Association for Computing Machinery's*
ACM SIGCHI ..... *Special Interest Group in Computer-Human Interaction*
ARIS ..... *Architektur integrierter Informationssysteme*
ASM ..... *Automotive Service Model*
ASML ..... *Automotive Services Modeling Language*
AViCoS ..... *Avatar based Virtual Co-driver System*
BMW ..... *Bayerische Motoren Werke AG*
CAN ..... *Controller Area Network*
CHI ..... *Computer-Human Interaction*
COMMAND ..... *Cockpit Management and Navigation Device*
DSM ..... *Display Signal Manager*, *A Domain-Specific-Modeling*
eIT ..... *Service Innovationsprozess am Beispiel der Elektromobilität*
EMF ..... *Eclipse Modeling Framework*
EPK ..... *Ereignisgesteuerten Prozessketten*
FIS ..... *Fahrerinformationssystem*
GEF ..... *Graphical Editing Framework*
GOMS ..... *Goals, Operators, Methods, and Selection rules*
GPS ..... *Global Positioning System*, *Global Positioning System*
GUI ..... *Graphical User Interface*
HGW ..... *HIMEPP Graphical Workbench*
HTML ..... *Hypertext Markup Language*
IAA ..... *Internationale Automobil-Ausstellung*
IKT ..... *Informations- und Kommunikationstechnologie*
INI.TUM ..... *Ingolstadt Institute der TUM*
LBVS ..... *Location Based Vehicle Services*
LTE ..... *Long Term Evolution*
MACS ..... *Mobile Dienste im Auto der Zukunft*
MDA ..... *Model Driven Architecture*
MDD ..... *Model-Driven Development*
Memex ..... *Memory Extender*
MEWSIC ..... *Method Engineering with Stakeholder Input and Collaboration*
MMI ..... *Multi-Media-Interface*, *Mensch-Maschine Interaktion*
MOF ..... *Meta Object Facility*
OMG ..... *Object Management Group*
OSGi ..... *Open Services Gateway initiative*
PC ..... *Personal Computer*
POI ..... *Points of Interest*
PTB ..... *Push-Turn-Button*
RCP ..... *Rich Client Plattform*
RUP ..... *Rational Unified Process*
SEI ..... *Software Engineering Institute*
SMS ..... *Short Message Service*
SMTP ..... *Simple Mail Transfer Protocol*
SWT ..... *Standard Widget Toolkit*
TLS ..... *Transport Layer Security*

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

TMC ..... *Traffic Message Channel*
TUM ..... *Technische Universität München*
uBase ..... *Ubiquitous Automotive Services Environment*
UML ..... *Unified Modeling Language*, *Unified Modeling Language*
UMTS ..... *Universal Mobile Telecommunications System*
URL ..... *Uniform Resource Locator*

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

1 Automotive Services - Innovationtreiber der Automobilindustrie

Automotive Services haben in den letzten Jahren eine beträchtliche Bedeutung in der Automobilindustrie gewonnen. Eine stetig steigende Verfügbarkeit von Systemressourcen, wie beispielsweise Rechenleistung, Speicherkapazitäten, oder hochauflösenden Bildschirmen erlaubt zunehmend komplexe Dienste in modernen Fahrzeugen. Zusätzlich beschleunigt ein kontinuierlicher Ausbau mobiler Breitbandanbindungen, wie zum Beispiel UMTS (3G) oder LTE (4G), diesen Trend (Golcar et al. 2010; Viswanathan et al. 2007). Zwei prominente Beispiele dieser Entwicklung sind ConnectedDrive von BMW (BMW Group 2010a) oder das Audi Multi-Media-Interface (Audi AG 2010a).

Getrieben durch den enormen Erfolg von Smartphones, wie beispielsweise dem Apple iPhone, hat sich eine ubiquitäre Verfügbarkeit von mobilem Internet im Alltag etabliert (West/Mace 2010; Apple Inc. 2010). Dies hat natürlich auch die Erwartungen an die digitalen Möglichkeiten in modernen Fahrzeugen maßgeblich beeinflusst. So erwarten Kunden die gleiche Funktionalität, Flexibilität und Kommunikation auch in ihren Fahrzeugen. Aufgrund der speziellen Nutzungssituation ist allerdings eine einfache Übertragung der existierenden Technologien nur schwer möglich. So sind Automotive Services komplexe Leistungsbündel aus Software, Dienstleistung und einem Produkt (dem Fahrzeug), die in diesem Nutzungskontext zum Einsatz kommen (Hoffmann 2010).

Zusätzlich sieht sich der Automobilmarkt selbst neuen Herausforderungen gegenüber. Im Zuge der aktuellen Umwelt und Klimadiskussionen sind die Automobilhersteller gezwungen neue, innovative Wege wie beispielsweise die Elektromobilität in den Fokus ihrer Entwicklungen zu stellen. Aus diesem Wettbewerbsdruck heraus entstehen teils „phänomenale“ Anforderungen an die Automobilhersteller (Tinham 2007; Hoffmann 2010). Beispielsweise verhindern technologische Limitationen, wie z.B. verfügbare Batteriekapazitäten und lange Ladezyklen, einen einfachen Wechsel von fossilen Brennstoffen hin zu sauberer Elektromobilität. Ein viel versprechender Ansatz dieser Problematik zu begegnen sind spezielle Nutzungsmodelle, wie zum Beispiel Miet-Modelle oder auch Konzepte für Zweitfahrzeuge, vor allem für den urbanen Raum. Um solche Modelle erfolgreich am Markt etablieren und sich so vom Wettbewerb abgrenzen zu können ist eine geeignete Unterstützung des Kunden durch Automotive Services unerlässlich (Becker 2002).

1.1 Motivation und Problemstellung

Vergleicht man die Domäne der Automotive Services mit anderen Servicedomänen, stellen sich die besonderen Nutzungsszenarien während der Fahrt als die wesentlichen Herausforderungen bei der Gestaltung von Automotive Services heraus. Neben dem Fahrer selber lassen sich der Beifahrer sowie die Passagiere als zusätzliche Nutzergruppen identifizieren. Jede dieser Gruppen stellt unterschiedliche Anforderungen und Bedingungen an die Nutzung von oftmals demselben Automotive Service. So dürfen Services den Fahrer zu keiner Zeit von seiner primären Aufgabe, dem sicheren Steuern des Fahrzeugs, ablenken. Daher sollten Automotive Services speziellen Design Guidelines genügen und beispielsweise eine Minimierung der visuellen Beeinträchtigung des Fahrers berücksichtigen. Natürlich müssen auch die Anforderungen des Beifahrers Beachtung finden. So darf beispielsweise die

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Darstellung eines Musikvideos für den Fahrer während der Fahrt nicht verfügbar sein, jedoch aber für den Beifahrer und die Passagiere auf der Rückbank. Automotive Services müssen also unterschiedliche Nutzungsbedingungen und daraus resultierende Anforderungen zur gleichen Zeit genügen.

Ein weiterer wesentlicher Faktor bei der Gestaltung von Automotive Services sind die limitierten Hardwareressourcen die im Fahrzeug zur Verfügung stehen. Typischerweise laufen Automotive Services auf speziellen gewichts- und energieoptimierten Embedded Systemen, die nur eingeschränkte Systemressourcen zur Verfügung stellen können. Zusätzlich erlaubt das Interieur eines Fahrzeugs in der Regel nur relativ kleine Bildschirme und spezielle Funktionstasten oder Bedienelemente, zum Beispiel „Dreh-Drück-Steller“, um mit den Nutzern zu interagieren (Harman Becker Automotive Systems GmbH 2003). Auch wenn moderne Fahrzeuge die Möglichkeit einer Sprachbedienbarkeit zur Verfügung stellen, bleibt die Interaktion der Nutzer mit dem Automotive Service während der Fahrt stark limitiert. Daher scheitert der Versuch erfolgreiche Dienste aus anderen Domänen direkt auf das Fahrzeug zu übertragen oftmals an einfachen Limitationen wie beispielsweise dem Mangel an einer geeigneten Tastatur für die Texteingabe.

Aufgrund der engen Verzahnung von technischen Design- und Sicherheitsanforderungen sind Automotive Services stark integrierte Bündel aus Hardware, Software und externen Services. Dies stellt eine weitere Herausforderung dar, da Hardware und Software unterschiedlichen Lebenszyklen unterworfen sind (Leimeister/Glauner 2008). Die durchschnittliche Lebensdauer eines Fahrzeugs bewegt sich in einen Zeitraum von 15-20 Jahren, was die verfügbaren Hardwareressourcen über einen langen Zeitraum konstant hält. Vergleicht man dies mit der typischen Lebensdauer moderner Software, so erkennt man hier schnell eine starke Diskrepanz. Trotzdem erwarten Kunden stets die neuesten Anwendungen und Möglichkeiten in ihrem Fahrzeug, wie zum Beispiel aktuell die Bereitstellung von mobilem Internet. Daher ist der Prozess von der Serviceidee bis hin zur Serviceverfügbarkeit im Fahrzeug riskant, aufwändig und zeitintensiv. Obwohl die Ansätze von Serviceplattformen, wie sie von einigen Automobilherstellern vorgeschlagen werden (Audi AG 2010b), die Problematik der Entkopplung von Hardware und Software Lebenszyklen adressieren, werfen sie neue Fragestellungen hinsichtlich von Design und Sicherheitsrestriktionen auf.

Zusammenfassend lassen sich somit drei wesentliche Kriterien für erfolgreiche Automotive Services feststellen. Zunächst ist es für Automotive Services unabdingbar die speziellen Bedingungen aus dem Kontext Automobilnutzung zu berücksichtigen. Desweiteren dürfen im Fahrzeug integrierte Services in keinem Fall die primären Aufgaben des Fahrers beeinträchtigen. Schließlich muss ein Automotive Service auch einen starken Business-Case darstellen um den ressourcenaufwändig Prozess von der Ideenfindung bis hin zur öffentlichen Verfügbarkeit zu rechtfertigen. Um die dargestellten Risiken bei der Gestaltung von Automotive Services für die Hersteller beherrschbar zu machen, ist ein eingehendes Verständnis der Wertschöpfungsmöglichkeiten von Ideen für Automotive Services von zentraler Bedeutung.

Erfolgreiche Automotive Services bieten einen signifikanten Mehrwert für das dem Fahrer, Beifahrer und Passagieren gebotene Fahrerlebnis und ermöglichen darüber hinaus kreative und innovative Herangehensweisen an Herausforderungen wie beispielsweise die der Elektromobilität. Um den zu Grunde liegenden Wertbeitrag von Automotive Services gestalten und nutzen zu können müssen technische und nichttechnische Stakeholder unterschiedlichster Anwendungsdomänen in den Gestaltungsprozess eng integriert werden. So

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

es ist die maßgebliche Aufgabe der Automotive Service Entwicklung das Know-How der Experten der Anwendungsdomänen zusammen mit dem der Experten der Umsetzungsdomänen nutzbar zu machen. Tabelle 1-1 ordnet diese Gruppen entlang ihrer unterschiedlichen Kompetenzprofile an.

	Hohes Application Domain Knowledge	Niedriges Application Domain Knowledge
Hohes Information Systems Knowledge	Automotive Service Experten	Experten der Umsetzungsdomänen
Niedriges Information Systems Knowledge	Experten der Anwendungsdomäne	Sonstige Stakeholder

Tabelle 1-1 Übersicht der wesentliche Stakeholdergruppen
Quelle: Eigene Darstellung in Anlehnung an Abschnitt 3.1.2

Grundsätzlich stellen Automotive Service Experten die zentralen Schlüsselfiguren entlang des Entwicklungsprozesses von Automotive Services dar. Diese werden unterstützt durch die Experten der Umsetzungsdomäne die wesentlich zur Realisierung und Bereitstellung der Services beitragen, sowie durch die Experten der Anwendungsdomäne deren Fokus auf inhaltlichen, strategischen und ökonomischen Faktoren liegt. Der Gruppe der sonstigen Stakeholder fällt zumeist die Rolle des Ideengebers oder Evaluators zu. Oft bilden Kunden diese Gruppe und können somit nur unter klar definierten Rahmenbedingungen, z.B. im Rahmen einer Kundenbefragung, in den Gestaltungsprozess mit einbezogen werden. Forschungsarbeiten rund um den Ansatz der Open Innovation legen nahe, dass eine frühe Integration externer Stakeholder in den Entwicklungsprozess zu wesentlich innovativeren Services führt (Chesbrough 2003; Faber 2009).

Eine integrierte Betrachtung und Repräsentation technischer als auch nichttechnische Charakteristiken von einzelnen Services ist von zentraler Bedeutung für das Verständnis des Wertbeitrages eines Automotive Services. Beispielsweise interessieren sich Kunden primär für den Nutzen den ein konkreter Service ihnen bietet, während Ingenieure eher den Fokus auf die technische Umsetzung und Sicherstellung der Verfügbarkeit legen. Zusätzlich müssen Automobilhersteller zu jeder Zeit die Einhaltung regulatorischer Bestimmungen für ihre Automotive Services dokumentieren und nachweisen können.

Brooks stellt bereits 1986 fest: “The hardest single part of building a software system is deciding precisely what to build. […] No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later” (Brooks 1986). Demzufolge hängt der Erfolg bei der Entwicklung von Automotive Services von der Fähigkeit ab, ein gemeinsames Verständnis sowie eine gemeinsame Vorstellung der Stakeholder und Entwickler zu finden und sicherzustellen. Zusätzlich müssen ausgewählte Bestandteile im Vorfeld hinsichtlich existierender Sicherheitsbestimmungen überprüft werden können. Es ist daher nicht weiter überraschend, dass sich in der Literatur und in der Praxis

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

eine Fülle an Ansätzen zur Erhebung und Erlangung eines solchen Verständnisses finden lassen (Briggs/Gruenbacher 2002; Brügge/Dutoit 2004; Krauß/Versteegen/Mühlbauer 2006).

Jedoch verstehen Stakeholder nur selten ihre Anforderungen vollständig zu Beginn eines Projekts. Oft fehlt ihnen eine klare Vorstellung was das System können soll oder sie ändern diese während des Entwicklungsprozesses. Zumeist werden sich Stakeholder erst dann klar darüber was sie eigentlich wollen, wenn sie ein System vor sich sehen dass alle wesentlichen Funktionalitäten umfasst[1] (Antón 2003). Breindahl schlägt daher vor ein solches System im Dialog mit den Nutzern zu gestalten (Breindahl 2008). Eine zentrale Rolle bei der Validierung des gemeinsamen Verständnisses fällt dabei Prototypen zu, die einen methodologischen Ansatz für eine evolutionäre Systementwicklung darstellen (Floyd et al. 1989). Dieser Gestaltung eines gemeinsamen Verständnisses unter Stakeholdern, Entwicklern und Kunden fällt besonders in der Automobilbranche eine besondere Bedeutung zu. Hinsichtlich ihrer speziellen Rahmenbedingungen, wie beispielsweise den Sicherheitsimplikationen oder dem besondere Einsatzumfeld, dem Fahrzeug, ist das Finden und Definieren valider Anforderungen von zentraler Bedeutung.

1.2 Vision der Arbeit

Ausgehend von den genannten Überlegungen wird in der vorliegenden Arbeit die Anwendung des Paradigmas des Design Thinking (Brown 2009) auf die Domäne der Gestaltung von Automotive Services vorgeschlagen. Ziel ist es, die Entwicklung von Automotive Services, also von Mehrwertdiensten im Fahrzeug, durch den Einsatz einer domänenspezifischen Modellierungssprache in einem iterativen Vorgehen zu unterstützen. Dies kann durch eine frühe Verfügbarkeit von Prototypen auf Basis der Modellierung erreicht werden.

Wie Eingangs dargelegt, zeigt sich in der Wissenschaft als auch in der Praxis deutlich, dass Prototyping das Finden des gemeinsamen Verständnisses des Wertbeitrags von Innovationen positiv unterstützt. Jedoch ist die Entwicklung von Prototypen eine sehr zeitaufwändige Aufgabe, deren Kosten mit der notwendigen Anzahl an Iterationen steigt (Grechanik/Conroy/Probst 2007). Konkret müssen bei der Gestaltung von Automotive Services Entwickler Prototypen dieser zunächst in das Fahrzeug integrierten, um einen Dialog mit den Stakeholdern zu ermöglichen. Das daraus resultierende Feedback wiederum muss in Änderungen einzelnen Komponenten der Prototypen, Hardware, Software und Services übertragen werden.

Jedoch benötigt man für die frühe Anwendung von Prototyping auf den Prozess der Ideenfindung und Ideenentwicklung einen flexiblen Ansatz um zwischen Erhebung und Validierung iterieren zu können. In die Praxis übertragen bedeutet dies, dass man eine Kommunikation und Kooperation der technischen und nichttechnischen Stakeholder sicherstellen muss. Wie bereits diskutiert liefern die nichttechnischen Stakeholder in diesem Prozess das Wissen und die Erfahrungen aus der Anwendungsdomäne, also das Verständnis darüber welchen Nutzen ein Service stiften kann, die technische Stakeholder entgegen tragen ihre Erfahrungen aus der Umsetzungsdomänen, das heißt Fragen der Umsetzbarkeit und des effizienten Betriebs, bei.

[1] Übersetzung durch den Author. Originalzitat: “stakeholders seldom understand their requirements fully at the beginning of a project. They often lack a clear vision of what the system should do and change their minds during development. In general, stakeholders only become surer about what they want when they see a system that does not exhibit the necessary features” (Antón 2003)

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Die vorliegende Arbeit stellt einen Ansatz zur modellgetriebenen Gestaltung, Definition und prototypische Umsetzung von Automotive Services vor. Dadurch wird in einem Prozess des Design Thinking ebendiese Kommunikation und Kooperation zwischen technischen und nichttechnischen Stakeholder bei der Gestaltung von Automotive Services unterstützt. Nichttechnische Stakeholder werden in die Lage versetzt, ihre Vorstellungen und ihr Verständnis von Automotive Services zu explizieren, ohne dabei ein Vorwissen über die technische Machbarkeit oder Fragen der Umsetzung zu benötigen. Dieses explizierte Wissen wiederum versetzt den technischen Stakeholder in die Lage, Aspekte der Umsetzung zu adressieren ohne dabei ein Verständnis über deren inhaltliche Gestaltung zu besitzen. Zusätzlich werden wiederverwendbarer Teile von Services sowohl in ihrer technischen als auch in ihrer nichttechnischen Ausgestaltung für eine einfache Wiederverwendung verfügbar gemacht. Auf diese Weise können neben der effizienten Unterstützung des interaktiven Gestaltungsprozesses auch die Fragen der Kosten- und Zeitfaktoren bei der Ideenfindung und Gestaltung betrachtet werden.

Der Mehrwert dieser Arbeit liegt somit in der Gestaltung einer domänenspezifischen Modellierungssprache für Automotive Services sowie in der Bereitstellung von entsprechenden Unterstützungswerkzeugen für deren Einsatz. Ein grafisches Modellierungswerkzeug hilft hierbei den nichttechnischen Stakeholdern bei der Explikation ihre Vorstellungen und Ideen. Zusätzlich versetzt eine entsprechende Laufzeitumgebung alle beteiligten Stakeholder in die Lage modellierte Services sowohl an ihrem Rechner zu simulieren als auch diese im Fahrzeug zu evaluieren. Schließlich liefert eine graphische Repräsentation eine gemeinsame Kommunikationsplattform, sowohl über den gesamten Service als auch über einzelne funktionale Komponenten.

1.3 Forschungsleitende Fragestellungen

Um die skizzierte modellgetriebene Gestaltung und Entwicklung von Automotive Services zu erreichen ist es notwendig diese Zielsetzung anhand von vier aufeinander aufbauenden forschungsleitenden Fragestellungen genauer zu betrachten. Zunächst wird systematisch die Domäne der Automotive Services untersucht um darauf aufbauend eine domänenspezifische Modellierung, die den speziellen Anforderungen der Domäne gerecht wird, gestalten zu können. Auf dieser Grundlage werden die notwendigen Softwarewerkzeuge erarbeitet um die entwickelte Modellierung geeignet zu unterstützen. Schließlich erfolgt in einem abschließenden Schritt eine Evaluation des Modellierungsansatzes um daraus Implikationen für Wissenschaft und Praxis ableiten zu können.

Forschungsfrage 1 ***Welche Bedeutung haben Automotive Services für die Automobilindustrie und welche Anforderungen leiten sich daraus für eine domänenspezifische modellgetriebene Gestaltung ab?***

Im Rahmen der ersten Forschungsfrage wird die Bedeutung von Automotive Services für die Automobilindustrie betrachtet. Ausgehend von einer begrifflichen Klärung werden Trends und Innovationstreiber der Branche diskutiert sowie die aktuellen Aktivitäten einiger bedeutender Automobilhersteller sowie Zulieferer betrachtet. Ausgehend davon werden Herausforderungen bei der Ermittlung der Anforderungen von Automotive Services sowohl in der Literatur als auch im Rahmen einer praktischen Vorstudie ermittelt. Es wurden Vertreter der Branchen Softwareentwicklung, Medizintechnik und Automotive zu deren Vorgehen,

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Problemen und Erfahrungen bei der Erhebung von Anforderungen befragt. In einem zweiten Schritt werden, ausgehend von den bisherigen Erkenntnissen, Anforderungen an eine Modellierungssprache für Automotive Services diskutiert. Hierzu wird insbesondere auf den zu Grunde liegenden Design Thinking Ansatz eingegangen um eine durchgängige Unterstützung der einzelnen Phasen zu gewährleisten. Zusätzlich werden diese konzeptionellen Anforderungen durch konkrete Anforderungen aus der Praxis ergänzt. Hierzu wird zunächst das vorliegende Forschungsfeld betrachtet um die wesentlichen Stakeholder zu identifizieren und diese im Rahmen eines Innovationsworkshops sowie einer anschließenden Ideen Community in den Anforderungsprozess integriert. Ergebnis dieser Forschungsfrage sind sowohl konzeptionelle Anforderungen an die Modellierung als auch praktische Anforderungen an deren Ausgestaltung und softwaretechnische Unterstützung.

Forschungsfrage 2 ***Wie sieht eine domänenspezifische Modellierungssprache für Automotive Services aus und was sind deren zentrale Designprinzipien und Gestaltungselemente?***

Ausgehend vom in der ersten Forschungsfrage aufgebauten Problemverständnis und den daraus abgeleiteten Anforderungen an eine domänenspezifische Modellierungssprache für Automotive Services, befasst sich Forschungsfrage zwei mit der Ausgestaltung der wesentliche Designprinzipien und Gestaltungselemente. Dass Ergebnis der zweite Forschungsfrage ist die Definition einer domänenspezifischen Modellierungssprache für Automotive Services sowie deren Erläuterung anhand zweier Case Studies.

Forschungsfrage 3 ***Welches sind die Gestaltungselemente einer durchgängigen Softwareunterstützung für die gefundene Modellierungssprache und welche Architekturmerkmale leiten sich aus diesen ab?***

Kern dieser Forschungsfrage ist die Gestaltung einer durchgängigen Werkzeugunterstützung für die Modellierung von Automotive Services. Ausgehend von einer Motivation der Werkzeugunterstützung wird zunächst deren grundlegende Architektur und Funktionsweise erarbeitet. Schließlich erfolgt eine detaillierte Betrachtung der einzelnen Elemente der Werkzeugunterstützung hinsichtlich deren konzeptionellen Grundlagen, architektonischen Eigenarten und Besonderheiten in der Umsetzung. Des Weiteren wird ein plattformunabhängiges Datenformat für den Austausch von Automotive Service Modellen diskutiert.

Forschungsfrage 4 ***Welche Implikationen an eine Weiterentwicklung und einen Einsatz des Modellierungskonzepts ergeben sich aus der praktischen Anwendung durch technische und nichttechnische Stakeholder?***

Ziel der vierten und letzten Forschungsfrage ist die Evaluation des entwickelten Modellierungskonzepts. Hierbei wird der Fokus insbesondere auf die Integration technischer und nichttechnischen Stakeholder in den Gestaltungsprozess von Automotive Services gelegt. Die in Forschungsfrage drei entwickelte Werkzeugunterstützung dient dabei den Probanden die Modellierung in einem realen Kontext einzusetzen und so belastbar Aussagen über dieses

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

gewinnen zu können. Die erzielten Erkenntnisse dienen einerseits einer zukünftigen Weiterentwicklung des Modellierungskonzeptes sowie einer funktionalen Verbesserung der unterstützenden Softwarewerkzeuge.

1.4 Forschungsmethodisches Design

Forschungsarbeiten um Umfeld der Informationssysteme können auf zwei unterschiedliche Arten erfolgen. Einerseits können bestehende Informationssysteme, also IT-Artefakte, in ihrem Nutzungskontext beobachtet und erforscht und andererseits neue Artefakte zur Lösung erkannter Probleme entwickelte werden. Der erste Ansatz, der empirisch orientierte Behaviorismus, befasst sich eingehend mit der Messung der Auswirkungen von IT-Systemen auf deren Nutzer (Bichler 2004). Allerdings können durch diese Konzentration auf die „Verwendung und Auswirkung von existierenden Informationssystemen auf Individuen, Gruppen und Organisationen […] Potentiale der Technologien zur Lösung organisatorischer Probleme“ (Bichler 2004) oftmals nicht schnell genug erforscht werden (March/Smith 1995; Jennex 2001; Hevner et al. 2004).

Diesem Problem begegnet der zweite Ansatz, die Design Science, durch die Befassung mit der Gestaltung neuer Artefakte zur Lösung von Problemstellungen sowie der Anwendung der daraus gewonnenen Erkenntnisse (Simon 1996, 114ff). Diese Artefakte können Konstrukte, Modelle, Methoden, verbesserte Theorien oder Instanziierungen sein (March/Smith 1995, 225ff.; Purao 2002) Einen wesentlichen Ansatz zur praktischen Umsetzung einer Design Science orientierten Forschung haben Takeda et al. (1990) beigetragen. Ihnen zufolge orientiert sich das Vorgehen an einem fünf Phasen umfassenden Zyklus, welcher ausgehend von der Entwicklung eines Problembewusstseins über die Erarbeitung eines Lösungsvorschlags, dessen Umsetzung und Evaluation hin zu einer abschließenden, den nächsten Zyklus einleitenden, Schlussfolgerung umfasst (Takeda et al. 1990):

- *Problembewusstsein:* Im ersten Schritt wird die spezifische Problemstellung identifiziert und spezifiziert. Dies geschieht in der Regel durch Identifikation der Anforderungen und Herausforderungen der speziellen Problemdomäne, welchen entweder gar nicht oder nicht ausreichend durch eine bekannte Lösung begegnet wird.
- *Lösungsvorschlag:* Im zweiten Schritt werden auf der Grundlage der identifizierten Problemstellung Lösungsvorschläge entwickelt die in der nächsten Phase des Vorgehens realisiert werden können.
- *Umsetzung:* Der Kern des Vorgehens ist die Umsetzung und Entwicklung einer Lösung für das erkannte Problem. In diesem Schritt wird die vormals vorgeschlagene Lösungsalternative in Form eines Artefakts realisiert
- *Evaluation:* Die Aufgabe der Evaluation ist die Bewertung der Ursache-Wirkungs-Beziehung aus Problemstellung und realisiertem Lösungsvorschlag. Letzterer wird demnach hinsichtlich seiner Eignung zur Lösung der beschriebenen Problemstellung bewertet. Sekundäre Problemstellungen, die sich aus der Gestaltung des Artefakts ergeben, stoßen ihrerseits wiederum einen neuen Design Science Zyklus an.
- *Schlussfolgerung:* In der letzten Phase werden schließlich die in der Evaluation gewonnen Erkenntnisse ausgewertet und für die nächste Iteration des Zyklus vorbereitet.

Der ursprüngliche Ansatz von Takeda et al. (1990) berücksichtigt jedoch vornehmlich den eigentlichen Problemfindungsprozess und vernachlässigt das Sammeln und Aufbewahren der gefundenen Erfahrungen und Erkenntnisse. Ziel einer gestaltungsorientierten Forschung ist

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

jedoch nicht nur das Lösen eines spezifischen Problems, sondern vielmehr das Erlangen von Erkenntnissen die auf ähnliche und verwandte Problemstellungen übertragen werden können. March und Smith (1995) teilen das Vorgehen nach Takeda et al. (1990) daher in zwei Aufgabenstellungen auf. Einerseits wird der Prozess der Gestaltung auf die Phasen der Entwicklung und Evaluation beschränkt und andererseits die Phasen der Theorisierung und Rechtfertigung eingeführt. Diese beiden Prozesse werden auch als Induktion und Deduktion bezeichnet (Simon 1996, 113). Unter Induktion versteht man die innere Umwelt der eigentlichen Artefaktgestaltung und Evaluation und unter Deduktion die äußere Umwelt der Problemstellung und Aufbewahrung der gewonnenen Erkenntnisse. An den Schnittstellen dieser beiden Umwelten entsteht demnach der wissenschaftliche Beitrag einer Design Science Forschung (Simon 1996, 113).

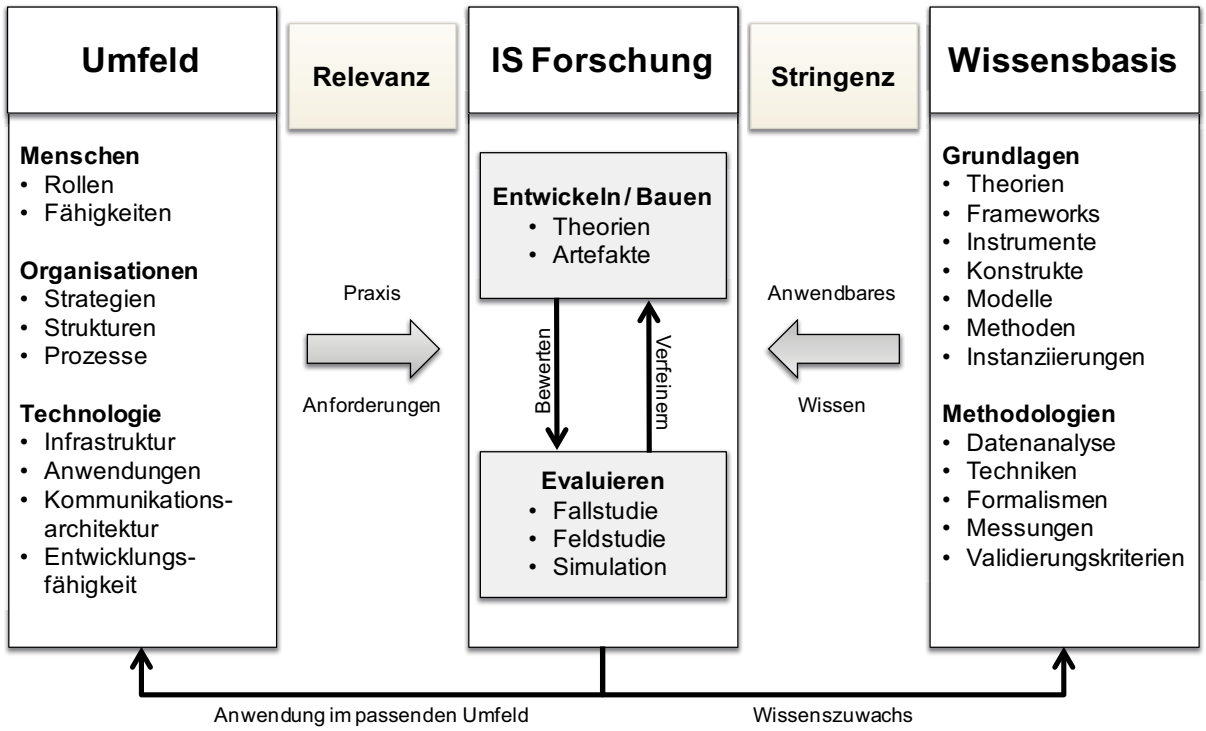


Abbildung 1-1 Rahmenwerk der Forschungsmethodik für Design Science
Quelle: Eigene Darstellung nach (Hevner et al. 2004)

Auf der Grundlage dieser Vorarbeiten haben Hevner et al. (2004) ein Rahmenwerk der Forschungsmethodik für Design Science vorgeschlagen. Die induktive Umwelt wird, wie in Abbildung 1-1 zu sehen, im Entwickeln und Evaluieren des zentralen Designprozesses beschrieben, wohingegen die deduktive Umwelt weiter in die beiden Bereiche Umfeld und Wissensbasis aufgeteilt wird. Problemstellungen werden demnach aus dem Umfeld abgeleitet und die entstandenen Artefakte auch wiederum in diesem Angewendet und Evaluiert. Durch diesen Prozess kann eine Relevanz der gestaltenden Forschungsarbeit sichergestellt werden. Die Aufgabe der Wissensbasis ist es hingegen das für die Gestaltung notwendige und anwendbare Wissen zur Verfügung zu stellen. Gleichzeitig werden Erkenntnisse, die aus der Artefaktgestaltung und Evaluation gewonnen werden, wieder zurück in die Wissensbasis gegeben und diese so erweitert. Auf diese Weise kann eine stringente Forschung erreicht werden.

Der wissenschaftliche Beitrag einer nach diesem Rahmenwerk ausgerichteten Forschungsarbeit ist somit einerseits ein Beitrag für die Praxis, der aus dem erstellten Artefakt und dessen praktischer Anwendung besteht, sowie andererseits einem Beitrag an die

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Wissenschaft, indem die in der Wissensbasis zur Verfügung gestellten Erkenntnisse das verfügbare Wissen erweitern.

1.5 Aufbau der Arbeit

Ziel der vorliegenden Arbeit ist das Ermöglichen und Unterstützen einer interaktiven Gestaltung von Automotive Services durch softwaregestützten Einsatz domänenspezifischer Modellierung. Im ersten Abschnitt der Arbeit wird die Motivation und Problemstellung sowie die Vision einer Modellierung von Automotive Services und die daraus abgeleiteten forschungsleitenden Fragestellungen dargelegt. Desweiteren erfolgt auf der Basis des zugrundeliegenden forschungsmethodischen Designs die Darstellung des Aufbaus der Arbeit.

In Abschnitt 2 wird die Bedeutung von Automotive Services für die Automobilindustrie betrachtet. Ausgehend von einer begrifflichen Klärung werden Trends wie Fahrerassistenzsysteme, Infotainmentsysteme oder mobile Dienste sowie Innovationstreiber, wie Elektromobilität oder Smartphones, diskutiert. Anschließend werden Beispiele der Aktivitäten von Automobilhersteller und Zulieferern im Bereich der Automotive Services vorgestellt um einen Überblick über die Komplexität und Entwicklung der Domäne zu erlangen.

Abschnitt 3 diskutiert die Herausforderungen bei der Gestaltung von Automotive Services. Hierzu werden zunächst der relevanten Anspruchsgruppen identifiziert und deren unterschiedliche Kompetenzprofile betrachtet. Ergänzt wird dies durch eine Analyse der Herausforderungen und Probleme der Anforderungsermittlung. Im Rahmen einer Vorstudie wird die aktuelle Situation des Requirements Engineering in der Praxis betrachtet. Hierzu wurden Unternehmen aus den Bereichen Softwareentwicklung, Medizintechnik und Automotive zu deren Erfahrungen und Einschätzungen befragt. Zusätzlich werden noch die Themenfelder der Mensch-Maschine-Interaktion sowie die Gestaltung von Infotainmentsystemen diskutiert. Zusammen mit den Beobachtungen des zweiten Abschnitts wird in diesem Kapitel die deduktive Umwelt des Umfelds der Forschungsarbeite betrachtet. Ergebnis ist ein informiertes Problembewusstsein auf dessen Grundlage im nächsten Schritt Lösungsvorschläge erarbeitet werden können.

Abschnitt 4 stellt die Anforderungen an eine graphische, domänenspezifische Modellierung von Automotive Services zusammen. Hierzu werden relevante Informationen und Theorien aus der deduktiven Umwelt der Wissensbasis zusammengetragen und diskutiert. Ausgehend von einer Betrachtung der Grundlagen der Modellierung von Automotive Services werden konzeptionelle Anforderungen aus dem Vorgehen des Design Thinking sowie konkrete Anforderungen der an diesem Forschungsvorhaben beteiligten Praxispartner diskutiert. Gemeinsam mit den beiden vorangegangen Abschnitten 2 und 3 stellt Abschnitt 4 die Beantwortung der ersten Forschungsfrage dar und beschreibt den Lösungsvorschlag nach Takeda et al. (1990).

In Abschnitt 5 folgt die Gestaltung einer domänenspezifischen Modellierungssprache für Automotive Services. Zunächst werden die grundlegenden Designprinzipien, die in Abschnitt 4 vorgeschlagen wurden, realisiert und darauf aufbauend die Elemente der Modellierungssprache beschrieben. Zuletzt wird der Einsatz der Modellierung am Beispiel von zwei Case Studies erläutert. Abschnitt 5 beantwortet somit die zweite Forschungsfrage und beschreibt den ersten Teil des zu entwickelten Artefakts.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Abschnitt 6 beschäftigt sich mit der Gestaltung einer durchgängigen Werkzeugunterstützung für die Modellierung von Automotive Services und umfasst somit die Bearbeitung der dritten Forschungsfrage. Ausgehend von einer Diskussion der Motivation der Werkzeugunterstützung erfolgt eine Architektur und Beschreibung der gesamten Werkzeugkette, welche im späteren Verlauf des Kapitels im Einzelnen bezüglich der jeweiligen Grundlagen, Architekturmerkmale und Umsetzungsbesonderheiten diskutiert wird. Des Weiteren werden in diesem Kapitel die zu Grunde liegenden technologischen Grundlagen der Modellierung als auch der Gestaltung der Laufzeitumgebung vorgestellt. Besonderes Augenmerk liegt hier auf dem Prototypingframework HIMEPP, welches im Rahmen der Forschungsarbeit von Hoffmann entstanden ist (Hoffmann 2010) und im Sinne einer domänenspezifischen Modellierung als domänenspezifisches Framework herangezogene wird (Tolvanen/Kelly 2004). In Anlehnung an Hevner et al. (2004) bilden die Ergebnisse dieses Kapitels zusammen mit dem letzten Abschnitt das Artefakt im Sinne des Rahmenwerks der Forschungsmethodik für Design Science (vgl. Abbildung 1-1).

Abschnitt 7 beschreibt die Evaluation des Modellierungskonzepts. Zunächst werden Evaluationsziele und Grundlagen der Artefaktevaluation behandelt um ausgehend von diesem Vorwissen das Evaluationsdesign zu gestalten. Im Rahmen von drei Fallstudien werden die zentralen Designprinzipien der Modellierung sowie deren Werkzeugunterstützung durch unterschiedliche Stakeholder evaluiert. Eine Darstellung der Evaluationsergebnisse und abschließende Diskussion derselben beantwortete somit die vierte und letzte Forschungsfrage.

Im abschließenden Abschnitt 8 werden nochmals die Ergebnisse der vorliegenden Forschungsarbeiten zusammengefasst und insbesondere auf den Beitrag der Arbeit zur Wissenschaft und Praxis eingegangen. Es werden also die Anwendung der Erkenntnisse im passenden Umfeld sowie der Zuwachs zur Wissensbasis beschrieben. Eine Diskussion der Limitationen der Arbeit führt schließlich zu einem Ausblick auf weiteren Forschungsbedarf.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

2 Stand der Technik: Automotive Services

Mit dem zunehmenden Einzug softwarebasierter Zusatzsysteme in Automobilen geht sowohl ein Wandel des Einsatzgebiets dieser Systeme einher, als auch ein Wandel im Selbstverständnis der Hersteller von reinen Produzenten zum nachgelagerten Dienstleister. Leimeister (2010) beschreibt resultierend die erleichterte Entwicklung kritischer Systeme und daneben die Bereitstellung einer Infrastruktur für neuartige mobile Dienste. Eben diese ermöglichen es wiederum den Herstellern, ihre bisherige Rolle zu erweitern und so dem Konkurrenzdruck durch stärkere Differenzierung zu entgehen. Golcar et al. (2010) führen in diesem Kontext an, dass Markenloyalität im Automobilsektor in wesentlichem Maße von der Kundenzufriedenheit beeinflusst wird und eine fortlaufende Kundenzufriedenheit durch die Integration dieser Dienste in die dem Kunden zur Verfügung gestellten Systeme erreicht werden kann. Mobile Mehrwertdienste im Automobil, die über das reine Bedienen des Fahrzeugs hinaus auf die Befriedigung verschiedenster Kundenbedürfnisse abzielen, werden in diesem Sinne als Automotive Services bezeichnet.

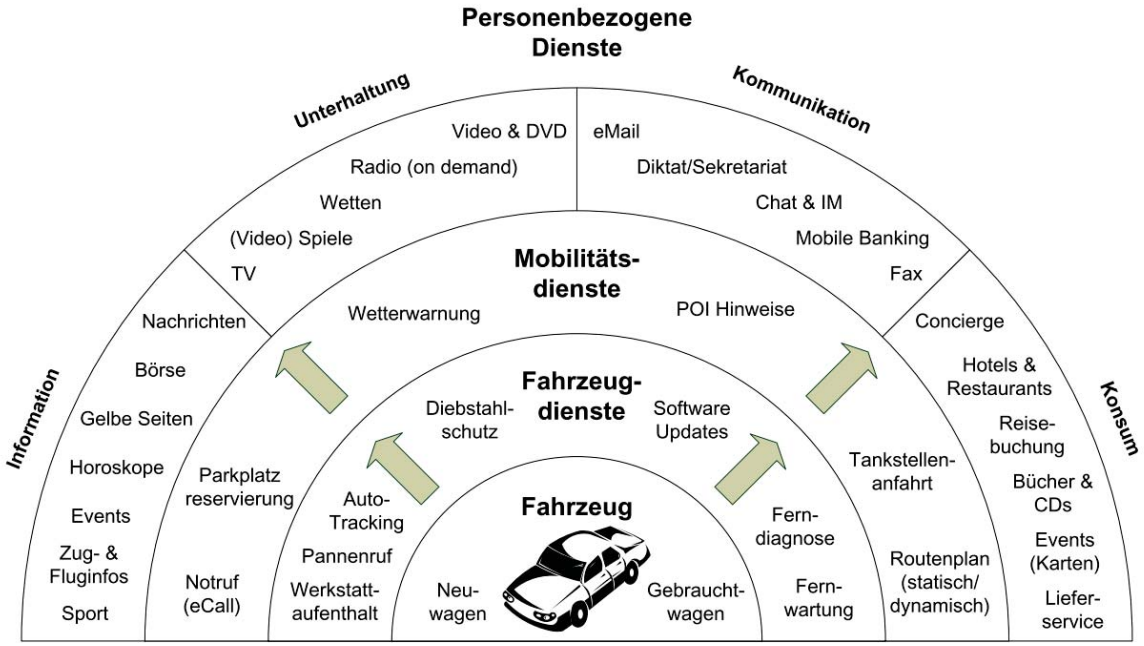


Abbildung 2-1 Klassifikation für fahrzeug- und kundenbezogene Dienste im Fahrzeug
Quelle: (Ehmer 2002; Hoffmann 2010)

Parallel zur aufgezeigten Entwicklung von rein fahrzeugorientierten Systemen hin zu universellen Dienstleistungsplattformen lässt sich nach Ehmer (2002) ein Klassifikationsschema für Automotive Services konstruieren. Dieses untergliedert, wie in Abbildung 2-1 zu sehen, mobile Dienste nach ihrem Bezug und ihrer Nutzenkategorie. Je personenbezogener und damit individueller ein Dienst wird, desto breiter fächert sich folglich das Spektrum. Automotive Services bieten ökonomisch betrachtet neben dem individuell für den Kunden gesteigerten Komfort auch das Potenzial für gänzlich neue Geschäftsmodelle und Wertschöpfungsketten.

Um das Feld der Automotive Services hinsichtlich einer Unterstützung des Entwicklungs- und Gestaltungsprozesses besser einordnen zu können, werden im nachfolgenden Abschnitt die existierenden Grundlagen dieser Domäne erläutert sowie wesentliche Innovationstreiber

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

diskutiert. Darüber hinaus werden aktuellen Entwicklungen einiger großer Automobilhersteller sowie Zulieferer betrachtet.

2.1 Grundlagen

Automotive Services als mobile Mehrwertdienste im Fahrzeug sind bereits seit längerem bekannt und werden auch in der Serie in vielfältiger Form eingesetzt. Ausgehend von den direkt fahrzeugbezogenen Fahrerassistenzsystemen haben sich darüber hinaus Infotainmentsysteme in modernenn Fahrzeugen etabliert. Neben diesen rein im Fahrzeug befindlichen Ansätzen fasst das Feld der Car2X Systeme die Ausweitung hinsichtlich einer Einbindung externer Systeme und Services zusammen. Die nachfolgenden Abschnitte geben einen Überblick über diese drei Aspekte von Automotive Services.

2.1.1 Fahrerassistenzsysteme

Fahrerassistenzsysteme sind (teil-)autonome Systeme im Kraftfahrzeug, deren Aufgabe es ist, den Fahrer bestmöglich zu unterstützen. Sie sollen die Fahrsicherheit erhöhen und den Fahrer entlasten (Baron/Swiecki/Chen 2006). Dies geschieht entweder durch Versorgen des Fahrers mit kontextsensitiven Informationen, die für das Betreiben des Automobils relevant sind, beispielsweise durch die Ausgabe von akustischen, visuellen oder haptischen Warnungen in Gefahrensituationen oder durch eigenständiges Eingreifen ins Fahrverhalten. Hinsichtlich des Kontexts bezieht sich ein Fahrerassistenzsystem auf fahrzeuginterne Sensorik, auf die Verkehrsumwelt und auf den Fahrer selbst (bspw. Erkennung von Aufmerksamkeitsdefiziten und Schlaf). Die Erfassung des Kontexts erfolgt unter anderem über Techniken wie elektromagnetische Funkwellen (Radar), Licht (Lidar - Light detection and ranging), Videoaufnahmen und Ultraschall. Zu den gängige Fahrerassistenzsysteme zählen Antiblockiersystem (ABS), Elektronisches Stabilitätsprogramm (ESP), Adaptive Cruise Control (ACC), Lane Departure Warning (LDW) und Traffic Sign Recognition (TSR) (Meroth/Tolg 2007a).

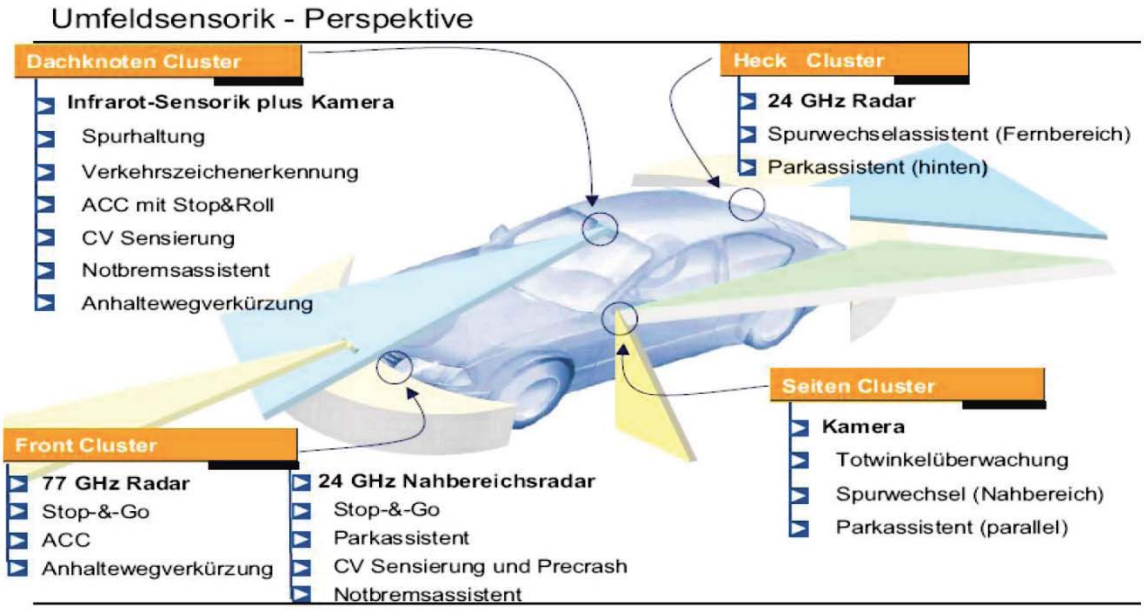


Abbildung 2-2 Sensornutzung für Fahrerassistenzsysteme
Quelle: (Niefünd 2004)

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Darüber hinaus können Fahrerassistenzsysteme den Fahrer auch in der Zielfindung unterstützen. Navigationssysteme errechnen mittels GPS optimale Routen und weisen den Weg zu potenziellen Orten von Interesse. Ergänzt werden solche Systeme durch die Verarbeitung von Live-Informationen zur aktuellen Verkehrslage, um so beispielsweise die Umgehung von Staus zu ermöglichen. Hier zeigen sich Überschneidungen mit den im nächsten Abschnitt diskutierten Infotainmentsystemen.

Systeme beider Kernbereiche des Infotainments lassen sich daher gemeinsam klassifizieren. Hoffmann (2010) zitiert Wandke, Wetzenstein und Polkehn (2005) für ein Klassifikationsschema nach sechs Phasen der Handlung von Mensch-Maschine-Interaktionen. Phase eins umfasst die Herstellung und Beibehaltung des Fahreraktivierungsniveaus und die Zielbildung. Wichtig sind die volle Aufmerksamkeit des Fahrers und ein Bewusstsein für das von ihm verfolgte Ziel. In Phase zwei werden Informationen des Fahrzeugs aufgenommen und dem Fahrer präsentiert. Dies kann redundant erfolgen (z.B. Einblendung von Verkehrszeichen) oder mit verändertem Modus (z.B. Pfeildarstellung und akustische Anweisungen des Navigationssystems). Die Informationen werden in der Folgephase verarbeitet. Hoffmann (2010) führt aus: „Im Fahrzeug kann eine Unterstützung dieser Phase zum Beispiel durch einen Demo-Modus erfolgen, bei welchem dem Fahrer die Funktionalität eines Systems oder seine Handlungsoptionen erklärt werden.“ Verarbeitete Informationen dienen als Grundlage für die Entscheidungsfindung in der vierten Phase. Assistenzsysteme geben hier Vorschläge und handeln je nach System bei impliziter oder expliziter Zustimmung durch den Fahrer in Phase fünf. In der finalen Phase erfolgt die abschließende Bewertung der gewählten Aktion. Diese Phase kann nur bedingt durch Assistenzsysteme unterstützt werden, da der Grad der Zielerreichung von bisherigen Systemen nicht erfasst wird.

2.1.2 Infotainmentsysteme

75 Jahre nach der Einführung des ersten Autoradios haben Infotainmentsysteme im Automobil einen wertbestimmenden Rang erobert (Meroth/Tolg 2007a). Das Kunstwort Infotainment ist ein Anglizismus, der sich aus den Wörtern Information und Entertainment (Unterhaltung) zusammensetzt. Entsprechend vereint der Begriff komfortsteigernde wie – im pragmatischen Sinne – informationsbringende Elemente. Die Mindmap in Abbildung 2-3 zeigt die vier Kernbereiche des Infotainments, die die wesentlichen Elemente des Konzepts prägen.

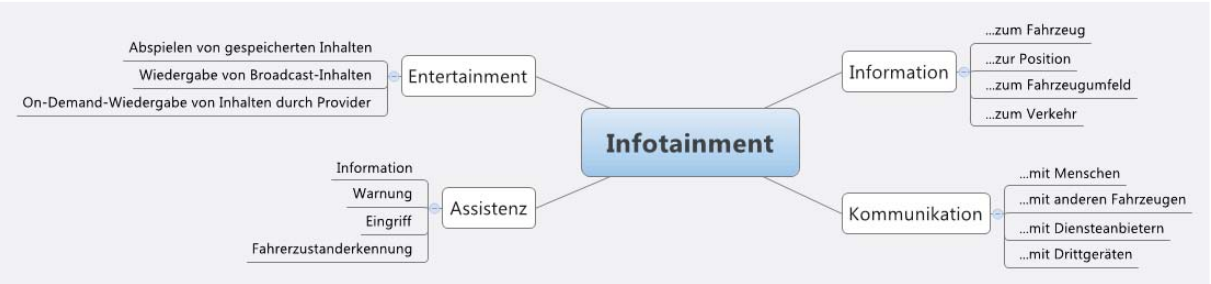


Abbildung 2-3 Kernbereiche des Infotainment
Quelle: (Meroth/Tolg 2007a)

Speziell in den 90er-Jahren avancierte der Entertainmentbereich zu einer wesentlichen Komponente der technischen Unterstützung im Fahrzeug. Während werkseitig über viele Jahre hinweg ein schlichtes, analoges Radio mit Kassettendeck als einziges Entertainmentgerät verbaut wurde, so ging mit der zunehmenden Datendigitalisierung auch ein steigender Anspruch des Kunden in diesem Bereich einher. Audio-CDs durchdrangen den

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Markt und wurden von MP3-Playern abgelöst. Unterhaltung in rein akustischer Form wurde durch visuelle Funktionalitäten ergänzt: DVD-Player fanden ihren Weg ins Fahrzeug, ebenso Fernsehen (DVB-T) und in den letzten Jahren auch Audio- und Video-Webstreaming über mobile Internetzugänge. Sogar Videospiele sind heute Teil des automotiven Entertainmentbereichs.

Entwicklungen des frühen 21. Jahrhunderts im Bereich der mobilen Datenübertragung ermöglichen die Integration des Kommunikationsbereichs in Fahrzeugcomputer. Telefonie erfolgt zwar auch heute noch meist über externe Mobiltelefone, doch Techniken wie Bluetooth erlauben die Kopplung beider Systeme. Durch die angesprochenen Entwicklungen neuer Techniken für mobilen Internetzugang lassen sich schnellere und stabilere Verbindungen realisieren, die damit auch die Nutzung von E-Mail-Diensten im Fahrzeug gestatten. Viele weitere Dienste lassen sich unter dem Schlagwort Car2X-Kommunikation vereinen.

Die Bereiche Information und Fahrerassistenz sind eng miteinander verwoben, da die bereitgestellten Informationen für die automatisierte Unterstützung des Fahrers weiterverwendet werden. Daten von externer Quelle wie Verkehrsinformationen (Traffic Message Control) und Geodaten (GPS) aber auch interne Daten wie Sensorauslesungen des Fahrzeugs bilden die Grundlage für Eingriffe des Fahrzeugcomputers in das Fahrverhalten des Fahrzeugs und ebenso für die Versorgung des Fahrers mit erweiterten Benachrichtigungen. Das Erfassen, Übermitteln und Verwerten von Verkehrsinformationen wird in der Literatur meist im Begriff Telematik zusammengefasst (Müller-Bagehl/Endt 2004).

Die Realisierung der Benutzerschnittstelle von Infotainmentsystemen ist eine der kritischsten und schwierigsten Aufgaben für Ingenieure: Interaktionen mit dem Fahrer müssen intuitiv und schnell von Statten gehen. Die Aufmerksamkeit des Fahrers muss stets dem Verkehr gelten, nicht dem System. In Abschnitt 3.5 wird auf diesen Aspekt im Rahmen der Analyse der Infotainmentsysteme von BMW, Audi und Mercedes-Benz genauer eingegangen.

2.1.3 Car2X-Kommunikation

Einen aktuellen Trend in der Entwicklung automobiler Dienste stellt die sogenannte Car-to-X-Kommunikation dar, kurz Car2X oder englisch V2X für Vehicle to X. Dieses Konzept umschreibt den mobilen Austausch von Informationen zwischen Fahrzeugen und externen Anlaufstellen. Je nach Art der letzteren, unterscheidet man hierbei zwischen mehreren Ausprägungen: Die Kommunikation zwischen je zwei Fahrzeugen wird als Car2Car-Kommunikation bezeichnet. Dem gegenüber steht der Informationsaustausch zwischen Fahrzeug und fest installierten Verkehrsinfrastrukturen wie z.B. Ampeln oder Stauschildern. Seltener sind in Literatur und Praxis die Varianten Car2Home und Car2Enterprise zu finden (Volkswagen AG 2010a).

Grundlage für Car2X-Kommunikation stellen ad-hoc-Netzwerke dar, die durch ihren spontanen Aufbau eine Vermittlung von Daten während der Fahrt erlauben. Diese Netzwerke basieren unter anderem auf den technischen Standards IEEE 802.11 (WLAN), UMTS und GPRS. In Anlehnung an das C2C-CC Manifesto (CAR 2 CAR Communication Consortium 2007) definiert Busch (2007) die Verbesserung der vier folgenden Punkte als wesentliche Ziele und Potenziale von Car2X:

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *Verkehrssicherheit:* Potenzielle Gefahren wie Stauenden oder Unfallstellen sollen an Autos im betroffenen Umfeld propagiert werden. Der Fahrer wird gewarnt und kann die entsprechende Vorsicht walten lassen.
- *Verkehrseffizienz:* Durch die Weiterleitung von Informationen über Orte mit hoher Verkehrsdichte können eben diese umfahren werden. So soll ein konstanter Verkehrsfluss gewährleistet werden. Häufig frequentierte Strecken werden entlastet.
- *Umweltverträglichkeit:* Eine Reduzierung von Wartezeiten durch die Vermeidung von Orten mit zähem Verkehrsfluss steigert nicht nur Sicherheit und Effizienz, sondern schont auch die Natur.
- *Komfort:* Durch Benachrichtigung des Fahrers über mögliche Orte von Interesse (Points of Interest) auf der Strecke kann der persönliche Nutzen für Reisende eklatant gesteigert werden. Gleiches gilt für die entgegengesetzte Richtung des Informationsflusses: Im Bedarfsfall können Informationen aus dem Fahrzeug (z.B. an eine Werkstatt) übertragen werden.

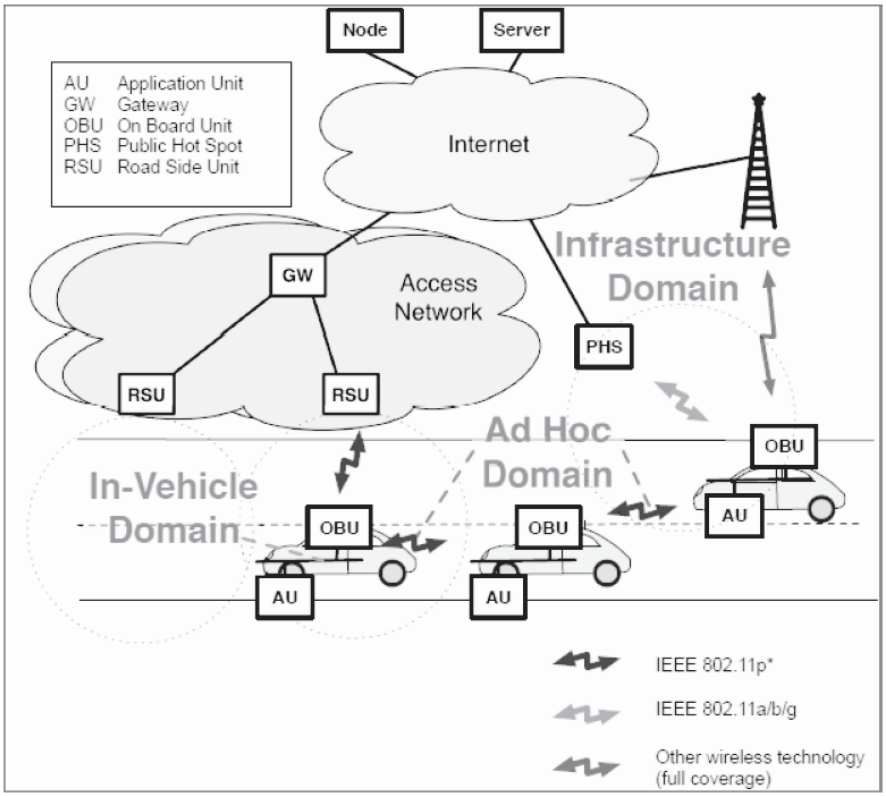


Abbildung 2-4 Szenario Car2X Kommunikation
Quelle: (Festag et al. 2008)

Es existieren jedoch auch einige weitere Herausforderungen bei der Einführung und dem Einsatz von Car2X-Systemen. Die technischen Anforderungen an geeignete Netzwerke sind hoch. So bleiben durch die hohe Geschwindigkeit fahrender Fahrzeuge nur Sekunden zum Aufbau der Verbindung und zur Übertragung der Daten. Netzwerke sind kurzlebig und instabil. Durch die Änderung des Verkehrsaufkommens und der Umgebung unterliegt die Netzwerk-Topologie ständiger Änderung. Überlastung der Kanäle und Kollisionen müssen auf technischer Seite unterbunden werden. Auch die Korrektheit der übertragenen Daten ist von außerordentlicher Wichtigkeit. Darüber hinaus treten organisatorische Erschwernisse in Erscheinung: Hersteller- und länderübergreifende Standards müssen gefunden werden. Zudem erschwert die anfänglich geringe Penetrationsrate einen sinnvollen Einsatz, da ein Nutzen erst ab einer gewissen Deckungsquote gegeben ist.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

2.2 Innovationstreiber

Es zeigt sich bei der Betrachtung von Automotive Services jedoch schnell, dass diese nicht ausschließlich von den Entwicklungen, Trends und Grundlagen der Automobilbranche abhängig sind, sondern durch eine Vielzahl externen Innovationsreiber vorangetrieben und beeinflusst werden. So wird das Thema der Informationsbereitstellung und Unterhaltung der Nutzer in den letzten Jahren durch die Entwicklungen im Bereich der Web-Technologien stark beeinflusst. Gleiches gilt auch für die zunehmende Bedeutung und Marktdurchdringung von Smartphones, welche mit einer zunehmenden und kosteneffizienten Verfügbarkeit von mobilen Breitbandverbindungen einhergeht. Letztere ist insbesondere für die Gestaltung und Umsetzung von Car2X Absätzen von zentraler Bedeutung.

Neben diesen allgemeinen Innovationstreibern kommt jedoch auch aus der Automobilbranche selber ein wesentlicher Innovationstreiber für Automotive Services. Im Zuge der aktuellen Klimadiskussion ist die Bedeutung von Elektromobilität in den Fokus der technologischen Diskussion gerückt. Jedoch existierend hier einige wesentliche Limitationen, die einen Wechsel von herkömmlichen Antriebskonzepten hin zu Elektromobilität erschweren. Automotive Services können hier ein entscheidender Faktor für eine erfolgreiche Nutzung sein. Schließlich ermöglichen Automotive Services darüber hinaus auch innovative Nutzungs- und Geschäftsmodelle.

Der nachfolgende Abschnitt gibt einen Überblick über wesentliche Innovationstreiber für Automotive Services. Ausgehend von den im letzten Abschnitt betrachteten Grundlagen werden die Innovationstreiber Web 2.0, Smartphones, mobile Breitbandverbindungen, Elektromobilität sowie innovative Nutzungs- und Geschäftsmodelle betrachtet.

2.2.1 Web 2.0

Wie vielen Schlagworten im Bereich der IT fehlt auch dem Begriff Web 2.0 eine genaue Definition. Evident ist der progressive Charakter des Begriffs, der eine neue evolutionäre Stufe des World Wide Web impliziert. Er wurde maßgeblich von Tim O'Reilly geprägt (Hass/Walsh/Kilian 2007), welcher Web 2.0 folgendermaßen umschreibt: „Like many important concepts, Web 2.0 doesn't have a hard boundary, but rather, a gravitational core. You can visualize Web 2.0 as a set of principles and practices that tie together a veritable solar system of sites that demonstrate some or all of those principles, at a varying distance from that core" (O'Reilly 2005). Betrachtet man den aggregierenden Charakter in dieser Definition, so präsentiert sich Web 2.0 als ein Kontext, der kollaborative Informationsdienste bündelt (Peters 2009). Daher erfolgt eine Visualisierung, wie in Abbildung 2-5 zu sehen, des Konzepts meist als Meme-Map, also in Form einer grafischen Vernetzung von inhaltlich verbundenen Schlagworten und Phrasen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

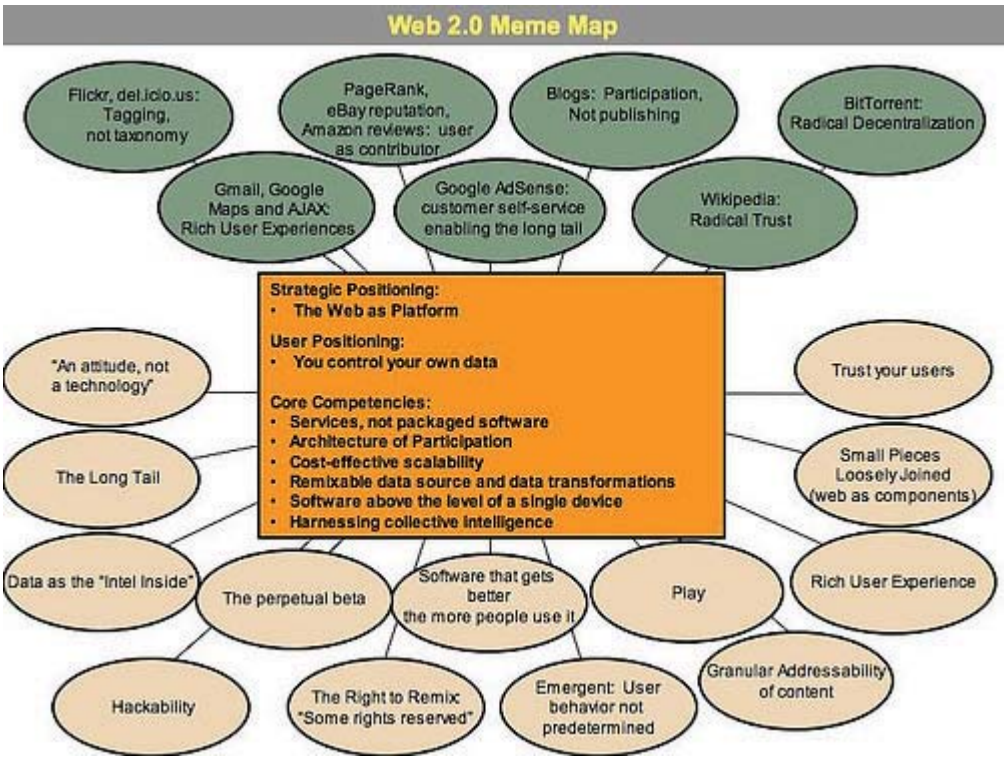


Abbildung 2-5 Web2.0 Meme Map
Quelle: (O'Reilly 2005)

Zentrale Eigenschaften des Web-2.0-Konzepts sind nach Koch und Richter (2007):

- *Dienste statt Software im Paket*[2]*:* Den Kern einer Web-2.0-Anwendung stellt der eigentlich bereitzustellende Dienst dar und nicht die spezifische Implementierung zur Bereitstellung derselben. Daher verfügen viele Web 2.0-Dienste über offene Schnittstellen. Dies ist auch vor dem Hintergrund neuer Nutzungsformen, bspw. mit mobilen Endgeräten zu sehen (Hass/Walsh/Kilian 2007).
- *Mischbare Datenquellen und Datentransformationen*[3]*:* Die Daten sind das wertvolle an der Anwendung. Sie sind in einer Weise bereitzustellen, die es ermöglicht, diese neu zu arrangieren und mit Daten aus anderen Quellen zu verbinden.
- *Eine Architektur der Beteiligung*[4]*:* Die Nutzer der Anwendung erstellen selbst deren Inhalt. Aus dem Konsument wird zugleich ein Produzent. Web 2.0 versteht sich als partizipatives Gebilde.

Es wird deutlich, dass es bei Web 2.0 weniger um Technologien geht (auch wenn bspw. Ajax und RSS oft mit Web 2.0 verbunden werden), als um Prinzipien grundsätzlicher Art (Mrkwicka/Kiessling/Kolbe 2009). Das wichtigste Prinzip und wohl der größte Unterschied zur vorherigen Generation des WWW stellt die Architektur der Beteiligung dar. Der Nutzer wird hierbei in den Mittelpunkt der Anwendung gerückt (Koch/Richter 2007). Obgleich sich durch Web 2.0 nicht zwangsweise neue Inhalte ergeben, erfolgt dennoch eine stärkere Betrachtung der Struktur der Informationsverknüpfung. Web 2.0-Applikationen können daher eher im Sinne von Social Software betrachtet werden, also als „internetbasierte

[2] Services, not packaged Software
[3] Remixable data sources and data transformations
[4] Architecture of Participation

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anwendungen, die Informations-, Identitäts- und Beziehungsmanagement in den (Teil-) Öffentlichkeiten hypertextueller und sozialer Netzwerke unterstützen“ (Koch/Richter 2007).

2.2.2 Smartphones

Eines der populärsten Schlagworte der Informations- und Kommunikationstechnik des 21. Jahrhunderts ist das Smartphone. Die begriffliche Herleitung lässt sich am besten historisch herbeiführen: In den 90er-Jahren des 20. Jahrhunderts durchdrangen zwei Klassen von mobilen Geräten den Markt. Während mobile Telefone, in Deutschland umgangssprachlich Handys, zu Zwecken der mobilen Kommunikation durch Telefon- und Kurznachrichtendienste genutzt wurden, so dienten Personal Digital Assistants (PDAs) vorwiegend dem persönlichen Informationsmanagement. Dieser, PIM abgekürzte, Begriff umfasst die Planung und Organisation von Terminen, Aufgaben und Adressen (Hansen/Neumann 2001). Ein wichtiger Aspekt ist dabei auch die Möglichkeit des Datenabgleichs zwischen mehreren Geräten (Synchronisation). Durch technischen Fortschritt im Bereich mobiler Datenübertragung wurde dieser Kern durch zusätzliche Funktionalitäten erweitert: So zählen heute auch Kommunikationsfunktionen wie elektronische Post (E-Mail) und Internetnutzung über einen Browser zu PIM. PDAs als mobile, miniaturisierte Computer zielten somit vorwiegend auf geschäftliche Nutzung ab, werden jedoch auch privat genutzt.

Die Vereinigung beider Geräteklassen lag nahe und resultierte im Smartphone, dem intelligenten Mobiltelefon. Gängige Abgrenzungsmerkmale von Smartphones gegenüber reinen Mobiltelefonen sind zumeist ein nicht-herstellerproprietäres Betriebssystem, eine vergleichsmäßig schnelle CPU, mehr Arbeitsspeicher, ein größerer Bildschirm, komfortablere Eingabemöglichkeiten über eine vollwertige QUERTZ-Tastatur und Datenschnittstellen wie WLAN und Bluetooth. Viele Geräte zeichnen sich zudem über Entertainmentfunktionen wie integrierte Kameras, Musik- und Videoabspielmöglichkeiten aus, welche jedoch nur bedingt dem Personal Information Management zuzuordnen sind.

Seit einigen Jahren warten viele Geräte auch mit Geolokalisationstechnik auf. Diese, vorrangig über GPS realisierte, Funktion ermöglicht dem Nutzer die Abfrage des aktuellen Standorts, etwa für standortbezogene Dienste wie Wetterprognosen, Navigationsdienste und umgebungsbasierten persönlichen Empfehlungen.

Smartphones bieten durch herstellerunabhängige Betriebssysteme und die Bereitstellung von Software Development Kits große Erweiterbarkeit, sodass ein Nutzer selbst über die Programme und damit Funktionen seines Geräts entscheiden kann (Maisch/Meckel 2009). Der gesteigerte, dynamische Funktionsumfang und die Flexibilität von Smartphones lassen die Grenzen zwischen Computern und mobilen Telefonen allmählich verschwinden. Smartphones bieten ubiquitären Zugang zum Internet und damit zu persönlichen Diensten, die Vernetzung erfordern, und ermöglichen die standortunabhängige und komfortable Verwaltung persönlicher Daten (Ballagas et al. 2006).

2.2.3 Mobile Breitbandverbindungen

Die Veränderung des World Wide Webs hinsichtlich Komplexität und damit Größe der Inhalte erforderte in den vergangenen Jahren nicht nur von leitungsgebundenen Breitbandverbindungen eine stetige Steigerung der realisierten Übertragungs-geschwindigkeiten. Mit der Einführung schnellerer mobiler Datenübertragungsdienste zur Jahrtausendwende wurde der Grundstein für die ortsunabhängige Nutzung webgebundener Dienste gelegt. Schnell wuchs jedoch auch hier der Bedarf nach großen Geschwindigkeiten.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Das 1997 eingeführte Wireless Application Protocol (WAP) gilt als erster Versuch, den mobilen Zugang zum Internet zu ermöglichen (Alby 2008, 21). Die langsame Geschwindigkeit und mangelnde Unterstützung verhinderten jedoch eine breite Akzeptanz von WAP. Erst die 2001 in Deutschland eingeführte General Packet Radio Services-Technik (GPRS) und der 2006 erstmals geschaltete Nachfolger EDGE (Enhanced Data Rates for GSM Evolution) konnten sich netz- und herstellerübergreifend durchsetzen. Bis heute (2010) gelten beide als gängige Standards für mobile Datenverbindungen und werden je nach Netz als Rückfalloption schnellerer Techniken eingesetzt.

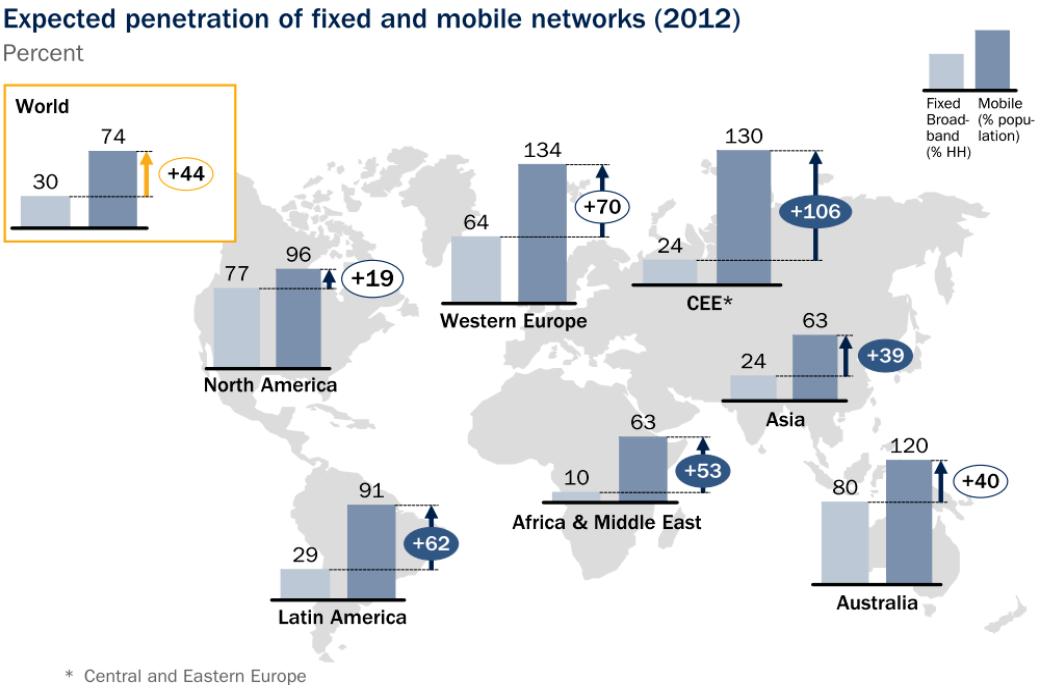


Abbildung 2-6 Ausbau leitungsgebundener und mobiler Breitbandanschlüsse
Quelle: (Buttkereit et al. 2009, 8)

2004 erfolgte schließlich die Einführung der dritten Generation (3G) der Mobilfunkübertragungstechnik, des Universal Mobile Telecommunications Systems (UMTS). Mit den Protokollerweiterungen HSxPA und HSxPA+ erfolgte schließlich nochmals ein großer Sprung hinsichtlich der theoretisch möglichen aber auch der im Alltag erreichten Übertragungsgeschwindigkeit, der diese auf das Niveau leitungsgebundener Techniken wie DSL hebt. WiMAX als konkurrierende Alternative zu HSxPA sei hier nur der Vollständigkeit halber erwähnt, konnte sich jedoch für kommerzielle Anwendungen nicht in der Breite durchsetzen. Seit Mitte 2010 befinden sich die nächste Generation der Long Term Evolution, kurz LTE, vor der Einführung.

Die volkswirtschaftlichen Auswirkungen der bevölkerungsdeckenden Penetration durch mobile Breitbandanschlüsse sind enormer Größe. Buttkereit et al. (2009) beziffern ein 10%iges Durchdringungswachstum in einem resultierend Wachstum des Bruttosozialprodukt von 0,1 bis 1,4%. Multiplikatoreffekte innerhalb der Industrie, aber auch indirekte, branchenübergreifende Effekte wie Produktivitätssteigerungen und Investitionsanreize werden gelistet. Im Rahmen des allgemeinen Enable-Align-Verhältnisses zwischen Informations- und Kommunikationstechnologie wirkt der Ausbau mobiler Breitbandverbindungen als Enabler: Neue Dienste können angeboten und neue Märkte erschlossen werden.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

2.2.4 Elektromobilität

Mit der Erfindung des Diesel- und des Ottomotors Ende des 19. Jahrhunderts begann das Zeitalter der Verbrennungsmotoren. Nach nun über 100 Jahren der intensiven, weltweiten Nutzung solcher Motoren, insbesondere im Verkehrswesen, wächst nun allerdings die Erkenntnis über die Abhängigkeit vom Öl und über die durch Emissionen verursachten Folgeschäden für Natur und Mensch. Unter dem Begriff Elektromobilität wird die Elektrifizierung der Antriebe als eine mögliche technologische Lösung für diese Probleme gesehen. Wie in Abbildung 2-7 dargestellt, würde die Abkehr von fossilen Brennstoffen hin zu erneuerbaren Energien im Verkehr keinen Bedarf wecken, der die prognostizierte Erzeugung übersteigt.

Das Bundesministerium für Wirtschaft und Technologie (2009, 8) nennt eine Reihe an Potenzialen von Elektromobilität:

- *Klimaschutz:* Elektromobilität kann einen wesentlichen Beitrag zur Verringerung der CO2-Emissionen im Verkehrssektor leisten.
- *Sicherung der Energieversorgung:* Fahren mit elektrischem Strom kann die Abhängigkeit vom Öl vermindern.
- *Ausbau des Technologie- und Industriestandorts:* Deutschland kann zum Leitmarkt für Elektromobilität werden und der deutschen Wirtschaft einen neuen Innovationsschub bringen.
- *Verringerung lokaler Emissionen:* Elektrofahrzeuge können die Städte von Schadstoffen, Feinstaub und Lärm befreien und so die Lebensqualität steigern.
- *Fahrzeuge in das Stromnetz integrieren:* Batteriefahrzeuge tragen zur Verbesserung der Effizienz der Netze bei und fördern den Ausbau der erneuerbaren Energien.
- *Neue Mobilität:* Elektrofahrzeuge können ein Baustein für intelligente und multimodale Mobilitätskonzepte der Zukunft sein.

In der Studie „Klimafreundliche Elektromobilität: Finanzielle Hürden zur Markteinführung bis 2020“ stellen Kortlüke und Pieprzyk die prognostizierte Energieerzeugung durch regenerative Energien dem prognostizierten Energiebedarf durch Elektromobilität gegenüber (Kortlüke/Pieprzyk 2010, 4).

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

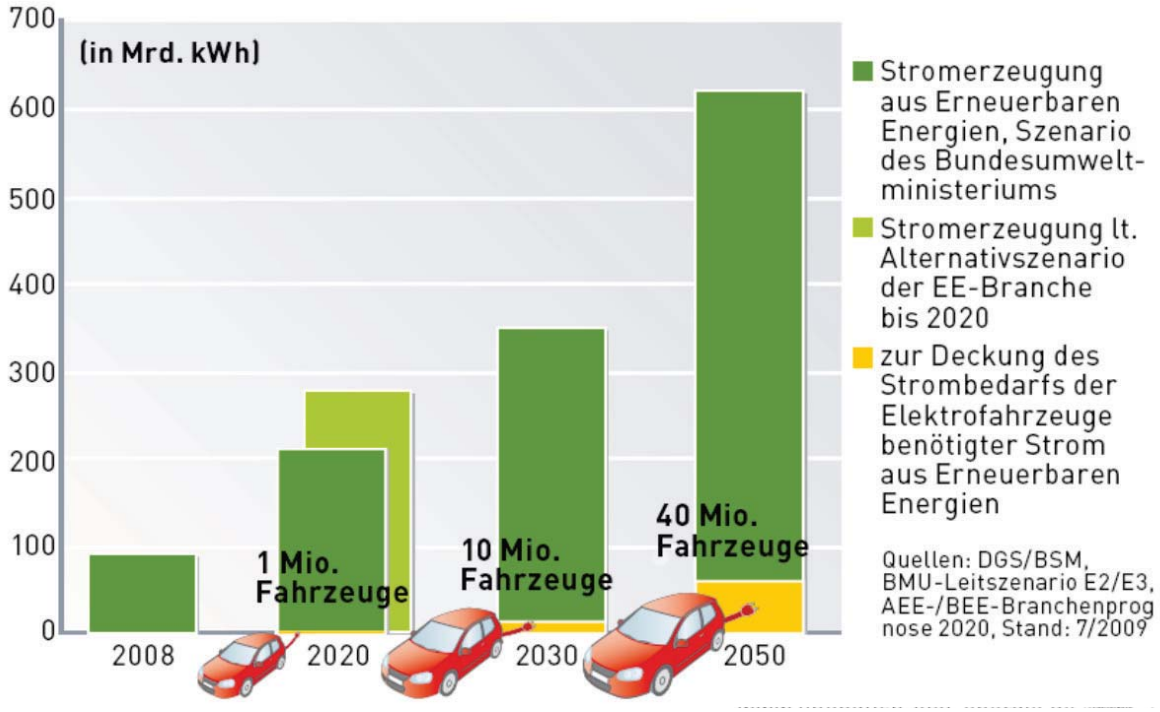


Abbildung 2-7 Energieerzeugung und Energiebedarf durch Elektromobilität
Quelle: (Kortlüke/Pieprzyk 2010, 4)

Die genannten Potenziale gehen jedoch auch mit Herausforderungen einher. Im Zentrum der Forschung steht hierbei die Frage nach der Energiebereitstellung: Es konkurrieren die Techniken der batteriebasierten Energiespeicherung und der brennstoffzellenbasierten Energiewandlung. Geeignete Batterien müssen nicht nur enorme Mengen an Energie speichern, sondern diese auch in möglichst kurzer Zeit aufnehmen können. Preis, Lebensdauer und realisierbare Fahrreichweite sprechen für den Einsatz von Brennstoffzellen. Für Batterien sprechen jedoch das geringere Gewicht und vor allem die bestehende Energieinfrastruktur.

2.2.5 Innovative Nutzungs- und Geschäftsmodelle

Technische Neuerungen hinsichtlich der Systeme im Automotive-Bereich und die verstärkte Öffnung der ehemals rein fahrzeugorientierten Systemgrenzen hin zu Mehrwertdiensten ermöglichen sowohl andersartige Ausprägungen bisheriger Nutzungs- und Geschäftsmodelle, wirken jedoch auch als Innovationstreiber und erlauben die Schaffung gänzlich neuer Modelle. Im Folgenden wird ein kurzer Überblick gegeben, der zeigt, welches ökonomische Potenzial hinter mobilen Diensten im Allgemeinen und Automotive Services im Speziellen steht.

Unter *Mobility Based Services* lassen sich Dienste zusammenfassen, deren Rückgrat mobile Datenübertragung bildet. Sie umfassen ein immenses Spektrum an möglichen Geschäftsmodellen, die jedoch alle auf die ubiquitäre Bereitstellung von Informationen für den Nutzer abzielen. Während Dienste dieses Bereichs über Notebooks und Smartphones schon vor einigen Jahren Einzug in den mobilen Alltag gehalten haben, so sind sie in Fahrzeugen bisher kaum vertreten. Wesentlicher Gesichtspunkt ist hier, wie bei allen Automotive Services, die angemessene Informationsbereitstellung über eine geeignete Mensch-Maschine-Schnittstelle.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Location Based Services beschreiben ortsabhängige Dienste, die den aktuellen Standort des Nutzers als Kontext und Basis für Dienstleistungen verwenden (Jacobsen 2004). Technische Grundlage bildet hier weitestgehend GPS, jedoch auch Datenübertragungsstandards, die eine Lokalisierung des Nutzers erlauben.

- *Navigation:* Einer der bereits heute gängigsten ortbasierten Dienste ist die Navigationsunterstützung über fortwährende Standorterfassung und davon abhängige Routenplanung. Verschiedenste Bezahlmodelle prägen dieses Marktsegment. Während beispielsweise Google mit Maps seit 2010 einen kostenlosen Dienst zur Verfügung stellt, bieten viele Hersteller regionalbeschränkt frei nutzbare Systeme zum Einwegpreis. Es werden jedoch auch Modelle angeboten, bei denen eine routenbasierte Bezahlung erfolgt, so ehemals von Nokia.
- *Verkehrsinformationen:* Ein weiteres Modell orientiert sich an Mehrwertdienstleistungen zu üblichen Navigationsunterstützungen. So lassen sich Zusatzdienste wie die Live-Ermittlung der aktuellen Verkehrslage auf der geplanten Strecke monetär vermarkten.
- *Selektive Informationsverbreitungs- und Anforderungsdienste:* Eine Schnittstelle zwischen allgemeinen mobilen Diensten und ortsbasierten Diensten bilden Services, die in Abhängigkeit vom Standort des Nutzers automatisch oder auf Anweisung des Nutzers Informationen anfordern oder diese verbreiten (Bodenstorfer/Hasenauer 2005). So können, gerade im Hinblick auf Car2X-Kommunikation, neue B2C-Leistungen definiert werden. Ein Beispiel für einen sowohl push- als auch pull-basierten Dienst im Automotive Sektor wäre beispielsweise das Auffinden einer nahegelegenen Werkstatt und das automatische Übertragen von Fahrzeuginformationen an diese. Im Gegensatz zur reinen Navigation liegt in diesem Bereich oft der Fokus auf Personalisierung. Die individuellen Vorlieben des Nutzers bilden eine wichtige Grundlage für ortsabhängige Dienste.
- *Werbung:* Eng verbunden mit dem letztgenannten Punkt sind ortsabhängige Werbungen, sogenannte Location Based Advertisements. Während rein push-orientierte Werbekonzepte kein Feedback durch den Nutzer ermöglichen, so erlaubt moderne Fahrzeugtechnik Pull-Konzepte, die eine individuelle Bedürfniserfassung garantieren.

2.3 Aktuelle Aktivitäten und Entwicklungen

Angesichts der steigenden wirtschaftlichen Bedeutung von Automotive Services für den Erfolg der Automobilindustrie ist es nicht weiter verwunderlich, das sich inzwischen nahezu alle Hersteller sowie auch deren Zulieferer in den letzten Jahren eingehend mit dieser Problematik befasst haben. Im folgenden Abschnitt werden einige ausgewählte Beispiele der Aktivitäten sowohl der Hersteller als auch externe Dritthersteller dargestellt. Die meisten der vorgestellten Automotive Services sind zum Zeitpunkt der Erstellung dieser Arbeit bereits am Markt verfügbar oder stehen kurz vor deren Markteinführung.

Die vorgestellten Serviceangebote wurden im Rahmen einer online Recherche, die sowohl die Angebote der Hersteller als auch die von Nachrichtenportalen beinhaltet, zusammengetragen. Aufgrund der Aktualität des dargestellten Themas kann dieser Abschnitt lediglich eine Momentaufnahme der Marktsituation liefern und ist sicherlich nicht als vollständig und erschöpfend anzusehen. Nichts desto trotz vermitteln die dargestellten Aktivitäten einen umfassenden Einblick in die aktuelle Marktsituation. Diese liefert im weiteren Verlauf der

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Arbeit ein grundlegendes Verständnis des Umfangs und der Funktionsweise von Automotive Services.

2.3.1 Hersteller

Im Rahmen der Recherche wurden sieben Automobilhersteller sowie fünf Dritthersteller und Zulieferern untersucht. Unter den Automobilhersteller sind die Firmen Audi, BMW, Ford, Daimler, Nissan, Toyota, und Volkswagen vertreten.

2.3.1.1 Audi

Die Audi AG hat für ihre aktuelle A8 Modellreihe die „Audi Online Dienste“ vorgestellt (Audi AG 2010b). Dieser Service stellt eine Erweiterung der bereits im Audi Multi-Media-Interface verfügbaren Offlinefunktionalitäten um zusätzliche Onlinefunktionalitäten dar. Technologisch basieren die Audi Online Dienste auf einem webgestützten Portal (Audi AG 2010a). Unter den neuen Funktionen befindet sich unter anderem eine online Suche nach Sonderzielen, eine Wettervorhersage sowie ein Bereich für aktuelle Nachrichten.

Sonderziele können über eine Freitextsuche im Fahrzeug eingegeben und anschließend über das Angebot des Suchmaschinenanbieters Google recherchiert werden. Gefundene Ergebnisse sind direkt mit den existierenden Funktionalitäten des Multi-Media-Interface verknüpft, so dass beispielsweise Sonderziele in der Navigation verwendet werden können. Die Wettervorhersage ist eine browserbasierte Anwendung, die Nutzer mit aktuellen Wetterinformationen versorgt. Die Daten werden immer aktuell über das Internet verfügbar gemacht und bieten zusätzliche Funktionalitäten wie beispielsweise ein Wetterradar. Einen weiteren Onlinedienst bietet Audi in Form von aktuellen Nachrichten an. Diese werden in Form eines RSS-Feed wiederum direkt im Browser dargestellt und versorgen so den Fahrer stets mit aktuellen Informationen.

Abbildung 2-8 Audi Online Wetterinformationen
Quelle: (Audi AG 2010b)

Als Voraussetzung, um die Audi Online Dienste nutzen zu können, setzt der Hersteller das Vorhandensein des so genannten "optionale Bluetooth Autotelefon online“ voraus. Für den Zugang zum Internet wird eine handelsübliche Sinnkarte verwendet. Somit ist das Angebot

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

von Audi nicht auf einen speziellen Mobilfunkanbieter begrenzt. Die Datenübertragung erfolgt aktuell über ein GPRS/EDGE Modul.

Zusätzlich zu den Onlineservices im Fahrzeug bietet Audi noch das Onlineportal myAudi an. Nutzer können hier unterschiedlicher personalisierte Services nutzen, wie beispielsweise Bilder oder Filme online stellen, um diese später zu verwenden. Des Weiteren bietet myAudi die Möglichkeit das eigene Fahrzeug anzulegen, um so personalisierte Onlinedienste wie Navigation und Adressdaten zu nutzen. Beispielsweise kann am heimischen PC via Google Maps eine Route oder ein Ziel erstellt werden das dann im Rahmen der Nutzung der Audi Online Dienste im Fahrzeug zur Verfügung steht.

2.3.1.2 BMW

Mit dem Dienstkonzept ConnectedDrive bietet BMW seinen Kunden einen persönlichen in das Fahrzeug integrierten Assistenten an (BMW Group 2010a). Im Kern besteht ConnectedDrive aus drei Diensten: BMW Assist, BMW Online und BMW TeleServices. Ebenfalls integriert BMW mit der Einführung des aktuellen 7er eine uneingeschränkte Internetnutzung im Fahrzeug.

BMW Assist ist eine Sammlung von Angeboten, im Rahmen derer Mitarbeiter den Kunden in unterschiedlichen Bereichen unterstützen. In erster Linie handelt es sich bei BMW Assist um einen Auskunftsdienst, der über das im Fahrzeug verfügbar Telefon funktioniert. Der Kunde kann sich beispielsweise beim Auffinden freier Parkhäuser, Geldautomaten, Hotel- und Restaurantreservierungen, der Address- und Kontaktvermittlungen oder der Fernverriegelung und Fernentriegelung des Fahrzeugs unterstützen lassen. Des Weiteren bietet BMW Assist aktuelle Verkehrsinformationen sowie eine Wettervorhersage. Im Falle eines Unfalls oder einer unvorhergesehenen Notlage kann das System einen automatischen Notruf übermitteln.

Abbildung 2-9 BMW Online Wetterinformationen
Quelle: (BMW Group 2010b)

Im Rahmen von BMW Online kann der Kunde standortbezogene Dienste in einem mobilen Internet Portals nutzen. Analog zum Nachrichtenangebot von Audi kommen RSS-Feed zum Einsatz, die individualisierte Informationen zu den Themen Deutschland, Welt, Wirtschaft, Sport und Unterhaltung anbieten. Zusätzlich können Kurzbeschreibungen und Informationen zu Hotels und Restaurants oder zu Sehenswürdigkeiten in der näheren Umgebung angezeigt werden. Schließlich umfasst BMW online auch Informationen zum Kulturprogramm der näheren Umgebung und bietet dem Kunden ein mobiles Büro mit integrierter E-Mail-Unterstützung (POP3/IMAP).

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Mit den BMW TeleServices unterstützt die Hersteller im Rahmen von ConnectedDrive den Kunden bei Wartung und Service des Fahrzeugs. Statusinformationen zu Wartungs- und Servicebedarf, die der Bordcomputer des Fahrzeugs ermittelt, können autonom und frühzeitig eine entsprechende BMW Service Partner übermittelt werden. Diese haben dann die Möglichkeit, absehbaren Reparaturbedarf des Fahrzeugs frühzeitig zu erkennen oder auch den Kunden auf einen anstehenden Servicetermin aufmerksam zu machen.

Auf der IAA 2009 wurde mit dem Concept Application Store ein weiterer Baustein von ConnectedDrive präsentiert. Beim Concept Application Store handelt es sich, analog zum iTunes Store von Apple, um einen Online Shop mit dem sich der Kunde neue Dienste nachträglich ins Fahrzeug integrieren kann. Als Beispiele hat der Hersteller Facebook, Twitter und Xing präsentiert. Der Concept Application Store ist allerdings bislang nicht in Serie verfügbar.

2.3.1.3 Ford

Seit 2008 bietet der Hersteller Ford in einigen seiner Fahrzeugreihen den Service Ford Sync an. Einerseits besteht Ford Sync aus fest in das Fahrzeug integrierten Diensten, andererseits wird großer Wert auf eine Integration externer Geräte wie MP3-Player oder Mobiltelefone gelegt. Technologisch setzt Ford stark auf den Bluetooth Standard zur Anbindung externen Geräte sowie auf eine möglichst durchgängige Steuerung via Sprachkommandos.

Der Kern des Ford Sync Konzepts ist die nahtlose Integration externer Geräte in das Fahrzeug. Diese werden einerseits über Sprachkommandos gesteuert andererseits aber auch über spezielle Funktionstasten die sich am Lenkrad oder der Bordkonsole befinden bedient. Über ein integriertes GPS Modul bietet Ford Sync die üblichen Navigationsfunktionalitäten an. Zusätzlich können über das Sprachkommando „Traffic“ aktuelle Verkehrsinformationen angefragt werden, die das System über die Lautsprecher des Fahrzeugs als Sprachausgabe ausgibt.

Die Integration gängiger Mobilfunktelefonen erfolgt wie bereits erwähnt drahtlos über Bluetooth. Übliche Funktionen wie das Annehmen und Tätigen von Telefonaten, das Durchstöbern der Kontakte oder das Versenden von SMS kann über Sprachsteuerung, das Lenkrad oder die Bordkonsole erfolgen. Beim Vorlesen von beispielsweise SMS Nachrichten erfolgen angepasste Optimierungen, so dass beispielsweise typische Abkürzungen wie „LOL“ erkannt werden. An einem Ford Sync System können gleichzeitig bis zu 12 Mobiltelefone angemeldet sein.

Abbildung 2-10 Ford Sync
Quelle: (Ford 2010)

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Die Integration von angeschlossenen MP3-Playern erfolgt direkt in das In-Car Audiosystem. Per Sprachbefehl kann das Abspielen der Musik gesteuert werden, wobei Informationen wie Titel oder Interpret im Fahrzeug angezeigt werden. Zusätzlich kann per Sprachbefehl nach ähnlicher Musik zu aktuell abgespielten Titeln gesucht werden. Über Musik hinaus kann das System auch Podcasts oder Audiobooks abspielen.

Die fest in das Fahrzeug integrierten Services, die Sync dem Fahrer zur Verfügung stellt, können auch ohne angeschlossenes Mobiltelefon genutzt werden. Hierzu zählen Informationssysteme aus den Bereichen Nachrichten, Sport und Wetter.

Für 2011 hat Ford eine Erweiterung von Ford Sync angekündigt (AppSync) mit der es möglich sein wird, speziell angepasste Apps, die der Nutzer auf seinem Smartphones hat, über das Fahrzeug zu steuern. In einer ersten Stufe sollen Anwendungen für die Smartphones Android und Blackberry unterstützt werden.

2.3.1.4 Daimler

Als erster Hersteller bietet Daimler für seine SMART-Fahrzeuge eine spezielle Integration des Apple iPhone an (Daimler AG 2010a). Das Smartphone kann über eine, mit der Bordelektronik vernetzte, Halterung mit dem Fahrzeug verbunden werden (Fürtsch 2010). Das besondere an diese Integration ist, dass nicht nur die üblichen Funktionen wie beispielsweise das Abspielen von Musik oder das Annehmen und Tätigen von Telefonaten über diese Integration möglich wird. Über das iPhone können zusätzliche Funktionen, wie ein Car-Finder und ein Navigationssystem aktiviert werden. In der Zukunft ist geplant, analog dem AppSync von Ford, weitere Apps über diese Schnittstelle zu integrieren. So soll es beispielsweise eine zusätzliche externe Kamera im Fahrzeug geben, mit deren Hilfe Geschwindigkeitlimits erkannt werden können um diese auf dem iPhone anzuzeigen.

Abbildung 2-11 SMART iPhone Integration
Quelle: (Fürtsch 2010)

Neben der Integration in das Fahrzeug nutzt ebenfalls die Mercedes-Benz Bank das iPhone um ihren Kunden Services rund um das Fahrzeug anzubieten. Mit der Mercedes-Benz Bank Sternhelfer App erhält der Fahrer schnelle Hilfe im Falle eines Unfalls oder eines Servicefalls (Mercedes-Benz Bank 2010). Neben Erste-Hilfe Informationen und Notruffunktionen besteht die Möglichkeit, Unfallschäden direkt mit dem iPhone aufzunehmen und via E-Mail an die

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Versicherung zu senden. Dabei werden Funktionalitäten wie GPS Ortung und die in das iPhone integrierte Kamera genutzt.

2.3.1.5 Nissan

Der Hersteller Nissan beschäftigt sich mit einer ganzen Reihe an innovativen Automotive Services die sowohl Hardwarekomponenten im Fahrzeug als auch spezialisierte Softwarekomponenten umfassen (Nissan 2010).

Das „Robotic Interface“ analysiert die Gefühlslage des Fahrers durch eine Beobachtung und Auswertung von Sprechweisen und Gesten. Mithilfe dieser Informationen versucht das System positiv auf den Fahrer einzuwirken, so dass eine glücklichere Gefühlslage zu besserem und sichererem Fahrverhalten führen soll.

Abbildung 2-12 Nissan Robotic Interface
Quelle: (Nissan 2010)

Im Kontext der aufkommenden Elektromobilität hat Nissan einen sowohl im Fahrzeug als auch außerhalb des Fahrzeugs nutzbaren Dienst für das Großserienelektrofahrzeug Nissan Leaf entwickelt (Gebhardt 2009). Mithilfe eines Smartphones kann auch außerhalb des Fahrzeugs der Ladezustand abgefragt und das Laden gesteuert werden. Auch per SMS oder Website ist der Ladezustand abrufbar. Während der Fahrt berechnet das System ausgehend von der derzeitigen Reichweite Informationen, die dem Fahrer im Navigationsgerät angezeigt werden.

Das Projekt ECO Drive Support System soll den Fahrer zu sparsamen Fahren animieren. Das System sammelte dazu Daten über die Strecke und das Fahrverhalten welche in der Bordkonsole angezeigt werden. Basierend auf diesen Informationen gibt das System Tipps zu besserem Fahrverhalten, berechnet Trends und erstellt ein Online-Ranking, in dem sich Fahrer mit anderen Fahrern vergleichen können. Alle Informationen können vom Fahrer während der Fahrt aktuell abgelesen und verglichen werden.

Das Drunk-driving Prevention Concept Car befasst sich mit der Vermeidung von alkoholisiertem Fahren. Anhand einer Gesichtsanalyse, einer Überprüfung des Alkoholanteils in der Luft, in den Handflächen des Fahrers, oder auf den Schaltknüppel sowie durch eine Auswertung des Fahrverhaltens wird ein möglicher alkoholisierter Zustand des Fahrers ermittelt. Mittels einer Vielzahl an Warnhinweisen, wie beispielsweise Tonsignalen, schriftliche Warnhinweisen oder dem Festziehen des Fahrergurtes, soll der Fahrer auf seinen Zustand aufmerksam gemacht werden. Wenn Alkohol in der Handfläche auf dem Schaltknüppel nachgewiesen wird, kann dieser nicht mehr benutzt werden.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

2.3.1.6 Toyota

Im Rahmen des Projekts LTE Connected Car kooperiert der Hersteller Toyota zusammen mit den Unternehmen Alcatel-Lucent, ngConnect, Atlantic Records, HP, Samsung und QNX. Ziel des Projekts ist die Bereitstellung einer Automotive Services Plattform, analog der von Apple geschaffenen Anwendungsplattform für das iPhone. Interessant in diesem Projekt ist ein auf „Cloud"-Technologie[5] basierender Ansatz, Anwendungen nicht herunter zu laden und zu installieren sondern diese direkt in der Cloud auszuführen. Hierzu setzt Toyota auf die mobile Datenanbindung mittels LTE. Laut Dawson setzen die Unternehmen auf Prototypentests durch Endkunden, um den Entwicklungsprozess bestmöglich zu unterstützen. Im Kontext der grafischen Repräsentation der Dienste im Fahrzeug tritt die Firma Adobe mit ihrer Flash-Plattform als Technologiepartner auf (Dawson 2010; Alcatel Lucent 2010; Banfield-Seguin Ltd. 2009).

Mit dem Toyota Prius wird die LTE Connected Car Technologie erstmalig als optionales Ausstattungsmerkmale erhältlich sein (Smith 2009). Besonders hervorzuheben ist dabei, dass über ein, in das Fahrzeug integriertes, WLAN die LTE Datenverbindung an verbundene Endgeräte weitergereicht werden kann.

Abbildung 2-13 LTE Connected Drive
Quelle: (Alcatel Lucent 2010b)

Anwendungsbeispiele für die im Projekt LTE Connected Car angestrebten Automotive Services werden insbesondere Location Based Services wie zum Beispiel Restaurantführer und lokalisierte Werbung genannt. Ebenso sind Musik und Videodienste, ähnlich wie bei Apple iTunes, angestrebt. Ein besonders wichtiger Aspekt bei der Integration von Automotive Services ist der direkte Kontakt der Automobilhersteller zu ihren Kunden. Im Unterschied zur bisherigen Praxis kann dieser durch den Einsatz von Automotive Services zukünftig auch im After-Sales-Bereich deutlich verstärkt werden, wovon sich die Hersteller eine Verbesserung der Markentreue der Kunden erhoffen.

[5] „Unter Cloud Computing versteht man ein IT-basiertes Bereitstellungsmodell, bei dem Ressourcen sowohl in Form von Infrastruktur als auch Anwendungen und Daten als verteilter Dienst über das Internet durch einen oder mehrere Leistungserbringer bereitgestellt wird." (Krcmar 2010; Böhm et al. 2010)

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Im Rahmen des Projekts hat Alcatel Lucent eine Studie zur Vermarktung von Automotive Services durchgeführt. Als Hauptzielgruppe wurden Kunden unter 35 Jahren identifiziert, wobei anzumerken ist, dass die Studie nicht nur auf den demographischen Faktor, sondern auch auf die Zahlungsbereitschaft der Kunden in Abhängigkeit von Alter und Nationalität eingeht. Kunden werden in der Studie in fünf Kundensegmente unterteilt: Early Adopters, Initiators, Aspirers, Active Drivers und Business Travelers. Bezüglich der Zahlungsbereitschaft der Kunden wurde für den europäischen Raum die Bereitschaft für ein monatliches Pauschalmodell als für die Kunden akzeptabel identifiziert. Besonders sollten die Hersteller hier allerdings auf Aspekte des Datenschutzes sowie Sicherheitsaspekte eingehen.

2.3.1.7 Volkswagen

Der Volkswagenkonzern befasst sich mit dem Thema Automotive Services im Rahmen des in den USA ansässigen Electronic Research Laboratory (ERL). In dieser Forschungseinrichtung werden künftig Infotainmentsystemen für Fahrzeuge des Volkswagen Konzerns entwickelt. Mit der Global Open Research Infotainment Architecture (GLORIA) wird ein System entwickelt, das vor allem auf neuartige Eingabemethoden wie zum Beispiel Multitouch und Fingergestenbedienung setzt. Bislang ist dieses System in der Serie allerdings noch nicht verfügbar. Haupt Unterscheidungsmerkmal von Gloria im Vergleich zu anderen Herstellerlösungen wie Ford Sync ist die modulare upgradefähige Umgebung (Cunningham 2009)(Kaufmann 2009).

Abbildung 2-14 Volkswagen "GLORIA" 3D GPS
Quelle: (Kaufmann 2009)

Mit dem App-my-ride Idea Contest hat der Volkswagenkonzern 2010 einen Open Innovation Wettbewerb zum Thema Automotive Services durchgeführt (Volkswagen AG 2010b). Ausgestattet mit einem Auf Adobe Flash basierenden Prototypingwerkzeug, konnten interessierte Kunden auch ohne Programmiererfahrung selbst Prototypen von Services entwickeln und zum Wettbewerb ein senden. Unter den eingesetzten Automotive Services befinden sich unter anderem Dienste zur Kostenkontrolle, Förderung eines umweltfreundlichen Fahrstils oder einem ortsabhängigen, auf Wikipedia basierenden Dienst der Points of Interest aufzeigt. Mit diesem Wettbewerb hat Volkswagen erstmalig den Kunden direkt in die Ideenfindungphase von Automotive Services integriert.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

2.3.2 Zulieferer

Neben den Automobilherstellern wurden als Dritthersteller und Zulieferer die Angebote der Firmen Alpine, Continental, Apple, Ross-Tech, und TomTom betrachtet.

2.3.2.1 Alpine

Der Zulieferer Alpine ist ein japanisches Unternehmen, spezialisiert auf Autoradios, Navigationssysteme und In-Car-Media-Lösungen. Neben Produkten wie MP3 Radios, Lautsprechersystemen, Audioprozessoren und GPS Navigationssystemen bietet das Unternehmen zunehmend Produkte die eine Integration mobiler Endgeräte ermöglichen an (Alpine Electronics Inc. 2010a).

Die so genannten Digital Media Stations bzw. Mobile Media Stations von Alpine kombinieren Dienste wie Audio, Video und Navigationssysteme und stellen diese über Touch-Displays zur Verfügung. Viele dieser Geräte sind mit dem Apple iPod kompatibel, können also auf dessen Musikbibliothek zugreifen.

Abbildung 2-15 Alpine Digital Media Station
Quelle: (Alpine Electronics Inc. 2010b)

Speziell für die Integration des Apple iPod in das Fahrzeug ermöglicht Alpine mit dem 425i iPod Video Interface eine komplette Bedienung des iPods über einen eigenen Touch-Monitor im Fahrzeug (Alpine Electronics Inc. 2010c). Die Menüstruktur ist dabei der des iPods nachempfunden und es lassen sich Videos oder Podcasts abspielen.

Im März 2007 gab Alpine außerdem bekannt, mit Nokia und NAVTEQ zu kooperieren um eine Anbindung von Smartphones zu ermöglichen. So sollen Smartphones in Alpinegeräte integriert werden können, wie etwa Nokia‘s Navigationsservice Ovi. Die Technologie hierfür nennt sich Terminal Mode und stammt von Nokia. Laut Alpine und Nokia soll es so möglich werden, widget-basierte Services wie Wettervorhersage, Musik oder App Stores im Auto zu ermöglichen. Alpine geht noch weiter und stellt eine Verbindung von Smartphones und Fahrzeugdaten wie etwa Tankfüllungen in Aussicht. Dies ermöglicht eine Erstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

intelligenter fahrzeugzentrierter Dienste. Ein Veröffentlichungsdatum für erste Geräte mit Terminal Mode Unterstützung ist derzeit nicht bekannt (Blandford 2010).

2.3.2.2 Continental

Die Firma Continental verfolgt seit 2009 eine eigene Fahrzeug-Infotainment-Plattform namens AutoLinQ die gemeinsam mit der Telekom-Tochter T-Systems angeboten wird. Technologisch basiert AutoLinQ auf dem Betriebssystem Android von Google (Continental Automotive GmbH 2010). AutoLinQ ermöglicht eine ortsunabhängige Interaktion des Nutzers mit dem Fahrzeug. Continental definiert für diesen Zweck unterschiedliche Nutzungsszenarien, die so genannten Views: Car View, Mobile View, Home View, Partner View. Dem Nutzer werden im Rahmen der vier Views kontextrelevante Informationen zugänglich gemacht. So kann er beispielsweise von zuhause aus mit dem PC oder von unterwegs mit dem Mobiltelefon oder Smartphone auf Fahrzeugfunktionen zugreifen und mit diesem interagieren. Beziehen sich die drei Views Car View, Mobile View und Home View auf den jeweiligen Nutzungskontext, beschreibt der Partner View die Möglichkeit Dritter, eigene Anwendungen für die AutoLinQ-Plattform bereitzustellen.

Der Car View ist der in das Fahrzeug integrierte Hauptfunktionsbereich und stellt somit den zentralen View dar. Funktional werden den Nutzern unter anderem eine erweiterte Routenplanung und Navigation, Social Networking, Kommunikation (Telefonie, E-Mail, Zugang zu Medien), Anwendungen für sicheres und spritsparendes Fahren sowie eine Reihe individualisierter, herstellerspezifischer Anwendungen geboten.

Abbildung 2-16 Continental AutoLinQ
Quelle: (Continental Automotive GmbH 2010)

Mit Home View können Nutzer bereits im Voraus Mithilfe einer webbasierten Navigation ihre zukünftigen Fahrten planen. Durch den Einsatz von Traffic-Overlay-Funktionen, werden die Nutzer von aktuellen Fahrverkehrsproblemen gewarnt und über alternative Routen informieren. Auch können über den Home View Sicherheitseinstellungen festgelegt und überwacht werden.

Mit Mobile Video bezeichnet Continental die Möglichkeit des Nutzers über das Mobiltelefon mit dem Fahrzeug zu kommunizieren. Als Kommunikationsmedium werden SMS eingesetzt mit denen unter anderem folgende Funktionen im Fahrzeug gesteuert werden können. So ist es möglich den aktuellen Fahrzeugzustand und Standort abzurufen sowie kontextabhängig die Fahrzeugsicherheit einzustellen und zu kontrollieren. Darüber hinaus können Nachrichten versendet und eine in das Fahrzeug integrierte Suchfunktion genutzt werden.

2.3.2.3 Apple

Auch wenn Apple hier im Kontext der Zulieferer aufgelistet wird, tritt der Konzern nicht selbst als Anbieter von Infotainmentlösungen oder Automotive Services auf. Vielmehr hat

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Apple durch die Einführung des iPhone eine mobile Plattform geschaffen, die es Dritten ermöglicht, vom Fahrzeug losgelöst Automotive Services anzubieten. Natürlich ist das iPhone nicht das einzige Gerät das für solche Dienste als Plattform geeignet ist. Aufgrund seiner großen Marktrelevanz wird es hier dennoch als Beispiel für eine geeignete mobile Plattform angeführt. Weitere verfügbare Plattformen sind beispielsweise Smartphones wie das Blackberry oder auf Google Android basierende Geräte. Im Folgenden werden exemplarisch die Anwendungen Carticipate, TripAlyzer, Dynolicious, iBoost, Take me to my car und iHUDDisplay vorgestellt.

Die Anwendung Carticipate (Carticipate 2008) unterstützt den Nutzer beim Finden und Planen von Mitfahrgelegenheiten. Fahrer können potentielle Mitfahrer kurzfristig über passenden Mitfahrgelegenheiten informieren. Für diesen Zweck wird das eingebaute GPS zur Positionsbestimmung genutzt und es sowohl dem Fahrer wie auch den Mitfahrern deutlich erleichtert, passenden Mitfahrgelegenheiten, oder Mitfahrer zeitnah zu finden.

Mit TripAlyzer (Surich Technologies Inc. 2010), können Fahrdaten ohne direkte Integration der Anwendung in das Fahrzeug aufgezeichnet und berechnet werden. Dies ist beispielsweise für ältere Fahrzeuge, die noch keine erweiterte Funktionalität besitzen, eine interessante Option. Abhängig von der Position und Beschleunigung, bestimmt mittels GPS, und Eingabe der relevanten Fahrzeugdaten ermittelter TripAlyzer Informationen zu den aktuellen Benzinkosten, zur Effizienz, zum Beschleunigungsverhalten, zur Geschwindigkeitsverteilung und zum CO2 Ausstoß.

Abbildung 2-17 Fahrdatenbestimmung auf dem iPhone mit TripAlyzer
Quelle: (Surich Technologies Inc. 2010)

Dynolicious (BunsenTech LLC 2010) ist ähnlich wie TripAlyzer eine Anwendung zur Erfassung dynamischer Fahrdaten. Mithilfe der Messung der Geschwindigkeit und des Beschleunigungsverhaltens werden Informationen und Vergleichsanzeigen für den Nutzer generiert. Unter anderem kann so beispielsweise die Kräfteeinwirkung einer Bremsung gemessen und dem Nutzer dargestellt werden. Durch die Möglichkeit unterschiedlicher Fahrzeugprofile ist eine Nutzung in verschiedenen Fahrzeugen und somit ein Vergleich dieser Fahrzeuge einfach möglich.

Eine Anwendung die im Kontext von Automotive Services weniger einen funktionalen Mehrwert sondern vielmehr auf den Fahrspaß fokussiert ist die Anwendung iBoost (Bonobo

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

2010). Diese erfasst Beschleunigungs- und Schaltvorgänge und spielt, über das Infotainmentsystemen des Fahrzeugs, realistische Fahrgeräusche ab.

Gerade für den urbanen Bereich ist die Anwendung Take me to my car eine sinnvolle Ergänzung. Die Anwendung speichert auf Knopfdruck den aktuellen Standort des Fahrzeugs und bietet dem Benutzer später die Möglichkeit, diesen über eine eingebaute Google-Maps-Karte wieder zu finden. Leider ist die Anwendung aufgrund ihrer Abhängigkeit von einem ausreichend genauen GPS-Signal in Parkhäusern oder Tiefgaragen nur bedingt oder gar nicht einsetzbar (Sadikov 2010).

Die Anwendung iHUDDisplay (Wessling 2010) nutzt die Reflektion der Windschutzscheibe um Informationen wie die aktuelle Geschwindigkeit, die Richtung, die Höhe oder die aktuelle Uhrzeit als simuliertes Heads-Up-Display (HUD) anzuzeigen. Die Anwendung ist ein Beispiel dafür, dass durch einen kreativen Einsatz von Technologie, angepasst auf die Rahmenbedingungen im Fahrzeug, sinnvolle Funktionen einfach gestaltet werden können.

2.3.2.4 TomTom

Die Firma TomTom ist traditioneller Anbieter von mobilen Navigationslösungen. Durch die aufkommende Verfügbarkeit mobiler Breitbandanbindungen hat TomTom seine Navigationslösungen um zusätzliche, so genannte TomTom LIVE Services ergänzt. Das kostenpflichtige Paket umfasst die sechs Dienste HD-Traffic, Sicherheitswarnungen, Kraftstoffpreise, Lokale Suche mit Google, Live QuickGPSFix und Wetter (TomTom International BV 2010a).

Durch den Service HD-Traffic werden aktuelle Verkehrsinformationen auf TomTom-Geräte übertragen und zur Verfügung gestellt. So werden Informationen zu Störungsmeldungen, genaue Informationen zu Verzögerungszeiten, Reisedauer, Ankunftszeit und Alternativrouten mobil übertragen. Informationen zur Verkehrslage werden unter anderem von den Geräten der Nutzer erhoben. Hierzu werden Informationen wie die Bewegungsgeschwindigkeit die Richtung in der sich ein Nutzer bewegt sowie dessen aktuelle Position zur Bestimmung von Verkehrsinformationen herangezogen und diese durch konventionelle Verkehrsinformationsdienste ergänzt.

Ein weiteres Element der TomTom LIVE Services betrifft aktualisierte Sicherheitswarnungen. Dazu zählen unter anderem Informationen über mobile und festinstallierte Radarkameras sowie Abschnittskontrollen zur Ermittlung der Durchschnittsgeschwindigkeit über eine längere Fahrstrecke. Diese Informationen werden einerseits vom TomTom selber als auch durch die Community der Nutzer bereitgestellt. TomTom Nutzer haben die Möglichkeit, gerichtete Geschwindigkeitsmessungen direkt über ihr Gerät anderen Nutzern zu melden.

Aktuelle Kraftstoffpreise werden mobil an Navigationsgeräte gesendet und dort den Nutzern bereitgestellt. Über die eingebauten Points of Interest, kann sich der Anwender so nicht nur zur nächstgelegenen Tankstelle, sondern auch zur günstigsten nächstgelegenen Tankstelle leiten lassen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

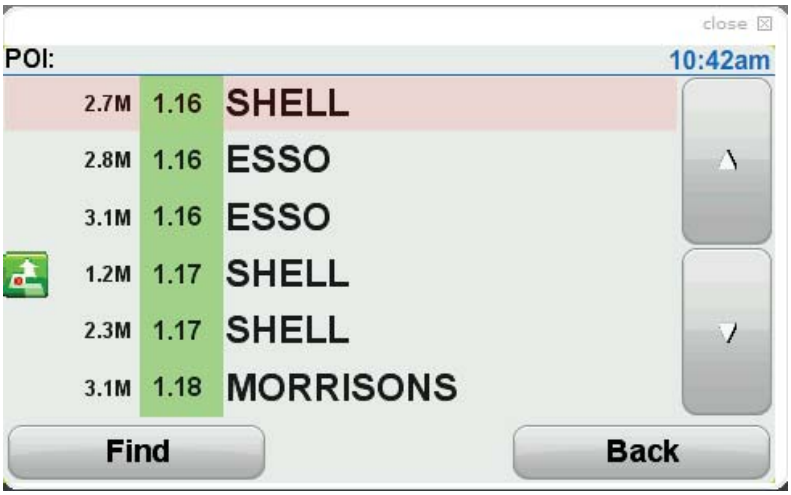


Abbildung 2-18 TomTom LIVE Services: Aktuelle Kraftstoffpreise
Quelle: (TomTom International BV 2010b)

Mit dem Dienst Lokale Suche mit Google hat TomTom die Möglichkeit einer Umgebungssuche integriert. Eingegebene Suchbegriffe werden von Google Maps in Einträge mit Namen, Adressen und gegebenenfalls Telefonnummern aufgelöst und den Nutzern direkt in der Navigation zur Verfügung gestellt. So kann beispielsweise der nächste Supermarkt gesucht und direkt eine Route dorthin berechnet werden.

Der LIVE Service Live QuickGPSFix nutzt die Möglichkeiten einer optimierten Positionsbestimmung. Mit Hilfe von vorbestimmten Positionsinformationen der GPS Satelliten können mit dem Service ausgestattet Geräte die Bestimmung der aktuellen Position wesentlich schneller durchführen. Laut TomTom kann bereits nach spätestens 30 Sekunden die Routeninformationen des Navigationsgeräts genutzt werden.

Der letzte Service integriert aktuelle Wetterinformationen in die Navigation. Einerseits werden aktuelle Wettervorhersagen für die nächsten fünf Tage zur Verfügung gestellt, andererseits können diese Informationen auch zu einer optimierten Routenberechnung genutzt werden. So kann das System beispielsweise Gegenden mit Sturmwarnungen meiden oder im Winter glatte Fahrbahnen berücksichtigen.

2.4 Diskussion

Die Grundlagen für Automotive Services, wie man sie aktuell in modernen Fahrzeugen vorfindet, lassen sich in die Bereiche Unterstützung des Fahrers, Information und Unterhaltung sowie Interaktion mit anderen Fahrzeugen und einem Backend untergliedern. Die drei Bereiche haben zwar aus Sicht des Fahrzeugs und des Fahrers sehr unterschiedliche Aufgaben, werden jedoch durch ihre enge Integration miteinander oftmals als ein einheitliches System aus der Sicht des Nutzers wahrgenommen.

Ausgehend von einfachen Unterhaltungssystemen, wie Autoradios, haben sich komplexe Infotainmentsysteme entwickelt. Diese umfassen die Bereiche der Information, der Unterhaltung, der Assistenz sowie der Kommunikation. In der Regel befassen sich Infotainmentsysteme mit nicht direkt fahrzeugbezogenen Problemstellungen, jedoch ist hier eine klare Trennung zu den diese umfassenden Unterstützungssystemen, den Fahrerassistenzsystemen, nur schwer zu ziehen. Deren Aufgabe ist die direkte Unterstützung des Fahrers bei der Steuerung und Bedienung des Fahrzeugs. Hierzu wird eine Vielzahl an

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

externen Sensoren mit Informationen des Fahrzeugs verbunden, um so für den Fahrer kritische Situationen zu entschärfen oder diesen bei Routineaufgabe zu entlasten.

Sowohl für Infotainmentsysteme als auch für Fahrerassistenzsysteme steht die Interaktion des Nutzers mit dem System im Kontext der Betrachtung. Der Gestaltung der dafür notwendigen Mensch-Maschine-Interaktion kommt somit eine herausragende Bedeutung bei der Umsetzung und Entwicklung von Automotive Services im Fahrzeug zu. Schließlich werden die Informations- und Unterstützungssysteme zunehmend untereinander sowie mit zentralen Backends verknüpft. Hierdurch können Potentiale, wie die Erhöhung der Verkehrssicherheit, die Verkehrseffizient, die Umweltverträglichkeit sowie der Komfort dieser Systeme für den Nutzer erhöht werden.

Aufgrund ihrer Ausrichtung auf Informations- und Unterhaltungsangebote, die meist in keinem direkten Zusammenhang mit der Funktion oder Situation des Fahrzeugs in Zusammenhang stehen, sind Automotive Services in einem großen Maße durch externe Innovationstreiber getrieben. Neue Technologien und Ansätze im Internet, die unter dem Begriff Web 2.0 zusammengefasst werden, fokussieren auf kollaborative Informationsdienste. Diese werden in Form von dynamischen Diensten anstatt konventionellen Softwarepaketen angeboten und vermischen unterschiedlichste Datenquellen und Transportmedien. Durch den konzeptionellen Ansatz der Beteiligung der Nutzer am Erstellen der genutzten Inhalte rück die Technologie in den Hintergrund und die Nutzung der verknüpften Informationen in den Fokus. Automotive Services müssen hier einen geeigneten Weg finden, diese Kollaboration auch im automobilen Umfeld zu ermöglichen.

Eine große Konkurrenz für Dienste im Fahrzeug bilden Smartphones. Entstanden aus der Kombination von Mobilfunkgeräten und PDAs bieten sie den Anwendern viele Optionen die auch in gängigen Infotainmentsystemen analog zu finden sind. Allerdings unterliegen Smartphones wesentlich kürzeren Innovationszyklen und sind durch die Nutzer stärker beeinflussbar. So können Anwender eigene Dienste für Smartphones entwickeln und hinzufügen und diese so individueller gestalten. Darüber hinaus verfügen Smartphones über eine Onlineverbindung und können häufig lokalisierte Dienste durch die integrierte GPS Ortung realisieren. Viele Automotive Services sind demnach auch auf Smartphones realisierbar, was folglich die Erwartungshaltung von Kunden an Automotive Services beeinflusst.

Die Grundlage für mobile Dienste bildet der mobile Breitbandanschluss an Datennetze. Durch die flächendeckende Verfügbarkeit von GSM (2G) basierten und UMTS (3G) basierten Netzten können viele Szenarien bereits heute im Fahrzeug realisiert werden. Mobile Breitbandverbindungen treten somit als Enabler für neue Automotive Services in Erscheinung. In Folge dessen können Services nicht nur die in Abschnitt 2.1 beschriebenen Aufgaben des Infotainment und der Unterstützung bereitstellen, sonder auch völlig neue Geschäftsmodelle wie Mobility Based Services und Location Based Services ermöglichen.

Schließlich bilden Automotive Services auch einen wesentlichen Bestandteil einer erfolgreichen Einführung von Elektromobilität. So geht es bei dieser nicht nur vornehmlich um die Umsetzung einer neuen Antriebstechnologie, sondern vielmehr um Themen wie Klimaschutz, Sicherung der Energieversorgung, Ausbau des Technologie- und Industriestandorts, Verringerung lokaler Emissionen, Integration von Fahrzeugen in das Stromnetz und neue Mobilitätsansätze. Gerade für letztere, im Hinblick auf die aktuellen

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

technischen Limitationen der Elektromobilität, kommt Automotive Services ein entscheidende Rolle zu.

Betrachtet man die diskutierten Angebote und Entwicklungen sowohl der Automobilhersteller als auch der Zulieferer, so lässt sich eine ganze Reihe von Trends ausmachen. Unabhängig von den aktuellen Angeboten, Forschungsinteresses und Strategien der einzelne Hersteller lassen sich unter den vormals genannten Gemeinsamkeiten sieben Trends zusammenfassen: Information, Navigation, Sicherheit, Integration externer Geräte, Integration dritter, Vernetzung, sowie Kundenbindung.

Die Bereitstellung von Informationen haben die Angebote aller Hersteller gemein. Hierzu zählen Nachrichtenprotale, die meist über RSS-Feeds bedient werden, Wetterinformationen sowie die Kommunikation über E-Mail und SMS. Darüber hinaus bieten viele Hersteller, wie beispielsweise Audi mit seinem Angebot Audi Online, den Kunden über das Internet zugängliche Portale an, mit denen sie fahrzeugbezogene Dienste nutzen können. Manche Hersteller gehen sogar soweit, die Inhalte des Internets ungefiltert und nicht verarbeitet ihren Kunden bereitzustellen. Beispiele hierfür sind BMW oder auch Toyota.

Der zweite für Automotive Services interessante Bereich ist der der Navigation. Obwohl nahezu alle Hersteller sowie eine Vielzahl an Zulieferer Navigationslösungen unterschiedlichster Couleur im Angebot haben, zeigt sich ein deutlicher Trend hin zu einer Navigation mit erweiterten Funktionen. Hierzu zählen beispielsweise die Nutzung von Onlineinhalten für die Suche nach Points of Interest oder die Bereitstellung aktueller Wetterinformationen zur Optimierung der Routenberechnung. Ein gutes Beispiel für diese Entwicklung sind die LIVE Services vom TomTom.

Ein weiterer Trend sind Automotive Services im Sicherheitsbereich. Hierzu zählen einfache Notrufe und Hilfefunktionen die viele Premiumhersteller bereits heute am Markt anbieten. Darüber hinaus gibt es noch eine ganze Reihe an innovativen Sicherheitsfunktionen wie beispielsweise dem Robotic Interface oder der Drunk-driving Prevention von Nissan. Schließlich können zu diesem Feld auch unterstützende Funktionen wie beispielsweise die Überwachung der Batterie und Ladekapazität von Elektrofahrzeugen gezählt werden.

Durch den Markterfolg und die damit einhergehende Verbreitung von Smartphones in den letzten Jahren, setzen viele Hersteller auf eine Integration derselben in ihre Fahrzeuge. So zeigen beispielsweise die Ansätze von Daimler für den Smart, von Alpine, aber auch die Angebote die für das Apple iPhone bereits verfügbar sind diesen Trend. Letztere gehen sogar so weit, Automotive Services komplett auf die externen Geräte zu verschieben und auf eine Integration in das Fahrzeug zu verzichten.

Als fünfter Trend lassen sich Bestrebungen hinsichtlich von Application Stores ausmachen. Viele Hersteller setzen daher auf modulare Systeme, wie beispielsweise VW GLORIA, um es Dritten zu ermöglichen, eigene Anwendungen auf den Systemen zu erstellen. Dieses ermöglicht den Herstellern einerseits völlig neue Geschäftsmodelle und schafft andererseits eine Plattform zur Integration Dritter in den Innovationsprozess. So können beispielsweise Automotive Services für das iPhone oder die LTE Connected Drive Plattform von Toyota von Dritten eigenständige entwickelt werden und so deren Know-how und Kreativität genutzt werden. VW geht sogar noch einen Schritt weiter. Im Sommer 2010 wurde in einem Innovationswettbewerb der Kunde direkt an der Gestaltung von Automotive Services beteiligt.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Die nahezu flächendeckende Verfügbarkeit von UMTS basierten Netzen, zumindest in Ballungsgebieten, sowie die aufkommende Möglichkeiten von LTE rücken Möglichkeiten vernetzter Automotive Services stärker in den Fokus. So setzt Toyota gar ganz auf eine Bereitstellung von Automotive Services in der Cloud und Continental forciert eine ubiquitäre Verfügbarkeit ihrer Services.

Der siebte und letzte Trend zeigt die Interessen der Hersteller und Zulieferer am Themenbereiche Automotive Services auf. So soll der Kunde im Aftersales Bereich stärker an das Unternehmen gebunden werden und so neue Phasen der Wertschöpfungskette für den Hersteller erschlossen werden. Darüber hinaus soll die Markentreue gerade junger Kunden, unter 35 Jahre, stärker ausgebaut werden. BMW versucht beispielsweise, dies durch persönliche Assistenzfunktionen die den Fahrer im Fahrzeug direkt mit BMW Mitarbeitern in Kontakt bringen, zu erreichen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

3 Herausforderungen bei der Gestaltung von Automotive Services

Wie in Abschnitt 2 dargelegt, hat die Entwicklung von Infotainmentsystemen in den vergangenen Jahren enorm an Bedeutung für die Automobilbranche gewonnen. Ausgehend von den grundlegenden, in der Serie bereits seit einigen Jahren etablierten und verfügbaren Infotainmentsystemen, hat sich, wie die Übersicht über aktuelle Entwicklungen zeigt, ein klarer Trend hin zu einer Vielzahl an Automotive Services in modernen Fahrzeugen entwickelt. Dabei spielen weniger technische Fragestellungen der Umsetzbarkeit und Machbarkeit die entscheidende Rolle, sondern vielmehr die Frage nach dem richtigen Service und einer geeigneten Umsetzung desselben für die Automotive Domäne.

Natürlich ist diese Fragestellung nach dem „Was“ umgesetzt werden soll im Bereich der Softwareentwicklung nicht neu. So befasst sich beispielsweise das Gebiet des Requirements Engineering mit der Fragestellung, wie Anforderungen erhoben, dokumentiert, analysiert und weiterentwickelt werden können. Darüber hinaus ist auch die Frage nach den relevanten Anspruchsgruppen von entscheidender Bedeutung. Nur wenn die richtigen Stakeholder in geeignete Art und Weise ihrer unterschiedlichen Kompetenzen mit in die Gestaltung von Automotive Services einbringen, können diese erfolgreich sein. Des Weiteren scheint die Frage nach einer geeigneten Interaktion der Fahrer mit Automotive Services ein wesentlicher Faktor zu sein. Durch die Limitationen der Domäne gewinnt die Frage der Interaktion der Nutzer mit dem System eine herausragende Bedeutung. Schließlich sind Automotive Services kein gänzlich neues Phänomen. Wie bereits angesprochen konnten in den letzten Jahren grundlegende Erkenntnisse und Erfahrungen aus den in der Serie erhältlichen Infotainmentsystemen gewonnen werden.

All diese Bereiche stellen somit wesentlichen Herausforderungen bei der Gestaltung von Automotive Services dar. Die nachfolgenden Abschnitte stellen den aktuellen Stand der Literatur sowie der Praxis zu diesen Themen vor, um daraus zu beachende Herausforderungen abzuleiten.

3.1 Relevante Anspruchsgruppen

Bei der Gestaltung von Automotive Services ist, wie auch bei anderen Softwareentwicklungsprojekten, die Identifikation und Auswahl von geeigneten Anspruchsgruppen (Stakeholdern) von entscheidender Bedeutung für einen Erfolg. Rupp definiert Stakeholder als Personen, die direkt oder indirekt Einfluss auf Anforderungen haben. Dazu zählen natürliche und juristische Personen als auch Personen die für eine ganze Gruppe an Personen stehen (Rupp 2007). Bei der Auswahl der Stakeholder gibt es jedoch eine Reihe an wichtigen Punkten die beachtet werden müssen. So muss einerseits sichergestellt werden, dass die richtigen Stakeholder in das Projekt integriert sind, andererseits dürfen bei diesem Prozess aber auch keine Stakeholder vergessen werden. Darüber hinaus dürfen nicht zu viele Stakeholder mit einbezogen werden. Insbesondere solche, die für den Rahmen des Projekts nicht relevant sind, müssen vermieden werden. Durch Letztere können Anforderungen hervorgehen, die in der Realität nicht benötigt werden (Pacheco/Tovar 2007).

Ausgehend von diesen Überlegungen werden nachfolgend Beiträge aus der Literatur diskutiert, die sich mit der Identifizierung von Stakeholder befassen. Analog zu den Überlegungen von Pacheco und Tovar (2007) erfolgt dabei eine Betrachtung Stakeholder

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

charakterisierender Arbeiten, Arbeiten mit dem Fokus auf der Interaktion von Stakeholder sowie der Einschätzung von Stakeholder. Anzumerken dabei ist, dass sich die Literatur auf eine Einschätzung und Kategorisierung fokussiert, die Problemstellung der eigentlichen Identifizierung der Stakeholder meist nur am Rande angesprochen wird.

Ausgehend von dieser Identifikation und Klassifizierung unterschiedlicher Stakeholder-gruppen, wird die Problemstellung der unterschiedlichen Kompetenzprofile der einzelnen Stakeholder diskutiert und anhand der Überlegungen von Khatri et al. (2006) die Situation der unterschiedliche Kompetenzprofile bei der Gestaltung von Automotive Services beleuchtet.

3.1.1 Identifikation von Anspruchsgruppen

Die Suche nach Stakeholder ist eine der ersten und wichtigsten Aufgaben der Entwicklung von Automotive Services. Sie sind es, die neue Services spezifizieren und mit ihren Anforderungen definieren. Dabei sollen nach Möglichkeit die Anforderungen aller relevanten Stakeholder und deren unterschiedliche Bedürfnisse mit erfasst werden. Eine Auswahl an möglichen Stakeholderrollen, wie sie Rupp (2007, 93) vorschlägt, sind in Abbildung 3-1 aufgelistet.

Verschiedene Stakeholderrollen	
Käufer	Technische Experten
Anwender	Marketing und Vertrieb
Entwickler	Produzenten des Produktes
Management	Prüfer und Auditoren
Produktlinienverantwortlicher	Schulungs- und Trainingspersonal
Meinungsführer und öffentliche Meinung	Sicherheitsbeauftragte
Standardisierungsgremium	Betriebsrat
Projekt- und Produktgegner	Personen aus anderen Kulturkreisen
Wartungs- und Servicepersonal	F&E-Repräsentanten
Experten für Prozessoptimierung und Ergonomie	Produktdesigner
Gesetzgeber	Controllingabteilung
Experten für das Systemumfeld	Produktbeseitiger

Tabelle 3-1 ***Verschiedene Stakeholderrollen***
Quelle: Eigene Darstellung in Anlehnung an Rupp (2007, 93)

Aus Hoffmanns (2001) Untersuchungen geht hervor, dass Stakeholder aufgrund ihres Fachkönnens und ihre Anwendungsdomäne charakterisiert werden können. Wichtig ist

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

hierbei eine frühe Integration der Domänenexperten in das Projekt. Eine Identifikation der Stakeholder kann durch eine Beratung mit initialen Stakeholder des Projekts erfolgen (Sommerville/Sawyer 1997). Eine weitere Möglichkeit ist die Einteilung des Auftraggebers in verschiedene Gruppen sortiert nach der Häufigkeit der Benutzung des angestrebten Produkts, Benutzercharakteristiken, Privilegsabstufungen und Fähigkeiten (Wiegers 2003).

Im SEI Technical Report (Christel/Kang 1992) werden Stakeholder entlang von fünf Gruppen eingeteilt: Kunden/Sponsoren/Benutzer, Entwickler, Qualitätspersonal, Sicherheitspersonal und Anforderungsanalysten. Es lassen sich in der Literatur noch eine ganze Reihe an Einteilungen und Klassifizierungen von Stakeholder finden. Hier ist die Capability Maturity Model Integration (CMMI Product Team 2002), der ISO/IEC Standart (Lawson 1999), die Arbeit von Pressman (2009), sowie der Rational Unified Process (Kruchten 2003) zu nennen. Auch wenn all diese Ansätze unterschiedliche Klassifizierungen vornehmen, lassen sich die beiden Gruppen der Softwareexperten, deren Expertise in der Umsetzung von Software liegt, sowie der Anwendungsexperten, die jeweils ihr spezielles Domänenwissen aus der Anwendungsdomäne mitbringen, ausmachen.

Einen alternativen Ansatz um Stakeholder zu identifizieren bietet die Fokussierung auf die Interaktion der Stakeholder. Sharp et al. (1999) verfolgen die Identifizierung aller Stakeholder mit der Entwicklung einer Gruppe von Baseline Stakeholdern. Von dieser Gruppe ausgehend, können Supplier Stakeholder, welche die Baseline Stakeholder mit Informationen und unterstützenden Maßnahmen versorgen, sowie Client Stakeholder, welche die Produkte kontrollieren, ermittelt werden. Eine weitere Gruppe stellen die Satellite Stakeholder dar, welche in Form von Informationssuche oder Kommunikation mit den Baseline Stakeholdern interagieren. Einen Ansatz entlang der Lebensphasen einer Software schlagen Preiss und Wegmann (2001) vor. Grundsätzlich wird zwischen der Entwicklungsphase und der Operationsphase unterschieden, um die für beide Phasen relevanten Stakeholder zu identifizieren.

Eine dritte Möglichkeit Stakeholder zu identifizieren, ist anhand ihrer Fähigkeiten und Einflussfaktoren. Mitchell et al. (1997) führen eine solche Einteilung anhand von drei leitenden Fragestellungen durch:

- Wie Befugt ist der Stakeholder um Anforderungen vorzuschlagen?
- Legitimität der Handlungen, die eine Personen in einem bestimmten Sozialsystem, welches auf der Definition von Normen basiert, durchführen.
- Dringlichkeit - Grad der Aufmerksamkeit die ein Stakeholder vom Projektmanager verlangt.

Ausgehend von diesen Überlegungen können mögliche Stakeholder nun hinsichtlich ihrer Priorität und Bedeutsamkeit eingestuft und so eine Entscheidung ob der Stakeholder für das Projekt relevant ist getroffen werden. Einen ähnlichen Ansatz verfolgt auch das Method Engineering with Stakeholder Input and Collaboration (MEWSIC) von Young et al. (2001). Stakeholder werden analog dem Vorgehen von Mitchell et al. (1997) anhand einer Einschätzung ihrer Person hinsichtlich des Projekts ausgewählt.

Gemein haben all diese Ansätze, dass sie zwar mögliche Kategorien für Stakeholder vorschlagen und auch Optionen aufzeigen, wie diese hinsichtlich ihrer Eignung und Relevanz für ein Projekt eingestuft werden können. Sie gehen jedoch nicht darauf ein wie sichergestellt werden kann, dass keine wesentlichen und relevanten Stakeholder vergessen werden. Da der Fokus dieser Arbeit auf der Gestaltung von Automotive Services und nicht auf deren Betrieb

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

liegt, werden in Anlehnung an Preiss und Wegmann (2001) nur die für die Entwicklungsphase relevanten Stakeholder betrachtet.

3.1.2 Unterschiedliche Kompetenzprofile

Neben einer Analyse der verschiedenen Rollen, die Stakeholder im Rahmen der Gestaltung von Automotive Services einnehmen können, ist die Frage des Wissens und der Fähigkeiten die sie aus ihren unterschiedlichen Domänen mitbringen von entscheidender Bedeutung. Grundsätzlich lassen sich hier zwei Gruppen beschreiben (Khatri et al. 2006): Application Domain Knowledge und Information Systems Knowledge (IS Knowledge).

Mit Application Domain Knowledge wird das Wissen sowie die Fähigkeiten des einzelnen Stakeholders hinsichtlich der Anwendungsdomäne eines Automotive Services beschrieben. Alexander definiert Domain Knowledge als "knowledge of the area to which a set of theoretical concepts is applied" (Alexander 1992; Khatri et al. 2006). In der wissenschaftlichen Literatur wird Domain Knowledge in einer Vielzahl von Arbeiten betrachtet (Alexander 1992; Alexander/Judy 1988). Hier wurden einerseits Domänen wie Physik und Ökonomie bis hin zu Geschichte und Literatur untersucht. Wichtig dabei ist die Fokussierung auf den domänenspezifischen Inhalt sowie die domänenspezifischen Fähigkeiten (McPeck 1990). Ein Mangel an Application Domain Knowledge führt oftmals zu unzureichenden Ergebnissen (Alexander/Judy 1988).

Information Systems Knowledge bezeichnet die Fähigkeiten der Stakeholder hinsichtlich einer Konzeption und Umsetzung von Automotive Services. Khatri et al. (2006) beschreiben IS Knowledge als „IS domain knowledge provides representations, methods, techniques, and tools that form the basis for development of application systems“. Mit der Hilfe der Fähigkeiten die IS Knowledge mitbringt, werden Stakeholder in die Lage versetzt, Probleme und Anforderungen aus der Anwendungsdomäne in Software, oder im Fall der Automotive Services, in neue Services zu implementieren.

Betrachtet man also die Menge der Stakeholder die an einem Softwareprojekt beteiligt sind hinsichtlich dieser beiden Gruppen, erscheint eine erfolgreiche Gestaltung von Automotive Services nur durch eine effektive Zusammenarbeit möglich. Trägt man beide Fähigkeiten in Form einer Matrix an, so folgt daraus eine Klassifikation von Stakeholdern in vier unterschiedliche Kompetenzprofile.

	Hohes Application Domain Knowledge	Niedriges Application Domain Knowledge
Hohes Information Systems Knowledge	○ Produzent des Produkts ○ Entwickler ○ F&E-Repräsentant ○ Produktlinienverantwortlicher	○ Technischer Experte ○ Standardisierungsgremium ○ Produktbeseitiger ○ Experten für das Systemumfeld ○ Wartungs- und Servicepersonal ○ Sicherheitsbeauftragte
Niedriges Information Systems Knowledge	○ Management ○ Prüfer und Auditoren ○ Produktdesigner ○ Experten für	○ Käufer ○ Anwender ○ Controllingabteilung ○ Gesetzgeber

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

	Prozessoptimierung und Ergonomie o Schulungs- und Trainingspersonal o Marketing und Vertrieb	o Personen aus anderen Kulturkreisen o Projekt- und Produktgegner o Betriebsrat o Meinungsführer und öffentliche Meinung

Tabelle 3-2 Kompetenzprofile der Stakeholderrollen
Quelle: Eigene Darstellung

Abbildung 3-2 stellt die von Rupp (2007, 93) beschrieben möglichen Stakeholderrollen (vgl. Abbildung 3-1) hinsichtlich ihrer unterschiedlichen Kompetenzprofile dar. Wie auch in anderen Bereichen als dem der Automotive Services finden sich die meisten Stakeholder naturgemäß in der rechten unteren Gruppe wieder. Gerade Kunden und Endanwender sowie weitere dritte Stakeholdergruppen verfügen oftmals weder über detaillierte Informationen der Anwendungsdomäne noch über geeignetes Wissen zur Umsetzung von Anwendungen. Die zweite Gruppe bilden jene Stakeholder die ein hohes Wissen hinsichtlich der Anwendungsdomäne besitzen jedoch ebenfalls nicht über die Fertigkeiten verfügen, diese geeignet umzusetzen.

Diesen beiden Gruppen gegenüber stehen die Experten der Umsetzungsdomänen. Auch diese lassen sich, wie Abbildung 3-2 zu entnehmen ist, in zwei Gruppen unterteilen. Die größere der beiden Gruppen hat zwar das notwendige Know-how Anwendungen technisch und konzeptionell umzusetzen, ihnen fehlt jedoch das detaillierte Wissen über die Anwendungsdomäne. Die letzte und kleinste Gruppe bilden diejenigen, die sowohl über das technische Know-how der Umsetzung als auch über das fachliche Know-how der Anwendungsdomäne besitzen.

Natürlich ist eine Aufteilung von Personengruppen oder einzelner Personen in die vier skizzierten Kompetenzprofile nicht immer trennscharf durchführbar. Sowie es sicherlich Kunden und Anwender gibt die über die technischen Fähigkeiten zur Umsetzung von Automotive Services verfügen, gibt es auch Vertreter aus dem Marketing und Vertrieb denen ein tiefes Verständnis der Anwendungsdomäne fehlt. Nichtsdestotrotz ist eine funktionierende Zusammenarbeit dieser vier Gruppen entscheidend für erfolgreiche Automotive Services.

3.1.3 Diskussion

Betrachtet man den Stand der Literatur zur Identifikation und Klassifikation geeigneter Stakeholder stellt man fest, dass es eine Vielzahl an Ansätzen und Möglichkeiten gibt. Offensichtlich können Stakeholder anhand unterschiedlichster Kriterien eingeteilt werden. Diese variieren von ihrer Bedeutung für das Projekt über ihre Rolle im Unternehmen hin zu den Kompetenzen die sie aus ihren unterschiedlichen Fachdomänen mitbringen. Hinsichtlich der Frage, wie man sicherstellen kann, dass alle relevanten Stakeholder identifiziert wurden ist in der Literatur nur wenig zu finden. So gibt es Ansätze Stakeholder ausgehend von initialen Stakeholder zu identifizieren oder auch diese entlang der unterschiedlichen Klassifikationen zu suchen.

Für die Gestaltung von Automotive Services ist es von entscheidender Bedeutung, das Domänenwissen der einzelnen Stakeholder in geeigneter Art und Weise nutzbar zu machen. Die zentrale Fragestellung ist daher nicht, wie Stakeholder in unterschiedliche Klassen eingeteilt werden, sondern wie sie ihre unterschiedlichen Kompetenzen nutzenstiftend

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

einsetzen können. Betrachtet man die in Abbildung 3-2 skizzierten Kompetenzprofile, so stellt sich hier die entscheidende Frage wie Stakeholder der vier Gruppen ihr Wissen in geeigneter Art und Weise den Stakeholder der anderen Gruppen kommunizieren können.

3.2 Ermittlung von Anforderungen

Neben der Identifikation von geeigneten Stakeholder ist die Frage welche Automotive Services wie umgesetzt werden soll und welche Anforderungen dabei berücksichtigt werden müssen die erste zentrale Herausforderung bei deren Gestaltung: “The hardest single part of building a software system is deciding precisely what to build. […] No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later” (Brooks 1986). Natürlich ist der Umgang mit Anforderungen sowie die Definition geeigneter Projektziele keine neue Herausforderung. In der Softwareentwicklung befasst sich das Themengebiet des Requirements Engineering mit dieser Aufgabe.

3.2.1 Begriffliche Klärung

In der Literatur gibt es eine Vielzahl an Begriffen und Terminologien für diese wissenschaftliche Disziplin. So finden sich die Begriffe Requirements Engineering, Requirements Management, Anforderungsmanagement und Anforderungstechnik, die bei den unterschiedlichen Autoren verschieden abgegrenzt werden. Die beiden am häufigsten gefundenen Begriffe sind dabei Requirements Engineering sowie Requirements Management. Ebert erläutert hierzu, dass es für den deutschen Sprachraum keine inhaltliche Unterscheidung der beiden Begriffe gibt (Ebert 2005). Die Unterteilung entstammt dem englischen Sprachraum, konnte sich im deutschen Sprachraum jede nicht durchsetzen.

Der Begriff Requirements Management, oder zu Deutsch Anforderungsmanagement, wie ihn Ebert (2005) verwendet, umfasst alle Prozesse die zur Erhebung, Strukturierung und Verwaltung von Anforderungen notwendig sind und schließt die Bereiche Anforderungsdefinition, Anforderungsverfolgung und Anforderungskontrolle mit ein (Ebert 2005). Allgemein ist Requirements Management für den organisatorischen Umgang mit Anforderungen verantwortlich. Während Kotonya und Sommerville (1998, 11) Requirements Management auf den Prozess des Management von Änderungen verschiedener Anforderung beschränken, bezeichnet Rupp (2007, 502) die Maßnahmen als Requirements Managements, welche die Anforderungsanalyse und die weitere Verwendung der Anforderung unterstützt. Diese umfasst die Verwaltung alle Informationen rund um Anforderungen, deren Weiterverwendung in späteren Projektphasen sowie die Wiederverwendung von Anforderungen. Wiegers definiert Requirements Management: "Das Ziel von Requirements Management ist es, qualitativ gute - nicht perfekte - Anforderungen zu generieren, die es erlauben, das Projekt mit einem akzeptablen Risiko zu beginnen“ (Wiegers 2003).

Requirements Engineering hingegen ist ein Requirements Management mit ingenieurmäßigen Vorgehen (Sommerville/Sawyer 1997). Es umfasst also die konkrete Umsetzung der beim Requirements Management beschriebenen Prozesse und Zielsetzungen in die Praxis. Eine der gebräuchlichsten Definitionen für Requirements Engineering entstammt der IEEE 610.12-1990 (IEEE Standards Board 1990):

- The process of studying user needs to arrive at a definition of system, hardware or software requirements.
- The process of studying and refining system, hardware and software requirements.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Darüber hinaus bezeichnen Kotonya und Sommerville den Requirements Engineering Prozess als einen strukturierten Ansatz von Aktivitäten zur Erlangung, Validierung und Pflege einer Anforderungsspezifikation (Kotonya/Sommerville 1998, 9). Der Begriff der Softwaretechnik entstammt der deutschen Übersetzung des Requirements Engineering, seine Verwendung ist jedoch unüblich (Partsch 2010). Darüber hinaus gibt es noch eine Reihe von weiteren Definitionen:

- „Software requirements engineering is the discipline for developing a complete, consistent unambiguous specification – which can serve as a basis for common agreement among all parties concerned – describing what the software product will do (but not how it will do it, this is to be done in the design specification)“ (Boehm 1984).
- "[…] software systems requirements engineering (RE) is the process of discovering that purpose, by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication, and subsequent implementation." (Nuseibeh/Easterbrook 2000)
- Requirements Engineering is a process referring "to all life-cycle activities related to requirements". This process consists of gathering, documentation and managing requirements. (Aurum/Wohlin 2005)

Eine Unterscheidung zwischen Requirements Engineering und Requirements Management erscheint jedoch künstlich, da beide Bereiche essenziell für die Disziplin des Requirements Engineering sind. Außerdem sieht man diese Entwicklung auch am Beispiel der jährlichen internationalen Konferenzen zum Thema Requirements Engineering, in denen inhaltlich alle Bereiche von der Theorie bis zur Praxis vertreten sind (Ebert 2005). Man kann also behaupten, dass die wissenschaftliche Disziplin des Requirements Engineering das Requirements Management einschließt (Rupp 2007).

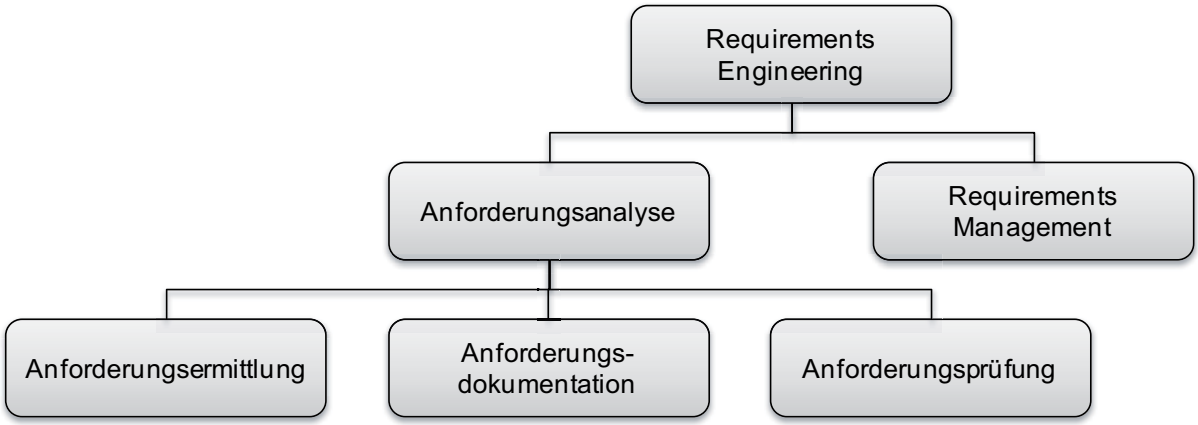


Abbildung 3-1 Begriffliche Abgrenzung Requirements Engineering
Quelle: Eigene Darstellung

Wie Abbildung 3-1 zeigt, schließt sich diese Arbeit einer begrifflichen Definition nach Rupp (2007, 14), Pohl (2008, 43), Wiegers (2003, 11), Partsch (2010, 25ff.), Kotonya und Sommerville (1998, 6), Hoffmann (2001, 27ff.) und Jirotka (1994, 19) an. Im Gegensatz zu den Definitionen von Ebert (2005, 18) und Schienmann (2001, 33) wird Requirements Engineering als Oberbegriff für die Aktivitäten des Requirements Management und der Anforderungsanalyse verwendet. Dabei ist die Anforderungsanalyse die Aktivität, um Anforderungen zu ermitteln, zu formulieren und zu validieren (Rupp 2007, 498). "Sie umfasst sowohl die Ermittlung der Anforderungen alle relevanten Stakeholder, die Transformation in

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Designanforderungen die für den Weiterentwicklungsprozess verwendet werden können als auch die Dokumentation einer geeigneten Spezifikation“ (Husen 2007, 33).

3.2.2 Aufgabenbereiche und Ansätze

Requirements Engineering beschreibt einen systematischen Weg um Anforderungen zu erheben, zu dokumentieren, zu prüfen und zu verwalten (Rupp 2007, 14). Ziel des Requirements Engineering ist es schließlich eine vollständige Anforderungsspezifikation, der alle am Projekt beteiligten Personen zustimmen können, zu erstellen (Pohl 2008, 43). Unpräzise formulierte Anforderungen, eine unzureichende Einbindung der Kunden oder späteren Benutzer sowie unkoordinierte Änderungen der Anforderungen im Verlauf eines Projektes sind entscheidende Gründe für dessen Scheitern (Ebert 2005, 39). Oftmals wird die Spezifikation von Anforderungen nicht gründlich genug durchgeführt was häufig zu Problemen führt die später nur schwierig und kostenintensiv beseitigt werden können (Partsch 2010, 20).

Eine Anforderung wird dabei in der Literatur unterschiedlich definiert. Kotonya und Sommerville beschränken Anforderung auf die frühen Phasen der Softwareentwicklung (Kotonya/Sommerville 1998, 6): „Requirements are defined during the early stages of a system development as a specification of what should be implemented.” Im Gegensatz dazu spricht Pohl von einem kontinuierlichen Requirements Engineering, da sich Anforderungen mit der Zeit ändern können oder am Anfang in unvollständiger, unklarer Art und Weise vorliegen. Die IEEE definierte Anforderungen wie folgt (IEEE Standards Board 1990):

(1) A condition or capability needed by a user to solve a problem or achieve an objective.
(2) A condition or capability that must be met or proposed by a system or system component to satisfy a contract, standard, specification or other formally imposed document.
(3) A documented representation of a condition or capability as in (1) or (2).

Um diesem Ziel gerecht werden zu können, werden in Literatur eine Reihe unterschiedlicher Ansätze vorgeschlagen: Modell getriebenes Requirements Engineering, visuelles Requirements Engineering, wissensbasiertes Requirements Engineering, Distributed Internet-Based Requirements Engineering und Agiles Requirements Engineering.

- *Modell getriebenes Requirements Engineering* beschreibt einen Ansatz, der die Ideen und Konzepte einer durchgehenden modellgetriebenen Softwareentwicklung auf den Bereich des Requirements Engineering überträgt. Model Driven Requirements Engineering eignet sich dabei besonders für solche Projekte, deren Anforderungen sich häufig ändern (Versteegen 2007).
- Das Prinzip des *visuellen Requirements Engineering* ist es, sehr komplexe Sachverhalte klar zu beschreiben. Visuelles Requirements Engineering eignet sich besonders für Fachabteilungen, die Pflichtenhefte für zu entwickelnde Software erstellen (Versteegen 2007).
- Das Ziel von *wissensbasiertem Requirements Engineering* ist es, Schwachstellen in spezifizierten Anforderungen in einem frühen Stadium systematisch zu identifizieren. Das wissensbasierte Requirements Engineering analysiert natürlich sprachliche Dokumente und identifiziert Schwachstellen durch vordefinierte Verben und Schlüsselwörter (Versteegen 2007).

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- DisIRE ist eine Abkürzung für *"Distributed Internet-Based Requirements Engineering"* und es ist eine Tool-basierte Methode für verteiltes, Internet basiertes Requirements Engineering. DisIRE basiert auf "theoretically founded and empirically validated approaches of collaborative requirements elicitation" (Geisser et al. 2007).
- *Agiles Requirements Engineering* bedeutet, dass Methoden des Requirements Engineering in das agile Software Engineering übernommen werden. Agiles Requirements Engineering beinhaltet die folgenden Konzepte: Interaktion mit dem Kunden, Validierung und Verifizierung, nicht-funktionale Anforderungen und Change-Management (Eberlein/Leite 2002).

3.2.3 Phasen des Requirements Engineering

Hinsichtlich der einzelnen Phasen des Requirements Engineering finden sich in der Literatur verschiedene Beschreibungen und Modelle. Vergleicht man, wie in Tabelle 3-3 dargestellt, die Ansätze von Rupp (2007), Pohl (2008), Kotonya und Sommerville (1998), Partsch (2010) und dem IEEE (1990), so lassen sich drei Hauptaktivitäten identifizieren: Anforderungsermittlung und Analyse, Anforderungsprüfung, und Anforderungsdokumentation. Zusätzlich führen Rupp (2007) und Pohl (2008) die begleitende Aktivität des Requirements Management mit an.

	Rupp (2007)	Pohl (2008)	Kotonya und Sommerville (1998)	Partsch (2010)	IEEE (1990)
Anforderungs-ermittlung und Analyse	Anforderungen erheben	Anforderungs-gewinnung	Anforderungen ermitteln	Ermittlung von Anforderungen	Anforderungen ermitteln
					Anforderungen formulieren
		Anforderungs-Übereinstimmung	Anforderungen analysieren und verhandeln		
Anforderungs-dokumen-tation					Anforderungen organisieren
	Anforderungen dokumentieren	Anforderungs-Dokumentation	Anforderungen dokumentieren	Beschreibung von Anforderungen	Anforderungen dokumentieren
Anforderungs-prüfung (Verifikation und Validierung)	Anforderungen prüfen	Anforderungs-Validierung	Anforderungen validieren	Analyse von Anforderungen	Anforderungen prüfen
Requirements Management	Anforderungen verwalten	Anforderungs Management			

Tabelle 3-3 Phasen des Requirements Engineering
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

3.2.3.1 Anforderungsermittlung und Analyse

Die erste Phase des Requirements Engineering befasst sich mit der Ermittlung und Analyse von Anforderungen. Hierbei sollen die Bedürfnisse und Ziele der Stakeholder erfasst und in eine für den Systementwickler verständliche Form gebracht werden. Wichtig dabei ist, dass Anforderungen aller Stakeholder, ausgehend von ihren unterschiedlichen Kompetenzprofilen, Berücksichtigung finden. Der Prozess der Anforderungsermittlung umfasst hierfür eine Reihe von Techniken und Aktivitäten, wie die Gewichtung von Anforderungen, die Kommunikation sowie die Zusammenarbeit mit den Stakeholdern. Zowghi und Coulin (2005) unterscheiden fünf grundlegende Aufgaben dieser Phase: Verstehen der Anwendungsdomäne, Identifikation der Quellen von Anforderungen, Analysieren der Stakeholder, Auswahl geeigneter Techniken und Tools sowie die Erhebung der Anforderungen aus den unterschiedlichen Quellen.

Zunächst ist es für eine erfolgreiche Anforderungsermittlung entscheidend, die Umgebung zu analysieren in der das zu erstellende System eingesetzt werden soll. Hier kommen politische, soziale und unternehmensspezifischer Aspekte zu tragen. Insbesondere Arbeitsprozesse und damit verbundener Herausforderungen, welche durch das zu erstellende System unterstützt oder gelöst werden sollen, müssen hier beschrieben werden.

In einem nächsten Schritt müssen die unterschiedlichen Quellen von Anforderungen identifiziert werden. Hierzu zählen Stakeholder, wie Benutzer und Fachleute, vorhandene Systeme oder Prozesse sowie sonstige zugängliche Unterlagen und Dokumente.

Im Rahmen der Identifikation aller verfügbaren und notwendigen Anforderungsquellen müssen auch die am Projekt beteiligten Stakeholder identifiziert werden. Eine eingehende Betrachtung dieser Aufgabe findet sich in Abschnitt 3.1.

Abhängig vom Projektkontext, den Anforderungsquellen sowie den beteiligten Stakeholdern müssen geeignete Techniken und Werkzeuge ausgewählt werden. Ein einziger Ansatz für die Anforderungsermittlung ist in der Regel nicht für alle Projektarten ausreichend (Zowghi/Coulin 2005, 22). In den meisten Projekten wird daher eine Auswahl unterschiedlicher Methoden eingesetzt.

Sind diese vier vorbereitenden Aufgaben durchgeführt worden, so kann mit der fünften und eigentlichen Aufgabe, der Anforderungsermittlung und Analyse, begonnen werden. Unter Einsatz der ausgewählten Techniken und Werkzeuge können nun Anforderungen identifiziert und analysiert werden. Dabei muss insbesondere auf die Wünsche und Bedürfnisse der einzelnen Stakeholder eingegangen werden.

3.2.3.1.1 Arten von Anforderungen

Wenn man von Anforderungen im Kontext des Requirements Engineering spricht, kann man diese in unterschiedliche Kategorien gruppieren. Sowohl Rupp (2007) als auch Hood et al. (2007) schlagen hier eine Einteilung in sieben Kategorien vor:

- Funktionalen Anforderungen,
- Technischen Anforderungen,
- Anforderungen an die Benutzungsschnittstelle,
- Qualitätsanforderungen,
- Anforderungen an sonstige Lieferbestandteile,
- Anforderungen an durchzuführende Tätigkeiten und
- Rechtlich – vertraglichen Anforderungen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Die wichtigste Gruppe bilden die funktionalen Anforderungen. Sie beschreiben die später vom System zur Verfügung gestellten Funktionen hinsichtlich der Interaktion mit den Nutzern, mit anderen Systemen sowie der internen Verarbeitung von Daten.

Neben den funktionalen Anforderungen gibt es noch eine ganze Reihe an weiteren Kategorien. Diese werden auch als nichtfunktionale Anforderungen bezeichnet. Hierzu zählen technische Anforderungen wie beispielsweise Hardwareanforderungen, Architektur Anforderungen oder Anforderungen an die zu verwendende Programmiersprache. Auch beinhalten nichtfunktionale Anforderungen, Anforderungen an die Benutzerschnittstelle, also die Form und Funktion der Ausgabegeräte und insbesondere deren graphische Gestaltung. Darüber hinaus gibt es noch Qualitätsanforderungen, Anforderungen an sonstige Lieferbestandteile, wie beispielsweise Handbücher, Projektpläne oder Trainingsunterlagen, Anforderungen durchzuführender Tätigkeiten, wie Systemeinführung, Schulungen oder Abnahmetests, sowie rechtlich und vertragliche Anforderungen wie Meilensteine, Eskalationspfade oder etwaige Vertragsstrafen.

3.2.3.1.2 Menschliche Einflussfaktoren

Auch wenn Anforderungen aus den unterschiedlichsten Quellen, wie beispielsweise existierenden Systemen, Prozessbeschreibungen und Dokumenten sowie rechtlichen Anforderungen entstammen, so nehmen doch die Anforderungen der Stakeholder hier eine zentrale Rolle ein. Es gibt eine ganze Reihe unterschiedlicher Einflussfaktoren die sich auf den Projektverlauf auswirken können. Entscheidend ist hier der zentrale Einfluss dieser Faktoren auf die Kommunikation mit den Projektbeteiligten. Rupp identifiziert hier acht mögliche Einflussfaktoren die es zu beachten gilt (Rupp 2007, 73ff.):

- *Motivation:* Oft mangelt es viele Stakeholder an der notwendigen Motivation sich aktiv und konstruktiv an einem Projekt zu beteiligen. Sie sind oft mit anderen, für sie persönlich wichtigeren Projekten befasst und können oder wollen daher nicht die notwendige Aufmerksamkeit aufbringen.
- *Kommunikative Fähigkeiten:* Stakeholder haben manchmal Schwierigkeiten ihre Gedanken oder Vorstellungen sprachlich richtig auszudrücken. Hier müssen besondere Techniken und Methoden sowie speziell geschulte Analytiker eingesetzt werden, um Stakeholder entsprechen zu unterstützen.
- *Art des Wissens:* Ein weiteres grundsätzliches Problem ist, dass sich nicht allen Stakeholdern ihren Anforderungen unmittelbar bewusst sind. Es gibt Anforderungen, die den Beteiligten durch entsprechende Methoden erst bewusst gemacht werden müssen. Andererseits sind für Stakeholder viele ihre Anforderungen so selbstverständlich, dass sie diese nicht angeben.
- *Homogenität der Stakeholdermeinungen:* In der Regel haben Stakeholder unterschiedlicher Gruppierungen auch unterschiedliche Zielvorstellungen und somit Meinungen zu einzelnen Anforderungen. Hier kommt dem Erheben und Analysieren von Anforderungen auch eine starke vermittelnde Rolle zu.
- *Abstraktionsvermögen:* Um Anforderungen, die aus praktischen Problemen oder Situationen heraus resultieren, geeignet formulieren zu können, müssen Stakeholder über ein gewisses Maß an Abstraktionsvermögen besitzen. Auch spielt der Hintergrund, aus dem die Stakeholder kommen, eine entscheidende Rolle.
- *Machtsituation und Gruppendynamik: D*ie Analyse von Anforderungen erfordert eine enge Zusammenarbeit der einzelnen Stakeholder miteinander. Aufgrund der

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

unterschiedlichen Fachabteilungen und Unternehmen aus denen diese Stakeholder stammen, führen komplexe Machtverhältnisse einzelner Gruppen untereinander oft zu problematischen Spannungen.

- *Kultur:* Ein oft vernachlässigter Faktor ist die unterschiedliche Kultur der Stakeholder. Dies zeigt sich einerseits in gesellschaftlichen Regeln und Verhaltensnormen, die eine Zusammenhalt erschweren, andererseits aber auch in unterschiedlichen Zielvorstellungen welche Aufgabe eine Software übernehmen sollen und darüber wie dieses gestaltet werden sollen.
- *Kompetenz der Stakeholder:* Wie bereits in Abschnitt 3.1 diskutiert, entstammen Stakeholder unterschiedlichen Gruppierungen und verfügen somit über unterschiedliche Kompetenzprofile. Dies hat einerseits den Vorteil, dass Anforderungen aus unterschiedlichen Kenntnisständen und Fähigkeiten gestaltet werden, andererseits wird dadurch eine gemeinsame Zielvorstellung und Kommunikation deutlich erschwert.

3.2.3.1.3 Ermittlungstechniken

Wie aus den unterschiedlichen Arten von Anforderungen sowie der großen Variabilität unterschiedlichster Stakeholder und der damit einhergehenden menschliche Einflussfaktoren hervorgeht, bedarf es zur Erhebung und Analyse von Anforderungen geeigneter Techniken und Methoden. Zunächst gilt es hier, die unterschiedlichen Rahmenbedingungen eines Projekts zu betrachten. "Eine Rahmenbedingung ist eine organisatorische oder technische Anforderungen, die die Art und Weise eingeschränkt, wie ein Produkt entwickelt wird" (Pohl 2008, 18). Rupp (2007, 72) unterscheidet drei Kategorien von Projektrahmenbedingungen: Mensch, organisatorische Rahmenbedingungen und fachlicher Inhalt. Die Wahl einer geeigneten Ermittlungstechnik erfordert eine Berücksichtigung all dieser Kategorie.

In der Literatur wird daher eine große Vielzahl an unterschiedlichen Techniken und Methoden vorgeschlagen. Diese entstammen den verschiedensten Disziplinen und Forschungsrichtungen, wie beispielsweise der Ethnographie bis hin zur technischen Gestaltung von Prototypen. In Anlehnung an Rupp (2007, 109ff.) lassen sich diese Ermittlungstechniken in fünf Kategorien einteilen: Kreativitätstechniken, Beobachtungstechniken, Befragungstechniken, vergangenheitsorientierte Techniken und unterstützende Techniken. Abbildung 3-4 gibt eine Übersicht über die in der Literatur von Rupp (2007, 115ff.), Kotonya und Sommerville (1998, 62ff.), Pohl (2008, 323ff.), Ebert (2005, 95ff.), IEEE Standards Board (1998, 16), Schwabe und Krcmar (1996) sowie Glaska et al. (2008) beschriebene Ermittlungstechniken und sortiert diese entlang der fünf genannten Kategorien.

Ermittlungstechniken	
Kreativitätstechniken	o Brainstorming o Brainstorming Paradox o Methode 6-3-5 o Wechsel der Perspektive o Walt Disney-Methode o Bionik / Bisoziation o Obsorn Checkliste

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Beobachtungstechniken	o Feldbeobachtung o Apprenticing o Needs Driven Approach
Befragungstechniken	o NLP o Fragebogen o Interview o Selbstaufschreibung o One – Site – Customer
Vergangenheitsorientierte Techniken	o Systemarchäologie o Wiederverwendung
Unterstützende Techniken	o Workshops o Mind Mapping o CRC – Karten o Snowcards o Audio o Video o Essenzbildung o Anforderungen erahnen o Anwendungsfälle o Prototyping

Tabelle 3-4 ***Übersicht Ermittlungstechniken***
Quelle: Eigene Darstellung (In Anlehnung an (Rupp 2007; Kotonya/Sommerville 1998; Pohl 2008; Ebert 2005; IEEE Standards Board 1998; Schwabe/Krcmar 1996; Glaska et al. 2008))

3.2.3.1.3.1 Kreativitätstechniken

Kreativitätstechniken werden in der Regel für die Gestaltung der initialen Anforderungen und somit der Diskussion einer ersten Idee und Vision eines Systems eingesetzt. Mit ihrer Hilfe kann ein Überblick über die Herausforderungen gewonnen, sowie innovative Ideen gesammelt werden. Auch werden Kreativitätstechniken eingesetzt, um unbewusste Anforderungen der Stakeholder zu ermitteln (Rupp 2007, 115). Zwei der bekanntesten Vertreter von Kreativitätstechniken sind das Brainstorming sowie die Walt Disney Methode. Weitere Techniken sind Abbildung 3-4 zu entnehmen.

In einem Brainstorming werden Ideen einer ganzen Gruppe von Stakeholder im Rahmen einer Sitzung generiert. Geleitet von einem Moderator können so viele Ideen in sehr kurzer Zeit gesammelt und mit der gesamten Gruppe weiterentwickelt werden. Schließlich werden diese Ideen analysiert um Anforderungen daraus zu entwickeln. (Rupp 2007,116). Eine interessante Variante des Brainstormings ist das Brainstorming Paradox. Hierbei geht es nicht darum, eine Lösung zu entwickeln, sondern vielmehr herauszufinden was nicht umgesetzt werden soll. So können in einer frühen Phase explizit Risiken entdeckt und Fehlentwicklungen eines Projekts vermieden werden (Rupp 2007, 117).

Die Walt Disney Methode versetzt Stakeholder, räumlich und zeitlich getrennt, in verschiedene Rollen und Sichten. Diese explizite Trennung und jeweils erfolgende Konzentration auf nur eine Sichtweise versucht ausreichend Raum für die unterschiedlichen Aspekte des Problems zu schaffen, um so neue Ideen zu entwickeln. Die Teilnehmer nehmen dabei die Sichtweisen des

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- Träumers in Visionärs, sie stehen für Fantasie, Kreativität und neue Ideen,
- des Realisten, sie stehen für Machbarkeit und Umsetzbarkeit und
- des Kritikers, sie stehen für Sinnhaftigkeit, Schwachstellen und negative Aspekte

ein.

3.2.3.1.3.2 Beobachtungstechniken

Beobachtertechniken werden eingesetzt, um den IST-Zustand eines Systems zu erfassen und die von einem System zu unterstützenden Arbeitsabläufe und Prozesse zu ermitteln. Dabei nehmen die Stakeholder eine passive Rolle ein, indem sie ihre Anforderungen nicht selber explizieren, sondern vom Anforderungsanalysten bei der Arbeit beobachtet und ihre Arbeitsschritte dokumentiert werden. Diese Techniken eignen sich besonders gut, wenn Stakeholder nicht in der Lage sind ihre Anforderungen ausreichend zu artikulieren oder aufgrund ihrer Zeit oder Motivation nicht für andere Erhebungstechniken infrage kommen. Bei den Beobachtungstechniken kann man zwischen Feldbeobachtungen und Apprenticing unterscheiden (Rupp 2007, 121).

Bei der Feldbeobachtung beobachtete ein Anforderungsanalyst die Stakeholder im Rahmen ihrer täglichen Arbeit. Schwabe und Krcmar schlagen hier mit dem Needs Driven Approach einen Leitfaden vor, der es dem Anforderungsanalysten ermöglicht diese Beobachtungen strukturiert durchzuführen (Schwabe/Krcmar 1996). Glaska et al. haben die Methode des Needs Driven Approach hinsichtlich einer Unterstützung der Dokumentation der Beobachtungen erweitert (Glaska et al. 2008).

Im Gegensatz zur reinen Beobachtung der Stakeholder wird mit der Technik des Apprenticing der Anforderungsanalyst in die Tätigkeiten der Stakeholder eingelernt. So kann er aus erster Hand seine eigenen praktischen Erfahrungen nutzen um daraus Anforderungen abzuleiten (Rupp 2007, 123).

3.2.3.1.3.3 Befragungstechniken

Im Gegensatz zu den Beobachtungstechniken, bei denen es vorwiegend um das implizite Wissen der Stakeholder geht, werden Befragungstechniken eingesetzt um das explizite Wissen der Stakeholder zu ermitteln. Befragungstechniken zählen zu den am häufigsten eingesetzten Ermittlungstechniken, da sie geeignet sind Anforderungen verschiedenster Detaillierungsgrade zu ermitteln. Stakeholder werden dabei gezielt nach ihren Wünschen und Vorstellungen gefragt und die resultierenden Antworten in Anforderungen übersetzt (Rupp 2007, 122). Es ist daher nicht verwunderlich, dass es in der Literatur eine ganze Reihe von Befragungstechniken gibt.

Die wohl am häufigsten eingesetzte Befragungstechniken ist das Interview. Dabei werden einem oder mehreren Stakeholdern Fragen gestellt und deren Antworten protokolliert. Diese Form der Befragung hat den Vorteil, dass man individuell auf den Interviewten eingehen kann und so spezifische Fragen direkt betrachten und daraus resultierende Anforderungen gewinnen kann. Vor allem am Anfang der Anforderungsermittlung sind persönliche Interviews sinnvoll, in denen gemeinsam die groben Anforderungen das System erarbeitet werden. Ein großer Nachteil ist jedoch, dass Interviews nur mit einer begrenzten Zahl an Stakeholder aufgrund ihrer zeitintensiven Natur durchgeführt werden können (Rupp 2007, 124).

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Eine Variante der Befragung, die diesen Nachteil des hohen Aufwands und der somit begrenzten Anzahl an Interviews begegnet, ist der Fragebogen. Im Rahmen eines Fragebogens werden den Stakeholdern einheitliche und vordefinierte Fragen gestellt, die diese schriftlich beantworten. Auf diese Weise kann eine sehr große Anzahl an Stakeholder gleichzeitige in die Anforderungsermittlung mit einbezogen werden. So ist es allerdings sehr schwer auf individuelle Fragen und Anforderungen einzugehen (Rupp 2007, 123).

3.2.3.1.3.4 Vergangenheitsorientierte Techniken

Gerade wenn es bei der Anforderungsermittlung um Anforderungen an eine Weiterentwicklung oder Ablösung eines Altsystems geht, können vergangenheitsorientierte Techniken einen großen Mehrwert bieten. Durch ihren Einsatz können bestehende Systeme bis ins Detail verstanden werden (Rupp 2007, 126). Zwei Vertreter dieser Techniken sind die Systemarchäologie sowie die Wiederverwendung.

Bei der Systemarchäologie werden Anforderung für das neu zu entwickelnde System aus einem existierten System oder den dazugehörigen Dokumenten entwickelt. Besonders Anleitungen und Handbücher eignen sich, um einen Überblick über das Verhalten des bestehenden Systems zu erlangen. Die Extraktion von Anforderungen aus Handbüchern ist jedoch sehr zeitaufwändig und eignet sich ausschließlich dazu, den existierten Funktionsumfang eines Altsystems zu ermitteln (Rupp 2007, 126).

Existieren bereits dokumentiert Anforderung von einem Altsystem oder einem ähnlichen System, so können diese Anforderungen wiederverwendet werden. Durch diese Wiederverwendung können Kosten und Aufwände eingespart werden, da Anforderungen nicht neu entwickelt werden müssen sondern lediglich an neuen Herausforderungen angepasst werden (Rupp 2007, 127).

3.2.3.1.3.5 Unterstützende Techniken

Die letzten von Rupp (2007) genannten Erhebungstechniken sind die so genannten unterstützenden Techniken. Unterstützende Techniken sind nicht für sich allein genommen einsetzbar, sondern werden in Kombination mit den bereits beschriebenen Techniken der anderen vier Kategorien genutzt. Sie dienen dazu, deren Effektivität zu verbessern und somit die Qualität der ermittelten Anforderungen zu erhöhen (Rupp 2007, 128). Zu den unterstützenden Techniken werden Workshops, Mind Mapping, Anwendungsfälle, CRC-Karten, Snowcards, Audioaufzeichnungen, Videoaufzeichnungen Essenzbildung, Anforderungen erahnen und Prototyping gezählt.

3.2.3.1.4 Qualitätskriterien für Anforderungen

Um hochwertige und qualitative Anforderungen zu gewährleisten, gibt es eine Reihe an Qualitätskriterien die diese erfüllen müssen. Im IEEE Standard 830-1998 werden acht Qualitätskriterien für die Anforderungsspezifikation festgelegt. Demnach muss eine Anforderungsspezifikation korrekt, eindeutig, vollständig, konsistent, bewertet, prüfbar, modifizierbar und verfolgbar sein (IEEE Standards Board 1998b). Rupp schlägt eine Liste von insgesamt 12 Qualitätskriterien vor, da sich diese nicht nur auf die Anforderungsspezifikation sondern vielmehr auf jede einzelne Anforderung beziehen sollen (Rupp 2007, 27ff.):

- *Vollständig:* Jede einzelne Anforderung muss die Funktionalität vollständig beschreiben.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *Korrekt:* Eine Anforderung ist korrekt, wenn sie die Vorstellung der Stakeholder wiedergibt.
- *Klassifizierbar:* Jede Anforderung sollte bezüglich der juristischen Verbindlichkeit klassifiziert werden können.
- *Konsistent:* Anforderungen müssen in sich und allen anderen Anforderungen gegenüber konsistent sein.
- *Prüfbar:* Anforderungen müssen so beschrieben werden, dass sie getestet werden können.
- *Eindeutig:* Anforderungen sollen nur auf eine Art und Weise interpretiert werden können.
- *Verstehbar:* Alle Stakeholder sollen die Anforderungen verstehen können.
- *Gültig und aktuell:* Jede Änderung soll dokumentiert und somit nachvollziehbar gemacht werden.
- *Realisierbar:* Jede Anforderung muss in Bezug auf bekannte Fähigkeiten, Grenzen des Systems und seiner Umgebung umsetzbar sein.
- *Notwendig:* Jede Anforderung sollte eine tatsächlich benötigte Eigenschaft oder Leistung fordern.
- *Verfolgbar:* Jede Anforderung soll eindeutig identifiziert und somit verfolgt werden können.
- *Bewertbar:* Anforderungen sollen nach Wichtigkeit oder Priorität bewertet werden können.

3.2.3.2 Anforderungsdokumentation

Die zweite Phase des Requirements Engineering befasst sich mit dem Festhalten und Persistieren der in der ersten Phase gewonnenen Anforderungen. Der Begriff Phase ist hier nicht zeitlich oder sequenzielle zu verstehen, vielmehr erfolgt einer Dokumentation während und parallel der Erhebung und Analyse der Anforderungen. Die Dokumentation nimmt somit einen sehr großen Stellenwert im Requirements Engineering ein. Anforderungsdokumente schaffen eine gemeinsame Informationsbasis auf die alle Projektbeteiligten zugreifen können und fördern somit die Kommunikation zwischen den beteiligten Stakeholdern. Auf diese Weise können Stakeholder, die im Rahmen unterschiedlicher Erhebungstechniken an der Anforderungserhebung und -analyse beteiligt sind, auf die Ergebnisse und Anforderungen der anderen Stakeholder zugreifen. Darüber hinaus führt diese explizite Dokumentation der Informationen dazu, dass über Inhalte reflektiert wird und Unvollständigkeiten und Fehler aufgedeckt werden können (Pohl 2008, 217).

3.2.3.2.1 Dokumentationstechniken

Abhängig vom Kontext des jeweiligen Projekts, den beteiligten Stakeholdern sowie den allgemeinen Rahmenbedingungen gibt es unterschiedliche Techniken Anforderungen zu dokumentieren. So können einerseits natürlich sprachliche Textdokumente und Tabellen und andererseits formale Techniken der Modellierung zum Einsatz kommen. Dabei werden in der Regel zusätzliche Informationen der Anforderungen wie Ziele, Definitionen und Anwendungsfälle integriert (Hood et al. 2007, 31-40).

3.2.3.2.1.1 Ereignisgesteuerte Prozessketten

Die Architektur integrierter Informationssysteme (ARIS) stellt einen Ordnungsrahmen für die Beschreibung von betrieblichen Informationssystemen dar. Informationssysteme werden

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

dabei aus einer Funktionssicht, Organisationssicht, Datensicht, Leistungssicht und Steuerungssicht betrachtet (Scheer 2001). Dazu bedient sich ARIS einer Reihe von grafischen Modellierungssprachen. Der wohl bekannteste Vertreter dürften die ereignisgesteuerten Prozessketten (EPK) sein. Im Rahmen der Dokumentation von Anforderungen werden EPKs häufig zur Dokumentation von fachlichen Abläufen und Prozessen genutzt (Rupp 2007, 186).

3.2.3.2.1.2 Kontextdiagramme

Kontextdiagramme beschreiben die Schnittstelle des Systems zu dessen Umgebung. Sie zeigen die Umwelt und die Datenflüsse eines Systems über dessen Grenzen hinaus. Die Nachbarsysteme werden als Rechtecke dargestellt, das Gesamtsystem durch einen Kreis repräsentiert. Datenflüsse werden als Doppelpfeile modelliert (Rupp 2007, 158).

3.2.3.2.1.3 Anwendungsfälle

Anwendungsfälle (engl. Use Cases) werden genutzt um die grundlegenden Funktionalitäten eines Systems und die Interaktion mit den wichtigsten Stakeholdern zu beschreiben. Ein einzelner Anwendungsfall beschreibt dabei die Interaktion eines Nutzers mit dem System. Die Darstellung von Anwendungsfällen erfolgt entweder in Form von UML UseCase Diagrammen oder textuell in Form von definierten Templates (Jacobson 1992).

3.2.3.2.1.4 Aktivitätsdiagramme

Aktivitätsdiagramme eignen sich besonders gut, Abläufe und die dazugehörigen Regeln darzustellen. Sie können zu unterschiedlichen Projektzeitpunkten mit unterschiedlichem Detaillierungsgrad erstellt werden und lassen sich somit vielfach im Projekt einsetzen. Aktivitätsdiagramme zeigen den Rahmen und die Regeln von Verhaltensabläufen. Die Elemente sind im wesentlichen Start- und Endpunkte, Verzweigungen, Bedingungen, Aktivitäten, Aktionen, Objektknoten, Kontrollelemente zur Ablaufsteuerung und verbindende Kanten (Rupp 2007, 166). Häufig werden sie alternativ zu ereignisgesteuerten Prozessketten eingesetzt.

3.2.3.2.1.5 Klassendiagramme

Klassendiagrammen werden genutzt um die Struktur des zu entwickelnden Systems darzustellen. Sie zeigen die wesentlichen Elemente des Systems und beschreiben deren Funktionalität und Beschaffenheit sowie deren Beziehungen zueinander. (Rupp 2007, 186). Klassendiagrammen können ebenfalls eingesetzt werden um für ein Projekt relevante Begriffe und deren Beziehungen zueinander zu beschreiben (Rupp 2007, 194).

3.2.3.2.1.6 Zustandsautomaten

Die formale Notation der Zustandsautomaten wird genutzt um das konkrete Verhalten eines Systems formal zu beschreiben. Sie werden häufig zur Analyse von Abläufen sowie der Simulation eines Systemverhaltens eingesetzt. Aufgrund ihrer formalen Natur eignen sie sich nur bedingt als Kommunikationsmedium, da sie beim Stakeholder ein sehr hohes Abstraktionsvermögen und formales Verständnis voraussetzen (Rupp 2007, 207).

3.2.3.2.1.7 Sequenzdiagramme

Sequenzdiagramme werden eingesetzt, die Kommunikation zwischen mehreren Kommunikationspartnern grafisch darzustellen. Der Fokus liegt hierbei auf dem Informationsaustausch und der Reihenfolge der Nachrichten. Sie befassen sich also mit dem

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

zeitlich, logischen Ablaufbedingungen. Grundelemente der Sequenzdiagramme sind die Kommunikationspartner sowie die Nachrichten die untereinander verschickt werden (Rupp 2007, 179).

3.2.3.2.1.8 Szenarios

Szenarios werden genutzt um Situationen, in denen Menschen Systeme nutzen, zu beschreiben. Im Zentrum eine Szenarios steht mindestens ein Akteur der mit dem System integriert um eine bestimmte Aufgabe zu lösen. Die Beschreibung ist in der Regel idealisiert, episodenhaft dargestellt, wobei der Kontext des Szenarios vorab definiert wird. Szenarios können mittels Freitext, strukturiertem Text oder Diagrammen dargestellt werden, meist wird jedoch eine Notation in natürlicher Sprache gewählt (Rupp 2007, 177).

3.2.3.2.1.9 Natürlicher Sprache

Pohl definiert einen natürlich sprachliche Anforderung als „eine Anforderung, die in einer natürlichen Sprache dokumentiert ist“ (2008, 229). In der Praxis bedeutet dies, dass Anforderungen in textueller Form beschrieben werden. Natürliche Sprache hat die Vorteile, dass sie vielseitig einsetzbar und sehr flexibel ist und eine Dokumentation in unterschiedlichen Abstraktions- und Detaillierungsgraden erlaubt. Auch werden keine besonderen Modellierungsfertigkeiten oder einzusetzende Softwarewerkzeuge benötigt oder vorausgesetzt. Der Nachteil von natürlicher Sprache ist, dass sie formal sehr unpräzise ist und deshalb zu unterschiedlichen Interpretationen der verschiedenen Stakeholder führt (Pohl 2008, 239).

3.2.3.2.1.10 Glossare

Glossare werden häufig in Kombination mit anderen Dokumentationstechniken eingesetzt. "Ein Glossar ist eine Sammlung von Fachbegriffen, die Bestandteile einer Sprache (Terminologie) sind. Das Glossar legt für diese Fachbegriffe deren spezifische Bedeutung fest" (Pohl 2008, 244). Durch Glossare können Mehrdeutigkeiten die durch Homonyme und Synonyme entstehen vermieden werden. Darüber hinaus beseitigt ein Glossar Unklarheiten bzgl. einzelner Begriffe und stellt sicher, dass diese von verschiedenen Personen nicht unterschiedlich interpretiert werden können (Pohl 2008, 244).

3.2.3.2.1.11 Syntaktische Anforderungsmuster

„Ein syntaktisches Anforderungsmuster dokumentiert wieder verwendbares Wissen über die Formulierung natürlichsprachlicher Anforderung, indem es syntaktische Strukturen für die Dokumentation von natürlichsprachlichen Anforderungen vorgibt" (Pohl 2008, 245).

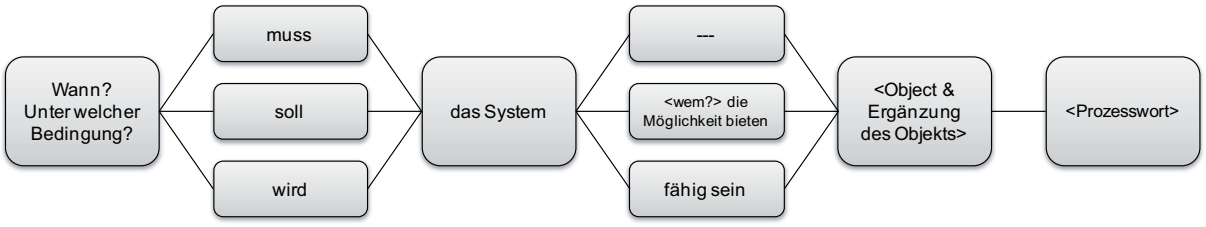


Abbildung 3-2 Anforderungsschablone
Quelle: (Rupp 2007, 234)

Syntaktische Anforderungsmuster unterstützen die Dokumentation von Anforderungen, indem sie konkrete syntaktische Strukturen zur Definition von Anforderungen vorgegeben.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Rupp (2007, 255) bezeichnet diese Technik als Anforderungsschablonen. Abbildung 3-2 zeigt ein Beispiel für eine solche Schablone.

3.2.3.2.1.12 Normsprachen

„Eine Normsprache besitzt eine im Bezug auf eine spezifische Domäne eingeschränkte natürlichsprachliche Grammatik (Syntax) und definiert eine Menge von Begriffen (Semantik), die unter Verwendung der vorgegebenen Grammatik zur Konstruktion von Aussagen über die Domäne verwendet werden können" (Rupp 2007, 247). Eine Normsprache stellt somit eine spezifische Fachsprache dar, die natürlichsprachliche Form mit fest definierten Regeln verknüpft. Aufgrund ihrer vereinfachten Grammatik und ihres festgelegten Wortschatzes reduziert sie Mehrdeutigkeiten der natürlichen Sprache und ist semantisch überprüfbar (Rupp 2007, 246).

3.2.3.2.2 Qualitätskriterien für Anforderungsdokumente

Eine Anforderungsspezifikation umfasst eine Menge formulierter Anforderungen, die zu einem bestimmten Zeitpunkt bekannt sind. Die Qualität eines Anforderungsdokuments hängt maßgeblich von den in Abschnitt 3.2.3.1.4 beschriebenen Qualitätseigenschaften der einzelnen Anforderungen sowie von übergreifenden Qualitätseigenschaften des Gesamtdokuments ab (Pohl 2008, 246). Nach Hood et al. (2007, 18-20) muss eine Anforderungsspezifikation einen angemessenen Umfang und eine klare Struktur aufweisen, sortierbar sein, qualitativ hochwertig gestaltet werden, vollständig sein, eine Verfolgbarkeit einzelne Anforderungen und Änderungen sicherstellen, modifizierbar und erweiterbar sein eine gemeinsame Zugreifbarkeit aller Stakeholder sicherstellen sowie bezüglich des gewählten Vorgehens der Anforderungsermittlung optimiert sein. Rupp definiert eine Reihe analoger Qualitätsmerkmale, die über jene der einzelnen Anforderungen hinausgehen (Rupp 2007, 32):

- *Angemessener Umgang und klare Struktur:* Die Anforderungsspezifikation sollte im Umfang begrenzt und nach Kriterien sortiert sein.
- *Sortierbarkeit:* Die Anforderungsspezifikation sollte es ermöglichen, Anforderungen nach verschiedenen Kriterien zu sortieren.
- *Qualitativ hochwertig:* Eine Anforderungsspezifikation sollte nur aus qualitativ hochwertigen Einzelanforderungen bestehen.
- *Vollständigkeit:* Eine Anforderungsspezifikation muss alle relevanten Anforderungen bezüglich Funktionalität, Qualität, usw. beinhalten sowie alle zugehörigen Grafiken, Diagramme und Tabellen umfassen.
- *Eindeutigkeit und Konsistenz:* Eine Anforderungsspezifikation sollte nur eindeutige und konsistente Einzelanforderungen beinhalten.
- *Traceability (Verfolgbarkeit):* In eine Anforderungsspezifikation sollten alle Zusammenhänge nachvollziehbar sein.
- *Modifizierbarkeit und Erweiterbarkeit:* Anforderungsspezifikationen sollten leicht modifizierbar und erweiterbar sein.
- Gemeinsam Zugreifbar: In einer Anforderungsspezifikationen sollte der Autor jedes Eintrags vermerkt werden. Gegenseitiges Überschreiben muss verhindert werden und der Zugriff auf die Dokumente sollte nur autorisierten Personen gestattet sein.
- *Optimiert bezüglich des gewählten Vorgehens:* Die Anforderungsspezifikation sollte so optimiert werden, dass sie den Zweck der Weiterverwendung in den anschließenden Phasen am besten erfüllt. (Detailierungsgrad, Notation, …)

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

3.2.3.2.3 Strukturierung von Anforderungsdokumenten

Sowie es für jede einzelne Anforderungen verschiedenste Techniken gibt diese zu dokumentieren, so gilt dies natürlich auch für die alle Anforderungen umfassende Gesamtspezifikation. Durch ein strukturiertes Anforderungsdokument, kann die Änderbarkeit, Erweiterbarkeit und Lesbarkeit der Anforderungen erleichtert werden. Ein Einsatz der gleichen Referenzstruktur in unterschiedlichen Projekten erleichtert es den Stakeholdern, sich in diesen Dokumenten zurechtzufinden oder verschiedene Spezifikationen miteinander zu vergleichen (Pohl 2008, 251). Die Literatur schlägt eine Reihe von Referenzstrukturen vor, die erprobtes Expertenwissen für zukünftige Projekte nutzbar machen. Zu den zahlreiche Gliederungen für Anforderungsspezifikationen, die ein einheitliches Gliederungsschema vorgeben, zählen:

- DIN 66272[6] (Deutschen Institut für Normung 1994)
- ISO 9126 (International Organization for Standardization 2000)
- Volere (Robertson/Robertson 1999)
- Gliederung nach Partsch (Partsch 2010)
- IEEE1233-1998 (IEEE Standards Board 1998a, 20ff.)
- IEEE 830-1998 (IEEE Standards Board 1998b, 11)

1. Introduction
   - 1.1. Purpose
   - 1.2. Scope
   - 1.3. Definitions, acronyms and abbreviations
   - 1.4. References
   - 1.5. Overview
2. Overall description
   - 2.1. Product perspective
   - 2.2. Product functions
   - 2.3. User characteristics
   - 2.4. Constraints
   - 2.5. Assumptions and dependencies
3. Specific requirements
   - 3.1. External interface requirements
   - 3.2. Functional requirements
   - 3.3. Performance requirements
   - 3.4. Design constraints
   - 3.5. Software system attributes
   - 3.6. Other requirements
4. Supporting information
5. Appendixes
6. Index

Abbildung 3-3 Struktur Anforderungsspezifikation nach IEEE 830-1998
Quelle: Eigene Darstellung in Anlehnung an (IEEE Standards Board 1998b, 11)

Laut Pohl (Pohl 2008, 252) basieren die am weitesten verbreiteten Strukturierungen von Anforderungsdokumenten auf den beiden IEEE Standards IEEE 1233-1998 (IEEE Standards

[6] Die Norm wurde im Mai 2006 durch das DIN ersatzlos zurückgezogen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Board 1998a) und IEEE 830-1998 (IEEE Standards Board 1998b). Abbildung 3-3 zeigt eine Gliederung, die nach IEEE 830-1998 (IEEE Standards Board 1998b, 11) als Kapitelstruktur vorgeschlagen wird. Eine detaillierte Beschreibung der Aufgabeninhalte der einzelnen Kapitel und Unterkapitel findet sich bei IEEE 830-1998 (1998b, 11), Pohl (2008, 252) und Rupp (2007, 389).

3.2.3.3 Anforderungsprüfung (Verifikation und Validierung)

Ausgehend von den erhobenen, analysierten und dokumentierten Anforderungen folgt eine Prüfung des Anforderungsdokuments auf Konsistenz, Vollständigkeit und Genauigkeit (Kotonya/Sommerville 1998, 86). Ziel dieser Prüfung ist es festzustellen, ob genau die Anforderungen dokumentiert wurden, die die Vorstellungen und Wünsche der Stakeholder an das Produkt beschreiben und ob diese den notwendigen Qualitätsanforderungen genügen. Eine Prüfung der Anforderungen sollte erst durchgeführt werden, wenn diese einen gewissen Grad an Vollständigkeit und Stabilität erreicht haben. In jedem Fall muss eine Überprüfung der Anforderungen durchgeführt sein, bevor mit der Umsetzung dieser begonnen werden kann. Fehler im Anforderungsdokument führen zu immensen Nachbearbeitungskosten: „Anforderungsfehler sind für 70-85 % der Nachbearbeitungskosten verantwortlich“ (Wiegers 2003, 13).

Im Idealfall sollte die Anforderungsspezifikation nicht nur durch eine einzelne Person, sondern vielmehr durch eine ganze Gruppe an Personen durchgeführt werden. Rupp sieht hier Anwender, fachlichen Verantwortliche, Analytiker, Entwickler, Tester, Projektleiter sowie projektexterne Personen als geeignete Kandidaten (Rupp 2007, 306).

Grundsätzlich können im Kontext des Prüfens von Anforderungen zwei Aktivitäten ausgemacht werden; Verifikation und Validierung. Verifikation von Anforderungen befasst sich damit, Anforderungsdokumente hinsichtlich bestimmter Qualitätskriterien, wie beispielsweise in Abschnitt 3.2.3.1.4 beschrieben, zu untersuchen. Eine Feststellung und Überprüfung hinsichtlich inhaltlichen Vorstellungen der Stakeholder erfolgt im Rahmen der Validierung der Anforderungen (Partsch 2010, 47). Im IEEE Standard 610.12-1990 werden die beiden Aktivitäten wie folgt definiert (IEEE Standards Board 1990, 80):

- “verification. The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. “
- “validation. The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements.”

Eine Verifikation und Validierung von Anforderungen wird unter Beteiligung der relevanten Personengruppen im Rahmen von Reviews durchgeführt. “A process or meeting during which a work product or set of work products, is presented to project personnel, managers, users, customers, or other interested parties for comment or approval” (IEEE Standards Board 1990, 64). Die Literatur unterscheidet zwischen verschiedenen Arten von Reviews, abhängig von der Art wie dieser durchgeführt werden (Rupp 2007; Kotonya/Sommerville 1998; Ebert 2005; Pohl 2008):

- *Stellungnahme:* Bei der Stellungnahme wird das Anforderungsdokument einem Kollegen zum Lesen übergeben, der Auffälligkeiten markierte. Anhand der Anmerkungen des Kollegen können sich Änderungen ergeben, die in das Dokument aufgenommen werden.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *Walkthrough:* Bei einem Walkthrough wird das Anforderungsdokument schrittweise zusammen mit dem jeweiligen Prüfer durchgegangen. Aufkommende Fragen und Gedanken zu einzelnen Anforderungen können so direkt bilateral besprochen und gegebenenfalls in das Anforderungsdokument übernommen werden.
- *Inspektion:* Im Gegensatz zu den beiden eher informellen Techniken Stellungnahme und Walkthrough, folgt die Inspektion einem fest vorgegebenen Prozess. Zunächst wird vom Inspektionsleiter geprüft, ob sich das Dokument in einem ausreichenden Zustand für eine Inspektion befindet. Anschließend wird ein konkreter Plan erstellt, der dem Prüferkreis, relevante Dokumente und Termine enthält. Auf der Grundlage dieses Prüfplans erstellen die einzelnen Prüfer nun ihre individuellen Anmerkungen, die im letzten Schritt, der Nachbearbeitung, vom Inspektionsleiter in das Anforderungsdokument übertragen werden.
- *Prototypen:* Prototypen werden eingesetzt, um ausgewählte Aspekte frühzeitig zu überprüfen und so neue oder verbesserte Anforderungen zu erhalten (Andriole 1994; Schrage 2004). Auch können sie eingesetzt werden ganze Systembereiche zu evaluieren (Sharp/Rogers/Preece 2007; Te'eni/Carey/Zhang 2006) oder Alternativen zu bewerten (Andriole 1994).

3.2.4 Diskussion

Im Angesicht der zahlreichen Forschungsarbeiten rund um das Requirements Engineering könnte man meinen, dass dieses Forschungsfeld umfassend betrachtet wurde und somit eine ausreichende Lösung für die Praxis bereitstellt. „Trotz enormer Anstrengungen […] im Bereich des Software Engineering sind Planung und Realisierung umfangreicher Softwaresysteme […] mit technischen und wirtschaftlichen Risiken verbunden, die sehr häufig in einer nicht verstandenen Problemstellung ihre Ursache haben“ (Partsch 2010, 14).

Aufgrund der Heterogenität der an diesem Prozess beteiligten Stakeholder sowie der Prozesse, Techniken und Methoden mit denen man im Verlauf des Requirements Engineering in Berührung kommt, verbleiben noch eine Reihe an grundlegende Problemstellungen die auch für den Bereich der Automotive Services eine stete Herausforderung darstellen. „Je komplexer eine Aufgabe ist, desto schwieriger ist es, sie präzise und vollständig zu beschreiben, um so klare Zielvorgaben zu haben“ (Partsch 2010, 16). Rupp identifiziert insgesamt acht typische Probleme des Requirements Engineering die ungeachtet der eingesetzten Prozesse, Methoden und Werkzeuge eine Herausforderung in diesem Bereich darstellen (Rupp 2007, 16).

- *Unklare Zielvorstellungen:* Software, und somit auch Automotive Services, werden von verschiedenen Personengruppen genutzt die unterschiedliche Produktmerkmale verwenden und sich hinsichtlich der Benutzungshäufigkeit sowie Bildungs- und Erfahrungsniveau unterscheiden. Eine frühe Identifikation all diese Stakeholder ist eine Grundvoraussetzung für die Erstellung vollständiger Zielvorstellungen.
- *Hohe Komplexität:* Zu entwickelnde Software gewinnt zunehmend an Komplexität, was folglich auch zu immer komplexeren Anforderungen führt. Die dadurch steigende Vielfältigkeit der wechselseitigen Abhängigkeiten einzelner Anforderungen zueinander verschärft dieses Problem zudem.
- *Sprachbarrieren:* Aufgrund ihrer unterschiedlichen Kompetenzprofile sowie ihrem oftmals auch stark unterschiedlichen kulturellen Hintergründe, fällt es Stakeholdern oftmals schwer, die gleiche Sprache zu sprechen. Durch unterschiedliche

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Formulierungen desselben Sachverhalts oder gleicher Formulierungen für unterschiedliche Sachverhalte entstehen Kommunikationsprobleme. Stakeholder aus verschiedenen Sprachräumen verschärfen dieses Problem zusätzlich.

- *Veränderliche Anforderungen:* Durch sich ändernde Rahmenbedingungen und einen Erkenntnisgewinn der Stakeholder im Projektverlauf entstehen naturgemäß stetig neue Anforderungen oder existierende Anforderungen werden verändert. Gerade in Projekten, in denen Anforderungen kontinuierlich verwaltet werden, kann dies zu großen Problemen führen.
- *Schlechte Qualität der Anforderungen:* Wie in Abschnitt 3.2.3.1.4 und Abschnitt 3.2.3.2.2 bereits diskutiert, ist eine ausreichende Qualität der Anforderungen von zentraler Bedeutung. So dürfen diese weder redundant sein noch eine widersprüchliche Interpretation zulassen. Eine schlechte Qualität der Anforderungen führt zu falsch oder schlecht umgesetzten Ergebnissen, die in keiner Weise den Ansprüchen und Erwartungen der Stakeholder entsprechen.
- *Unnötige Merkmale:* Neben widersprüchlichen und schlecht formulierten Anforderungen sind Anforderung die unnötige Merkmale beschreiben ein weiteres großes Problem im Requirements Engineering. So schaffen sie zusätzlichen Aufwand bei der Umsetzung oder führen gar zu einem, für den Kunden nicht mehr richtig einsetzbaren, Produkt.
- *Ungenaue Planung:* Unzureichend formulierte Anforderungen haben oft zur Folge, dass die Komplexität eines Projekts nicht erkannt wird. Dies führt zu erheblicher Verlängerung der Projektlaufzeit und erhöht somit die Kosten des Projekts.

3.3 Requirements Engineering in der Praxis

In den vorangegangenen Abschnitten wurde, ausgehend von einer Betrachtung der für die Gestaltung von Automotive Services relevanten Anspruchsgruppen und deren unterschiedliche Kompetenzprofile, das Forschungsfeld des Requirements Engineerings als zentrales Gestaltungsinstrument identifiziert und diskutiert. Wie sich bei der Betrachtung des aktuellen Standes der Literatur zeigt, gibt es in diesem Bereich eine Vielzahl an Vorgehensmodellen, Methoden und Werkzeugen. Man stellt jedoch schnell fest, dass es eine ganze Reihe an Herausforderungen und Probleme bei der Erhebung und Gestaltung von Anforderungen existieren. Aus diesem Grund wurden im Rahmen einer Vorstudie Experten aus der Praxis zu ihren Erfahrungen und Einschätzungen befragt.

Analog des von Hug vorgeschlagenen Vorgehens, gliedert sich die Vorstudie in folgende Bereiche (Hug 2001, 24). Nach einer Diskussion der Ziele der Vorstudie werden methodische Grundlagen der empirischen Sozialforschung, die das methodische Fundament der Studie bilden, betrachtet. Aufbauend darauf, erfolgt die Darstellung der Vorbereitung und Planung, der Erhebung, der Aufbereitung, sowie die Auswertung und Diskussion der gewonnen Erkenntnisse.

3.3.1 Ziele der Vorstudie

Ziel dieser Vorstudie ist es, neben dem bereits diskutierten aktuellen Stand der Literatur auch die etablierten Vorgehensweisen und Herausforderungen aus der Praxis zu betrachten. Zentrale Fragestellung ist, wie Unternehmen existierende Vorgehensmodelle, Methoden und Werkzeuge einsetzen, in welchen Bereichen sie weiteren Entwicklungsbedarf sehen und wo aus ihrer praktischen Erfahrung heraus die größten Schwierigkeiten und Probleme liegen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Zu diesem Zweck wurden Unternehmen aus den Bereichen Softwareentwicklung, Medizintechnik sowie der Automobilindustrie befragt. Neben der für diese Arbeit wichtigen Automobil- und Softwareindustrie wurden zusätzlich vier Unternehmen aus den Bereichen Medizintechnik in die Untersuchung mit aufgenommen. Auch wenn die Medizintechnik keinen offensichtlichen und unmittelbaren Bezug zu Automotive Services hat, so können doch einige Parallelen entdeckt werden. Analog der Automobilindustrie sieht sich die Medizintechnik der Herausforderung gegenüber, Softwareanwendungen und Dienste für einen sehr speziellen Nutzungskontext zu entwickeln, in dem ein sehr hohes Maß an Sicherheitsbestimmungen und -anforderungen existieren.

3.3.2 Grundlagen Empirischer Sozialforschung

„Empirische Sozialforschung ist die systematische Erfassung und Deutung sozialer Erscheinungen“ (Atteslander/Cromm/Grabow 2006, 4). „Empirisch“ bedeutet dabei, dass theoretisch formulierte Annahmen an spezifischen Wirklichkeiten überprüft werden. „Systematisch“ weist darauf hin, dass dieser Vorgang nach bestimmten Regeln vor sich gehen muss (Atteslander/Cromm/Grabow 2006, 4f).

Die Wissenschaft bedient sich je nach Disziplin verschiedener Methoden zur Gewinnung neuer Erkenntnisse. Auch in der Sozialwissenschaft gibt es eine Vielzahl von Techniken zur Erhebung, Erfassung und Auswertung von Daten. Diekmann bezeichnet die Gesamtheit dieser Techniken als „Methodenvielfalt“ (Diekmann 2007, 18). Die Methoden der empirischen Datenerhebung haben die Funktion, Ausschnitte der Realität, die für eine Untersuchung interessant sind, möglichst genau zu beschreiben oder abzubilden (Bortz/Döring 2006, 137).

3.3.2.1 Untersuchungsarten der empirischen Sozialforschung

In empirischen Human- und Sozialwissenschaften unterscheidet man eine Vielzahl von Untersuchungsarten. Grundsätzlich wird zwischen qualitativer und quantitativer Forschung unterschieden. In der qualitativen Forschung werden die Daten zunächst verbalisiert, wohingegen sie in der quantitativen Forschung numerisch beschrieben werden. Man unterscheidet jedoch nicht nur das zu verarbeitende Datenmaterial, sondern auch die Forschungsmethoden, den Gegenstand und das Wissensverständnis (Bortz/Döring 2006, 295).

Die quantitative Forschung orientiert sich am naturwissenschaftlichen Forschungsverständnis und basiert auf numerischen Daten. Beim quantitativen Ansatz wird eine Quantifizierung bzw. Messung von Ausschnitten der Beobachtungsrealität durchgeführt und die daraus resultierenden Messwerte anschließend statistisch verarbeitet (Bortz/Döring 2006, 295). Das Methodenspektrum umfasst standardisierte Befragungstechniken, schematisierte Beobachtungsformen, inhaltsanalytische und statistische Verfahren, experimentelle Vorgehensweisen, Tests und Skalierungsverfahren (Hug 2001, 22). Quantitative Methoden eignen sich gut um Hypothesen und Theorien zu überprüfen (Bortz/Döring 2006, 137f.).

Beim qualitativen Ansatz wird die Erfahrungswirklichkeit verbalisiert und anschließend interpretativ ausgewertet. Im Gegensatz zu den quantitativen Verfahren wird dabei auf Messungen und Zahlen verzichtet. Stattdessen bedient man sich qualitativen Materials, welches meistens aus Texten wie z.B. Beobachtungsprotokollen, Interviewtexten und Briefen oder aus anderen Objekten wie z.B. Zeichnungen, Fotos und Kleidungsstücken besteht. Dieses qualitative Material enthält viel mehr Informationen als ein Messwert (Bortz/Döring 2006, 296). Das Methodenspektrum umfasst eine Reihe von Interviewformen, Gruppen-diskussionsverfahren, (nicht-) teilnehmende Beobachtungsvarianten, ethnographische

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Vorgangsweisen, inhaltsanalytische Verfahren und qualitative Experimente (Hug 2001, 22). Klassifiziert man empirische Untersuchungen nach ihrer Zielsetzung, so unterscheidet man zwischen explorativen, explanativen und deskriptiven Untersuchungen (Bortz/Döring 2006, 360).

Die explorative oder erkundende Forschung ist besonders bei einem wenig bekannten Forschungsgegenstand hilfreich. Diese Art der Forschung kann eingesetzt werden, um einen ersten Einblick in einen bestimmten Bereich zu bekommen. Explorationsstudien sind eine Phase im Forschungsprozess, die auf die Entwicklung wissenschaftlich prüfbarer Hypothesen abzielt. Die explorative Forschung ist eine Vorstudie zur Vorbereitung einer darauf folgenden genaueren Untersuchung. Zu diesem Zweck wird sie oft den explanativen Untersuchungen vorgeschaltet (Bortz/Döring 2006, 360).

„Explanative Untersuchungen dienen der Prüfung von Theorien und Hypothesen“ (Bortz/Döring 2006, 360). „Wissenschaftliche Hypothesen sind Annahmen über reale Sachverhalte […]. Sie weisen über den Einzelfall hinaus […] und sind durch Erfahrungsdaten widerlegbar[...]“(Bortz/Döring 2006, 8). Im Gegensatz zu explorativen Untersuchungen erfordern hypothesenprüfende Untersuchungen Vorkenntnisse die es ermöglichen, vor Beginn der Untersuchung präzise Hypothesen zu formulieren. Bei dieser Art der Untersuchung werden Annahmen über Zusammenhänge, Unterschiede und Veränderungen ausgewählter Merkmale getestet. Das Ziel dieser Untersuchung ist es festzustellen, ob die Untersuchungsergebnisse die Hypothesen untermauern oder den Hypothesen widersprechen. Aus den Ergebnissen sollen Erklärungen und Prognosen ableitbar sein (Bortz/Döring 2006, 491).

Deskriptive Untersuchungen dienen der Beschreibung von Populationen, Phänomenen, Einzelfällen und Stichproben (Bortz/Döring 2006, 360). Bei dieser Art der Untersuchung geht es um die Beschreibung von Grundgesamtheiten oder Populationen auf Basis von Stichprobenergebnissen. Dabei ist eine Grundgesamtheit zu definieren, die hinsichtlich eines oder mehrerer Merkmale zu beschreiben ist (Bortz/Döring 2006, 370).

3.3.2.2 Methoden der Empirischen Sozialforschung

Abhängig vom Ziel und von der Fragestellung einer Untersuchung gibt es unterschiedliche Methoden zur Gewinnung von Daten. „Je nach Forschungsinteresse kommen unterschiedliche Gegenstandsbereiche in Betracht und damit auch unterschiedliche Methoden […]“ (Atteslander/Cromm/Grabow 2006, 48). Grundsätzlich gibt es vier Methodengruppen: die Inhaltsanalyse, die Beobachtung, die Befragung und das Experiment (Atteslander/Cromm/Grabow 2006, 48). Diese Methoden sind sowohl in qualitativen als auch in quantitativen Ausprägungen beschrieben worden.

„Mittels Inhaltsanalysen lassen sich Kommunikationsinhalte wie Texte, Bilder und Filme untersuchen, wobei der Schwerpunkt auf der Analyse von Texten liegt“ (Atteslander/Cromm/Grabow 2006, 181). Sie dient zum einen der primären Datengewinnung und zum anderen kann sie zur Verarbeitung von erhobenen Daten wie z.B. Protokollen eingesetzt werden. Das Ziel der Inhaltsanalyse ist es, die zentralen Themen und Bedeutungen von Texten und anderen Objekten herauszuarbeiten und aus den Merkmalen auf Zusammenhänge seiner Entstehung und Verwendung zu schließen. Dabei kann das Material mittels qualitativer oder quantitativer Inhaltsanalyse ausgewertet werden (Atteslander/Cromm/Grabow 2006, 182).

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

„Unter Beobachtung verstehen wir das systematische Erfassen, Festhalten und Deuten sinnlich wahrnehmbaren Verhaltens zum Zeitpunkt seines Geschehens“ (Atteslander/Cromm/Grabow 2006, 67). Bei der Beobachtung verhält sich das forschende Subjekt weitgehend passiv und versucht in einem nichtkommunikativen Prozess mit Hilfe sämtlicher Wahrnehmungsmöglichkeiten Erfahrungen zu sammeln. Sie bedient sich Instrumenten, die die Selbstreflektiertheit, Systematik und Kontrolliertheit der Beobachtung gewährleisten. Beobachtung kann sowohl qualitativ als auch quantitativ betrieben werden. Sie kann quantitative Daten produzieren, die zu einer statistischen Hypothesenprüfung geeignet sind oder sie kann explorativ eingesetzt werden, um einen interpretativen Zugang zu einem Beobachtungsfeld zu schaffen (Bortz/Döring 2006, 262).

„Befragung bedeutet Kommunikation zwischen zwei oder mehreren Personen. Durch verbale Stimuli (Fragen) werden verbale Reaktionen (Antworten) hervorgerufen: Dies geschieht in bestimmten Situationen und wird geprägt durch gegenseitige Erwartungen. Die Antworten beziehen sich auf erlebte und erinnerte soziale Ereignisse, stellen Meinungen und Bewertungen dar“ (Atteslander/Cromm/Grabow 2006, 101). Die Befragung ist die in der empirischen Sozialforschung am häufigsten angewendete Methode. Man kann zwischen der mündlichen Befragung in Form von Interviews und der schriftlichen Befragung in Form von Fragebögen unterscheiden (Bortz/Döring 2006, 237).

Atteslander definiert ein Experiment als: „eine wiederholbare Beobachtung unter kontrollierten Bedingungen; dabei werden eine bzw. mehrere unabhängige Variablen so manipuliert, das eine Überprüfungsmöglichkeit der zugrunde liegenden Hypothese, d.h. der Behauptung eines Kausalzusammenhangs, in unterschiedlichen Situationen gegeben ist“ (Atteslander/Cromm/Grabow 2006, 167). Man kann zwischen Feldexperiment und Laboratoriumsexperiment unterscheiden. Bei Laboratoriumsexperimenten wird eine Experimentalgruppe in einer künstlichen Situation untersucht, wohingegen der zu untersuchende Gegenstand bei einem Feldexperiment nicht aus seiner natürlichen Umgebung herausgelöst wird (Atteslander/Cromm/Grabow 2006, 168).

3.3.2.3 Formen der Befragung

Generell kann man verschiedene Formen der Befragung nach der verwendeten Kommunikationsart und dem Grad der Strukturierung der Kommunikationsform unterscheiden. Atteslander unterscheidet sieben verschiedene Typen der Befragung (Atteslander/Cromm/Grabow 2006, 123ff.): wenig strukturierte, teilstrukturierte und stark strukturierte Interviews sowie wenig strukturierte, teilstrukturierte und stark strukturierte schriftliche Befragung und eine Kombination aus mündlicher und schriftlicher Befragung. Wenig strukturierte und teilstrukturierte Befragungstechniken eignen sich zur Erhebung qualitativer Daten, wohingegen die Erhebung quantitativer Daten den Einsatz von stark strukturierten Befragungen erfordert. Denn eine Quantifizierung der Daten ist nur möglich, wenn standardisierte Fragen gestellt werden, deren Antworten vergleichbar sind.

Wenig strukturierte und teilstrukturierte Befragungen werden nur sehr selten schriftlich durchgeführt, da die Probanden eher zu mündlichen Äußerungen bereit sind als zu schriftlichen Ausarbeitungen eines Themas (Bortz/Döring 2006, 308). Mündliche Befragung wird in Form von Interviews durchgeführt, bei denen man den Grad der Strukturiertheit unterscheidet.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Bei wenig strukturierten Interviews arbeitet der Forscher ohne Fragebogen. Dadurch ergibt sich eine flexible Gesprächsführung, da der Forscher seine Fragen individuell an die befragte Person anpassen kann. Das Gespräch folgt nicht den Fragen des Forschers, sondern die jeweils nächste Frage ergibt sich aus dem Gespräch. Der Forscher hat die Aufgabe das Gespräch in Gang zu halten und die Meinung des Befragten zu erfassen, ohne seine Reaktionsmöglichkeit einzuschränken (Atteslander/Cromm/Grabow 2006, 124).

Anders verhält es sich bei stark strukturierten Interviews. Hier muss zunächst ein Fragebogen konstruiert werden, der die Freiheitsspielräume des Forschers und des Befragten stark einschränkt. Der Fragebogen legt den Inhalt, die Anzahl und die Reihenfolge der Fragen fest. Darüber hinaus wird bereits bei der Konstruktion des Fragebogens über die Formulierung der Frage und die zugehörigen Antwortkategorien entschieden. Diese Art der Befragung liefert vergleichbare Antworten, kann unabhängig vom Forscher und damit in großer Zahl durchgeführt werden. Sie eignet sich besonders zur Hypothesenprüfung (Atteslander/Cromm/Grabow 2006, 125).

Bei teilstrukturierten Interviews handelt es sich um Gespräche, die auf Basis vorbereiteter und vorformulierter Fragen stattfinden. Dabei ist die Reihenfolge der Fragen nicht festgelegt und der Forscher hat jederzeit die Möglichkeit interessante Themen aufzugreifen und sie weiter zu vertiefen. In der Regel wird ein Gesprächsleitfaden benutzt, um teilstrukturierte Interviews durchzuführen (Atteslander/Cromm/Grabow 2006, 125). Meistens werden dabei explorative Ziele im Rahmen der qualitativen Forschung verfolgt (Atteslander/Cromm/Grabow 2006, 129).

Das Leitfadeninterview ist die gängigste Form qualitativer Befragung. Durch den Leitfaden und die darin enthaltenen Fragen kann man Daten erheben. Da bei mehreren Interviews derselbe Leitfaden verwendet wird sind die Ergebnisse der Interviews einigermaßen vergleichbar. Dennoch bleibt dem Forscher die Möglichkeit spontan auf die Interviewsituation zu reagieren und neue Fragen zu stellen die nicht im Leitfaden stehen oder die Reihenfolge der Fragen zu verändern. Der Vorteil von Leitfadeninterviews ist die Flexibilität, da der Forscher frei in der Gestaltung des Interviews ist und bei interessanten Themen in die Tiefe gehen kann, indem er zusätzliche Fragen stellt. Dennoch verfügt er über einen Katalog mit vordefinierten Fragen, was die Ergebnisse einigermaßen vergleichbar macht (Bortz/Döring 2006, 315).

Das Experteninterview stellt eine Form des Leitfadeninterviews dar (Atteslander/Cromm/Grabow 2006, 132). Das Experteninterview ist ein Sammelbegriff für offene oder teilstrukturierte Befragungen von Experten zu einem vorgegebenen Bereich oder Thema (Bortz/Döring 2006, 314). Menschen, die über besonderes Wissen zu einem bestimmten Thema verfügen, werden als Experten bezeichnet. Dieses Wissen kann sich auf alle Bereiche des Lebens beziehen, dazu zählt auch Wissen über die Abläufe und Tätigkeiten in einem bestimmten Bereich eines Unternehmens, in dem die Befragten arbeiten. Mitarbeiter eines Unternehmens haben aufgrund ihrer Position und ihrer persönlichen Beobachtung und Mitarbeit ein spezielles Expertenwissen zu einem bestimmten Sachverhalt. Die Experten sind allerdings nicht das Objekt der Untersuchung, sie sind lediglich Zeugen und Wissensträger der den Forscher interessierenden Informationen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

3.3.3 Planung und Vorbereitung

Die Vorstudie basiert auf der Grundlage von qualitativen Leitfadeninterviews. Ein allgemeines Merkmal von qualitativen Interviews ist, dass die Frageformulierungen und die Reihenfolge der Fragen nicht vorab festgelegt werden, sondern je nach Situation angepasst werden können. Da es bei dieser Untersuchung nicht nur darum geht, die eingesetzten Techniken und Vorgehensweisen des Requirements Engineerings zu ermitteln, was auch mit einem strukturierten Fragebogen möglich gewesen wäre, sondern auch Probleme und deren Ursachen aufgedeckt werden sollen, erscheint diese Art der Befragung am zweckmäßigsten.

Der Leitfaden ist in diesem Sinne Orientierung für den Gesprächsverlauf. Er ermöglicht es dem Forscher spontan und individuell auf interessante Anmerkungen des Interviewten einzugehen und Problemfelder individuell aufzudecken sowie durch gezieltes Nachfragen Ursachen und Gründe für diese zu erforschen. Darüber hinaus bleibt eine weitestgehende Vergleichbarkeit der Interviews untereinander weiterhin bestehen.

Auf der Grundlage der diskutierten Literatur wurde zunächst ein Leitfaden für die Interviews entwickelt, der sich grob in zwei Bereiche unterteilen lässt. Der erste Fragenblock bezieht sich auf allgemeine Informationen zum Unternehmen sowie zu den Produkten und den Projekten zu denen der Interviewte Expertise besitzt. Ziel des ersten Fragenblocks ist einerseits, Hintergrundinformationen zum Experten zu halten, andererseits aber auch um eine für den Experten angenehme Gesprächssituation zu schaffen.

Im zweiten Fragenblock wurden schließlich die in der Literatur identifizierten Themengebiete betrachtet. Das erste Themengebiete beschäftigt sich mit dem übergeordneten Vorgehen des Requirements Engineering und erfragt ein ganzheitliches Verständnis des Experten zu diesem Gebiet. Ausgehend davon werden anschließend die Themengebieten Anforderungsermittlung und Analyse, Anforderungsdokumentation, Verifikation und Validierung von Anforderungen sowie der Einsatz von Prototypen besprochen. Tabelle 3-5 stellt den Interviewleitfaden tabellarisch dar.

Fragenblock	Themengebiet	Fragen
1. Fragenblock	Allgemeine Informationen zum Unternehmen	o Umsatz gesamt, Deutschland, in Mio. € o Mitarbeiter gesamt
	Produkte des Unternehmens	o Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten? o Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren o Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?
	Projekte des Unternehmens (ein bestimmtes Projekt)	o Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern? o Könnten Sie das Projekt kurz beschreiben? o Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

		○ Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch? ○ Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.
2. Fragenblock	Requirements Engineering	○ Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes? ○ Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung? ○ Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht? ○ Wer ist am Prozess der Verwaltung von Anforderungen beteiligt? ○ Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht? ○ Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu? ○ Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt? ○ Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...) ○ Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen? ○ Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert? ○ Von wem kommen die Änderungen? ○ Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem? ○ Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?
	Anforderungsermittlung und Anforderungsanalyse	○ Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)? ○ Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten? ○ Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen) ○ Werden die Anforderungen priorisiert? Falls ja, wie?
	Anforderungsdokumentation	○ Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren? ○ Welche Informationen werden dokumentiert? ○ Wird die Verwaltung von Anforderungen mit Werkzeugen

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

		unterstützt? Falls ja, mit welchen? Falls nein, warum nicht? o Verifikation und Validierung o Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden? o Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?
	Verifikation und Validierung	o Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden? o Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?
	Prototypen	o Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …) o Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt? o Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …) o Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein? o Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?

Tabelle 3-5 Aufbau des Interviewleitfaden
Quelle: Eigene Darstellung

Das Erstellen des Leitfadens wurde unter Einbeziehung von Testinterviewpartnern iterativ durchgeführt. Nach Fertigstellung des Leitfadens wurden Unternehmen aus den drei Bereichen Automotive, Medizintechnik und Softwareentwicklung identifiziert, die für die Studie infrage kommen. Insgesamt konnten fünf Interviewpartner aus drei Automobilunternehmen, acht Interviewpartner aus dem Bereich der Softwareentwicklung sowie vier Experten aus der Medizintechnik für die Studie gewonnen werden. Von den fünf Interviewpartnern der Automobilunternehmen waren jeweils zwei vom gleichen Hersteller, so dass insgesamt drei Automobilhersteller an der Studie beteiligt waren.

Branche	Größe	Umsatz 2007	Mitarbeiter 2007	Anzahl der Unternehmen	Anzahl der Interviews
Automotive	groß	ab 50 Mio €	500 u. mehr	3	5
Software	groß	ab 50 Mio €	500 u. mehr	5	5
	mittel	von 1 bis 50 Mio €	von 10 bis 499	3	3
Medizintechnik	groß	ab 50 Mio €	500 u. mehr	4	4

Tabelle 3-6 Klassifizierung der Studienteilnehmer
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Tabelle 3-6 gibt einen Überblick über die beteiligten Unternehmen, klassifiziert nach Branche, Größe und Anzahl der Interviews. Die Größe der Unternehmen ist ausgehend vom Jahresumsatz aus dem Jahr 2007 sowie der Anzahl der Mitarbeiter in die Kategorien, groß, mittel und klein unterteilt. Unter den 15 beteiligten Unternehmen, waren drei große Automobilhersteller, vier große Medizintechnikunternehmen, fünf große Softwareentwicklungsunternehmen sowie drei mittlere Softwareunternehmen.

Interviewpartner	Position des Interviewpartners	Art des Interviews	Größe des Unternehmens	Branche
Alpha	Geschäftsführer; IT-Consultant, Software Engineer	persönlich	mittel	Software
Beta	Executive Partner; Manager, an der Konzeption des Vorgehens für die Durchführung der Projekte beteiligt	persönlich	groß	Software
Gamma	PMO - Qualitätsmanagement FAZIT Projektteam IT FAZIT; Projektmanagement; zuständig für das Qualitätsmanagement; Mitglied der Projektmanagement-gruppe	persönlich	groß	Automotive
Delta	ERP Lösungen und Dienste; Verantwortliche für das Anforderungsmanagement aus der zentralen IT	persönlich	groß	Automotive
Epsilon	Verantwortlicher für das fachliche Anforderungsmanagement	persönlich	groß	Automotive
Zeta	Projektleiter, Business Development Manager	persönlich	groß	Medizintechnik
	Robotics & Platforms			

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Eta	IT Management Consulting	persönlich	groß	Software
Theta	Verantwortlicher für das Requirements & Usability Engineering	persönlich	groß	Automotive
Iota	Leiter Kompetenzgruppe RE; Mitglied des Fachgebiets Business Engineering	persönlich	groß	Automotive
Kappa	Manager; Principal Health Care	persönlich	groß	Software
Lambda	Strategic Account Manager	telefonisch	groß	Medizintechnik
My	Verantwortliche im Bereich IT-Beratung	telefonisch	groß	Medizintechnik
Ny	Managing Director	persönlich	groß	Software
Xi	Geschäftsführer	persönlich	mittel	Software
Omikron	Senior Manager; Leiter Business Area Supply Chain Execution (Senior Manager Business Area Supply Chain Execution (Leiter des Bereiches Beschaffung), Schnittstelle zu Requirements Engineering)	telefonisch	mittel	Software
Pi	Seniorberater; Manager, an der Konzeption des Vorgehens im Bereich des Requirements Engineering beteiligt	persönlich	groß	Software
Rho	Projektleiter Project Manager Surgery C-Arms	persönlich	groß	Medizintechnik

Tabelle 3-7 ***Übersicht Interviewpartner***
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Tabelle 3-7 Liste die einzelnen Interviewpartner anonymisiert auf. Der Tabelle ist die Position des Interviewpartners im Unternehmen, die Art wie das Interview durchgeführt wurde (persönlich, telefonisch) sowie die Größe und die Branche des Unternehmens zu entnehmen.

3.3.4 Datenerhebung

Die meisten der durchgeführten Interviews fanden vor Ort im Unternehmen des jeweiligen Befragten statt. Lediglich drei der Interviews wurden aus organisatorischen Gründen telefonisch durchgeführt. Während der Interviews hatten die Experten die Möglichkeit, Unterstützungsmaterialien wie Dokumente und Artefakte heranzuziehen. Die Interviews dauerten zwischen 44 und 92 Minuten, die durchschnittliche Länge eines Interviews Betrug dabei 73 Minuten.

Zu Beginn des Interviews fand eine Informations- und Begrüßungsphase statt, in der sich die Interviewpartner gegenseitig vorstellten sowie die Interviewten die Möglichkeit erhielten, Fragen zu Ziel und Ablauf des Forschungsvorhabens zu stellen. In diesem Rahmen wurde den Gesprächspartnern versichert, dass die erhobenen Daten und Informationen ausschließlich im Rahmen dieser Studie in anonymisierter Form verwendet werden. Des Weiteren wurden die Interviewten um ihr Einverständnis gebeten das Interview zum Zweck der späteren Auswertung aufzuzeichnen. Ausgenommen des Interviews Kappa konnten alle durchgeführten Interviews somit mitgeschnitten werden. Zusätzlich wurden Gesprächsnotizen während des Interviews angefertigt.

Wie bereits erwähnt, gliedert es sich der Fragebogen in zwei wesentliche Bereiche. Zunächst wurden die Interviewpartner gebeten sich selbst und ihre Rolle im Unternehmen zu beschreiben sowie im Verlauf des ersten Fragenblocks wesentliche Eckdaten zum Unternehmen zu benennen. Besonders wichtig war hier eine Beschreibung der Art der Projekte die im Bereich der Experten durchgeführt werden. Aus diesen Projekten sollten die Interviewten ein bestimmtes Projekt exemplarisch herausgreifen, an dessen Beispiel die Fragen aus Fragenblock 2 diskutiert wurden. Natürlich hatten die Interviewpartner die Möglichkeit, Informationen auch aus anderen Projekten oder allgemeinen Vorgaben des Unternehmens in die Beantwortung der Fragen mit einfließen zu lassen. Im Verlauf der Interviews hatte sich deutlich gezeigt, dass das Heranziehen eines konkreten Beispiels die Beantwortung der Fragen für die Experten oftmals wesentlich erleichtert.

Die einzelnen Fragen des zweiten Fragenblocks wurden anschließend in einer, für das jeweilige Interview individuellen, Reihenfolge behandelt. Die Interviewten hatte somit die Möglichkeit frei zu erzählen, ohne sich an eine vorgegebene Gliederung halten zu müssen. Das Interview wurde in Form eines natürlichen Gesprächs geführt, in dem die Befragten zu gegebenen Stichwörtern und Fragestellungen frei erzählen konnten. Der Fragebogen selbst, der auch den Interviewten vorab zur Verfügung gestellt wurde, diente im Wesentlichen dem Interviewer einen Überblick über die bereits behandelten sowie noch ausstehenden Fragen zu behalten. Eine offene Formulierung der einzelnen Fragen erleichterte es den Befragten, eine individuelle Antwort auf das jeweilige Themengebiet zu geben.

3.3.5 Aufbereitung der Daten

Typischerweise arbeitet man in der qualitativen Sozialforschung mit Stichproben von wesentlich kleinerem Umfang als in der quantitativen Sozialforschung (Diekmann 2007, 455).

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Hinsichtlich der Anzahl der durchgeführten Interviews, erscheint eine ausschließliche Quantifizierung des Datenmaterials daher nicht sinnvoll. Aus einer Stichprobe dieser Größenordnung lassen sich keine allgemeingültigen Aussagen aggregieren. Die Daten wurden daher thematisch zusammengefasst und interpretativ aufbereitet. Hierbei dienen die Themengebieten aus dem zweiten Fragenblock als Gliederung. Im nachfolgenden Abschnitt werden daher die Antworten der einzelnen Unternehmen hinsichtlich der Themengebiete des zweiten Fragenblocks diskutiert. Darüber hinaus wurden die zentralen Antworten der Interviews quantitativ zusammengefasst, um eine Gewichtung und Tendenz ausmachen zu können. Wichtig ist es hierbei zu erwähnen, dass diese Angaben keine statistische Relevanz besitzen und lediglich der besseren Darstellung der gewonnen Erkenntnisse dienen.

Da bei dieser Auswertung nicht die Einzelantworten einzelner Interviewpartner im Vordergrund stehen sondern ein Einblick in die Praxis entwickelt werden soll, werden die Aussagen aller Interviewpartner jeweils thematisch besprochen. Da die Fragen des ersten Fragenblocks der Einordnung des Interviewten sowie dessen Unternehmen dienen und darüber hinaus den zweiten Fragenblock vorbereiten, werden nachfolgend ausschließlich die Themengebieten des zweiten Fragenblocks behandelt.

Zu Aufbereitung des erfassten Materials wurden unter Zuhilfenahme der Tonbandaufnahmen sowie der während dem Interview gemachten Gesprächsnotizen, Mitschriften der Einzelinterviews angefertigt. Zur Gliederung dieser wurde wiederum der Interviewleitfaden herangezogen. Die Mitschriften spiegeln daher lediglich die inhaltlichen Aussagen der Experten zu den einzelnen Themengebieten und Fragestellungen wieder, ohne dabei auf sonstige prägnante Merkmale des Verlaufs, wie zum Beispiel Tonhöhe, Pausen, Lachen oder gleichzeitiges Sprechen einzugehen (Bortz/Döring 2006, 312). Da einerseits nicht alle Interviews aufgenommen werden konnten, sowie nonverbale und paraverbale Äußerungen für die spätere Interpretation der Untersuchung nicht von Bedeutung sind, wurde auf eine vollständige Transkription der Interviews verzichtet und auf das Instrument des schriftlichen Gesprächsprotokolls zurückgegriffen (Mayring 2007).

Da verschriftete Gespräche durch unvollständige Sätze, verschluckte Silben und umgangssprachliche Wendungen oft holprig und schlecht formuliert wirken (Bortz/Döring 2006, 312), wurde beim Anlegen der Gesprächsprotokolle der Text gelegentlich geglättet, ohne dabei die erfassten Inhalte zu verfälschen. Desweiteren wurden Aussagen der Interviewten, die in keinem Kontext zum Interviewleitfaden standen, weggelassen. Schließlich wurden die Gesprächsprotokolle anonymisiert. Die vollständigen Gesprächsprotokolle der Interviews können im Anhang A nachgelesen werden. In Anhang B finden sich darüber hinaus die Aussagen der einzelnen Interviews gegliedert nach den Fragestellungen des Interviewleitfadens.

3.3.6 Untersuchungsergebnisse

Nachfolgend werden die Aussagen der 17 Interviewpartner zu den fünf Themengebieten Requirements Engineering, Anforderungsermittlung und Anforderungsanalyse, Anforderungsdokumentation, Verifikation und Validierung sowie Prototyping besprochen. Die jeweiligen Aussagen der einzelnen Interviewpartner können im Anhang B nachgeschlagen werden.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

3.3.6.1 Requirements Engineering

Im folgenden Teil werden die Aussagen der Unternehmen zum Thema Requirements Engineering zusammengefasst. Wie dem Interviewleitfaden zu entnehmen ist, wurden die Gesprächspartner über den Ablauf, die Dokumentation sowie die Verantwortlichkeiten im Prozess befragt.

Zunächst sollten die Interviewten darstellen, ob ihr Unternehmen Anforderungen nur am Anfang eines Projekts erhebt oder diese kontinuierlich entlang der Projektphasen verwaltet. Die Mehrheit der befragten Unternehmen gab hier an, dass Anforderungen entlang der gesamten Projektdauer verwaltet werden (11). Drei der befragten Unternehmen erfassen Anforderungen lediglich zu Beginn eines Projekts, wobei zwei dieser Unternehmen einen definierten Change Management Prozess haben um auf spätere Änderungen der Anforderungen eingehen zu können. Ein Unternehmen verwaltet Anforderungen lediglich zu Beginn und am Ende des Projekts, nicht aber dazwischen und ein Unternehmen gab an, dass diese Entscheidung projektspezifisch fällt, Anforderung jedoch meist kontinuierlich verwaltet werden. Ein Interviewpartner gab keine Antwort auf diese Frage.

Auf die Frage nach dem Prozess der Verwaltung der Anforderungen, insbesondere der einzelnen Phasen des Verwaltungsprozesses, können die Antworten der Unternehmen in zwei Gruppen klassifiziert werden. 10 der befragten Unternehmen haben detailliert die Phasen ihres Verwaltungsprozesses berichtet, die anderen sieben machten lediglich allgemeine Aussagen über Existenz, Eigenschaften und Werkzeuge der Unterstützung ihres Requirements Engineering Prozesses.

So haben die Unternehmen der zweiten Gruppe angegeben, dass Anforderungen in Form einer Liste, meist einer Excel-Liste, dokumentiert werden. Ein Änderungsmanagement wird meist mithilfe eines Ticketing-System durchgeführt. Das Unternehmen, das hier projektspezifisch vorgeht, gab an sich an klassischen Vorgehensmodellen, wie vormals in der Literatur diskutiert, zu orientieren.

Phase Literatur	Genannte Phase	Anzahl der Angaben
Anforderungserhebung	o Aufschreiben der Anforderungen o Grobe Beschreibung o Erheben	8
Interpretation und Strukturierung	o Interpretation und Umformulierung o Analyse o Aufwand Evaluierung o Priorisierung und Bewertung o Verständliches Aufschreiben o Ausformulierung	9
Änderungsmanagement	o Änderung	5
Validation und Verifikation		7
Anforderungsverfolgung	o Review	4
Dokumentation	o Eintrag in Lastenheft	5

Tabelle 3-8 ***Angaben zu den Phasen des Requirements Engineering Prozesses***
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Die Antworten der ersten Gruppe, wurden zunächst gesammelt und anhand der Phasen des Requirements Engineering aus der Literatur strukturiert. Anschließend wurde gezählt, wie viele der Unternehmen aus der ersten Gruppe welche Phasen des Prozesses für wichtige erachten und durchführen. Die Ergebnisse sind in Tabelle 3-8 dargestellt. Abbildung 3-4 stellt die gefundene Gewichtung der einzelnen Phasen nochmals grafisch dar.

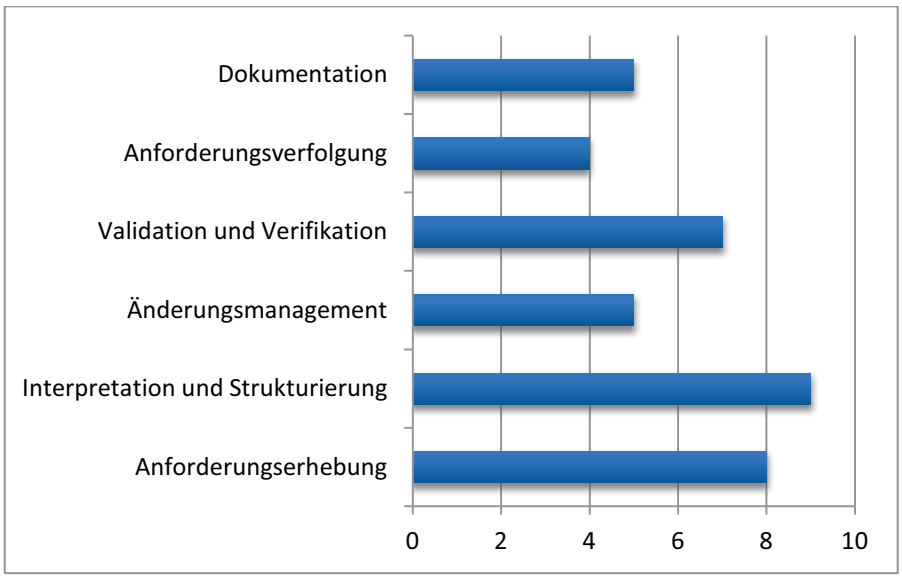


Abbildung 3-4 Gewichtung der Phasen des Requirements Engineering Prozesses
Quelle: Eigene Darstellung

In der nächsten Frage wurde geklärt aus welchen unterschiedlichen Quellen Anforderungen kommen und ob diese, abhängig von ihrer Herkunft, unterschiedlich behandelt werden. Die Mehrzahl der Unternehmen gab als Quelle von Anforderungen den Kunden an. An zweiter Stelle wurden am häufigsten interne Abteilungen genannt, die sich unmittelbar mit der Projektrealisierung befassen. Die dritte Gruppe repräsentieren die Produktendanwender, die jedoch häufig nicht von Kunden unterscheidbar sind. Des Weiteren können Anforderungen noch von anderen Fachbereichen, Lieferanten, Management, Vertrieb oder Serviceabteilungen sowie dem Gesetzgeber stammen. Wie Abbildung 3-5 zu entnehmen ist, bildet die Gruppen der Kunden und Produktendanwender die mit Abstand größte Quelle an Anforderungen.

Auf die Frage nach den Beteiligten am Verwaltungsprozess haben die Befragten sehr unterschiedliche Antworten gegeben. Einige der Unternehmen haben nur die dafür zuständigen Teams genannt (in sieben Fällen das Projektteam beziehungsweise Ausführungsteam, in einem Fall das Marketingteam, sowie in zwei Fällen eine spezialisierte Zentralstelle). Die restlichen Unternehmen haben auf diese Frage zuständige Rollen der einzelnen Teammitglieder angegeben. Hier würden der Produktmanager/Projektleiter sowie Anforderungsanalysten und Reviewer genannt. In einem Fall wurde explizit der Kunde, unterstützt durch das Unternehmen, als Hauptverantwortlicher für die Verwaltung der Anforderungen angegeben. Ein Unternehmen machte zu dieser Frage keine Angabe.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

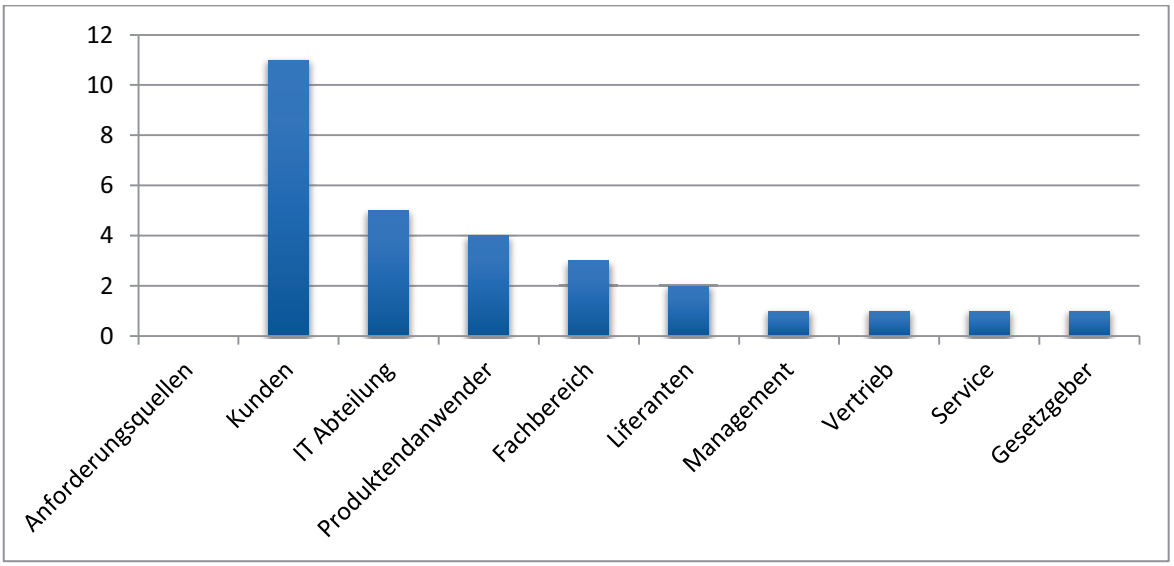


Abbildung 3-5 Quellen von Anforderungen
Quelle: Eigene Darstellung

Da Anforderungen oft in einer speziellen Fachsprache definiert werden und die am Prozess Beteiligten unterschiedliche Domänen sowie unterschiedliches Fachwissen besitzen, Erfolg bei den meisten der befragten Unternehmen eine Übersetzung (Vereinheitlichung) der erhobenen Anforderungen. Laut den Angaben der Befragten werden Anforderungen meistens in einer Kooperation zwischen dem Anforderungsteam (Entwickler) und den beteiligten Fachabteilungen beziehungsweise Kunden durchgeführt. Zwei der befragten Unternehmen haben angegeben, dass Anforderungen von Anfang an in natürlicher Sprache verfasst werden und somit keine Übersetzung notwendig ist. Zwei der Unternehmen haben in ihren Projekten spezialisierte Anforderungsmanagementteams, die auch die Übersetzung durchführen. In zwei der befragten Unternehmen wird diese Aufgabe vom Projektleiter übernommen.

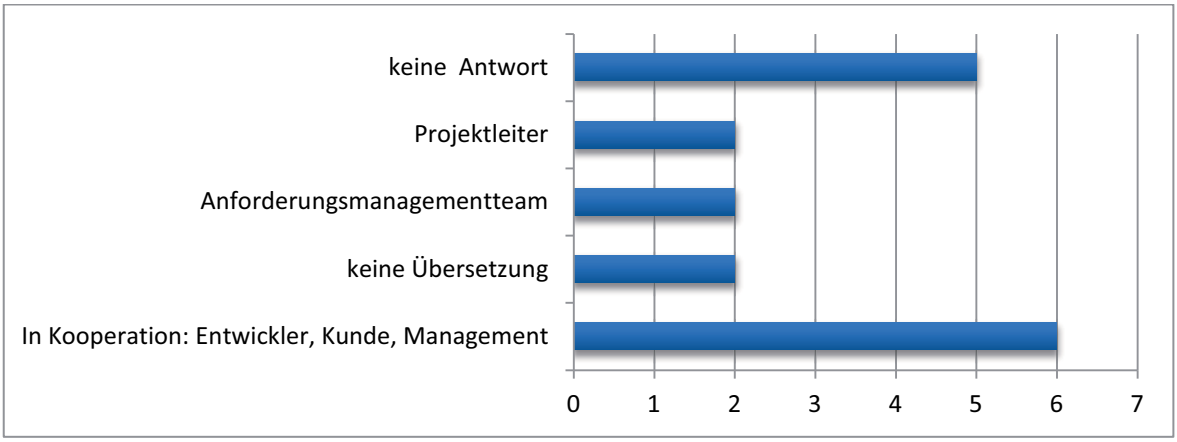


Abbildung 3-6 Zuständigkeit Übersetzung von Anforderungen
Quelle Eigene Darstellung

Um den Anforderungsentstehungsprozess besser bewerten zu können, wurden die Teilnehmer nach einer Einschätzung des Anteils der Anforderung, die schon am Beginn des Projekts bekannt sind, beziehungsweise wie viele erst im Verlauf des Projekts hinzukommen, gefragt. Die Antworten variieren hier von genauen prozentualen Einschätzungen über Angaben des Prozentsatzes der Änderungen bis hin zu allgemeinen Aussagen, dass dies sehr projektspezifisch sei. Aufgrund der sehr unterschiedlichen Antworten lässt sich hier kaum ein klarer Trend ausmachen. So haben je zwei Unternehmen die Spannen 40-60 %, 60-70 % und

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

80-90 % angegeben. Gemein haben alle Unternehmen allerdings, dass im Schnitt mehr als die Hälfte der Anforderungen erst im Verlauf des Projekts bekannt werden oder sich zumindest ein Anteil von 10-20 % der Anforderungen im Verlauf des Projekts maßgeblich ändert. Zwei der befragten Unternehmen machten keine Angaben zu dieser Frage, ein Befragter wies nochmal darauf hin, dass besonders spezielle, feingranulare Anforderungen erst später hinzu kommen.

Ausgehend von der Frage nach dem Anteil der später hinzukommenden oder sich im Projektverlauf veränderlichen Anforderungen, wurden die Befragten im weiteren Verlauf nach einer Einschätzung gebeten, wie mit solchen Anforderungen im weiteren Projektverlauf umgegangen wird und ob diese auch umgesetzt werden. Bis auf zwei Unternehmen die keine Angaben zu dieser Frage gemacht haben, gaben alle Befragten an, dass neue Anforderungen in jedem Fall aufgenommen werden. Nach der Aufnahme der Anforderungen werden diese typischerweise analysiert, wobei die verschiedenen Unternehmen unterschiedliche Kriterien und Ziele für diese Analyse haben. Zwei der Unternehmen priorisieren neue Anforderungen, zwei Unternehmen führen eine Kostenanalyse durch wobei zusätzlich eine Zeitanalyse durchgeführt wird. Einer der Befragten gab an, dass neue oder geänderte Anforderungen lediglich dokumentiert werden. Die weiteren Befragten konnten keine allgemeinen Aussagen zum Umgang mit neu hinzukommenden Anforderungen machen, da dies projektspezifisch sei und von der Leistungsbeschreibung des Projekts abhängt. Abbildung 3-7 zeigt den Umgang der befragten Unternehmen mit neuen oder geänderten Anforderungen.

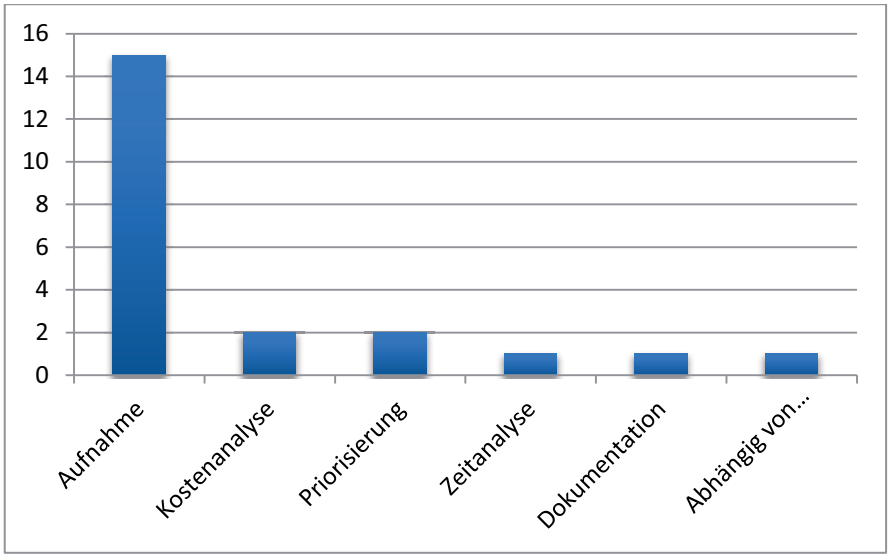


Abbildung 3-7 Umgang mit neuen oder geänderten Anforderungen
Quelle: Eigene Darstellung

Auf die Frage, ob die während der Entwicklung geänderten oder neu hinzugekommenen Anforderungen umgesetzt werden, gab es keine eindeutigen Antworten. Für jede dieser Anforderung wird eine einzelne Entscheidung gefällt. Die Entscheidungen werden dabei von einem Gremium, beziehungsweise einer speziell geformter Runde (3 Unternehmen) oder in Absprache mit dem Kunden (3 Unternehmen) getroffen. In jeweils einem Fall entscheidet das Marketing oder der Projektleiter über die Umsetzung der Anforderungen. Einige Unternehmen machten hier nur allgemeine Aussagen wie, die Anforderungen werden teilerweise umgesetzt oder die Entscheidung für oder gegen eine Anforderung wird aufgrund der Analyse getroffen, ohne genauer zu spezifizieren, welche Kriterien die Entscheidung beeinflussen oder von wem sie getroffen wird.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Im Folgenden wurde von den Befragten spezifiziert, welche Arten von Anforderungen typischerweise erst später hinzu kommen. Neun der befragten Unternehmen haben funktionale Anforderungen genannt. Des Weiteren nannten drei der befragten Unternehmen nichtfunktionale Anforderungen und zwei der Befragten gaben an, dass alle Arten von Anforderungen neu hinzukommen können. Drei der befragten Unternehmen machten hierzu keine Angaben. Besonders ist anzumerken, dass keiner der Befragten aus der Automotive Branche angegeben hat, dass funktionale Anforderungen später hinzu kommen. Im Gegensatz dazu kam diese Antwort bei den Befragten aus Softwareunternehmen sowie der Medizintechnik sehr häufig vor.

Anschließend sollten die Befragten eine Einschätzung des Aufwandes der später hinzugekommenen Anforderungen abgeben. Konkrete Angaben in Prozent machte hierzu nur ein Befragter; spät hinzukommende Anforderungen verursachen nicht mehr als 10% Zusatzaufwand, wobei dieser Aufwand nicht hinsichtlich Zeit, Kosten oder anderen Faktoren beziffert wurde. Ein weiteres Unternehmen gab an, dass Aufgrund der für die Umsetzung zusätzlichen Iterationen des Entwicklungsprozesses ein sehr hoher Aufwand entsteht. Am häufigsten wurden Antworten wie projektspezifisch, schwer zu sagen oder unterschiedlich genannt. Einige Unternehmen haben jedoch konkrete Faktoren die den Aufwand beeinflussen angegeben:

- Abhängigkeiten zu anderen Anforderungen
- Projektgröße (Je umfangreicher das Projekt ist, desto aufwändiger sind auch spät hinzukommende Anforderungen zu integrieren.)
- Projektstadium (Je später Anforderung hinzukommen, desto schwieriger ist es diese zu integrieren.)

Schließlich wurden die Befragten nach dem Vorgehen bei nachträglichen Änderungen gefragt. Bei einer Analyse der Antworten stellt sich heraus, dass bei acht der befragten Unternehmen das Vorgehen im Wesentlichen analog dem Vorgehen bei neuen Anforderungen entspricht. Änderungen werden zunächst bewertet und anschließend entschieden ob diese umgesetzt werden sollen. Unterschiedlich ist lediglich die Frage, wer die Entscheidung bezüglich der Umsetzung der Anforderungen trifft.

Des Weiteren, wurden die Unternehmen nach der Herkunft von Anforderungsänderungen befragt. Tabelle 3-9 listet die Antworten der Befragten auf. Besonders hervorzuheben hierbei ist, dass interne Stakeholder (Fachbereiche, Qualitätsabteilung, Vertrieb, Produktmanagement oder IT) hier eine deutlich größere Rolle spielen als bei der initialen Erhebung von Anforderungen. Fünf der Befragten machten zu dieser Frage keine Angaben.

Quellen von Anforderungsänderungen	**Anzahl der Angaben**
Intern o Fachbereiche o Qualitätsabteilung o Vertrieb o Produktmanagement o IT	10

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Kunden	9
Lieferanten	3
Gesetzgeber	1

Tabelle 3-9 ***Quellen von Anforderungsänderungen***
Quelle: Eigene Darstellung

Die nächste Frage befasst sich mit der Dokumentation von Anforderungen. Hierbei sollten die Befragten darauf eingehen, ob Lasten- bzw. Pflichtenhefte erstellt werden. Unter den Antworten wurden verschiedene Bezeichnungen für Lasten und Pflichtenhefte genannt, wie zum Beispiel Grob- und Feinkonzept. Unter den Unternehmen, die entweder nur ein Lasten- oder Pflichtenheft erstellen, gaben zwei nur das Pflichtenheft sowie eins nur das Lastenheft an. Ein Unternehmen erstellt ein integriertes Dokument, das sowohl Informationen des Pflichten- als auch des Lastenheftes enthält. Abbildung 3-8 stellt die Verteilung der Antworten dar.

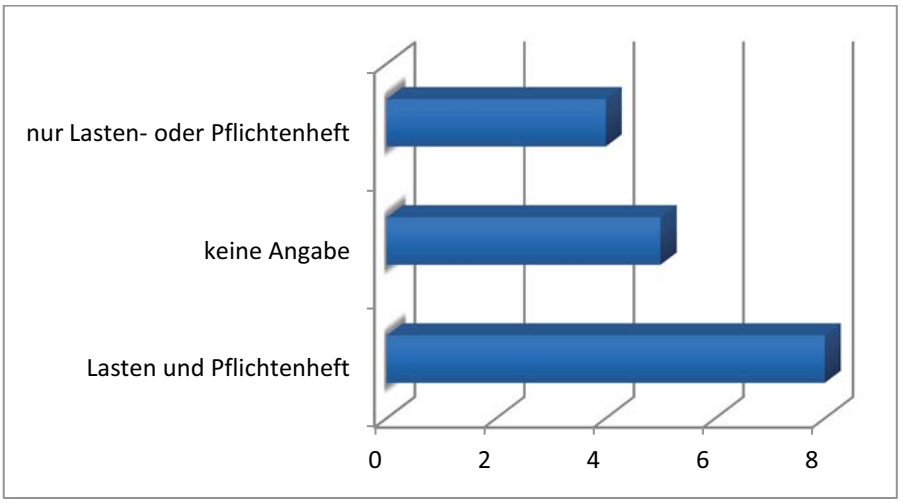


Abbildung 3-8 ***Dokumentation von Anforderungen***
Quelle: Eigene Darstellung

Bei den Unternehmen die Lasten- und/oder Pflichtenhefte erstellen, wurde konkreter nachgefragt von wem und wann dies erledigt wird. Eins der befragten Unternehmen hat angegeben, dass Lastenhefte vom Produktmanagement erstellt werden, in einem Fall ist die Marketingabteilung für die Erstellung der Lastenhefte verantwortlich. Auf die Frage nach Verantwortlichkeiten für die Fertigung der Pflichtenhefte gab es drei unterschiedliche Antworten: Lieferanten, IT-Abteilung und die IT zusammen mit dem Kunden. Die Mehrzahl der Unternehmen (13) machte keine Angaben zu dieser Frage.

Die nachfolgende Tabelle 3-10 listet nochmals die wesentlichen Angaben der Befragten zum Themengebiet Requirements Engineering auf. In der linken Spalte sind die einzelnen Detailfragen angegeben, die rechte Spalte enthält die wichtigsten genannten Stichworte.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Frage	Key Facts
Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?	○ Meistens kontinuierlich ○ Projektspezifisch ○ Anforderungen bleiben stabil
Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?	○ Excel Liste ○ Ticketing-System ○ Anforderungserhebung ○ Interpretation uns Strukturierung ○ Änderungsmanagement ○ Validierung und Verifikation ○ Anforderungsverfolgung ○ Dokumentation
Woher kommen die Anforderungen? (Kunde, intern, Berater...). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?	○ Kunden ○ IT-Abteilung ○ Produktendanwender
Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?	○ Projektteam bzw. Ausführungsteam ○ Produktmanager, Projektleiter
Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?	○ Kooperation ○ Entwickler ○ Fachabteilung ○ Kunden ○ Management ○ Anforderungen sofort in natürlicher Sprache fassen
Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?	○ Projektspezifisch ○ 80-90% bekannt am Start ○ 60-70% bekannt am Start ○ 40-60% bekannt am Start
Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?	○ Auf jeden Fall aufnehmen ○ Analyse/Kostenanalyse ○ Priorisierung ○ Zeitanalyse ○ Gremium - Entscheidung ○ Ansprache mit Kunden
Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)	○ Funktionale Anforderungen ○ Nicht-funktionale Anforderungen
Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?	○ Projektspezifisch ○ Schwer zu sagen ○ Anfand beeinflusst von Abhängigkeiten mit

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

	anderen Anforderungen, Projektgröße, Projektstadium
Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?	o Bewerten, Entscheiden, Umsetzen o Gleich wie bei Initialanforderungen
Von wem kommen die Änderungen?	o Intern o Kunden o Lieferanten
Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?	o Lastenheft: Produkt Management, IT-Abteilung o Pflichtenheft: Lieferanten, IT, Kunden
Welche Informationen sind im Pflichtheft enthalten? Welche Informationen sind im Lastenheft enthalten?	o Pflichtenheft ist in meisten Fällen Konkretisierung, detaillierte Ausarbeitung des Lastenheftes o Anforderungen o Ideen, Ausschreibung der Ideen o Lösungsbeschreibung

Tabelle 3-10 Key Facts Requirements Engineering
Quelle: Eigene Darstellung

3.3.6.2 Anforderungsermittlung und Anforderungsanalyse

Ausgehend von der allgemeinen Betrachtung des Requirements Engineerings im ersten Fragenteil werden anschließend die Themenbereiche der Ermittlung von Anforderungen sowie deren Analyse eingehender untersucht. Zunächst wurden die Unternehmen befragt, in welcher Art und Weise Anforderungen in ihren Projekten typischerweise erhoben werden, beziehungsweise welche Techniken sie dafür einsetzen. Die Mehrzahl der Unternehmen gab hier an Anforderungen hauptsächlich im Rahmen von Workshops und Interviews zu erheben. Unter anderem wurden aber auch Besprechungen, Analysen bestehender Systeme und Prozesse sowie der Einsatz von Prototypen genannt. Tabelle 3-11 stellt die eingesetzten Techniken sortiert nach Häufigkeit dar.

Technik zu Anforderungserhebung	Anzahl der Angaben
Workshop	14
Interview	7
Besprechungen	4
Analyse der bestehenden Prozesse/Systeme	3
Prototypen	3

Tabelle 3-11 Technik zu Anforderungserhebung
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Besonders ist zu bemerken, dass die Unternehmen in ihren Angaben nicht immer strukturierte und nichtstrukturierte Interviews unterschieden haben. Daher wurden beide Formen in Tabelle 3-11 zusammengefasst. Des Weiteren wurden Diskussion und Brainstorming zu Workshops hinzu gezählt, obwohl diese in einigen Fällen sicherlich auch als einfache Besprechungen anzusehen sind. Ein befragtes Unternehmen führt keine proaktive Anforderungserhebung durch. Im Schnitt setzen die Unternehmen ein bis zwei unterschiedliche Techniken zur Erhebung ein. Sechs der befragten Unternehmen setzen jeweils ein bis zwei Techniken ein, ein Unternehmen drei Techniken sowie drei Unternehmen bis zu vier unterschiedliche Techniken.

Auf der Grundlage der genannten Techniken zur Erhebung von Anforderungen wurde nach Problemen und Schwierigkeiten bei derselben gefragt. Die Antworten der 17 befragten Unternehmen lassen sich in drei Gruppen einteilen: Kommunikation, Prozessorganisation und Qualität der Anforderungen.

- *Kommunikation:* Die häufigste Antwort auf die Frage nach Problemen und Schwierigkeiten war die Kommunikation der Beteiligten. Außer allgemeinen Kommunikationsproblemen wurden insbesondere die Punkte Aufwand der Absprachen, mangelnde Informationsintegrität durch schlechte Kommunikation, Schwierigkeiten die richtigen Informationen zu bekommen, sowie Verständnisprobleme wegen unterschiedlicher Terminologien und Sichtweisen genannt. Dabei haben einige Befragte die Kommunikation mit Kunden als besonders problematisch hervorgehoben. Insbesondere wird es als schwierig angesehen qualitativ hochwertiges Feedback von Kunden zu bekommen. Der Auftraggeber/Kunde kann oftmals nicht genau sagen was er sich von einem Produkt erhofft und erwartet, ist sich über die Ziele des zu entwickelnden Systems nicht im klaren und hat oftmals unrealistische und somit auch nicht realisierbare Vorstellungen des Ergebnisses.
- *Prozessorganisation:* ein großes Problem stellt auch die Identifikation der richtigen Ansprechpartner und Stakeholder dar. Oftmals muss sehr großer Aufwand betrieben werden um diese zunächst zu identifizieren und später in geeigneter Weise in das Projekt zu integrieren.
- *Qualität der Anforderungen:* Ein weiteres großes Problem stellt die Qualität der Anforderungen dar. Diese werden häufig unpräzise und schlampig formuliert. Dies führt im weiteren Verlauf dazu, dass Anforderungen schwierig zu priorisieren sind und somit kein geeignetes Ranking der umzusetzenden Anforderungen gefunden werden kann.

Zusammenfassend lässt sich festhalten dass die meisten Probleme bei der Erhebung von Anforderungen im Kommunikationsbereich angesiedelt sind. Hier fallen besonders die unterschiedlichen Fertigkeiten und Hintergründe der projektbeteiligten Stakeholder ins Gewicht.

Bezüglich der Analyse der aufgenommenen Anforderungen wurden die Unternehmen hinsichtlich der an dieser Aufgabe beteiligten Rollen sowie im Detail durchgeführten Aktionen befragt. Bei der Analyse von Anforderungen sind sowohl die Kunden als auch die Entwickler maßgeblich beteiligt. Darüber hinaus wurden noch das Anforderungsteam, der Anforderungsverantwortliche sowie das Projektmanagement genannt. Anzumerken ist dabei jedoch, dass der Kunde ausschließlich in Zusammenarbeit mit einem der anderen Beteiligten an dieser Aufgabe partizipiert.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Hinsichtlich der einzelnen Aktionen und Aktivitäten zeichnen die Befragten ein recht einheitliches Bild. So werden Anforderungen zunächst konkretisiert und verfeinert. Ausgehend davon erfolgt eine Sortierung und Strukturierung die schließlich in einer Gruppierung der gewonnenen Anforderungen mündet. Sieben der befragten Unternehmen setzen in dieser Phase auch vordefinierte Templates oder spezielle Softwarewerkzeuge, wie zum Beispiel Modellierungstools, ein.

Abschließend wurde noch die Frage der Präzisierung von Anforderungen bei der Ermittlung und Analyse besprochen. Alle Unternehmen gaben hier an, dass Anforderungen priorisiert werden. Ein Unternehmen priorisiert jedoch lediglich konkurrierende Anforderungen. Hinsichtlich des Schemas nach dem Anforderungen priorisiert werden lassen sich verschiedene Formen identifizieren:

- muss/soll/kann
- 1 (must have) - 6 (nice to have)
- sofort zu erledigen/pending/on hold
- Ampelsystem
- ja/nein

Einige der Befragten haben zusätzlich Entscheidungsfaktoren für eine Priorisierung angegeben. So wurden in vier Fällen Kosten, in drei Fällen zeitliche Faktoren, einmal Komplexität, einmal Aufwand-/Nutzenanalyse, einmal der Kundennutzen und einmal die Realisierbarkeit genannt. Oft gibt es hier allerdings kein Standardvorgehen, sondern eine Priorisierung wird individuell für das jeweilige Projekt in Abstimmung mit den beteiligten Stakeholdern durchgeführt.

Tabelle 3-12 stellt nochmal die wesentlichen Aussagen der Befragten zur Anforderungserhebung und Anforderungsanalyse dar.

Frage	Key Facts
Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z.B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?	o Workshop o Interview o Besprechungen o Analyse der bestehenden Prozesse/Systeme o Prototypen
Was sind die häufigsten Probleme, die bei Ermittlung von Anforderungen auftreten?	o Kommunikation o Prozessorganisation o Qualität der Anforderungen
Wie werden die Anforderungen analysiert?(Unter Analyse versteht man die Konkretisierung von Anforderungen)	o Rollen: Kunden, Entwickler, Anforderungsteam, Projektmanagement o Aktionen: Konkretisierung, Verfeinerung, Strukturierung, Gruppierung o Vordefinierte Templates o Modellierungstools
Werden die Anforderungen priorisiert? Falls ja, wie?	o Nummerierung o muss/soll/kann

Tabelle 3-12 Key Facts Anforderungserhebung und Anforderungsanalyse
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

3.3.6.3 Anforderungsdokumentation

Im nächsten Themengebiet wurde die Dokumentation von Anforderungen diskutiert. Insbesondere ging es dabei darum, wie Anforderungen dokumentiert werden, ob dafür Vorlagen eingesetzt werden, welche Informationen diese Vorlagen umfassen sowie um die Frage, in wie fern Software Tools eingesetzt werden.

12 der befragten Unternehmen dokumentieren Anforderungen in fest definierten Templates. Es werden dabei vornehmlich gewöhnliche Textdokumente auf der Basis der Templates erstellt. Zwei der befragten Unternehmen gaben zusätzlich an, dass die Dokumentation der Anforderungen durch Modelle ergänzt wird. Ein Unternehmen hat keine Angaben zu dieser Frage gemacht.

Bezüglich der Art der Informationen die zu den einzelnen Anforderungen erfasst werden, wurde eine ganze Reihe von Optionen genannt. Die nachfolgende Tabelle 3-13 listet diese, gewichtet nach der Häufigkeit in der Nennung, auf.

Dokumentierte Information	Anzahl der Angaben
Beschreibung	10
Autor	7
Priorität	6
Verweis auf weitere Artefakte	6
Ansprechpartner	5
Identifikationsnummer	5
Name/Thema	4
Release	3
Termin	3
Testfälle	2
Updates an Anforderung	2
Geschätzter Aufwand	2

Tabelle 3-13 Dokumentierte Information zu Anforderungen
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Zusätzlich zu den genannten Informationen werden von vereinzelt auch Reifegrad, Nutzen, Vorbedingungen, Nachbedingungen, Fehler, Datum und Kommentare zu einzelnen Anforderungen dokumentiert.

Hinsichtlich einer Unterstützung der Anforderungsdokumentation durch Softwarewerkzeuge setzen alle, bis auf ein Unternehmen welches keine Angaben zu dieser Frage machte, diese ein. Eines der Unternehmen schränkte allerdings ein, dass Softwareunterstützung nur in Ausnahmefällen zu Anforderungsdokumentation eingesetzt wird. Hervorzuheben ist, dass 10 der befragten Unternehmen Microsoft Office als Softwarewerkzeug zur Dokumentation von Anforderungen einsetzen. Konkret wurden viermal Microsoft Excel sowie zweimal Microsoft Word genannt, die anderen Unternehmen machten hierzu keine genaueren Angaben. Darüber hinaus wurden noch folgende Softwarewerkzeuge genannt:

- Doors (4 Antworten)
- Requisit Pro (3 Antworten)
- HP Quality Center (2 Antworten)

Zusätzlich wurden noch die Produkte die Eclipse, Mercury Test Director sowie Caliber Requirements Management genannt. Eins der befragten Unternehmen hat eine eigene Softwareentwicklung im Einsatz, ein Weiteres gab an mehrere Werkzeuge gleichzeitig zu nutzen.

Die nachfolgende Tabelle 3-14 listet die wesentlichen Aussagen der befragten Unternehmen zur Dokumentation von Anforderungen auf.

Frage	Key Facts
Wie werden die Anforderungen dokumentiert? Gibt es ein Muster(Vorlage), um Anforderungen zu dokumentieren?	○ Fest definierte Templates ○ Textdokument
Welche Informationen werden dokumentiert?	○ Beschreibung ○ Autor ○ Priorität ○ Verweis auf anderen Artefakte ○ Ansprechpartner ○ ID ○ Name/Thema ○ Release ○ Termin ○ Testfälle ○ Updates an Anforderungen ○ Geschätzter Aufwand
Wie wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nicht, warum nicht?	○ Microsoft Office ○ Doors ○ Requisit Pro

Tabelle 3-14 Key Facts Requirements Engineering
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

3.3.6.4 Verifikation und Validierung

Der Fragenkomplex Verifikation und Validierung befasst sich im Wesentlichen mit der Fragestellung, welche Anforderungen umgesetzt werden und wie dies im Verlauf des Projekts verifiziert und sichergestellt wird.

Auf die Frage wer die Entscheidung trifft, dass eine einzelne Anforderung umgesetzt wird, hat die Mehrheit der Befragten (9) angegeben, dass diese vom Kunden getroffen wird. Auch wurde das Projektmanagement oder die Projektleitung als Zuständiger genannt. In drei der betrachtet Unternehmen erfolgt die Entscheidung anhand der vormals festgelegten Priorisierung der Anforderungen. Auch werden die Entwickler genannt, die hinsichtlich der Realisierbarkeit einzelne Anforderungen entscheiden wann und ob diese umgesetzt werden.

Ob die Eigenschaften des entwickelten Produktes mit den definierten Anforderungen übereinstimmen wird in den Unternehmen durch abschließende Tests sichergestellt. Unter den Antworten wurden allgemeine Tests, ohne eine genaue Spezifizierung wie dieser durchgeführt werden, Abnahmetests durch den Kunden, sowie interne Tests in Kombination mit Abnahmetests durch den Kunden genannt.

Tabelle 3-15 fasst nochmal die wichtigsten Antworten der Befragten zur Verifikation und Validierung von Anforderungen zusammen.

Frage	Key Facts
Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?	o Entscheidung von Kunden o Anhand der Priorisierung o Entwickler entscheiden über Realisierbarkeit
Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?	o Tests zu Überprüfung der Anforderungserfüllung o Abnahmetest o Interne Tests

Tabelle 3-15 Key Facts Verifikation und Validierung
Quelle: Eigene Darstellung

3.3.6.5 Prototyping

Im letzten Fragenabschnitt wurde das Thema Prototyping sowie der Einsatz von Prototypen im Rahmen des Anforderungsmanagements betrachtet. Insbesondere sollten die Befragten beschreiben, welchen Zweck oder welche Funktionen Prototypen im Rahmen des Anforderungsmanagements bei ihnen erfüllen, für welche Zielgruppen diese entwickelt werden, wer Prototypen entwickelt und welcher Aufwand und welche Kosten Prototyping bedeutet. Schließlich sollten die Befragten noch sagen, was für oder gegen einen Einsatz von Prototypen im Anforderungsmanagement spricht.

Prototypen dienen einer Vielzahl an Aufgaben. So wurde von neun der befragten Unternehmen die Kommunikation mit dem Kunden als wichtigste Aufgabe genannt. Insbesondere dienen Prototypen der Demonstration des Projektstands, dem Aussehen des Endprodukts, der Klärung der Kundenerwartungen sowie der Spezifizierung von Kundenanforderungen. Eine weitere Aufgabe ist das Testen von Funktionalitäten, das

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Sicherstellen der Machbarkeit sowie die Überprüfung von Anforderungen. Schließlich werden Prototypen auch zur Visualisierung sowie zur Gestaltung des User Interface eingesetzt.

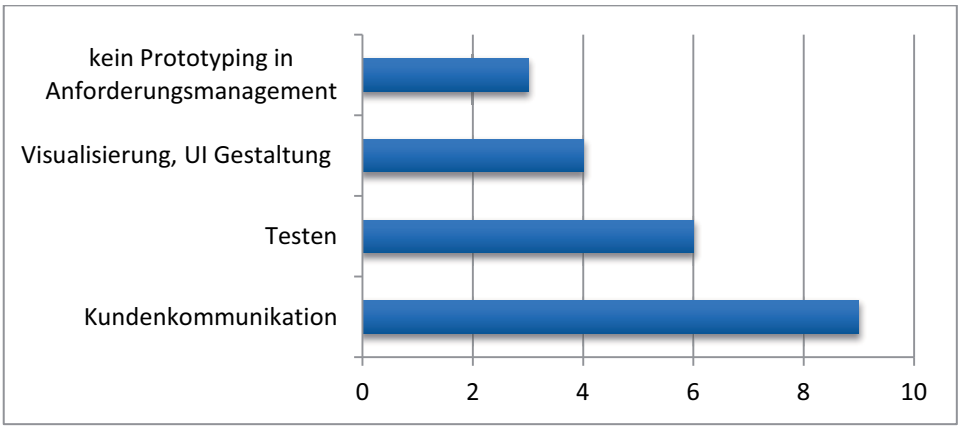


Abbildung 3-9 Einsatz von Prototypen im Anforderungsmanagement
Quelle: Eigene Darstellung

Abbildung 3-9 zeigt die Häufigkeit der Ziele von Prototypen. Dabei ist allerdings anzumerken, dass Prototypen meist nicht einem sondern häufig mehreren Zwecken dienen, wie zum Beispiel der Kundenkommunikation und dem Systemtest. Drei der befragten Unternehmen haben angegeben, dass sie keine Prototypen im Rahmen der Anforderungsermittlung erstellen. Eins der befragten Unternehmen setzt Prototypen explizit zur Konfliktlösung ein.

Auf die Frage für welche Zielgruppen Prototypen erstellt werden, hat die Mehrheit der Unternehmen den Kunden genannt. Als Kunde wurden sowohl der Auftraggeber als auch andere Fachabteilungen genannt. Zwei Unternehmen haben den Endanwender als Zielgruppe angegeben, trennen diese allerdings nicht klar von der Zielgruppe Kunde ab. Darüber hinaus wurden noch das Management sowie das Integrationsteam genannt.

Als nächstes wurden die Techniken beziehungsweise Werkzeuge (Sprachen, Frameworks,...) die für das Prototyping eingesetzt werden besprochen. Sieben der befragten Unternehmen gaben dabei an, für die Entwicklung von Prototypen die gleiche Technologie einzusetzen die auch im Endprodukt eingesetzt werden soll. Oft werden Prototypen auch zu fertigen Endprodukten ausgebaut. Viele Unternehmen nutzen projektspezifische Werkzeuge, diese variieren von Office Tools wie zum Beispiel Power Point über HTML und Adobe Photoshop hin zu andere GUI Werkzeugen.

Bei der Einschätzung des Aufwands und Erfolgs von Prototypen im Anforderungsmanagement, nannten einige Unternehmen prozentuelle Angaben hinsichtlich der Gesamtkosten oder des allgemeinen Projektaufwands. Die Zahlen variieren hier von 5-10 % des gesamten Projektaufwands hin zu 20 % des Gesamtkostenanteils. Viele Unternehmen konnten den Aufwand allerdings nicht genauer spezifizieren und gaben nur allgemeine Antworten, wie zum Beispiel „zu aufwändig“ oder „nur geringer Aufwand“. Betrachtet man die Verteilung der Antworten über die Branchen hinweg genauer, zeichnet sich ab, dass Unternehmen der Automobilbranche den Einsatz von Prototypen für zu aufwändig halten und diese meist nur Einsetzen, wenn Prototypen von einzelnen Anforderungen gefordert werden. Auf der anderen Seite gibt es aber auch Unternehmen die nahezu keinen Aufwand im Einsatz von Prototypen sehen, da die Vorteile der Risikominimierung etwaige Nachteile deutlich ausgleichen. Auch wurde festgestellt, dass das Erstellen von Prototypen im Laufe der letzten Jahre deutlich vereinfacht wurde.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Abschließend wurden die Interviewpartner noch gebeten eine Einschätzung zu geben, welche Faktoren für oder gegen den Einsatz von Prototypen im Anforderungsmanagement sprechen.

Gegen einen Einsatz von Prototypen wurden folgende Punkte genannt:

- Es besteht die Gefahr nicht perfekter Prototypen, die als Ausgangsbasis für das Endprodukt genutzt werden.
- Prototypen sind zu aufwändig und zu konzeptionell.
- Nicht alle Projektbeteiligten wissen mit Prototypen umzugehen und denken nach dem der erste Prototypen erstellt wurde, dass das Projekt in drei Tagen fertig sei.
- Prototypen, die ohne einen bestimmten Zweck erstellt werden, sorgen für Verwirrung.

Für den Einsatz von Prototypen wurden nachfolgende Punkte genannt:

- Prototypen erlauben eine bessere Einschätzung der Anforderungen und können als Entscheidungsgrundlage dienen.
- Prototypen sind sinnvoll um die Machbarkeit von Anforderungen zu überprüfen
- Prototypen bringen bei geringem Aufwand ein großes Maß an Sicherheit.
- Prototypen eignen sich sehr gut um neue Techniken auszuprobieren.
- Prototypen sind einsetzbar im Erwartungsmanagement.

Generell wird die Entscheidung für oder gegen einen Prototypen auf der Grundlage einer Kosten/Nutzen Abschätzung durchgeführt. Drei der befragten Unternehmen machten zu den Vor- und Nachteilen des Prototyping keine Angaben.

Tabelle 3-16 stellt die wichtigsten Aussagen der Unternehmen zum Einsatz von Prototyen im Anforderungsmanagement dar.

Frage	Key Facts
Welche Zweck oder welche Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen?	○ Kommunikation ○ Kundekommunikation ○ Demonstration von Projektstand ○ Aussehen von dem Endprodukt ○ Klärung der Kundenerwartungen ○ Spezifizierung der Kundenanforderungen
Für welche Zielgruppen (Stakeholder) werden Prototypen entwickelt?	○ Der Kunde ○ Endanwender ○ Management
Wie werden diese Prototypen entwickelt?	○ Gleiche Technologien, die auch im Endprodukt eingesetzt werden ○ Projektspezifisch
Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?	○ Um Kosten zu sparen arbeitet man inkrementell die Prototypen zu Lösungen aus ○ Fast kein Aufwand, dadurch wird das Risiko für die Entwicklung reduziert ○ Leider meist zu hoher Aufwand

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?	o Gegen Einsatz: Prototypen sind zu aufwendig, Nicht alle kennen wie es mit einem Prototyp umzugehen ist, o Für Einsatz: Prototypen erlauben bessere Einschätzung der Anforderungen, sehr gut um neue Techniken auszuprobieren, Einsetzbar in Erwartungsmanagement

Tabelle 3-16 ***Key Facts Prototyping***
Quelle: Eigene Darstellung

3.3.7 Diskussion

Der strukturierte Umgang mit Anforderungen spielt für alle befragten Unternehmen eine entscheidende Rolle im Verlauf ihrer Projektarbeit. Dabei werden Anforderungen als über den kompletten Projektverlauf veränderlichen wahrgenommen. Die initialen Anforderungen zu Beginn eines Projekts werden maßgeblich vom Kunden oder den Endanwendern gestellt. Darüber hinaus kommen in dieser Phase auch Anforderungen aus anderen Fachabteilungen oder gesetzliche Vorgaben zum Tragen. Es lässt sich aber klar feststellen, dass der Kunde hier die mit Abstand größte Bedeutung hat. Betrachtet man die Techniken die Unternehmen für die Erhebung von Anforderungen einsetzen, zeigt sich ein klarer Trend zu Workshops und Interviews. In der Regel setzen die Unternehmen jedoch ein bis zwei unterschiedliche Techniken gleichzeitig ein, dies kann aber projektabhängig variieren.

Ein Problem dabei ist jedoch, dass Anforderungen unterschiedlichen Fachdomänen entstammen und daher in eine einheitliche Sprache und Darstellungsform gebracht werden müssen. Diese Übersetzung und Interpretation der Anforderungen erfolgt in der Regel in Zusammenarbeit mit dem Kunden. In Einzelfällen wird dieser Arbeit aber auch von der Projektleitung oder den Entwicklern durchgeführt. Dies hat den Nachteil, dass Anforderungen einer fremden Fachdomäne interpretiert werden müssen und so oftmals hohe Reibungsverluste entstehen.

Sind Anforderungen erhoben und dokumentiert, erfolgt eine Priorisierung der Anforderungen. In der Regel werden Anforderungen in einem Skalensystem priorisiert, das von einfachen Ja/Nein-Einteilungen hin zu einem Schulnotensystem reicht. Als Grundlage für die Priorisierung werden Kosten, zeitliche Faktoren sowie Komplexität der Umsetzung herangezogen. Die Priorisierung wird häufig, neben der expliziten Entscheidung des Kunden, als Grundlage für eine Umsetzungsentscheidung genutzt. Nach Abschluss des Projekts werden diese Kriterien auch im Rahmen von Abnahmetests eingesetzt.

Nach Angaben der Unternehmen ändern sich bis zu 60 % der initialen Anforderungen im Projektverlauf. Dies kann bedeuten, dass völlig neue Anforderungen nachträglich hinzukommen aber auch, dass bereits erhobenen Anforderungen geändert oder bearbeitet werden. Alle Unternehmen waren sich einig, dass mindestens 10-20 % der Anforderungen diesem Änderungsprozess unterworfen sind. Inwieweit neue oder geänderte Anforderungen umgesetzt werden ist dabei sehr projektspezifisch. Betrachtet man die Art dieser Anforderungen, so stellt man fest, dass es sich maßgeblich um funktionale Anforderungen handelt. Auffällig ist, dass die Vertreter der Automobilbranche als einzige keine großen Änderungen der funktionalen Anforderungen angaben. Offensichtlich ist dies ein Phänomen

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

softwarenaher Projekte, wie man sie natürlich in der Softwarebranche aber auch in der Medizintechnik findet.

Angaben über den Mehraufwand variieren hier stark zwischen den Befragten. Von Aussagen über einen sehr hohen Aufwand durch zusätzliche Iterationen des Entwicklungsprozesses bis hin zu Angaben, hinzukommende Anforderungen verursachen nicht mehr als 10 % Zusatzaufwand, waren hier alle Meinungen vertreten. Abhängig scheint dieser Aufwand von der Vernetzung der Anforderungen zu anderen Anforderungen, der Projektgröße sowie dem Stadium in dem sich das Projekt befindet zu sein. Im Gegensatz zu den initialen Anforderungen, die maßgeblich vom Kunden und Endanwender stammen, werden Änderungen meist von internen Fachbereichen, Qualitätsabteilungen oder der IT initiiert. Diese Tatsache ist besonders entscheidend für eine Interpretation der Anforderungen, da die geänderte Personengruppe auch einen anderen fachlichen Hintergrund mitbringt.

Dokumentiert werden Anforderungen in der Regel in Lasten- bzw. Pflichtenheften. Hierzu werden häufig textuelle Vorlagen, Templates, oder Listen eingesetzt. In der Regel wird für die Dokumentation Microsoft Office als Werkzeuge genutzt. Hier kommen sowohl Word als auch Excel zum Einsatz. Darüber hinaus werden spezialisierte Softwarelösungen, wie Doors oder RequisitPro eingesetzt.

Eines der zentralsten Probleme beim Erheben von Anforderungen ist die Kommunikation mit den beteiligten Stakeholdern. Aufgrund unterschiedlicher Terminologien und Sichtweisen ist es oft schwierig oder unmöglich die richtigen Antworten und Anforderungen zu erhalten. Auch sind häufig, wie bereits in der Literatur beschrieben (Antón 2003), ungenaue Vorstellungen sowie unklare Zielvorstellungen der Stakeholder ein großes Problem. Meist sind diese unrealistisch und damit auch nicht realisierbar. Besonders hervorzuheben ist hier die Kommunikation mit dem Kunden, da die unterschiedlichen Fachdomänen hier oft eine sprachliche und inhaltliche Hürde darstellen. Ein weiteres Problem ist die Identifikation der richtigen Ansprechpartner und Stakeholder. Auch wenn man diese ausfindig gemacht hat, ist eine geeignete Integration oft nur schwer möglich. Aus den beiden genannten Problemen der Kommunikation und Organisation entsteht schließlich auch das dritte genannte Problem, die Qualität der Anforderungen. Diese werden häufig unpräzise und schlampig formuliert was dann eine spätere Nutzung erschwert oder teilweise auch unmöglich macht.

Ein Ansatz, den viele Unternehmen wählen um den genannten Problemen zu begegnen, ist der Einsatz von Prototypen. Dabei hinterlassen die geführten Gespräche allerdings ein sehr differenziertes Bild über die Vor- und Nachteile von Prototyping. So werden Prototypen häufig als zu aufwändig und zu konzeptionell wahrgenommen und es zeigt sich, dass Projektbeteiligte nicht mit Prototypen umzugehen wissen. So entsteht oft der Eindruck, dass, wenn zentrale Funktionen im Rahmen eines Prototyps bereits demonstrierbar und kommunizierbar sind, damit auch die Anforderungen des Projekts erfüllt wurden. Dies führt dazu, dass Prototypen häufig zum Endprodukt ausgebaut und weiterentwickelt werden. Auf der anderen Seite zeigt der Einsatz von Prototypen sehr viele positive Aspekte. So können Anforderungen besser eingeschätzt, die Machbarkeit von Anforderungen überprüft sowie neue Techniken risikofrei ausprobiert werden. Hinsichtlich der Kommunikation mit den Stakeholdern, dem größten genannten Problem im Anforderungsmanagement, werden Prototypen als Mittel des Erwartungsmanagement eingesetzt.

Zusammenfassend lässt sich feststellen, dass Anforderungsmanagement eine zentrale Bedeutung bei der Gestaltung von Softwarelösungen einnimmt. Dabei existiert eine Vielzahl

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

an Vorgehensmodellen und Methoden die auch in der Praxis Anwendung finden. Die großen Probleme liegen hier in der Kommunikation mit den beteiligten Stakeholder sowie der großen Veränderlichkeit von Anforderungen im Projektverlauf. Kommunikation bedeutet hier einerseits die Integration unterschiedlicher Fachdomäne, sowie eine gemeinsame Präsentation von Anforderungen. Ausgehend von den Aussagen der Unternehmen lässt sich feststellen, dass die größte Herausforderung in der Unterstützung dieser Kommunikation liegt.

3.4 Gestaltung der Mensch-Maschine Interaktion

Im Gegensatz zu vielen Softwaresystemen, gerade im Bereich der betrieblichen Informationssysteme, steht bei der Gestaltung von Automotive Services die Interaktion mit den Nutzern, also dem Fahrer, den Beifahrern und weiteren Passagieren im Zentrum der Aufmerksamkeit. In der Regel geht es bei Automotive Services nicht um hoch komplexe Dienste, bei denen Funktionalität oder ein komplexes Datenbankdesign im Vordergrund stehen, sondern vielmehr sind Automotive Services, analog zu mobilen Diensten die man auf Smartphones findet, eher von kleinem Funktionsumfang. Die Herausforderung liegt vielmehr auf einer effizienten und effektiven Interaktion der Nutzer mit dem Service. Erst diese ermöglicht einen sinnvollen Einsatz im Fahrzeug. Das Forschungsfeld das sich mit eben dieser Fragestellung befasst ist die Mensch-Maschine Interaktion.

3.4.1 Begriffliche Klärung

Mensch-Maschine Interaktion (MMI)[7] befasst sich mit der Schnittstelle zwischen interaktiven Systemen und Menschen. Der Fokus liegt hierbei auf einer benutzergerechten Gestaltung dieser Systeme. Caroll bezeichnet MMI als „[…] die Lehre und Ausübung von Usability“ (Carroll 2001). Usability, zu Deutsch Gebrauchstauglichkeit, ist das "Ausmaß in dem ein Produkt durch bestimmte Benutzer in einen bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen" (Geis 2005; International Organization for Standardization 1998)[8]. Die Mensch-Maschine Interaktion als Bestandteil der Informatik vereint somit Erkenntnisse aus den Bereichen Interaktionsdesign, Informationsdesign, Softwareergonomie und Psychologie.

Auch wenn eine Notwendigkeit der Betrachtung der Gebrauchstauglichkeit schon zu Beginn der Entwicklung von Rechenmaschinen existierte, so ist die wissenschaftliche Disziplin der Mensch-Maschine Interaktion 1982 auf der Konferenz “Human Factors in Computer Systems“ in Gaithersburg (USA) entstanden. Gleichzeitig bildete sich die Special Interest Group in Computer-Human Interaction (ACM SIGCHI) der Association for Computing Machinery's (ACM) (Special Interest Group in Computer-Human Interaction (ACM SIGCHI) 2009).

3.4.2 Grundlagen der Mensch-Maschine Interaktion

Als Grundlage für die Entstehung der wissenschaftlichen Disziplinen der Mensch-Maschine Interaktion werden vier Entwicklungen gezählt (Carroll 2001): Die interaktive Softwareentwicklung und der damit einhergehende Bau und Einsatz von Prototypen, die Softwarepsychologie, die Kognitionswissenschaften sowie die Entwicklung des User Interface Design.

[7] engl. Human-Computer Interaction (HCI)

[8] engl. “Extend to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use”(International Organization for Standardization 1998)

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

3.4.2.1 Iterative Softwareentwicklung und Prototyping

Durch die fortschreitende Entwicklung der verfügbaren Hardware konnten im Laufe der Zeit immer umfangreichere, leistungsfähigere und komplexere Programme entwickelt und auf diese Hardware betrieben werden. Dijkstra beschrieb die Problematik dieses Trends in seiner Dankesrede zum Turing Award: „the major cause is... that the machines have become several orders of magnitude more powerful! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming had become an equally gigantic problem." (Dijkstra 1972). Folglich stiegen ebenfalls die Kosten für die Entwicklung von Software stark an, was letztendlich zur Entwicklung der Disziplin des Software Engineering führte. Balzert klassifiziert diese als die „Zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Softwaresystemen" (Balzert 1996, 36).

Mit dem Einzug des Software Engineering haben sich auch Vorgehensmodelle für die Entwicklung von Software etabliert. Zunächst wurde ein ingenieurmäßiges, sequenzielles Vorgehen verfolgt, beispielsweise in Form des Wasserfallmodells. Da Anforderungen an Softwaresysteme jedoch im Verlauf deren Entwicklung einem steten Wandel unterliegenden (vgl. Abschnitt 3.2), konnten sich später iterative Softwareentwicklungsmodelle durchgesetzt. Tabelle 3-17 gibt einen Überblick über verfügbaren Vorgehensmodelle sortiert nach ihrem Grad an Formalisierung und ihrem sequentiellen oder iterativen Charakter (Krcmar 2010a, 193).

	Sequentiell	Iterativ
Stark formalisiert	○ Wasserfallmodell ○ V-Modell ○ V-Modell (200x) ○ V-Modell XT ○ W-Modell ○ Inkrementell strukturierter Ansatz ○ (Neo-) Hermes ○ …	○ Spiralmodell ○ RUP ○ Prototyping ○ OO Lifecycle-Modell ○ Feature Driven Development ○ …
Wenig formalisiert		○ Extreme Programming ○ Object Engineering Process ○ Partizipative Softwareentwicklung ○ SCRUM ○ MDA ○ …

Tabelle 3-17 Überblick über Vorgehensmodelle
Quelle: Eigene Darstellung in Anlehnung an (Krcmar 2010, 193)

Im Zuge des iterativen Vorgehens bei der Entwicklung von Software gewann auch der Einsatz von Prototypen an Bedeutung. Ein wesentlicher Meilenstein auf diesem Weg ist das Spiralmodell nach Boehm, welches eine zyklische Weiterentwicklung von Prototypen in

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Abstimmung mit den Stakeholdern propagiert (Boehm 1988). Grundsätzlich kann man zwischen drei Arten des Prototyping unterscheiden (Floyd et al. 1989):

- *Exploratives Prototyping*, das auf Kommunikation mit den Stakeholder fokussiert.
- *Experimentales Prototyping*, das zur Präsentation einzelner Aspekte genutzt wird.
- *Evolutionäres Prototyping*, welches wohl am ehesten der Vorstellung von Boehm entspricht und eine iterative Weiterentwicklung der Prototypen hin zum Produkt verfolgt.

Prototypenbau und iterative Softwareentwicklung beziehen also den Menschen in den Design- und Entwicklungsprozess mit ein, um auf diese Weise die Qualität von Softwareprodukten zu steigern.

3.4.2.2 Software Psychologie

Dieses stärkere Einbeziehen von Menschen in die Entwicklungsprozesse hat enorme Auswirkungen auf das Erscheinungsbild von Computern, Software und Programmierung. Entwickler von Software müssen sich verstärkt mit unterschiedlichen Stakeholdern und deren Anforderungen zurechtfinden, gleichzeitig aber immer komplexer werdende technische Grundlagen beherrschen. „Although attention to improving programming environments continues, emphasis within CHI[9] has shifted from helping programmers-as-users to helping programmers develop better interfaces for non-programming end-users“ (Grudin 1990, S.263). Diese Entwicklung der zunehmende Bedeutung von Stakeholder in der Gestaltung von Software führt zu einer steigenden Bedeutung des „Human Factor“[10].

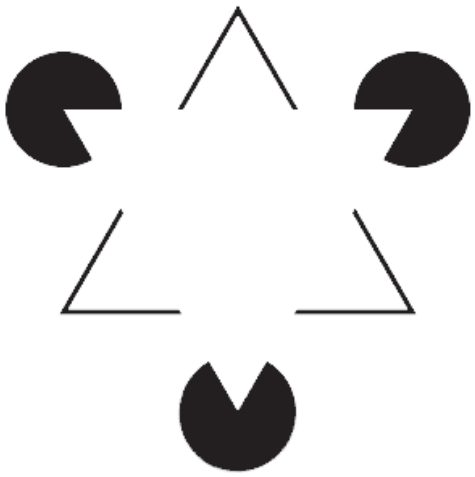


Abbildung 3-10 Kognitive Wahrnehmung des Kanizsa Dreiecks
Quelle: (Engel/Gold 1998, 178)

Der Begriff Human Factor geht auf Gerald Weinberg zurück (Molzberger 1983). Er erkannte die Bedeutung der Psychologie für die Entwicklung von Software, die sich in den vergangenen Jahren als zunehmend bedeutend herausgestellt hat. „Wir machen uns nicht klar, wie stark unsere Ausgangsannahme die Art beeinflusst, wie wir Daten sammeln und auswerten“ (Calvin 2004, 131). Das menschliche Gehirn umfasst zahlreiche kognitive Vorgänge die sich mit der Umwelt des Menschen beschäftigen und dabei nicht bewusst wahrgenommen werden (Roth 1997,30-33). Der Mensch interpretiert unterbewusst eigene Erfahrungen in Interaktion mit seiner Umwelt, wobei geringe Unklarheiten mit eigenem

[9] Computer-Human Interaction
[10] Faktor Mensch

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Wissen aufgefüllt werden. Dies führt aber nicht notwendigerweise zu richtig Entscheidungen und Einschätzungen. Der Effekt kann mit der optischen Wahrnehmungstäuschung des Kanizsa-Dreiecks veranschaulicht werden.

Abbildung 3-10 zeigt das Kanizsa Dreieck, in dem der Betrachter ein weißes Dreieck im Zentrum der Abbildung wahrnimmt, obwohl dieses eigentlich nicht vorhanden ist. Diese fälschlicherweise wahrgenommenen kognitiven Konturen veranschaulichen den beschriebenen Effekt: Die bekannte Form des Dreiecks wird unterbewusst an eine passende Stelle interpretiert. Das gleiche Phänomen tritt auch bei mangelhaften und ungenauen Softwareanforderungen auf. Unklarheiten und Lücken in der Spezifikation sowie Unstimmigkeiten zwischen Anforderungen, werden im Rahmen der kognitiven Wahrnehmung unterbewusst ergänzt. So entstehen unterschiedliche Wahrnehmungen der Stakeholder bei gleichem Informationsstand. Um diesem Effekt entgegen zu wirken ist es wichtig, Missverständnisse frühzeitig aufzudecken oder ganz zu verhindert um Unstimmigkeiten möglichst beseitigen zu können.

3.4.2.3 Modelle der Kognitionswissenschaft

Neben den psychologisch relevanten Faktoren im Rahmen der Kommunikation in Softwareprojekten gibt es darüber hinaus psychologische Faktoren die den Umgang und die Wahrnehmung von Software durch Stakeholder beeinflussen. Die Kognitionswissenschaft beschäftigt sich mit der „Entwicklung all der Funktionen […], die zum Wahrnehmen eines Gegenstandes oder zum Wissen über ihn beitragen“ (Miller 1994). Ausgehend von den Erkenntnissen der Kognitionswissenschaft, die ihre Wurzeln in der Psychologie, Linguistik, Neurowissenschaft, Philosophie und Informatik hat, gibt es in der MMI-Forschung Modelle und Richtlinien für die Gestaltung von Software. Ziel ist es Vorhersagen über mögliche Probleme zu treffen und Lösungsvorschläge zum Umgang mit diesen bereitzustellen. „In the ideal, such models produce a priori quantitative predictions of performance at an earlier stage in the development process than prototyping and user testing. That is, they predict execution time, learning time, errors, and they identify those parts of an interface that lead to these predictions, thereby providing a focus for redesign effort“ (John/Kieras 1996).

Ein bekannter Vertreter diese Ansätze ist das GOMS-Model (Card/Moran/Newell 1986). Das Modell versucht Aussagen über die Effizienz und die Gebrauchstauglichkeit von Software in einem sehr frühen Stadium der Softwarentwicklung zu treffen. Es werden Annahmen getroffen wie man eine Software gestalten muss damit der Benutzer diese intuitiv verwenden kann um auf diese Weise unbekannte oder selten genutzt Aufgaben zu beschleunigen: „it is a representation of the "how to do it" knowledge that is required by a system in order to get the intended tasks accomplished” (Kieras 1996, 2). Im Focus von GOMS stehen die Bedürfnisse und Interaktionen des Nutzers mit dem Rechner. Diese werden in elementare Bestandteile zerlegt und dann betrachtet. Diese Bestandteile können physisch, wahrnehmend oder kognitiv sein und umfassen die Konstrukte: **G**oals (Ziele), **O**perators (Aktionsmöglichkeiten des Benutzers), **M**ethods (Aneinanderreihung von Operators die zu einem Ziel führt) und **S**elections (Situationsabhängige Auswahlregeln für Methods). In den letzten Jahren wurden weitere GOMS-Varianten entwickelt, um zusätzliche Faktoren zu untersuchen.

Ein zweites bekanntes Modell ist das Cognitive Engineering. „[…] The goal of Cognitive Engineering is to come to understand the issues to show how to make better choices when they exist, and to show what the tradeoffs are when, as is the usual case, an improvement in one domain leads to deficits in another” (Norman/Draper 1986, 31). Im Gegensatz zu GOMS

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

hat Cognitive Engineering den Fokus nicht auf den konstruktiven sondern auf den gestalterischen Aspekten von Software. Es soll also nicht die Benutzeraktion in ihre elementaren Bestandteile zerlegt, sondern vielmehr die Darstellung und Interaktion mit dem System von dessen Funktionalität getrennt betrachtet werden. Nach Normen ist die Gestaltung der entsprechenden Benutzeroberflächen nicht Teil der Fähigkeiten eines Programmierers und sollte daher einem speziellen Designer überlassen werden (Norman 1996, 208). Ein Beispiel für eine solche Trennung ist die Gestaltung eines einfachen Mausklicks. Für den Programmierer steht hier das Aktivieren einer Funktion im Vordergrund. Aufgabe des Designers ist es hingegen, die Darstellung und Interaktion mit dem Nutzer so zu gestalten, dass diese dessen Erwartungen entspricht.

Sowohl GOMS als auch das Cognitive Engineering stellen den Benutzer in den Mittelpunkt ihrer Überlegungen. Aus ihren beiden unterschiedlichen Betrachtungswinkeln heraus verfolgen sie das Ziel gut funktionierende Software zu präsentieren die die Erwartungen der Nutzer und Stakeholder umsetzt.

3.4.2.4 User Interface Design

Die vierte Entwicklung, die als Grundlage für die Disziplin der Mensch-Maschine Interaktion angesehen wird, sind grafische Benutzeroberflächen. Ursprünglich waren Computer reine Rechenmaschinen deren primäre Aufgabe in der Berechnung von Aufgabenstellungen und nicht in der Darstellung von Ergebnissen lag (Carroll 2001). In diesem Szenario werden Computer ausschließlich von speziell geschultem Personal genutzt, so dass eine intuitive und flexible Interaktion mit diesen, wie sie im Bereich der Automotive Services zwingend notwendig ist, nicht entscheidend für eine erfolgreiche Nutzung ist.

Abbildung 3-11 Xerox Star
Quelle: (Wikipedia 2010a)

Die Entwicklung grafischer Benutzeroberflächen beginnt bereits 1945 mit dem Memex-Apparat[11] von Werner von Bush (Busch 1945, 4). Mit der Entwicklung der Software „Skatchpad“ von Ivan Sutherland (Sutherland 1964), wurde die erste auf Software basierte grafische Interaktion ermöglicht, die zu einer der wichtigsten und einflussreichsten in der MMI Geschichte gehört (Sears/Jacko 2007, 5). Um Benutzeroberflächen auch einfach bedienen zu können, wurde 1963 die Computermaus am Stanford Research Institute unter Mitwirkung von Douglas C. Engelbart und William Englisch entwickelt (Reimer 2005, 2).

[11] Schreibtisch, der Mikrofilme als Datenspeicher und eingebaute Bildschirme zur Informationsdarstellung nutzt (Bush 1945, 3).

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Eingesetzt wurde die Maus das erste Mal 1973 am Xerox Alto. Diese Maschine des Palo Alto Research Center ist der erste Computer mit einem Graphischen User Interface (GUI) (Abbildung 3-11).

Der Alto wurde aber nur zu experimentellen Zwecken genutzt, während die ersten kommerziellen Personal Computer, Xerox Star 1981 und Lisa von Apple 1983, folgten. Beide PCs hatten jedoch keinen wirtschaftlichen Erfolg, welcher aber 1984 durch die Weiterentwicklung von Lisa zum Macintosh erreicht wurde (Abate 2004). Das Betriebssystem Windows 1.0 (Abbildung 3-12) kam 1985 auf den Markt und war die erste grafische Benutzeroberfläche die nicht auf ein bestimmtes Computermodell zugeschnitten war, sondern auf unterschiedlichen PCs eingesetzt werden konnte. Benutzeroberflächen und dafür notwendige Gerätschaften (Computermaus, Tastatur, Bildschirm, usw.) ermöglichen Computeranfängern ein einfaches Arbeiten am Computer und sind somit die Hauptverantwortlichen für die Alltagstauglichkeit des Computers.

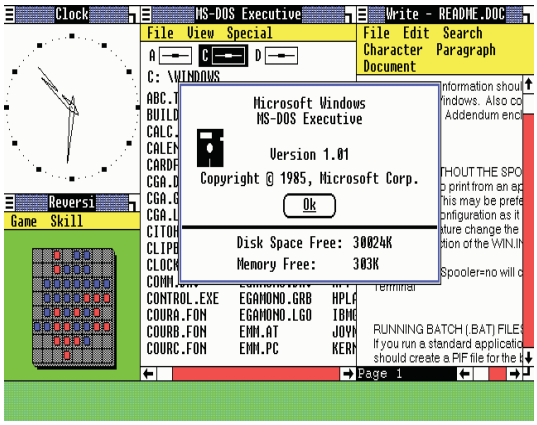


Abbildung 3-12 Windows 1.0
Quelle: (Wikipedia 2010b)

Mit der Schaffung graphischer Benutzeroberflächen rück die Interaktion des Nutzers mit dem System in den zentralen Fokus der Entwicklung. Überträgt man diese Beobachtung auf den Bereich der Automotive Services, so ist diese Bedeutung in einem verstärkten Maße zu beobachten. Durch die umgebungsbedingten Einschränkungen im Fahrzeug ist eine Nutzung über simple Eingabemechanismen wie z.B. eine Konsole nicht mehr möglich und die Gestaltung der graphischen Oberfläche von zentraler Bedeutung für den Erfolg eines Services.

3.4.3 Human Factor

Neben dem ingenieurmäßigen Erfassen von Softwareanforderungen im Rahmen des klassischen Requirements Engineering, verfolgt das Forschungsfeld der Mensch-Maschine Interaktion eine stärkere Integration der Stakeholder in den Gestaltungsprozess. Floyd et al. entwickeln ein alternatives, benutzerorientiertes Vorgehen in dem Lösungen durch alle Beteiligten gemeinsam entwickelt werden (Floyd et al. 1989). Der Ansatz entwickelte sich im skandinavischen Raum (Dänemark, Schweden und Norwegen), da diese Länder nach Ansicht der Autoren eine nicht zu stark ausgeprägte feudale Gesellschaftsform wie andere europäische Länder haben und die Menschen es daher eher gewöhnt sind Entscheidungen gemeinsam zu fällen. Ausgehend von den Gedanken der Humanisierung und Demokratisierung stellen Floyd et al. mit dem skandinavischen Ansatz die enge Einbindung aller Stakeholder zu Sicherstellung einer Nutzbarkeit entstehender Software in den Vordergrund.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Wie bereits in Abschnitt 3.1.2 angedeutet, können Stakeholder aufgrund ihrer unterschiedlichen Kompetenzprofile gemeinsam mit Technologieexperten Usability von Software sicherstellen. So entsteht ein Ansatz, der Gedanken der Sozialwissenschaften mit denen der Informatik verknüpft (Floyd et al. 1989). Allerdings muss bei der Gestaltung grafischer Systeme der Faktor Mensch in die Überlegungen mit einbezogen werden. Sollen Stakeholder, die keine Technologieexperten sind, in die Gestaltung von Softwaresystemen integriert werden, spielt die biologische Psychologie des Menschen eine große Rolle. Besonders die Art und Weise, wie Stakeholder die Interaktion mit einem technischen System wahrnehmen, beeinflusst maßgeblich dessen Gestaltung: „Die biologische Psychologie erforscht die Zusammenhänge zwischen biologischen Prozessen und Verhalten“ und die physiologische Psychologie beschreibt „[…] die interdisziplinäre Forschung über die Beziehungen zwischen Gehirn und Verhalten […]. Die physiologische Psychologie ist also ein Teilgebiet der biologischen Psychologie […]“ (Birbaumer/Schmidt 2002, 3).

Zunächst muss daher betrachtet werden wie Menschen von Softwaresystemen verursachte Sinnesreizungen wahrnehmen. Hier spielen vor allem das Sehen und das Höhere eine entscheidende Rolle. Im Bereich der Automotive Services kommt zusätzlich dem Tasten eine vermehrte Bedeutung zu, da viele Bedienelemente „blind“ erfühlt und genutzt werden müssen. Die beiden verbleibenden Sinneswahrnehmung des Menschen, das Riechen und Schmecken werden bislang kaum eingesetzt.

Abbildung 3-13 Farbkreis nach Johannes Itten
Quelle: (Wikipedia 2010c)

Aus den biologischen und psychologischen Erkenntnissen hinsichtlich der Sinneswahrnehmung des Menschen lässt sich eine Reihe von Designaspekten für Benutzeroberflächen ableiten. So kann durch verschiedene Farbtöne der Blick der Nutzer auf bestimmte Punkte gezogen und durch den Einsatz von Komplementärfarben (vgl. Abbildung 3-13) (Itten 1974) sowie gute Kontrastverhältnisse (Birbaumer/Schmidt 2002, 379) dessen Aufmerksamkeit gezielt gesteuert werden. Analysiert man die so genannten Sakkaden (vgl. Abbildung 3-14), also die sprunghaften Augenbewegungen die beim aufmerksamen Betrachten eines Objekts stattfinden, stellt man fest, dass das menschliche Auge dazu neigt zunächst Fixpunkte im linken oberen Viertel einer Benutzeroberfläche zu fokussieren um dann, ähnlich wie beim Lesen, nach rechts und schließlich nach unten zu wandern (Birbaumer/Schmidt 2002, 379).

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

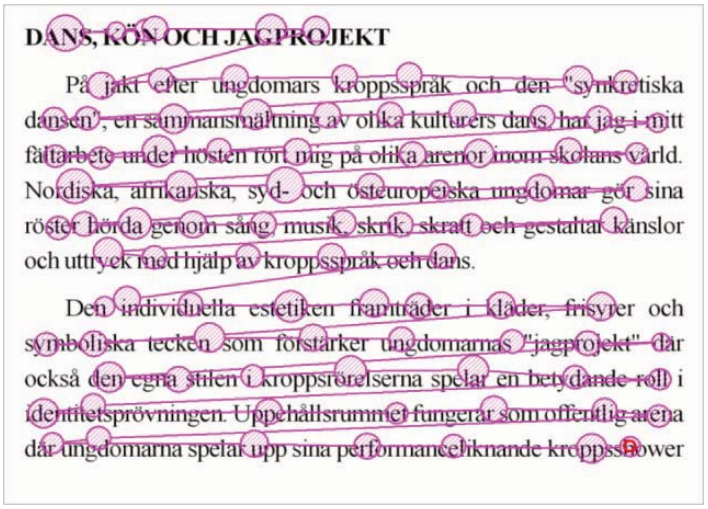


Abbildung 3-14 Sakkaden beim Lesen von Text
Quelle: (Wikipedia 2010d)

Neben dem Auge, was in gewisser Hinsicht den menschlichen Leitsinn darstellt, Spiel auch das Ohr und der Tastsinn eine bedeutende Rolle. So werden akustische Signale eingesetzt um Nutzern Informationen zu vermitteln ohne deren gesamte Aufmerksamkeit zu binden. Ein bekanntes Beispiel hierfür ist ein Signalton für das Eintreffen einer E-Mail oder einer SMS Nachricht. In Bereichen, in denen es für den Nutzer oft unmöglich ist visuell mit einem System zu interagieren wird Akustik auch genutzt um eine Interaktion zu ermöglichen (Nicolescu 2009, 22). Der Tastsinn des Menschen wird verstärkt im Bereich der Computerspiele eingesetzt. So kann über einen gezielten Einsatz von Vibration ein haptisches Feedback vermittelt werden. Ein weiteres Feld in dem haptische Interaktion mit Computersystemen eine bedeutende Rolle spielt ist der Einsatz von Braille Displays[12] für sehbehinderte Menschen.

3.4.4 Gestaltungsgesetze

Ausgehend von den Erkenntnissen über die Bedeutung des menschlichen Faktors für die Entwicklung gebrauchstauglicher Softwaresysteme zeigt sich, dass die Wahrnehmung einzelner Individuen bei gleicher Sinnesreizung stark unterschiedlich ausfallen. Das bedeutet, dass in der Wahrnehmungspsychologie unterschiedliche Erfahrungen und Persönlichkeiten zu unterschiedlichen Wahrnehmungen führen können. Diese Ansicht beruht auf der dualistischen Hypothese, welche annimmt, das seelische und neuronale Prozesse aufeinander einwirken, jedoch völlig unterschiedlichen Prozesses sind (Schmidt/Lang/Heckmann 2007, 275).

Es gibt jedoch auch zahlreiche Erkenntnisse im Bereich der Wahrnehmungspsychologie, die auf alle Menschen zutreffen. Beispielsweise neigt ein Mensch dazu Ganzheiten wahrzunehmen, wie beispielsweise die kognitiven Konturen des Kanizsa Dreiecks (vgl. Abbildung 3-10). Des Weiteren versucht die menschliche Psyche aus Einzelelementen der betrachteten Objekte stets Dinge zu bilden die über Gestaltqualitäten verfügen (Knauer 2002, 16). Das heißt der Mensch versucht stets einfache Formen zu sehen und ergänzt fehlende Stücke oder blendet Elementdetails aus. Knauer fasst diese allgemein gültigen Gesetzmäßigkeiten folgendermaßen zusammen (Knauer 2002, 16):

[12] Computer-Ausgabegerät für Blinde, das Zeichen in Brailleschrift darstellt.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *Gesetzt der Geschlossenheit:* Die Wahrnehmung neigt zum Schließen von Punkt- und Linienelementen, auch wenn dabei offene zu geschlossenen Umrissen werden, aber geschlossene Figuren werden gegenüber anderen Teilen, die nicht geschlossen sind oder weniger eindeutig verlaufen, dominiert.
- *Gesetz der Gleichheit:* Sind formal verschiedene Elemente gegeben, so neigt die menschliche Wahrnehmung dazu, die jeweils gleichartigen als zusammengehörige Figur aufzufassen.
- *Gesetz der zusammenfassenden Nähe:* Benachbarte Elemente, die den geringsten Abstand haben, werden zu einer Gruppe oder einer Figur zusammengefasst.
- *Gesetzt der einfachen Gestalt:* Ein nach Anzahl und Lage der Elemente einfacher Aufbau gilt als prägnant gegenüber einem komplexen; geometrische und symmetrische Gebilde werden bevorzugt, denn oft genügt es zur Erfassung der Formstruktur, einen Teil wahrzunehmen, da der Rest vom Bewusstsein hinzugefügt wird.
- *Gesetz von Figur und Grund:* Hier existiert eine Konkurrenz untereinander, wobei das Auffallendere stets als Figur gesehen wird und das Andere zum Hintergrund abgewertet wird.
- *Gesetz der Erfahrung:* Bekannte Formen können erkannt werden, auch wenn diese nicht geschlossen sind.
- *Gesetz der Innenseite:* Linien, die eine Tendenz aufweisen einen Innenbereich zu bilden oder zu umschließen werden bevorzugt wahrgenommen.
- *Gesetz der durchgehenden Linie:* Der Mensch geht davon aus, dass Linien einen einfachen und bekannten Verlauf haben der ähnlich einem Weg einer bestimmten Richtung folgt.

Die Aufgabe der Gestaltungsgesetze ist es, inhaltliche Schwerpunkte richtig zu setzen und den Nutzer so in seiner Interaktion mit dem System zu unterstützen. Beispielsweise kann das Lesen von Wörtern erleichtert werden wenn der Zwischenraum von Buchstaben kleiner als der doppelte Durchschnittsabstand ist (Galitz 2007, 199). Ein weiterer Aspekt der im Rahmen der Gestaltungsgesetze beachtet werden muss, ist die Theorie des Kurzzeitgedächtnisses (Miller 1994). Miller stellte fest, dass Menschen üblicherweise 5-9 Informationsartefakte in ihrem Kurzzeitgedächtnis behalten können. Eine geeignete Gestaltung von grafischen Benutzeroberflächen erlaubt es einem Entwickler mehr als diese Anzahl an Informationen gleichzeitig darzustellen ohne die Nutzer dabei zu überfordern (Sears/Jacko 2007, 51). Generell sollte ein Bildschirm jedoch nicht zu mehr als 30 % an Informationen ausgefüllt sein und Informationsobjekte sollten lokal um einen Fixpunkt angeordnet werden (Galitz 2007, 118). Generell sollten Benutzeroberflächen einen klaren Aufbau der Struktur aufweisen und Animationen, gerade am Blickfeldrand des Benutzers, nach Möglichkeit vermeiden, da diese Aufmerksamkeit auf sich ziehen und den Benutzer von seiner momentanen Tätigkeit ablenken. Analog zu den von Knauer beschriebenen Gestaltungsgesetzen fast Shneiderman acht goldene Designregeln für Gebrauchstauglichkeit und Softwareergonomie zusammen (Shneiderman 1997, 575):

(1) *Konsistenz:* Konsistenz bedeutet, dass ähnliche Ziele in ähnlichen Situationen auf gleichartige Weise durchgeführt werden können. Ein Beispiel hierfür ist der Schließen

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Knopf von Fenstern, der sich stets in der rechten oberen Ecke eines Fensters befindet[13].

(2) *Individuelle Benutzbarkeit:* Wie bereits diskutiert bringen unterschiedliche Nutzer unterschiedliche Erfahrungswelten und somit Vorlieben wie sie Systeme verwenden möchten mit. Die individuelle Benutzbarkeit trägt dem Rechnung, indem sie vorschreibt, dass es für gleichartige Funktionen unterschiedliche Lösungswege geben soll. So kann das Speichern von Dokumenten üblicherweise durch eine Tastenkombination, einen Mausklick auf ein Symbol oder eine Auswahl im Menü erfolgen.

(3) *Feedback:* Aktionen des Nutzers sollen stets ein sofortiges Feedback zurückgeben. Ist dies aufgrund einer beispielsweise längeren Ausführungszeit nicht möglich, so ist der Nutzer darüber zu informieren, da Reaktionszeiten die länger als etwa 2 Sekunden dauern den Eindruck einer Fehlfunktion vermitteln.

(4) *Abgeschlossenheit:* Der Nutzer soll stets den Überblick über den Fortschritt einer Interaktion behalten. Beispielsweise musste ein Fragebogen der mehrere Anzeigen erfordert einen Fortschrittsbalken oder einen Zähler bereitstellen der den bereits erfolgten Fortschritt und die noch ausstehenden Anzeigen darstellt.

(5) *Fehlervermeidung:* Typische Benutzungsfehlern die in einer Software durch ungenaues Verhalten der Nutzer entstehen können, sollten nach Möglichkeit im Vorfeld vermieden werden. Ein Beispiel hierfür ist die Warnung von Textverarbeitungen die den Nutzer auffordern das Dokument zu speichern, wenn dieser die Textverarbeitung schließt. So kann verhindert werden, dass ein versehentliches Löschen des Dokuments erfolgt.

(6) *Umkehrbarkeit:* Eine weitere Art der Fehlervermeidung ist die Umkehrbarkeit. Diese Regel fordert, dass der Nutzer stets in der Lage sein sollte eine gewisse Anzahl an Aktionen, die er ausgeführt hat, rückgängig zu machen. So kann beispielsweise das Löschen einer Grafik oder das Überschreiben eines Absatzes in einer Textverarbeitung einfach rückgängig gemacht werden.

(7) *Benutzerkontrolle:* Der Nutzer muss stets die Kontrolle über das Softwaresystem behalten. So dürfen auch langwierige Berechnungen oder ein Fehlverhalten des Systems nicht dazu führen, dass der Nutzer diese nicht abbrechen oder unterbrechen kann.

(8) *Entlastung des Kurzzeitgedächtnisses:* Wie bereits diskutiert ist das menschliche Kurzzeitgedächtnis nur in der Lage, 5-9 Elemente gleichzeitig zu erfassen (Miller 1994). Software sollte daher stets nur die aktuell notwendigen Informationen anzeigen und so eine Überforderung des Nutzers vermeiden.

3.4.5 Diskussion

Betrachtet man die Entwicklungen und Erkenntnisse aus dem Forschungsfeld der Mensch-Maschine Interaktion lassen sich zwei grundlegende Elemente identifizieren. Auf der einen Seite erfordert die zunehmende Komplexität interaktiver Softwaresystemen sowie gestiegene Verfügbarkeit leistungsfähiger Hardware eine enge Integration der Stakeholder in den Gestaltungsprozess. Auf der anderen Seite erfordert der Mensch als entscheidender Faktor bei der Gestaltung interaktive Systeme die Beachtung grundlegender Gestaltungs-gesetzmäßigkeiten und Designregeln.

[13] Bei Systemen wie Microsoft Windows; kann bei unterschiedlichen Herstellern auch abweichen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Systeme dienen nicht nur dazu, rechenintensive Aufgaben zu lösen, sondern haben vielmehr die Interaktion mit den Nutzern als zentrales Gestaltungsziel gewonnen. Somit steht nicht mehr ausschließlich die Umsetzung funktionaler Anforderungen sondern auch eine gebrauchstaugliche Gestaltung der Interaktion mit den Nutzern im Vordergrund. Mit der Entwicklung des Cognitive-Engineering stieg die Gestaltung dieser Gebrauchstauglichkeit, welche auf die Schnittstelle der Mensch-Maschine Interaktion fokussiert, in ihrer Bedeutung. (Herczeg 2006, 183). Die Integration des Cognitive-Engineering in den Softwareentwicklungsprozess führt schließlich zur Entwicklung des Usability-Engineering: "Die wissenschaftliche Disziplin und das systematische Studium, die/das sich mit der Aufklärung der Wechselwirkungen zwischen menschlichem und anderen Elementen eines Systems befasst, unter Berufszweig, der die Theorie, die Prinzipien, Daten und Methoden auf die (System) Gestaltung anwendet mit dem Ziel, das Wohlbefinden des Menschen und die Leistung des Gesamtsystems optimieren"(International Organization for Standardization 2004; Geis 2005).

Der Einsatz von Konzepten des Prototyping sowie der iterative Softwareentwicklung erweisen sich dabei als die geeignetsten Methoden um anhand von Nutzerinformationen Rückschlüsse auf das Design ziehen zu können und die Software anhand der gefundenen Anforderungen optimal auf die Nutzer anzupassen (Van Buskirk/Moroney 2003). Es wird also nicht mehr nur beschrieben was umgesetzt werden soll, sondern auch die Frage wie Software am Ende aussehen und benutzt werden soll.

Wie bereits diskutiert, spielt der menschliche Faktor bei der Gestaltung interaktiver Systeme eine entscheidende Rolle. Dabei müssen persönliche Vorlieben und individuelle Eigenarten einzelner Nutzer Berücksichtigung finden. Darüber hinaus gibt es noch eine ganze Reihe grundlegender Anforderungen und Regeln die eine Gebrauchstauglichkeit von interaktiven Softwaresystemen ermöglichen. Problematisch in diesem Zusammenhang ist, dass weder Stakeholder, mit ihren fachspezifischen Kompetenzen, noch Programmierer und Entwickler über die notwendigen Fähigkeiten und Voraussetzungen für das Design dieser Systeme verfügen (Norman 1996, 208). Dieses Wissen muss demzufolge von einer dritten Gruppe, den Designern, zur Verfügung gestellt werden.

3.5 Gestaltung von Infotainmentsystemen

Ein weiterer wesentlicher Punkt bei der Gestaltung von Automotive Services sind Erkenntnisse die aus der Entwicklung und dem Einsatz von Infotainmentsystemen abgeleitet werden können. Als Weiterentwicklung des klassischen Autoradios (Meroth/Tolg 2007b) vereinen Infotainmentsysteme das Informations- und Unterhaltungsangebot im Fahrzeug. Neben diesen beiden Elementen, aus denen sich der Begriff Infotainment[14] ableitet, werden auch Aspekte der Fahrerassistenz sowie der Kommunikation in Infotainmentsystemen abgebildet (Meroth/Tolg 2007b). Sie stellen somit die technische und konzeptionelle Grundlage für Automotive Services dar.

Die beiden für den Nutzer zentralen Hauptbestandteile von Infotainmentsystemen sind ein Display und ein dazugehöriges Bedienteil. Das Display ist meist in das Armaturenbrett zwischen Fahrer und Beifahrer integriert, es können jedoch auch zusätzliche oder alternative Elemente, wie die Darstellung im Fahrerinformationssystem (FIS) oder ein Head-Up Display als Ausgabemedium zum Einsatz kommen. Darüber hinaus sind Infotainmentsysteme mit der

[14] Information (Info-) und Entertainment (-tainment)

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Bordelektronik des Fahrzeugs verbunden, so dass sie eine Schnittstelle zu fahrzeugrelevanten Informationen bereitstellen und externe Geräte wie GPS oder mobile Datenanbindung integrieren.

Infotainmentsysteme sind bereits seit Mitte der 90er Jahre in Fahrzeugen etabliert (Meroth/Tolg 2007). Somit verfügen aktuelle Systeme über eine große Menge an Erfahrungen im operativen Einsatz beim Kunden. Für die Gestaltung erfolgreicher Automotive Services ist es daher unerlässlich, diese Systeme genauer zu betrachten um Gestaltungsgrundlagen, gerade im Bereich der Nutzerinteraktion, zu erlangen. Daher werden im Anschluss die aktuell am Markt erhältlichen Infotainmentsysteme der Hersteller Audi (Multi-Media-Interface), BMW (iDrive) und Mercedes-Benz (Command) untersucht. Der nachfolgende Abschnitt fast die wesentlichen Merkmale der betrachteten Infotainmentlösungen zusammen und diskutiert die daraus ableitbaren Gestaltungselemente für Automotive Services.

3.5.1 Betrachtung etablierter Ansätze

Bei der Betrachtung der Infotainmentsysteme der Hersteller Audi, BMW und Mercedes-Benz wird der Schwerpunkt auf die Analyse der Interaktion mit den Nutzern gelegt. Hierbei werden insbesondere das Bedienteil und das verfügbare Display, der Funktionsumfang, das Bedienkonzept sowie grafische Gestaltungselemente betrachtet. Auf eine Analyse der Systeme hinsichtlich ihrer technischen Spezifikation oder verfügbaren Hardwarefunktionalität wird mit Hinblick auf die Analyse der Herausforderungen der Gestaltung von Automotive Services verzichtet.

Die Untersuchung der Infotainmentsysteme wurde im Rahmen einer Forschungsarbeiten (Forrai 2010) bei der Audi AG durchgeführt. Im Rahmen dieser Arbeit wurden die Seriensysteme der drei Hersteller in aktuellen Fahrzeugen (bzw. an einem Testaufbau eines Vorserienmodells bei Audi) untersucht.

3.5.1.1 Audi Multi-Media-Interface

In der Studie Avantissimo welche im Jahr 2001 auf der Internationale Automobil-Ausstellung (IAA) präsentiert wurde, stellt Audi erstmals ein Infotainmentsysteme für seine Fahrzeuge vor (Schätzl 2001). Im Jahr 2003 wurde das Multi-Media-Interface genannte Infotainmentsysteme (MMI) im Audi A8 (Typ D3) erstmals in einem Serienfahrzeug angeboten. Im aktuellen Audi A8 (Typ D4) ist mittlerweile die Version 3G+ verbaut.

Zum Zeitpunkt der Analyse war das aktuelle MMI 3G+ noch nicht in einem Serienfahrzeug verfügbar und wurde daher anhand eines Versuchsaufbaus, der aus den Originalkomponenten der Serie zusammengestellt war, durchgeführt. Anzumerken ist hierbei, dass der Testaufbau nicht über das im aktuellen Audi A8 verbaut Touchpad zur Eingabe von Ziffern und Buchstaben verfügte. Auch war die eingespielte Softwareversion nicht vollständig, so dass Teile des Kartenmaterials für die Navigation sowie die Funktion Car, welche die direkt fahrzeugbezogenen Informationen und Einstellungen beinhaltet, nicht verfügbar waren.

3.5.1.1.1 Bedienteil und Display

Das Audi Multi-Media-Interface benutzt zur grafischen Interaktion mit dem Fahrer ein im Armaturenbrett des Fahrzeugs integriertes farbiges Display, welches mit einer Auflösung von 800x480 Pixel angesteuert wird. Die Bedienung des Infotainmentsystems erfolgt primär mittels eines Dreh-Drück-Stellers (Push-Turn-Button, PTB) sowie durch zusätzliche Funktionstasten, mit deren Hilfe sich speziell definierte Funktionen aufrufen lassen. Diese

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Funktionen stellen die Hauptbereiche des MMI dar, deren Aktivierung mit einer roten Kontrolllampe neben der entsprechenden Funktionstaste signalisiert wird. Neben den Funktionstasten gibt es noch die so genannten Steuerungstasten (Softkey) mit denen spezielle Aktionen in der Benutzeroberfläche gesteuert werden können, sowie einen Return-Taste und eine Vorwärts- und Rückwärtstaste. Wie bereits einleitend erwähnt, verfügt die Serienversion der MMI 3G+ über ein Touchpad zur Eingabe von Buchstaben und Ziffern. Abbildung 3-15 zeigt das Bedienteil einer MMI 3G+ mit dem zentral angeordneten Dreh-Drück-Steller, den vier silbernen Steuerungstasten, die über dem Dreh-Drück-Steller angeordneten Funktionstasten sowie dem Touchpad auf der linken Seite des Bedienteils.

Abbildung 3-15 Bedienteil Audi MMI 3G+
Quelle: (Audi AG 2010c, 41)

3.5.1.1.2 Funktionsumfang

Das Audi Multi-Media-Interface bietet Funktionen aus vier Bereichen an: Multimedia, Navigation, Assistenz und Kommunikation. Über die acht Funktionstasten des Bedienteils kann der Anwender die einzelnen Funktionen der vier Bereiche aufrufen. Der Bereich Multimedia umfasst dabei die Funktionen Radio, Media und Tone, die es ermöglichen neben der Wiedergabe von Medien unterschiedlichster Quellen analoges und digitales Radio abzuspielen. Der zweite Bereich des MMI befasst sich mit Navigation. Über die Funktionstasten Nav und Info lassen sich die integrierte Routennavigation, TMC-Nachrichten sowie die in Abschnitt 2.3.1.1 vorgestellten Audi Online Dienste für Wetter oder Nachrichten aufrufen. Der Bereich Kommunikation wird über die Funktionstaste Tel aktiviert. Er beinhaltet ein, über eine Freisprecheinrichtung nutzbares Telefon sowie ein Adressbuch. Der letzte Bereich, die Assistenz, wird über die Funktionstaste Car erreicht. Hier können Nutzer Einstellungen am Fahrzeug und an den Fahrerassistenzsystemen vornehmen. Neben der direkten Anwahl der Funktionsbereiche über die Funktionstasten können diese auch über ein Hauptmenü, welches über die Funktionstaste Menu erreicht werden kann, genutzt werden.

Innerhalb der einzelnen Funktionen können die Steuerungstasten zur Anwahl spezieller Aktionen genutzt werden. Die Anordnung der Tasten auf dem Bedienteil entspricht dabei der Darstellung der Aktionen in den vier Ecken des Displays.

3.5.1.1.3 Bedienkonzept

Grundsätzlich lässt sich das Bedienkonzept der MMI 3G+ in zwei unterschiedliche Bereiche gliedern. Auf der einen Seite können Funktionsbereiche über das Hauptmenü oder die

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Funktionstasten aufgerufen werden, zum anderen können Anzeigen innerhalb der Funktionsbereiche über graphische Softkeys, die über die Steuerungstasten ausgewählt werden, aktiviert werden. Die Hierarchie der einzelnen Anzeigen ist hierbei in einer Baumstruktur hierarchisch aufgebaut, deren zentraler Knoten das Hauptmenü darstellt.

Wird ein Funktionsbereich erstmalige aufgerufen, wird die Startanzeige des entsprechenden Bereichs angezeigt. Wie bereits beschrieben ist es dabei unerheblich, ob der Funktionsbereich über die Funktionstaste oder das Hauptmenü aufgerufen wird. Wird die Funktionstaste erneut betätigt, so gelangt der Nutzer zurück ins Hauptmenü. Wechselt der Nutzer hingegen zwischen verschiedenen Funktionsbereichen, so wird bei Rückkehr in einen bereits aktivierten Funktionsbereich die zuletzt aktive Anzeige wieder aktiviert und der Fokus auf das zuletzt fokussierte Elemente gesetzt.

Innerhalb der Funktionsbereiche werden so genannte Softkey-Menüs durch die Steuerungstasten aktiviert. Hierbei ist es im Gegensatz zu den Funktionstasten unerheblich, ob der Nutzer das Menü bereits aufgerufen hat oder es zum ersten Mal betritt. Das Betätigen der Steuerungstaste führt stets zur initialen Anzeige des Softkey-Menüs und der Fokus wird auf das erste Objekt gesetzt.

Wie bereits erwähnt, ist die Struktur der MMI hierarchisch angelegt. Mithilfe der Return-Taste kann der Nutzer zur jeweils nächst höheren Ebene zurückspringen, unabhängig davon, wie er die aktuelle Anzeige erreicht hat. Die Logik ist somit nicht historisch sondern hierarchisch angelegt und endet mit dem Hauptmenü.

Eine Besonderheit bei der Funktionalität des Dreh-Drück-Stellers ist, dass dessen Bewegungen direkt umgesetzt werden. Eine Drehbewegung nach links oder rechts hat somit einen direkten Wechsel des Fokus auf das entsprechend nächste Element zur Folge. Tabelle 3-18 fasst das grundsätzliche Bedienkonzept der MMI 3G+ zusammen.

Aktion	Verhalten
Funktionstaste	*Erstmalige Betätigung:* Anzeige der erste Ebene *Erneute Betätigung:* Anzeige der letzten angezeigten Ebene *Fokus:* Speicherung des zuletzt fokussierten Objekts
Steuerungstaste	*Erstmalige Betätigung:* Anzeige der ersten Ebene *Erneute Betätigung:* Anzeige der ersten Ebene *Fokus:* Fokussierung des ersten Objekts
Return-Taste	Hierarchisch
PTB	*Links-Drehung:* Fokus nach unten *Rechts-Drehung:* Fokus nach oben

Tabelle 3-18 Bedienkonzept Audi MMI 3G+
Quelle: Eigene Darstellung in Anlehnung an (Forrai 2010, 33)

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

3.5.1.1.4 Graphische Gestaltungselemente

Die Darstellung des Audi Multi-Media-Interface erfolgt auf einem zentralen, im Armaturenbrett integrierten, Display. Die Darstellung betrug beim untersuchten Versuchsaufbau 800x480 Pixel, was auch der Auflösung in der aktuellen Serienausführung entspricht. Der dargestellte Bildschirm lässt sich wie in Abbildung 3-16 zu sehen von oben nach unten in fünf Bereiche untergliedern.

Der erste Bereich enthält links und rechts einen Softkey (diese entsprechen der linken und rechten oberen Steuerungstaste am Bedienteil) sowie einem Feld für den Titel der aktuellen Anzeige zwischen den beiden Softkeys. Der zweite Bereich direkt darunter stellt den Untertitel der aktuellen Anzeige dar und erstreckt sich über die gesamte Breite der Anzeige. Der nächste Bereich ist der Hauptanzeige, welche mit unterschiedlichen Inhalten gefüllt werden kann die im Nachfolgenden genauer erläutert werden. Anschließend an die Hauptanzeige folgt eine Zeile die analog dem ersten Bereich die verbleibenden beiden Softkey beinhaltet. Der Bereich zwischen den beiden Softkeys (diese entsprechen der linken und rechten unteren Steuerungstaste am Bedienteil) bleibt dabei frei. Der letzte Bereich findet sich im unteren Bildschirmrand und beherbergt eine Statusleiste an der, wie in Abbildung 3-16 zu sehen ist, die Verfügbarkeit von TMC-Nachrichten oder die Stärke der aktuellen Funknetzverbindungen angezeigt wird.

Abbildung 3-16 Aufbau Audi Multi-Media-Interface
Quelle: (Forrai 2010, 34)

Die Farbgebung des Audi MMI orientiert sich an den vier Funktionsbereichen die in Abschnitt 3.5.1.1.2 beschrieben wurden. Der Bereich Information und Navigation wird in blau, der Bereich Car in Rot, der Bereich Multimedia (Radio, Media und Tone) in Orange und der Kommunikationsbereich im Funktionsmenü Telefon in Grün dargestellt. Zusätzlich gibt es für die Darstellung neutraler Inhalte, wie beispielsweise dem Hauptmenü, eine weiße Farbgebung.

Der Hauptbereich der Anzeige kann je nach Funktion und gewünschter Interaktion mit dem Nutzer verschiedener Anzeigen beherbergt. Diese werden unter Berücksichtigung der entsprechenden Farbgebung in unterschiedlichen Funktionsbereichen durchgängig verwendet. Der Hauptbereich stellt auf der linken Seite eine Scrollleiste dar, so dass die einzelnen Anzeigen größer sein können als der auf dem Bildschirm verfügbare Platz. Einzige Ausnahme

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

stellen Pop-Up Anzeigen dar, die kurzzeitig über die aktuelle Anzeige geblendet werden. Ein Beispiel für ein solches Pop-Up ist die Änderung der Raumtemperatur, die über der aktuellen Anzeige eingeblendet wird. Insgesamt stehen acht unterschiedliche Anzeigen zur Verfügung (Forrai 2010, 34ff): Wizard, Liste, Speller, RotaryKnob, Fadenkreuz, Dialog, Browser und Karte.

Der Wizard ist eine grafische Darstellung eines Menüs, bei dem mehrere Objekte auf einer Ellipse angeordnet sind, die durch Drehen des Dreh-Drück-Stellers ausgewählt werden. Die Darstellung der einzelnen Objekte kann textuell, grafisch oder textuell und grafisch sein. Zusätzlich kann der Wizard optional über ein Hintergrundbild verfügen. Abbildung 3-17 zeigt verschiedene Varianten des Wizards.

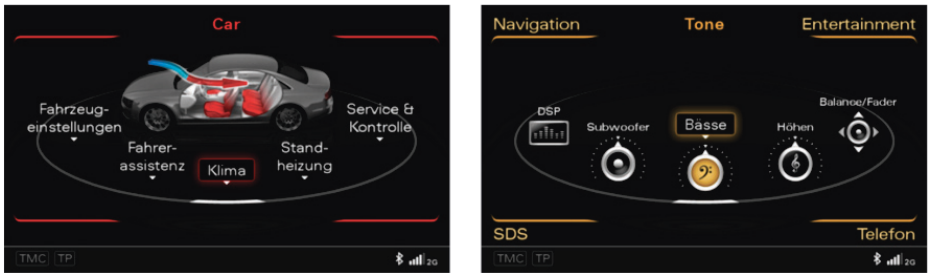


Abbildung 3-17 Darstellung eines Wizards (Audi MMI)
Quelle: (Forrai 2010, 35)

Listen werden benutzt um eine Menge an Elementen darzustellen und diese mit dem Dreh-Drück-Steller auszuwählen. Die einzelnen Elemente der Liste können dabei eine Checkbox, komplexe Elemente (bestehend aus Bild und Text), ausklappbare Unterlisten, Drop-Down Liste zur Auswahl von Optionen, sowie ein Piker, mit dem Zeiteinstellungen vorgenommen werden können, sein. Abbildung 3-18 zeigt verschiedene Listen.

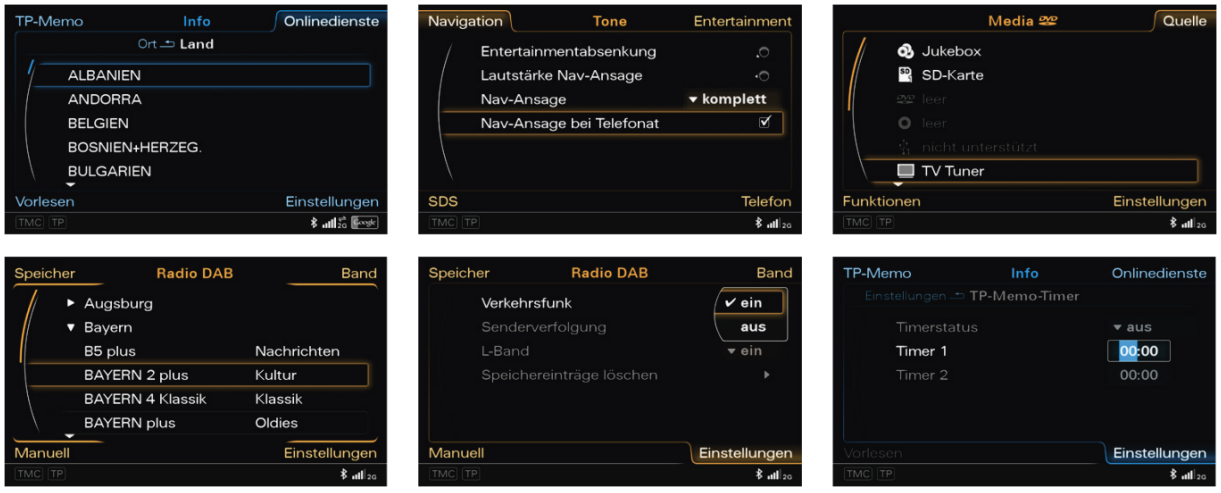


Abbildung 3-18 Darstellung einer Liste (Audi MMI)
(Liste, Checkbox, Komplexe Liste, Ausklappbare Liste, Drop-Down Liste und Picker)
Quelle: (Forrai 2010, 35ff)

Texteingabe erfolgt mittels eines so genannten Spellers. Er stellt ein kreisförmig angeordnetes Alphabet auf der linken Seite des Bildschirms dar, in dessen Mittelpunkt ein rundes Auswahlobjekt dargestellt wird. Auf der linken Seite des Bildschirms befindet sich das Eingabefeld in dem die einzelnen selektierten Buchstaben eingetragen werden. Unter dem Eingabefeld besteht die Möglichkeit, passende Begriffe, beispielsweise aus dem Adressbuch, anzuzeigen. Abbildung 3-19 zeigt einen Speller zur Eingabe des Vornamens für das Telefonbuch.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Abbildung 3-19 Darstellung eines Spellers (Audi MMI)
Quelle: (Forrai 2010, 37)

Weitere zur Verfügung stehende Anzeigen der sogenannte RotaryKnob, der optisch ähnlich dem Speller eine Auswahl auf einer Skala erlaubt. Mithilfe des Fadenkreuzes kann eine Position auf einer Grafik ausgewählt werden was beispielsweise bei Einstellung von Balance und Fader im Sound System oder der Auswahl von Points of Interest (POI) in der Navigation Anwendung findet. Dialoge stellen eine besondere Form der Liste dar, die im oberen Bereich der Anzeige erklärenden Text und im unteren Bereich eine Liste mit Auswahloptionen beinhalten. Die letzten beiden verfügbaren Anzeigen sind ein Browser sowie die Kartendarstellung der Navigation. Abbildung 3-20 gibt einen Überblick über diese Anzeigeelemente.

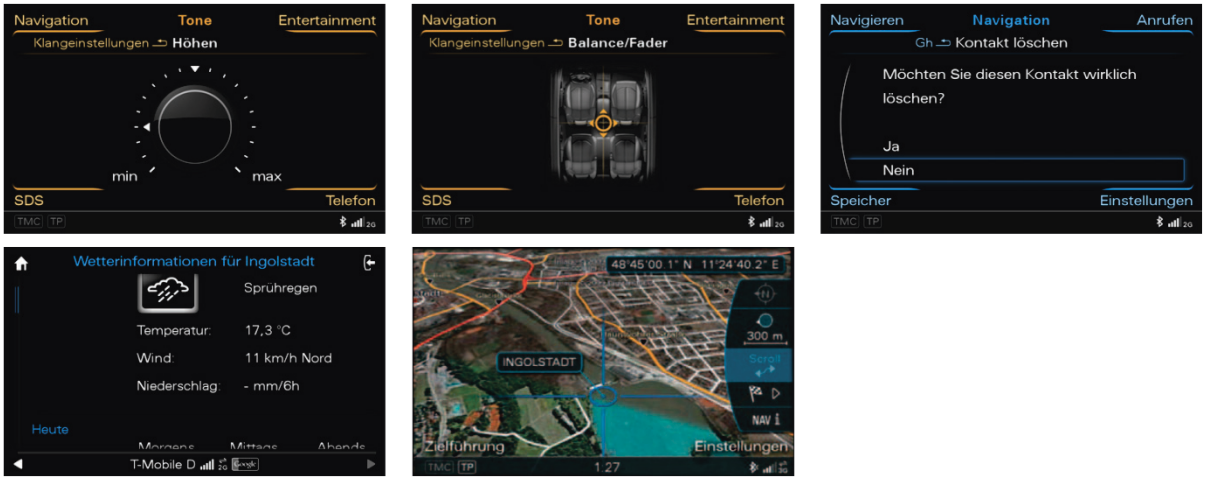


Abbildung 3-20 Darstellung weitere Anzeigen (Audi MMI)
(RotaryKnob, Fadenkreuz, Dialog, Browser und Karte)
Quelle: (Forrai 2010, 37; Audi AG 2010, 90)

3.5.1.2 BMW iDrive

Das Infotainmentsystemen von BMW trägt den Namen iDrive und wurde erstmals 2001 mit der BMW 7er Reihe eingeführt. Nach eingangs schlechten Kritiken wurde das System nochmals überarbeitet und ist heute in Version 2 verfügbar (Autosieger 2008). Analysiert wurde das System in einem BMW 730d (Baujahr 2008).

3.5.1.2.1 Bedienteil und Display

Analog zum Multi-Media-Interface von Audi ist das iDrive Infotainmentsystemen von BMW im Armaturenbrett des Fahrzeugs und der Mittelkonsole installiert. Das Anzeigedisplay hat eine Auflösung von 1280x480 Pixel und befindet sich mittig im Armaturenbrett zwischen Fahrer und Beifahrer. Das Bedienteil hat als zentrales Steuerelement einen Dreh-Drück-Steller sowie oberhalb angeordnet vier Funktionstasten für CD, Telefon, Radio und Navigation. Zusätzlich findet sich auf dem Bedienteil eine Back-Taste und eine Option-Taste. Abbildung 3-21 zeigt das Controller genannte Bedienteil mit dem zentralen Dreh-

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Drück-Steller, den darüber angeordneten Funktionstasten sowie der darunter befindlichen Back- und Option-Taste.

Abbildung 3-21 Controller BMW iDrive
Quelle: (BMW Group 2010c)

3.5.1.2.2 Funktionsumfang

Die Funktionen des BMW iDrive umfassen die vier Bereiche Multimedia, Navigation, Assistenz und Kommunikation. Im Bereich Multimedia werden die Funktionen CD und Multimedia (externer Media Player), digitales und analoges Radio sowie digitales Fernsehen zusammengefasst. Hinter dem Bereich Navigation befindet sich das integrierte Navigationssystem mit Staumeldungen und Points of Interest. Der Bereich Assistenz wird durch die Funktion Fahrzeuginformation vertreten und im Bereich Kommunikation finden sich eine Telefonfunktion sowie der Zugang zu den BMW Onlinediensten, innerhalb derer speziell angepasste Onlineinhalte sowie freies Browsen nutzbar sind.

Im Gegensatz zum Multi-Media-Interface von Audi können nicht alle Funktionsbereiche direkt über eine Funktionstaste erreicht, sondern lediglich über das Auswahlmenü ausgewählt werden. Die Funktionen CD, Telefon, Radio und Navigation sind die einzigen mit Funktionstasten direkt erreichbaren Bereiche. Zusätzlich haben alle Bereiche noch ein Optionsmenü, welches durch die Optionstaste am Bedienteil erreicht werden kann. Alternativ können diese auch durch das Drücken des Controllers nach rechts erreicht werden.

3.5.1.2.3 Bedienkonzept

Das Bedienkonzept von iDrive ist hierarchisch aufgebaut und beginnt, wie auch das System von Audi, mit einem Hauptmenü als Ausgangspunkt. Jede der achte im iDrive vorhandenen Funktionen verfügt wiederum über ein eigens Hauptmenü sowie hierarchisch darunter angeordnete Anzeigen. Die unterste Anzeige dieses Stapels ist jeweils das Optionsmenü. Bei der Auswahl der einzelnen Funktionsbereiche unterscheidet sich das Verhalten des Systems je nachdem, ob eine Funktion mit der Funktionstaste am Bedienteil oder einer Auswahl im Funktionsmenü aktiviert wurde.

Wird eine Funktion aktiviert indem der Benutzer die Funktionstaste drückt, so gelangt er direkt in die zweite Ebene der ausgewählten Funktion und nicht in deren Funktionsmenü. Erst ein wiederholtes Betätigen der Funktionstaste ruft schließlich das Funktionsmenü auf. Ein erneutes Betätigen der Funktionstaste führt den Nutzer wieder zurück in das Hauptmenü. Wurde ein Menü vom Nutzer bereits früher schon einmal aufgerufen, so gelangt dieser direkt zur zuletzt aktiven Anzeige.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Wird eine Funktion hingegen im Hauptmenü ausgewählt, so wird bei erstmaliger Auswahl die oberste Ebene der jeweiligen Funktion angezeigt. Analog zum Aufruf durch die Funktionstaste wird bei wiederholtem Aufruf einer Funktion die zuletzt aktive Ebene aktiviert. Der Fokus wird dabei nicht auf das zuletzt selektierte, sondern auf das zuletzt aktivierte Objekt gesetzt. Ein Betätigen der Menütaste führt den Nutzer zurück zum Hauptmenü.

Die einzelnen Anzeigen werden beim BMW iDrive, mit Ausnahme des Hauptmenüs, nicht als Vollbild anzeigen sondern werden als übereinanderliegende Layer realisiert. Durch Drücken des Controllers wechselt die Anzeige zur nächst höheren beziehungsweise nächst niedrigeren Ebene. Da die Struktur von iDrive von unten nach oben gestapelt wird, befindet sich das Hauptmenü auf der untersten Ebene und die Optionsmenüs jeweils auf der obersten Ebene. Ein Drücken nach links führt somit zur nächsten darunter liegende Ebene, ein Drücken nach rechts zur nächsthöheren Ebene oder wenn diese nicht mehr vorhanden ist zum Optionsmenü.

Ein Drehen des Dreh-Drück-Stellers verschiebt den Fokus auf das nächste oder vorangegangene Element einer Anzeige. Dabei verfolgte die iDrive Logik einen spiegelverkehrten Ansatz zum Audi Multi-Media-Interface. Eine Linksdrehung verschiebt den Fokus nach oben und eine Rechtsdrehung nach unten. Im Gegensatz zum MMI wird lediglich das nächste Element ausgewählt, ohne dass dabei der Fokus direkt verschoben wird.

Die Back-Taste ist bei iDrive historisch belegt. Ein Druck auf diese führt den Nutzer auf den zuletzt angezeigten Bildschirm, unabhängig davon wo sich der Nutzer in der Hierarchie gerade befindet. Tabelle 3-19 fasst das grundsätzliche Verhalten der Bedienung des BMW iDrive zusammen.

Aktion	Verhalten
Funktionstaste	*Erstmalige Betätigung:* Anzeige der zweiten Ebene *Erneute Betätigung:* Anzeige der letzten angezeigten Ebene *Fokus:* Fokussierung des zuletzt aktiven Objekts
Funktionsmenü	*Erstmalige Betätigung:* Anzeige der erste Ebene *Erneute Betätigung:* Anzeige der letzten angezeigten Ebene *Fokus:* Fokussierung des zuletzt aktiven Objekts
Back-Taste	Historisch
Controller	*Links-Drehung:* Fokus nach oben *Rechts-Drehung:* Fokus nach unten *Links-Drücken:* Wechsel der Anzeige zur darunterliegenden Ebene *Rechts-Drücken:* Wechsel der Anzeige zur nächsthöheren Ebene falls vorhanden, ansonsten Wechsel in das Optionsmenü
Option-Taste	Anzeige des Optionsmenüs

Tabelle 3-19 Bedienkonzept BMW iDrive
Quelle: (Forrai 2010, 46)

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

3.5.1.2.4 Graphische Gestaltungselemente

Grundsätzlich lassen sich zwei unterschiedliche Arten der Aufteilung der Anzeige im iDrive unterscheiden. In der untersten Ebene lässt sich der Bildschirm in vier Bereiche unterteilen. Im linken Bereich wird das Hauptbereich dargestellt, darüber befindet sich ein Feld in dem Titel des aktuellen Bereichs angezeigt wird. Auf der rechten Seite des Bildschirms befindet sich eine Vorschau auf die nächst höhere Ebene, die beim Drücken des Controllers nach rechts aktiviert wird. Diese Anzeige ist ausgegraut, so dass für den Nutzer nur die linke Hälfte des Bildschirms aktiv ist. Rechts neben der Vorschau befindet sich ein graphischer Indikator, der das Vorhandensein eines Optionsmenüs anzeigt. Ebenfalls in Abbildung 3-22 zu sehen ist auf der rechten Seite die Aufteilung des Bildschirms der zweiten Ebene. Die Anzeige der zweiten Ebene wird in Form eines Layers über die Anzeige der ersten Ebene dargestellt, welche am linken Bildschirmrand noch darunter liegend zu erkennen ist. Die Anzeige der zweiten Ebene gliedert sich in die Bereiche Titel (links oben), Statusleiste (rechts oben) und Hauptbereich. Auf der rechten Seite des Bildschirms ist wiederum der Indikator für das Optionsmenü zu sehen. Weitere Ebenen werden analog der Darstellung der zweiten Ebene in Form von Layern über die darunter liegenden Ebenen geblendet. Diese darunter liegenden Ebenen bleiben dabei auf der linken Bildschirmseite sichtbar.

Abbildung 3-22 Aufbau BMW iDrive
(links: erste Ebene; rechts: zweite Ebene)
Quelle: (Forrai 2010, 47)

Die Farbgebung des iDrive orientiert sich, analog dem Audi MMI, an den dargestellten Funktionsbereichen. BMW unterscheidet hier drei Systemfarben. Das Farbschema Blau wird für Kontakte, BMW Dienste, Einstellungen, Fahrzeuginformationen sowie das Telefon benutzt. Der Multi Media Bereich mit CD und Radio verwendet eine beige Anzeige. Für die Navigation wird die Grundfarbe Grün genutzt. Die Farbgebung beeinflusst dabei lediglich die Überschriften und Eingrenzung des Hauptbereichs, die restlichen Anzeigen behalten ihren graublauen Grundton bei.

Für den Fall, dass die Anzeige im Hauptbereich mehr als den zur Verfügung stehenden Platz einnimmt, sieht BMW zwei unterschiedliche Mechanismen vor dieses dem Nutzer anzuzeigen. Variante eins ist ein Scrollbalken, der am rechten Bildschirmrand (links des Indikators für das Optionsmenü) dargestellt wird. Die zweite Variante ist die Aufteilung der Anzeige in einzelne Seiten, die mit einem Seitenzähler oberhalb der Anzeige angezeigt wird. Besteht eine Anzeige beispielsweise aus drei Seiten, so wird dies mit der Darstellung „2/3“ verdeutlicht. Der Nutzer befindet sich also auf der zweiten von drei Seiten.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Zur Darstellung einer Fortschrittsanzeige wird ein Prozessgauge benutzt. Dieses Gauge wird im iDrive innerhalb von Fortschrittsdialogen verwendet und animiert dargestellt. Es macht jedoch keine Angaben zum Fortschritt sondern stellt lediglich eine Aktivität des Systems dar.

Der Fokus eines Objekts im Infotainment System von BMW wird durch eine rote Umrandung des Objekts dargestellt. Jede Anzeige des iDrive enthält auf der linken Seite eine graphische Darstellung des Controllers, die den Wechsel zwischen einzelnen Anzeigen andeutet. Das aktuelle fokussierte Objekt der Anzeige ist mit dieser Repräsentation des Controllers durch eine rote Linie verbunden. Ist in einer Anzeige, beispielsweise einer Textanzeige, kein Objekt fokussierbar wird der Scrollindikator auf der linken Bildschirmseite fokussiert (Abbildung 3-23 links). Abbildung 3-23 zeigt die beiden Varianten des Selektors.

Abbildung 3-23 Darstellung des Selektors (BMW iDrive)
(links: Scrollindikator; rechts: Scrollleiste)
Quelle: (Forrai 2010, 50)

Das zentrale Gestaltungselemente zur Darstellung von Informationen und Interaktion mit dem Nutzer im BMW iDrive sind Listen. Neben einfachen Listen, die lediglich kurzen Text darstellen, sind auch komplexe Listenelemente mit mehreren Textfeldern und Grafiken möglich (vgl. Abbildung 3-24). Darüber hinaus kennt iDrive noch das Konzept der Checkboxen, so dass einzelne Listenelemente vom Nutzer an und abgewählt werden können. Ein weiteres Element das in Listen zur Anwendung kommt, ist der so genannte Piker. Diese spezielle Art der Zahleneingabe, die sich auch im Audi Multi-Media-Interface findet, wird verwendet, um speziell formatierte Werte, wie Uhrzeit oder Datum, einzugeben. Alternativ können Werte auch durch Slider dargestellt werden, die grafisch in Form eines Balkens Werte zwischen einem Minimum links und einem Maximum rechts anzeigen. Sowohl Picker als aus Slider werden durch Drücken des Controllers aktiviert und durch Drehen desselben verändert. Ein erneutes Drücken des Controllers beendet die Eingabe und speichert den Wert.

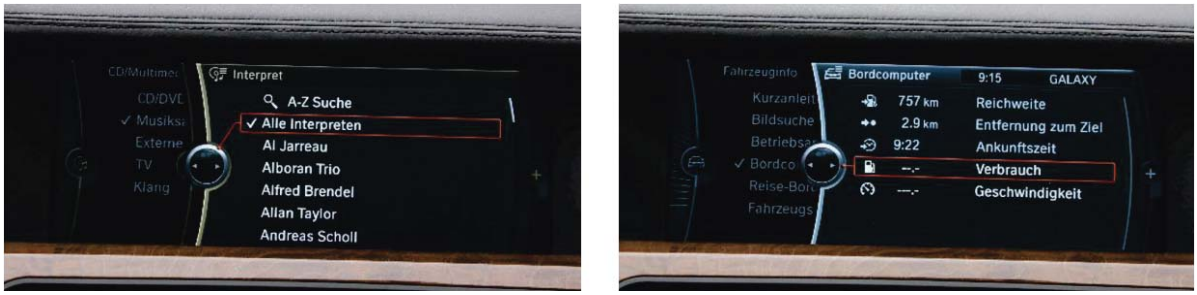


Abbildung 3-24 Darstellung einer Liste (BMW iDrive)
(Links: Einfache Liste; Rechts: Komplexe Liste mit einer Grafik und zwei Textfeldern)
Quelle: (Forrai 2010, 48)

Die Texteingabe im iDrive erfolgt über einen Speller. Das Alphabet wird in einem Kreis rund um den grafisch dargestellten Controller angeordnet. Durch Drehen des Controllers können die Elemente des Alphabets ausgewählt werden. Über dem Speller befindet sich ein Textfeld

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

in dem die Eingabe dargestellt wird. Rechts des Spellers werden gefundene Eingaben, beispielsweise Adressen, in Form einer Liste dargestellt (vgl. Abbildung 3-25).

Abbildung 3-25 Darstellung des Spellers (BMW iDrive)
Quelle: (Forrai 2010, 49)

Weitere verfügbare Anzeigen sind Dialoge, Browser, Kartendarstellung sowie ein ImageMapper. Dialoge stellen einen Text oder eine Fortschrittsgauge dar und bieten dem Nutzer eine Liste an Auswahlmöglichkeiten für seine Antwort. Der Internetbrowser unterstützt zwei unterschiedliche Modi. Im ersten Modus werden spezielle von BMW erstellte Informationen zu Reisen und Auskunftsdiensten sowie Nachrichten dargestellt. Grafisch orientiert sich diese Anzeige an den restlichen Anzeigetypen des iDrive. Eine weitere Option ist der freie Internetbrowser, der im Gegensatz zu den restlichen Anzeigen als Vollbild dargestellt wird. Er besteht aus einer Menüleiste am linken Bildschirmrand sowie dem eigentlichen Browserfenster, welches den restlichen Bildschirm einnimmt. Eine Besonderheit im iDrive ist der so genannte ImageMapper. Analog einer HTML ImageMap werden Bilder mit grafischen Marken versehen die über den Controller ausgewählt werden können. Diese Darstellung wird z.B. zur Anzeige der Bedienungsanleitung verwendet.

Abbildung 3-26 zeigt von links nach rechts die Darstellung eines Dialogs, Onlinedienste im Browser, freier Internetbrowser sowie ImageMapper.

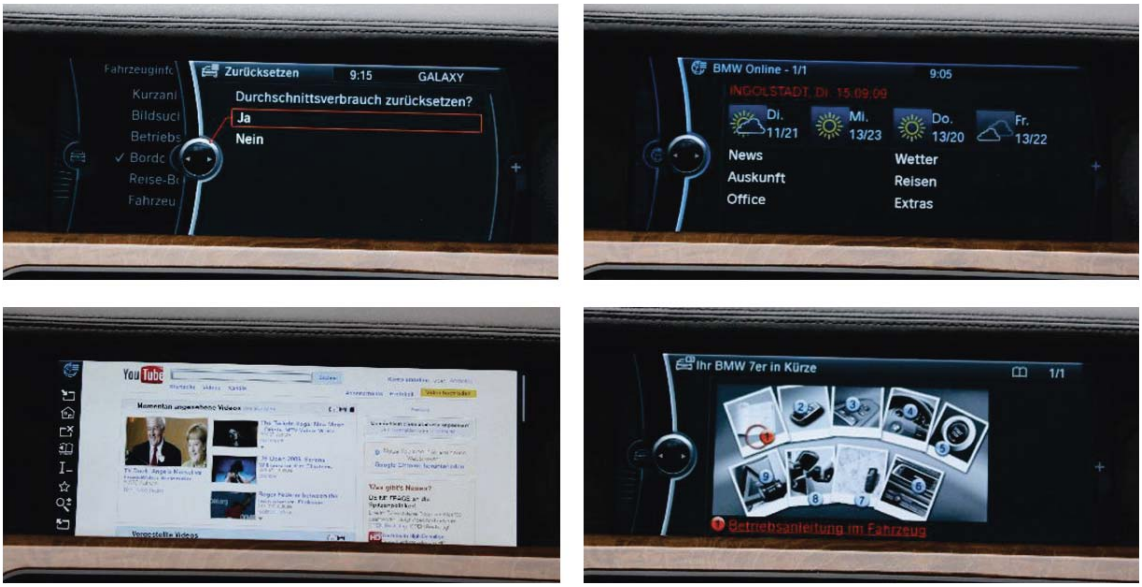


Abbildung 3-26 Darstellung weitere Anteigen (BMW iDrive)
(Dialog, Onlinedienst im Browser, Freier Internetbrowser, ImageMapper)
Quelle: (Forrai 2010, 51ff)

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

3.5.1.3 Mercedes-Benz COMAND

Das Infotainmentsystemen von Mercedes Benz wurde erstmalig 1996 unter dem Namen COMMAND eingeführt (Hanrieder 2004). Die Abkürzung COMMAND steht dabei für Cockpit Management and Navigation Device (Cunningham 2007). Analysiert wurden die aktuelle Version von COMMAND in einem Mercedes S-Klasse S450 Hybrid aus dem Jahr 2008.

3.5.1.3.1 Bedienteil und Display

Das Infotainmentsystemen von Mercedes Benz (MB) ist, wie auch die Systeme von Audi und BMW, in das Armaturenbrett und die Mittelkonsole des Fahrzeugs integriert. Im Armaturenbrett befindet sich das Display, welches eine 16:9 Auflösung bietet. Daneben befinden sich Funktionsknöpfe zur Einstellung des Displays. Integriert in die Mittelkonsole befindet sich das Bedienteil, welches einen Dreh-Drück-Steller, zwei Funktionstasten eine Zurücktaste und eine Tastatur für die Nummerneingabe der Telefonfunktion besitzt. Besonders zu bemerken ist beim Dreh-Drück-Steller, dass dieser in der Lage ist dem Nutzer beim Erreichen des Endes einer Liste ein haptisches Feedback zu geben. Dies wird realisiert, indem ein weiteres Drehen des Dreh-Drück-Stellers in die entsprechende Richtung nicht mehr möglich ist. Ein weiteres besonderes Merkmal ist die doppelte Belegung der beiden Funktionstasten. Die linke Funktionstaste führt zu Multimediainhalten, die Rechte zu Telefon und Navigation. Abbildung 3-27 zeigt das Bedienteil in der Mittelkonsole des Fahrzeugs. Zentrale in der Mitte befindet sich der Dreh-Drück-Steller, oberhalb links und rechts der Tasten für den Warnblinker und die Lendenstütze des Sitzes die Funktionstasten und die Zurücktaste und unterhalb des Dreh-Drück-Steller die Tastatur.

Abbildung 3-27 Bedienteil Mercedes Benz COMMAND
Quelle: (Daimler AG 2010b)

3.5.1.3.2 Funktionsumfang

COMMAND besitzt kein zentrales Hauptmenü, es können dennoch alle Funktionen ohne das Betätigen einer Funktionstaste aufgerufen werden. Die Anzeige von Command ähnelt sehr stark einer Desktop-Anwendung und besitzt ein Menü am oberen Bildschirmrand. Innerhalb dieses Menüs kann jede Funktion von Command aufgerufen werden. Die vier Funktionen

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Disk, Radio, Telefon und Navigation können sowohl über dieses Menü als auch über die beiden Funktionstasten am Bedienteil aufgerufen werden. Die Funktionsbereiche Disk und Radio sowie Telefon und Navigation teilen sich jeweils einen Funktionstaste. Neben den vier genannten Funktionen gibt es noch eine Fahrzeugfunktion, in der Systemeinstellungen, wie beispielsweise die Nachbeleuchtungszeit, eingestellt werden können. Die Funktion Fahrzeugdienste ist nur über das Menü zuggreifbar.

3.5.1.3.3 Bedienkonzept

Das Bedienkonzept von COMMAND folgt wie auch die anderen beiden betrachteten Systeme einer hierarchischen Struktur. Allerdings verfügt COMMAND über kein übergeordnetes Hauptmenü, so dass der initial angezeigte Bildschirm beim Einschalten des Systems die Radiofunktion ist. Wird eine andere Funktion über das Menü am oberen Bildschirmrand ausgewählt, wechselt die Anzeige auf den Startbildschirm der jeweiligen Funktion. Analog hierzu ist auch die Funktionslogik der beiden Funktionstasten. Durch die Doppelbelegung der Tasten werden die darunter liegenden Funktionen jedoch nur sequenziell aufgerufen. Das bedeutet, dass bei erstmaligem Betätigen der Disk/Radio Taste die Disk Funktion aufgerufen wird. Ein erneutes betätigen derselben Taste wechselt von der Disk Funktion in die Radio Funktion. Mithilfe der Funktionstasten können also jeweils die beiden Funktionen durchgeschaltet werden, wobei das Hauptmenü der jeweiligen Funktion aktiviert wird.

Die Zurück-Taste von COMMAND ist analog der Return-Taste von Audi hierarchisch angelegt. Befindet sich der Nutzer also in einem Untermenü, beispielsweise der Radiofunktion, so führt ihn ein Betätigen der Zurück-Taste in die nächst höhere Ebene dieser Funktion. Da COMMAND über kein Hauptmenü verfügt, endet dieses hierarchische Verhalten mit dem Erreichen des Hauptmenüs der jeweiligen Funktion.

Wie in Abschnitt 3.5.1.3.4 genauer dargelegt, ist die Anzeige von Command in verschiedene Bereiche eingeteilt. Durch ein Drücken des Dreh-Drück-Stellers in die jeweilige Richtung kann zwischen den verschiedenen Bereichen der Anzeige navigiert werden. Innerhalb eines Bereichs werden einzelne Elemente durch Drehen des Dreh-Drück-Stellers ausgewählt und durch Drücken des Dreh-Drück-Stellers aktiviert. Eine Drehung nach links verschiebt dabei den Fokus ein Element nach unten, eine Drehung nach rechts entsprechend nach oben. Tabelle 3-20 fasst das Verhalten der einzelnen Elemente des Bedienteils von COMMAND zusammen.

Aktion	Verhalten
Funktionstaste	*Erstmalige Betätigung:* Anzeige des Hauptmenüs der ersten Funktion hinter der Funktionstaste *Erneute Betätigung:* Anzeige der zweiten Funktion hinter der Funktionstaste *Fokus: Fokus des ersten Objekts*
Zurück-Taste	Hierarchisch
PTB	*Links-Drehung:* Fokus nach oben *Rechts-Drehung:* Fokus nach unten

Tabelle 3-20 Bedienkonzept Mercedes Benz COMMAND
Quelle: (Forrai 2010, 56)

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

3.5.1.3.4 Graphische Gestaltungselemente

Die Aufteilung der Anzeige ist in COMMAND in alle Funktionsbereichen identisch umgesetzt und gliedert sich in fünf horizontal ausgerichtete Bereiche. Am oberen Bildschirmrand befindet sich eine Statusleiste die unter anderem Informationen zur Signalstärke der Mobilfunkverbindung anzeigen. Darunter befindet sich das Menü, welches die Funktionsbereiche von Command umfasst. Der nächste Bereich ist der zentrale Bereich der Anzeige dessen Inhalt im Folgenden genauer dargestellt wird. Unterhalb des Hauptbereichs erfolgt analog der Menüzeile ein Bereich für Untermenüs, welcher Funktionen abhängig von der gewählten Funktion darstellt. Den Abschluss am unteren Ende des Bildschirms bildet eine zweite Statusleiste, in der unter anderem Informationen zur Klimaanlage angezeigt werden.

Aufgrund der desktopähnlichen Struktur ist das Menü analog der Menüleiste an einem Computer über alle Funktionsbereiche hinweg vorhanden und kann somit nicht mit den Hauptmenüs von Audi und BMW verglichen werden. Diesen Charakter unterstreicht auch die horizontale Anordnung der Menüeinträge. Zwischen den einzelnen Bereichen des Bildschirms kann durch Drücken des Dreh-Drück-Stellers nach oben oder unten gewechselt werden. Um vom Hauptmenü in das Untermenü zu gelangen, muss diese also zweimal nach unten gedrückt werden um zunächst in den Hauptbereich und dann in das Untermenü zu gelangen. Abbildung 3-28 gibt einen Überblick über die Struktur der Anzeige von COMMAND.

Abbildung 3-28 Aufbau Mercedes Benz COMMAND
Quelle: (Forrai 2010, 57)

Zwischen den einzelnen Funktionsbereichen wird in COMMAND, im Gegensatz zu den anderen beiden Systemen, nicht farblich unterschieden. Sämtliche Menüs und Untermenüs sind in denselben beigen Farbtönen gehalten und der Hauptbereich hat eine schwarze Hintergrundfarbe.

Analog zu den Hauptbereichen bei BMW und Audi besteht auch in COMMAND die Möglichkeit, mehr Informationen im Hauptbereich einer Anzeige darzustellen als dort auf einmal hinein passen. Ein grafisch stark an PC-Systeme angelehnter Scrollbalken auf der rechten Seite des Bildschirms dient dem Nutzer zur Orientierung. Auch kann COMMAND die Nutzer über den Fortschritt bei langwierigen Operationen unterrichten. Hierzu wird in Form eines Popupfensters eine Tortengrafik angezeigt, die sich zunehmend schwarz einfärbt und so den prozentualen Fortschritt angezeigt.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Eine Selektion von Elementen wird in COMMAND durch einen roten Unterstrich sowie einen Wechsel des Hintergrunds des selektierten Elements in eine hellbeige Farbe angezeigt. Befindet sich ein fokussiertes Objekt innerhalb eines aufgeklappt Kastens, so verfärbt sich der Hintergrund des Kastens ebenfalls hellbeige, um so den grafischen Fokus auf sich zu ziehen.

Anders als im Multi-Media-Interface und iDrive bilden Listen nicht das zentrale Gestaltungselemente in COMMAND. Informationen werden zunächst mithilfe einer Grafik visualisiert und Listen nur bei Bedarf zur Auswahl einzelner Werte genutzt. Hierzu stehen COMMAND einfache Listen, komplexe Listen, bestehend aus Grafik und Text, sowie Checkboxen zur Verfügung. Listen werden dabei in Form von Pop-Ups an der entsprechenden Stelle über den Bildschirm eingeblendet (vgl. Abbildung 3-29).

Abbildung 3-29 Darstellung von Listen (MB COMMAND)
Quelle: (Forrai 2010, 59)

Texteingabe erfolgt in COMMAND nicht mit einem runden Speller, sondern werden Mithilfe eines vertikal angeordneten Alphabets am unteren Bildschirmrand dargestellt. Mithilfe des Dreh-Drück-Stellers kann zwischen einzelnen Elementen des Alphabets durch Drehen gewechselt und diese durch Drücken ausgewählt werden. Eine Texteingabe kann dabei nicht nur ein Textfeld umfassen, sondern mehrere, die Mithilfe eines nach oben und nach unten Drückens des Dreh-Drück-Stellers ausgewählt werden können.

Abbildung 3-30 Darstellung Picke (MB COMMAND)
Quelle: (Forrai 2010, 60)

Zur Eingabe von speziell formatiert Zahlen verfügt das System von Mercedes Benz ebenfalls über einen Picker. Dieser wird in einem Pop-Up Fenster angezeigt und stellt die entsprechenden Informationen zusätzlich in Form einer Grafik dar. Wie in Abbildung 3-30 zu sehen, wird die einzustellende Uhrzeit zusätzlich durch eine analoge Uhr dargestellt.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Wie auch bei anderen Elementen werden Slider in einem Popupfenster dargestellt. Aktiviert der Nutzer einen entsprechenden Eintrag im Hauptbereich öffnet sich ein Pop-Up Fenster in dem der Slider durch Drehen des Dreh-Drück-Stellers geändert werden kann. Durch Drücken des Dreh-Drück-Stellers wird das Pop-Up Fenster geschlossen und die Anzeige im Hauptbereich aktualisiert. Eine weitere Möglichkeit Werte auf einer Skala einzustellen ist ein Fadenkreuz. Analog zum System von Audi stellt COMMAND ein Fadenkreuz über einer Grafik dar, das durch Drücken des Dreh-Drück-Stellers in die verschiedenen Himmelsrichtungen positioniert werden kann. Ein Beispiel für solches Fadenkreuz ist die Einstellung von Balance und Fader im Soundsystem. (vgl. Abbildung 3-31)

Abbildung 3-31 Auswahl von Werten (MB COMMAND)
(links: Slider; rechts: Fadenkreuz)
Quelle: (Forrai 2010, 61)

Schließlich bietet COMMAND auch die Möglichkeit Dialoge anzuzeigen. Auch diese werden in Form eines Pop-Up Fensters dargestellt. Dialoge beinhalten einerseits anzuzeigende Informationen, bestehend aus Texten und Grafiken, sowie horizontal angeordnete Buttons, die der Nutzer durch Drehen des Dreh-Drück-Stellers auswählen und so durch Drücken desselben eine Antwort geben kann.

3.5.2 Diskussion

Vergleicht man die drei Systemen von BMW, Audi und Mercedes Benz stellt man fest, dass sich diese im Laufe der letzten Jahre sehr stark einander angenähert haben. Alle drei Systeme nutzen zur Darstellung ihrer Inhalte ein zentral in das Armaturenbrett integriertes Display. Darüber hinaus wird vermehrt das Fahrerinformationssystem, welches sich im Kombigerät befindet, zur Anzeige ausgewählte Informationen genutzt. So wird dort zum Beispiel der aktuelle Radiosender oder Richtungspfeile der Navigation angezeigt.

Für die Interaktion des Nutzers mit dem System setzen alle drei Hersteller auf einen Dreh-Drück-Steller als zentrales Steuerelement. Funktional kann der Nutzer diesen drehen, drücken und in die vier Himmelsrichtungen schieben. Neben dem Dreh-Drück-Steller gibt es bei allen Herstellern noch eine Reihe an zusätzliche Funktionstasten. Diese werden entweder genutzt einzelne Funktionsbereiche das Infotainmentsystemen direkt zu erreichen oder um allgemeine Aktionen wie das Zurückspringen oder den Aufrufen des Hauptmenüs zu aktivieren.

Darüber hinaus bieten Infotainmentsysteme einen Zugang zur Bordelektronik des Fahrzeugs. Über die am CAN[15]-Bus vorliegenden Informationen können so fahrzeugspezifische Daten eingesehen und in bedingtem Maße manipuliert werden. Auch ist ein Zugriff auf die, für die

[15] Netzwerk zur Übermittlung von Daten zwischen einzelnen Steuergeräten im Fahrzeug.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Navigation erforderliche, GPS-Positionierung und die Mobilfunkkomponente der Fahrzeuge möglich. So können Telefonfunktionen sowie ein mobiler Internetzugriff in allen Infotainment-Lösungen genutzt werden.

Betrachtet man die Systeme hinsichtlich des Funktionsumfangs, bieten sie dem Nutzer im Wesentlichen die gleichen Funktionalitäten an. Zur Grundausstattung der Systeme gehören Radio (digital und analog), CD, Telefon, Adressbuch, Navigation und Staumeldungen. Natürlich ist das Vorhandensein der einzelnen Funktionen abhängig von der jeweiligen Modellreihe und kann je nach den vom Kunden bestellten Ausstattungsmerkmalen variieren. Des Weiteren haben die Systeme integrierte Onlineangebote, die meist von den Herstellern optimiert zur Verfügung gestellt werden. Einzig iDrive von BMW bietet ein freies Browser an.

Auch hinsichtlich ihrer Bedienkonzepte weisen die Systeme einige Gemeinsamkeiten auf. So sind alle drei Lösungen hierarchisch aufgebaut und beginnen mit einem Hauptmenü. Eine Ausnahme bildet hier COMMAND, welches das Hauptmenü in die Funktionsbereiche integriert und somit nicht als allein stehendes Menü anbietet. Einzelne Funktionsbereiche können entweder über das Menü oder über die speziellen Funktionstasten direkt angesprochen werden und sind nahezu vollständig über den Dreh-Drück-Steller steuerbar. In der Bedeutung der Return-Taste zeichnen sich zwei unterschiedliche Wege ab. Audi und Mercedes Benz verfolgen hier einen hierarchischen Ansatz, BMW einen historischen.

Der Aufbau der Anzeigen erfolgt jeweils anhand eines vorgegebenen Schemas. Der Bildschirm wird in Bereiche unterteilt die ihre Bedeutung, mit nur wenigen Ausnahmen, über alle Funktionsbereiche hinweg behalten. Zusätzlich werden Farbschemata eingesetzt um unterschiedliche Funktionsbereiche für den Nutzer visuell zu unterscheiden (mit Ausnahme von COMMAND). Zentrales Element dieses Schemas ist stets ein Hauptbereich, der als Container für die funktionsbezogenen Inhalte genutzt wird und ein einheitliches Scrollen für mehrseitige Anzeigen bereitstellt. Besonders dabei ist, dass es über alle Funktionsbereiche hinweg eine begrenzte Anzahl an Elementen gibt die im Hauptbereich angezeigt werden können. Zusätzlich können Informationen auch in Form von Pop-Up Dialogen über dem Schema eingeblendet werden.

Das wesentliche Bedienkonzept, welches man im Hauptbereich vorfindet, sind Listen. Diese können unterschiedlichste Informationen sowohl textuell als auch grafisch darstellen und ermöglichen dem Nutzer eine Auswahl einzelner oder mehrerer Elemente. Je nach Hersteller variieren Listen von einfachen, eindimensionalen Listen hin zu mehrdimensionalen Listen. Für die Eingabe von Text wird entweder ein Speller oder ein Alphabet genutzt. Abgesehen von der grafisch unterschiedlichen Darstellung sind beide Ansätze analog über den Dreh-Drück-Steller nutzbar, jedoch ermöglichen sie aufgrund ihrer langsamen Bedienung nur eingeschränkte Texteingabe. Um diesem Problem entgegenzuwirken beginnen die Hersteller Touchpads (Audi) oder Tastaturen (Mercedes Benz) zu nutzen.

Für die direkte Interaktion mit dem Nutzer stehen Dialoge zur Verfügung. Dialoge bestehen aus einem Bereich für die Informationsanzeige (Text und Grafik) sowie einer Liste an Antwortmöglichkeiten die der Nutzer über den Dreh-Drück-Steller auswählen kann. Weitere verfügbare Elemente sind ein Browser, eine Kartendarstellung sowie Slider, Fadenkreuz und Picker für die Eingabe numerischer Werte.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

3.6 Fazit und Implikationen

Wie die vorangegangen Abschnitte zeigen, ist die Gestaltung erfolgreicher Automotive Services von einer Vielzahl an Herausforderungen aus unterschiedlichen Bereichen geprägt. Neben der Auswahl und Beachtung der verschiedenen Fähigkeiten und Kompetenzen der relevanten und notwendigen Stakeholder, sind Erkenntnisse die die Softwareentwicklung aus dem Bereich des Requirements Engineering besitzt sowie grundlegende Herausforderungen wie eine Gestaltung der Mensch-Maschine-Interaktion im Fahrzeug effizient umgesetzt werden kann, entscheidend. Darüber hinaus gibt es eine Reihe an praktischen Erfahrungen sowohl was den Umgang mit Stakeholder, den Anforderungen als auch die Gestaltung von Infotainmentsystemen betrifft.

Stakeholder in dieser Domäne lassen sich in vier grundsätzliche Kompetenzprofile unterteilen: Automotive Service Experten, Experten der Anwendungsdomäne, Experten der Umsetzungsdomänen sowie sonstige Stakeholder. Wie auch die Teilnehmer der Vorstudie bestätigen, ist die Gruppe der sonstigen Stakeholder maßgeblich für Anforderungen an einen Automotive Service verantwortlich. Zwar haben diese weder das technische Know-How Services umzusetzen noch kennen sie sich ausreichend in der Automotive Domäne aus. Trotzdem sind sie als die späteren Kunden und Nutzer von Automotive Services von entscheidender Bedeutung. Die wesentliche Frage die sich hier stellt ist, wie diese Stakeholder in die Lage versetzt werden können ihre Vorstellungen und Anforderungen nutzenstiftend zu explizieren und so den anderen drei Gruppen zur Verfügung zu stellen. Analog stellt sich natürlich auch die Frage, wie die Experten der Anwendungsdomäne als auch die Experten der Umsetzungsdomänen ihr Wissen geeignet zur Verfügung stellen können um die anderen Stakeholdergruppen in die Lage zu versetzen ein vollständiges Bild eines Automotive Service zu zeichnen.

Die größten Probleme im Requirements Engineering sind die Beschreibung der Zielvorstellungen der Stakeholder sowie deren Verständnis der vorliegenden Problemstellung. Unklare Zielvorstellungen führen zur Umsetzung der falschen Lösung und verhindern so ein erfolgreiches Resultat. Ein weiteres Problem ist die hohe Komplexität die hinter einem Automotive Service steckt. Diese Komplexität, sowohl auf der Umsetzungs- als auch auf der Anwendungsseite, erschwert es viele Stakeholder, Anforderung in geeigneter Art und Weise zu verstehen oder selbst beizusteuern. Ein ähnliches Problem bilden Sprachbarrieren. Diese können einerseits fachlicher Natur sein andererseits aber auch schlicht daher kommen, dass Stakeholder unterschiedlichen Kulturen und Sprachräumen entstammen. Auch hier ist ein geeignetes Medium der Kommunikation unverzichtbar. Des Weiteren sind Anforderungen im Verlauf eines Projektes stark veränderlich. So ist es nur schwer möglich diese in Form eines sequenziellen Vorgehens von der Erhebung bis zur Umsetzung und Betrieb zu verfolgen. Vielmehr scheint ein Einsatz eines geeigneten, iterativen Vorgehens, welches Stakeholder in geeigneter Art und Weise integriert, vonnöten. Ein weiteres Problem ist die oftmals schlechte Qualität der dokumentierten Anforderungen. Aufgrund der hohen Komplexität von Automotive Services und der starken inhaltlichen Verschränkung der einzelnen Anforderungen untereinander, ist eine geeignete Dokumentation unverzichtbar. Neben der Qualität von Anforderungen ist die Frage nach den richtigen Anforderungen, und damit den richtigen Merkmalen des entstehenden Service entscheidend. Hier können Prototypen einen wesentlichen Beitrag leisten, indem sie allen Beteiligten frühzeitig die Konsequenzen und Möglichkeiten einzelner Anforderungen aufzeigen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Wie bereits angesprochen sind Kunden und Endanwender, also Vertreter der Gruppe der sonstigen Stakeholder, die größte Quelle der initialen Anforderungen. Erst im späteren Projektverlauf, wenn Anforderungen aufgrund unterschiedlichster Gründe geändert oder erweitert werden, kommen interne Experten, also Vertreter der Experten der Anwendungsdomäne, zum Tragen. Anforderungen werden in der Praxis vorwiegend im Rahmen von Workshops und Interviews erhoben. Es zeigt sich, dass der Einsatz von Modellen hier ein geeignetes Medium zur Kommunikation und Interaktion darstellt. Dies unterstreicht auch die Beobachtung, dass in der Regel unterschiedliche Fachdomäne an der Entwicklung und Gestaltung beteiligt sind. Ein Modell stellt hier eine gemeinsame Sprache zur Kommunikation und Kooperation dar. Durch den Einsatz von Modellen in Kombination mit Prototypen kann so die Kommunikation entscheidend verbessert werden und eins der wesentlichen Probleme, die Explikation von Anforderungen sowie die Sicherstellung der Qualität sichergestellt werden.

Hinsichtlich der Gestaltung der Interaktion der Nutzer mit dem Automotive Service weist die Forschung der Mensch-Maschine-Interaktion zwei wesentliche Aspekte auf. So werden auf der einen Seite Systeme immer komplexer und erfordern somit eine stärkere Integration der Stakeholder. Dies hat natürlich auch zum Effekt, dass es einer Vielzahl dieser zunehmend schwer fällt, System hinreichend zu verstehen. Auf der anderen Seite stellt der Mensch selber, als Nutzer der Systeme, einen entscheidenden Faktor bei deren Gestaltung dar. Bei der Nutzung und somit die Interaktion der Systeme müssen die kognitiven Eigenarten des Menschen bei der Gestaltung dieser Berücksichtigung finden. Es zeigt sich, dass hier ein integrativer Ansatz sowie ein möglichst früher Einsatz von Prototypen zielführend ist. Problematisch in diesem Zusammenhang ist jedoch, dass weder Stakeholder, mit ihren fachspezifischen Kompetenzen, noch Programmierer und Entwickler über die notwendigen Fähigkeiten und Voraussetzungen für ein Design dieser Systeme verfügen (Norman 1996, 208). Dieses Wissen über das Gestalten von und Interaktion mit Automotive Services muss also in geeignete Art und Weise verfügbar gemacht werden.

Schließlich gibt es eine ganze Reihe an Erkenntnissen Errungenschaften die aus dem Bereich der Infotainmentsysteme in den letzten Jahren gewonnen werden konnten. Da diese Systeme sowohl die technische als auch die konzeptionelle Grundlage für den Einsatz von Automotive Services im Fahrzeug darstellen, stellen deren Besonderheiten wesentliche Herausforderungen dar. Für die Interaktion mit den Nutzern stehen in der Regel ein Display sowie ein Bedienteil mit einem Dreh-Drück-Steller als zentralem Steuerelement zur Verfügung. Zusätzlich erlauben diese Systeme den Zugriff auf Boardsysteme, welche eine enge Integration von Automotive Services in das Fahrzeug ermöglichen. Hinsichtlich der grafischen Gestaltung und Struktur aktueller Infotainmentsysteme zeigt sich, dass diese hierarchisch aufgebaut sind und über feste Bildschirmschemata verfügen. Besonders zu bemerken ist hierbei, dass unabhängig der Funktion, die ein Nutzer in einem Infotainmentsystemen gerade verwendet nur eine Hand voll Bedienkonzepte wie Listen oder Speller zum Einsatz kommen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

4 Gestaltung von Automotive Services durch den Einsatz von Modellen

Ein viel versprechender Ansatz mit den Herausforderungen und Eigenarten der Gestaltung von Automotive Services umzugehen ist der Einsatz von Modellen. In vielen Bereichen in denen komplexe Sachverhalte einer Vielzahl unterschiedlicher Stakeholder, die unterschiedliche Kompetenzen mitbringen, verdeutlicht werden müssen, kommen Modelle zum Einsatz. Beispiele hierfür sind die Gestaltung von Prozessabläufen die oftmals mithilfe von ARIS-Modellen oder BPMN-Modellen modelliert werden (Krcmar 2010a, 143). Auch bei der Gestaltung von Softwaresystemen sind Modelle inzwischen unverzichtbar und werden einerseits zur Beschreibung von Anforderungen und andererseits zur Darstellung von Umsetzungsdetails genutzt (Brügge/Dutoit 2004, 24). Auf diese Weise kann sowohl die Kommunikation als auch die Dokumentation wesentlicher Aspekte von Automotive Services sichergestellt werden.

Allerdings ist ein Werkzeug alleine nicht in der Lage diese Herausforderungen zu meistern. Vielmehr sind ein effizienter Einsatz sowie ein geeignetes Vorgehen in Kombination mit einem dazu passenden Werkzeug ein Erfolg versprechender Ansatz. Die Methode des Design Thinking ist ein solcher Ansatz, Automotive Services in enger Integration mit den Stakeholdern zu gestalten und so einen „systematic approach to innovation“ (Brown 2009, 157) zu beschreiten. Die Methode folgt einem iterativen Vorgehensmodell und unterstützt dabei die kreative Gestaltung neuer Ideen. Der folgende Abschnitt befasst sich mit den Grundlagen der Modellierung sowie den Eigenarten des Design Thinking, um auf dieser Basis eine Modellierung von Automotive Services zu entwickeln.

Aus beiden Betrachtungen werden Anforderungen an eine Modellierung von Automotive Services sowie einer geeigneten Werkzeugunterstützung erarbeitet. Neben dieser rein theoretischen Betrachtung der Problemstellung werden schließlich in einem dritten Abschnitt die Anforderungen der Praxis an eine solche Modellierung erhoben. Hierzu wurden Stakeholder im Rahmen eines Workshops sowie einer darauf aufbauenden Ideenplattform in den Anforderungsprozess integriert.

4.1 Modellierung von Automotive Services

Modelle werden eingesetzt, komplexe und schwierige Sachverhalte in einer einfachen und klar definierten Art und Weise zu repräsentieren. Eine Modellierung von Automotive Services versetzt Stakeholder in die Lage, neue und innovative Ideen auf der Grundlage bereits etablierte Modellierungskonzepte zu explizieren ohne dabei Ressourcen in die Konzeption und Entwicklung technischer Aspekte investieren zu müssen. Insbesondere ein speziell auf die Domäne der Automotive Services angepasstes Modellieren kann Stakeholder mit unterschiedlichen Kompetenzen bei ihrer gestalterischen Arbeit maßgeblich unterstützen.

4.1.1 Begriffliche Klärung

Der Begriff des Modells leitet sich vom lateinischen Wort *modus* (Maß, Art, Weise) bzw. *modulus* (Maßstab) ab. Im alltäglichen Sprachgebrauch offenbart sich eine große Bedeutungsvariabilität. Je nach Kontext finden sich in Lexika synonyme Verwendungen der Begriffe Muster, Vorbild, Entwurf und Abbild. Eine einheitliche Definition erfolgt auch in der Fachliteratur nur abstrakt und anhand von Aspekten, so beispielsweise die in der allgemeinen

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

(und damit domänenunabhängigen) Modelltheorie von Stachowiak (1974) entwickelte Definition mittels folgender drei Charakteristika:

- *Abbildungsmerkmal:* Ein Modell ist ein Abbild bzw. eine Repräsentation eines Objekts oder Sachverhalts von Interesse. Dieses Objekt kann real oder gedacht sein.
- *Verkürzungsmerkmal:* Ein Modell erfasst nicht alle Facetten, Zusammenhänge und Attribute seines Vorbilds, es abstrahiert und vereinfacht.
- *Pragmatisches Merkmal:* Jedes Modell dient einem Zweck und orientiert sich an einem Nutzen. Modelle werden entworfen, um Probleme zu lösen.

Abbildung 4-1 vereint die genannten Aspekte mit der Darstellung der Zusammenhänge eines Modells mit seiner Umwelt und seinem Kontext.

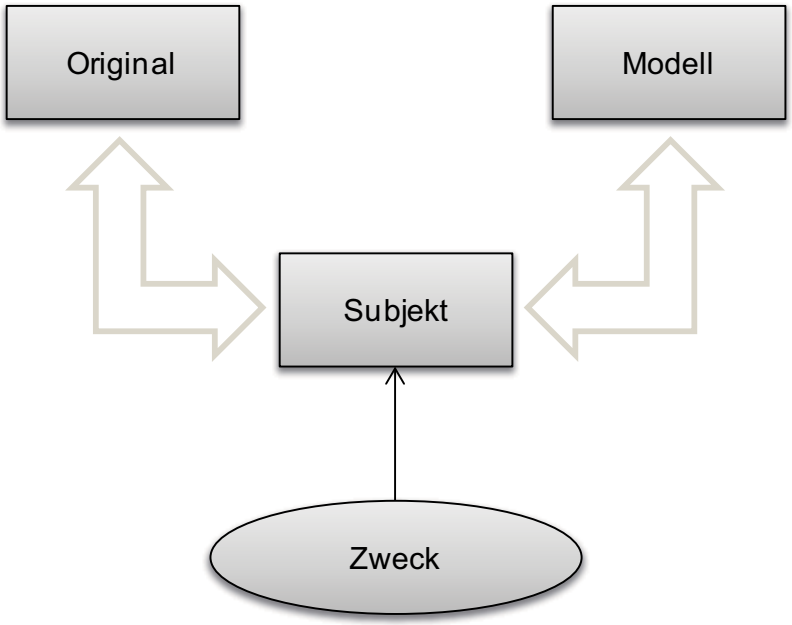


Abbildung 4-1 Modell und Kontext
Quelle: (Thomas 2001)

Die verschiedenen wissenschaftlichen oder alltagsgebräuchlichen Ausprägungen des Modellbegriffs lassen sich anhand dieser Abbildung leicht aufzeigen: Je nach Kontext variieren ein oder mehrere Komponenten. Zweck und Original bestimmen Form und Granularität des Modells. Original und Modell können dabei konkret wie ein Gebäude und sein Architekturmodell sein oder abstrakt wie die zahlenmäßige Entwicklung einer Population und das entsprechende mathematische Modell (Kastens/Büning 2008). Der Charakter des einen gibt jedoch nicht zwangsweise den des anderen vor.

Zwischen Original und Modell besteht eine selektiv merkmalerhaltende, abstrahierende Abbildungsrelation. Unter Einfluss einer Zweckorientierung wird diese Relation vom Subjekt festgelegt. Dieser Vorgang wird Modellierung genannt und erfolgt anhand der Modellierungstechnik. Braun (2007, 16) konkretisiert: „Die Modellierungstechnik schreibt vor, wie ein Modell zu konstruieren ist. Sie besteht aus einer Modellierungssprache und einer Vorgehensweise […]. Von der Vorgehensweise (Vorgehen im Kleinen) als Vorschrift zur Erstellung eines Modells ist die Sprache zu unterscheiden, mit der das Modell beschrieben wird. Die Sprache legt die Elemente fest, die bei der Erstellung eines Modells verwendet werden können“ (vgl. Abbildung 4-2).

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

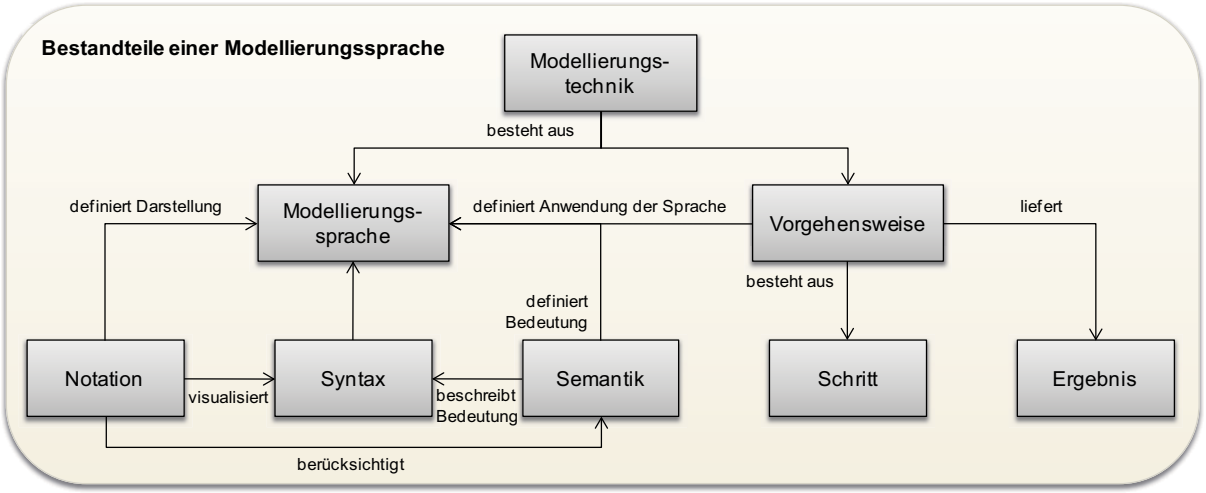


Abbildung 4-2 Die Bestandteile einer Modellierungssprache
Quelle: (Braun 2007, 16)

Strukturell und semantisch betrachtet lassen sich bei der Abbildung der Attribute von Original auf Modell vier Fälle unterschieden (Stachowiak 1974):

- *Präterion:* Als unwichtig betrachtete Attribute des Originals werden nicht in das Model übernommen.
- *Abundanz:* Dem Modell werden Attribute hinzugefügt, die sich nicht auf das Original abbilden lassen. Dies kann beispielsweise zum Zweck der Komplexitätsreduktion erfolgen.
- *Kontrastierung:* Attribute werden gezielt hervorgehoben.
- *Transkodierung:* Attribute werden mit einer anderen Bedeutung belegt.

4.1.2 Der Modellbegriff in der Wirtschaftsinformatik

Schermann (2009, 16) stellt die nach Greiffenberg (2004, 13) in der Wirtschaftsinformatik vorwiegend verwendeten Modellbegriffe der abbildungsorientierten Definition und der konstruktionsorientierten Definition dem allgemeinen Modellbegriff nach Stachowiak (1974) gegenüber.

Der abbildungsorientierte Modellbegriff zielt auf Eindeutigkeit in der Abbildung ab und unterscheidet sich dadurch maßgeblich vom allgemeinen Modellbegriff. Die Beziehung zwischen Attributen von Modell und Original ist damit durch den Wegfall von Abundanz in der Regel ein strukturerhaltender Homomorphismus (Strahringer 1996). Wird auf eine Vereinfachung des Originals verzichtet, ergibt sich der Sonderfall einer isomorphen Beziehung. Schermann führt auch das resultierende Problem der Notwendigkeit omniszienter Kenntnis von Ursache-Wirkungsbeziehungen bei deren homomorpher Modellierung an (Schermann 2009, 17).

Der zweite relevante Modellbegriff rückt den namensgebenden Konstruktionsprozess in den Fokus des Interesses (Schütte 1998). Das Original wird vom Modellersteller für die Zwecke des Modellnutzers rekonstruiert, der Konstruktionsprozess bestimmt so den Charakter der Abbildung des Originals durch das Modell (Schermann 2009, 16). Abbildung 4-3 illustriert diesen Hergang: Ausgangspunkt ist sowohl beim Modellersteller als auch beim Modellnutzer zunächst das interne Modell als Rekonstruktion des Originals. Die Explikation des internen Modells des Erstellers unter Mitwirkung des Nutzers formt schlussendlich eine externe Modellmanifestierung im alleinigen Sinne des Nutzers. In diesem Kontext unterstreicht das

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Prädikat konzeptionell den durch das Problemempfinden des Modellnutzers definierten Zweck des Modells zur Analyse und Lösungsfindung. Darüber hinaus listet Strahringer (1996) weitere Funktionen konzeptioneller Modelle auf: Beschreibung, Erklärung, Entscheidung oder Ermittlung von Entscheidungsgrundlagen.

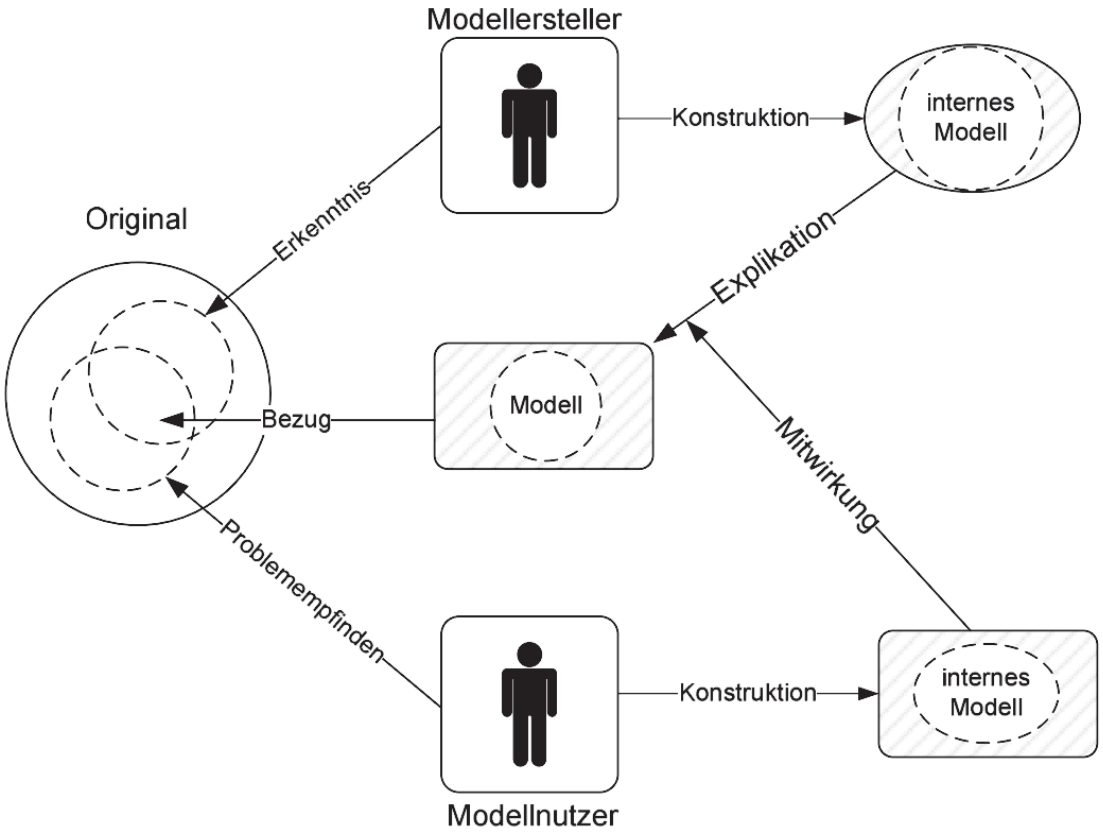


Abbildung 4-3 Modellentwicklung nach dem konstruktionsorientierten Modellbegriff
Quelle: (Schermann 2009, 17)

4.1.3 Begriffliche Ausprägungen und Spezialformen eines Modells

Ein Modell, das seinerseits aus mehreren Einzelmodellen besteht, wird als komposites Modell bezeichnet (Pitschke 2009). Komposite Modelle profitieren daher von den Erkenntnissen, die in deren Teilmodellen bereits expliziert wurden. Ein Modell mit normativem Charakter, das also als Entwurfsmuster oder Vergleichsobjekt herangezogen werden kann, gilt in der Literatur als Referenzmodell. Dem gegenüber steht das Applikationsmodell als referenzierendes Modell (Fettke/Loos 2003).

Falls das Original eines Modells selbst ein Model ist, so spricht man von Metamodellierung. Das aus diesem Prozess resultierende Modell wird folglich Metamodell genannt (Strahringer 1996). In der Praxis dienen Metamodelle dazu, Grundlagen für kontextidentische Modellierung zu schaffen. Sie definieren uniforme Techniken um sicherzustellen, dass Originale unterschiedlicher Ausprägung mit den gleichen Komponenten und dem gleichem Abstraktionsgrad modelliert werden. Da entsprechend der Kontextidentität auch der Zweck unverändert bleibt, sind Modelle die auf Metamodellen basieren stets hinsichtlich wichtiger Merkmale, die in die Definition des Metamodells einfließen, vergleichbar.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

4.1.4 Qualität konzeptioneller Modelle

Nicht jedes Modell kann jedoch per se den Anspruch erheben das Nutzenpotenzial im Sinne der in Abschnitt 4.1.2 genannten Punkte voll zu entfalten. Vielmehr hängt die Qualität eines Modells von mehreren Faktoren ab und muss individuell für diese beurteilt werden. Schermann (2009, 24) differenziert fünf Dimensionen des Qualitätsbegriffs nach Krogstie (1998):

- Die *physische Qualität* des Modells nimmt den Wissensstand von Modellersteller und Modellnutzer zum Gegenstand der Betrachtung. Zum einen wird untersucht inwieweit es dem Ersteller möglich ist sein Wissen zu externalisieren, zum anderen muss sichergestellt sein, dass eine folgelogische Internalisierung durch den Nutzer erfolgen kann. Kriterium der Qualität in dieser Dimension ist dabei nur die Deckungsgleichheit von vorhandenem Wissen der Akteure und der Aussagen durch das Modell.
- Die Übereinstimmung der Strukturen und Aussagen zwischen Modell und Original (Domäne) und damit die Validität des Modells ist Gegenstand der *semantischen Qualitätsbewertung*. Darüber hinaus wird die Vollständigkeit des Modells beurteilt. Anzumerken ist, dass semantische Qualität, wie in Abbildung 4-4 dargestellt, nur indirekt bewertet werden kann.
- Bei der Beurteilung der *syntaktischen Qualität* werden Aussagen im Modell auf Deckung mit den jeweiligen Regeln der verwendeten Modellierungssprache untersucht.
- Unter *sozialer Qualität* versteht man den Grad der Übereinkunft der Beteiligten über den Modellinhalt und die daraus zu ziehenden Schlussfolgerungen.
- Das entscheidende Kriterium der *pragmatischen Qualität* ist die vollständige und korrekte Interpretation des Modells durch den Nutzer.

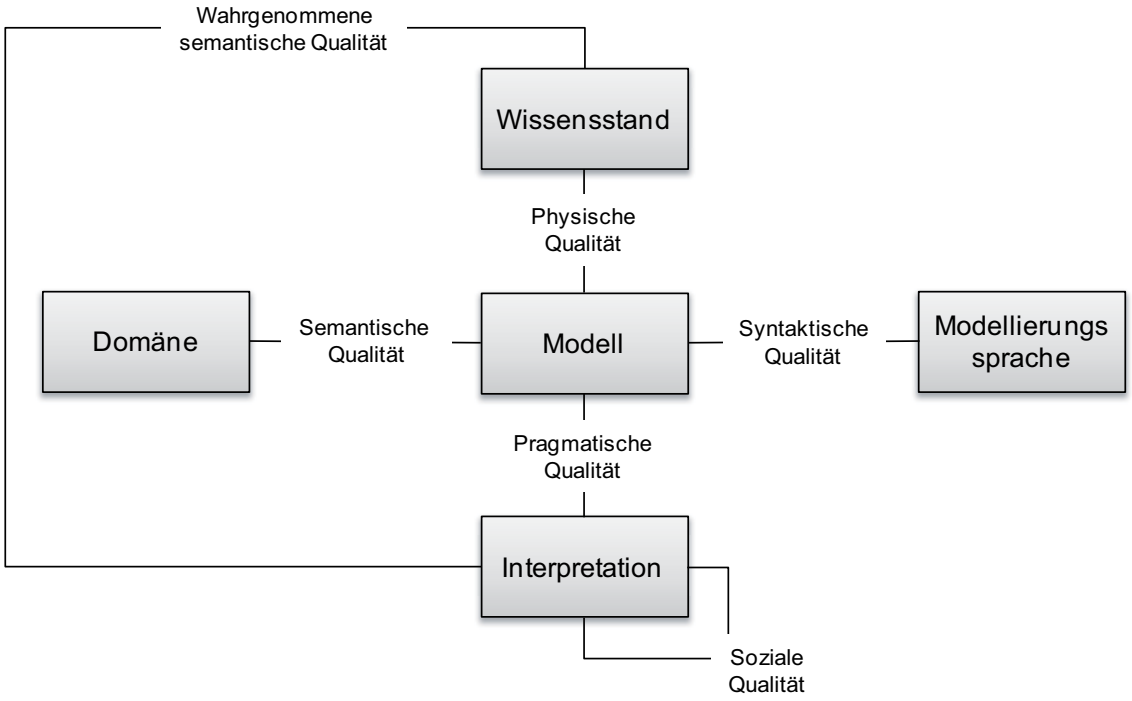


Abbildung 4-4 Einflussfaktoren auf die Qualität konzeptioneller Modelle
Quelle: (Schermann 2009, 24)

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

4.1.5 Modelle in der Softwareentwicklung

„Models are abstractions that portray the essentials of complex problems or structures by filtering out non-essential details, thus making the problem easier to understand. […] We build models of complex systems because we cannot comprehend such systems in their entirety. There are limits to human capacity to understand complexity. Models help us organize, visualize, understand and create complex things“ (Quatrani 1997, 13).

Durch den gezielten Einsatz von Modellen ergeben sich in jeder Anwendungsdomäne eklatante Vorteile: Die Verständlichkeit des Problems und der potenziellen Lösung wird durch die Abstraktion von irrelevanten Aspekten erhöht, der Komplexität größerer Projekte wird entgegengewirkt. Darüber hinaus können Modelle den Grundbaustein für Automatisierung und damit Produktivitätssteigerung liefern.

In der Softwaretechnik führte die verstärkte Orientierung auf allgemeine Metamodelle wie die Unified Modeling Language (UML) zum Paradigma der modellgetriebenen Entwicklung, dem Model-Driven Development (MDD) (Stahl/Voelter/Czarnecki 2006, 4). Das Fundament der Entwicklung bildet hierbei das namensgebende Modell eines zu entwerfenden Systems, der Fokus verschiebt sich von der manuellen Programmierung zur Codegenerierung: Anstelle der direkten Implementierung einer Lösung wird diese zunächst modelliert, die Lösung anschließend anhand des Modells generiert und in Folgeschritten angepasst.

Der Grad der Automatisierung durch Codegenerierung wird durch Modellierungskonstrukte definiert. Nur wenn der Abstraktionsgrad von Code und Modell hoch ist, kann von echter Generierung gesprochen werden. Allgemeine Modellierungssprachen wie UML basieren jedoch auf Elementen die in enger Anlehnung an Programmierkonstrukte geschaffen wurden. Modell und entstehender Programmcode liegen damit auf dem gleichen Abstraktionsgrad. Tolvanen (2006) beschreibt die resultierende Redundanz als Roundtrip-Problem: Die gleiche Information muss konsistent an zwei verschiedenen Stellen verwaltet werden. Aufgrund der Deckungsgleichheit wird kein Vorteil gewonnen, denn das Modell beschreibt Funktionalitäten und Verhalten auf die gleiche Weise wie Programmcode, hätte also direkt als solcher umgesetzt werden können. Die Transformation einer Klasse von einem Modelldiagramm in Code beschreibt keine echte Generierung.

4.1.6 Domänenspezifische Modellierung

Vielfach wird in der Fachliteratur die Analogie der Abstraktionssteigerung bei Programmiersprachen herangezogen: Während alle Programmiersprachen auf der untersten Ebene Assemblercode erzeugen, so konnte erst durch die Einführung stärker abstrahierender Sprachen der dritten Generation höhere Produktivität erzielt werden. Der logische Ansatzpunkt für eine Verbesserung ist folglich eine Erhöhung des Abstraktionsgrades.

Diese kann durch Spezialisierung der Modellierungskonstrukte auf die Problemdomäne erreicht werden. Modellierungselemente werden nicht länger in Anlehnung an Programmierungselemente gestaltet, sondern in Anlehnung an Entitäten und Zusammenhänge des Problemgebiets, denn jede Domäne besitzt eigene Abstraktionen, Konzepte und Regeln (Kelly 2005). Beschränkungen und Richtlinien der Domäne werden als Constraints in die Modellierungssprache eingebunden um die Gültigkeit entstehender Modelle für jeden Anwendungsfall zu garantieren. So erscheint es beispielsweise im Kontext der Entwicklung von Automotive Software logisch, Begrifflichkeiten der Domäne wie „Auswahlmenü“, „Sprachbefehl“ oder „aktueller GPS-Standort“ direkt und als eigene Konstrukte in die

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Modellierungssprache zu übernehmen, statt ihre Funktionalitäten mittels Aggregation visualisierter Codeklassen nachzubauen.

Tolvanen und Kelly (2004) nennen Flexibilität als einen der großen Vorteile der domänenspezifischen Modellierung (DSM): Da Änderungsanforderungen zumeist der Problemdomäne und nicht der Implementierungsdomäne entstammen, können Änderung leicht und intuitiv in ein vorhandenes Modell integriert werden. Darüber hinaus garantiert die Fokussierung auf eine spezifische Problemdomäne, dass sich das Expertenwissen hinsichtlich dieser Domäne in jedem Modell wiederfindet. Jeder mögliche Problemkontext ist ohne Aufwand zu modellieren, ungültige Situationen werden ausgeschlossen. Durch die augenscheinliche Loslösung von der Implementierungsebene – diese ist nun in das Modell bzw. den Codegenerator integriert – ist es Nicht-Programmierern möglich, Lösungen direkt in der Problemdomäne zu gestalten, ohne sie zunächst in die Umsetzungsdomäne zu transformieren und ohne neue Semantiken erlernen zu müssen.

4.1.6.1 Beispiel domänenspezifischer Modellierung

Tolvanen (2006) zeigt den Einsatz einer domänenspezifischen Modellierungssprache im Bereich der Programmierung digitaler Armbanduhren. Wie in Abbildung 4-5 zu sehen, basiert die Modellierung einer Uhr hierbei auf Zustandsautomaten. Übergänge werden durch die modellierten Knöpfe „Mode“ und „Set“ ausgelöst. Durch die Orientierung auf Elemente des Anwendungsgebiets und das Fehlen jeglicher codebasierter Konstrukte im Modell ist die Erhöhung des Abstraktionsniveaus evident. Dies garantiert eine einfache Erweiterung des Modells im Fall von Anforderungsänderungen: Da die Realität nicht erst über eine abweichende Grammatik in ein Modell übersetzt werden muss, ist keine Anpassung dieser Grammatik vonnöten. Eine Beziehung zwischen Code und Modell wird erst in einem späteren Schritt durch den Codegenerator hergestellt.

Auch ist aus Abbildung 4-5 ersichtlich wie domänenspezifische Beschränkungen durch Constraints nativ in das Modell integriert wurden. Zustandsübergänge können nicht ohne Interaktion mit dem Nutzer, also ohne Drücken eines Knopfes, angestoßen werden. Aktionen können nur bestimmten Zuständen entspringen. Darüber hinaus kann eine Aktion, die mit einer Transition einhergeht, nur Daten vom Typ „MI“ oder „HO“ ändern.

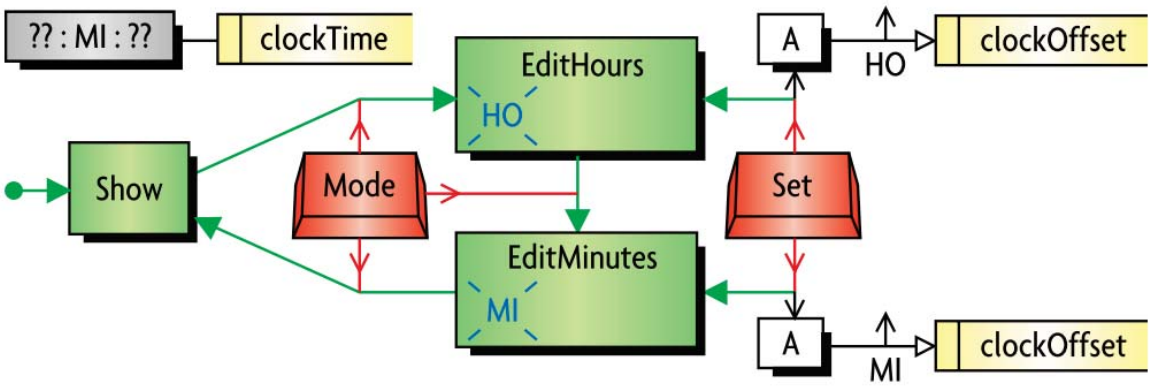


Abbildung 4-5 Beispiel domänenspezifischer Modellierung
Quelle: (Tolvanen 2006)

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

4.1.6.2 Architektur domänenspezifischer Modellierung

Tolvanen und Kelly (2004) beschreiben die Architektur eine domänenspezifische Modellierung aus zwei unterschiedlichen Blickwinkeln. Einerseits, wie in Abbildung 4-6 zu sehen, beschreiben sie die Sicht der Domäne auf die Problemstellung, andererseits die des Gestalters einer neuen Lösung. Aus Sicht der Domäne besteht eine domänenspezifische Modellierung aus drei Bestandteilen: Der domänenspezifischen Modellierungssprache, einem domänenspezifischen Code-Generator sowie einem Domänen-Framework (Code Umsetzung domänenspezifischer Konstrukte). Aus Sicht des Gestalters einer neuen Lösung übersetzen sich diese drei Konstrukte zum einen in ein Modell der Lösung sowie den daraus entstehenden Produktcode. Dieser setzt sich, transparent für den Nutzer, aus dem Domänen-Framework und der domänenspezifischen Codegenerierung zusammen.

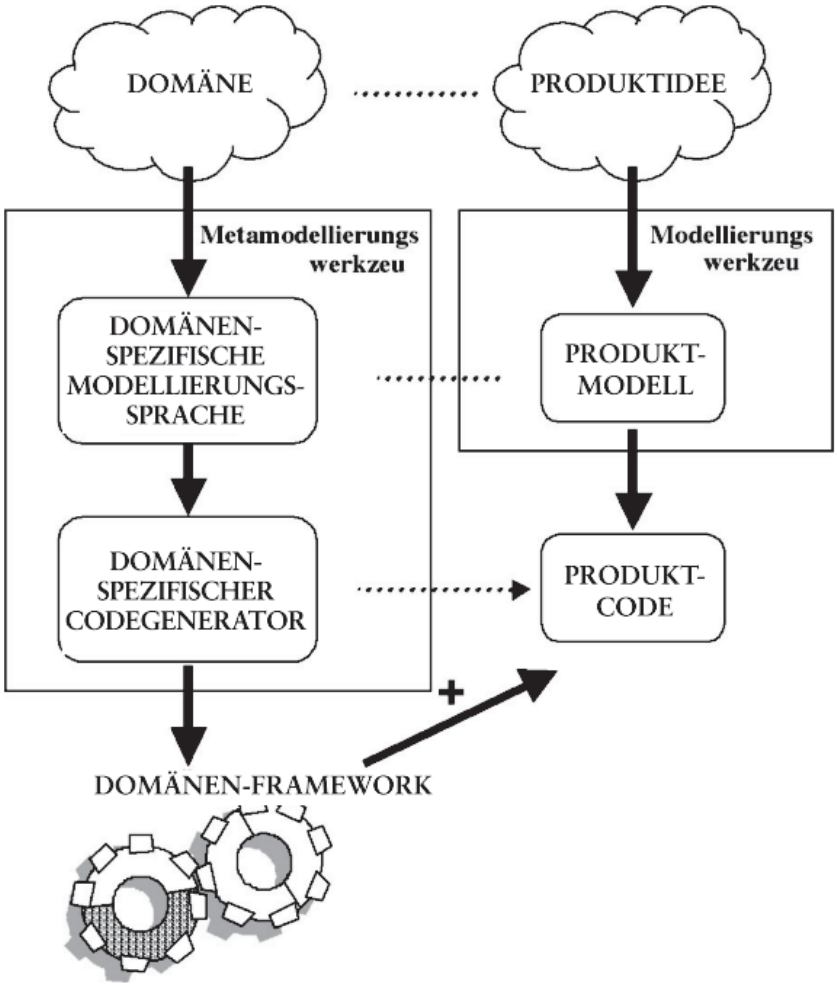


Abbildung 4-6 Architektur einer DSM-Umgebung
Quelle: (Tolvanen/Kelly 2004)

Die domänenspezifischen Modellierungssprache, welche vom Gestalter zur Umsetzung eines Lösungsmodells eingesetzt wird, umfasst dabei drei grundlegende Aspekte (Tolvanen 2006): Konzepte der Problemdomäne, eine Notation sowie Regeln für den Modellierungsprozess die eine gültige Semantik sicherstellen.

Die Konzepte der Problemdomäne sind es, die eine Abstraktion der Modellierung auf ein der Problemdomäne angemessenes Niveau heben. Es werden also nicht funktionale Aspekte der

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Umsetzung sondern vielmehr gestalterischer und konzeptionelle Bestandteile des Problems modelliert. Diese Konzepte sind es, die das domänenspezifische Wissen Nutzbar machen.

Die Konzepte der Domäne alleine machen allerdings noch keine syntaktisch sinnvolle Modellierung aus. Im Rahmen der Notation werden die Konzepte aus der Domäne in Modellierungskonstrukte übertragen. Die Notation umfasst also alle jene Elemente, die zur Darstellung der Konzepte der Problemdomäne benötigt werden.

Der letzte Aspekt der domänenspezifischen Modellierung ist das Wissen über die Zusammenhänge und Regeln der einzelnen Konzepte. Diese Regel legen fest, wie die, mithilfe der in der Notation, beschriebenen Konzepte zusammenhängen können und welche Arten von Verbindungen zwischen einzelnen Elementen erlaubt sind.

4.1.7 Diskussion

Es lässt sich also feststellen, dass eine domänenspezifische Modellierung ein geeignetes Werkzeug für eine Gestaltung von Automotive Services darstellt. Auf diese Weise können komplexe Sachverhalte und technische Details transparent abstrahiert und im Rahmen der Modellierung visualisiert werden. Technische Details der Fahrzeuge und verfügbare Funktionalität, wie beispielsweise GPS Navigation oder der Zugriff auf den CAN-Bus, können so konzeptionell erfasst und unter Berücksichtigung einer geeigneten Notation genutzt werden. Darüber hinaus können die Regeln, die ein Zusammenspiel der einzelnen Konzepte ermöglichen, ebenfalls im Rahmen der Modellierung hinterlegt werden.

Auf diese Weise können durch den Einsatz einer domänenspezifischen Modellierung von Automotive Services Stakeholder der unterschiedlichen Kompetenzprofile effizient zusammenarbeiten. Auf der einen Seite sind Domänenexperten in der Lage, geeignete Konzepte zu explizieren um diese anderen Stakeholder zugänglich zu machen. Auf der anderen Seite können die Experten der Umsetzungsdomänen diese Konzepte im Rahmen des Domänen-Frameworks mit funktionalen Komponenten hinterlegen. Auf diese Weise steht sowohl die Konzeption im Modell als auch die Funktion im Code zur Verfügung. Ausgehend von den Beiträgen dieser beiden Gruppen können einerseits alle Stakeholder Automotive Services komponieren und in diesem Rahmen zusätzliche Konzepte identifizieren. Somit ist eine domänenspezifische Modellierung von Automotive Services in der Lage, mit ihrer Nutzung zu wachsen und stellt somit ein evolutionäres Konzept dar, welches auch mit zukünftigen, bislang noch unbekannten, Problemstellungen zurechtkommt. Schließlich können Automotive Service Experten komplexere Funktionen in Form von Modellen definieren, die wiederum Bestandteile komplexerer, kompositer Modelle sein können.

4.2 "Design Thinking" als Vorgehensmodell

Eine erfolgreiche Modellierungssprache setzt jedoch auch ein geeignetes Vorgehen für dessen Nutzung voraus (Braun 2007, 16). Bevor also die einzelnen Konzepte sowie eine geeignete Notation und darauf aufbauende Regeln definiert werden können, muss zunächst das Vorgehen der Gestaltung, dessen einzelne Schritte sowie die aus diesen Schritten folgenden Resultate betrachtet werden. Neben den domänenspezifischen Eigenschaften bilden die Herausforderungen des Vorgehens die wesentlichen Gestaltungsgrundlagen für eine Modellierung von Automotive Services.

Design Thinking ist ein Ansatz für die Integration von technischen und nichttechnischen Stakeholdern in die praktische, kreative Lösung von Problemen oder relevanter

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Fragenstellungen. Die Methode fördert die Generierung und Gestaltung innovativer Automotive Services durch eine Konzentration auf heterogene Gruppen von Akteuren. Design Thinking ist im Gegensatz zu analytischem Denken ein Konzept, das sich rund um Kreativität und den Aufbau von neuen Ideen dreht. Eingeführt von Terry Winogard (1996) eliminiert Design Thinking die Angst vor Versagen und fördert größtmöglichen Einsatz und Teilnahme an der Gestaltungs- und Prototypenphase. Der Grundgedanke dahinter ist, eine kluge Verknüpfung einer bottom-up Experimentation mit einer begleitenden Anleitung von oben zu erreichen. Tim Brown illustriert diesen Gedanken anhand von sechs einfachen Regeln (Brown 2009, 73):

(1) Die besten Ideen entstehen, wenn nicht nur den Designern, Entwicklern oder dem Management sondern vielmehr dem gesamten Ökosystem der Organisation Raum zum Experimentieren gelassen wird.
(2) Diejenigen, die externem Wandel (neue Technologien, strategischen Herausforderungen, ...) ausgesetzt sind, sind auch diejenigen, die am geeignetsten sind auf diesen zu reagieren.
(3) Einzelne Ideen sollten nicht daran gemessen werden woher sie stammen. Wichtiger ist die Idee und nicht die Stellung desjenigen der sie hatte.
(4) Ideen, die Aufmerksamkeit auf sich ziehen, sollten bevorzugt werden. Erst wenn über Ideen gesprochen wird sollten dies auch organisatorisch unterstützt werden.
(5) Die strukturierenden Fähigkeiten des Managements sollten genutzt werden neue Ideen zu pflegen, zu entwickeln und schließlich auch umzusetzen.
(6) Ein übergeordnetes Ziel artikuliert eine klare Richtung und verhindert das Gefühl einer stetigen Überwachung.

Das Vorgehen des Design Thinking stellt also kein klar definiertes Vorgehensmodell dar, sondern ist vielmehr ein kulturelles und konzeptionelles Rahmenwerk für die partizipative Gestaltung von Innovationen. Es ist wichtig, das Entstehen und die Entwickeln von Ideen zu unterstützt. Nichts desto trotz gibt es eine Reihe an Elementen, die diesen Prozess lenken und unterstützen. Für eine erfolgreiche Gestaltung von Automotive Services müssen also im Sinne des Design Thinking diese Elemente durchlaufen und beachtet werden. Folglich muss auch ein Modellierungsansatz die Eigenarten und Aspekte dieser Elemente betrachten.

4.2.1 Elemente des Design Thinking

Abbildung 4-7 zeigt die sieben Elemente des Design Thinking Zyklus. Die einzelnen Elemente verstehen sich nicht als sequenziell oder parallel abzuarbeitende Phasen, sondern stellen vielmehr Aktivitäten dar, die erfolgreiche Kreativität unterstützen. Je nachdem welche Ausgangssituation initial vorherrscht, kann beispielsweise mit einer Auswahl vorhandener Prototypen oder aber auch mit dem Lernen aus einer existierten Implementierung begonnen werden. Auch ist die Reihenfolge der Elemente dabei nicht streng festgelegt. Natürlich gibt es jedoch eine Reihe an Vorbedienungen die gegeben sein müssen. So kann eine Auswahl der besten Ideen erst dann erfolgen, wenn diese zunächst definiert wurden.

Abbildung 4-7 stellt die sieben Elemente des Design Thinking rund um die Gestaltung von Automotive Services dar. Betrachtet man den Ansatz als sequenzielles Vorgehen, so beginnt man mit der Wahl der besten Fragestellung, der Definition von Bedürfnissen und der Feststellung von Herausforderungen. Darauf Aufbauend folgt einer Phase der Recherche und Forschung, die zur Inspiration, zum Erkenntnisgewinn und zum Ausloten von Rahmenbedingungen dient. Ein anschließendes Brainstorming hilft Ideen zur Befriedigung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

der identifizierten Anforderungen zu gewinnen. Diese dürfen in Anlehnung an die Regeln von Braun nicht bewertet werden (Brown 2009, 73). Ausgehend von den definierten Ideen werden Prototypen und Entwürfe kreiert, um ausgewählte Konzepte testen und verbessern zu können. Auf der Grundlage dieser Prototypen kann nun die beste Idee ausgewählt und im Rahmen eines Projekts anschließen umgesetzt werden. Das letzte Element dient dem Lernen aus den vorangegangenen Aktivitäten. Erkenntnisgewinne aus der Umsetzung fließen in eine iterative Weiterentwicklung der Ideen ein.

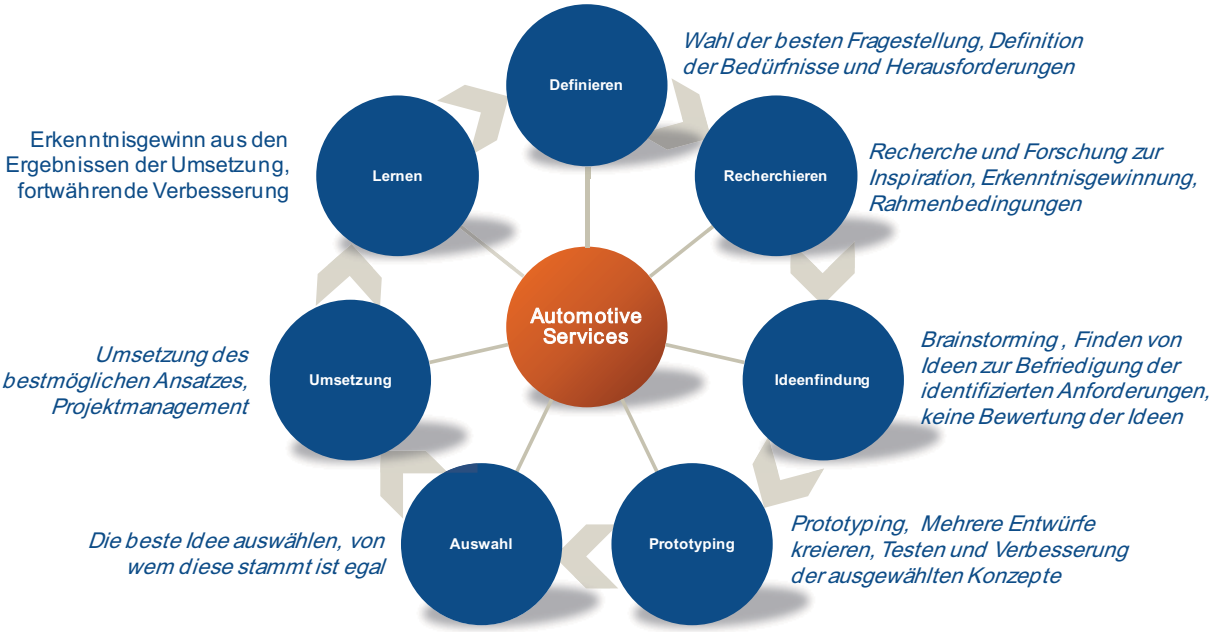


Abbildung 4-7 Design Thinking bei der Gestaltung von Automotive Services
Quelle: Eigene Darstellung in Anlehnung an (Winograd 1996)

Die Leitidee beim Durchlaufen dieser sieben Elemente ist Geschwindigkeit. Design Thinking fordert seine Teilnehmer auf, die einzelnen Iterationen lieber schnell und häufig zu durchlaufen, als sich besonders lange an einzelnen Aktivitäten aufzuhalten: „The faster we make our ideas tangible, the sooner we will be able to evaluate them, refine them, and zero in on the best solution“(Brown 2009, 89). Die nachfolgenden Abschnitte geben einen genauen Überblick über die sieben Elemente und beschäftigen sich mit der Fragestellung, welche Anforderungen eine domänenspezifische Modellierung von Automotive Services erfüllen muss, um ein Design Thinking geeignet zu unterstützen.

4.2.1.1 Definieren

Das erste Element des Design Thinking ist die Definitionsphase. In diesem Schritt werden die zentralen Fragestellungen die im Rahmen des Projekts beantwortet werden sollen von Stakeholdern gemeinsam festgelegt. Darüber hinaus werden die wesentlichen und entscheidenden Bedürfnisse sowie Herausforderungen die dabei beachtet werden müssen definiert. Im Einzelnen bedeutet dies, dass die initialen Teilnehmer entscheiden, welche Problemstellungen gelöst werden und insbesondere für welches Zielpublikum dies geschehen soll. Abhängig von diesen beiden Aspekten werden eventuell zusätzliche Stakeholder integriert. Schließlich wird noch ein gemeinsamer Glossar angelegt um eine Begriffsbasis für alle Beteiligten zu schaffen.

Obwohl es in diesem Schritt maßgeblich um das Installieren einer initialen Projektbasis geht und weniger um die eigentliche Innovation, impliziert dieser Phase die Notwendigkeit nach

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

einem gemeinsamen Ausgangspunkt zur Beschreibung des aktuellen Problems als auch des angestrebten Lösungsraums.

Anforderung 1: Gemeinsamen Ausgangspunkt zur Beschreibung des aktuellen Problems und angestrebten Lösungsraums für die Beteiligten schaffen.

4.2.1.2 Recherchieren

Das zweite Element des Design Thinking befasst sich mit dem Suchen und Erforschen der Problemstellung. Ziel ist es, Inspiration und Information zu fördern und gleichzeitig ein innovatives Umfeld zu schaffen. Die Teilnehmer analysieren die Historie der Problemstellung und tragen Hintergrundinformationen zusammen. Darüber hinaus werden existierende Versuche die Problemstellung zu lösen, sei es in der gleichen oder einer anderen Domäne, betrachtet. An dieser Stelle erfordert das Design Thinking potenziell wichtige zusätzliche Stakeholder zu identifizieren, beispielsweise Projektunterstützter, Investoren und vor allem auch Kritiker. Desweiteren schlägt Design Thinking vor, sowohl Nutzer als auch Technologieführer zu identifizieren um wichtige Constraints zu erfassen.

Um diese Aufgabe zu unterstützen müssen Ideen und Designalternativen schnell dokumentiert und expliziert werden können. Auf diese Weise können Rechercheergebnisse auf einfache Art und Weise nutzbar gemacht werden.

Anforderung 2: Ermöglichen eines schnellen und einfachen Wegs zu Explikation bereits existierende Ideen und Designalternativen.

4.2.1.3 Ideenfindung

Das dritte Element ist eine der wichtigsten Phasen des Design Thinking. In diesem Schritt generieren und entwickeln die Teilnehmer unterschiedliche Ideen und Lösungsvorschläge für Probleme und Herausforderungen auf der Grundlage des bislang zusammengetragenen Wissens. Hierbei ist insbesondere wichtig, dass die Teilnehmer eine qualitative Beurteilung sowie eine Einschätzung der Machbarkeit der einzelnen Ideen vermeiden (Brown 2009, 74). Die Erarbeitung von Ideen erfordert es von den Teilnehmern, Ähnlichkeiten und Unterschiede der einzelnen Ideen schnell zu identifizieren und diese mit den individuellen Charakteristika, Bedürfnissen und Motiven der angestrebten Nutzer in Einklang zu bringen.

Eine wesentliche Grundlage für eine erfolgreiche Ideenfindung ist es, Ideen und Gestaltungsalternativen schnell kommunizieren zu können. Eine effektive Zusammenarbeit der einzelnen Ideengeber basiert maßgeblich auf einer effizienten Kommunikation der einzelnen Ideen. Wichtig ist dabei, dies an einer geeigneten Art und Weise durchzuführen um eine pragmatische Qualität in Anlehnung an Schermann und Krogstie sicherzustellen (Schermann 2009; Krogstie 1998).

Anforderung 3: Effektive und effiziente Kommunikation von Ideen und Designalternativen.

4.2.1.4 Prototyping

Die Gestaltung unterschiedlicher Lösungsvorschläge für die vielversprechendsten Ideen durch die Teilnehmer ist das vierte Element des Design Thinking. Das Ziel dieser Phase ist es, die

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

dahinterliegenden Konzepte und Lösungen zu validieren und nach Möglichkeit zu verbessern. Um unterschiedliche Designalternativen zu erhalten werden Einzelideen kombiniert, erweitert und verändert. Prototypen kritischer Lösungseigenschaften helfen dabei ein durchdringendes Verständnis dieser Lösung zu entwickeln. Des Weiteren können Stakeholder konkretes Feedback zur Machbarkeit und Nützlichkeit einzelner Designalternativen liefern. In jedem Fall sollte der Fokus aber weiterhin auf der Vielfältigkeit und Kreativität liegen, ohne dass einzelne Alternativen dabei bewertet werden und die Neutralität der Teilnehmer hinsichtlich einzelner Ideen gefährdet ist.

Da die Gestaltung und Umsetzung von Lösungsvorschlägen und Prototypen ein zeitaufwändiger Prozess ist, ist dieser Schritt oftmals schwierig durchzuführen. Versetzt man die Teilnehmer jedoch in die Lage, einzelne Aspekte von Lösungsvorschlägen kreativ zu den kombinieren anstatt diese Aspekte jeweils neu zu entwickeln, können Vorschläge und Prototypen leicht erweitert und abgeändert werden.

Anforderung 4: Unterstützen einer einfachen Gestaltung, Reorganisation und Abänderung von Lösungsvorschlägen.

4.2.1.5 Auswahl

Sind mehrere Lösungsvorschläge vorhanden, so ist die Aufgabe im fünften Schritt des Design Thinking die, die vielversprechendste Lösung auszuwählen. Wurden bislang alle Ideen als gleichwertig und gleich wertvoll betrachtet, erfolgt in diesem Schritt eine Fokussierung und Festlegung auf einen einzelnen Lösungskandidaten. Dies bedeute jedoch nicht, dass die anderen Ideen und Lösungsvorschläge verworfen werden. In weiteren Iterationen des Vorgehens können diese entweder als eigenständige Lösungen oder auch als Ideengeber und Erweiterungen der gewählten Lösung zum Tragen kommen. Um diese Aufgabe durchzuführen, bewerten die Teilnehmer die Ziele des Projekts und wählen die dazu passendste Designalternative aus.

Von zentraler Bedeutung in diesem Schritt ist es, das die Teilnehmer emotionale Aspekte beiseitelassen und auf eine Urheberschaft einzelner Ideen verzichten. Natürlich geht es an dieser Stelle nicht um den Konsens des kleinsten gemeinsamen Nenners, sondern vielmehr um eine möglichst objektive Auswahl der geeignetsten Lösung. Um dies zu erreichen müssen die Teilnehmer in die Lage versetzt werden, über den Wertbeitrag einzelner Lösungsvorschläge objektiv zu diskutieren.

Anforderung 5: Einschätzen des Wertbeitrags einer Lösung ermöglichen.

4.2.1.6 Umsetzung

Die Umsetzung der gewählten Lösung ist das nächste Element im Design Thinking. Abhängig von den Rahmenbedingungen des Projekts, der angestrebten Anwendungsdomäne sowie möglichen weiteren externen Anforderungen können hier unterschiedliche Technologien und Lösungswege beschritten werden.

In Anbetracht der Tatsache, dass die Gestaltung von Automotive Services in einem schnellen, iterativen Vorgehen durchgeführt werden muss, um ein möglichst zeitnahes Feedback auf die implementierte Lösung zu erhalten, muss auch die Umsetzung und dass Deployment dieser schnell erfolgen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anforderung 6: Ermöglichen einer schnellen Umsetzung der gewählten Lösungsalternative.

4.2.1.7 Lernen

Mit dem siebten und letzten Element, dem Lernen, wurden alle Elemente des Design Thinking einmal durchlaufen. In diesem Schritt erheben die Teilnehmer Evaluationsdaten aus der realisierten Lösung, beispielsweise durch die Beobachtung und Befragung von Lead Usern, die diese nutzen. Auf der Grundlage dieser Informationen können die Teilnehmer potentielle Verbesserungen diskutieren und somit den nächsten Zyklus des Design Thinking vorbereiten.

Eine Modellierung von Automotive Services muss somit das Dokumentieren und Vorbereiten des Lernens aus dem Einsatz der Umsetzung für zukünftige Lösungsvorschläge unterstützen.

Anforderung 7: Vorbereiten von Schlussfolgerungen für zukünftige Lösungsvorschläge.

4.2.2 Diskussion der Herausforderungen

Zusammenfassend lässt sich feststellen, dass das Vorgehen des Design Thinking einen geeigneten Rahmen für die Gestaltung von Automotive Services darstellt. Durch die konzeptionelle Ausrichtung technische als auch nichttechnische Stakeholder in einer partizipativen Gestaltung von Problemstellungen zu unterstützen, eignet sich die Methode besonders die vier identifiziert Stakeholdergruppen (vgl. Abschnitt 3.1.2) zu integrieren.

Eine Betrachtung der einzelnen Elemente des Vorgehens zeigt jedoch, dass ein effektives Design Thinking eine gemeinsame Repräsentation von Ideen und Lösungsvorschlägen voraussetzt. Insbesondere das sechste Element, die Umsetzung der gewählten Lösungsalternative, stellt eine besondere Herausforderung dar. Ein begrenzter Zeithorizont sowie eingeschränkten Ressourcen zwingen die Teilnehmer eine bestimmte Lösung für die Umsetzung zu wählen. Jedoch bedrohen die in der Automobilindustrie üblichen langen Entwicklungszyklen die Vorteile des Design Thinking für die Entwicklung von Automotive Services. Es ist daher von entscheidender Bedeutung, den Zyklus des Design Thinking möglichst rasch zu durchlaufen und trotzdem keine der Stakeholdergruppen dabei zu verlieren. Abbildung 4-8 stellt die aus den Anforderungen des Design Thinking abgeleiteten Herausforderungen an eine Modellierung von Automotive Services dar.

Die erste Herausforderung bei der Nutzung von Design Thinking für die Gestaltung von Automotive Services ist es, die beiden Gruppen der nichttechnischen Stakeholder, insbesondere die der sonstigen Stakeholder, in die Lage zu versetzen, ihre Visionen und Anforderungen an Automotive Services zu explizieren. Betrachtet man die Anforderungen die das erste, dritte und fünfte Element des Design Thinking an eine unterstützende Modellierung stellen, das Schaffen eines gemeinsamen Ausgangspunkt zur Beschreibung des aktuellen Problems und des angestrebten Lösungsraum (A1), die effektive und effiziente Kommunikation von Ideen und Designalternativen (A3) sowie das Ermöglichen der Einschätzen des Wertbeitrags einer Lösung (A5), so fordern alle drei die Unterstützung einer einfachen Explikation und Kommunikation domänenspezifischer Ideen und Visionen. Wie in Abschnitt 4.1.5 diskutiert, eignen sich domänenspezifische Ansätze besonders für diese Herausforderung. Folglich erfordert ein Design Thinking eine domänenspezifische

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Repräsentation der Aktivitäten und Aufgaben von Automotive Services, um Stakeholder in die Lage zu versetzen den erwarteten Wertbeitrag eins Services zu explizieren.

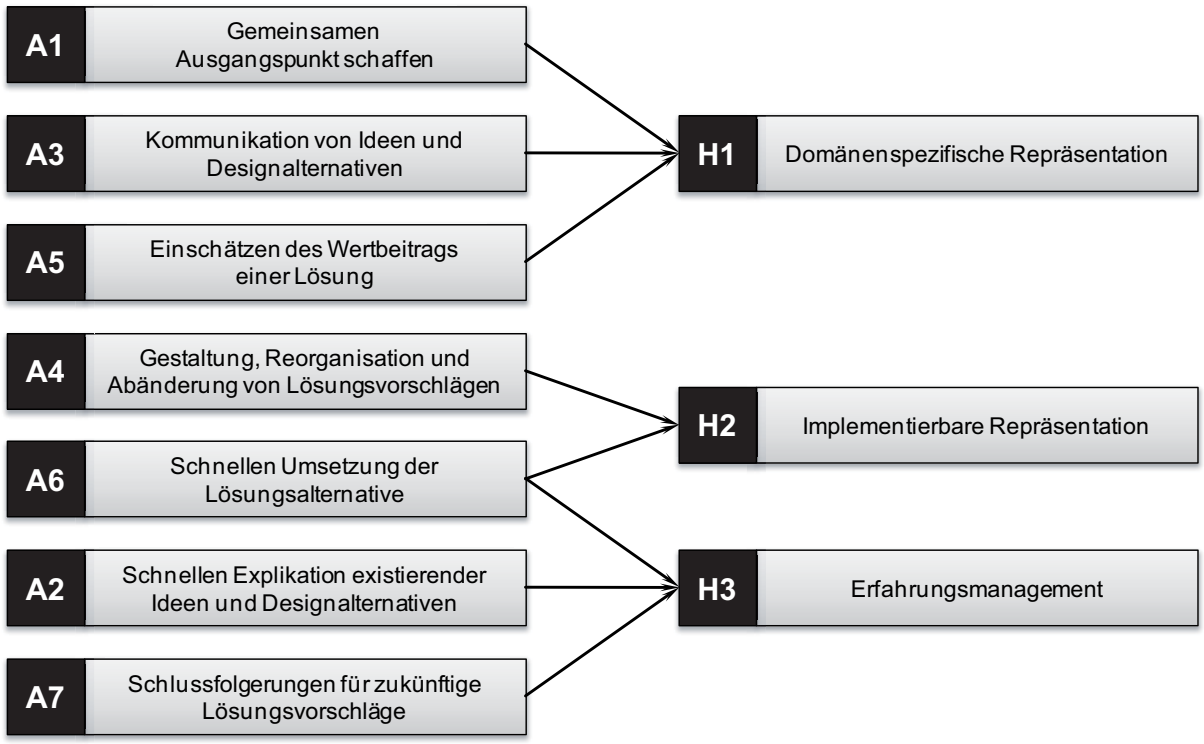


Abbildung 4-8 Herausforderungen des Design Thinking für Automotive Services
(Anforderungen wurden sortiert um überlappende Pfeile zu vermeiden)
Quelle: Eigene Darstellung

Aus den Anforderungen des vierten und sechsten Elements des Design Thinking folgt die zweite Herausforderung für eine Modellierung von Automotive Services. Die Unterstützung einer einfachen Gestaltung, Reorganisation und Abänderung von Lösungsvorschlägen (A4) sowie das ermöglichen einer schnellen Umsetzung der gewählten Lösungsalternative (A6) fordert eine implementierbare Repräsentation der Services. Die Teilnehmer müssen in die Lage versetzt werden, existierende und vorgeschlagene Möglichkeiten kreativ zu kombinieren um den aktuellen Stand der Diskussion zeitnah zu reflektieren. Beispielsweise dürfen Lösungsvorschläge nicht mit Sicherheitsbestimmungen oder übergeordneten Unternehmenszielen kollidieren. Die von den Stakeholdern vorgeschlagenen Lösungsmodelle müssen daher alle notwendigen Informationen für eine Implementierung des Automotive Services beinhalten.

Die dritte Herausforderung setzt sich schließlich aus den Anforderungen des sechsten, zweiten und dritten Elements zusammen. Eine schnelle Umsetzung der gewählten Lösungsalternative (A6), ein schneller und einfacher Weg zu Explikation bereits existierende Ideen und Designalternativen (A2) sowie das Festhalten von Schlussfolgerungen für zukünftige Zyklen (A7) erfordert das Vorhandensein einer Sammlung an bewerten und implementierten Bestandteilen von Automotive Services. Durch dieses Nutzbarmachen bereits gestalteter Lösungsaspekte werden Stakeholder in die Lage versetzt, existierende Ideen und Designalternativen zu bewerten und so eine möglichst schnelle Umsetzung einer gewählten Lösungsalternative sicherzustellen. Eine Modellierung von Automotive Services muss daher ein effektives Erfahrungsmanagement der Ergebnisse und Erkenntnisse vorangegangener Design Thinking Zyklen unterstützen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

4.3 Anforderungen aus der Praxis

Neben den Herausforderungen der Gestaltung von Automotive Services sowie den konzeptionellen Anforderungen aus dem operativen Einsatz von Modellierung in Anlehnung an das Vorgehen des Design Thinking, gibt es noch eine Reihe an praktischen Anforderungen, die sich aus dem Forschungsumfeld dieser Arbeit ergeben. Der nachfolgende Abschnitt diskutiert daher die Anforderungen die aus dem Einbezug der Stakeholder des Forschungsprojekts resultieren. Hierzu wird zunächst das spezielle Forschungsumfeld betrachtet, um auf dieser Grundlage relevante Stakeholder zu identifizieren. Ausgehend hiervon wird das Vorgehen bei der Integration dieser Stakeholder sowie die daraus resultierenden Anforderungen diskutiert.

4.3.1 Betrachtung des Forschungsumfelds

Die Erstellung einer interaktive Gestaltung von Automotive Services durch softwaregestützten Einsatz domänenspezifischer Modellierung erfolgt im Rahmen des Projekts HIMEPP Graphical Workbench (HGW), dass aus einer Kooperation des Automobilherstellers Audi AG aus Ingolstadt sowie der Technischen Universität München (TUM) entstanden ist. Im Rahmen der Public-Private-Partnership „Ingolstadt Institute der TUM“ (INI.TUM) findet eine regelmäßige Kooperation der beiden Institutionen statt. Ziel dieser Zusammenarbeit ist es, Forschungsprojekte unterschiedlichster Ausrichtung in enger Kooperation zwischen je einer Fachabteilung der Audi AG und einem Lehrstuhl der Universität voranzutreiben.

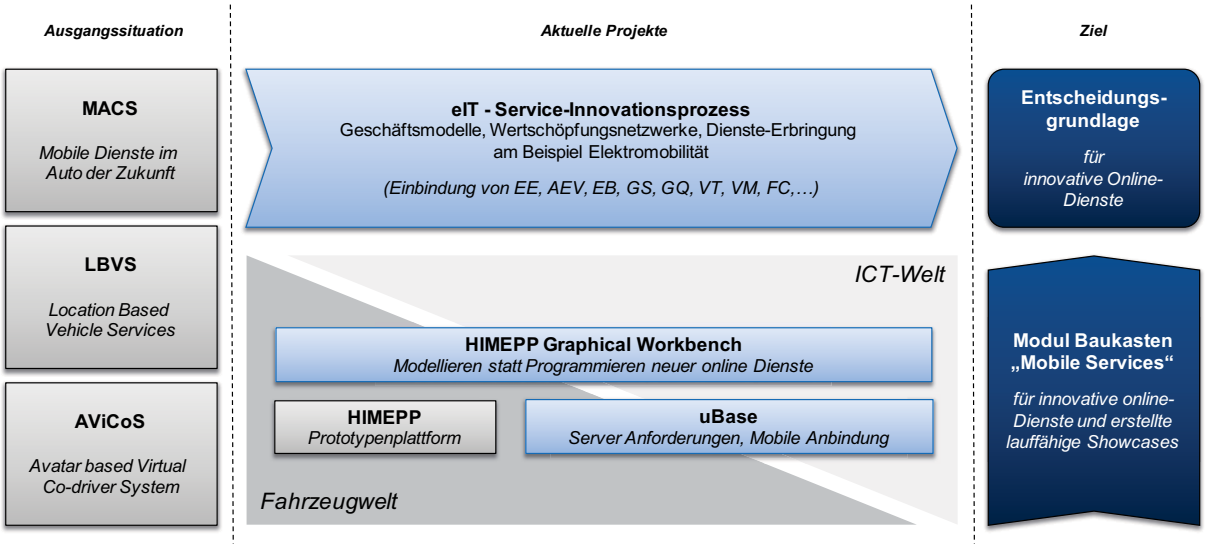


Abbildung 4-9 Projektübersicht "Entscheidungsgrundlage für Online-Dienste"
Quelle: Eigene Darstellung

Das Projekt HIMEPP Graphical Workbench ist innerhalb der INI.TUM Projekte in einen Kanon an Projekten eingegliedert, deren Ziel die Entwicklung einer Entscheidungsgrundlage für innovative Onlinedienste ist. Hierzu werden, wie in Abbildung 4-9 zu sehen, die Ergebnisse unterschiedlichster Forschungsprojekte zur Entwicklung dieser Entscheidungsgrundlage herangezogen. Zum Zeitpunkt der Erstellung dieser Arbeit bereits abgeschlossene Projekte sind hellgrau, aktuelle laufende Projekte hellblau und Ziele dunkelblau dargestellt.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Ausgangsituation für die Forschungsarbeiten sind die Projekte Mobile Dienste im Auto der Zukunft (MACS), Location Based Vehicle Services (LBVS) und Avatar based Virtual Co-driver System (AViCoS).

Das initiale Projekt MACS analysiert durch die Betrachtung der Problemfelder prototypische Umsetzung mobiler Services auf Basis innovativer Informations- und Kommunikationstechnologie (IKT), dem Aufbau einer verteilten IT-Infrastruktur als Basis für mobile Dienste, einem Vorgehensmodell zur Entwicklung und Implementierung mobiler Services, dem Aufbau und der Gestaltung eines kooperativen Wertschöpfungsnetzwerkes im Kontext wissensintensiver mobiler Telematikdienste und der Berücksichtigung sicherheitsrelevanter Fragestellungen sowie der Erschließung kooperativer Telematikdienste rund ums Automobil die Bedeutung und den Kontext von Automotive Services (Reichwald/Krcmar/Reindl 2007).

Aufbauend auf dem Ist-Zustand der Mobilfunkinfrastruktur wurden im Projekt LBVS Kriterien zur Typisierung mobiler Dienste abgeleitet und vorgestellt. Im Weiteren wurden diese Kriterien zu Evaluierung von Automotive Services durch Simulation bestimmt. Hier wurde neben dem Aufbau und der Funktionsweise des Simulationstools auch auf die Rahmenbedingungen und Limitationen der Simulation mobiler Dienste unter möglichst realen Bedingungen eingegangen. Außerdem wird die Datenbasis, auf der die Simulation beruht, ausführlich dargestellt und beschrieben (Golcar et al. 2010).

Mit Hilfe einer avatarbasierten, natürlich-sprachlichen Schnittstelle wird im Projekt AViCoS dem Fahrer auf einem vertrauten Weg die Möglichkeiten und das Verhalten des Fahrzeugs mitgeteilt. Die Kombination verschiedener Medien wie Ton und Bild erbringt dabei in verschiedenen Situationen eine Unterstützung in der Bedienung und der Interpretation des Fahrzeugs (Nicolescu 2009).

Auf der Grundlage der Betrachtungen der unterschiedlichen Aspekte von Automotive Services in den drei Ausgangsprojekten wird im Rahmen der aktuell laufenden Projekte ein Modul Baukasten „Mobile Services“ entwickelt. Hierbei sollen einerseits die technischen Aspekte der Fahrzeugwelt mit denen der IKT-Welt verknüpft werden und andererseits Geschäftsmodelle, Wertschöpfungsnetzwerke und Fragen der Dienste-Erbringung Berücksichtigung finden.

Das Projekt Highly Integrated Modular Embedded Prototyping Platform (HIMEPP), welches ebenfalls bereits abgeschlossen ist, liefert ein Werkzeug zum Rapid-Prototyping mobiler Dienste, das für die unterschiedlichen Anforderungen gleichermaßen benutzbar ist. Mit Hilfe von HIMEPP wird es möglich, kostengünstig und zeitnah Dienste für Kunden prototypisch in der automobilen Infrastruktur umzusetzen (Hoffmann 2010).

Als Erweiterung der Ergebnisse von HIMEPP von der Fahrzeugwelt in die IKT-Welt wird mit dem Ubiquitous Automotive Services Environment (uBase) ein Prototyp für die Erbringung von Diensten außerhalb des Fahrzeugs und deren Integration in das Fahrzeug entwickelt. Ziel ist eine Integration der in uBase entstandenen Komponenten in die HIMEPP Graphical Workbench.

Neben den technischen Aspekten werden im Projekt Service Innovationsprozess am Beispiel der Elektromobilität (eIT) nichttechnische Aspekte von Automotive Services betrachtet. Der Einsatz von Informationstechnik ermöglicht es im Bereich der Elektromobilität beispielsweise noch vorhandene Defizite bestehender Antriebskonzepte durch intelligente Routenplanung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

bzw. Lademanagement zu kompensieren, wodurch ein frühzeitiges Angebot elektrisch betriebener Fahrzeuge möglich wird. Gleichermaßen ermöglicht erst der sinnvolle Einsatz von IT die Synchronisation und Kopplung verschiedener Stakeholder im Kontext der Elektromobilität. Somit stellt der Informationsaustausch zwischen Fahrzeughersteller, Energielieferanten und Energiedistributoren eine Grundvoraussetzung der Elektromobilität dar.

4.3.2 Identifikation der Stakeholder

Wie bereits angesprochen wurde die vorliegende Forschungsarbeit im Kontext der Ingolstadt Institute der Technischen Universität München durchgeführt. Für eine Identifikation relevanter Stakeholder ergeben sich in diesen Projektkontext daher zwei unterschiedliche Herangehensweisen diese zu identifizieren. Einerseits müssen Vertreter der drei am Projekt beteiligten Institutionen, der Audi AG, der Technischen Universität München und INI.TUM, integriert werden. Andererseits ist diese Arbeit in ein übergeordnetes Forschungsvorhaben eingegliedert, so dass Stakeholder aus den parallel durchgeführten Projekten ebenfalls berücksichtigt werden müssen.

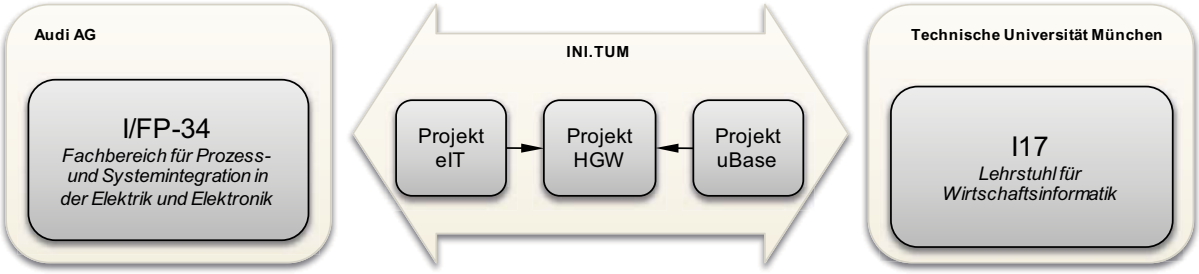


Abbildung 4-10 Organisationen und deren Zusammenarbeit im Forschungsumfeld
Quelle: Eigene Darstellung

Abbildung 4-10 stellt diesen Zusammenhang grafisch dar. Die Audi AG wird dabei durch den Fachbereich für Prozess und Systemintegration in der Elektrik und Elektronik (I/FP-34) und die Technische Universität München durch den Lehrstuhl für Wirtschaftsinformatik (I17) vertreten. Die Ingolstadt Institute der TUM werden doch die Bearbeiter der drei Projekte eIT, uBase und HGW repräsentiert.

Insgesamt waren am, im Anschluss vorgestellten, Vorgehen der Anforderungsermittlung 12 Stakeholder beteiligt. Aus dem Fachbereich von Audi waren acht Stakeholder und von Seiten des Lehrstuhls für Wirtschaftsinformatik der Technischen Universität München ein Stakeholder vertreten. Die restlichen drei Stakeholder sind die Projektbearbeiter der drei parallelen Forschungsarbeiten im Rahmen von INI.TUM.

4.3.3 Vorgehen

Für eine Integration der identifizierten Stakeholder wurde ein Vorgehen mit zwei aufeinander aufbauenden Aktivitäten gewählt. Zunächst wurden die Projektziele und vorhandenen Grundlagen im Rahmen eines Innovationsworkshops erläutert um anschließend im Rahmen des Workshops Anforderungen an eine interaktive Gestaltung von Automotive Services durch softwaregestützten Einsatz domänenspezifischer Modellierung zu gewinnen. In einem zweiten Schritt wurden die Anforderungen den Stakeholder in Form einer Ideen Community zugänglich gemacht, so dass diese weitere Anforderungen hinzufügen oder existierende weiter entwickeln können.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

4.3.3.1 Innovationsworkshop

Eine der bekanntesten Kreativitätstechniken für Gruppen ist das Brainstorming. Dabei werden Ideen in einer Gruppe während einer Sitzung gesammelt und von einem Moderator notiert. Die Ideen werden in kurzer Zeit gefunden und von mehreren Personen weiterentwickelt sowie anschließen einer Analyse unterzogen (Rupp 2007, 116). Im Rahmen der Anforderungsermittlung für die HIMEPP Graphical Workbench wurde das Werkzeug eines Brainstormings zusammen mit einer anschließenden Kategorisierung im Innovationsworkshop eingesetzt.

Für die Durchführung des Innovationsworkshops wurden alle Stakeholder in einen Seminarraum der Technischen Universität München eingeladen. Der Workshop wurde am 22. Januar 2010 von 12:30 Uhr bis 18:00 Uhr durchgeführt. Anwesend waren alle in Abschnitt 4.3.2 identifizierten Stakeholder, die Moderation übernahm der Projektbearbeiter des Projekts HIMEPP Graphical Workbench.

Im ersten Teil des Workshops wurden die Teilnehmer über den aktuellen Projektstand informiert sowie die weitere Projektplanung erläutert. Im Einzelnen wurden zunächst die Herausforderungen bei der Gestaltung von Automotive Services sowie die Motivation für das Projekt diskutiert. Anschließend wurde das angedachte Vorgehen zur Entwicklung von Automotive Services sowie die Idee des Einsatzes von Modellierung besprochen. Darauf folgte einen Vorstellung einer möglichen Architektur sowie eine Übersicht der geplanten Meilensteine für die Projektlaufzeit. Die Präsentation dieser Informationen diente einerseits dazu, alle Projektbeteiligten auf den gleichen Informationsstand zu bringen und andererseits dazu, eine gemeinsame Vorstellung und Vision des Projektergebnisses zu initiieren. Die Unterlagen der Präsentation finden sich in Anhang C.

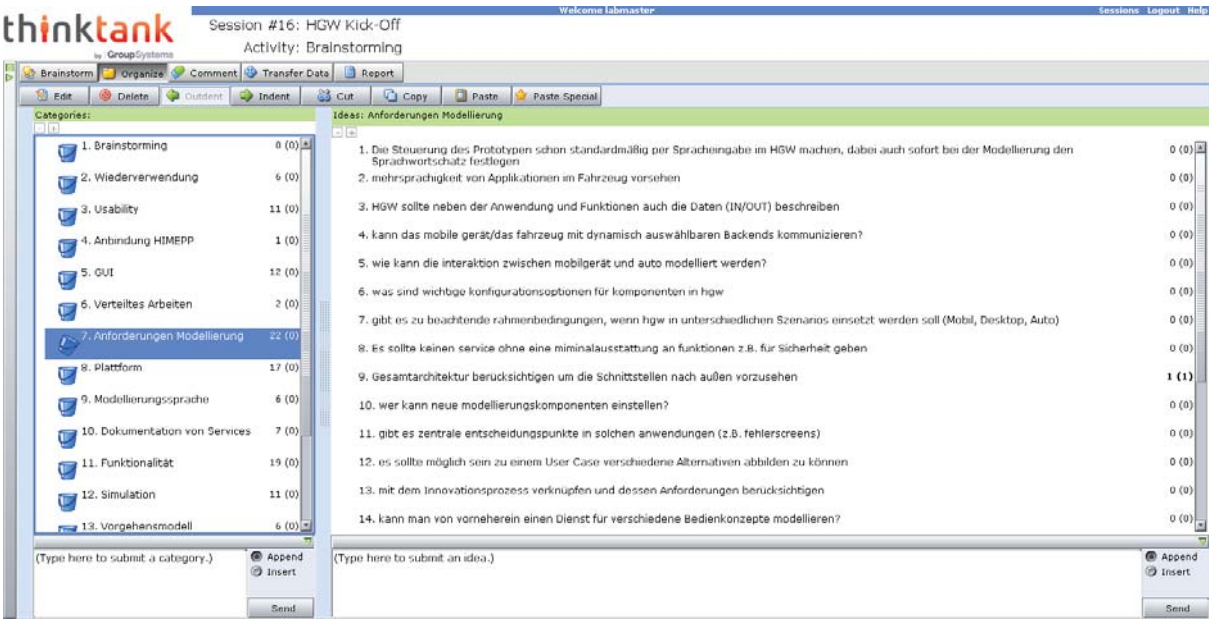


Abbildung 4-11 Electronic Meeting System ThinkTank von GroupSystems
Quelle: Eigene Darstellung

Im Anschluss an den Präsentationsteil erfolgt die eigentliche Erhebung der Anforderungen in Form eines Brainstormings. Die Stakeholder wurden hierzu jeweils mit einem eigenen Notebook ausgestattet, welches mit dem Electronic Meeting Systems (EMS) ThinkTank der Firma GroupSystems ausgestattet war. Wie in Abbildung 4-11 zu sehen, bietet ThinkTank den Teilnehmern eine einfache, browserbasierte Oberfläche, in der sie in anonymisierter Form

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Ideen beitragen können. Jeder der Stakeholder konnte dabei die Ideen der anderen Teilnehmer live auf seinem Bildschirm nachvollziehen und diese bei Bedarf kommentieren. Die Dauer des Brainstormings Betrug 45 Minuten und es wurden insgesamt 124 einzelne Ideen zusammengetragen.

Nach einer 15-minütigen Pause wurden im Anschluss an das Brainstorming die gesammelten Ideen im Plenum mit den Stakeholdern diskutiert und auf dieser Grundlage 14 Kategorien (Abbildung 4-11 links) für Anforderungen definiert. Anschließend wurden die einzelnen Ideen in diese 14 Kategorien gegliedert. Zum Abschluss des Innovationsworkshops wurde noch ein Ausblick auf die Ideen Community gegeben, mit deren Hilfe die Ideen im weiteren Verlauf des Projekts entwickelt und ergänzt werden sollten.

4.3.3.2 Ideen Community

Im Anschluss an den Innovationsworkshop wurden die gefundenen Anforderungen an eine Modellierung von Automotive Services den Stakeholder zur Ergänzung und zur Weiterentwicklung in Form einer Ideen Community online zur Verfügung gestellt. Die Teilnehmer haben somit über den kompletten Projektverlauf die Möglichkeit, Anforderungen zu ergänzen, zu kommentieren oder weiterzuentwickeln. Aufgrund der Tatsache, dass die Ideen Community über das Internet zuggreifbar ist, wurde die Teilnahme auf einen registrierten Personenkreis begrenzt. Die Stakeholder erhielten dafür einen Benutzernamen sowie ein Passwort um sich dem System gegenüber zu identifizieren.

Abbildung 4-12 Hauptbereich der Ideen Community
Quelle: (Riedl 2010a)

Die Ideen Community bietet zwei unterschiedliche Sichten auf Anforderungen an. In der Hauptansicht (vgl. Abbildung 4-12) werden Anforderungen in Form einer tabellarischen Übersicht dargestellt. Anforderung können dabei nach unterschiedlichen Kriterien, wie Datum der letzten Änderung oder Popularität sortiert und erforscht werden. Im Hauptbereich der Ansicht werden die wesentlichen Informationen der einzelnen Anforderungen zusammengefasst dargestellt und Stakeholder haben die Möglichkeit Bewertungen

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

einzusehen. Darüber hinaus besteht noch die Option neue Anforderungen (Ideen) zu erstellen, mit anderen Mitgliedern der Ideen Community in Kontakt zu treten oder Anforderungen nach häufigen Begriffen in Form einer Tag Cloud zu durchsuchen.

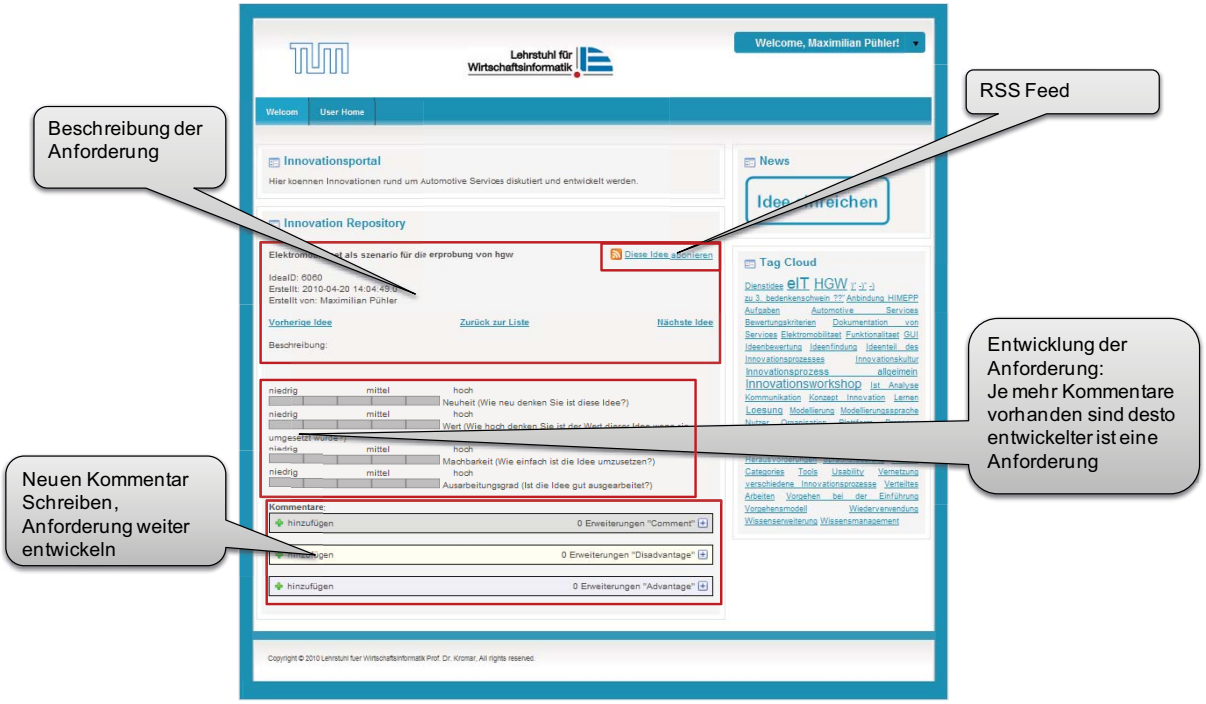


Abbildung 4-13 Einzelne Anforderung in der Ideen Community
Quelle: (Riedl 2010b)

In der Detailansicht (vgl. Abbildung 4-13) zu einer einzelnen Anforderung haben die Stakeholder die Möglichkeit die detaillierte Beschreibung einer Anforderung einzusehen und diese in Form eines RSS-Feed zu abonnieren. Auch kann hier eine Bewertung hinsichtlich unterschiedlichster Kriterien erfolgen und so der Reifegrad einer Anforderung verfolgt werden. Das wesentliche Werkzeug dieser Ansicht stellt jedoch das Kommentieren von Anforderungen dar. Auf diese Weise können sich die Stakeholder hinsichtlich einzelner Anforderungen untereinander austauschen und diese inhaltlich weiter entwickeln.

4.3.4 Diskussion der Anforderungen

Wie bereits erwähnt wurden im Verlauf des Innovationsworkshops als auch in der anschließenden Weiterentwicklung insgesamt 124 Anforderungen durch die Stakeholder formuliert. Anforderungen gliedern sich dabei in die Kategorien Wiederverwendung, Usability, Anbindung HIMEPP, GUI, Verteiltes Arbeiten, Anforderungen Modellierung, Plattform, Modellierungssprache, Dokumentation von Services, Funktionalität, Simulation, Vorgehensmodell, Nutzer und Sprachsteuerung. Anzumerken dabei ist, dass einzelne Anforderungen der Stakeholder in mehrerer Kategorien einsortiert werden konnten und unterschiedliche Antworten oftmals einen vergleichbaren Sachverhalt beschreiben. Aus diesem Grund werden die Anforderungen im Nachgang, sortiert nach den Kategorien, zusammengefasst und formuliert. Sich wiederholende Anforderungen sowie Verständnisfragen, die nicht in Form einer Anforderung beschrieben werden können, wurden dabei aussortiert. Des Weiteren ist zu bemerken, dass die meisten beteiligten Stakeholder an den in Abschnitt 4.3.1 beschriebenen vorausgegangenen Projekten in unterschiedlicher Intensität beteiligt waren und viele Anforderungen somit vor dem Hintergrund einer

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Weiterführung der in diesen Projekten entstandenen Ideen und Vorstellungen zu verstehen sind.

Auch wenn alle Stakeholder angaben, die Ideen Community genutzt zu haben, konnten dort keine Änderungen einzelner Anforderungen oder neue Anforderungen festgestellt werden. Den Stakeholdern stand die Ideen Community für einen Zeitraum von sechs Monaten im Anschluss an den Innovationsworkshop zur Verfügung. Die nachfolgende Tabelle 4-1 der Anforderungen basiert daher auf den 124 Anforderungen und 14 Kategorien wie sie in Anhang D zu finden sind. Insgesamt wurden 70 Anforderungen aus der Praxis aus den Anforderungen der Stakeholder extrahiert.

Kategorie	Anforderung
Wiederverwendung	A1 Sicherstellen eines methodischen Vorgehens. A2 Wiederverwendbarkeit von Services ermöglichen. A3 Wiederverwendbarkeit von Komponenten ermöglichen. A4 Erstellen von Templates ermöglichen.
Usability	A5 Einfache, nicht "überladene" Werkzeugunterstützung. A6 Erfahrungswissen anderer Modellierer bereitstellen. A7 Integration einer Anleitungen und Beispiele in der Werkzeugunterstützung. A8 Integritätsüberprüfung während der Modellierung. A9 Vorschau für den modellierten Dienst anbieten. A10 Unterstützung einer Stichwortsuche für Komponenten. A11 Keine IT-Kenntnisse voraussetzen. A12 Bereitstellen eines Repository für Funktionalitäten.
Anbindung HIMEPP	A13 Fahrzeugbezug der Modellierung durch spezielle Funktionen (GPS, CAN Bus Signale, ...) erweitern.
GUI	A14 Verteiltes Modellieren ermöglichen. A15 Unterstützung unterschiedlicher Eingabemedien (Dreh-Drück-Steller, Touchscreen, ...). A16 Unterstützung unterschiedlicher Ausgabemedien (FIS, MMI Screen). A17 Regeln für die Gestaltung von User Interfaces hinterlegen. A18 Bereitstellung von Standards und Modellierungsgrundlagen. A19 Unterstützung unterschiedlicher Look and Feels. A20 Ermöglichen einer Internationalisierbarkeit der modellierten Automotive Services. A21 Berücksichtigung verschiedener Fahrzeugmodelle.
Verteiltes Arbeiten	A22 Möglichkeit gemeinsam an Modellen/Prototypen zu arbeiten. A23 Enge Verknüpfung mit dem Innovationsprozess.
Anforderungen Modellierung	A24 Erweiterbarkeit hinsichtlich neuer Funktionalitäten wie Backend oder Mobile Devices. A25 Spracheingabe bei Services soll modellierbar sein.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

	A26 Einzelne Komponenten sollen konfigurierbar sein. A27 Ein Backend soll dynamisch in der Modellierung gewählt werden können. A28 Interaktionen zwischen unterschiedlichen Geräten modellieren. A29 Möglichkeit schaffen, Vorgaben wie Sicherheit zu modellieren. A30 Einzelne Komponenten können variabel auf verschiedene Geräte verschoben werden. A31 Automatisches anlegen einer Logdatei beim Ausführern von Automotive Services. A32 Modellieren von verteilten Automotive Services.
Plattform	A33 Berücksichtigung einer Touchpad/Touchscreen Unterstützung. A34 Modellieren direkt vor Ort im Fahrzeug. A35 Direktes Deployment ins Fahrzeug vorsehen. A36 Spezielle Komponenten zur Steuerung des Fahrzeugs. A37 Komponenten sollen dynamisch konfigurierbar sein A38 Interaktionen der Nutzer nicht nur im Fahrzeug, sondern auch auf dem Server (Web Server). A39 Browserbasierte Werkzeugunterstützung der Modellierung. A40 Integration externer Services.
Modellierungssprache	A41 Unterstützung einer Gestaltung von neuen Komponenten. A42 Hierarchisches Modellieren zur Vereinfachung der Modelle. A43 Experten- und Normalversion des Modellierungswerkzeugs. A44 Formale Beschreibung von Komponenten sollte rückwärtskompatibel sein.
Dokumentation von Services	A45 Direkte Beschreibung und Dokumentation von modellierten Diensten. A46 Versionierung der Modelle. A47 Export der Modelle in Lastenhefte unterstützen. A48 Dokumentation von Automotive Services durch Modelle.
Funktionalität	A49 Modelle sollen auch auf mobilen Plattformen ausführbar sein. A50 Automatisch grundlegende Properties anlegen. A51 Bereitstellung eines Gesamtübersichtscreens analog dem Audi Wizard. A52 Simulation der Ausführung von Modellen am Rechner. A53 Verfolgung der Ausführung graphisch im Modell. A54 Dynamisches Nachladen von Funktionen zur Laufzeit. A55 Prüfmechanismus, ob ein Modell valide ist. A56 Feedback der Nutzer direkt im Modell hinterlegen. A57 Komponenten können später noch aktualisiert werden oder neu hinzukommen. A58 Modelle sind plattformunabhängig. A59 Nutzung neuer, noch nicht implementierter Komponenten ermöglichen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Simulation	A60 Simulation von Diensten am Rechner ermöglichen. A61 Simulation hardwaregebundener Funktionen am Rechner ermöglichen (z.B. GPS). A62 Integration der Modellierung in ein Brainstorming. A63 Qualitätsmetriken für die Usability (Auswertung der Nutzerbedienschritte) neuer Prototypen integrieren und unterstützen. A64 Dummy-Schnittstellen für (noch) nicht verfügbare Komponenten.
Vorgehensmodell	A65 Anbindung/Portierung von iPhone Apps. A66 Berücksichtigung unterschiedlicher Perspektiven (Nutzer, Diensterbringen, ...). A67 Modelle sollen auch Grundlage für einen Serieneinsatz sein.
Nutzer	A68 Modelleirung und dazugehöriges Werkzeug sollen im Rahmen eines Wettbewerbs eingesetzt werde. A69 Berücksichtigung der unterschiedlichen Stakeholder-Kompetenzen in der Gestaltung der Modellierung.
Sprachsteuerung	A70 Modellierung der Sprachsteuerung ermöglichen.

Tabelle 4-1 Anforderungen aus der Praxis
Quelle: Eigene Darstellung auf der Grundlage des Innovationsworkshops (Anhang D)

4.4 Zusammenfassung

Das Problemfeld der Automotive Services sowie die, aus den in Abschnitt 3 diskutierten Aspekten der Gestaltung hervorgehenden Herausforderungen, lassen sich durch den Einsatz einer domänenspezifischen Modellierung gepaart mit einem partizipativen und iterativen Vorgehen im Rahmen des Design Thinking maßgeblich unterstützen.

Aus der Betrachtung der Grundlagen der Modellierung lässt sich ableiten, dass eine Erhöhung des Abstraktionsgrades, mit dem Automotive Services beschrieben werden, eine wesentliche Verbesserung bei deren Gestaltung nach sich zieht. Insbesondere werden auf diese Weise auch all jene Stakeholder, die nicht über technisches Vorwissen hinsichtlich einer Umsetzung verfügen, in die Gestaltung integriert. Eine domänenspezifische Modellierung besteht aus einer Modellierungssprache, einem Codegenerator sowie einem domänenspezifischen Framework (Tolvanen/Kelly 2004). Aus Sicht der Stakeholder, die diese Modellierung nutzen, bleibt eine graphische Beschreibung des Services sowie eine funktionale Umsetzung sichtbar.

Als grundlegende Gestaltungselemente für die Modellierungssprache ergeben sich drei wesentliche Aspekte: Konzepte, Notation und Regeln. Konzepte abstrahieren das relevante domänenspezifische Wissen in einer Form, die von allen Stakeholdern, auch ohne technisches Vorwissen, verstanden werden kann. Eine entsprechende Notation hilft dabei, diese Konzepte visuell darzustellen. Ein wichtiger Aspekt hierbei ist, dass Konzepte veränderlich sind und daher existierende Konzepte erweitert werden oder neue hinzu kommen können. Regeln beschreiben schließlich das Verhalten und die Interaktion einzelner Konzepte untereinander.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Ein nutzenstiftender Einsatz von Modellierung erfordert jedoch ein geeignetes Vorgehen. Die in sieben Elemente gegliederte Methode des Design Thinking liefert so einen Ansatz, da sie ihren Fokus auf der kreativen Gestaltung von Innovationen hat und durch ein interaktives und partizipatives Vorgehen eine enge Kopplung der Gestaltung des daraus resultierenden Services sicherstellt. Des Weiteren liefert eine Beachtung der Aktivitäten des Design Thinking grundlegender Anforderungen die eine Modellierungstechnik für Automotive Services berücksichtigen muss. Wie in Abschnitt 4.2.2 ersichtlich, konnten aus den sieben Elementen des Design Thinking drei Herausforderungen an eine entsprechende Modellierungssprache abgeleitet werden.

Schließlich wurden noch die direkt an diesem Projekt beteiligten Stakeholder hinsichtlich ihrer praktischen Anforderungen befragt. Neben den theoretischen und konzeptionellen Grundlagen für eine Modellierung von Automotive Services liefert das Erfahrungswissen diese Stakeholder wesentliche Details einer erfolgreichen, praktikablen Umsetzung. Dabei beziehen sich die resultierenden Anforderungen sowohl auf die zu gestaltende Modellierungstechnik als auch die dafür notwendige Werkzeugunterstützung.

Im nachfolgenden Abschnitt werden auf der Grundlage dieser drei Herausforderungen die wesentlichen Gestaltungsmerkmale einer Modellierungssprache für Automotive Services diskutiert und anhand der drei Aspekte eine domänenspezifische Modellierungssprache für Automotive Services entwickelt.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

5 Domänenspezifische Modellierung von Automotive Services

Im nachfolgenden Abschnitt wird auf der Grundlage der besprochenen Aspekte der Gestaltung von Automotive Services durch den Einsatz von Modellen, eine domänenspezifische Modellierungssprache für Automotive Services entwickelt. In Anlehnung an Tolvanen und Kelly (2004) fokussiert dieser Abschnitt auf den modellierenden Teil, die Ausgestaltung des Code Generators sowie des domänenspezifischen Frameworks findet sich im Anschluss in Abschnitt 6. Eine Beschreibung der Werkzeugunterstützung für die Modellierungssprache befindet sich ebenfalls im anschließenden Abschnitt 6.

Die Entwicklung der Modellierungssprache erfolgt in zwei Schritten: Zunächst werden die aus den Elementen des Design Thinking abgeleiteten Herausforderungen zur Definition der wesentlichen Designprinzipien herangezogen. Im Anschluss daran werden auf dieser Grundlage Gestaltungselemente des domänenspezifischen Modells, also die Konzepte, die Notation sowie die dazugehörigen Regeln entwickelt. Schließlich werden im Rahmen zweier Case Studies praktischer Beispiele für die Gestaltung von Automotive Services mit Hilfe der Modellierung gezeigt.

5.1 Automotive Services Modeling Language (ASML)

Da es sich bei der zu gestaltenden Modellierungssprache um eine domänenspezifische Modellierungssprache für Automotive Services handelt, wird diese im Anschluss als *Automotive Services Modeling Language (ASML)* bezeichnet. Die in dieser Sprache gestalteten Modelle tragen entsprechend den Namen *Automotive Service Model (ASM)*.

Das Ziel der Automotive Services Modeling Language ist es, Stakeholder in die Lage zu versetzen, ihre Visionen und Anforderungen zu explizieren ohne dafür technologisches Grundwissen zu benötigen. Darüber hinaus soll auf der Grundlage von ASML ein Prototyping dieser modellierten Services ermöglicht werden. Modelle können also ohne großen Aufwand in die Fahrzeugumgebung übertragen werden. Die Grundlage von ASML ist die Idee der Produktivierung bestehender Serviceideen, was in einer modularen Architektur von Servicefunktionen resultiert. Diese Architektur ermöglicht es Stakeholdern, den Umfang ihres angestrebten Automotive Services auf eine Reihe von vordefinierten Entitäten zu projizieren.

5.1.1 Designprinzipien

Gestaltungorientierte Forschung in der Wirtschaftsinformatik zielt auf die Lösung praktischer Probleme durch die Entwicklung innovativer und nützlicher IT Artefakte (Rosemann/Vessey 2008). Wenn es das Ziel eines Artefakts ist, nützliche Lösungen für praktische Probleme zu schaffen, sollte es das Ziel der gestaltungorientierten Forschung sein, präskriptives Wissen über die Lösung einer, dem Problem verwandten, Klasse von Problemen zu sammeln und zu verfeinern, d.h. Designprinzipien bereitzustellen (Baskerville 2008; Walls/Widmeyer/El Sawy 1992). Allerdings, so argumentiert Weber (1987), ist gestaltungorientierte Forschung subjektiv und anekdotisch und leidet unter der Anhäufung und Verfeinerung präskriptiven Wissens. Aus diesem Grund wurde theoriebasiertes Design vorgeschlagen, in dem Theorien als rechtfertigendes Wissen im Rahmen der Designentscheidungen herangezogen werden (Briggs 2006; Baskerville 2008).

Die Reduktion der gestaltungorientierten Forschung auf einen "hypothetisch-deduktiven" (Baskerville 2008) Forschungsansatz steht dabei jedoch im Konflikt mit der ursprünglichen

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Idee der Lösung praktischer Probleme, also relevanter und möglicherweise neuer Phänomene die bislang noch nicht effektiv angegangen wurden. Darüber hinaus würde dieser Forschungsansatz die zentrale Rolle der Kreativität in der Gestaltungorientierung unterdrücken (Aaen 2008; Markus/Majchrzak/Gasser 2002). Daher folgt diese Arbeit dem Verständnis von Briggs (2006), nachdem theoriebasiertes Design in innovativen Artefakte, die „seat-of-the-pants"-Ansätze wesentlich übertreffen, resultiert. Theoriebasiertes Design hilft dem Forscher, Designargumente zu verbessern und besonders innovative Designalternativen zu identifizieren.

Van Aken illustriert die Vorteile von theoriebasiertem Design: "… one can design an airplane wing on the basis of tested, technological (black-box) rules, but such wings can be designed much more efficiently grounded on the laws and insights of aerodynamic and mechanics" (van Aken 2004). In ihrer Arbeit schlagen Gehlert et al. (2009) einen methodischen Ansatz zur Begründung und Dokumentation von Designentscheidungen auf der Grundlage von Theorien vor, um die Intersubjektivität des Designartefaktes zu verbessern (Walls/Widmeyer/El Sawy 1992b).

Schermann et al. (2007) empfehlen die Entwicklung von kohärenten Gestaltungsprinzipien zu spezifischen Designfragen. Die Zerlegung von Problemen der gestaltungorientierten Forschung in kohärente Gesetzmäßigkeiten reduziert die Komplexität der Forschungsaufgabe. Darüber hinaus können Designprinzipien individuell durch das Testen der zur Verfügung gestellten Hypothesen bewertet werden. Referenzen zwischen den Designprinzipien helfen, den unveränderlichen Kern des Prinzips zu analysieren und bauen auf bestehenden Designprinzipien auf (Weber 1987, 13). Schließlich können Designprinzipien einzeln angewendet werden und reduziert den Aufwand des Lern- und Anpassungsprozesses. So kann der individuelle Nutzen eines Designprinzips leichter ermittelt werden. Ergebnisse der Evaluation von Designprinzipien resultieren in lokalen Änderungen der Prinzipien. Auf diese Weise erleichtern Designprinzipien das inkrementelle Verbessern der Kenntnisse über das Design. In Summe leiten theoriebasierte Designprinzipien den Forscher bei der Nutzung von Theorien zur Gestaltung innovativer und nützlicher Artefakte an. Insbesondere fördern theoriebasierte Designprinzipien die Begründung von Designargumenten und besonders innovativen Designalternativen.

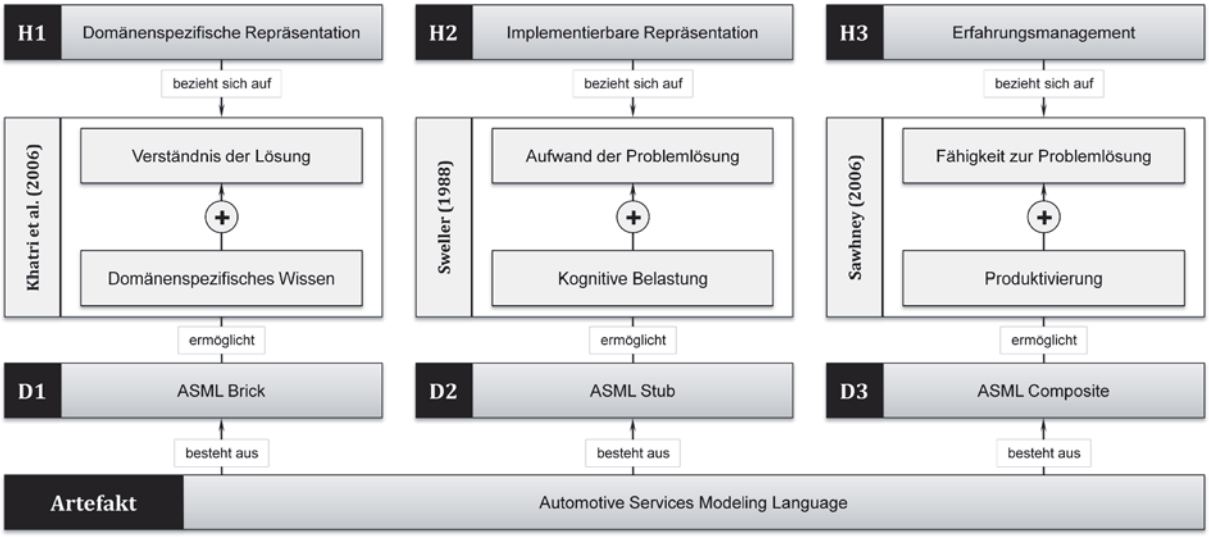


Abbildung 5-1 Designprinzipien der Automotive Service Modelling Language
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Abbildung 5-1 zeigt, wie ASML den Herausforderungen des Design Thinking mit drei entsprechenden Designprinzipien begegnet. Jede Herausforderung bezieht sich auf ein empirisch überprüfbares Wirkungskonstrukt, im Fall der Bereitstellung einer domainspezifischen Repräsentation ist dies das Konstrukt des „Verständnisses der Lösung", dass den Grad zu dem Stakeholder in der Lage sind einen bestimmten Lösungsvorschlag zu verstehen, beschreibt. Dem gegenüber wurden in der Literatur entsprechende Theorien identifiziert die mögliche passende Ursachekonstrukte beschreiben. So zeigen zum Beispiel Khatri et al. (2006), dass die Adaption von Domänenwissen die Fähigkeit zum Verständnis einer Lösungen positiv beeinflusst. Daher realisiert das Designprinzip „Brick" diese Ursache-Wirkung-Beziehung durch die Bereitstellung einer domainspezifischen Modellierungssprache.

Im Folgenden werden die drei Designprinzipien skizziert und deren Auswirkungen diskutiert. Zunächst werden die Designprinzipien dargestellt und anschließend anhand der identifizierten theoretischen Kenntnisse untermauert (Gregor 2006). Schließlich werden empirisch überprüfbare Hypothesen für diese Ursache-Wirkung-Beziehungen vorgeschlagen.

5.1.1.1 Brick

Das erste Designprinzip der Automotive Services Modelling Language sind Bricks. Sie sind die Grundbausteine eines Automotive Services. Bricks sind diskrete Entitäten die verschiedene Funktionen unterschiedlichster Granularität eines Automotive Services repräsentieren. So können diese Funktionen von der grafischen Darstellung einer Liste von Elementen bis hin zum Versenden einer E-Mail beschreiben. Innerhalb eines Modells dürfen dabei mehrere Bricks des gleichen Typs verwendet werden. Eine Ausnahme bilden hier die sogenannten Singleton Bricks. Singleton Bricks vertreten üblicherweise technische Elemente, wie z.B. eine GPS-Positionierung. Diese werden durch spezifische Hardwareelemente realisiert und können daher auch nur einmal instanziiert werden.

Bricks werden mit Hilfe von Streams miteinander verbunden. Ein Stream beschreibt den Kontrollfluss zwischen Bricks und verbindet immer genau zwei Bricks miteinander. Sobald ein Signal am Ausgang eines Bricks anliegt wird der Brick deaktiviert, das Signal an den nachfolgenden Brick weitergegeben und dieser schließlich aktiviert. Neben der Beschreibung des Kontrollflusses dienen Stream auch zur Steuerung des Datenflusses zwischen Bricks. Wird ein Brick durch einen Stream aktiviert, erstellt er eine Kopie des Streams und reichert diese mit seinen eigenen Ergebnissen an. Diese Kopie des originalen Stream wird am Ausgang für die Verbindung mit dem nächsten Brick verwendet. Durch dieses fortgesetzte Kopieren der Streams bleibt der ursprüngliche Stream unverändert und es kann eine parallele Ausführung ermöglicht werden, ohne dass es zu gegenseitigen Beeinflussungen kommt.

Das Wissen auf dem das Designprinzip Brick basiert, stammt aus der Arbeit von Khatri et al. (2006). Basierend auf der „cognitive fit" Theorie argumentieren sie, dass für das Lösen von Problemen zwei Arten von Wissen erforderlich sind. Auf der einen Seite sind dies technische Kenntnisse um Lösungsvorschläge hinsichtlich ihrer Machbarkeit zu überprüfen und anschließend eine Umsetzung einer konkreten Lösung vorzuschlagen. Auf der anderen Seite ist Fachwissen notwendig, um Visionen und Anforderungen der Stakeholder zu artikulieren. Üblicherweise besitzen Stakeholder über großes Fachwissen aber nur über wenig technisches Know-How. Khatri et al. (2006) zeigen, dass Stakeholder, die in der Lage sind ihre Ideen in Form von konzeptionellen Modellen zu explizieren, d.h. in der Lage sind ihr Fachwissen anzuwenden, ein besseres Verständnis der Herausforderungen des Designs und der

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Eigenschaften der vorgeschlagenen Lösung besitzen. Bricks weichen von traditionellen Modellierungskonstrukten wie Klassen, Methoden und Assoziationen ab und führen domainspezifische Konstrukte wie GPS-Positionierung, Kalender oder Sprachmemos ein. So sind Stakeholder in der Lage, ihre Vorstellungen von Automotive Services in Form von Tätigkeiten und Aufgaben im Fahrzeug zum Ausdruck zu bringen, ohne ein Verständnis der zugrunde liegenden Implementierung zu benötigen.

Die zu überprüfende Hypothese ist daher, ob Stakeholder mit geringen bis gar keinen technischen Kenntnissen der Automotive Domäne und ihrer IT-Komponenten in der Lage sind, realisierbare Automotive Services zu gestalten.

Hypothese 1: Stakeholder mit geringen bis gar keinen technischen Kenntnissen der Automotive Domäne und ihrer IT-Komponenten sind in der Lage, realisierbare Automotive Services zu gestalten.

Allerdings fordert dieses Designprinzip, welches Stakeholder in die Lage versetzt ihre Anforderungen in Form von domänenspezifischen Konstrukten zu spezifizieren, von technischen Stakeholdern, entsprechende IT-Artefakte zu Verfügung zu stellen, die den domänenspezifischen Anforderungen gerecht werden. Jedoch muss davon ausgegangen werden, dass technische Stakeholder über wenig bis gar kein domänenspezifisches Fachwissen verfügen. Mit dieser Herausforderung befasst sich das nächste Designprinzip.

5.1.1.2 Stub

Das zweite Designprinzip konzentriert sich auf die Umwandlung domainspezifischer Anforderungen in IT-Artefakte, die durch technische Stakeholder umgesetzt werden. Da anzunehmen ist, dass Automotive Services von Stakeholdern die über großes domainspezifisches Fachwissen und nur über geringes technisches Wissen verfügen definiert werden, ist es die Aufgabe von Stubs, nichttechnische Stakeholder im Prozess der Festlegung der erforderlichen Spezifikation für eine Umsetzung zu unterstützen. Umfassen Bricks also sowohl das domänenspezifische Fachwissen als auch die technologischen Grundlagen einer Umsetzung, so stellen Stubs lediglich das domänenspezifische Fachwissen dar.

Stubs basieren auf dem Gedanken der modellgetriebenen Entwicklung. Sie ermöglichen nichttechnischen Stakeholdern das Definieren der im letzten Designprinzip vorgestellten Bricks und insbesondere das ausgestalten deren spezifischen Eigenschaften. Anzumerken ist dabei, dass kein technisches Wissen für diese Aufgabe benötigt wird. Beispielsweise möchte ein Stakeholder einen neuen Baustein verwenden der es ermöglicht Sprachnachrichten aufzunehmen. Hierzu erstellt der Stakeholder ein neues Stub mit dem Namen "Sprachmemo" und definiert dessen Eigenschaften. Eine softwaretechnische Unterstützung dieser Aktivität wird in Abschnitt 6.2.1 vorgesellt. Ein Ergebnis dieses neuen Stubs wäre z.B. die aufgenommene Sprachnotiz. Im Prozess der Modellierung des Automotive Services stellt der Stakeholder fest, dass Sprachnotizen mit dem Ort der Aufnahme verbunden sein sollten. Er kehrt zur Definition des Stubs zurück und fügt den benötigten Input für die GPS-Ortung hinzu. Der so definierte Stub stellt eine Vorlage für die Realisierung durch die technischen experten im Domänenframework dar.

Die Grundlage für die Gestaltung von Stubs entstammt der “cognitive load“ Theorie von Sweller (1988). Die Theorie besagt, dass das Lernen durch die Art der Präsentation von Informationen verbessert werden kann. "The ease with which information may be processed

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

in working memory is a prime concern of cognitive load theory. Working memory load may be affected either by the intrinsic nature of the material (intrinsic cognitive load), or alternatively, by the manner in which the material is presented, or the activities required of students (extraneous cognitive load)" (Sweller 1988). Leider kann die intrinsische Natur des Wesens der technischen Details, die für eine Umsetzung von Automotive Services notwendig sind, nicht vereinfacht werden um nichttechnische Stakeholder leichter zu integrieren. Das gleiche gilt auch für die von den Stakeholdern erforderlichen Aktivitäten. Die Innovatoren von Automotive Services können den Aufwand, den Stakeholder in das Lernen investieren, nicht erhöhen; diese Variable kann also ebenfalls nicht beeinflusst werden.

Die Lösung für das “cognitive load“ Problem besteht also darin, die Anzahl der Elemente, mit denen sich Stakeholder befassen müssen, zu reduzieren und komplexe Informationen in das Langzeitgedächtnis zu verschieben. "Once a schema has been constructed, the interacting elements are incorporated within the schema and do not need to be considered individually within working memory. The schema can act as a single element in working memory and will impose minimal working memory demands, especially if it is automated….once constructed, this schema can act as an interacting element in higher order schemas" (Sweller 1988). Daraus folgt, dass Umsetzungsdetails einzelner Stubs vor den nichttechnischen Stakeholdern verborgen bleiben müssen.

Die zu überprüfende Hypothese ist daher, ob Stakeholder mit geringen bis gar keinen technischen Kenntnissen der Automotive Domäne und ihrer IT-Komponenten in der Lage sind, umsetzbare Anforderungen an eine Implementierung einzelner Komponenten zu definieren. Auf dieser Grundlage sollten technische Experten entworfene Stubs implementieren und so in vollständige Bricks umwandeln können.

Hypothese 2: Stakeholder mit geringen bis gar keinen technischen Kenntnissen der Automotive Domäne und ihrer IT-Komponenten sind in der Lage, umsetzbare Anforderungen an eine Implementierung einzelner Komponenten zu definieren.

Wie jedoch weiter oben erörtert, sollten Stakeholder ebenfalls in der Lage sein, bestehende Ideen und Designalternativen effektiv zu Nutzen sowie spätere Design Thinking Zyklen auf der Grundlage der Lösungen des aktuellen Zyklus zu ermöglichen. Das nächste und letzte Designprinzip der Automotive Services Modeling Language befasst sich daher mit dieser Herausforderung.

5.1.1.3 Composite

Das dritte Designprinzip konzentriert sich auf die Herausforderung eines Erfahrungsmanagements und der Ermöglichung zukünftiger Design Thinking Zyklen durch die Bereitstellung einer Methode für eine umfassende Artikulation von modelliertem domänenspezifischem Fachwissen.

Composites ermöglichen es Stakeholdern verschiedenste Bricks beliebig zu kombinieren und so eine noch umfangreichere Repräsentation von Automotive Services zu erhalten. Man nehme beispielsweise die oben diskutierte Sprachaufzeichnung. Stakeholder könnten den Sprachaufzeichnungs-Brick mit einem E-Mail-Brick kombinieren. So haben zukünftige Nutzer einen Baustein, mit dem Sie Sprachmemos aufzeichnen und per E-Mail an eine bestimmte E-Mail-Adresse verschicken können. Das entstehende Composite ermöglicht es

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

den entstandenen zusammengesetzten Service zu nutzen und sich auf weitergehenden wertschöpfenden Aspekten zu konzentrieren. Abbildung 5-2 zeigt die Grundidee eines Composites.

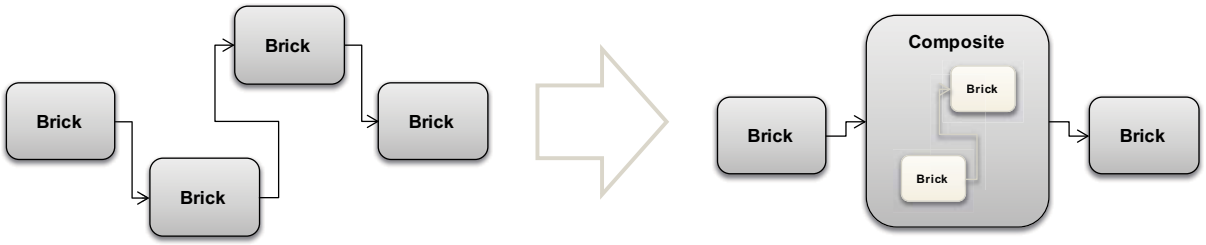


Abbildung 5-2 Designprinzip Composite
Quelle: Eigene Darstellung

Composites werden durch das Zusammenfassen mehrerer, durch Streams verbundener, Bricks und Stubs erstellt. Das entstandene Composite verhält sich aus der Sicht des Modellierers wie ein gewöhnlicher Brick, dessen Funktionalität durch das neu entstandene Teilmodell realisiert wird. Darüber hinaus ist es Möglich, Composites wiederum als Teil eines weiteren Composites zu nutzen. Da Composites wie gewöhnliche Bricks verwendet werden, müssen hier keine besonderen Aspekte beachtet werden.

Die Grundlage für Composites ist die Forschung zur Produktivierung von Services. Produktivierung von Services bedeutet in diesem Zusammenhang das Lernen aus verschiedenen existierenden individuellen Services (Sawhney 2006; Davies/Brady/Hobday 2007). Im Gegensatz zu bestehenden Entwicklungsprozessen für Services fordert Produktivierung die Analyse bestehender Lösungen und die Entwicklung neuer Services durch die Identifikation von Gemeinsamkeiten unterschiedlicher implementierter Lösungen, Prozesse und Strukturen.

In der Automotive Services Modeling Language erfüllen Composites genau diese Anforderung, indem sie komplexe, realisierte Aspekte der Lösung zusammenfassen und die darin enthaltenen Strukturen und Prozesse wiederverwendbar machen. Die daraus resultierenden produktiverten Automotive Services können von künftigen Stakeholdern genutzt werden, innovative Aspekte neuer Automotive Services zu adressieren, d.h. die Konzentration auf neue Aktivitäten und Funktionen im Fahrzeug. So steigert Produktivierung die Fähigkeit zur Problemlösung (Galbraith 2002).

Die zu überprüfende Hypothese ist daher, ob Stakeholder in der Lage sind existierende Elemente wieder zu verwenden, um neuen Funktionalitäten für künftige Nutzer zu definieren. Darüber hinaus sollten sich die Fähigkeiten der Automotive Services Modeling Language im Laufe der Zeit weiter entwickeln.

Hypothese 3: Stakeholder sind in der Lage, existierende Elemente wieder zu verwenden um neue Funktionalitäten für künftige Nutzer zu definieren und so die Fähigkeiten der Automotive Services Modeling Language zu erweitern.

5.1.2 Elemente des domänenspezifischen Modells

Die Automotive Services Modeling Language besteht demnach aus drei grundlegenden Gestaltungselementen: Bricks dienen dazu, domänenspezifisches Wissen in einer, von

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

technischen Grundlagen abstrahierten, Form zu repräsentieren und so verfügbar zu machen. Zu den Bricks gehören noch Streams, die funktionale und logische Übergänge von einem Brick zum nächsten darstellen. Schließlich erweitern Composites die Modellierungssprache im Sinne einer evolutionären Weiterentwicklung. Durch die Komposition komplexer Automotive Services werden neue Bausteine geschaffen, die in zukünftigen Services wieder verwendet werden können. Um mit der Fragestellung gänzlich neuer Herausforderung umzugehen, werden Stakeholder durch die Möglichkeit eigene Stubs zu definieren in die Lage versetzt, neue, domänenspezifische Konzepte zu nutzen.

Diese Designprinzipien bilden für sich alleine genommen allerdings noch keine domänenspezifische Modellierungssprache. Vielmehr ist es notwendig, eine geeignete Notation für eine graphische Repräsentation und Modellierung der einzelnen Prinzipien zu definieren sowie Regeln für deren Anwendung zu spezifizieren. Darüber hinaus müssen die domänenspezifischen Konzepte, die Bricks repräsentieren, definiert werden.

5.1.2.1 Notation

Die Notation einer Modellierungssprache definiert die syntaktische Repräsentation der durch die Sprache modellierten Konzepte. Im Falle der Automotive Services Modeling Language besteht diese Notation aus fünf Elementen: *Bricks* repräsentieren Konzepte der Anwendungsdomäne und sind mit einer diese realisierenden Implementierung im domänenspezifischen Framework hinterlegt. *Stubs* beschreiben ebenfalls ein Konzept der Anwendungsdomäne, sie sind allerdings im Gegensatz zu Brick noch nicht implementiert. *Composites* repräsentieren aggregierte Teilmodelle die wiederum in der Notation der Automotive Services Modeling Language definiert sind. Zu diesen drei Elementen, die die Designprinzipien von ASML realisieren, besteht die Notation noch aus zwei weiteren Bestandteilen. *Start / Autostart / Stopp* ist ein besonderer Baustein, der den Beginn und das Ende eines Modells symbolisiert. Schließlich repräsentieren *Streams* den Kontrolle- und Informationsfluss im Modell.

5.1.2.1.1 Brick

Bricks werden in ASML unabhängig ihrer Funktion oder des Konzepts welches sie repräsentieren stets gleich dargestellt. Dieser Grundsatz verhindert, dass sich Stakeholder einer Vielzahl an Elementen in der Notation gegenüber sehen die unterschiedliche Konzepte und Sachverhalte beschreiben. Natürlich müssen inhaltliche Unterschiede verschiedener Bricks in der Notation deutlich gekennzeichnet werden, um eine Verwechslung zu verhindern und ein schnelles Erfassen des modellierten Services zu gewährleisten. Der Grundbaustein gliedert sich daher in vier unterschiedliche Bereiche, die die einzelnen beschreibenden Attribute des Bricks beherbergt. Abbildung 5-3 gibt einen Überblick über die graphische Notation eines Bricks.

Das oberste Element ist der Titel des Bricks, der sich über dessen gesamte Breite erstreckt. Der Titel beinhaltet den Typ des Bricks, eine Versionsnummer sowie einen Index, mit dessen Hilfe mehrere Bricks des gleichen Typs innerhalb eines Modells unterschieden werden können. Darunter befindet sich ein Bereich für eine Vorschau auf das Konzept, welches durch den Brick repräsentiert wird. Dieser Bereich eines Bricks ist optional und erscheint nur bei solchen Bricks, bei denen eine Vorschau sinnvoll erscheint. Dies ist zum Beispiel bei Bedienkonzeptbausteinen, die eine graphische Anzeige zur Interaktion mit dem Nutzer

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

beschreiben, gegeben. Um eine einfache und übersichtliche Darstellung der Modelle zu ermöglichen, kann dieser Bereich vom Modellierer dynamisch auf- und zugeklappt werden.

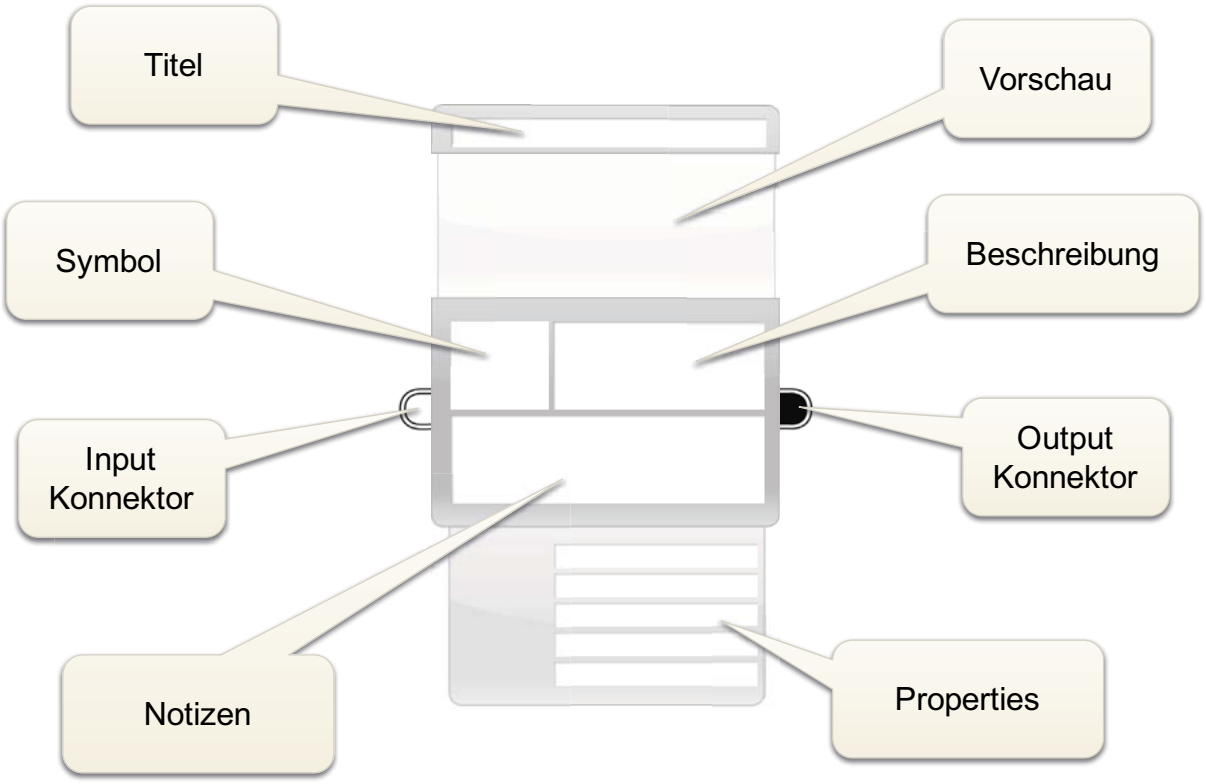


Abbildung 5-3 Notation Brick
Quelle: Eigene Darstellung

Im zentralen Bereich des Bricks befindet sich in der ersten Zeile, links oben, ein Feld für ein Symbol, das den Baustein visuell kennzeichnet. Für den vormals angesprochenen E-Mail-Brick ist dies zum Beispiel ein Bild eines Briefkuverts. Neben dem Symbol befindet sich ein Textfeld, welches eine Beschreibung des Bricks beinhaltet. Diese Beschreibung ist unabhängig vom aktuellen Modell und kann vom Modellierer nicht verändert werden. Für eigene Anmerkung und Notizen steht in jedem Brick ein Notizbereich bereit. Dieser befindet sich unterhalb des Symbols und der Beschreibung und erstreckt sich über die gesamte Breite des Bausteins. Hier können z.B. Beobachtungen, Anmerkungen oder Reaktionen von Nutzern hinterlegt werden.

Der untere Bereich eines Bricks dient der Darstellung von Properties. Properties beschreiben einzelne Parameter, mit denen die Funktionsweise eines Bricks beeinflusst werden kann. Beispielsweise kann hier die Systemfarbe eines Bedienkonzept-Bricks eingestellt werden. Analog zum Vorschaubereich, steht der Propertiesbereich nur in solchen Bricks zur Verfügung, die auch über Properties verfügen. Zusätzlich kann dieser Bereich ebenfalls vom Modellierer dynamisch auf- und zugeklappt werden, um eine übersichtliche Darstellung des Modells zu ermöglichen.

Links und rechts des Bricks befinden sich die so genannten Konnektoren. Konnektoren sind diskrete Eingangs- und Ausgangspunkte, über die Bricks durch die Verwendung von Streams untereinander verknüpft werden. Eingangskonnektoren werden stets auf der linken Seite des Bricks in Form eines nicht ausgefüllten Halbkreises dargestellt. Ein Eingangskonnektor repräsentiert eine Funktion, die ein Brick zur Verfügung stellt. Am Beispiel des E-Mail-Bricks wäre dies z.B. die Funktion „Sende E-Mail“. Informationen, die ein

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Eingangskonnektoren benötigt um die entsprechende Funktion aufzurufen (beispielsweise den Empfänger der E-Mail), werden im Rahmen der Streams übergeben. Nachdem ein Brick eine Funktion bearbeitet hat, signalisiert er dies durch ein Signal am Ausgangskonnektor, der auf der rechten Seite durch einen schwarz ausgefüllten Halbkreis dargestellt wird. Über diesen Konnektor werden auch die Ergebnisse des Bricks an den nachfolgenden Stream weitergegeben. Sowohl Eingangs- als auch Ausgangskonnektoren können in beliebiger Anzahl vorhanden sein und werden jeweils untereinander auf der entsprechenden Seite dargestellt. Da die einzelnen Bricks sehr spezifische Konzepte repräsentieren, sollte die Anzahl der Konnektoren nach Möglichkeit klein gehalten werden. Eine große Anzahl an Konnektoren indiziert die Notwendigkeit einen neuen Brick zu definieren.

Abbildung 5-4 zeigt die Darstellung eines Karten-Brick, der das Bedienkonzept der Darstellung einer Landkarte und Anzeige von Informationen auf dieser Karte repräsentiert. Von links nach rechts wird der Brick in den vier Darstellungsvarianten dargestellt, die sich aus dem Auf- und Zuklappen des Vorschau- und Propertiesbereichs ergeben. An diesem Beispiel werden nachfolgend die Gestaltungselemente eines Bricks in der Automotive Services Modeling Language, wie Sie Abbildung 5-3 dargestellt werden, zusammengefasst.

Abbildung 5-4 Notation Brick am Beispiel "Karten"
Quelle: Eigene Darstellung

5.1.2.1.1.1 Titel

Der Titel eines Bricks besteht aus einem, für den Typ des Bricks eindeutigen Namen, einer optionalen Versionsnummer sowie einem, in runden Klammern angegebenen, Index. Existiert nur eine Version des Bausteins, so entfällt, wie in Abbildung 5-4 zu sehen, die optionale Versionsnummer. Der Index identifiziert ein bestimmtes Brick innerhalb eines Modells. Wird beispielsweise ein Karten-Brick mehrmals innerhalb eines Modells verwendet, können diese Instanzen durch den Index voneinander unterschieden werden.

5.1.2.1.1.2 Vorschau

Die Vorschau ist ein optionales Element, welches nur in solchen Bricks verfügbar ist für die eine Vorschau sinnvoll ist. Dies ist beispielsweise für den gezeigten Karten-Brick zutreffend. Die Vorschau ist in diesem Fall ein Screenshot der zeigt, wie sich dieser Brick einem Nutzer gegenüber im Fahrzeug darstellt.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

5.1.2.1.1.3 Symbol

Das Symbol ermöglicht eine graphische Identifikation einzelner Typen von Bricks. Jeder Brick hat also ein eindeutiges Symbol, welches, analog zu dessen Namen, eine visuell Identifikation ermöglicht. Symbole können ebenfalls verwendet werden einzelne Bricks, zum Beispiel in einer Liste, zu repräsentieren. Am Beispiel des Karten-Brick wird eine stilisierte Weltkarte als Symbole genutzt.

5.1.2.1.1.4 Beschreibung

Die Beschreibung eines Bricks dient dazu, dass dahinterliegende Konzept textuell zu erläutern. Die darin enthaltenen Informationen können von einer einfachen Feststellung der Grundfunktion des Bricks, wie am Beispiel des Karten-Bricks *„Baustein zur Darstellung von Landkarten“*, bis hin zu Hinweis auf spezifische Details reichen.

5.1.2.1.1.5 Notizen

Mithilfe von Notizen können Stakeholder beliebige Informationen, die sich auf einen Automotive Service beziehen, im Modell hinterlegen. Notizen haben keine semantische Bedeutung hinsichtlich des domänenspezifischen Modells und dienen allein der Kommunikation mit anderen Stakeholdern. Im Karten-Brick Beispiel in Abbildung 5-4 wird der Zweck und die Funktion des Karten-Bricks im speziellen Modell beschrieben: *„Stellt die aktuellen Termine als Points of Interest auf der Landkarte dar.“*

5.1.2.1.1.6 Properties

Durch Properties kann die Funktionsweise eines Bricks individuell an die Gegebenheiten im Modell angepasst werden. Im Karten-Brick sind dies fünf Properties: der Zoom-Faktor der Karte, also der Maßstab in dem die Karte angezeigt werden soll, der Kartenprovider (es stehen Google-Maps und OpenStreetMaps zur Verfügung), der Titel sowie der Untertitel und eine Liste von Aktionen, die der Nutzer aktivieren kann.

Das Kartenbeispiel stellt auch den einzigen Sonderfall einer Property dar. Üblicherweise beziehen sich Properties nur auf die inhaltlichen Funktionen eines Bricks und nicht auf dessen Repräsentation im Modell. Eine Ausnahme bildet hier das Property „Actions“. Aktionen die angegeben werden (in Form einer Komma separierten Liste), stehen dem Modellierer direkt als Ausgangskonnektoren des Bricks zur Verfügung. Mit diesem Property kann also der Funktionsumfang des Bricks beeinflusst werden.

5.1.2.1.1.7 Input Konnektor

Inputkonnektoren repräsentieren die Funktionen die ein Brick zur Verfügung stellt. Am Beispiel des Karten-Bricks sind dies drei Inputkonnektoren: *„Zentriert die Karte auf eine bestimmte Position“, „Stellt Points of Interest grafisch auf der Karte dar“, „Stellt eine Route als Linie grafisch auf der Karte dar“*. Informationen, die Inputkonnektoren benötigen, werden zusammen mit dem eingehenden Stream übergeben.

5.1.2.1.1.8 Output Konnektor

Outputkonnektoren signalisieren das Ende einer Funktion oder eine Eingabe des Nutzers auf die durch einen Übergang zum nächsten Brick reagiert werden soll. Sobald ein Signal an einem Ausgangskonnektor anliegt, wird der aktuelle Brick beendet, dessen Informationen auf den ausgehenden Stream übertragen und der nachfolgende Brick aktiviert. Im Karten-Brick

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Beispiel steht die vom Nutzer ausgelöste Aktion „Zurück“ als Outputkonnektor zur Verfügung.

5.1.2.1.2 Stub

Aus Sicht des Modellierers werden Stubs wie gewöhnliche Bricks verwendet. Sie verfügen also ebenfalls über einen Titel, einen Hauptbereich in dem ein Symbol, eine Beschreibung sowie ein Bereich für Notizen vorhanden ist, sowie einen vom Nutzer auf- und zuklappbarer Propertiesbereich am unteren Ende des Stubs. Eine Vorschau auf die Darstellung des durch den Stub beschriebenen Konzepts ist aufgrund der mangelnden Implementierung im domänenspezifischen Framework nicht vorhanden. Ein Stub kann also, wie in Abbildung 5-5 zu sehen, in den beiden Varianten mit Properties und ohne Properties angezeigt werden. Ebenfalls analog der Darstellung eines Bricks werden Eingangskonnektoren durch nicht ausgefüllte Halbkreise auf der linken Seite und Ausgangskonnektoren als schwarz ausgefüllten Halbkreis am rechten Rand des Stubs dargestellt.

Abbildung 5-5 Notation Stub am Beispiel "Karten v2"
Quelle: Eigene Darstellung

Stubs können vom Modellierer nur durch ihr Symbol von gewöhnlichen Bricks unterschieden werden. Ein aufgeschlagenes Buch mit einer darüber befindlichen Lupe symbolisiert den nicht implementierten Charakter eines Stubs. Stubs können vom Nutzer auf zwei Arten erzeugt werden: Entweder ein bestehendes Brick oder ein bestehender Stub wird um neue Funktionalität erweitert, oder ein gänzlich neues Element wird angelegt. Im ersten Fall kann der Name des Stubs nicht verändert werden und eine Versionsnummer wird zwischen Titel und Index eingefügt (vgl. Abbildung 5-5). Anzumerken ist, dass hier keine hierarchische Struktur im Sinne einer Vererbung angelegt wird sondern der alte Baustein lediglich als Vorlage für den neuen Baustein kopiert wird. Es ist also nicht nur möglich, Eigenschaften des alten Bausteins zu Verändern und neue Hinzuzufügen, sondern auch, beispielsweise Eingangskonnektoren, zu entfernen. Eine Rückwärtskompatibilität von Stubs und Bricks ist somit nicht zwingend gegeben.

Für das Anlegen von Stubs steht dem Nutzer ein spezieller Wizard, wie in Abschnitt 6.2.1 beschrieben, zur Verfügung. Die in diesem Wizard erstellten neuen Elemente stehen einerseits direkt in der Modellierung als neue Bausteine zur Verfügung und werden andererseits, in Form einer formalisierten textuellen Beschreibung, den Entwicklern zur Umsetzung im domänenspezifischen Framework zur Verfügung gestellt. Sobald für einen Stub eine passende Implementierung existiert, wird dieser automatisch zu einem vollständigen Brick, ohne dass

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

weitere Anpassungen notwendig sind. Aus Sicht des Modellierers ändert sich also lediglich das Symbol des Stubs und er erhält, je nach Implementierung, optional eine zusätzliche Vorschau. Existierende Modelle, in denen das Stub Verwendung findet, müssen nicht angepasst werden.

5.1.2.1.3 Composite

Wie auch Stubs werden Composites, wie in Abbildung 5-6 ersichtlich, in Form von, der Darstellung von Bricks analogen, Bausteinen dargestellt. Ein Composite verfügt über einen Titel, einen Hauptbereich sowie Eingangs- und Ausgangskonnektoren an den Seiten des Bausteins. Die Informationen im Titel beschränken sich auf den Namen des Composite, der gleichzeitig der Name des dahinterliegenden Modells ist, sowie einem Index zur Unterscheidung mehrere Instanzen. Eine Versionierung von Composite ist nicht möglich und wird diese im Titel auch nicht angezeigt. Im Hauptbereich verfügt ein Composite, analog zu Bricks und Stubs, über ein Symbol, eine Beschreibung sowie einem Bereich für Notizen. Aufgrund der Tatsache, dass Composite nicht durch eine direkte Implementierung im domänenspezifischen Framework sondern über ein eigenes Modell definiert werden, steht für sie weder eine Vorschau noch ein Bereich für Properties zur Verfügung. Beide Elemente können dem, dass Composite beschreibenden, Model entnommen werden.

Die einzige Möglichkeit, ein Composite von einem Stub oder einem Brick zu unterscheiden, ist, wie auch bei Stubs, das Symbol. Eine stilisierte Zeichnung auf einer weißen Zeichenfläche symbolisiert dabei den definierenden Charakter des dahinterliegenden Modells (vgl. Abbildung 5-6). In der, in Abschnitt 6 beschriebenen, Modellierungsumgebung kann dieses Modell durch einen Doppelklick auf das Symbol geöffnet werden.

Abbildung 5-6 Notation Composite
Quelle: Eigene Darstellung

Eine Besonderheit bei Composites sind die Konnektoren. Es sind zwar sowohl Eingangs- als auch Ausgangskonnektoren vorhanden, diese werden aber in ihrer Funktionsweise gänzlich unterschiedlich denen der Stubs und Bricks behandelt. Eine Beschreibung dieses Verhaltens findet sich im Rahmen der Regeln im Abschnitt 5.1.2.2. Zunächst stehen in einem Composite alle Eingangs- und Ausgangskonnektoren aller Elemente des darunter liegenden Modells zur Verfügung. Diese muss der Modellierer jedoch zunächst auswählen, bevor diese angezeigt werden. Ein neues Composite besitzt also initial keine Konnektoren in der Darstellung. Eine Auswahl von Konnektoren erfolgt über einen Doppelklick auf den Titel des Bausteins.

5.1.2.1.4 Start / Autostart / Stop

Die letzten zur Gestaltung von Modellen verfügbaren Bausteine in der Automotive Services Modeling Language sind die Symbole für Start, Autostart und Stopp. Streng genommen

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

handelt es sich bei diesen Bausteinen ebenfalls um Bricks, die durch Konnektoren über Streams mit anderen Bricks verbunden werden. Da diese drei Elemente allerdings semantisch eine besondere Bedeutung besitzen, werden sie im Modell durch eigene Symbole dargestellt. Wie in Abbildung 5-7 zu sehen, werden (von links nach rechts) Start, Autostart und Stopp als Kreise mit doppeltem Rand angezeigt.

Der Bereich zwischen dem Doppelrand ist der Konnektor. Start und Autostart verfügen lediglich über einen Ausgangskonnektor, da diese Bausteine den Beginn des Modells symbolisieren und somit nicht über Streams aufgerufen werden können. Entsprechend verfügt das Stopp-Element nur über einen Eingangskonnektor, da das Erreichen dieses Elements das Modell beendet und keine weiteren Streams möglich sind.

Abbildung 5-7 Notation Start / Autostart / Stop
Quelle: Eigene Darstellung

5.1.2.1.5 Stream

Neben den Bausteinen die für die einzelnen domänenspezifischen Konzepte stehen, stehen in ASML Streams zur Beschreibung des Kontroll- und Informationsflusses im Modell zur Verfügung. Streams werden, wie in Abbildung 5-8 dargestellt, als Pfeile die jeweils von einem Ausgangskonnektor zu einem Eingangskonnektor reichen symbolisiert. Zum Anlegen eines Streams in der Modellierungsumgebung muss der Nutzer lediglich auf einen Ausgangskonnektor klicken und diesen per Drag'n'Drop mit einem Eingangskonnektor verbinden. Streams werden automatisch um Modellierungsbausteine herum geroutet und dürfen diese nicht schneiden.

Abbildung 5-8 Notation Stream
Quelle: Eigene Darstellung

Für die Darstellung der Validität eines Streams stehen verschiedene Darstellungsvarianten und Farbkodierungen zur Verfügung. Zunächst wird zwischen synchronen und asynchronen Streams unterschieden. Erstere werden, wie in Abbildung 5-8 links zu sehen, mit einer durchgezogenen Linie und letztere (rechts) durch eine gestrichelte Linie dargestellt. Zusätzlich weist ein Gelb markierter synchroner Stream auf eine fehlerhafte Verknüpfung der Informationen des Eingangskonnektors mit Informationen des Streams hin, ein roter asynchrone Stream auf einen grundsätzlichen semantischen Fehler im Modell. Dies kann einerseits ebenfalls eine fehlerhafte Verknüpfung der Informationen sein, andererseits aber auch ein Verstoß gegen die in Abschnitt 5.1.2.2.3 beschriebenen Regeln für asynchrone Streams. Anzumerken ist, dass diese kennzeichnende Validierung lediglich ein Hinweis auf besonders schwere Fehler ist und eine schwarze Darstellung des Streams keine formal Gültigkeit des Modells garantiert. Darüber hinaus werden Nutzer nicht an einer regelwidrigen

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Modellierung in ASML gehindert. Erkannte Fehler werden lediglich als solche gekennzeichnet, so dass der Modellierer im Rahmen seiner eigenen Vorstellungen auf diese reagieren kann.

5.1.2.2 Regeln

Neben der syntaktischen Darstellung und semantischen Bedeutung der im letzten Abschnitt beschriebenen Gestaltungselemente der Automotive Services Modeling Language fehlen noch die für eine domänenspezifische Modellierungssprache notwendigen Regeln. Neben den grundlegenden Aspekten, wie diese Elemente im Rahmen von Modellen zu einem Automotive Service Model zusammengesetzt werden können, muss zusätzlich die Bedeutung der einzelnen Gestaltungselemente und die damit modellierte Funktionsweise eines Automotive Services festgelegt werden. Aus diesem Grund werden die Gestaltungselemente der beschriebenen Notation nochmals hinsichtlich der dahinter stehenden Regeln besprochen. Stubs werden in diesem Zusammenhang mit Bricks zusammengefasst, da sie das gleiche Konzept wie diese beschreiben und sich lediglich durch eine fehlende Implementierung im domänenspezifischen Framework von Bricks unterscheiden.

5.1.2.2.1 Brick

Aus funktionaler Sicht des Modells stellen Bricks atomare Elemente dar, deren Ausführung vom Aufruf des Bricks durch den Eingangskonnektor bis hin zum Beenden des Bricks durch den Ausgangskonnektor aus Sicht der Modellierung transparent ist. Sie werden dabei in einem eigenen Prozess ausgeführt, so dass das restliche Modell und somit andere Gestaltungselemente nicht durch einen einzelnen Blick blockiert werden können. Aus Sicht des Modells können Bricks also nur über die Eingangskonnektoren und Properties beeinflusst werden.

Die im Modell spezifizierten Properties werden einem Brick beim erstmaligen Aktivieren durch einen Stream übergeben. Sie dienen der grundsätzlichen Konfiguration des Bricks und sind für jedes Brick individuell. Existieren innerhalb eines Modells mehrere Instanzen des gleichen Bricks, so werden diese auch in der Ausführung als unabhängige Instanzen realisiert. Jede dieser Instanzen hat somit ihre eigene Properties, entsprechend den Angaben im Modell. Wird ein Brick im Zuge der Ausführung des Modells zum wiederholten Mal aufgerufen, bleiben die alten Properties weiterhin gültig und das Brick befindet sich in dem internen Zustand, den es nach der letzten Aktivierung besaß.

Eine Ausnahme bilden die so genannten Singleton Bricks. Singleton Bricks sind eine besondere Form von Bricks, die für einmalig existierende Konzepte stehen. Ein Beispiel für ein solches, einmalig existierendes Konzept ist die GPS-Positionierung im Fahrzeug. Da hinter diesem Konzept eine singulär verfügbare Hardwarekomponenten steht, kann diese folglicherweise auch nicht mehrfach im Modell instanziiert werden. Trotzdem unterscheiden sich diese Bricks im Modell nicht von gewöhnlichen Bricks. Lediglich in ihrer Ausführung werden sie in einigen Punkten unterschiedlich behandelt. Zunächst werden Singleton Bricks nur durch eine gemeinsame Instanz, die in einem eigenen Prozess läuft, realisiert. Zusätzlich werden jedes Mal, wenn ein Singleton Brick durch einen Stream aufgerufen wird, die im Modell hinterlegten Properties aktualisiert. So können auch Singleton Bricks im Modell an verschiedenen Stellen unterschiedliche Properties besitzen, die in der Ausführung entsprechende Berücksichtigung finden.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

5.1.2.2.2 Composite

Wird ein Composite durch einen Stream an einem bestimmten Konnektor aktiviert, so wird diese Aktivierung und die im Stream enthaltenen Informationen an den originären Baustein des Konnektors transparent weitergegeben und dieser aktiviert. Die weitere Ausführung des Modells erfolgt nun im, vom Composite gekapselten, Modell, bis ein Stream dort wiederum einen Konnektor erreicht der vom Nutzer im Composite ausgewählt wurde. Der Stream wird nun nicht mehr im beschreibenden Modell weitergeführt, sondern die Ausführung wieder eine Ebene nach oben auf den Ausgang des Composite geführt.

Man kann sich dieses Vorgehen in Form von übereinander geschichteten Ebenen vorstellen. Auf diese Weise ist eine, sowohl für die Modellierung als auch für eine spätere Ausführung, vollständig transparente Staffelung einer beliebigen Anzahl an Ebenen, also an Composites, möglich. Eine Einschränkung hierbei ist jedoch die Validität der eingehenden und ausgehenden Streams. Die durch das Composite beschriebenen Teilmodelle müssen in sich vollständige und gültige Modelle sein, die nicht auf Informationen des übergeordneten Modells angewiesen sind. Besonderes Augenmerk muss auch auf Singleton Bricks innerhalb eines Composites gelegt werden. Aufgrund des einmaligen Charakters dieser Bausteine über Modellgrenzen hinweg, kann es zu einem unvorhergesehenen Verhalten in der Ausführung kommen.

5.1.2.2.3 Stream

Neben den atomaren, funktionalen Gestaltungselementen wird die Logik eines Modells im Wesentlichen über dessen Streams definiert. So legen diese einerseits die Reihenfolge der anzuwendenden domänenspezifischen Konzepte fest und definieren andererseits auch den Informationsfluss zwischen diesen Bausteinen. Abhängig von den verknüpften Informationen kann die Funktionsweise und Teils auch die Bedeutung einzelner Bricks inhaltlich stark variieren.

Wie in der Beschreibung der Notation von Streams in Abschnitt 5.1.2.1.5 bereits angedeutet, wird grundsätzlich zwischen synchronen und asynchrone Streams unterschieden. Der wesentliche Unterschied ist hier, dass nach einem synchronen Stream das ausgehende Brick deaktiviert wird und nur noch das neue Brick seine Funktion ausführt. Synchrone Streams stellen also einen einfachen Kontrollfluss von einem Baustein zum nächsten dar. Asynchrone Streams hingegen ermöglichen eine parallele Ausführung von Bricks. Nach einem asynchronen Stream sind also sowohl der ausgehende Brick als auch der nachfolgende Brick aktiv und beide können Signale an ihren jeweiligen Ausgangskonnektoren hervorbringen. Durch das Prinzip des jeweiligen Kopierens des alten Streams beim Erreichen eines Bricks und der Ausführung einzelner Bricks in jeweils eigenen Prozessen, stehen nach einem asynchronen Stream zwei parallele Ausführungen zur Verfügung, die vollständig unabhängig voneinander sind. Ein Beispiel für den Einsatz eines asynchronen Streams ist die Steuerung eines Audioplayers durch einen entsprechenden Bedienkonzeptbaustein (vgl. Abschnitt 5.1.2.3.1). Durch den asynchronen Aufruf des Audioplayers bleibt die Anzeige und Steuerung im Rahmen des Bedienkonzeptbausteins definiert und ist für den Nutzer weiterhin aktiv.

Eine Besonderheit, die sowohl bei synchronen als auch bei asynchronen Streams beachtet werden muss, sind Bricks die eine Benutzeroberfläche repräsentieren. Beschreibt ein Stream einen Übergang von einem Brick mit Benutzeroberfläche hin zu einem Brick ohne Benutzeroberfläche, darf die Darstellung nicht einfach „schwarz“ werden. Bei einem synchronen Aufruf wird der alte Brick inaktiv, bleibt aber weiterhin sichtbar. Erst wenn der

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Kontrollfluss ein weiteres Brick mit einer Benutzeroberfläche erreicht, wird das alte Brick unsichtbar und das neue angezeigt. Bei asynchronen Streams hingegen muss unterschieden werden, wie viele Anzeigen zur Verfügung stehen. Existiert nur eine Anzeige im Fahrzeug, sind asynchrone Streams nur erlaubt, wenn im nachfolgenden Modell kein Brick mit einer Benutzeroberfläche folgt. Im Falle mehrerer Anzeigen ist dies, entsprechend der Anzahl der verfügbaren Anzeigen, möglich.

Neben dem Kontrollfluss übernehmen Streams auch die Aufgabe den Informationsfluss zwischen Bricks zu steuern. Hierzu merkt sich ein Stream alle Informationen die am entsprechenden Ausgangskonnektor des ausgehenden Bricks zur Verfügung gestellt werden. Eine eindeutige ID ermöglicht eine Identifikation der Quelle der jeweiligen Information. Zusätzlich verfügt jeder Stream, wie bereits erwähnt, über alle Informationen der vorausgegangenen Streams. Liegt am Ausgangskonnektor eine Information mit einer ID vor, die im Stream bereits vorhanden ist, so wie diese durch die neue Information überschrieben und die alte steht nicht mehr zur Verfügung. Am Eingangskonnektor eines Bricks werden die Informationen des Streams mit den Parametern, die der Eingangskonnektor benötigt, verknüpft. Hier können alle, im Stream verfügbaren Informationen aller vorausgegangenen Bricks verwendet werden. Ist diese Verknüpfung unvollständig oder werden inkompatible Datentypen zugeordnet, wird dies im Modell, entsprechend der Notation, farblich dargestellt.

5.1.2.2.4 Start / Autostart / Stop

Die Bedeutung der drei Elemente Start, Autostart und Stopp lässt sich leicht erahnen. Start und Autostart sind jeweils die Einstiegspunkte in die Ausführung des Modells. Ein Erreichen des Stoppelements beendet dieses. Die Sonderform Autostart des Startelementes indiziert, dass das Modell in der Ausführungsumgebung ohne Zutun des Nutzers automatisch gestartet werden soll. Sowohl Start als auch Autostart dürfen im Modell insgesamt nur einmal vorhanden sein. Modelle die kein Start und Autostart Element enthalten, können nicht allein ausgeführt werden, stehen aber in Form von Composites in anderen Modellen zur Verfügung.

Das Stopp Element hingegen darf beliebig oft im Modell vorkommen. Semantisch bedeutet das Erreichen des Stoppelements, dass das gesamte Modell, inklusive aller synchronen und asynchronen Pfade, beendet wird. Dies gilt auch für die Teilmodelle, die durch Composites integriert sind. Es ist daher anzuraten, Stoppelemente nicht in Composites zu verwenden, da diese eventuell zu einem unbeabsichtigten Ende der Ausführung führen können. Eine Verwendung des Stoppelements in Composites ist jedoch sowohl syntaktisch als auch semantisch erlaubt.

5.1.2.3 Konzepte

Neben einer geeigneten Notation und den dazugehörigen Regeln sind die domänenspezifischen Konzepte der dritte, und wesentliche Aspekte einer domänenspezifischen Modellierungssprache. Zwar müssen auch, wie weiter oben diskutiert, die Notation sowie die Regeln den speziellen Eigenschaften der Domäne angepasst sein, doch sind es erst die Konzepte, die die notwendige Erhöhung des Abstraktionsgrades der Modellierungskonstrukte realisiert.

Aus der Betrachtung des Stands der Technik in Abschnitt 2 sowie der Analyse der bestehenden Infotainmentsysteme der Hersteller Audi, BMW und Mercedes Benz in Abschnitt 3.5.1 lassen sich zwei grundsätzliche Arten von Konzept für Automotive Services identifizieren: Bedienkonzepte und Funktionskonzepte.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Auf der einen Seite gibt es Bedienkonzepte zur Interaktion mit dem Nutzer. Diese beinhalten sowohl das Wissen und die Erfahrungen die eine geeignete Gestaltung und Darstellung von Informationen im Fahrzeug sicherstellen und andererseits die notwendigen Grundlagen, die eine Interaktion der Nutzer erst ermöglichen. Es zeigt sich, dass für die Darstellung von Automotive Services keine Vielzahl an unterschiedlichen Bedienkonzepten notwendig ist, sondern diese lediglich aus wenigen universellen Elementen besteht (vgl. Abschnitt 3.5.1).

Auf der anderen Seite gibt es noch Funktionskonzepte. Funktionskonzepte zeichnen sich dadurch aus, dass sie vom Nutzer eines Automotive Services nicht visuell oder haptisch wahrgenommen werden können, sondern jeweils eine spezielle Funktionalität realisieren. Ein Beispiel für ein Funktionskonzept ist die GPS-Positionierung oder eine Sprachsynthese. Beide Konzepte verfügen weder über eine Anzeige am Bildschirm noch können sie vom Nutzer über Funktionsknöpfe oder einen Dreh-Drück-Steller direkt gesteuert werden.

Ein Funktionskonzept, das in dieser Definition aus dem Rahmen fällt, ist die Spracheingabe. Sie ist auf der einen Seite ein Funktionskonzept, da sie weder über eine graphische Anzeige noch über eine Manipulationsmöglichkeit über Funktionstasten oder Dreh-Drück-Steller verfügt, auf der anderen Seite aber auch ein Bedienkonzept, da sie eine direkte Interaktion mit dem Nutzer repräsentiert. Aus diesem Grund wird die Definition eines Bedienkonzepts dahingehend erweitert, dass Bedienkonzepte zwingend eine graphische Repräsentation besitzen müssen. Die Spracheingabe zählt also demnach zu den Funktionskonzepten.

Auch wenn sich aus den Betrachtungen der vorangegangen Abschnitte eine Vielzahl an Konzepten ableiten lassen, werde im Nachgang nur einige wenige Beispiele exemplarisch erläutert, die im Rahmen dieser Arbeit zum Zwecke der Evaluation im domänenspezifischen Framework realisiert wurden. Es ist die originäre Aufgabe der Stakeholder, ihr domänenspezifisches Wissen als Konzepte zur Verfügung zu stellen. Die vorgestellten Konzepte bilden somit nur einen Kern der Modellierung, der durch die Definition neuer Konzepte, und damit neuer Bricks, stetig erweitert werden kann.

5.1.2.3.1 Bedienkonzepte

Bedienkonzepte beschreiben Konzepte, die die graphische Interaktion der Nutzer mit einem Automotive Service realisieren.

5.1.2.3.1.1 Liste

Das Bedienkonzept Liste repräsentiert die Darstellung und Interaktion einer Menge an Elementen auf dem Bildschirm. Eine Liste verfügt über einen Eingangskonnektor, bei dessen Aktivierung die übergebenen Parameter angezeigt und wenn angegeben, ein bestimmtes Element ausgewählt wird. Als Ausgangskonnektor stehen dem Modellierer zum einen die Signalisierung der Selektion eines Listenelements durch den Nutzer (Drehen des Dreh-Drück-Stellers) sowie die Aktivieren eines Listenelements (Drücken des Dreh-Drück-Stellers) zur Verfügung. Als Ergebnis werden jeweils die Position sowie der Inhalt des selektierten Elements an den nächsten Stream übergeben. Parametrisieren lässt sich eine Liste durch Angabe des Titels, dass Untertitels, einer Liste von Elementen sowie einer Liste von Aktionen. Die Liste der Elemente stellt Standardelemente dar, die durch die Parameter des Eingangskonnektors überschrieben werden.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

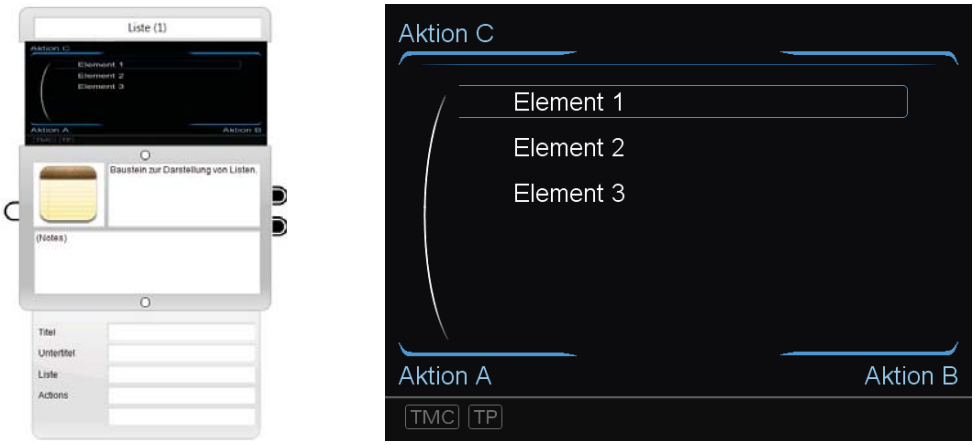


Abbildung 5-9 Bedienkonzept Liste
Quelle: Eigene Darstellung

5.1.2.3.1.2 Text

Das Bedienkonzept Text wird zur Darstellung von Fließtext am Display genutzt. Analog zum letzten Konzept stehen als Parameter ein Titel, einen Untertitel ein initialer Text, der durch den Parameter des Eingangskonnektors überschrieben wird, sowie Aktionen bereit. Neben diesen Aktionen hat das Konzept keinen Ausgangskonnektor. Als Eingangskonnektor steht die Funktion *„Stellt einen Text dar„* zu Verfügung, die als Parameter den Text sowie den Untertitel der Anzeige entgegennehmen kann. Ist der anzuzeigende Text größer als die zur Verfügung stehende Darstellung, kann durch Drehen des Dreh-Drück-Stellers nach Oben und Unten gescrollt werden.

Abbildung 5-10 Bedienkonzept Text
Quelle: Eigene Darstellung

5.1.2.3.1.3 Browser

Das dritte realisierte Bedienkonzept ist ein Browser. Mithilfe des Browsers können beliebige Internetseiten im Fahrzeug dargestellt werden. Funktional wird der Browser durch den Browser Chrome realisiert und verfügt somit über dessen Funktionalität hinsichtlich der Darstellung von Inhalt und der Nutzung von Plugins. So ist beispielsweise die Darstellung von Adobe Flash im Fahrzeug möglich. Als Parameter zur Konfiguration stehen ein Titel, einen Untertitel sowie eine anzuzeigende Internetadresse zur Verfügung. Das Konzept verfügt des Weiteren über einen Eingangskonnektor, der das Darstellen einer Internetseite initialisiert sowie über zwei Ausgangskonnektoren, die auf die Auswahl spezieller Links in der HTML Seite reagieren. Durch das Integrieren von Links vom Typ „ACTIONURL“ und „ASML“

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

können somit browserbasierte Inhalte in beliebige Automotive Service Modelle integriert werden.

Abbildung 5-11 Bedienkonzept Browser
Quelle: Eigene Darstellung

5.1.2.3.1.4 Karten

Das letzte und vierte realisierte Bedienkonzept sind Karten. Dieses Konzept kapselt die Darstellung von Landkarten im Fahrzeug. Der Typ der Karten ist dabei grundsätzlich austauschbar, in der realisierten Implementierung im domänenspezifischen Framework wurden als Anbieter Google-Maps und OpenStreetMaps realisiert. Die Nutzung des Karten-Bedienkonzepts setzt eine aktive Internetverbindung voraus, da das Kartenmaterial nicht lokal gespeichert sondern direkt aus dem Internet angezeigt wird. Für die Parametrisierung stehen die Properties Zoom, Kartentyp, Titel, Untertitel und Aktionen zur Verfügung. Der Zoom-Faktor bezieht sich dabei jeweils auf die Definition des genutzten Kartenanbieters. Als Eingangskonnektor stehen das Zentrieren der Karte auf eine angegebene GPS-Position, das Anzeigen eines oder mehrerer Points of Interest, sowie die Darstellung einer Route, bestehend aus einer Liste von GPS-Positionen, in Form einer farbigen Linie zur Verfügung.

Abbildung 5-12 Bedienkonzept Karten
Quelle: Eigene Darstellung

5.1.2.3.2 Funktionskonzepte

Funktionskonzepte beschreiben Konzepte, die die atomaren Funktionen eines Automotive Services repräsentieren.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

5.1.2.3.2.1 Audio Player

Das Funktionskonzepte Audio Player kapselt die Funktionalität der Wiedergabe von Audioinhalten im Fahrzeug. Audioinhalte werden dabei mittels einer URL angegeben, so dass diese sowohl lokal als auch online verfügbar sein können. Der Audio Player verfügt über keine Properties mit deren Hilfe er parametrisierte werden könnte. Als Eingangskonnektor stehen Start, Stop und Pause der Wiedergabe zur Verfügung. Ein erneuter Aufruf des Eingangskonnektors Pause startet die Wiedergabe an der vormals pausierten Stelle. Als Ausgangskonnektoren stehen Signale, die das Ende des Audiostreams sowie einen Fehler bei der Wiedergabe anzeigen.

Abbildung 5-13 Funktionskonzept Audio Player
Quelle: Eigene Darstellung

5.1.2.3.2.2 Audio Recorder

Das nächste Funktionskonzept ist der Audio Recorder. Wie auch der Audio Player verfügte er über keine Properties und ist somit ebenfalls nicht parametrisierbar. Als Eingangskonnektor stehen das Starten sowie das Stoppen der Aufnahme zur Verfügung. Der Eingangskonnektor „Starten der Aufnahme“ nimmt als Parameter optional die Dauer der Aufnahme entgegen. Als Ausgangskonnektor gibt es die beiden Signale *„Aufnahme wurde gestartet“* und *„Aufnahme wurde beendet“*. Das resultierende Audiofile wird lokal gespeichert und dessen URL auf den nächsten Stream geschrieben.

Abbildung 5-14 Funktionskonzept Audio Recorder
Quelle: Eigene Darstellung

5.1.2.3.2.3 GPS Position

Das Funktionskonzept GPS Position kapselt die GPS Positionierung des Fahrzeugs. Der Baustein ist ebenfalls nicht konfigurierbar. Als Eingangskonnektor steht die Anfrage der aktuellen GPS Position zur Verfügung, als Ausgangskonnektor die Signalisierung der Feststellung eben dieser. Neben der geographischen Position wird auch der, vom Satelliten empfangene, Zeitstempel an den ausgehenden Stream übergeben.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Abbildung 5-15 Funktionskonzept GPS Position
Quelle: Eigene Darstellung

5.1.2.3.2.4 GeoCoder

Das Funktionskonzept GeoCoder dient dazu, eine gegebene Adresse in eine Geo-Koordinate zu verwandeln. Die angegebene Adresse wird dazu mithilfe einer online Datenbank in eine GPS-Position referenziert. Die beiden Eingangskonnektoren nehmen entweder eine oder mehrere Adressen zur Kodierung entgegen, deren Ergebnisse an den beiden entsprechenden Ausgangskonnektoren signalisiert wird. Zusätzlich existiert noch ein dritter Ausgangskonnektor, der anzeigt, wenn die Geo-Kodierung nicht korrekt durchgeführt werden konnte. Als Ergebnis wird entweder eine einzelne Koordinate oder entsprechend einer Liste von GPS Koordinaten auf den Stream geschrieben.

Abbildung 5-16 Funktionskonzept GeoCoder
Quelle: Eigene Darstellung

5.1.2.3.2.5 Kalender (Google)

Zum Zugriff auf Kalendereinträge steht das Funktionskonzept Kalender (Google) zur Verfügung. Mit den beiden Eingangskonnektoren können entweder alle Kalendereinträge für den aktuellen Tag oder alle Einträge für einen vorgegebenen Zeitraum, der mithilfe von Parametern aus den vorangegangenen Stream übergeben werden muss, aus dem Kalender extrahiert werden. Der einzige Ausgangskonnektor signalisiert das Ende dieser Aktion und übergibt die gefundenen Kalendereinträge an den nachfolgenden Stream. Als Parameter stehen diesem Funktionskonzept der Benutzername sowie das Passwort für den Zugriff auf den Kalender zur Verfügung.

Abbildung 5-17 Funktionskonzept Kalender (Google)
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

5.1.2.3.2.6 Kontakte (Google)

Analog zum Funktionskonzept Kalender wurde auch das Funktionskonzept Kontakte (Google) realisiert. Mithilfe des Bausteins kann auf die in Google verwalteten Kontakte zugegriffen werden. Wie auch im letzten Baustein stehen die beiden Parameter Benutzername und Passwort für eine Authentifizierung als Properties zur Verfügung. Mit den insgesamt fünf Eingangskonnektoren können die Namen der Kontakte, die E-Mail-Adressen sowie die Telefonnummern des Adressbuchs angefordert werden. Als Ausgangskonnektoren stehen sechs Signale zur Verfügung, die die extrahierten Daten auf die jeweiligen nachfolgenden Stream schreiben.

Abbildung 5-18 Funktionskonzept Kontakte (Google)
Quelle: Eigene Darstellung

5.1.2.3.2.7 Mail

Das Funktionskonzepten Mail erlaubt das Versenden von E-Mails über beliebige SMTP Server. Für die Konfiguration des Konzepts stehen die Parameter SMTP Benutzername, SMTP Passwort, SMTP Host, TLS, und Absender zur Verfügung. Der Parameter TLS nimmt die Werte *true* und *false* entgegen und konfiguriert, ob das Versenden der E-Mail verschlüsselt erfolgen soll. Als Eingangskonnektor steht die Funktion *„Sende E-Mail“* mit den Parametern Empfänger, Betreff und Text zur Verfügung. Die beiden Ausgangskonnektoren signalisieren einerseits ein erfolgreiches Versenden der Mail und andererseits einen Fehlerfall. Es werden keine Informationen auf den nachfolgenden Stream geschrieben.

Abbildung 5-19 Funktionskonzept Mail
Quelle: Eigene Darstellung

5.1.2.3.2.8 RSS Reader

Nachrichtenangebote können mithilfe des Funktionskonzepts RSS Reader in Automotive Services genutzt werden. Die URL des auszulesenden RSS Feeds kann dabei entweder als Property oder als Parameter für den Eingangskonnektor angegeben werden. Als Ergebnis der

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Aktivitäten stehen die beiden Ausgangskonnektor *„RSS Feed wurde eingelesen“* und *„RSS Feed konnte nicht eingelesen werden“* zur Verfügung. Die Informationen des eingelesenen RSS Feeds werden in Form von textuellen Listen auf den nachfolgenden Stream geschrieben.

Abbildung 5-20 Funktionskonzept RSS Reader
Quelle: Eigene Darstellung

5.1.2.3.2.9 Routing

Für einfache Navigationsaufgaben steht das Funktionskonzept Routing zur Verfügung. Auf der Grundlage der Geoinformationen von OpenStreetMaps kann einen Route berechnet werden. Der Eingangskonnektor nimmt als Parameter entweder eine Start und eine Zielposition oder eine Liste von Positionen, die in der angegebenen Reihenfolge abgefahren werden sollen, entgegen. Die beiden Ausgangskonnektoren signalisieren, ob die Route berechnet werden konnte oder nicht. Als Ergebnis wird eine Liste von GPS Positionen, die die einzelnen Waypoints der Route darstellen, auf den nachfolgenden Stream geschrieben. Die Berechnung erfolgt online, so dass eine ausreichende Internetverbindung verfügbar sein muss.

Abbildung 5-21 Funktionskonzept Routing
Quelle: Eigene Darstellung

5.1.2.3.2.10 Spracherkennung

Ein weiteres Funktionskonzept ist die Spracherkennung. Mithilfe der Spracherkennung können diskrete Befehle in einer vorgegebenen Sprache im Rahmen eines Automotive Services erkannt werden. Die zu erkennenden Befehle werden in Form von Aktionen als Property angegeben. Als implementierte Sprachen stehen Deutsch und Englisch zur Verfügung. Nach Aktivierung des Konzepts durch den Eingangskonnektor wartet der Baustein bis ein Begriff erkannt wurde und signalisiert dies am entsprechenden Ausgangskonnektor. Für jeden zu erkennenden Begriff existiert also genau ein Ausgangskonnektor. Zusätzlich wird der erkannte Begriff auf den nachfolgenden Stream geschrieben.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Abbildung 5-22 Funktionskonzept Spracherkennung
Quelle: Eigene Darstellung

5.1.2.3.2.11 Sprachsynthese

Analog zum Audio Player dient das Funktionskonzept Sprachsynthese zur Ausgabe von Audio im Fahrzeug. Allerdings werden in diesem Fall keine Audiostreams wiedergegeben, sondern textuelle Inhalte als Sprache synthetisiert. Das Konzept benötigt keine Properties zur Konfiguration und verfügt über, dem Audio Player analoge, Konnektoren (vgl. Abschnitt 5.1.2.3.2.1). So steht je ein Eingangskonnektor für das Starten, Stoppen und Pausieren der Sprachsynthese sowie ein Ausgangskonnektor für das Signalisieren des Endes und eines eventuellen Fehlers der Sprachausgabe. Der auszugebene Text wird als Parameter an den entsprechenden Eingangskonnektor übergeben.

Abbildung 5-23 Funktionskonzept Sprachsynthese
Quelle: Eigene Darstellung

5.2 Case Study: Modellierung von Automotive Services

Eine domänenspezifische Modellierungssprache ist natürlich viel mehr als die Summe ihrer einzelnen Bestandteile, und so ist auch bei der Automotiv Services Modeling Language der geeignete Einsatz der Modellierungskonstrukte und Elemente der Sprache von entscheidender Bedeutung. Erst eine geeignete Kombination macht aus diesen Bestandteilen ein wertschöpfendes Modell. Um die Möglichkeiten und den Einsatz der Modellierungssprache am praktischen Beispiel zu verdeutlichen, werden im Anschluss beispielhaft zwei Modelle diskutiert.

Beim ersten Modell handelt es sich um einen einfachen Nachrichtendienst, wie er analog auch im Projekt MACS (Reichwald/Krcmar/Reindl 2007) realisiert wurde (vgl. Abschnitt 4.3.1). Das Beispiel zeigt einerseits, wie vormals aufwändig zu realisierende Dienste mithilfe der Automotiv Services Modeling Language in einer einfachen Art und Weise beschrieben werden können und illustriert andererseits in diesem Zuge auch den Einsatz der Gestaltungselemente der Sprache.

Das zweite Modell ist eine Kalenderapplikation, mit deren Hilfe der Nutzer sich seine tagesaktuellen Termine im Fahrzeug anzeigen und vorlesen lassen kann. Darüber hinaus besteht noch die Möglichkeit, einzelne Termine auf der Landkarte zu referenzieren und eine

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Tagesroute zu planen. Durch die höhere Komplexität dieses zweiten Modells eignet es sich auch, den Einsatz von Composites zu illustrieren.

5.2.1 ASML MyNews

ASML MyNews ist ein einfacher Automotive Service, der aktuellen Nachrichten, die aus unterschiedlichen Onlinequellen stammen können, im Fahrzeug darstellen. Für die Umsetzung des Dienstes werden, wie in Abbildung 5-24 zu sehen, lediglich fünf Bausteine sowie die dazugehörigen, verbindenden, Streams, benötigt.

Ausgehend von einem Startelement wird mithilfe des ersten Streams synchron das Funktionskonzept RSS Reader aufgerufen. Der Brick wird über eine Property bereits während der Modellierung mit der URL der Quelle des RSS Feeds versorgt. An dieser Stelle ist es unerheblich, welcher RSS Feed gewählt wird, solange dieser mit dem RSS Standard kompatibel ist. Da es sich bei dem Baustein um das erste Brick dieses Typs im Modell handelt, wird der Index im Titel mit einer 1 angegeben. Notizen sind, wie in Abbildung 5-24 zu sehen, nicht vorhanden.

Wurde der RSS Feed erfolgreich eingelesen, so werden dessen Informationen am entsprechenden Ausgangskonnektor auf den nachfolgenden Stream geschrieben, welcher wieder synchron den nächsten Baustein aufruft. Das Bedienkonzept Liste dient dazu, die Titel der einzelnen Einträge des RSS Feeds darzustellen. Am Eingangskonnektors wird also die Informationen die aus dem ersten Brick stammen, als Input für die Liste verknüpft. Als Properties werden der Titel *„ASML mein News“* und der Untertitel *„Spiegel Online“* sowie eine Aktion *„Reload“* spezifiziert. Beim Titel handelt es sich um den Namen des Automotive Services, der Untertitel gibt den Namen des RSS Feeds wieder. In diesem Fall wurde als Beispiel der RSS Feed des Nachrichtenmagazins Spiegel[16] genutzt.

An welchen der drei Ausgangskonnektoren nun der nächste Stream aktiviert wird, hängt von der Interaktion des Nutzers mit dem Bedienkonzeptbaustein Liste ab. Ihm stehen, wie in Abschnitt 5.1.2.3.1.1 beschrieben, das Selektieren und das Aktivieren eines Listenelements zur Verfügung. Zusätzlich gibt es noch die im Modell über die Properties spezifizierte Aktion *„Reload“*. Wie in Abbildung 5-24 zu sehen, wurde die Selektion nicht mit einem Stream verknüpft, so dass diese Aktion des Nutzers ohne Auswirkungen bleibt. Wählt der Nutzer die Aktivität *„Reload“,* so wird der RSS Reader erneut aufgerufen. Dies hat zur Folge, dass auch der RSS Feed erneut aus dem Internet geladen und interpretiert wird und die Informationen auf den Stream mit diesen neuen, aktualisierten Informationen überschrieben werden. Aus Sicht des Nutzers hat die Aktivierung der Aktion *„Reload“* also zur Folge, dass die Informationen die in der Liste angezeigt werden, aktualisiert werden.

Die zweite Möglichkeit für den Nutzer ist das Aktivieren eines Listenelements. Wie in Abbildung 5-24 zu sehen, gehen in diesem Fall zwei Streams am entsprechenden Ausgangskonnektor los. Der synchrone Stream aktiviert den nächsten Bedienkonzeptbaustein, der asynchrone aktiviert die Sprachsynthese. Das Aktivieren eines Listenelements hat also zur Folge, dass der Inhalt des entsprechenden RSS Eintrags sowohl vorgelesen als auch gleichzeitig textuell am Bildschirm angezeigt wird. Um dies zu ermöglichen, wird auf beiden Streams der Inhalt des RSS-Eintrags an den Eingangskonnektor des nachfolgenden Bausteins übergeben. Durch den Einsatz eines asynchronen Streams für die Sprachsynthese wird dieser

[16] http://www.spiegel.de/schlagzeilen/index.rss

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

in einem parallelen Prozess gestartet, so dass nun sowohl die Sprachsynthese als auch das Bedienkonzept Text aktiv sind. Dies bedeutet für den Nutzer, dass er die Funktionen des Bricks Text steuern kann, ohne dabei die Sprachsynthese zu beeinflussen oder auf deren Terminierung warten zu müssen. Das Text Brick wurde wiederum mit dem Titel *„ASML MyNews“* und Untertitel *„Spiegel Online“* initialisiert. Aus Sicht des Nutzers bleiben also der Titel und der Untertitel der Anzeige bei einem Wechsel vom ersten Bedienkonzeptbaustein zu diesem Bedienkonzeptbaustein identisch. Zusätzlich wurde noch die Aktion *„Zurück“* als Property spezifiziert, so dass der Nutzer über deren Aktivierung zur Liste der RSS Nachrichten zurückkehren kann. Diese Rückkehr erfolgt wiederum über einen synchronen Stream.

Aktiviert der Nutzer also die Funktion *„Zurück“*, wird das Text Brick deaktiviert und der Listen Brick wieder aktiviert. Sollte die Sprachsynthese noch nicht beendet worden seien, so läuft diese, bis zu ihrem Ende, weiter. Erst wenn wieder ein Listeneintrag vom Nutzer aktiviert wird, wird die noch laufende Sprachsynthese durch die Synthese des nun ausgewählten Eintrags ersetzt. Die alte Ausgabe bricht also ab und die Neue wird gestartet.

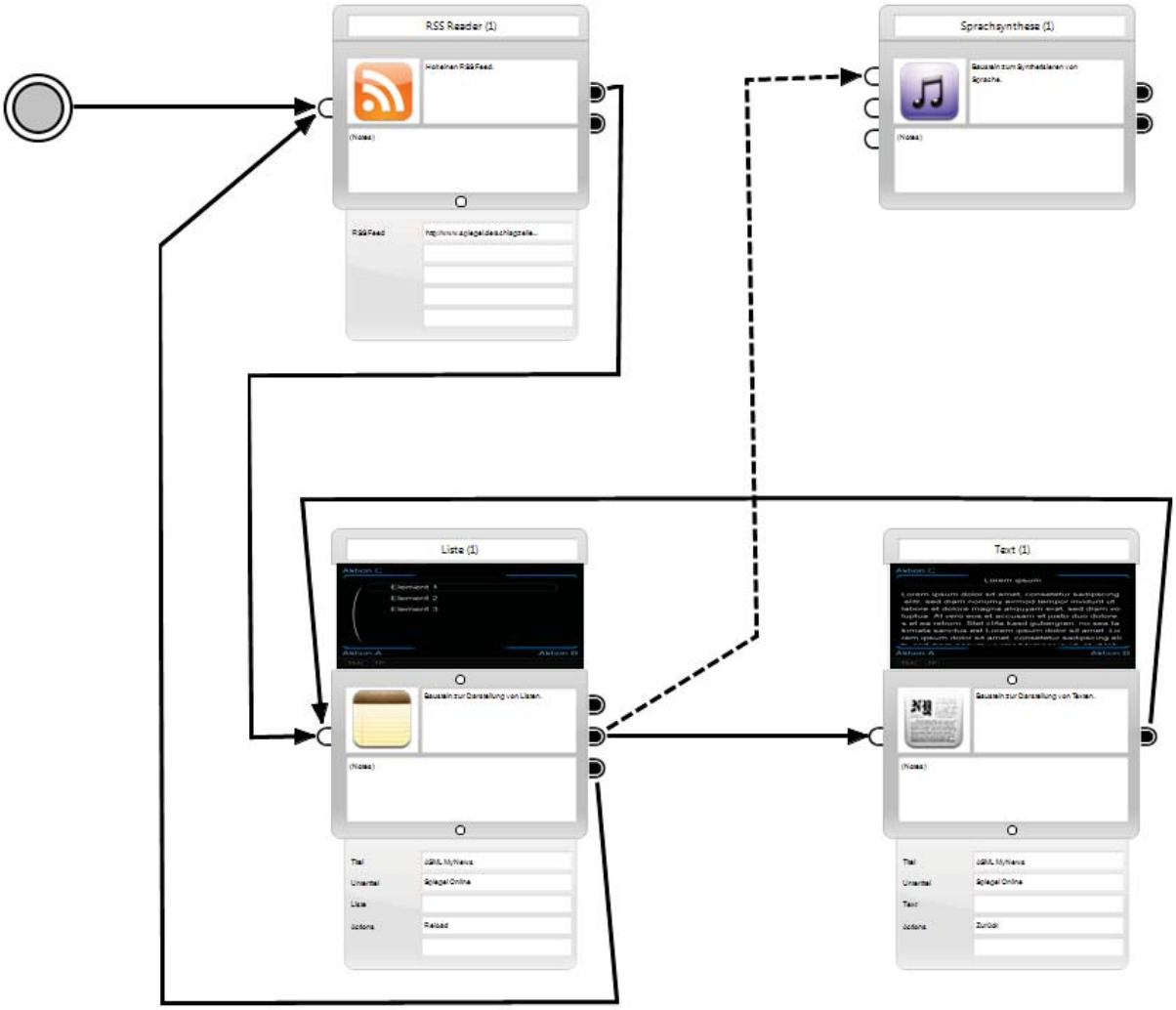


Abbildung 5-24 Case Study ASML MyNews
Quelle: Eigene Darstellung

Dieser, aus lediglich fünf Bausteine bestehende Dienst, stellt sich aus Sicht des Nutzers im Fahrzeug als simpler Automotive Service dar, der aus lediglich zwei unterschiedlichen Anzeigen besteht. Sobald der Dienst gestartet wird, erscheint eine Liste von Nachrichten auf

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

dem Bildschirm, die der Nutzer mithilfe des Dreh-Drück-Stellers auswählen und aktivieren kann. Sobald der Dreh-Drück-Steller gedrückt wird, wird die gewählte Nachricht vorgelesen am Bildschirm angezeigt. Durch die Aktion *„Zurück“* gelangt der Nutzer zurück zur Übersicht der Nachrichten, welche er durch Betätigen von *„Reload“* aktualisieren kann. Bei Auswahl einer Nachricht wird diese vorgelesen und auch bei der Rückkehr zur Nachrichtenübersicht wird dieses Vorlesen zu Ende geführt.

5.2.2 ASML MyDay

Der zweite Automotiv Service ASML MyDay ist ein Beispiel für einen deutlich komplexeren Service, wie ihn der im letzten Abschnitt vorgestellte Dienst ASML MyNews darstellt. Inhaltlich dient der Automotiv Services dazu, die Termine des aktuellen Tages in Form einer Liste darzustellen und unterschiedliche Aktionen auf diesen Terminen auszuführen. So kann der Nutzer sich die Beschreibung eines Termins Anzeigen und Vorlesen, den Ort an dem der Termin stattfinden soll auf einer Landkarte einzeichnen oder eine Route, ausgehend von der aktuellen Position des Fahrzeugs, über alle Termine hinweg berechnen lassen. Als Grundlage für diese Terminplanung wird ein Google Kalender genutzt, indem für jeden Termin, neben Angaben zu Datum und Uhrzeit, ein Name, eine Beschreibung sowie eine Adresse des Termins hinterlegt sind.

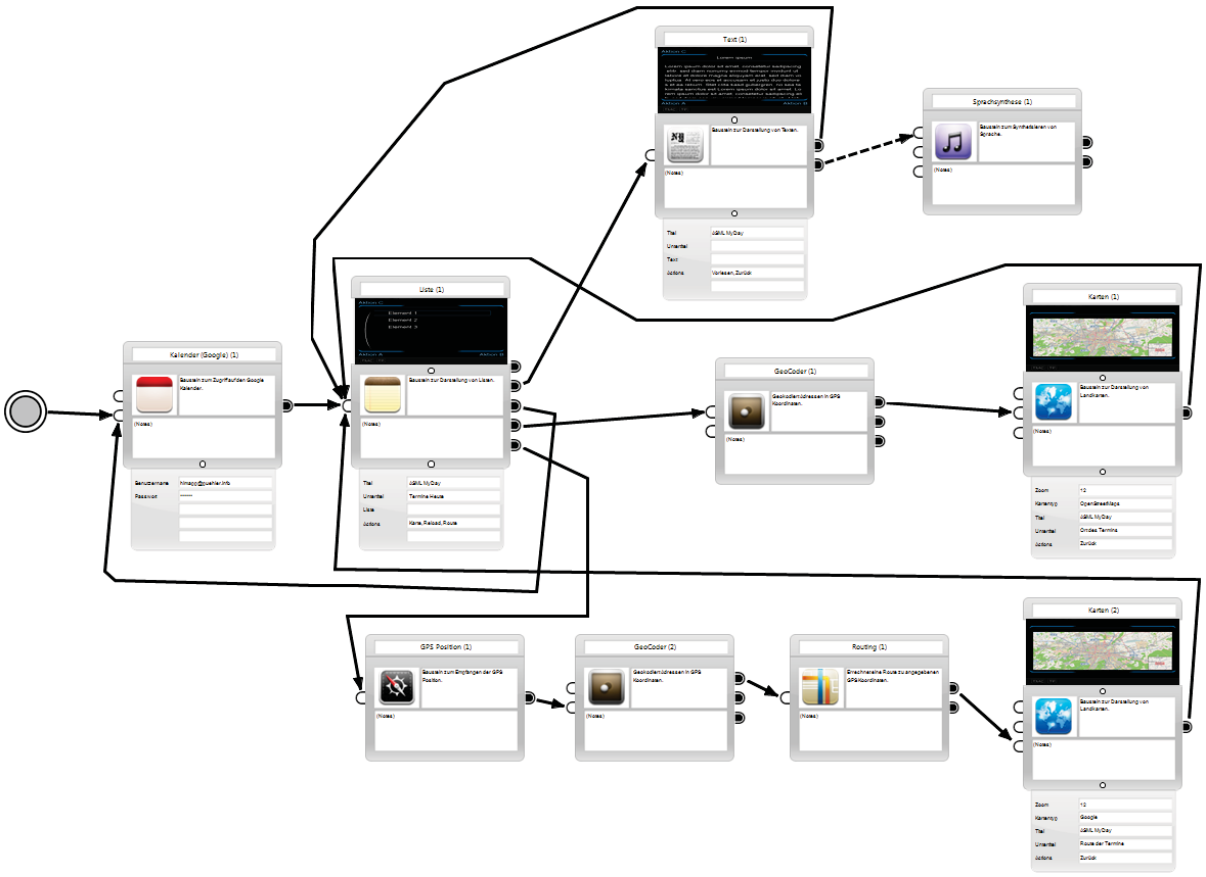


Abbildung 5-25 Case Study ASML MyDay
Quelle: Eigene Darstellung

Wie in Abbildung 5-25 zu sehen, wird ausgehend vom Startelement analog dem letzten Service eine Funktionskonzept Brick aufgerufen. Der Baustein Kalender (Google) wird am Eingangskonnektor *"Hole alle Termine für den aktuellen Tag"* aufgerufen. Über die Properties Benutzername und Passwort kann der Brick so auf den Onlinekalender zugreifen

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

und die relevanten Informationen extrahieren. Diese werden anschließend am Ausgangskonnektor auf den nachfolgenden Stream geschrieben.

Mithilfe dieses Streams wird das Bedienkonzept Liste aufgerufen und die Titel der Kalendereinträge als Listenelemente übergeben. Neben dem Titel *"ASML MyDay"* und dem Untertitel *"Termine Heute"* werden in den Properties noch die drei Aktionen „*Karte*“, „*Reload*“ und „*Route*“ spezifiziert. Zusammengenommen mit den beiden Ausgangskonnektoren für die Selektion und Aktivierung von Listenelementen stehen nun fünf Ausgangskonnektoren zur Verfügung. Wie auch im letzten Beispiel ist das Selektieren eines Listenelements nicht mit einem Stream verknüpft und hat somit keine Aktivität zur Folge. An den verbleibenden vier Ausgangskonnektoren kann der Nutzer vier unterschiedliche Aktionen starten.

Über die Funktion „*Reload*“ wird analog dem letzten Beispiel der Funktionsbaustein Kalender (Google) erneut aufgerufen. Dies hat zur Folge, dass die Termine und somit auch die angezeigte Liste aktualisiert werden.

Der Ausgangskonnektor „*Aktiviert*“ ist über einen synchronen Stream mit einem Text Brick verbunden. Die Anzeige der aktuellen Liste wird also beendet und deaktiviert und das nachfolgende Text Brick aktiviert und angezeigt. Dieses nun aktive Brick ist über den Property Titel *"ASML MyDay"* sowie die Aktionen „*Vorlesen*“ und „*Zurück*“ parametrisierte. Der Untertitel wird zusammen mit dem anzuzeigenden Text am Eingangskonnektor mit dem Stream übergeben. Als Untertitel wird der Name des Kalendereintrags und als anzuzeigender Text dessen Beschreibung verwendet. Die beiden Ausgangskonnektoren führen zum einen Zurück zur Anzeige der Liste der aktuellen Termine und zum anderen zum Vorlesen der Beschreibung des aktuellen Termins. Dieses Vorlesen wird durch einen asynchronen Stream aufgerufen und wird somit unabhängig von den weiteren Aktionen des Nutzers vollständig durchgeführt.

Die dritte Aktivität, die der Nutzer auswählen kann, ist die Aktion „*Karte*“. Mit diesem Ausgangskonnektor wird über einen synchronen Stream das Funktionskonzept GeoCoder verknüpft. Am Eingangskonnektor wird diesem Brick die Adresse des in der Liste gewählten Termins übergeben. Mithilfe einer online Datenbank wird diese Adresse in eine Geokoordinate übersetzt und diese am entsprechenden Ausgangskonnektor an den ausgehenden Stream übergeben. Dieser führt wiederum zum Bedienkonzept Karten, der den Namen des in der Liste gewählten Termins als Point of Interest an der vom GeoCoder kodierten Koordinate auf einer Landkarte anzeigt. Das Karten Brick wird dazu mit den Properties Zoom (12), Kartentyp (OpenStreetMaps), Titel (ASML MyDay), Untertitel (Ort des Termins) und der Aktion „*Zurück*“ parametrisiert. Die einzige Aktion, die der Nutzer am Karten Brick nun ausführen kann ist „*Zurück*“. Durch einen synchronen Stream wird die Liste der Termine aufgerufen der Karten Brick deaktiviert.

Die vierte und letzte Aktion, die der Nutzer mit den Terminen aufrufen kann, ist die Aktion „*Route*“. Wird dieser Ausgangskonnektor am Listenbrick aktiviert, wird mit einem synchronen Stream das Funktionsbrick GPS Position aufgerufen. Sobald die aktuelle Koordinate identifiziert werden konnte, wird über einen weiteren synchronen Stream der Ausgangskonnektor des Bricks mit dem nachfolgenden Brick GeoCoder verknüpft. Dieser GeoCoder kodiert nun wiederum die Adressen aller Termine zu GPS-Koordinaten. Anzumerken ist hierbei, dass dies das zweite Vorkommen des GeoCoder Bricks im Modell ist und dieser somit im Titel mit dem Index 2 identifiziert wird. Sobald die Koordinaten

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

identifiziert wurden, wird wiederum über einen synchronen Stream das Funktionskonzept Routing aufgerufen. An dessen Eingangskonnektoren werden die, in den vorangegangenen Bricks gesammelten Informationen, als Eingangsparameter genutzt. So wird sowohl die GPS-Position des GPS Bricks als auch die Positionen der Termine aus dem GeoCoder als Input verwendet. Sobald die Route berechnet wurde, wird dies am entsprechenden Ausgangskonnektor signalisiert und über einen weiteren synchronen Stream ein Kartenbrick aufgerufen. Dieses Kartenbrick wird ebenfalls mit dem Index 2 identifiziert, da es sich um das zweite Vorkommen dieses Bricks im Modell handelt. Bis auf den Untertitel werden hier die gleichen Properties angegeben wie im ersten Kartenbrick. Als Untertitel wird *„Route der Termine“* angegeben. Der Kartenbrick zeigt dem Nutzer eine Landkarte, auf der die Route über alle Termine ausgehend vom aktuellen Ort angezeigt wird. Als einzige Aktion steht wiederum *“Zurück“* zur Verfügung. Dem Nutzer gelangt so wieder zur Liste der Termine.

Aus Sicht des Nutzers besteht ASML MyDay aus einem Dienst mit vier Anzeigen: Initial wird eine Liste der Termine des heutigen Tages am Bildschirm angezeigt. Ausgehend von dieser Liste kann sich der Nutzer die Beschreibung des Termins anzeigen und vorlesen, den Termin auf einer Karte einzeichnen und eine Route über alle Termine hinweg berechnen anzeigen lassen.

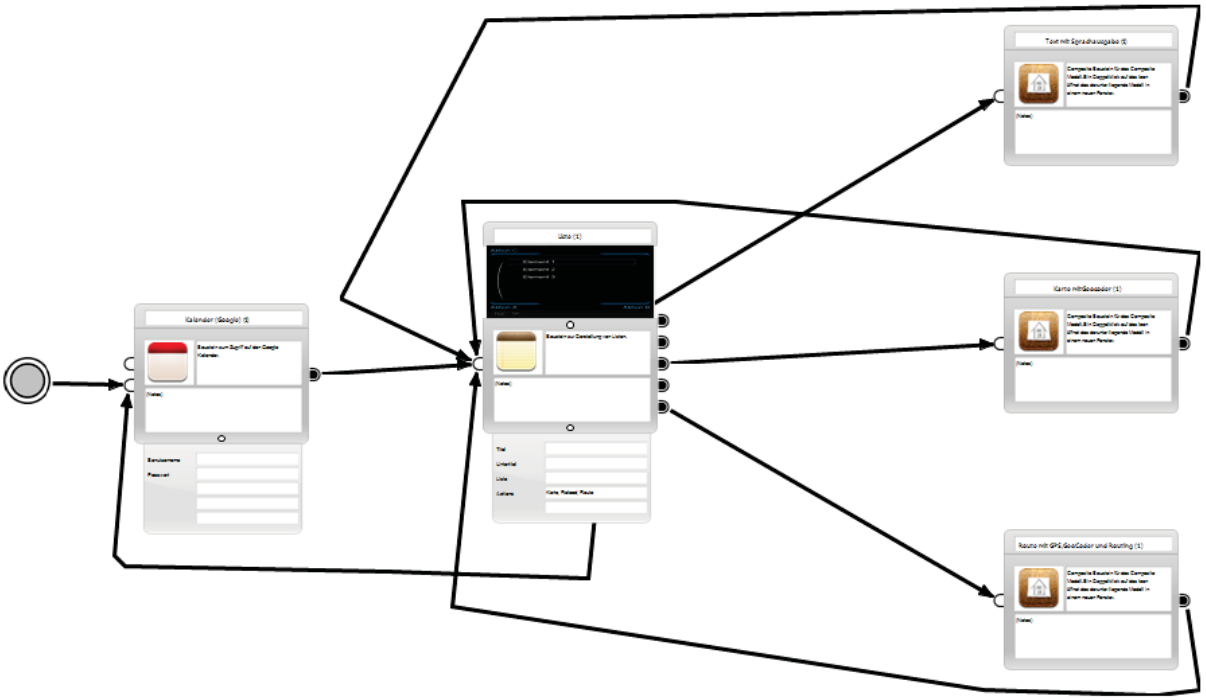


Abbildung 5-26 Case Study ASML MyDay(Composites)
Quelle: Eigene Darstellung

Diese drei Aktionen können auch in Form von Composites als wieder verwendbares Wissen zusammengefasst werden. Abbildung 5-26 zeigt das gleiche Modell mit dem Unterschied, dass die drei möglichen Aktionen jeweils durch ein Composite zusammengefasst werden. Somit muss sich der Modellierer nicht mit den Details dieser Aktionen befassen, sondern kann sich auf den wertschöpfenden Beitrag weitere Aktionen und Funktionen im Modell konzentrieren. Darüber hinaus ist das Modell auf diese Weise deutlich einfacher in der Interpretation geworden.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

5.3 Zusammenfassung

Auf der Grundlage eines theoriebasierten Designs im Rahmen eines gestaltungsorientierten Forschungsansatzes wurde den zentralen Herausforderungen, die sich aus den einzelnen Schritten des Design Thinking an eine Unterstützung durch ein domänenspezifisches Modellieren von Automotiv Services ergeben, durch drei zentrale Designprinzipien begegnen.

Durch die Einführung von Bricks wird der Notwendigkeit einer domänenspezifischen Repräsentation entsprochen, um so Stakeholder in die Lage zu versetzt ihr domänenspezifisches Wissen zu nutzen und so ein besseres Verständnis der Lösung zu erlangen. Eine Implementierbarkeit der Repräsentation wird durch das Designprinzip Stub realisiert. Durch die Möglichkeit, domänenspezifische Konzepte in Form von Stubs zu spezifizieren, wird der Aufwand der Problemlösung für die einzelnen Stakeholder wesentlich reduziert, da sie sich zukünftig nur noch mit einem Element, dem Stub, und nicht mehr mit einem komplexen Konzept auseinandersetzen müssen. Dies ermöglicht eine Reduktion der kognitiven Belastung und führt auf der einen Seite dazu, das Stakeholder ihre Vorstellungen leichter im Rahmen des Modells realisieren, und Konzepte, auf der anderen Seite, einfacher in das domänenspezifische Framework integriert werden können. Mit dem dritten Designprinzip, dem Composite, wird der Herausforderung einer Notwendigkeit für ein Erfahrungs-management begegnet. Durch die Produktivierung erprobter Services kann so die Fähigkeit zur Problemlösung maßgeblich verbessert werden, da bei der Gestaltung zukünftiger Services dieses erprobte Wissen in Form von Composites einfacher wieder verwendet werden kann. Ein Erfahrungsmanagement versetzt Stakeholder in die Lage, sich jeweils nur mit den aktuellen Problemen beschäftigen zu müssen und bereits gelöste Aspekte einfach wieder zu verwenden.

Auf der Grundlage dieser drei Designprinzipien konnte eine domänenspezifische Modellierungssprache für Automotiv Services gestaltet werden. Wie dargestellt, besteht die Notation aus fünf Elementen, die zusammen mit einem entsprechenden Regelwerk und den domänenspezifischen Konzepten die Automotive Services Modeling Language (ASML) bilden. Im Rahmen von Case Studies wurde der Einsatz der Modellierungssprache am Beispiel von zwei Automotive Services gezeigt. Insbesondere das Designprinzip der Composites hilft hier, Komplexität zu reduzieren und erprobtes Wissen wiederverwendbar zu machen. Um den Erfolg und die Nützlichkeit dieser Modellierung bewerten zu können, wurden darüber hinaus drei Hypothesen entwickelt und formuliert. So besagt die erste Hypothese, dass Stakeholder mit geringen bis gar keinen technischen Kenntnissen der Automotive Domäne und ihrer IT-Komponenten in der Lage sein sollen, realisierbare Automotive Services zu gestalten. Darüber hinaus besagt die zweite Hypothese, dass Stakeholder mit geringen bis gar keinen technischen Kenntnissen der Automotive Domäne und ihrer IT-Komponenten in der Lage sein sollen, umsetzbare Anforderungen an eine Implementierung einzelner Komponenten zu definieren. Schließlich fordert die dritte Hypothese, dass Stakeholder in der Lage sind, existierende Elemente wieder zu verwenden um neue Funktionalitäten für künftige Nutzer zu definieren und so die Fähigkeiten der Automotive Services Modeling Language zu erweitern.

Um also die domänenspezifische Modellierungssprache für Automotive Services intersubjektiv nachvollziehbar bewerten zu können, ist eine Überprüfung dieser drei Hypothesen notwendig. Um diese Evaluation durchzuführen, fehlt allerdings ein domänenspezifischer Code Generator sowie eine Umsetzung der Konzepte im Rahmen eines

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

domänenspezifischen Frameworks (Tolvanen/Kelly 2004). Darüber hinaus müssen die Automotive Service Modelle in einer interpretierbaren Form vorliegen, was durch den Einsatz eines Modellierungswerkzeugs realisiert werden kann. Der nachfolgende Abschnitt befasst sich daher mit der Gestaltung und Umsetzung dieser Elemente.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

6 Gestaltung einer durchgängigen Werkzeugunterstützung

Wie in Abschnitt 4.1.6.2 diskutiert besteht nach Tolvanen und Kelly (2004) eine domänenspezifische Modellierung aus insgesamt drei Elementen: Im vorangegangen Abschnitt wurde die *domänenspezifische Modellierungssprache*, ihre Notation, Regeln sowie einer Auswahl domänenspezifischer Konzepte auf der Basis von drei gestaltenden Designprinzipien definiert. Zusätzlich zu dieser Modellierungssprache fehlen noch die beiden Elemente des *domänenspezifischen Code Generators* sowie des *domänenspezifischen Frameworks*.

Der Prozess von der Produktidee hin zum Automotive Service wird demnach in zwei Schritten vollzogen. Zunächst wird die Produktidee mit Hilfe der domänenspezifischen Modellierungssprache in einem Modell beschrieben und spezifiziert. Aus Sicht des Nutzers ist dies das Produktmodell. Im zweiten Schritt wird dieses Modell als Grundlage herangezogen, um mit Hilfe des Code Generators und des domänenspezifischen Frameworks den entsprechenden Produktcode zu generieren. Dieser Prozess ist für den Nutzer transparent, so dass die Modellierung einer Produktidee automatisch zum daraus resultierenden Produktcode führt.

Im nachfolgenden Abschnitt werden diese noch fehlenden Elemente einer domänenspezifischen Modellierung diskutiert. Zusammen mit der Modellierungssprache bilden sie die Grundlage für die Evaluation des beschriebenen Konzepts.

6.1 Architektur

Die Architektur zur Unterstützung der Automotive Services Modeling Language basiert auf dem beschriebenen Prozess von der Produktidee über das Produktmodell zum Produktcode in zwei Schritten. Ausgehend von der Produktidee wird ein grafisches Modellierungswerkzeug genutzt um ein, die Ideen repräsentierendes, Produktmodell zu entwickeln. Die Gestaltung dieses Modells basiert auf der Grundlage der in Abschnitt 5 definiert Modellierungssprache für Automotive Services und basiert somit im Wesentlichen auf einer Verknüpfung von Bedienkonzepten und Funktionskonzepten. Abbildung 6-1 stellt diesen Zusammenhang graphisch dar. Das Modellierungswerkzeug wiederum besteht aus zwei Elementen. Auf der einen Seite befindet sich der Editor, der das eigentliche Modellierungswerkzeug für ein Automotive Service Modell darstellt, sowie auf der anderen Seite die Workbench, die die Anwendung rund um den Editor realisiert.

Der Editor ist eine graphisch Modellierungsumgebung mit deren Hilfe die Nutzer mittels Drag‘n‘Drop Automotive Service Modelle zusammenstellen können. Für diese Aufgabe werden als Gestaltungselement die Bricks der Modellierungssprache benutzt. Natürlich genügt es für ein Modellierungswerkzeug nicht, lediglich ein einzelnes Modell gestalten zu können. Vielmehr ist eine Verwaltung einer Vielzahl an Modellen sowie unterstützende Funktionalitäten, wie das Rückgängig machen von Aktivitäten oder das Umbenennen eines Modells, für ein vollständiges Modellierungswerkzeug unabdingbar. Ebenso müssen Composites, die ihrerseits wiederum durch ein eigenes Automotive Service Modell beschrieben sind, von der Workbench verwaltet werden. Darüber hinaus ist es die Aufgabe der Workbench, die Nutzer bei der Gestaltung und Bearbeitung neuer Stubs zu unterstützen.

Das Ergebnis dieses ersten Schritts ist also ein domänenspezifisches Produktmodell, welches bestehend aus den Elementen der Modellierungssprache die Produktidee beschreibt. Der

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

nächste, nun folgende Schritt, ist der Übergang vom Produktmodell zum Produktcode. Wie bereits erwähnt, soll dieser Schritt transparent für den Nutzer erfolgen, so dass gestaltete Produktmodelle direkt als Produktcode ausgeführt werden können.

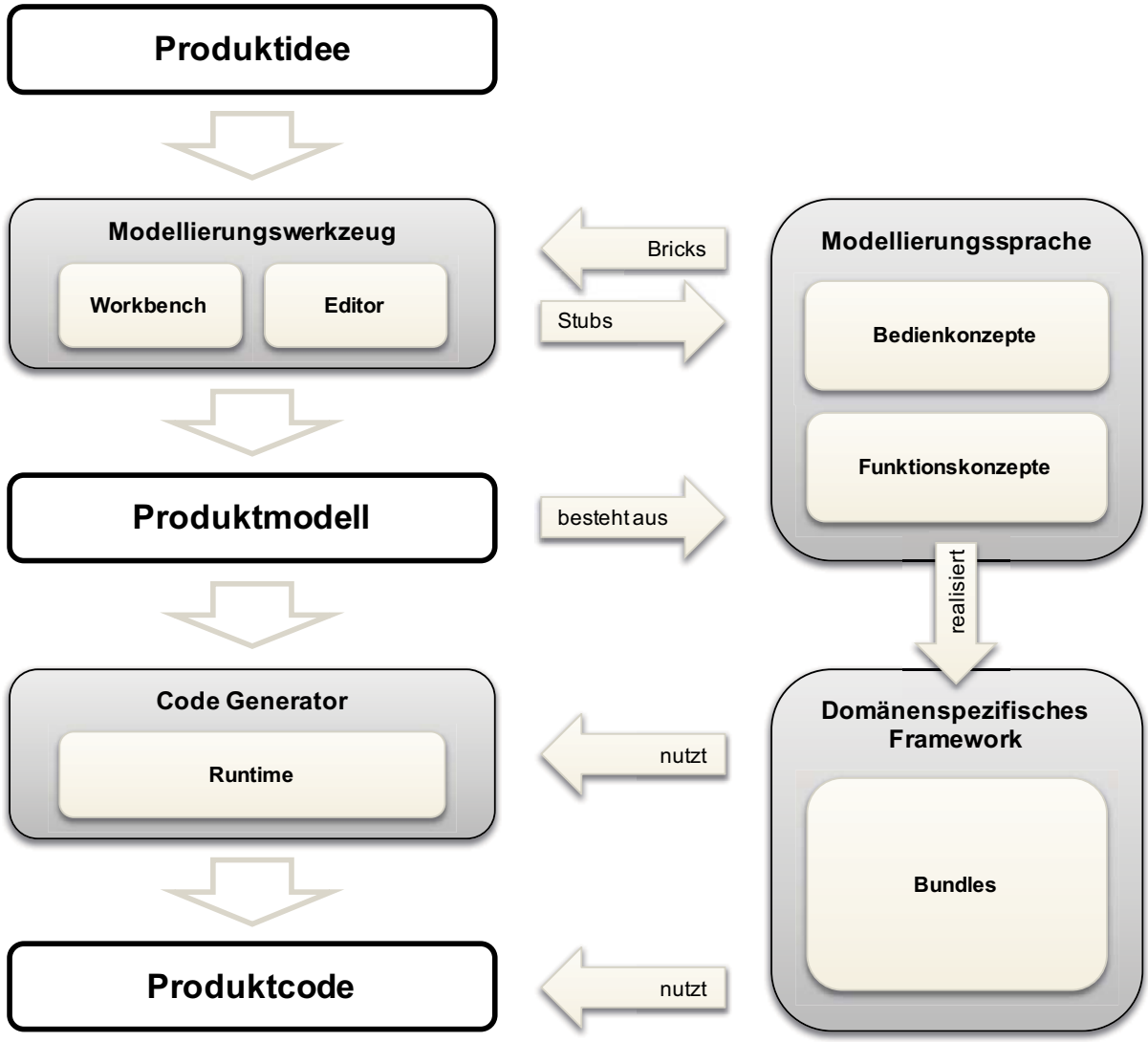


Abbildung 6-1 Architektur der Werkzeugunterstützung
Quelle: Eigene Darstellung

Nach Tolvanen und Kelly (2004) ist für diesen Übergang einerseits ein Code Generator und andererseits ein domänenspezifisches Framework notwendig. Wie in Abbildung 6-1 zu sehen, besteht das domänenspezifische Framework aus Bundles, die die einzelnen Bedienkonzepte und Funktionskonzepte der Modellierungssprache realisieren und implementieren. Für jedes Konzept existiert im Framework somit genau ein Bundle das dieses implementiert. Darüber hinaus gibt es im domänenspezifischen Framework noch eine Reihe an zusätzlichen Funktionalitäten, wie zum Beispiel die Interpretation des Bedienteils oder die Verwaltung der Internetverbindung. Das domänenspezifische Framework ist im eigentlichen Sinne nicht Teil dieser Arbeit, sondern basiert auf der Highly Integrated Modular Embedded Prototyping Plattform (HIMEPP), die im Rahmen der Forschungsarbeit von Hoffmann (2010) entstanden ist.

Das zweite Element, der Code Generator, wird durch die Einführung eine Runtime realisiert. Anstatt das Produktmodell in ausführbaren Code zu transformieren, wird dieses von einer Runtime zur Laufzeit interpretiert. Somit ist für die Ausführung einzelner Produktmodelle,

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

sofern alle Konzepte im domänenspezifischen Framework realisiert sind, kein Generieren von Code notwendig. Dieser Ansatz der direkten Modellausführung hat des Weiteren den Vorteil, dass aus Sicht des Nutzers der eigentliche Schritt der Umwandlung des Modells in den Produktcode entfällt. Wie in Abbildung 6-1 zusehen, nutzen also sowohl der Code Generator als auch der Produktcode die Funktionalität des domänenspezifischen Frameworks.

6.2 Umsetzung der Elemente der Werkzeugunterstützung

Für die Umsetzung der einzelnen Komponenten der Modellierungsumgebung wurde als Grundlage die Eclipse Plattform genutzt. „Die Eclipse Plattform ist eine Open Source Plattform für die Erstellung von Anwendungen und Entwicklungswerkzeugen…“ (Hoffmann 2010, 233). Dies hat zum einen historische Gründe, die sich aus der in Abschnitt 4.3.1 beschriebenen Forschungslandschaft dieser Arbeit ergeben, zum anderen ist das domänenspezifische Framework HIMEPP auf der Grundlage der hinter Eclipse stehenden Technologien entstanden (Hoffmann 2010, 135). Darüber hinaus bietet die Eclipse Plattform, welche auf der offenen, objektorientierten Programmiersprache Java aufsetzt, eine Vielzahl an Funktionalitäten, die die Umsetzung der zu entwickelnden Elemente der Werkzeug-unterstützung unterstützt (Eclipse Foundation 2010a). Die Eclipse Plattform wurde 2001 von IBM im Rahmen der Eclipse-Community als Open Source Projekt freigegeben und wird seitdem stetig weiterentwickelt und verbessert.

Die Grundlage für die Eclipse Plattform bildet ein Plugin-Konzept auf Basis der Eclipse eigenen OSGi[17] Implementierung Equinox. Equinox ist ein auf Java aufbauendes Framework für die komponentenorientierte Gestaltung produktiverter Konzepte. Dass den „Open Services Gateway initiative“ Standard implementierende Framework bildet sowohl die Grundlage für die Eclipse Entwicklungsumgebung als auch für die darunter liegende Laufzeitumgebung der Eclipse erweiternden Plugins. Im domänenspezifischen Framework HIMEPP wird Equinox genutzt, einzelne Bundles zu realisieren (Hoffmann 2010, 135). Das Framework entspricht also den Grundsätzen der Produktivierung und eignet sich daher insbesondere, domänenspezifische Konzepte der Modellierung im Rahmen des Frameworks zu realisieren.

Auf der einen Seite erlaubt die Eclipse Plattform also eigene Funktionalitäten und Werkzeuge zu entwickeln, auf der anderen Seite dient sie als Basis für die Realisierung neuer Stubs im domänenspezifischen Framework. Ausgehend davon, dass das domänenspezifische Framework in Form von HIMEPP bereits realisiert ist und die Modellierungssprache in Abschnitt 5 definiert wurde, müssen für eine vollständige Umsetzung dieser Architektur noch vier Elemente gestaltet werden: Dies sind die *Workbench*, der *Editor* sowie die *Runtime*. Zusätzlich muss noch das *Persistieren der Produktmodelle* betrachtet werden, da diese nur so von einem Werkzeug zum nächsten weitergegeben werden können.

6.2.1 Modellierungswerkzeug: Workbench

Der erste Bestandteil des Modellierungswerkzeugs ist die ASML Workbench. Die Aufgabe der Workbench ist es, die Funktionalität des Modellierens von Automotive Service Modellen im Rahmen einer, für den Nutzer leicht verwendbaren, Applikationen zu kapseln. So übernimmt die Workbench verwaltende und unterstützende Aufgaben, legt das Layout und die Darstellung der Anwendung fest und übernimmt das Speichern und Erstellen neuer Modelle. Schließlich dient die Workbench als Container für einzelne Editoren.

[17] Open Services Gateway initiative

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

6.2.1.1 Grundlagen

Die Grundlage für die Entwicklung des Modellierungswerkzeugs bildet die Eclipse Rich Client Plattform (RCP). Die Ideen der Rich Client Platform ist es, wiederkehrende Aufgaben die graphische Anwendungen gemeinsam haben, wie beispielsweise das grundsätzliche Layout des anzuzeigenden Fensters, Statusleisten oder Menüeinträge, die aus nahezu allen Anwendungen bekannt sind, im Rahmen der Rich Client Platform zu kapseln. Den für die jeweilige Anwendung spezifischen Teil können Entwickler in Form von so genannten Plugins beisteuern. Darüber hinaus bietet dieser Ansatz den Vorteil, dass bestehende Plugins in eigene grafische Anwendungen einfach integriert werden können. Abbildung 6-2 stellt diesen Ansatz grafisch dar.

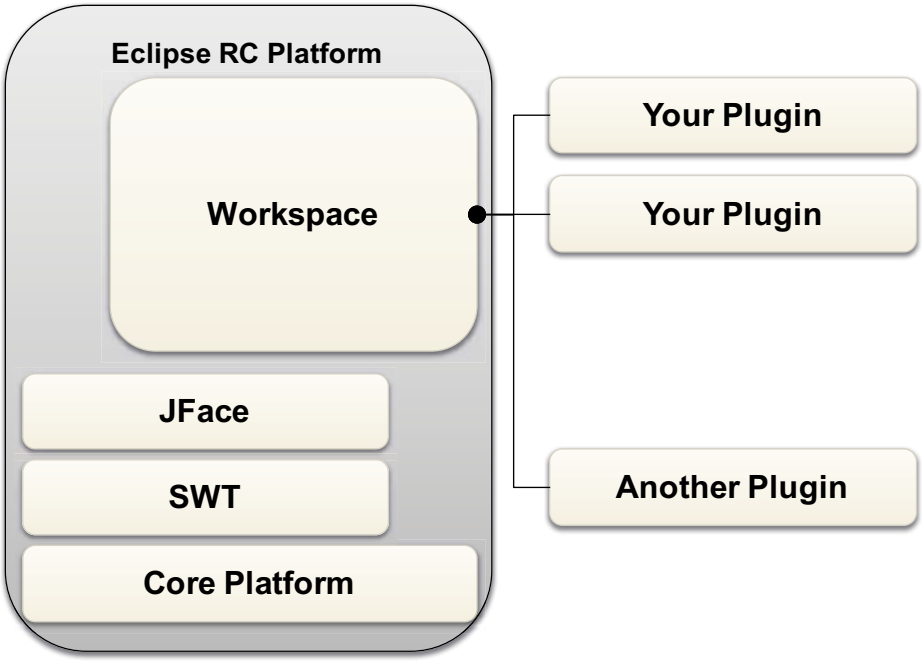


Abbildung 6-2 Architektur Eclipse Rich Client Platform
Quelle: Eigene Darstellung in Anlehnung an (Rogalski 2005)

Die Eclipse Rich Client Plattform besteht in ihrer Architektur aus vier Schichten. Die unterste Schicht ist die Core Plattform, welche durch das bereits angesprochene komponentenorientierte Framework Equinox realisiert wird. Die in Equinox realisierten Bundles umfassen dabei die Möglichkeit, Code, Ressourcen und Metainformationen bereitzustellen sowie Applikationen, Services und Pakete zu verwalten, die durch andere Bundles wiederum verwendet werden können. Darüber hinaus können Bundles auch einfach versioniert werden (Rogalski 2005).

Die nächste Schicht stellt das Standard Widget Toolkit (SWT) dar. SWT ist analog zu Swing eine in Java geschriebene Bibliothek für die Gestaltung grafischer Benutzeroberflächen, welche stark mit den darunter liegenden, nativen Funktionen des Betriebssystems verknüpft ist. Dies hat zur Folge, dass in SWT realisierte Applikationen in ihrer Gestaltung identisch mit nativen Anwendungen des jeweiligen Betriebssystems sind. Allerdings stellt SWT einen plattformunabhängigen Ansatz dar, so dass in SWT realisierte Applikationen auf unterschiedlichen Systemen verfügbar gemacht werden können.

Die dritte Schicht stellt eine weitere Abstraktion der grafischen Bibliothek SWT dar. Aufbauend auf dem Grundgedanken der Rich Client Plattform, wiederkehrende Elemente und

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Aufgaben im Rahmen der Plattform zu kapseln, bietet JFace komplexe grafische Elemente wie Dialoge, Wizards oder Rich Text Editoren an.

Die vierte Schicht stellt den eigentlichen Workspace dar. Sie besteht im Wesentlichen aus drei Elementen (Vogel 2010): Das *Hauptprogramm* stellt den Einstiegspunkt in eine RCP Applikation dar und ist in ihrer Funktionsweise in der Regel für die meisten Applikationen identisch. Neben dem Hauptprogramm gibt es noch einen sogenannten *Workbench Advisor*, der verwaltende Aufgaben wie das Erscheinungsbild einer Applikation, also die Gestaltung der Menüs, Toolbars, etc. übernimmt. Das letzte Element sind *Perspektiven*. Perspektiven legen die Aufteilung der Oberfläche der Anwendung, insbesondere die Darstellung von Views und Editoren, fest. Views und Editoren sind Bereiche, in denen die Nutzer mit der Anwendung interagieren. Ein View ist dabei ein passives Element, dessen Inhalt der Nutzer nicht direkt verändern kann, ein Editor hingegen dient gerade dieser Änderung von Inhalten.

Wie in Abbildung 6-2 zu sehen, kann die Funktionalität dieser Architektur durch eigene oder fremde Plugins erweitert werden. Hinsichtlich der Gestaltung eines Modellierungswerkzeugs für ASML ist es also notwendig, einerseits einen Workspace mit den entsprechenden Views für eine Verwaltung der Modelle und Unterstützung der Modellierung zu gestalten, als auch den eigentlichen Editor in Form eines Plugins verfügbar zu machen. Die ASML Workbench umfasst somit den linken Teil der oben dargestellten Architektur und nutzt den Editor, welcher in Abschnitt 6.2.2 diskutiert wird, zum Editieren und Darstellung von Automotive Service Modellen.

6.2.1.2 Umsetzung

Die Gestaltung der ASML Workbench gliedert sich in vier Bereiche. Der obere, in Abbildung 6-3 mit 1 gekennzeichnete Bereich umfasst die Menüleiste und die Toolbar der Workbench. Er erstreckt sich über die gesamte Breite der Anwendung. Beide Elemente haben die Aufgabe, Funktionen, die sich entweder auf einzelne Bereiche der Workbench oder auf die Workbench insgesamt beziehen, zur Verfügung zu stellen. Der zweite Bereich ist der so genannte Model Explorer. In diesem Bereich werden die in der Workbench verfügbaren Modelle in alphabetischer Reihenfolge angezeigt und der Status ihrer Validität durch ein, dem Namen vorangestelltes, Symbol gekennzeichnet. Das mit Outline bezeichnete Übersichtsfenster ist der dritte Bereich der Workbench. Das jeweils aktive Modell wird hier als Übersicht dargestellt, so dass der Nutzer bei einer Vergrößerung des Modellausschnitts einen Überblick über das Gesamtmodell behält. Der vierte und letzte Bereich ist der Editorbereich. Hier werden die in 6.2.2 vorgestellten Editorfenster dargestellt.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

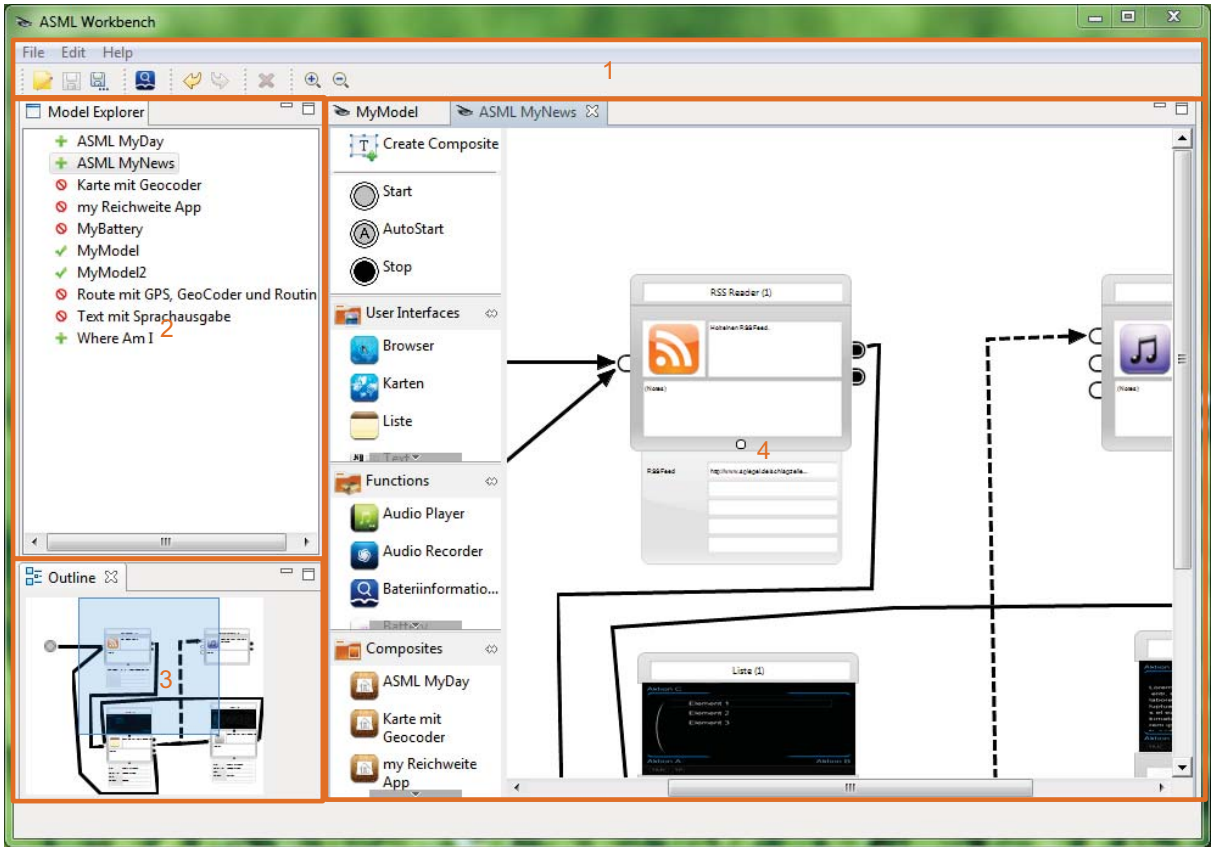


Abbildung 6-3 Bereiche der Workbench
Quelle: Eigene Darstellung

6.2.1.2.1 Toolbar

In der Toolbar werden je nach Kontext die einzelnen Elemente der Toolbar aktiv oder inaktiv dargestellt. Beispielsweise ist der Löschen-Knopf nur verfügbar, wenn entweder ein oder mehrere Modelle im Model Explorer oder Elemente im aktiven Modell ausgewählt sind. Von links nach rechts umfasst die Toolbar folgende Aktivitäten:

- *Anlegen eines neuen Modells:* Betätigen Nutzer diese Funktion, erscheint ein Popup-Fenster in dem der Name des anzulegenden Modells angegeben werden muss. Die ASML Workbench stellt dabei sicher, dass der gewählte Name noch nicht verwendet wurde, so dass ein versehentliches Überschreiben existierender Modelle verhindert wird. Mit dem Anlegen eines neuen Modells wird automatisch die entsprechende Datei zum Speichern des Modells im Dateisystem angelegt.
- *Speichern:* Diese Funktion speichert das aktuell aktive Modell. Wird das Modell oder Teile davon in anderen geöffneten Modellen als Composite verwendet, so werden diese über das Speichern des Modells informiert um gegebenenfalls auf inhaltliche Änderungen reagieren zu können.
- *Speichern unter:* Mithilfe dieser Funktion kann ein existierendes Modell unter einem neuen Namen abgespeichert werden. Das alte Modell bleibt in diesem Fall weiterhin existent und es wird eine Kopie angelegt.
- *Brick Wizard:* Der Brick Wizard ist eine Komponente, mit dessen Hilfe Nutzer Stubs definieren und konfigurieren können. Die detaillierte Funktionsweise des Brick Wizard wird in Abschnitt 6.2.1.3.5 erläutert.
- *Undo / Redo:* Mit Hilfe der Funktionstasten Undo und Redo können einzelne Aktionen, die der Nutzer am Modell durchgeführt hat, rückgängig gemacht werden.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Die eigentliche Umsetzung dieser Funktion ist Aufgabe des Editors, so dass die Semantik dieser Funktionstasten vom jeweils aktiven Editor abhängt.

- *Löschen:* Mit Hilfe dieser Funktion können ganze Modelle oder einzelne Elemente eines Modells gelöscht werden. Anzumerken ist dabei, dass das Löschen eines Modells im Gegensatz zum Löschen einzelner Elemente eines Modells nicht rückgängig gemacht werden kann.
- *Zoom:* Durch Zoom kann die Vergrößerung des aktiven Modells verändert werden. So ist es zum Beispiel möglich, in einem komplexen Modell Einzelbereiche detailliert darzustellen oder einen Gesamtüberblick über das Modell zu erhalten.

Die Menüleiste umfasst im Wesentlichen die gleichen Elemente wie die Toolbar. Die meisten Funktionen können so auch über Tastenkombinationen ausgeführt werden. Zusätzlich befinden sich im Menü noch Elemente zum Schließen von einzelnen oder mehreren Modellen, zum Beenden der ASML Workbench sowie eine About-Funktion.

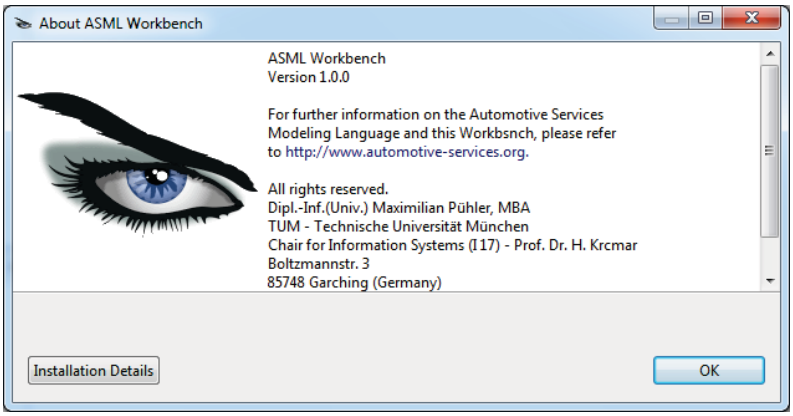


Abbildung 6-4 Anzeige der About-Funktion der ASML Workbench
Quelle: Eigne Darstellung

6.2.1.2.2 Model Explorer

Der Model Explorer dient der Verwaltung der zur Verfügung stehenden Modelle. Diese werden, alphabetisch sortiert, als flache Liste dargestellt. Durch Selektion einzelner oder mehrerer Elemente können diese für das Löschen über die Toolbar markierte werden. Darüber hinaus können Nutzer durch einen Doppelklick auf einen Modelleintrag diesen auf der rechten Seite der Workbench in einem Editor zur weiteren Bearbeitung öffnen.

Eine zweite wesentliche Aufgabe im Model Explorer ist, die Nutzer über die Qualität der einzelnen Modelle zu informieren. Wie in Abbildung 6-3 zu sehen, befinden sich vor dem Namen jedes einzelnen Modells jeweils ein Symbol, welches den Zustand des Modells wiedergibt. Ein grünes Plus gibt an, dass das Model syntaktisch in Ordnung ist und über ein Start Element verfügt. Das Modell wird somit von der Runtime als gültiges Modell erkannt und dem Nutzer zur Ausführung angeboten. Zwei kreisförmig angeordnete grüne Pfeile bedeuten analog zum grünen Plus, dass es sich um ein valides, ausführbares Modell handelt. Die Pfeile geben dabei an, dass das Modell einen Autostart enthält und von der Runtime somit automatisch ausgeführt wird. Ein grüner Haken schließlich beschreibt ein gültiges aber nicht ausführbares Modell. Diese können beispielsweise als Composites genutzt werden. Ein rotes Stoppzeichen schließlich symbolisiert ein fehlerhaftes Modell. Diese werden von der Runtime ignoriert und können nicht ausgeführt werden. Anzumerken ist, dass ein Modell auch als fehlerhaft markiert wird, wenn ein darin verwendetes Composite fehlerhaft ist.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

6.2.1.2.3 Outline

Die Links unten befindliche Outline dient dem Nutzer zur Übersicht über komplexe Modelle. In diesem Bereich wird stets das gesamte Modell angezeigt, unabhängig in welcher Zoom Stufe sich der dazugehörige Editor findet. Sind mehrere Editoren geöffnet, zeigt diese Ansicht stets den Inhalt des aktiven Editors an. Der blaue Kasten, der Transparent über das Modell eingezeichnet wird, zeigte den aktuell sichtbaren Bereich des Modells im Editor. Durch Verschieben des blauen Kastens mit der linken Maustaste kann der im Editor angezeigte Ausschnitt verschoben werden.

6.2.1.2.4 Editor Bereich

Der Editorbereich auf der rechten Seite der ASML Workbench stellt den Hauptarbeitsbereich dieser dar. Wie bereits erwähnt, wird die Funktionalität des Editors in Form eines Plugins realisiert, so dass dieser unabhängig von der ASML Workbench entwickelt und verwendet werden kann. Innerhalb der Workbench können beliebig viele Editoren gleichzeitig geöffnet und auf der rechten Seite in Form von Tabs angezeigt werden. Dieser Bereich dient also dazu, einen Platzhalter für Editoren in der Workbench zu schaffen und, auf der technischen Seite, Zugriff auf die Workbench zu erlauben.

6.2.1.2.5 Brick Wizard

Das in Abschnitt 5.1.1.2 besprochene Designprinzip der Stubs fordert, dass Stakeholder in der Lage sind, neue, bislang noch nicht in der Modellierung verfügbare domänenspezifische Konzepte auf eine einfache Art und Weise nutzbar zu machen. Darüber hinaus soll diese Spezifikation neuer Stubs alle Informationen beinhalten, die von technischen Experten gebraucht werden, diese im domänenspezifischen Framework zu realisieren und somit einen neuen Brick zu schaffen. Um dieser Anforderung gerecht zu werden, verfügt die ASML Workbench über ein unterstützendes Werkzeug, mit dessen Hilfe Stakeholder genau diese Aufgabe erfüllen können.

Mit dem Brick Wizard werden Nutzer Schritt für Schritt durch die Definition eines neuen Stubs geführt. Als Ergebnis dieser Aktivität wird eine formale Beschreibung erstellt, die Grundlage für eine Umsetzung im domänenspezifischen Framework ist. Darüber hinaus können diese Stubs direkt in Modellen eingesetzt werden. Der Brick Wizard erfasst diese Informationen in vier Schritten: Allgemeine Angaben erfassen, Properties definieren, Eingangskonnektoren anlegen und Ausgangskonnektoren anlegen. Sollte eine dieser Aktivitäten für ein bestimmtes Stub nicht notwendig seien, da dieses beispielsweise über keine Properties verfügt, können einzelne Schritte mit *„Next“* übersprungen werden. Sind alle Informationen vorhanden, wird der Wizard mit *„Finish“* beendet.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

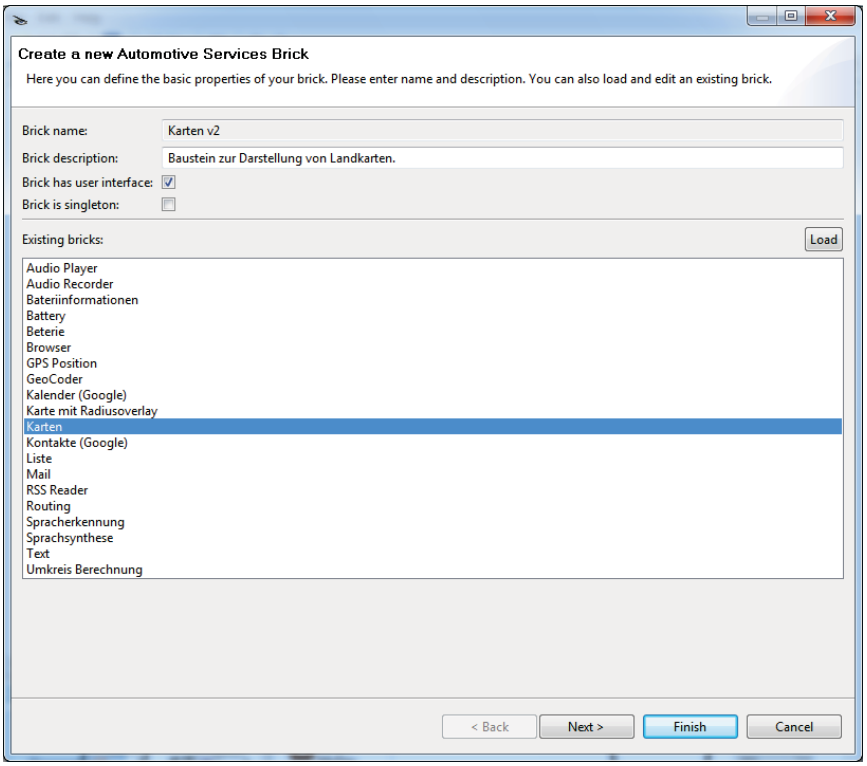


Abbildung 6-5 Erstellen von Stubs: Brick Wizard I
Quelle: Eigene Darstellung

Wie in Abbildung 6-5 zu sehen, werden im ersten Schritt vier allgemeine Informationen zum zu erstellenden Stub vom Nutzer erfragt. Hierzu zählen der Name des Stubs, eine Beschreibung sowie die Information ob es sich beim Stub um ein Bedienkonzept oder um Singleton handelt. Diese Informationen können entweder im oberen Teil des Wizards manuell eingegeben oder aus einem existierten Brick übernommen werden. Um letzteres durchzuführen, wählt der Nutzer den entsprechenden Brick in einer Liste aus und lädt dessen Informationen über die *„Load“*-Funktion. Neben den allgemeinen Informationen werden dadurch auch Informationen der weiteren Schritte aus dem existierten Brick übernommen. Da es sich in diesem Fall um die Weiterentwicklung eines bestehenden Konzepts handelt, kann dessen Name nicht vom Nutzer verändert werden. Stattdessen wird der Name des geladenen Bricks übernommen und durch Anhängen einer Versionsnummer erweitert.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

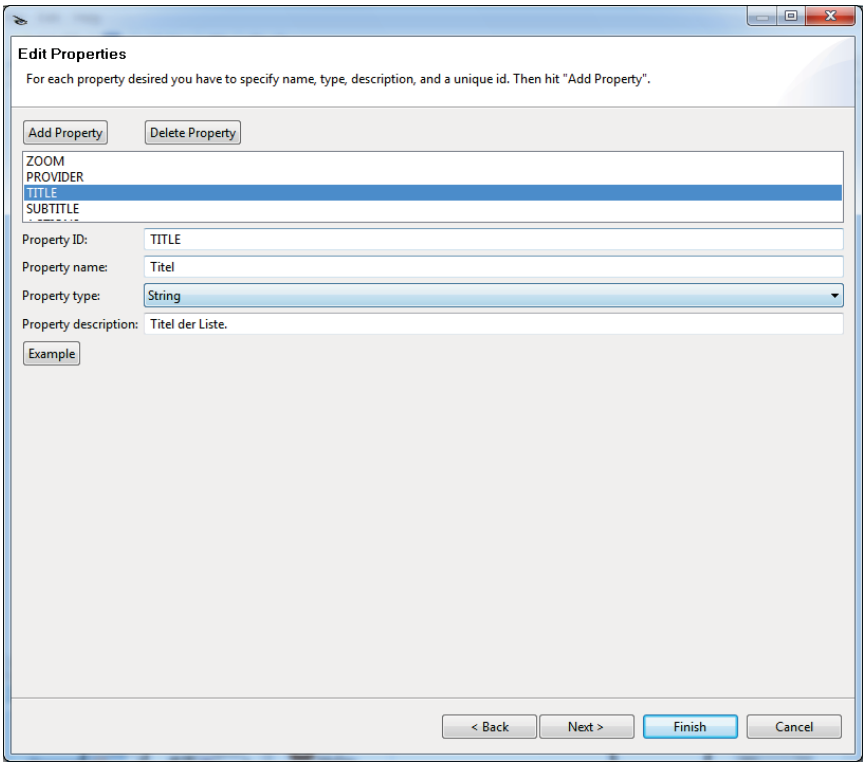


Abbildung 6-6 Erstellen von Stubs: Brick Wizard II
Quelle: Eigene Darstellung

Im zweiten Schritt können die Properties des Stubs definiert werden. Im oberen Fenster wird eine Liste aller Properties angezeigt, die durch die Funktionstasten *„Add Property“* und *„Delete Property“* geändert werden kann. Um einen neuen Property hinzuzufügen, müssen die vier dafür notwendigen Parameter angegeben werden. Die ID stellt eine eindeutige Bezeichnung des Properties dar und wird von den Entwicklern für die Umsetzung im domänenspezifischen Framework benötigt. Der Name hingegen beschreibt, wie der Property im Editor bezeichnet werden soll. Den Inhalt eines entsprechenden Tooltipps mit weiterführenden Informationen kann der Nutzer über die Description angeben. Schließlich muss noch der Datentyp das Property ausgewählt werden. Hierzu stehen über ein Dropdown Menü unterschiedliche Optionen zur Verfügung.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

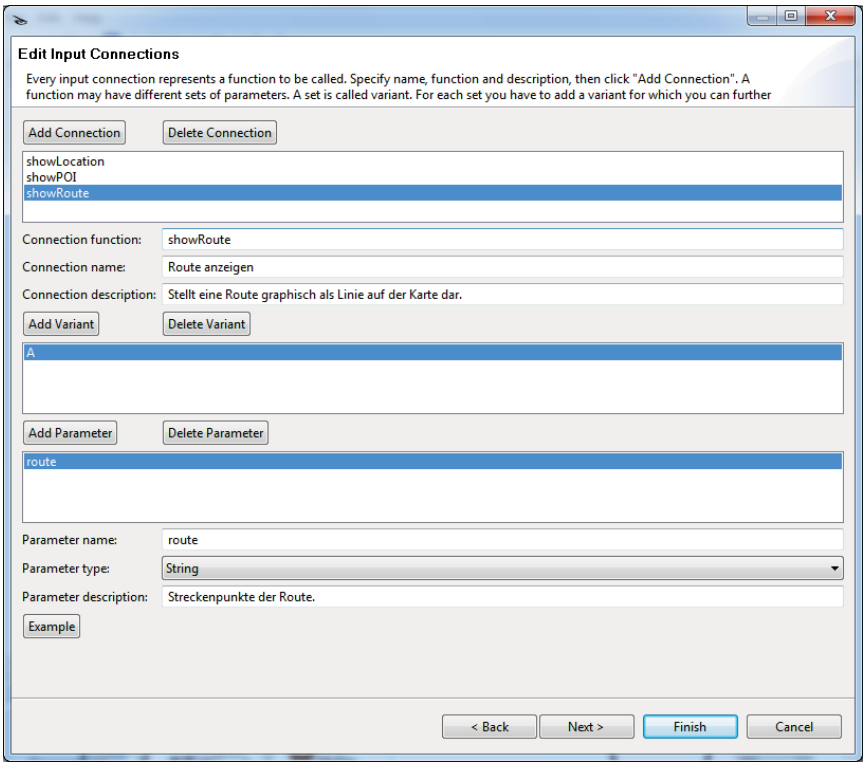


Abbildung 6-7 Erstellen von Stubs: Brick Wizard III
Quelle: Eigene Darstellung

Als nächstes werden die Eingangskonnektoren des Stubs konfiguriert. Ein Konnektor wird dabei durch eine Funktion, einen Namen und eine Beschreibung definiert. Die Funktion ist analog der ID der Properties für die Umsetzung im domänenspezifischen Framework notwendig. Name und Beschreibung geben wiederum die Bezeichnung in der Modellierungsumgebung sowie den entsprechenden Tooltipp an. Des Weiteren können für einen Eingangskonnektor unterschiedliche Varianten definiert werden. Eine Variante in diesem Zusammenhang bedeutet, dass der Konnektor zwar den gleichen Namen und die gleiche Funktion repräsentiert, jedoch einen unterschiedlichen Satz an Parametern benötigt. Ein Beispiel für ein Eingangskonnektor mit zwei Varianten ist der Listen Brick. Hier gibt es den Konnektor *„Liste anzeigen“*, der entweder einen Parameter mit der Liste der Elemente oder zwei Parameter mit der Liste der Elemente und dem Index des zu fokussierenden Elements aufgerufen werden kann. Der Eingangskonnektor hat also zwei Varianten mit einmal einem und einmal zwei Parametern. Diese Parameter können im letzten Bereich dieses Schritts im Wizard angegeben werden. Ein Parameter verfügt über einen Namen, eine Beschreibung sowie eine Datentyp.

Im letzten Schritt werden die Ausgangskonnektoren definiert. Ein Ausgangskonnektor besteht aus einer ID, einem Namen einer Beschreibung (Tooltipp) sowie einer Liste an Parametern. Diese Parameter repräsentieren die Ergebnisse, die beim Anlegen eines Signals an diesen Ausgangskonnektor an den nachfolgenden Stream übergeben werden. Parameter bestehen ihrerseits aus einer ID, einem Namen, einer Beschreibung und einem Datentypen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

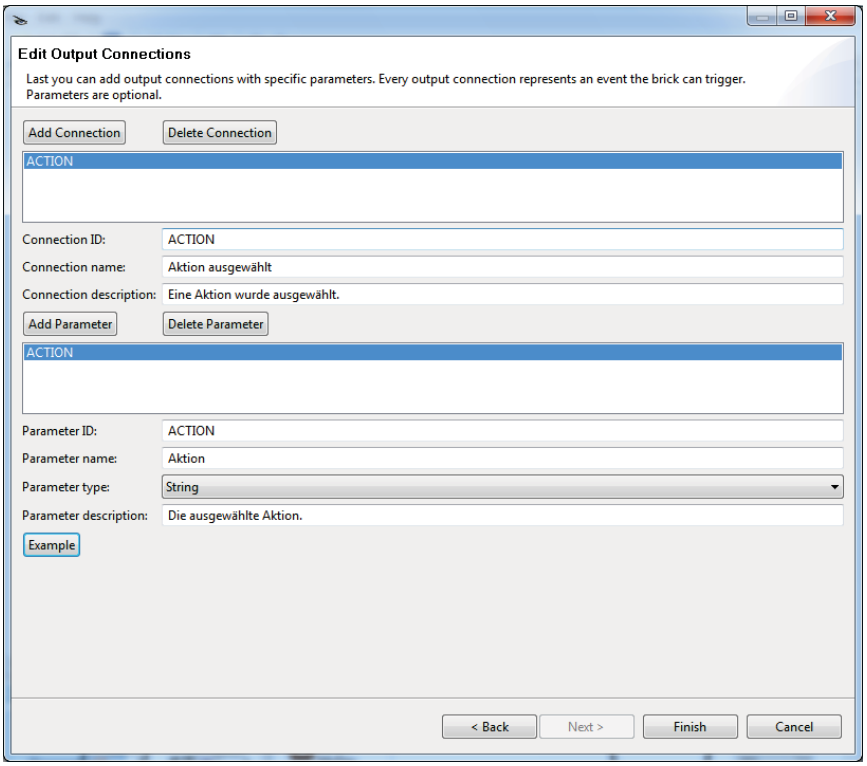


Abbildung 6-8 Erstellen von Stubs: Brick Wizard IV
Quelle: Eigene Darstellung

6.2.1.3 Architektur

Die Architektur der Workbench orientiert sich an der vorgestellten Referenzarchitektur einer Eclipse Rich Client Platform. Der Hauptbereich der Anwendung sowie die Gestaltung der Perspektiven wird, wie in Abbildung 6-9 zu sehen, im Workbench Paket realisiert. Das dritte Element der Referenzarchitektur, der Workbench Advisor, befindet sich im Advisor Paket. Zusätzlich zu diesen beiden Paketen existiert noch ein View Paket, welches den Umfang des Model Explorers beschreibt sowie ein Wizard Paket, in dem sich die Realisierung des Brick Wizard findet. Das letzte Paket beherbergt eine Sammlung einzelner Aktionen, wie beispielsweise das Anlegen eines neuen Modells, welche von der Workbench zur Verfügung gestellt werden.

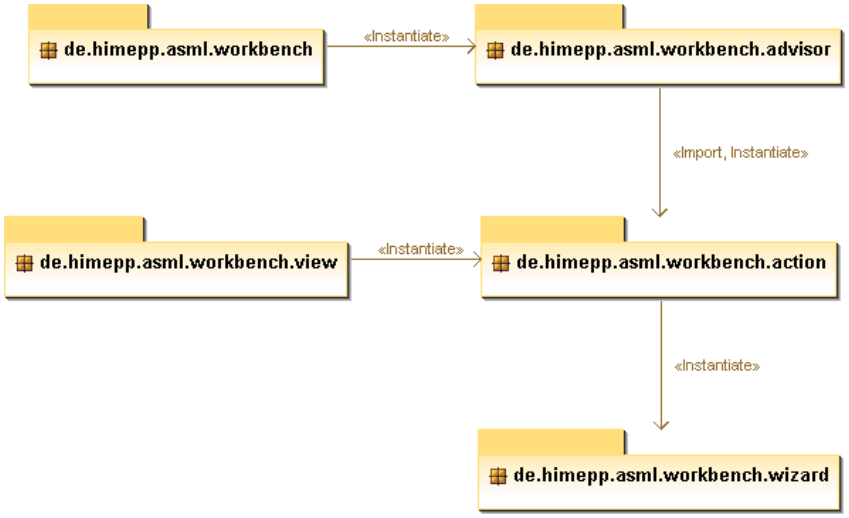


Abbildung 6-9 Übersicht der Bestandteile der Workbench
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Die nachfolgenden Abschnitte geben einen Überblick über die in diesen Paketen enthaltenen Klassen sowie eine Beschreibung deren wesentlichen Aufgaben und Funktionen.

6.2.1.3.1 Workbench

Wie bereits erwähnt ist es die Aufgabe des Workbench Pakets die grundlegenden Bestandteile der Rich Client Platform zu realisieren. Zu diesem Zweck besteht das Packet aus insgesamt drei Klassen. Die Application Klasse ist gewissermaßen die zentrale Klasse der Applikation, da sie sich um das Starten und Stoppen der ASML Workbench kümmert. Da es sich bei der Workbench streng genommen selber um ein Plugin aus Sicht des RCP Frameworks handelt, dient die Activator Klasse dazu, den Lebenszyklus dieses Plugins zu verwalten. Die dritte Klasse, Perspective, legt schließlich die Darstellung und Anordnung der einzelnen Elemente auf der graphischen Oberfläche fest.

Abbildung 6-10 Übersicht der Bestandteile des Pakets Workbench
Quelle: Eigene Darstellung

6.2.1.3.2 Advisor

Das Advisor Paket enthält ebenfalls drei Klassen, deren Aufgabe es ist, Funktionalitäten der RCP Applikation zu definieren. Der ApplicationWorkbenchAdvisor initialisiert den ApplicationWorkbenchWindowAdvisor, legt die zu verwendende Perspektive fest und definiert, ob die Anwendung ihr Layout beim Beenden speichern soll. Der Application-WorkbenchWindowAdvisor initialisiert wiederum den ApplicationActionBarAdvisor und entfernt darüber hinaus unerwünschte Elemente aus dem Menü und der Toolbar, die durch andere Plugins automatisch eingetragen werden können. Schließlich konfiguriert der ApplicationActionBarAdvisor die in der Workbench verfügbaren Funktionen, wie z.B. das Anlegen eines neuen Modells, sowie die entsprechenden Einträge in der Toolbar und im Menü.

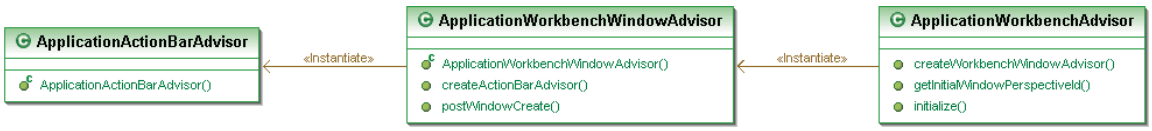


Abbildung 6-11 Übersicht der Bestandteile des Pakets Advisor
Quelle: Eigene Darstellung

6.2.1.3.3 Action

Das Action Paket implementiert die im Advisor Paket festgelegten Funktionen der Workbench. Da einige allgemeine Funktionen, wie Beenden oder Zoomen, bereits zum Umfang des RCP Frameworks gehören, müssen hier nur noch drei Aktionen definiert werden. Im Interface ICommandIds werden diese Aktionen mit einer eindeutigen ID referenziert. Wie in Abbildung 6-12 zu sehen, werden die Aktionen DeleteModelAction (Löschen eines Modells), NewModelAction (Anlegen eines neuen Modells) sowie BrickWizardAction (Starten des Brick Wizard) in diesem Paket realisiert.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

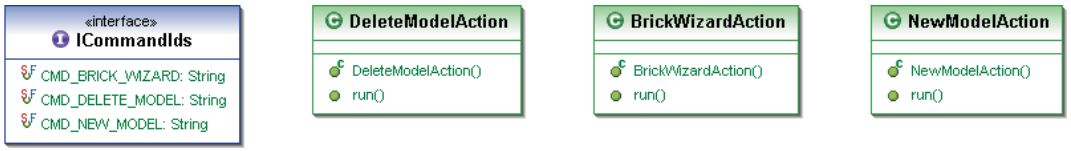


Abbildung 6-12 Übersicht der Bestandteile des Pakets Action
Quelle: Eigene Darstellung

6.2.1.3.4 View

Das View Paket umfasst den Model Explorer. Die Klasse ModelExplorerView erweitert die vom Framework bereitgestellte allgemeine ViewPart Klasse und implementiert sowohl die Darstellung als auch die Funktionalität des Model Explorers. Die beiden Hilfsklassen ModelExplorerContentProvider und ModelExplorerLabelProvider werden für das Erstellen der anzuzeigenden Informationen genutzt. Hier ist besonders anzumerken, dass der ModelExplorerLabelProvider auch das überprüfen der Modelle auf Validität initialisiert.

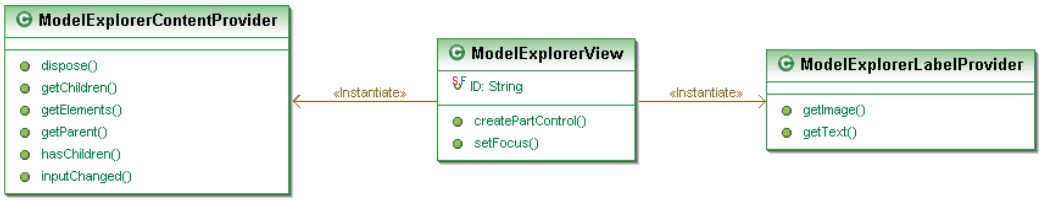


Abbildung 6-13 Übersicht der Bestandteile des Pakets View
Quelle: Eigene Darstellung

6.2.1.3.5 Wizard

Schließlich umfasst das fünfte und letzte Paket die Realisierung des Brick Wizard. Die zentrale Klasse BrickWizard wird durch die BrickWizardAction im Action Paket aufgerufen und übernimmt die Konfiguration und Durchführung des Wizards. Darüber hinaus existieren in dieser Klasse interne Klassen, die zum Speichern der zu erstellenden Informationen dienen: Connection, Parameter, Property und Variant.

Für die Darstellung der einzelnen Seiten des Wizards steht jeweils eine Klasse zur Verfügung. Die Aufgabe dieser Klassen ist es, einerseits das Layout der jeweiligen Seite zu gestalten und andererseits die vom Nutzer bereitgestellten Informationen im BrickWizard zu speichern. Wird der Wizard beendet, werden die gesammelten Informationen mit Hilfe des XmlWriters abgespeichert. Die Klasse selber verfügt über keine semantische Information hinsichtlich des Speicherformats für Stubs. Diese Informationen werden ebenfalls vom BrickWizard realisiert.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

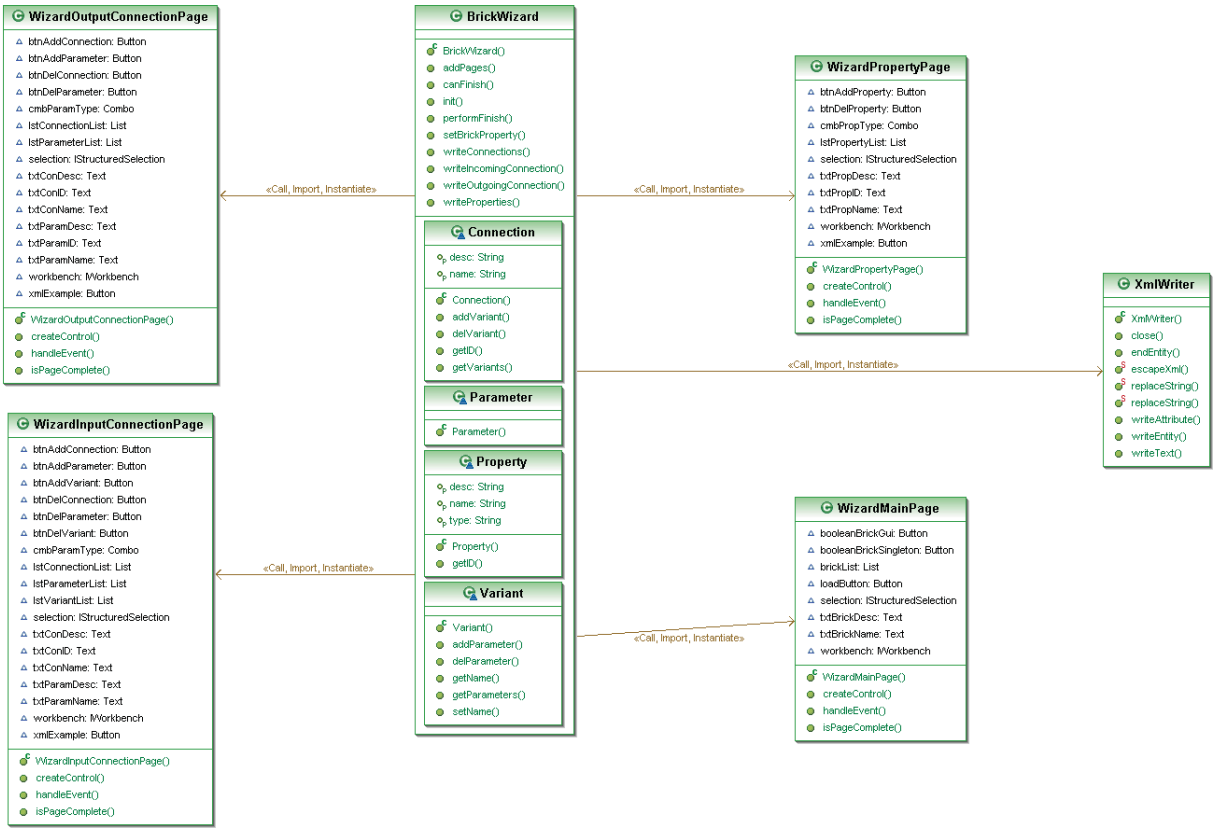


Abbildung 6-14 Übersicht der Bestandteile des Pakets Wizard
Quelle: Eigene Darstellung

6.2.2 Modellierungswerkzeug: Editor

Das zentrale Element der Modellierungsunterstützung ist der Editor. Der ASML Editor ist ein grafisches Gestaltungswerkzeug für Automotive Service Modelle die mit der Automotive Services Modeling Language modelliert sind. Für die Nutzung von Composites und die Unterstützung der Nutzer durch weiterführende Funktionen wie das Verwalten mehrere Modelle ist der Editor auf eine enge Integration in die, in Abschnitt 6.2.1 beschriebene, Workbench angewiesen. Ausgehend von den technologischen Grundlagen werden im Nachfolgenden einerseits die Umsetzung, also die Darstellung und Interaktion des Editors mit den Nutzern, als auch die Architektur und somit softwaretechnische Realisierung des Editors beschrieben.

6.2.2.1 Grundlagen

Bei der Gestaltung von grafischen Editoren ist man mit einem grundsätzlichen Problem konfrontiert. Auf der einen Seite existiert ein Modell, also eine Repräsentation der zu bearbeiten Informationen im Arbeitsspeicher, und auf der anderen Seite die Darstellung dieses Modells auf dem Bildschirm. Interagiert der Nutzer nun mit der Darstellung auf dem Bildschirm, so müssen diese Aktionen zum einen in Modifikationen des Modells und zum anderen in Modifikationen der Darstellung umgesetzt werden. Die Folge dieser Situation sind Inkonsistenzen zwischen Modell und Darstellung, da die gleiche Aktion zweimal in unterschiedlichen Umgebungen umgesetzt werden muss. Abbildung 6-15 verdeutlicht diesen Zusammenhang.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

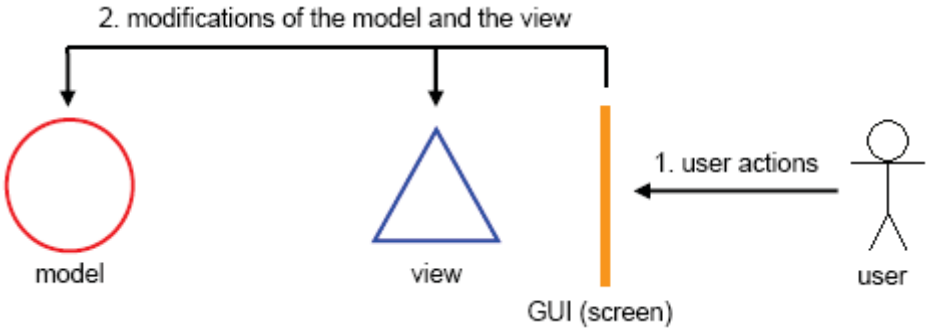


Abbildung 6-15 Problemstellung grafischer Editoren
Quelle: (Eclipse Foundation 2010b)

Das Graphical Editing Framework (GEF) der Eclipse Foundation adressiert genau dieses Problem mit einem Model-View-Controller Ansatz. Wie in Abbildung 6-16 zu sehen, werden die Aktionen des Nutzers mit dem Bildschirm nicht direkt in Modifikation des Modells und der Darstellung übersetzt, sondern zunächst nur das Modell aktualisiert, welches wiederum eine Aktualisierung der Darstellung nach sich zieht.

Konkret bedeutet dies, dass das Modell alle wesentlichen Informationen, die vom Nutzer editiert werden können, beinhalten muss. Da keine direkten Änderungen der Darstellung erfolgen, müssen auf der Seite des Modells alle relevanten Informationen gespeichert werden. Des Weiteren bedeutet dieser Ansatz, dass das Modell keinerlei Information über die eigentliche Darstellung benötigt. Das Modell propagiert lediglich die in ihm beinhalteten Informationen sowie Änderungen an diesen an die Darstellung. Auch darf es keinen Rückkanal von der Darstellung zurück zum Modell geben, um wiederum Inkonsistenzen zu vermeiden. Eine konkrete Beschreibung des in Editor verwendeten Modells findet sich in Abschnitt 6.2.3.

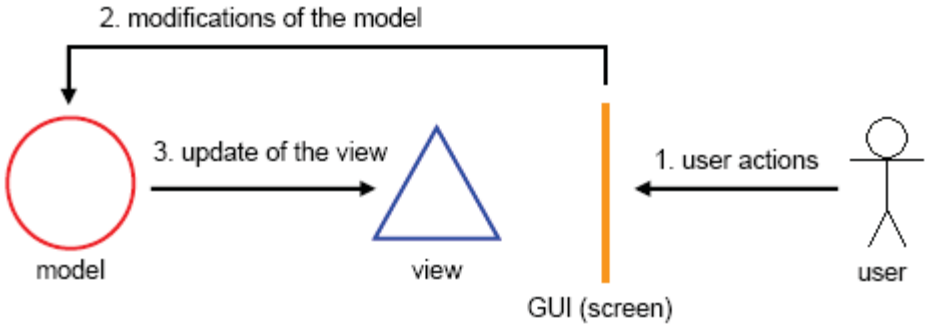


Abbildung 6-16 Model-View-Controller Ansatz GEF
Quelle: (Eclipse Foundation 2010b)

Die Darstellung der Modelle kann dieser Architektur entsprechend „dumm“ ausfallen. Die Elemente der Darstellung haben somit lediglich zwei Aufgaben. Zunächst stellen sie, angestoßen von Informationen des Modells, dieses grafisch für den Nutzer dar. Die zweite Aufgabe ist das Propagieren von Nutzeraktionen an das Modell. Für die in Abbildung 6-16 dargestellte Interaktion fehlen also noch die Bestandteile: Die die Modifikation des Modells realisieren (*„2. modifications of the model“*) sowie jene, die die Aktualisierung der Darstellung übernehmen (*„3. Update of the view“*).

Für das Propagieren der Aktionen der Nutzer an das Modell werden in GEF Policies und Commands eingesetzt. Mithilfe von Policies kann für die einzelnen Aktionen, die ein Nutzer auf einer Oberfläche durchführen kann, wie das Selektieren eines Elements, einen Doppelklick oder das Betätigen der Löschtaste, spezifiziert werden, welches Kommando als

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Folge der Aktion ausgeführt werden soll. Dieses Kommando wiederum realisiert die jeweilige Änderung am Modell. Darüber hinaus werden diese Kommandos einem Command Stack hinzugefügt, so dass diese sowohl ausgeführt als auch wieder rückgängig gemacht werden können und somit die Undo- und Redo-Funktionalität des Editors realisieren.

Das zweite verbindende Element sind EditParts. EditParts stehen zwischen dem Modell und der Darstellung und verbinden diese beiden Elemente miteinander. Ein EditPart hört auf Änderungen des Modells und ruft bei Bedarf entsprechende Informationen im Modell ab. Mithilfe dieser Informationen gestalteten EditParts die Darstellung des Modells auf der Grundlage der Views auf dem Bildschirm.

6.2.2.2 Umsetzung

Die Darstellung des Editors gliedert sich in drei Bereiche: Am oberen Rand des Editors befindet sich die Titelleiste, in der der Name des im Editor geöffneten Modells angezeigt wird. Der zweite Bereich auf der linken Seite des Editors ist die so genannte Toolbar. Die Aufgabe der Toolbar ist es, dem Nutzer die notwendigen Werkzeuge für die Gestaltung von Modellen zur Verfügung zu stellen. Der dritte Bereich, der in Abbildung 6-17 mit 3 markierte ist, ist der Canvas, also die Zeichenfläche, in der Modelle erstellt und dargestellt werden.

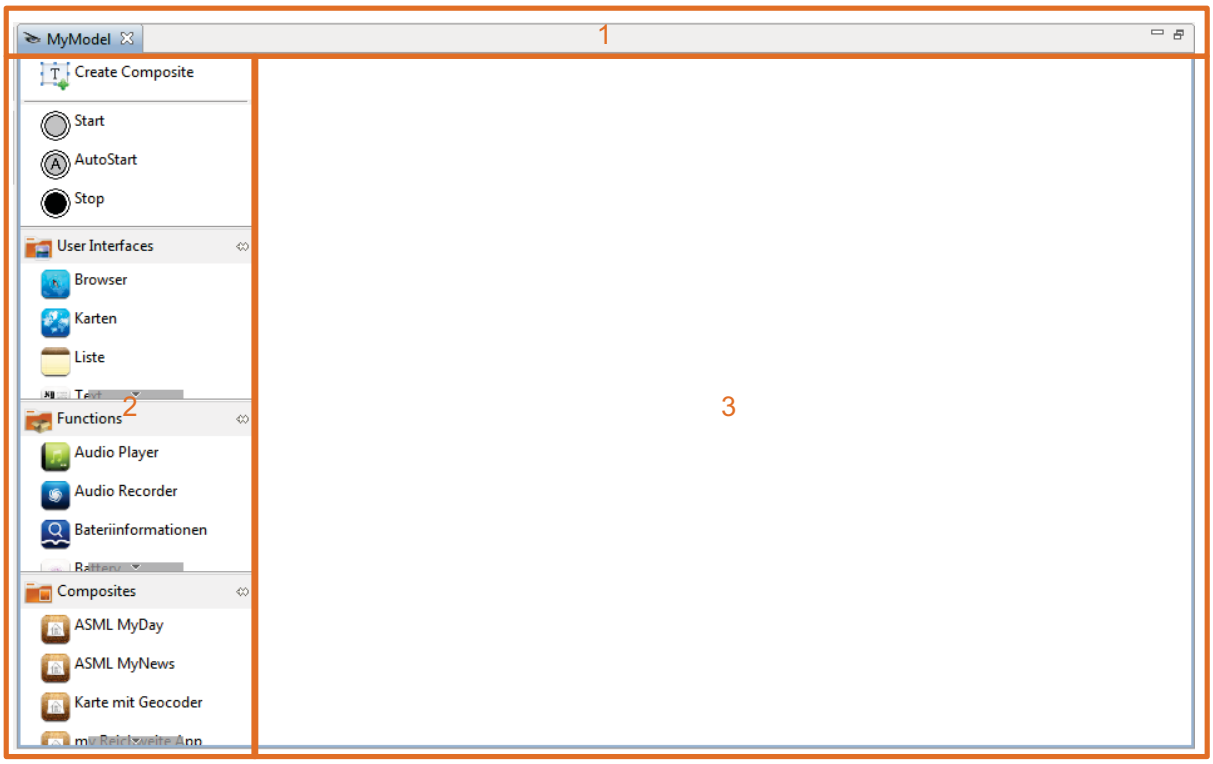


Abbildung 6-17 Bereiche des Editors
Quelle: Eigene Darstellung

Die Aufgabe der Titelleiste am oberen Rand ist die Anzeige des Namens des aktuell aktiven Modells. Darüber hinaus kann über den, rechts vom Namen befindlichen, Schließenknopf das Modell in der Workbench geschlossen werden. Hat sich das Modell in der Zwischenzeit verändert, was durch ein Sternchen vor dem Namen des Modells angezeigt wird, wird der Nutzer zum Speichern des Modells aufgefordert. Auf der rechten Seite der Titelleiste befinden sich Funktionstasten zum Minimieren und Maximieren des Editorfensters. Diese Funktionen signalisieren der Workbench, dem Editor entweder den gesamten Bereich der Workbench einzuräumen und andere Elemente zu verkleinern, oder aber dass der Editor selbst verkleinert

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

werden kann um anderen Elementen mehr Platz einzuräumen. Zusätzlich können in der Titelleiste mehrere geöffnete Editoren in Form von Tabs dargestellt werden. Durch Anklicken der einzelnen Tabs kann zwischen Editoren gewechselt werden. Die Titelleiste ist also ein verteiltes Element, welches für alle aktiven Editoren gemeinsam verfügbar ist.

Die auf der linken Seite befindliche Toolbar gliedert sich selber wiederum in fünf Bereiche. Der oberste Bereich enthält genau ein Element, welches die Nutzer zum Erstellen neuer Composites verwenden können. Im Bereich direkt darunter befinden sich die Elemente Start, Autostart und Stopp. Abgesehen vom Element zum Anlegen von Composites können alle Elemente der Toolbar per Drag'n'Drop auf den rechts befindlichen Canvas gezogen und so das entsprechende Element instanziiert werden. Sowohl der erst als auch der zweite Bereich der Toolbar hat eine feste Größe und kann vom Nutzer somit weder verschoben noch minimiert werden. Dies ist jedoch für die drei nachfolgenden Bereiche möglich. Ein Klick auf die Titelleiste minimiert oder maximiert diesen entsprechend dem aktuellen Zustand des Bereichs. Im dritten Bereich von oben befinden sich Bricks und Stubs, die Bedienkonzepte repräsentieren. Im Bereich darunter, dem vierten Bereich, solche, die Funktionskonzepte realisieren. Im letzten Bereich werden schließlich die Composites zur Verfügung gestellt. Als Composite ist grundsätzlich jedes Modell verfügbar, welches in der, den Editor umschließenden, Workbench angezeigt wird.

Der dritte Bereich, auf der rechten Seite, welcher den meisten Platz im Editor einnimmt, ist der Canvas, also die Zeichenfläche für Automotive Service Modelle. Elemente können dort durch Drag'n'Drop aus der Toolbar erstellt werden und mithilfe von Maus und Tastatur manipuliert werden. Zur Unterstützung des Modellierers stellt der Canvas eine Reihe an Funktionalität zur Verfügung. So werden Nutzer beim Layout ihrer Modelle durch ein, in der Anzeige nicht sichtbares, Gitter unterstützt. Elemente werden bevorzugt an den Kanten dieses Gitters ausgerichtet, was eine Ausrichtung der Elemente untereinander erleichtert. Zusätzlich werden den Nutzern noch Hilfslinien angezeigt, die automatisch erscheinen wenn ein Element, das der Nutzer gerade verschiebt, beispielsweise an der oberen Kante bündig mit dem linken Nachbarelement ist. Des Weiteren verfügt der Canvas über eine Zoom-Funktionalität, mit deren Hilfe der Nutzer die Darstellung des Modells vergrößern oder verkleinern kann. Diese kann einerseits über die Funktionstasten in der Toolbar der Workbench oder andererseits über das Drehen des Scrollrads auf der Maus bei gleichzeitigem Drücken der Steuerungstaste auf der Tastatur erreicht werden. Die Größe des Canvas ist dabei unbegrenzt. Elemente können also in alle vier Himmelsrichtungen beliebig weit verschoben werden. Durch Zoomen und Scrollen des Canvas kann innerhalb dieser Zeichenfläche navigiert werden. Die Größe der rechts und unten angeordneten Scrollleisten indiziert dabei die Größe des Canvas.

6.2.2.2.1 Arbeiten mit Bricks und Stubs

Zum Arbeiten mit Bricks und Stubs wird die in Abschnitt 5.1.2.1 beschriebenen Notation der Automotive Services Modeling Language verwendet. Zum Anlegen eines Bricks muss lediglich das Symbol aus der Toolbar auf den Canvas des Editors gezogen werden. Um Bricks wieder zu löschen oder auf dem Canvas zu verschieben, müssen diese durch einfaches Anklicken mit der Maus selektiert werden. Eine schwarze Umrandung signalisiert dann den selektierten Zustand. Ein Verschieben auf der Zeichenfläche erfolgt durch Verschieben mit gedrückter Maustaste, wobei auch ein Verschieben mehrere gleichzeitig markierter Elemente

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

möglich ist. Zum Löschen eines markierten Bricks wird entweder die Entfernentaste auf der Tastatur oder die Löschentaste in der Workbench genutzt.

Abbildung 6-18 Arbeiten mit Bricks
Quelle: Eigene Darstellung

Wie in der Notation bereits dargelegt, gliedert sich der Brick selber in mehrere Bereiche. Der obere Bereich, mit Titel und Vorschau ist vom Nutzer nicht Editierbar, die Anzeige der Vorschau kann allerdings durch einfaches Klicken auf das Kreissymbol unterhalb der Vorschau geöffnet beziehungsweise geschlossen werden. Ein längeres Verweilen des Mauszeigers über der Vorschau öffnet einen Tooltipp, in dem die Vorschau in vergrößerter Form dargestellt wird. Im mittleren Bereich kann der Nutzer eigene Notizen angeben. Durch einfaches Anklicken des Notizenbereichs erscheint direkt an dieser Stelle ein kleines Fenster zum Editieren der Notiz. Dies ist in Abbildung 6-18 weiter unten am Property Titel zu sehen. Analog zu den Notizen können also auch einzelne Properties direkt im Canvas bearbeitet werden. Auch ist ein Auf- und Zuklappen das Propertybereichs durch Anklicken des darüber befindlichen Kreissymbols möglich. Um zusätzliche Informationen zu den links und rechts befindlichen Konnektoren zu erhalten, kann der Nutzer, wie auch bei der Vorschau, in einem Tooltipp textuelle Informationen angezeigt bekommen. Hierzu muss wiederum der Mauszeiger über dem Konnektor eine längere Zeit verweilen.

6.2.2.2.2 Arbeiten mit Streams

Streams können im Editor direkt an den Ausgangskonnektoren der Bricks und Stubs erzeugt werden. Durch Drag'n'Drop eines Ausgangskonnektors auf einen Eingangskonnektor wird automatisch ein neuer Stream zwischen diesen beiden Elementen erzeugt. Der Verlauf des Streams wird dabei automatisch um bestehende Objekte herum geroutet, so dass ein Schneiden von Bricks durch Streams verhindert wird. Darüber hinaus kann der Nutzer den Stream manuell anpassen. Analog zu Bricks wird ein Stream durch einfaches Anklicken mit der Maus markiert. Markierte Streams werden dadurch kenntlich gemacht, das so genannte Bendpoints in ihrem Verlauf angezeigt werden. Eine Bendpoint ist ein Punkt, an dem der Nutzer die Route des Streams mit der Maus verändern kann. Durch Anklicken und Verschieben eines Bendpoints wird ein neuer Punkt angelegt, durch den der Stream verlaufen soll. Auf diese Weise kann die Übersichtlichkeit der Modelle verbessert werden.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Neben der grafischen Darstellung sowie der Aussage welcher Ausgangskonnektor mit welchen Eingangskonnektor verknüpft werden soll, können an einem Stream noch eine ganze Reihe an weiteren Informationen vom Nutzer angegeben werden. Diese Einstellungen können in einem Popupfenster vorgenommen werden, welches durch Doppelklick auf den Stream geöffnet wird. Im oberen Bereich dieses Fensters können die von Eingangskonnektor benötigten Parameter der verschiedenen Varianten mit den am Stream verfügbaren Informationen der vorangegangenen Bricks verknüpft werden. Hierzu wählt der Nutzer zunächst auf der linken Seite eine entsprechende Variante aus, was in einer Anzeige der für diese Variante notwendigen Parameter resultiert. Als nächstes wählt der Nutzer einen dieser Parameter per Mausklick aus. Auf der rechten Seite des Fensters erscheint automatisch eine Liste aller, zu diesem Parameter passenden, verfügbaren Informationen auf dem Stream. Diese sind sortiert nach den Ausgangskonnektoren der vorangegangenen Bricks, an denen sie auf den Stream geschrieben wurden. In Abbildung 6-19 ist beispielsweise zu sehen, das zur „Listenposition des Texts“ in Variante F der Parameter *„Position des ausgewählten Elements“* vom Ausgangskonnektor *„Listenelement wurde aktiviert“* aus dem Listen Brick passt. Durch Anklicken dieses Elements auf der rechten Seite wird der Parameter mit der entsprechenden Information verknüpft. Dies wird im Fenster verdeutlicht, indem beim Anklicken des Parameters auf der linken Seite automatisch das verknüpft Elemente auf der rechten Seite markiert wird.

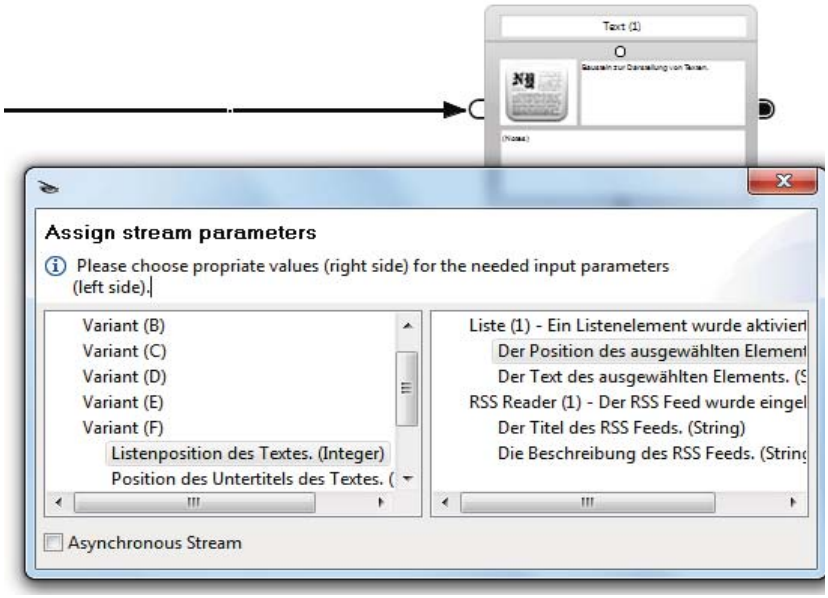


Abbildung 6-19 Anlegen von Streams
Quelle: Eigene Darstellung

Zusätzlich zur Verknüpfung der Parameter kann der Nutzer in diesem Fenster noch wählen ob ein Stream asynchron sein soll. Dies kann mithilfe des An- und Abwählens der links unten befindlichen Checkbox erfolgen. Informationen, die der Nutzer in diesem Fenster angibt, werden automatisch auf das Modell übertragen, so dass ein weiteres Bestätigen nicht notwendig ist. Das Fenster kann also durch Betätigen des Schließenknopfs rechts oben oder Drücken der Esc-Taste auf der Tastatur geschlossen werden.

6.2.2.2.3 Arbeiten mit Composites

Eines der zentralen Designprinzipien der Automotive Services Modeling Language sind Composites. Durch sie wird der Nutzer in die Lage versetzt, erprobtes Wissen, welches in

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Form von Automotive Service Modellen vorliegt, für zukünftige Entwicklungen nutzbar zu machen. Dies kann im Rahmen des Editors auf eine sehr einfache Art und Weise erfolgen. Der Nutzer klickt hierzu auf das *„Create Composite“* Symbol in der Toolbar und markierte anschließend mit der Maus alle Elemente des Modells, die das zu erstellende Composite umfassen soll. Der Editor öffnet daraufhin ein Popupfenster, in dem der Name des neuen Composites angegeben werden muss. Dieser Name kann vom Nutzer beliebig gewählt werden, darf allerdings nicht dem Namen eines bereits bestehenden Modells entsprechen, was vom Editor jedoch sichergestellt wird. Sobald der Nutzer den Namen mit OK bestätigt, werden die markierten Elemente aus dem aktuellen Modell entfernt und in ein neues Modell übertragen. Dieses Modell wird von der Workbench wie jedes andere Modell im Model Explorer angezeigt. An der Stelle der entfernten Elemente wird ein neues Composite eingefügt und alle Konnektoren, die sich am Rand des neuen Modells befinden, automatisch ausgewählt. Zusätzlich erscheint das Composite im unteren Bereich der Toolbar als neuer nutzbarer Baustein.

In ähnlicher Weise wird auch mit Streams verfahren. Alle Streams zwischen Elementen, die sich nun im neuen Modell befinden, werden ebenfalls inklusive aller Einstellungen auf das neue Modell übertragen. Streams hingegen, die vom ursprünglichen Modell zu Elementen des neuen Composite Modells führen, werden mit dem Composite verbunden. Aus Sicht des Modellierers wird also ein ganzer Bereich des Modells durch einen einzelnen Baustein ersetzt, dessen Konnektoren mit allen eingehenden und ausgehenden Streams entsprechend verknüpft werden. Abbildung 6-20 zeigt das Anlegen eines neuen Composites.

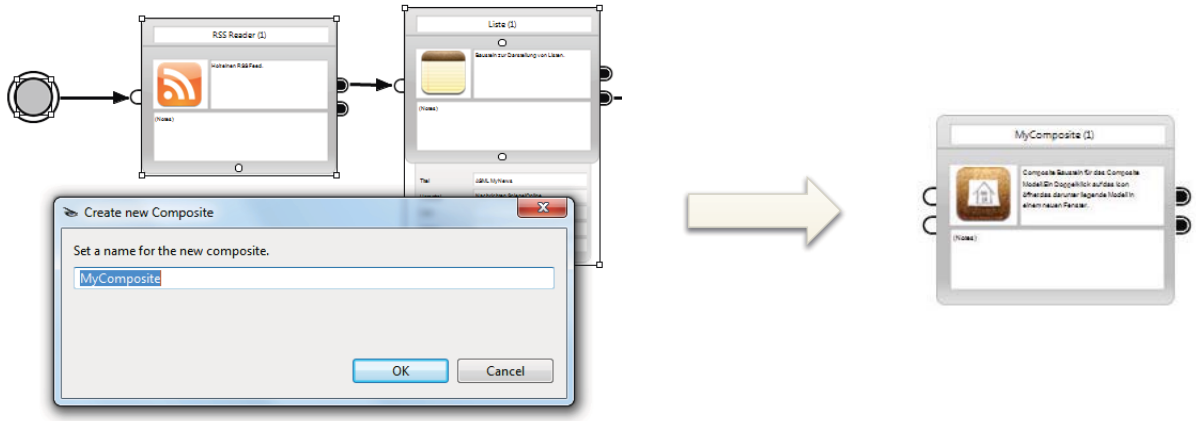


Abbildung 6-20 Anlegen von Composites
Quelle: Eigene Darstellung

Bis auf zwei Ausnahmen verhält sich ein Composite genauso wie ein gewöhnlicher Brick. Die erste Ausnahme ist das Festlegen der genutzten Konnektoren. Grundsätzlich stehen alle Konnektoren aller Elemente des darunter liegenden Modells zur Verfügung. Da dies unter Umständen sehr viele Elemente sein können, die auf der linken und rechten Seite des Konnektors dargestellt werden müssten, werden automatisch nur solche angelegt, die auch mit Streams verknüpft sind. Möchte ein Nutzer weitere Konnektoren des Composites nutzen, so kann er dies durch einen Doppelklick auf den Titel des Composites festlegen. Der Editor zeigt in diesem Fall ein Popupfenster an, auf dessen linken Seite alle verfügbaren Eingangskonnektoren und auf dessen rechten Seite alle verfügbaren Ausgangskonnektoren angezeigt werden. Durch Markieren dieser Elemente können Konnektoren zur Anzeige im Modell an- und abgewählt werden. Anzumerken ist, dass als Ausgangskonnektoren auf der rechten Seite nur solche angezeigt werden, die durch die gewählten Eingangskonnektoren auf

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

der linken Seite auch erreicht werden können. Abbildung 6-21 zeigt das Auswählen von Konnektoren am Beispiel des ASML MyNews Composites, welches das in der Case Study in Abschnitt 5.2.1 erläuterte MyNews Modell beschreibt.

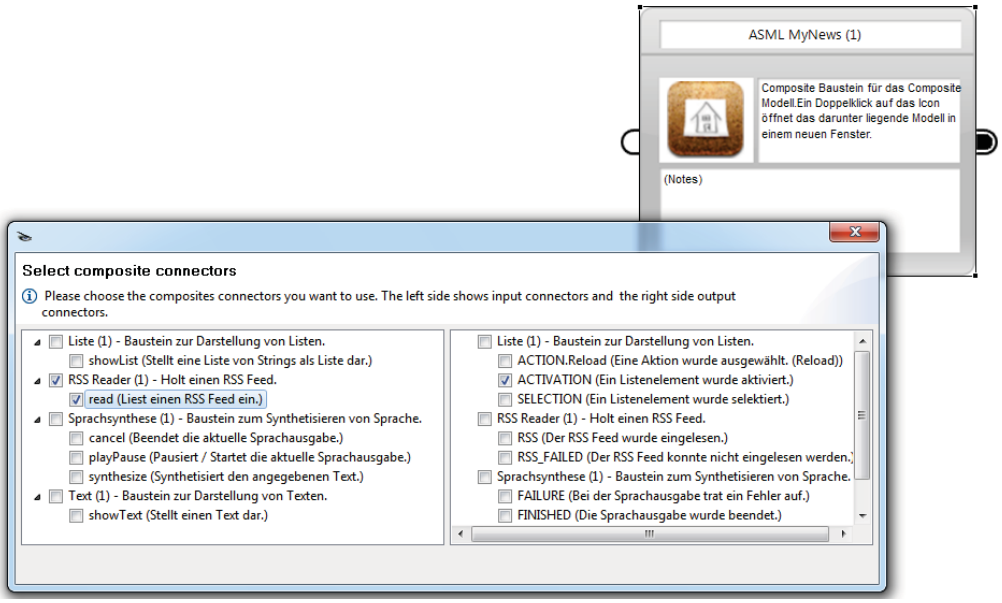


Abbildung 6-21 Auswählen von Composite Konnektoren
Quelle: Eigene Darstellung

Die zweite Ausnahme ist das Ändern des darunter liegenden Modells. Dies wird aus Sicht der Workbench als ganz normales Modell betrachtet und enthält keine Informationen darüber, dass es sich um ein Composite handelt. Daher ist es für den Nutzer möglich, dieses Modell über den Model Explorer auf der linken Seite der Workbench zu öffnen. Um eine einfache Verknüpfung des Composites mit seinem Modell zu ermöglichen, kann dieses zusätzlich durch einen Doppelklick auf das Symbol des Composites in einem neuen Editor geöffnet werden. Sobald das darunter liegende Modell geändert wird, wird das Composite über diese Änderungen informiert und passt sich entsprechend den neuen Gegebenheiten an. Wird zum Beispiel ein Baustein im Modell entfernt, dessen Konnektor im Composite verwendet wird, wird dieser automatisch aus dem Composite entfernt und ein, eventuell mit diesem Konnektor verknüpfter Stream, gelöscht.

6.2.2.3 Architektur

Zur Implementierung der vorgestellten Umsetzung des Editors wurde eine, an den in den Grundlagen dargestellten Ansatz angelehnte, Architektur gewählt. Abbildung 6-22 gibt einen Überblick über die einzelnen Bestandteile des Editors. Das zentrale Paket ist das Editor Paket. Seine Aufgabe ist es, den Editor zu initialisieren und die einzelnen Bestandteile zusammenzubringen. Darüber hinaus werden hier auch die Bestandteile, die eine Verknüpfung des Editors mit der Workbench ermöglichen realisiert. Die Umsetzung der Darstellung der Modelle erfolgt im View Paket. Das Modell wird, wie bereits erwähnt, in einem eigenen Plugin realisiert und ist somit nicht Teil des Editors, auch wenn ein Großteil der hier realisierten Komponenten auf Elemente des Modells angewiesen ist.

Die Interaktion der Nutzer mit dem Modell wird in den beiden Paketen Command und Policies realisiert. Policies legen dabei fest, welcher Command bei welcher Aktion des Nutzers ausgeführt werden soll. Zur Unterstützung dieser Aufgaben, stehen noch die Funktionalitäten der beiden Pakete Directedit und Dialog zur Verfügung. Im Ersten wird das

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Editieren von Properties und Notizen und im Zweiten die Popupfenster für Composites und Streams realisiert.

Die Aktualisierung der Anzeige, ausgehend von Änderungen am Modell, wird im Controller Paket realisiert. Dieses Paket hat demzufolge auch die engste Verzahnung mit der Implementierung des Modells. Darüber hinaus ist dieses Paket zusammen mit dem Editor Paket für das initiale Erstellen der Anzeige des Modells verantwortlich.

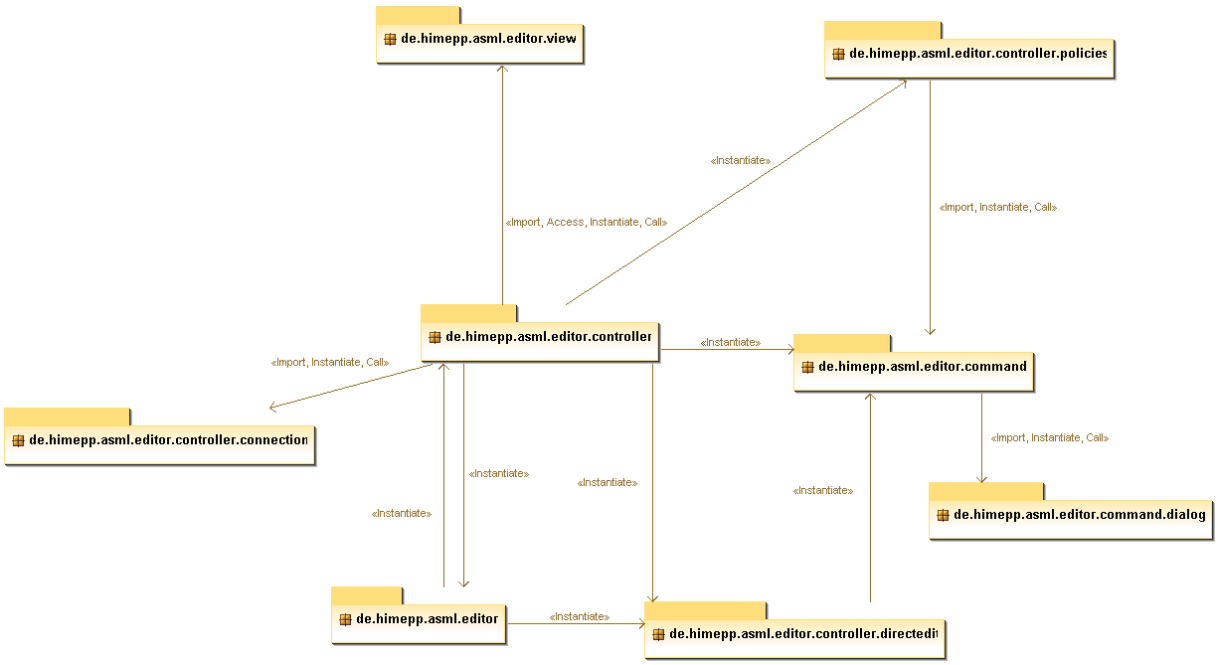


Abbildung 6-22 Übersicht der Bestandteile des Editors
Quelle: Eigene Darstellung

Das letzte Paket, Connection, fasst Hilfsfunktionen zusammen, die für die Darstellung und das Routing von Streams verantwortlich sind. Im Anschluss werden die Bestandteile der einzelnen Pakete zusammen mit deren Funktionalität genauer erläutert.

6.2.2.3.1 Editor

Die Bestandteile des Editor Pakets sind die zentralen Elemente des Editors. Der Klasse Editor kommt dabei eine besondere Bedeutung zu. Sie ist für das Laden und Speichern des Modells, für die Verwaltung des Command Stacks sowie für das Layout und die unterstützenden Funktionen des Canvas zuständig. Um diese Aufgaben zu erfüllen, stehen eine Reihe von unterstützenden Klassen in diesem Paket zur Verfügung.

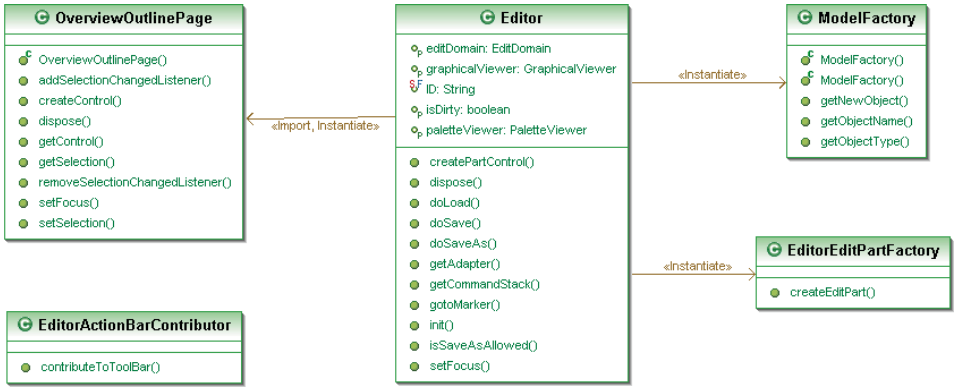


Abbildung 6-23 Übersicht der Bestandteile des Pakets Editor
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Die Klasse OverlayOutlinePage realisiert die Übersicht über gezoomte Modelle in der linken unteren Ecke der Workbench. Da diese Funktion stark mit dem Editor verknüpft ist, wird sie daher auch im Editor und nicht in der Workbench realisiert. Daneben hilft die Klasse EditorActionBarContributor bei der Integration in die Workbench. In dieser Klasse registriert sich der Editor für Funktionen der Workbench, wie z.B. das Speichern des Modells oder das Löschen von Elementen.

Für das Erstellen von Modellen und deren Bestandteile sind die beiden Klassen ModelFactory und EditorEditPartFactory verantwortlich. Der ModelFactory kommt dabei die Aufgabe zu, neue Elemente im Modell zu instanziieren, wenn der Nutzer diese durch Drag'n'Drop aus der Toolbar anlegt. Die EditorEditPartFactory dagegen erstellt die dazu passenden EditParts und sorgt somit für eine Darstellung des neuen Elements auf dem Bildschirm.

6.2.2.3.2 View

Die Darstellung der einzelnen Modellelemente wird vom Paket View übernommen. Ausgehend vom DiagramView, welcher die oberste Ebene darstellt, stehen Klassen für die Darstellung von Bricks (BrickView), Composites (CompositeView) und Start, Stop und Autostart (StartStopView) zur Verfügung. Für die Darstellung von Stubs wird ebenfalls die Klasse BrickView verwendet. Wie in Abbildung 6-24 zusehen, stellen die Klassen CompositeView und StartStopView eine Erweiterung der Klasse BrickView dar, da sie dessen Funktionen lediglich anpassen, jedoch zum Großteil wiederverwenden.

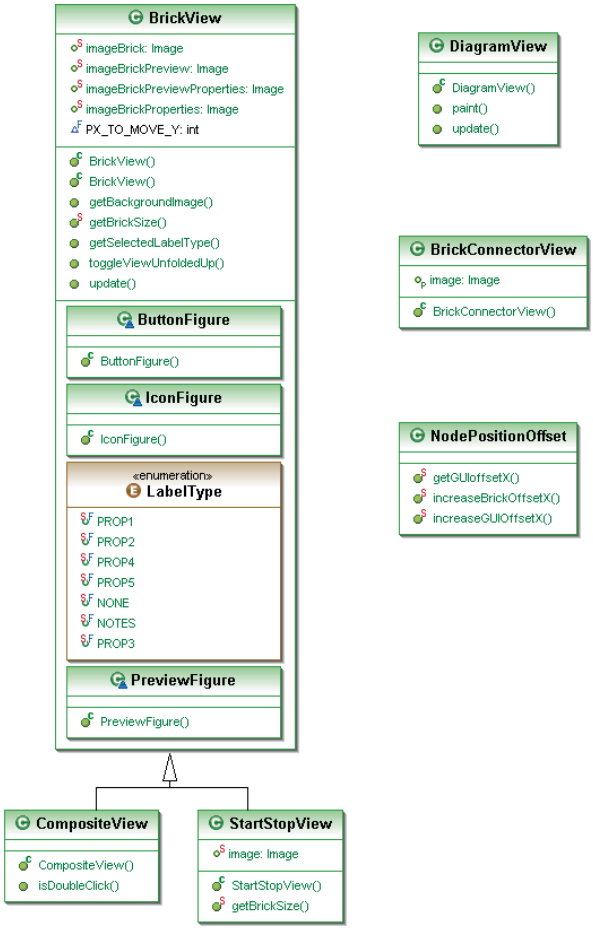


Abbildung 6-24 Übersicht der Bestandteile des Pakets View
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Für die Darstellung von Konnektoren wird eine eigene Klasse verwendet, da die diese im Modell logisch von ihrem Brick getrennt sind und somit auch durch ein eigenes Element repräsentiert (BrickConnectorView) werden. Für die Berechnung der relativen Position eines Konnektors zu dessen Birck wird noch die Hilfsklasse NodePositionOffset verwendet.

6.2.2.3.3 Connection

Das Hilfspaket Connection beinhaltet nur eine einzelne Klasse. Der ShortestPathConnection-Router hilft bei der Darstellung von Streams im Modell. Da die Darstellung von Streams eine Standartfunktion von GEF ist, wird sie nicht durch eine eigene Klasse realisiert sondern direkt aus dem Framework übernommen. Das Routen der auf dem Bildschirm anzuzeigenden Linie wird allerdings im Editor individuell über dieses Paket realisiert.

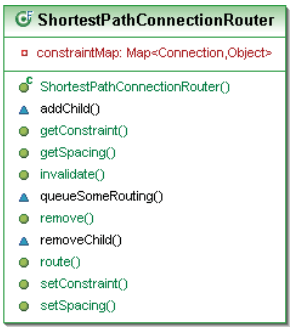


Abbildung 6-25 Übersicht der Bestandteile des Pakets Connection
Quelle: Eigene Darstellung

6.2.2.3.4 Command

Für die Interaktion der Nutzer mit dem Modell steht eine ganze Reihe an Commands zur Verfügung. Diese lassen sich grob in drei Kategorien einteilen: Commands für Bricks, Commands für Streams und Commands für Composites. Dabei ist zu beachten, das Stubs, Composites und Start, Stop und Autostart ebenfalls als Bricks behandelt werden. Alle Commands implementieren dabei die von GEF zur Verfügung gestellte Klasse Command, so dass sie jeweils über das gleiche Interface verfügen. Alle Commands haben also eine execute-, undo- und redo-Methode, die vom Command Stack genutzt wird, um die Undo- und Redo-Funktion zu realisieren.

Für das Bearbeiten von Bricks gibt es insgesamt fünf Commands. Mit dem Create-BrickCommand werden neue Bricks erstellt und mit dem DeleteBrickCommand wieder gelöscht. Das Verschieben auf dem Bildschirm steuert der MoveBrickCommand. Um die Parameter eines Bricks zu Verändern, werden die beiden Commands ChangeBrickNotes-Command und ChangeBrickPropertyCommand verwendet.

Die Interaktion mit Streams erfordert hingegen sieben Commands. Mit dem CreateStream-Command werden neue Streams nach einem Drag'n'Drop eines Ausgangskonnektors auf einen Eingangskonnektor erzeugt und mit dem DeleteStreamCommand können diese wieder entfernt werden. Das Öffnen des Popupfensters zum Editieren von Streams wird durch den StreamSetParametersCommand realisiert. Die verbleibenden vier Commands sind für die Manipulation von Bendpoints zuständig: BendPointCommand, CreateBendPointCommand, DeleteBendPointCommand und MoveBendPointCommand.

Zuletzt gibt es noch zwei Commands für das Anlegen von neuen Composites (ExtractCompositeCommand) und das Öffnen des Popupfensters zum Auswählen der zu

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

verwendenden Eingangs- und Ausgangskonnektoren (CompositeSetConnectorsCommand). Desweiteren werden für die Interaktion mit Composites die Commands der Bricks genutzt.

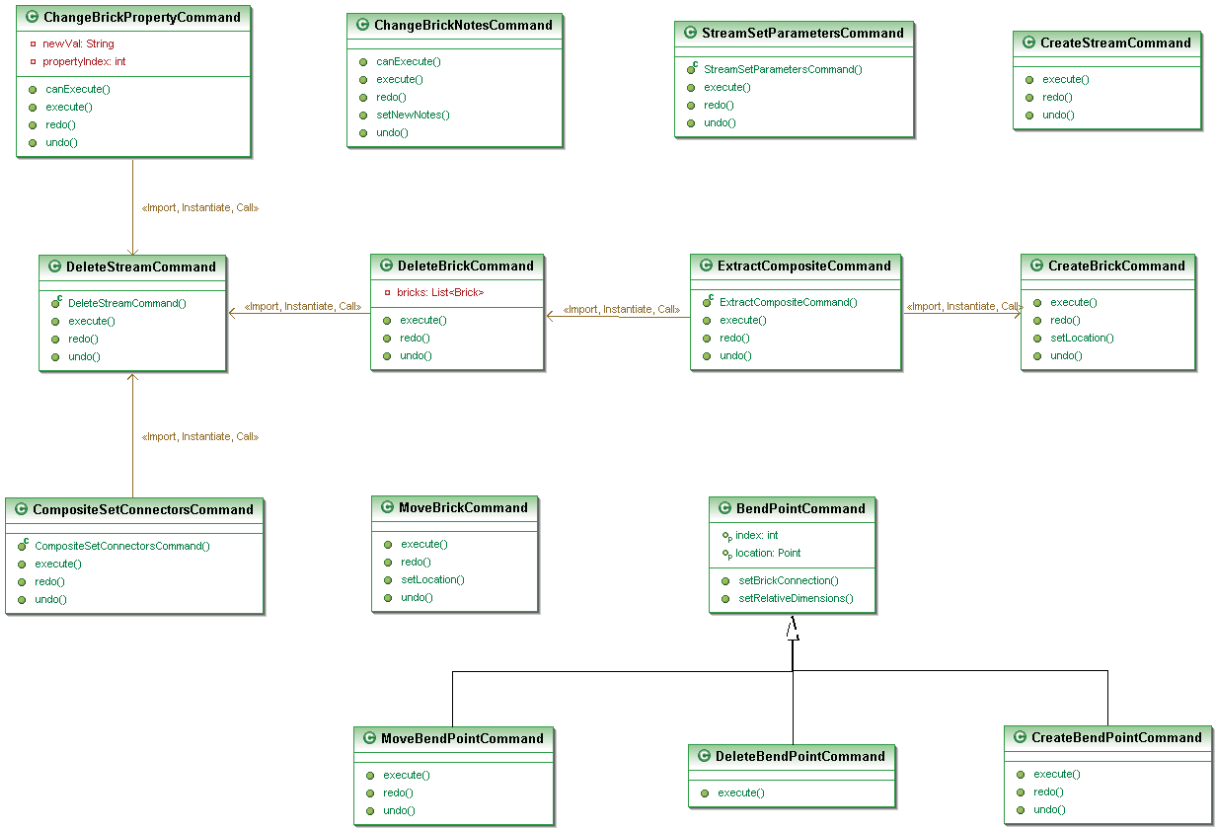


Abbildung 6-26 Übersicht der Bestandteile des Pakets Command
Quelle: Eigene Darstellung

6.2.2.3.5 Policies

Die Aufgabe der Policies ist es festzulegen, welcher Command bei welcher Aktion des Nutzers mit einem Element der View ausgelöst werden sollen. Policies werden dafür in GEF für alle Aktionen die auf der Benutzeroberfläche möglich sind registriert und instanziieren und initialisieren den jeweils notwendigen Command.

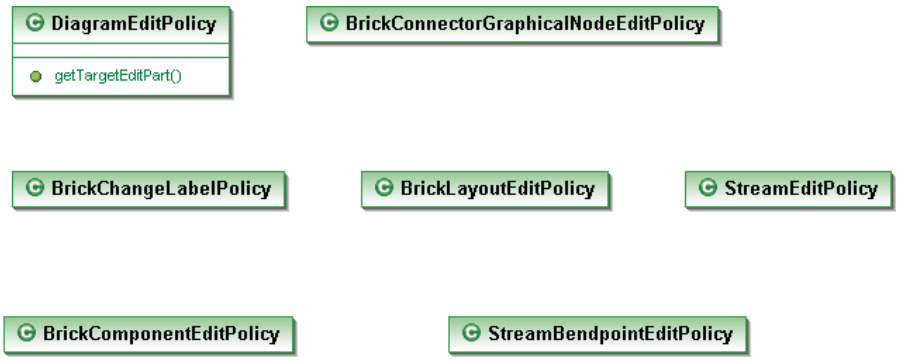


Abbildung 6-27 Übersicht der Bestandteile des Pakets Policies
Quelle: Eigene Darstellung

Wie in Abbildung 6-27 zu sehen, wurden insgesamt sieben Policies für den Editor realisiert: BrickChangeLabelPolicy, BrickComponentEditPolicy, BrickConnectorGraphicalNodeEditPolicy, BrickLayoutEditPolicy, DiagramEditPolicy, StreamBendpointEditPolicy und StreamEditPolicy.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

6.2.2.3.6 DirectEdit

Das Paket DirectEdit ist für das Editieren von Properties und Notizen direkt im Modell verantwortlich. Darüber hinaus befindet sich hier auch die Funktionalität zum Erstellen neuer Composites direkt aus dem Modell heraus.

Der EditorDirectEditManager wird vom entsprechenden Command verwendet, ein Fenster zum Editieren von Inhalten direkt im Modell anzuzeigen. Für die Lokalisierung der dafür passenden Position im Modell wird die Klasse LabelCellEditorLocator verwendet.

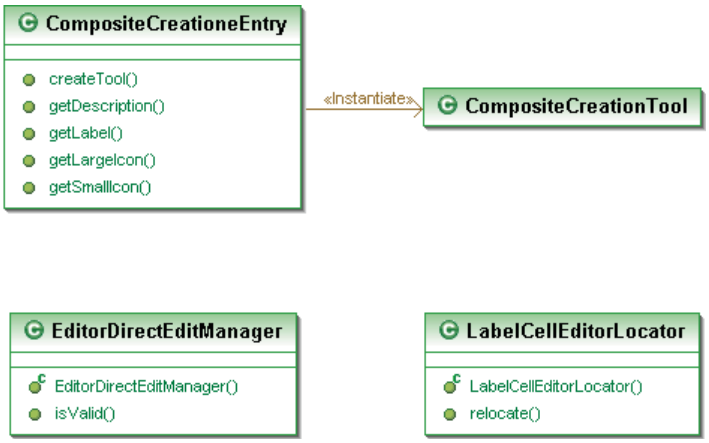


Abbildung 6-28 Übersicht der Bestandteile des Pakets DirectEdit
Quelle: Eigene Darstellung

Die anderen beiden Elemente dieses Pakets implementieren den „Create Composite“-Eintrag in der Toolbar. Der CompositeCreationEntry kümmert sich um die Anzeige des entsprechenden Eintrags und das CompositeCreationTool um die Umsetzung der eigentlichen Funktionalität.

6.2.2.3.7 Dialog

Wie in Abschnitt 6.2.2.2 besprochen, stehen dem Nutzer zwei Dialoge zur Auswahl der in Composites genutzten Konnektoren sowie zum Konfigurieren von Streams zur Verfügung. Beide Dialoge werden, wie in Abbildung 6-29 zu sehen, jeweils durch eine Dialogklasse und einen dazu passenden Content Provider realisiert. Die Dialogklassen sind für die Darstellung der Dialoge auf dem Bildschirm sowie deren Interaktion mit den Nutzern verantwortlich. Der anzuzeigende Inhalt wird von den beiden Klassen CompositeSetConnectorsDialog-ContentProvider und StreamSetParametersDialogContentProvider zur Verfügung gestellt. Die Dialoge werden von den entsprechenden Commands aufgerufen, welche schließlich auch die gesammelten Informationen an das Modell weitergeben.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

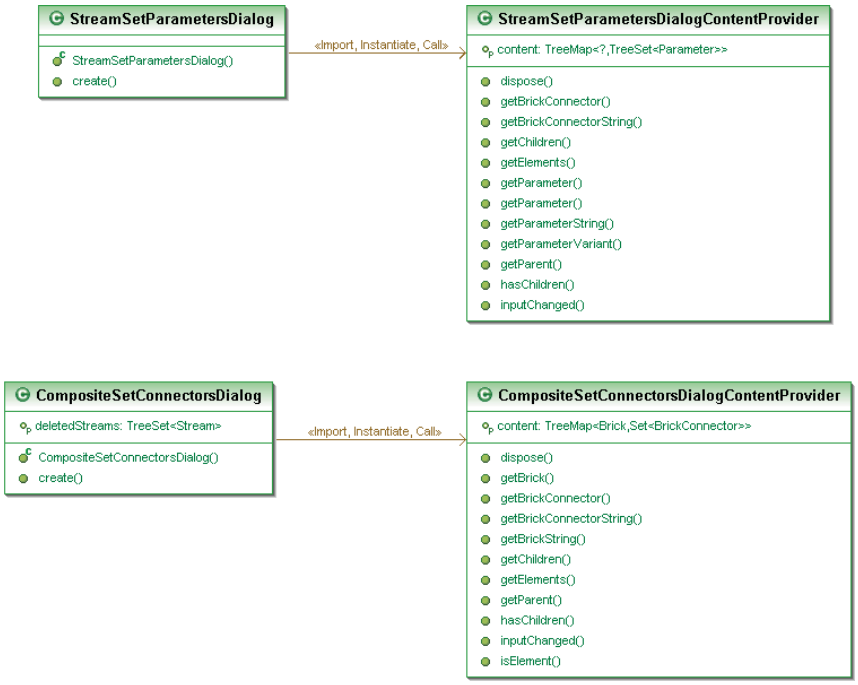


Abbildung 6-29 Übersicht der Bestandteile des Pakets Dialog
Quelle: Eigene Darstellung

6.2.2.3.8 Controller

Das letzte Bindeglied in der in Abschnitt 6.2.2.1 vorgestellten Realisierung des Editors ist die Verbindung zwischen Modell und View. Für jedes Element des Modells steht in diesem Paket eine Klasse zur Verfügung, die diese Verbindung umsetzt. Es finden sich hier also ähnliche Elemente wie im View Paket oder in der später vorgestellten Umsetzung des Modells. Die Hier realisierten EditParts haben zum einen die Aufgabe, die entsprechende View zu initialisieren und zum anderen diese über Änderungen am Modell zu Informieren.

Wie in Abbildung 6-30 zu sehen, stehen die Klassen BrickConnectorEditPart, BrickEditPart, CompositeConnectorEditPart, CompositeEditPart, DiagramEditPart, StartStopEditPart, StreamEditPart zur Verfügung. StartStopEditPart und CompositeEditPart sind wiederum vom BrickEditPart abgeleitet, da sie nur Spezialfälle dieses darstellen. Gleiches gilt auch für den CompositeConnectorEditPart, der die Funktionen des BrickConnectorEditPart erbt.

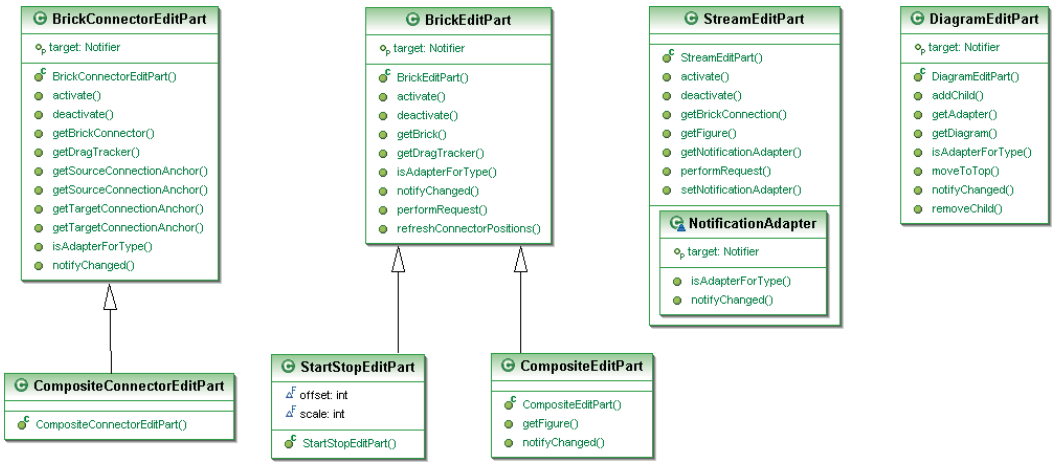


Abbildung 6-30 Übersicht der Bestandteile des Pakets Controller
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

6.2.3 Produktmodell: Persistieren der Modelle

Nachdem in den beiden vorangegangenen Abschnitten mit der ASML Workbench und dem ASML Editor die beiden Bestandteile des Modellierungswerkzeugs für Automotive Service Modelle entwickelt wurden, fehlt für deren Einsatz noch ein geeigneter Ansatz, dass durch diese Werkzeuge entstehende Produktmodell zu speichern. Darüber hinaus müssen dem Editor auch die, in Form von Bricks und Stubs, zur Modellierung zur Verfügung stehenden domänenspezifischen Konzepte verfügbar gemacht werden. Insbesondre Stubs werden nicht vom domänenspezifischen Framework realisiert. Aus diesem Grund muss zusätzlich eine geeignete Form des Persistierens dieser Konzepte, unabhängig vom domänenspezifischen Framework, gefunden werden. Der nachfolgende Abschnitt befasst sich daher sowohl mit dem Persistieren der Produktmodelle, also der Automotive Service Modelle, als auch dem Persistieren der domänenspezifischen Konzepte.

6.2.3.1 Grundlagen

Aufgrund der Tatsache, dass es sich bei beiden zu persistierenden Elementen um wohl strukturierte Objekte handelt, die vollständig durch textuelle Attribute beschrieben werden können, eignet sich die Extensible Markup Language (XML) für diese Aufgabe. Der Einsatz von XML hat den Vorteil, dass die entstehenden Dateien vollständig unabhängig von der gewählten Implementierung der einzelnen Werkzeuge sind und darüber hinaus durch einen einfachen Texteditor bearbeitet und gelesen werden können. Dadurch können sowohl die definierten Konzepte als auch die entstandenen Modelle in unterschiedlichen Werkzeugen auf unterschiedlichen Plattformen genutzt werden, ohne dass dabei spezielle proprietäre Komponenten zur Interpretation der Informationen zwingend notwendig sind. Ein weiterer Vorteil ist der grundsätzliche, logische Aufbau in Elemente und Attribute. Diese hierarchische Struktur entspricht in ihren Grundzügen einer objektorientierten Entwicklung und eignet sich daher besonders, die in der Implementierung als Objektmodell vorliegenden Informationen zu speichern.

Grundsätzlich unterscheidet man zwischen drei Arten von XML Dokumenten: Dokumentzentrierte Dokumente werden genutzt um vom Menschen lesbaren Dokumente zu erstellen, in denen XML zum semantischen Anreichern der Informationen genutzt wird. Die zweite Art sind datenzentrierte Dokumente. Diese sind sehr stark strukturiert und eignen sich mehr für die Interpretation durch maschinelle Verarbeitung. Der dritte Ansatz, die semi-strukturierten Dokumente stellen eine Mischform aus den beiden anderen dar. Aufgrund der Tatsache, dass sowohl die Konzepte als auch die Modelle durch die zur Verfügung gestellten Werkzeuge erstellt und interpretiert werden, wird in diesem Fall eine datenzentriert Orientierung gewählt.

Diesem Ansatz kommt auch die Verfügbarkeit von standardisierten Softwareframeworks für deren Verarbeitung entgegen. Für den konkreten Einsatz im Rahmen der ASML Umgebung, unter Berücksichtigung der Tatsache, dass für die Gestaltung des Editors des GEF Framework verwendet wird, kommt hier das zu diesem passende Eclipse Modeling Framework (EMF) zum Einsatz (Budinsky et al. 2003). Das Ziel dieses Framework ist es, eine enge Verknüpfung zwischen den im Arbeitsspeicher verfügbaren Modellen und deren persistierter Form im Dokument herzustellen. Zu diesem Zweck stellt EMF ein eigenes Metamodell namens Ecore zur Verfügung. Dieses, in Abbildung 6-31 dargestellte, Metamodell implementiert den Meta Object Facility Standard (MOF) der Object Management Group (OMG).

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

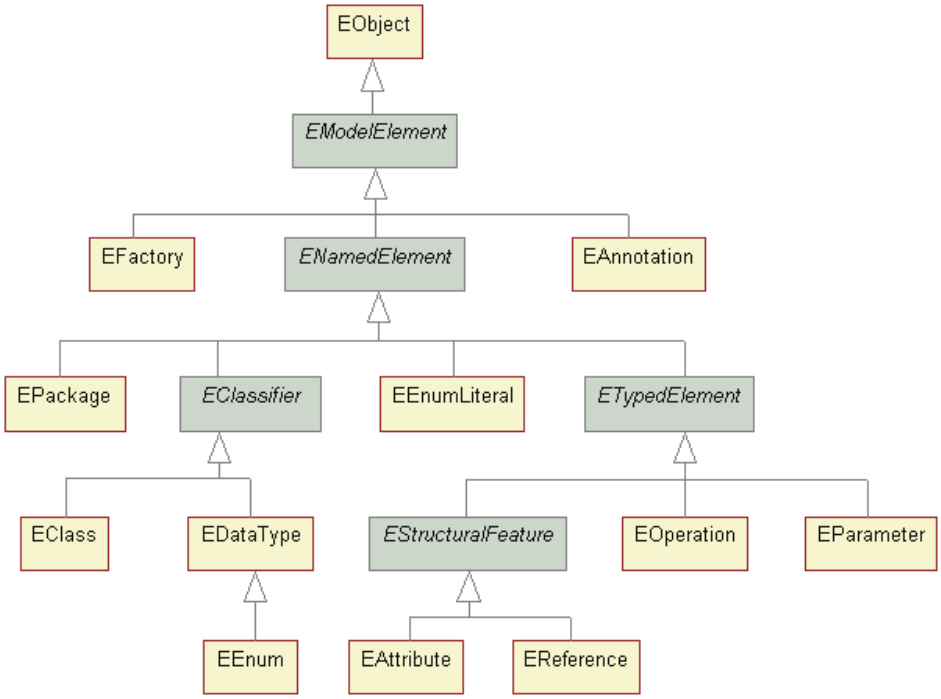


Abbildung 6-31 Hierarchie des EMF Ecore-Modells
Quelle: (Eclipse Foundation 2005)

Für die Umsetzung der Werkzeugunterstützung der Automotive Services Modeling Language wird ein, auf dem Ecore-Modell aufbauendes, Modell als Zwischenmodell für die Persistierung der Automotive Service Modelle genutzt. Diese werden also aus ihrer internen Repräsentation im Arbeitsspeicher in ein Ecore-Modell transformiert, welches dann, mit Hilfe der in EMF verfügbaren Mechanismen, in das datenzentrierte XML Dokument gespeichert wird. Dieses Dokument wird mit der Endung *„asm“* gekennzeichnet. Für das Persistieren der domänenspezifischen Konzepte kommt analog ein datenzentriertes XML Dokument zum Einsatz, die Dateien werden mit dem Namen *„asml-description.xml“* versehen. Diese Trennung der Konzepte und Modelle hat den Vorteil, dass die jeweils entstehenden Dateien sehr klein sind und daher leicht über eine schmalbandige Mobilfunkverbindung ins Fahrzeug übertragen werden können.

6.2.3.2 Persistierung domänenspezifischer Konzepte

Wie bereits erwähnt werden Bricks und Stubs durch datenzentrierte XML Dateien gespeichert. Die Struktur dieser Dateien ist in einer einfachen Baumstruktur gehalten, die die beschreibenden Elemente eines Konzepts umfasst. Analog den vier Seiten des Brick Wizard kann die Struktur der „asml-description.xml“ genannten Dateien in vier Bereiche gegliedert werden: Allgemeine Informationen, Properties, Eingangskonnektoren, und Ausgangskonnektoren. Die nachfolgenden Abbildungen zeigen zusammengesetzt eine verkürzte Version der asml-description.xml des Liste Brick, welches in Abschnitt 5.1.2.3.1.1 vorgestellt wurde.

```
<?xml version="1.0" encoding="utf-8"?>
<brick name="Liste"
       description="Baustein zur Darstellung von Listen."
       gui="true"
       singleton="false">
```

Abbildung 6-32 asml-description.xml Listen Brick Teil I: Allgemeine Informationen
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Abbildung 6-32 zeigt den ersten Teil der asml-descripton.xml. In der ersten Zeile werden die Version des XML Standards sowie die Kodierung der Textdatei in einem XML Metatag angegeben. Als Kodierung wird UTF-8 eingesetzt, um eine Übertragbarkeit der Inhalte auf unterschiedliche Plattformen zu ermöglichen. Das erste Element des Dokuments trägt den Namen Brick und hat vier Attribute. Diese Attribute, die im Hauptbereich des Brick Wizard abgefragt werden, sind Name (name), Beschreibung (description), Bedienkonzept (gui) und Singelton (singleton). Die letzteren beiden Werte werden mit den booleschen Werten *true* und *false* angegeben.

```
<properties>
    <property id="TITLE"
              type="String"
              name="Titel"
              description="Titel der Liste."
    />
</properties>
```

Abbildung 6-33 asml-description.xml Listen Brick Teil II: Properties
Quelle: Eigene Darstellung

Der zweite Bereich befindet sich innerhalb des Brick Elements, welches das gesamte Dokument umfasst. Innerhalb eines Containers mit dem Namen *„properties“* können einzelne Properties, wie in Abbildung 6-33 zu sehen, angegeben werden. Die Informationen, die das Property beschreiben, werden wiederum in Form von Attributen spezifiziert. Diese umfassen die ID (id), den Datentyp (type), den Namen (name) sowie die Beschreibung (description) des Property.

```
<connections>
    <input>
        <connection function="showList"
                    name="Liste anzeigen"
                    description="Stellt eine Liste von Strings als Liste dar."
                    >
            <parameters>
                <variant id="A"/>
                <variant id="B">
                    <parameter type="List(String)"
                               name="items"
                               description="Liste der anzuzeigenden Strings."
                    />
                </variant>
            </parameters>
        </connection>
    </input>
```

Abbildung 6-34 asml-description.xml Listen Brick Teil III: Eingangskonnektoren
Quelle: Eigene Darstellung

Der dritte Bereich umfasst die Beschreibung der Eingangskonnektoren. Zunächst wird ein Container geöffnet, der sowohl Eingangs- als auch Ausgangskonnektoren umfasst und in diesem ein spezieller Container für die Eingangskonnektoren erstellt (input). Analog zu den Properties können in diesem Container nun beliebig viele einzelne Konnektoren (connection) beschrieben werden. Die für einen gesamten Konnektor geltenden Informationen wie Funktion (function), Name (name) und Beschreibung (description) werden dabei als Attribute angegeben. Parameter eines Eingangskonnektors werden wiederum in einem Container (parameters) zusammengefasst in dem mehrere Varianten (variant) enthalten sein können. Jede dieser Variante wird dabei mit A, B, C, … im ID-Attribut (id) durchnummeriert. Eine

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Variante kann optional einen oder mehrere Parameter enthalten. Ein Parameter ist wiederum ein XML Element welches durch die drei Attribute Datentyp (type), Name (name) und Beschreibung (description) definiert wird.

```
        <output>
            <connection id="SELECTION"
                        name="Element selektiert"
                        description="Ein Listenelement wurde selektiert."
                        >
                <parameters>
                    <parameter id="SELECTION_CONTENT"
                               type="String"
                               name="Element Text"
                               description="Der Text des ausgewählten Elements."
                    />
                </parameters>
            </connection>
        </output>
    </connections>
</brick>
```

Abbildung 6-35 asml-description.xml Listen Brick Teil IV: Ausgangskonnektoren
Quelle: Eigene Darstellung

Im letzten Bereich werden die Ausgangskonnektoren beschrieben. Dieser Bereich befindet sich immer noch innerhalb des Containers für Konnektoren, der in der letzten Abbildung geöffnet wurde. Ein Container für Ausgangskonnektoren (output) umschließt diesen Bereich. Analog zum letzten Container werden hier wiederum einzelne Konnektoren beschrieben, die, da es sich um Ausgangskonnektoren handelt, mit den Attributen ID (id), Name (name) und Beschreibung (description) definiert werden. Wiederum kann ein Konnektor mehrere Parameter habe, die in einem diese umschließenden Container (parameters) definiert werden. Da Ausgangskonnektoren keine unterschiedlichen Varianten an Parametern haben, werden diese direkt in diesem Container beschrieben. Parameter der Ausgangskonnektoren haben die Attribute ID (id), Datentyp (type), Name (name) und Beschreibung (description). Schließlich werden noch alle geöffneten XML Elemente wieder geschlossen.

6.2.3.3 Persistierung von Automotive Service Modellen

Um das Datenformat für das Persistieren von Automotive Service Modellen zu beschreiben, wird im Anschluss ein einfaches Modell als Beispiel gewählt. Abbildung 6-36 zeigt ein aus vier Elementen bestehendes Modell, welches ausgehend vom einen Start Element ein Text Brick anzeigt. In diesem Brick wird der Nutzer über den Zweck dieses Services informiert und kann mit Hilfe der Aktion „Nachrichten“ den ASML MyNews Composite aufrufen. Das hinter diesem Composite stehende Modell wurde im Rahmen der Case Study in Abschnitt 5.2.1 bereits erläutert. Als Eingangskonnektor wurde “Lies einen RSS Feed ein“ des RSS Reader Brick gewählt. Sobald der Nutzer im Composite eine Nachricht auswählt wird nicht die weitere Funktion des Composites ausgeführt, sondern der Name der Nachricht mit einer Textsynthese ausgegeben. Dieses einfache Modell zeigt alle wesentlichen Elemente, die für das Speichern von Automotive Service Modellen von Interesse sind.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

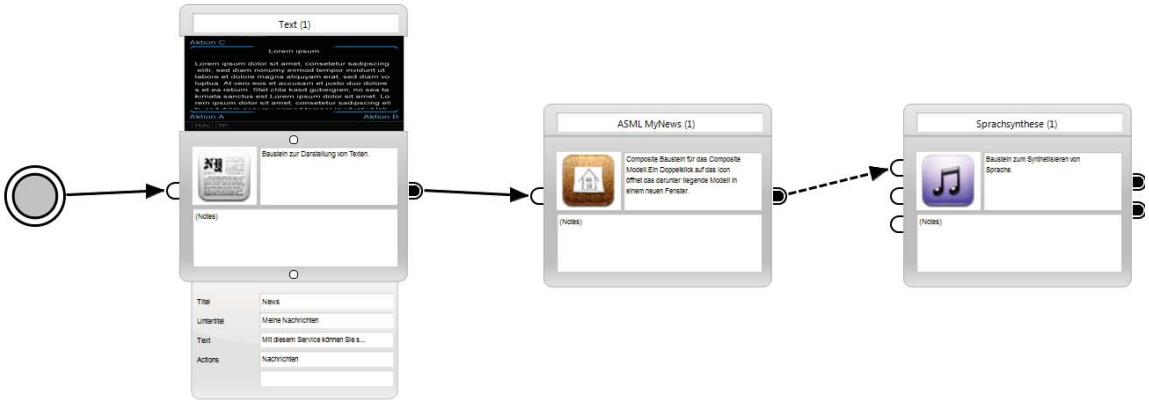


Abbildung 6-36 Persistieren von Modellen: Screenshot Beispielmodell
Quelle: Eigene Darstellung

Analog des XML Formats für Konzepte, wird auch hier in der ersten Zeile die XML Version sowie die zu nutzende Kodierung des Dokuments spezifiziert. Hier kommt ebenfalls utf-8 zum Einsatz. Als nächstes wird, wie in Abbildung 6-36 zu sehen, der Typ des Dokuments festgelegt. EMF nutzt dafür das XML Metadata Interchange Format (XMI), welches anbieterneutral ist und den Datenaustausch von Objekten auf Basis des Meta Object Facility Standard erlaubt. Das xmi:XMI-Element legt somit fest, dass es sich um die XMI Version 2.0 handelt und sowohl der „http://www.omg.org/XMI“ als auch der „http://www.automotive-services.org/ASML“ Namensraum genutzt wird. Letzter legt die im Nachfolgenden verwendeten „ASML:*“ Elemente fest. In der letzten Zeile der Abbildung 6-36 ist noch das ASML:Diagram Element zu sehen. Dieses hat die Aufgabe eine eindeutige ID für jedes Modell zu speichern, welche im Attribut ID (id) hinterlegt ist und mit deren Hilfe Modelle, auch wenn sie umbenannt worden sind, identifiziert werden können.

```
<?xml version="1.0" encoding="utf-8"?>
<xmi:XMI xmi:version="2.0"
         xmlns:xmi="http://www.omg.org/XMI"
         xmlns:ASML="http://www.automotive-services.org/ASML">
    <ASML:Diagram id="7cd23d8a-cfbe-487d-b97f-b363be0f30df"/>
```

Abbildung 6-37 Persistieren von Modellen: XML Struktur Teil I
Quelle: Eigene Darstellung

Im zweiten Teil der XML Struktur wird in Abbildung 6-38 das erste Brick gespeichert. Bei diesem Baustein handelt es sich um das Start Brick, welches weder über Properties noch spezielle Konnektoren verfügt. Dementsprechend besteht das ASML:Brick Element aus sechs Attributen. Die ersten vier speichern die ID (id), den Typ (type), den Namen (name) sowie die vom Nutzer angegebenen Notizen (notes). Hier fällt auf, dass obwohl ein Startelement keine Notizen und auch keinen angezeigten Titel besitzt, diese Informationen hier trotzdem gespeichert werden. Dies hat den Grund, dass es sich bei diesem Element logisch betrachtet ebenfalls um einen Brick handelt und daher auch alle Standardattribute von Bricks vorhanden sind.

Besondere Bedeutung kommen den beiden Attributen Typ und Name zu. Der Typ legt fest, um was für eine Art von Brick es sich handelt (StartStop, Brick, Composite) und der Name gibt das konkrete Konzept für diesen Brick an. Ersteres Attribut wird für die Anzeige und das zu verwendende Modellelement genutzt und zweiteres legt fest, welches Konzept und damit welche asml-description.xml durch dieses Brick repräsentiert wird. Wie bereits erwähnt, enthält das Modell keine Informationen über einzelne Konzepte, so dass diese stets aus der entsprechenden asml-description.xml übernommen werden müssen. Die letzten beiden

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Attribute, xPosition und yPosition, speichern schließlich noch die Position des Elements auf dem Canvas des Editors.

```
<ASML:Brick id="Start (1)"
            type="StartStop"
            name="Start"
            notes="(Notes)"
            xPosition="154"
            yPosition="614"/>
```

Abbildung 6-38 Persistieren von Modellen: XML Struktur Teil II
Quelle: Eigene Darstellung

Das nächste Element ist wiederum ein ASML:Brick und beschreibt den im Modell verwendeten Textbaustein. Analog zum letzten ASML:Brick Element sind wieder die Attribute ID (id), Typ (type), Notizen (notes) und Position (xPosition und yPosition) vorhanden. Der Typ ist in diesem Fall „Brick“ und der Name „Text“, was kennzeichnet, dass es sich um das Brick mit dem domänenspezifischen Konzept Text handelt. Neben diesen Attributen finden sich noch drei weitere in diesem Element. Da das Bedienkonzept sowohl über eine Vorschau als auch Properties verfügt, sind in der Darstellung auch die beiden auf- und zuklappbaren Bereiche vorhanden. Die Attribute unfoldedUp und unfoldedDown speichern den Zustand dieser Bereich, also ob sie offen oder geschlossen sind. Im Beispiel sind, wie in Abbildung 6-36 zu sehen, beide geöffnet.

Das letzte Attribut, Text2Property, verweist auf die zu diesem Brick gehöhrenden Properties. Diese werden als eigenständige Elemente gespeichert und durch das Attribut mit dem Brick verknüpft. Hier sieht man, das die Struktur des Dokuments grundsätzlich flach organisiert ist, also sich alle Elemente auf der gleiche Ebene befinden. Die in Abbildung 6-39 zu sehenden Werte /3, /4, /5 und /6 geben an, das das 3. bis 6. Element des Dokuments zu diesem Brick gehöhrende Properties beschreibt. Zu beachten ist dabei, das ein Model stets mit dem ASML:Diagram Element an Position 0 beginnt.

Die vier nachfolgenden ASML:Property Elemente speichern somit die zum Text Brick gehöhrenden Properties. Dies erfolgt in den drei Attributen ID (id), Wert (value) und Property2Brick. Die ID bezieht sich auf die in der asml-description.xml festgelegte ID der Property (vgl. Abbildung 6-33) und im Attribut value werden die Angaben des Nutzers im Modell hinterlegt. Das letzte Attribut verweist wiederum auf den Brick zu dem dieses Element gehört.

```
<ASML:Brick id="Text (1)"
            type="Brick"
            name="Text"
            notes="(Notes)"
            xPosition="341"
            yPosition="419"
            unfoldedUp="true"
            unfoldedDown="true"
            Brick2Property="/3 /4 /5 /6"/>
<ASML:Property id="TITLE"
               value="News"
               Property2Brick="/2"/>
<ASML:Property id="SUBTITLE"
               value="Meine Nachrichten" Property2Brick="/2"/>
<ASML:Property id="TEXT"
               value="Mit diesem Service k&#xf6;nnen Sie sich ihre Pers&#xf6;hnlichen Nachrichten anzeigen lassen."
               Property2Brick="/2"/>
<ASML:Property id="ACTIONS"
               value="Nachrichten"
               Property2Brick="/2"/>
```

Abbildung 6-39 Persistieren von Modellen: XML Struktur Teil III
Quelle: Eigene Darstellung

Da nun die ersten beiden Elemente des Modells vorhanden sind, kann der zwischen diesen Elementen befindliche Stream gespeichert werden. Dies geschieht mit Hilfe eines

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

ASML:Stream Elements. Das Element umfasst fünf Attribute: ID (id), Ausgangsbrick (sourceBrick), Ausgangskonnektor des Ausgangsbricks (sourceBrickConnector) sowie das Zielbrick (targetBrick) und den daran befindlichen Eingangskonnektor (targetBrick-Connector).

```
<ASML:Stream id="78efbfa6-5274-4828-b8f6-cee3f8089883"
             sourceBrick="Start (1)"
             sourceBrickConnector="Start"
             targetBrick="Text (1)"
             targetBrickConnector="showText"/>
```

Abbildung 6-40 Persistieren von Modellen: XML Struktur Teil IV
Quelle: Eigene Darstellung

Als nächstes wird das dritte Element des Modelles beschrieben. Zu den für Bricks typischen Attributen ID (id), Typ (type), Name (name), Notizen (notes) und Position (xPosition und yPosition) werden in diesem Fall noch weitere Attribute gespeichert. Wie am Typ Attribut ersichtlich, handelt es sich bei diesem Brick um ein Composite, so dass noch das dieses Composite beschreibende Model referenziert werden muss. Dies geschieht in den beiden Attributen Model ID (modelID) und Model Datei (modelFile). Beide geben analog das Modell an, wobei die Model ID Priorität über die Angabe des Dateinamens hat. Die Model ID bezieht sich dabei auf die im ASML:Diagram Element festgelegte ID.

```
<ASML:Brick id="ASML MyNews (1)"
            modelId="7cd23d8a-cfbe-487d-b97f-b363be0f30df"
            modelFile="ASML MyNews.asm"
            type="Composite"
            name="ASML MyNews"
            notes="(Notes)"
            xPosition="755"
            yPosition="532"
            Brick2CompositeConnector="/9 /10"/>
<ASML:CompositeConnector connector="RSS Reader (1)"
                         brick="read"
                         CompositeConnector2Brick="/8"/>
<ASML:CompositeConnector connector="Liste (1)"
                         brick="ACTIVATION"
                         CompositeConnector2Brick="/8"/>
```

Abbildung 6-41 Persistieren von Modellen: XML Struktur Teil V
Quelle: Eigene Darstellung

Zusätzlich werden noch die verwendeten Konnektoren angegeben. Dies ist für normale Bricks nicht notwendig, da diese Information in der asml-description.xml hinterlegt ist. Da der Nutzer für ein Composite die verwendeten Konnektoren allerdings individuell auswählen kann, müssen diese hier gesondert im Modell gespeichert werden. Analog den Properties im Brick werden die Konnektoren des Composite über eigene Elemente definiert (ASML:CompositeConnector) und über das Attribut Brick2CompositeConnector referenziert. Die beiden in Abbildung 6-41 gezeigten ASML:CompositeConnectoren werden über die Attribute Konnektor (connector) und Brick (brick) beschrieben. Diese beiden Werte geben, analog der Referenzierung in Streams, das Brick und den dazugehörigen Konnektor im, das Composite beschreibenden, Modell an. Das letzte Attribut, CompositeConnector2Brick, verweist wieder zurück auf das Composite Element.

```
<ASML:Stream id="fcea8ac6-78f2-4bce-bb1a-8d50f1ca1b8d"
             sourceBrick="Text (1)"
             sourceBrickConnector="ACTION.Nachrichten"
             targetBrick="ASML MyNews (1)"
             targetBrickConnector="RSS Reader (1).read"/>
```

Abbildung 6-42 Persistieren von Modellen: XML Struktur Teil VI
Quelle: Eigene Darstellung

Nachdem nun auch das dritte Element vorhanden ist, kann der zweite Stream zwischen dem Text Brick und dem Composite gespeichert werden. Dies erfolgt analog dem letzten Stream über die 5 Attribute ID (id), Ausgangsbrick (sourceBrick), Ausgangskonnektor des

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Ausgangsbricks (sourceBrickConnector) sowie das Zielbrick (targetBrick) und den daran befindlichen Eingangskonnektor (targetBrickConnector).

Schließlich erfolgt die Definition des letzten Bricks des in Abbildung 6-36 gezeigten Modells. Nachdem dieser Sprachsynthese Brick über keine Properties verfügt, wird er analog dem StartStop Brick durch die Attribute ID (id), Typ (type), Name (name), Notizen (notes) sowie die Position auf dem Canvas (xPosition und yPosition) beschrieben.

```
<ASML:Brick id="Sprachsynthese (1)"
            type="Brick" name="Sprachsynthese"
            notes="(Notes)"
            xPosition="1164"
            yPosition="532"/>
```

Abbildung 6-43 Persistieren von Modellen: XML Struktur Teil VII
Quelle: Eigene Darstellung

Zuletzt folgt noch der verbleibende dritte Stream, der das Composite mit dem Sprachsynthese Brick verbindet. Zunächst finden sich in Abbildung 6-44 die gleichen Attribute wie im letzten Stream in Abbildung 6-42: ID (id), Ausgangsbrick (sourceBrick), Ausgangskonnektor des Ausgangsbricks (sourceBrickConnector), Zielbrick (targetBrick) und den daran befindlichen Eingangskonnektor (targetBrickConnector). Da es sich bei diesem Stream allerdings um ein asynchrones Stream handelt, wird diese Information über ein weiteres Attribut isAsynchronous angegeben.

```
    <ASML:Stream id="acf4e06d-9f56-4f93-9360-268e42618c5b"
                 isAsynchronous="true"
                 sourceBrick="ASML MyNews (1)"
                 sourceBrickConnector="Liste (1).ACTIVATION"
                 targetBrick="Sprachsynthese (1)"
                 targetBrickConnector="synthesize"
                 Stream2Parameter="/14"/>
    <ASML:Parameter name="input"
                    sourceParameterVariant="A"
                    sourceBrick="ASML MyNews (1)"
                    sourceBrickConnector="ACTIVATION"
                    sourceParameter="SELECTION_CONTENT"
                    Parameter2Stream="/13"/>
</xmi:XMI>
```

Abbildung 6-44 Persistieren von Modellen: XML Struktur Teil VIII
Quelle: Eigene Darstellung

Darüber hinaus werden über diesen Stream auch Parameter vom Composite an die Sprachsynthese weitergegeben. Diese Zuordnung der Parameter wird über ein eigenes Element ASML:Parameter realisiert, welches im ASML:Stream über das Attribut Stream2Parameter referenziert ist. Das Attribut Name (name) verweist auf den Parameter des Eingangskonnektors sowie das Attribut sourceParameterVariant auf die vom Nutzer gewählte Variante des Konnektors. Der an diesen Parameter zu übergebende Inhalt wird durch Angabe seiner Quelle spezifiziert. Dies geschieht durch die Attribute Ausgangsbrick (sourceBrick), Ausgangskonnektor des Ausgangsbricks (sourceBrickConnector) sowie den dort an den Stream übergebenen Parameter (sourceParameter). Zuletzt wird noch der ASML:Stream, zu dem die Parameterübergabe gehört, angegeben (Parameter2Stream).

Zur vollständigen Beschreibung von Automotive Service Modellen sind also sowohl die das Modell beschreibende .asm-Datei sowie alle in dieser verwendeten asml-description.xml Dateien notwendig. Darüber hinaus werden Composites ebenfalls in Form von unabhängigen Modellen beschrieben, die bei Einsatz von Composites ebenfalls vorhanden sein müssen.

6.2.3.4 Architektur

Für eine Umsetzung der beschriebenen Persistierung der Automotive Service Modelle sowie der durch Bricks beschriebenen Konzepte besteht die Architektur des Modells aus 3 Paketen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Das Model Paket beinhaltet das eigentliche Modell, welches im Editor verwendet wird um durch Commands verändert und durch EditParts und die entsprechenden Views dargestellt wird. Alle Informationen, die einzelne Elemente des Modells aus den asml-description.xml Dateien beziehen, werden mit Hilfe des Properties Pakets geladen und verwaltet. Das Tools Paket ist schließlich für das Transformieren des Modells in das Ecore Modell zuständig, welches anschließend mit den Mechanismen des EMF Frameworks in die XML Struktur der asm-Dateien überführt wird. Darüber hinaus wird hier auch der Weg in die umgekehrte Richtung realisiert, also das Laden von asm-Dateien in Ecore Modelle durch EMF sowie das Transformieren derselben in die interne Repräsentation des Models.

Abbildung 6-45 Übersicht der Bestandteile des Produktmodells
Quelle: Eigene Darstellung

6.2.3.4.1 Model

Das Model Paket besteht aus je einer Klasse für die in ASML verfügbaren Elemente: Die Diagram Klasse bildet den zentralen Knoten eines Modells und verwaltet als ihre Kind-Elemente alle Bricks, was, wie Abbildung 6-46 zu entnehmen ist, auch Composites und StartStop Elemente umfasst. Dem entsprechend ist die Klasse Brick, welche Bricks und Stubs realisiert, die Oberklasse für Composites und StartStop Elemente. Connectoren werden, da sie in unterschiedlicher Anzahl vorhanden sein können, in einer eigenen Klasse realisiert. Besonders anzumerken ist, das Composites keine gewöhnlichen Konnektoren verwenden, sonder wiederum von diesen abgeleitete CompositeConnectors. Diese haben die Aufgabe, im darunter liegenden Composite Modell enthaltene, Konnektoren anderer Bricks zu referenzieren.

Schließlich gibt es noch die beiden Klasse Stream und StreamBendpoint. Streams werden, als relative Elemente die zwei Konnektoren miteinander verbinden, nicht als direktes Kind-Element des Diagramms verwaltet. Stattdessen werden sie von den jeweils entsprechenden Konnektoren direkt referenziert. Alle in Streams enthaltenen Informationen, wie die Zuordnung von Parametern zu Eingangskonnektoren, werden innerhalb der Stream Klasse gespeichert. Das letzte Element ist der StreamBendpoint. Diese Klasse beschreibt einen bestimmten Punkt auf dem Canvas durch den ein Stream geroutet werden soll.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

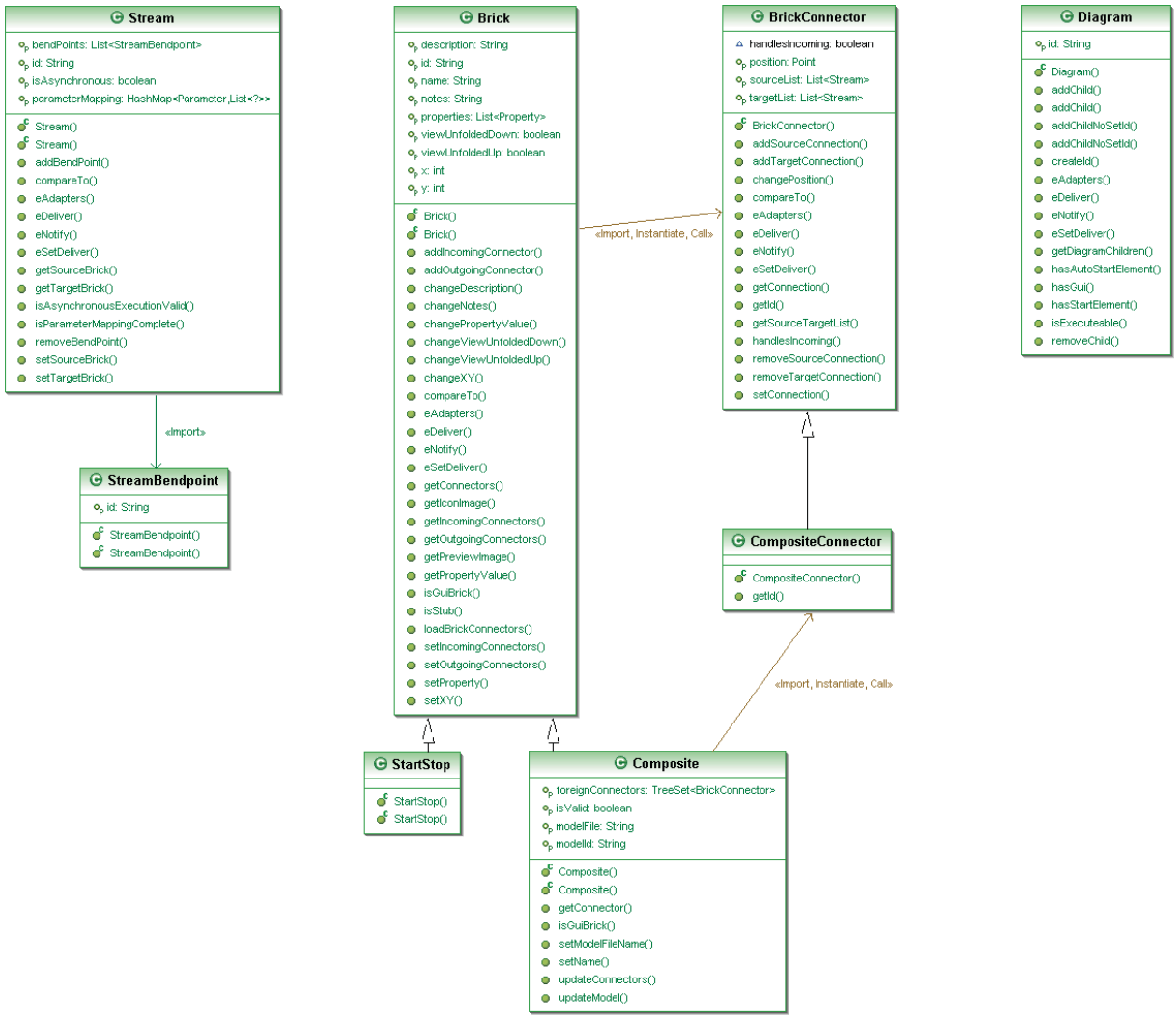


Abbildung 6-46 Übersicht der Bestandteile des Pakets Model
Quelle: Eigene Darstellung

6.2.3.4.2 Properties

Das zweite Paket der Architektur zur Persistierung der Modelle ist das Properties Paket. Die Aufgabe dieses Pakets ist es, die in den asml-description.xml Dateien enthaltenen Informationen über domänenspezifische Konzepte einerseits aus den Dateien zu Laden und andererseits als Objekte für die Modellelemente zur Verfügung zu stellen. Da diese Informationen zur Laufzeit nicht veränderlich sind, werden sie getrennt von den Modellinformationen in diesem Paket gesondert verwaltet. Dies bedeutet, das zwei Instanzen eines Bricks, die das gleiche Konzept beschreiben, durch zwei Objekte des Typs Brick aus dem Model Paket realisiert werden, sich aber gemeinsam eine Instanz der aus der asml-description.xml stammenden Properties teilen.

Das Properties Paket besteht, wie in Abbildung 6-47 zu sehe, aus sieben Klassen. Die zentrale Klasse ist der BrickProperty, welcher alle allgemeinen Informationen über ein Konzept enthält sowie Referenzen auf die weiteren Elemente des Pakets verwaltet. Für Composites und StartStop gibt es hier jeweils eine speziell abgeleitete Variante, da diese Elemente nicht durch eine eigene asml-description.xml beschrieben werden. Für die Verwaltung von Properties wird die Klasse Property verwendet.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Die verfügbaren Eingangs- und Ausgangskonnektoren werden über die Klassen Connector, ParameterVariant und Parameter beschrieben. Die Klasse Connector steht dabei sowohl für Eingangs- als auch für Ausgangskonnektoren. Für einen Konnektor notwendige, oder entsprechend von diesem an den nachfolgenden Stream übergebene, Parameter, werden in der Klasse Parameter realisiert. Da Eingangskonnektoren über unterschiedliche Varianten an Parametern verfügen können, werden Parameter über die Klasse ParameterVariant an Konnektoren gebunden. Ein Konnektor hat also viele ParameterVariants und diese wiederum viele Parameter. Die gleiche Struktur wird auch für Ausgangskonnektoren genutzt. Zwar haben diese keine unterschiedlichen Varianten an zu übergebenen Parametern, dies wird jedoch transparent über genau eine ParameterVariant pro Ausgangskonnektor realisiert.

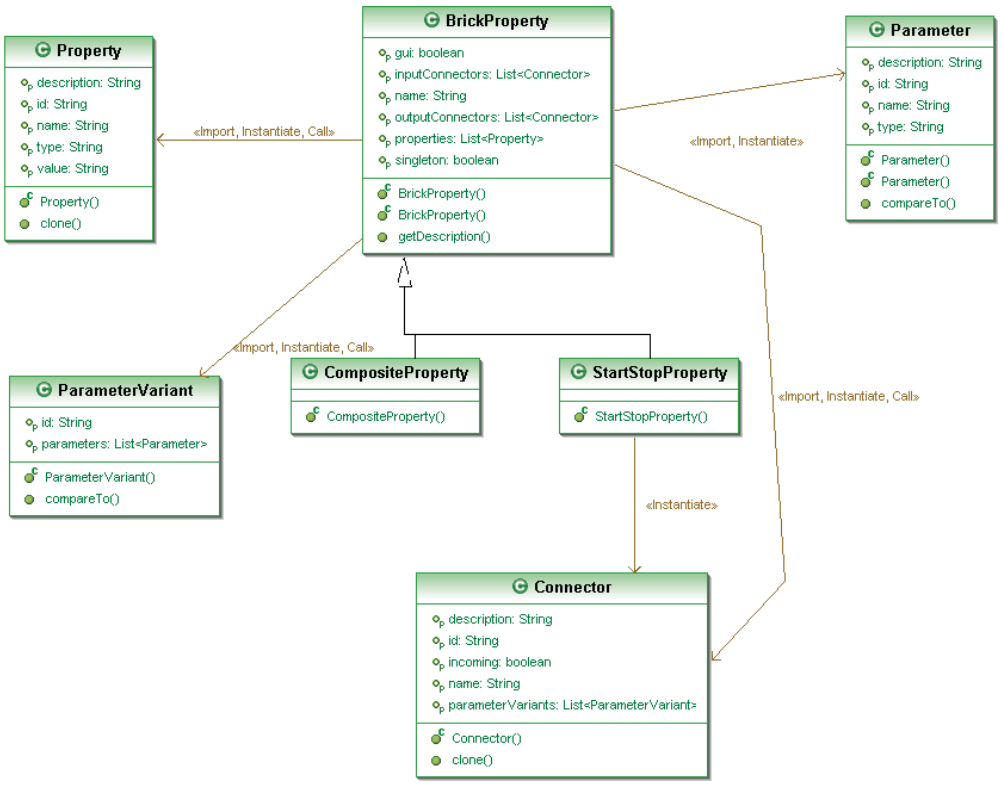


Abbildung 6-47 Übersicht der Bestandteile des Pakets Properties
Quelle: Eigene Darstellung

6.2.3.4.3 Tools

Das letzte Paket ist das Tools Paket. Dieses Paket beinhaltet zwei funktionale Hilfsklassen, die einerseits für das Transformieren von Modellen und andererseits für das Konvertieren von Datentypen genutzt werden.

Die Klasse TypeCaster ist für letztere Aufgabe zuständig. Wie in der Beschreibung der XML Formate der Modelle sowie der Darstellung des Editors bereits diskutiert, verfügen Properties und Parameter über unterschiedliche Datenformate. Beispielsweise wird für die Repräsentation von Text das Format String und für eine Liste von textuellen Elemente eine Liste von Strings verwendet. Die Aufgabe der Klasse TypeCaster ist es einerseits, diese Datentypen nach Möglichkeit ineinander zu Konvertieren oder andererseits, wenn eine Konvertierung nicht möglich ist, dies festzustellen. In der aktuellen Referenzimplementierung dieser Klasse stehen folgende Datentypen zur Verfügung:

- *String* Zeichenkette variabler Länge.
- *Integer* Ganzzahliger Wert.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *Double* Gleitkommazahl.
- *URL* Identifizierung und Lokalisierung eine Ressource.
- *Point2D* Punkt bestehend aus zwei Gleitkommazahlen.
- *Timestamp* Zeitstempel (Millisekunden seit 01.01.1970 00:00:00 GMT).
- *Date* Repräsentation des aktuellen Datums inkl. Uhrzeit.
- *List<String>* Liste von Zeichenkette.
- *List<Integer>* Liste von ganzzahligen Werten.
- *List<Double>* Liste von Gleitkommazahlen.
- *List<URL>* Liste von Ressourcen.
- *List<Point2D>* Liste von Punkten.
- *List<Timestamp>* Liste von Zeitstempeln.
- *List<Date>* Liste von Daten.

Die Klasse ECoreFactory stellt schließlich die zentrale Speicher- und Laderoutine für Modelle bereit. Mit einer Reihe an statischen Funktionen (in Abbildung 6-48 mit einem roten „S“ markiert) werden die verfügbaren domänenspezifischen Konzepte aus ihren asml-description.xml Dateien geladen. Dies erfolgt hier statisch, so dass die Informationen zur Laufzeit nur einmal geladen werden müssen und in der Workbench allen Editoren zur Verfügung stehen. Die Informationen werde also zum einen für das Laden und Speichern von Modellen als zum anderen auch für das Anlegen neuer Bausteine aus der Toolbar des Editors genutzt.

Daneben hat die Klasse noch die Funktionen loadModel, saveModel, saveModelAs und getModelId. Die letztere Funktion lädt nur die erste Zeile eines Modelles, um die darin befindliche ID zu erhalten. Die Funktion loadModel lädt eine asm-Datei in ein Ecore Modell unter Zuhilfenahme der EMF Funktionalität und überführt dieses anschließend in die interne Repräsentation des Model Pakets. Entsprechend transformieren die Methoden saveModel und saveModelAs das interne Modell in ein Ecore Modell und speichern dieses anschließen mit EMF als XML Datei ab. Erstere Funktion überschreibt dabei alte Dateien, die zweite Funktion legt eine Kopie des Modells in einer neuen Datei unter einem neuem Namen an. Bei diesem Vorgang wird auch eine neue Modell ID erstellt, so dass es sich anschließend formal um ein neues Modell handelt.

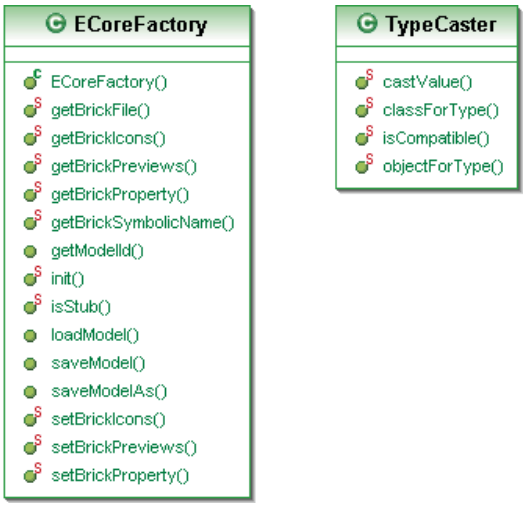


Abbildung 6-48 Übersicht der Bestandteile des Pakets Tools
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

6.2.4 Code Generator: Runtime

Der letzte noch ausstehende Teil der Werkzeugunterstützung für eine domänenspezifische Modellierungssprache ist noch Tolvanen und Kelly (2004) der Code Generator. Wie in der Beschreibung der Architektur in Abbildung 6-1 zu sehen, handelt es sich im vorliegenden Fall um keinen Generator im eigentlichen Sinne, sondern vielmehr um eine Laufzeitumgebung, die die in Form von ASML-Modellen vorliegenden Automotive Services ausführt.

Zu diesem Zweck werden die domänenspezifischen Konzepte unabhängig voneinander in einem domänenspezifischen Framework realisiert und anschließend, auf der Grundlage der in den Modellen hinterlegten Informationen, in der Laufzeitumgebung zum letztendlichen Automotive Service zusammengesetzt. Der nachfolgende Abschnitt beschreibt ausgehend von den technischen Grundlagen des domänenspezifischen Frameworks wie sich die Laufzeitumgebung für den Nutzer darstellt und wie diese schließlich technisch realisiert werden kann.

6.2.4.1 Grundlagen

Wie in Abschnitt 6.1 beschrieben, wird für eine Umsetzung der Laufzeitumgebung für die Automotive Services Modeling Language sowie des dazugehörigen domänenspezifischen Frameworks die Prototypenplattform HIMEPP genutzt. Ziel von HIMEPP ist es, wiederkehrende Funktionalitäten in Form von klar definierten Komponenten zu realisieren um diese in zukünftigen Projekten wiederverwenden zu können. Überträgt man diesen Ansatz auf die Realisierung der domänenspezifischen Konzepte für die Automotive Services Modeling Language, so eignen sich diese HIMEPP Komponenten insbesondere, einzelne Konzepte wiederverwendbar umzusetzen. Darüber hinaus wurde im Rahmen der Entwicklung von HIMEPP die Integration der Prototypen ins Fahrzeug betrachtet. Automotive Services auf dieser Grundlage können also im Fahrzeug unter der Verwendung der Hardware des verfügbaren Infotainmentsystems genutzt werden.

Für eine Integration der Automotive Services in die Serienhardware wird ein spezieller CarPC genutzt. Dieser CarPC stellt im Fahrzeug eine Hardwareplattform zur Verfügung, auf der die in HIMEPP realisierten Automotive Services ausgeführt werden können. Für die Interaktion mit dem Fahrzeug wird der CarPC über einen speziellen Display Signal Manager (DSM) mit dem Bildschirm sowie dem Bedienteil verknüpft. Zusätzlich stehen noch weitere Anschlüsse für die Mobilfunkantenne, die GPS Komponente sowie den CAN-Bus des Fahrzeugs zur Verfügung. Eine detaillierte Beschreibung der Fahrzeugintegration findet sich in Nicolescu (2009, 142).

Abbildung 6-49 stellt die grundlegende Architektur von HIMEPP dar. Die unterste, in Grau dargestellte, Schicht bildet die OSGi Service Platform. Hier wurden zur Realisierung die Eclipse eigene Implementierung Equinox herangezogen. Aufgabe dieser Schicht ist es, ein einheitliches Komponentenmodell zu definieren sowie eine entsprechende Laufzeitumgebung für dieses Komponentenmodell zur Verfügung zu stellen. Elemente dieser Plattform, denen in HIMEPP eine besondere Bedeutung zukommt, sind die OSGi Declarative Services, der EventAdmin sowie die OSGi LogServices. Declarative Services realisieren das Laden und Entladen einzelner Komponenten und deren Abhängigkeiten zur Laufzeit, so dass bestehende Komponenten während der Ausführung entfernt oder ersetzt und neue Komponenten hinzugefügt werden können. Der OSGi EventAdmin übernimmt die Koordination der Kommunikation der einzelnen Komponenten untereinander. Über ein Registrieren am

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

EventAdmin können so über diesen versendete Events empfangen werden. Schließlich liefern die LogServices eine standardisierte Umgebung um Statusnachrichten und Laufzeithinweise in Form von Logdateien zu speichern.

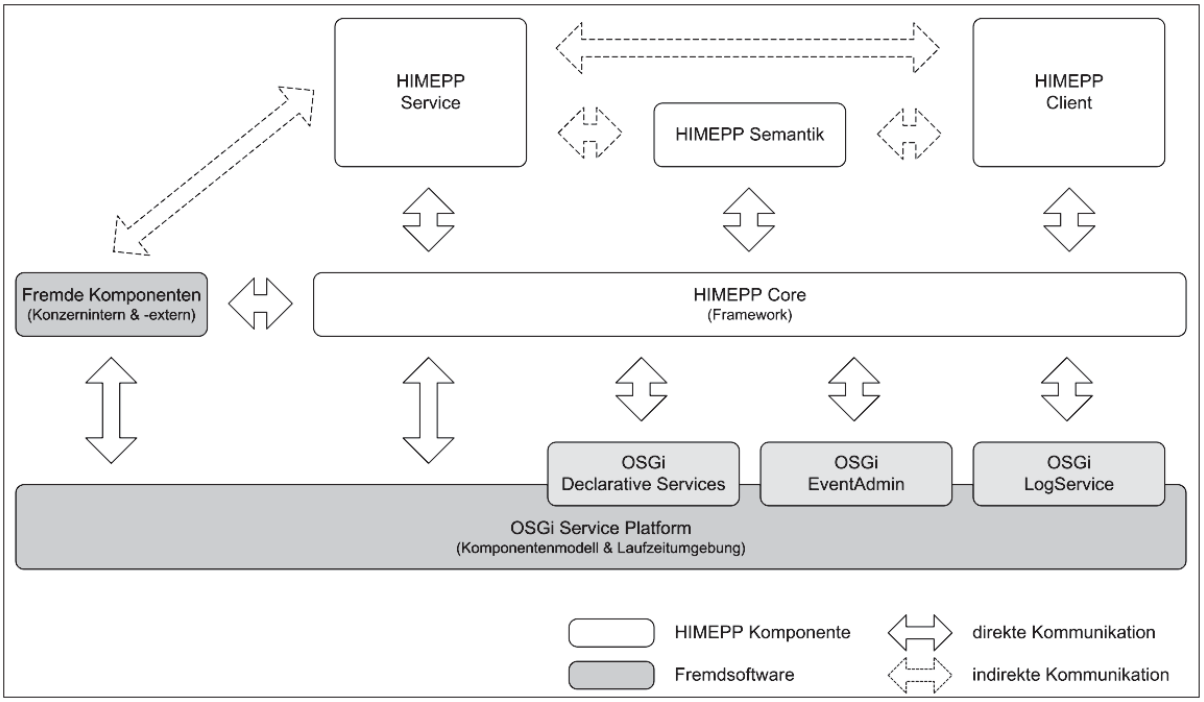


Abbildung 6-49 Übersicht über die HIMEPP Systemarchitektur
Quelle: (Hoffmann 2010, 158)

In der Mitte der Abbildung ist die zentrale Komponente des Frameworks, HIMEPP Core, zu sehen. Die Aufgabe von HIMEPP Core ist es, die Elemente der OSGi Serviceplattform transparent für die weitere Nutzung zur Verfügung zu stellen und wiederkehrende Aufgaben, wie das Anmelden am EventAdmin, zu übernehmen. Aus Sicht der einzelnen HIMEPP Komponenten abstrahiert somit HIMEPP Core die Funktionalität des OSGi Frameworks. Darüber hinaus stellt HIMEPP Core auch eine Schnittstelle für den Zugriff auf externe Komponenten zur Verfügung. Diese können beispielsweise genutzt werden um fahrzeugspezifische Hardware zu realisieren.

Für den Gestalter von Automotive Services sind jedoch im Wesentlichen die drei oben zu sehen Komponenten entscheidend. HIMEPP Services, HIMEPP Semantik und HIMEPP Client bilden die Elemente aus denen Automotive Services zusammengesetzt werden. Einem Service kommt dabei die Aufgabe zu einzelne funktionale Aspekte zu realisieren. Ein Beispiel für eine solche funktionale Komponente ist das Bedienteil im Fahrzeug. Auf der anderen Seite übernehmen Client Komponenten die Realisierung der logischen Funktionalität eines Automotive Services sowie die Realisierung der Interaktion mit dem Nutzer. Zwischen diesen beiden Komponenten finden sich die Semantik Komponenten. Ihre Aufgabe ist es, Service Komponenten und Client Komponenten logisch voneinander zu entkoppeln. Ein Beispiel für eine solche Entkopplung ist wiederum das Bedienteil: Die Service Komponente ist für die technische Kommunikation mit den physischen Bedienteil zuständig und reagiert auf dessen elementare Funktionen, wie beispielsweise das Drehen des Dreh-Drück-Stellers nach links oder nach rechts. Die Semantik Komponente ist für die semantische Interpretation dieser Ereignisse zuständig. Dies ist in diesem Fall der Befehl nach Oben oder nach Unten. Aus Sicht der Client Komponente wird also nicht der Dreh-Drück-Steller gedreht, sondern der

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Befehl nach Oben oder nach Unten ausgelöst. Diese Entkopplung hat den Vorteil, dass anstelle des Dreh-Drück-Stellers auch beispielsweise ein Funktionsknopf oder eine Gestenerkennung diese Aktion auslösen kann.

Für eine Umsetzung der domänenspezifischen Konzepte müssen also Service Komponenten und Client Komponenten realisiert werden. Service Komponenten übernehmen dabei die Aufgabe Funktionskonzepte zu realisieren. Dementsprechend werden Bedienkonzepte über Client Komponenten implementiert. Da die Logik eines Automotive Services im Rahmen der Automotive Services Modeling Language auf das Modell übergeht, verbleibt für die Client Komponente also nur die Realisierung der Interaktion mit dem Nutzer. Semantik Komponenten treten aus Sicht der Modellierung nicht direkt in Erscheinung. Allerdings fällt ihnen auch in diesem Fall weiterhin die Aufgabe zu, Aktionen einzelner Service Komponenten in semantische Befehle zu transformieren, die wiederum die einzelnen Bedienkonzepte beeinflussen. Eine weitere Besonderheit sind nicht in der Modellierung sichtbare Servicekomponenten. So wird beispielsweise das Bedienteil nicht direkt als Brick modelliert, sondern erscheint nur mittelbar als Interaktionsmöglichkeit mit einem Bedienkonzept.

Für eine detaillierte Beschreibung des domänenspezifischen Frameworks HIMEPP wird auf die Arbeiten von Hoffmann (2010) und Nicolescu (2009) verwiesen.

6.2.4.2 Umsetzung

Der Grundgedanke der Laufzeitumgebung der Automotive Services Modeling Language ist es, vom Nutzer gestaltete Modelle ohne weitere Konfiguration oder notwendige Einstellungen durch den Nutzer direkt am Rechner oder im Fahrzeug auszuführen. Dafür werden die einzelnen domänenspezifischen Konzepte, aus denen sich ein Modell zusammensetzt, wie im vorangegangen Abschnitte beschrieben, durch entsprechende Komponenten im HIMEPP Framework realisiert. Dieses Framework wird für eine durchgängige Werkzeugunterstützung von ASML um eine Runtime Schicht erweitert. Die Aufgabe dieser Schicht ist es, Automotive Service Modelle zu Laden, diese zu interpretieren und durch Verknüpfung von realisierten Komponenten auszuführen.

Um dies durchzuführen werden beim Start der ASML Runtime alle, für diese verfügbaren, Modelle mit Hilfe der in Abschnitt 6.2.3.4 beschriebenen Modellimplementierung in den Arbeitsspeicher geladen. Anschließend werden die Modelle mit der im Editor, in Abschnitt 6.2.2, vorgestellt Funktionen der Validierung auf Ausführbarkeit hin untersucht und dem Nutzer anschließend alle Modelle angezeigt, die dieser starten kann. Modelle, die aus Sicht der Runtime nicht ausführbar sind, zum Beispiel durch das Fehlen einer implementierten Komponente in HIMEPP, werden in der Ansicht versteckt. Um diese Aufgabe zu erfüllen, verfügt die ASML Runtime über zwei Anzeigen. Die erste Anzeige ist der so genannte Warte-Bildschirm, der dem Nutzer bei länger andauernden Operationen gezeigt wird. Die zweite Anzeige ist der so genannte Home-Bildschirm, in dem entsprechend die verfügbaren Modelle aufgelistet werden. Abbildung 6-50 zeigt die erste der beiden Anzeigen.

Die beiden häufigsten Situationen, in denen der Warte-Bildschirm zum Einsatz kommt ist das Laden der Automotive Services Modelle beim Initialisieren der ASML Runtime sowie beim erstmaligen Starten einzelner, im Home-Bildschirm aufgelisteter, Modelle. Um die andauernde Aktivität zu signalisieren, wird auf der linken Seite ein animiertes Symbol dargestellt. Das Symbols trifft keine Aussage darüber, wie weit die aktuelle Aktivitäten

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

fortgeschritten ist, sondern signalisiert lediglich das die ASML Runtime aktuell aktiv ist. Rechts neben dem Symbol erscheint noch ein kurzer Hinweistext über die Art der aktuellen Aktivität. Beim Initialisieren der ASML Runtime wird, wie in Abbildung 6-50 dargestellt, „Starting Automotive Services…“ angezeigt.

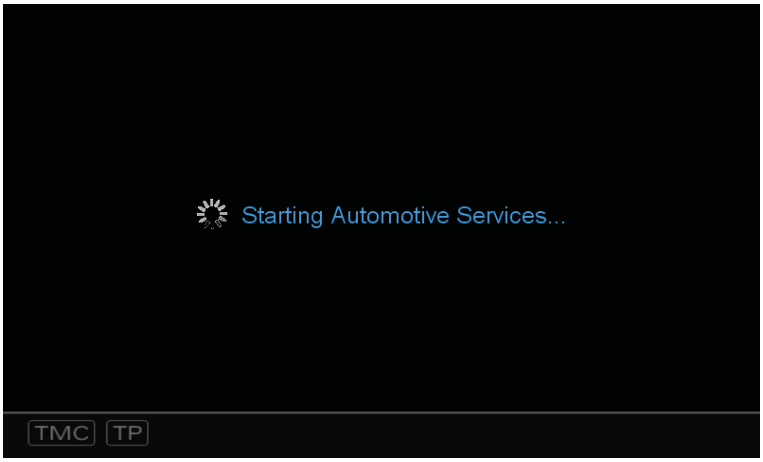


Abbildung 6-50 Warte- Bildschirm der Runtime
Quelle: Eigene Darstellung

Abbildung 6-51 zeigt den Home-Bildschirm. In diesem werden alle verfügbaren, ausführbaren Modelle alphabetisch sortiert nach ihrem Namen angezeigt und können durch Drehen des Dreh-Drück-Stellers und Drücken desselben ausgewählt werden. Sobald der Nutzer auf diese Weise einen bestimmten Service aktiviert hat, analysiert die Runtime das entsprechende Modell und initialisiert alle dazu notwendigen Komponenten. Als nächstes erscheint wieder ein Warte-Bildschirm, der das Starten des jeweiligen Dienstes indiziert. Sobald innerhalb eines Dienstes ein Stop Element erreicht wird, so entfernt die ASML Runtime alle nicht mehr benötigten Komponenten aus dem Arbeitsspeicher, beendet den Dienst und kehrt zum Home-Bildschirm zurück. Eine alternative Möglichkeit zum Home-Bildschirm zurückzukehren hat der Nutzer durch Betätigen der am Bedienteil befindlichen Return-Taste. Ein Druck auf diese Taste pausiert das aktuell laufenden Modell und zeigt den Home-Bildschirm an. Kehrt der Nutzer zu einem, auf diese Weise pausierten, Automotive Service zurück, so wird dessen Ausführung an der letzten aktiven Stelle wieder aufgenommen.

Abbildung 6-51 Home- Bildschirm der Runtime
Quelle: Eigene Darstellung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Für die Ausführung der Modelle spielen Streams innerhalb der Runtime eine herausragende Rolle. Die nachfolgenden Schritte beschreiben das Vorgehen beim Interpretieren und Ausführen von Automotive Service Modellen. Zunächst wird das Start Element des Modells gesucht und alle von diesem Start Elemente ausgehenden Streams initialisiert und ausgeführt (Schritt 4).

Schritt 1: Der Stream wird initialisiert indem alle, in den vorausgegangenen Bricks erstellten, Informationen auf diesen kopiert werden. Handelt es sich beim Stream um einen vom Start Elemente ausgehenden, so sind noch keine Informationen vorhanden und der Stream bleibt somit leer.

Schritt 2: Der Stream wird, unter Zuhilfenahme des OSGi EventAdmin, am Ausgangskonnektor des ausgehenden Bricks angemeldet. Der Stream wartet also darauf, dass der vorangegangene Baustein ein Signal am Ausgangskonnektor anlegt und so den Stream aktiviert. Ist der Ausgangsbrick ein Start Element, wird direkt zu Schritt 4 gesprungen.

Schritt 3: Wird der Stream durch ein Signal am Ausgangskonnektor aktiviert, so werden alle auf diesem Ausgangskonnektor verfügbaren Informationen zu den bereits am Stream vorhandenen Informationen kopiert. Befinden sich auf dem Stream bereits Informationen, die durch die am Ausgangskonnektor verfügbaren Informationen überschrieben werden müssten, so werden die alten Informationen verworfen und die neuen behalten.

Schritt 4: In diesem Schritt wird nun der aktivierte Stream ausgeführt. Zunächst wird das Ausgangsbrick deaktiviert und anschließend das nachfolgende Brick durch Übergabe der im Modell spezifizierten Properties initialisiert und anschließend aktiviert. Der Kontrollfluss ist somit also von ausgehenden Brick auf das nachfolgende Brick übergegangen. Handelt es sich um einen asynchronen Stream, so bleibt auch der Ausgangsbrick aktiv, so dass nun beide Bricks aktiv sind.

Schritt 5: Handelt es sich beim ausgehenden Brick um ein Bedienkonzept, so muss noch entschieden werden, ob dieses ausgeblendet werden muss. Ist das nachfolgende Brick wiederum ein Bedienkonzept, so wird dieses eingeblendet und das vorangegangene ausgeblendet. Handelt es sich beim nachfolgenden Brick um ein Funktionskonzept, so bleibt das alte Bedienkonzept so lange sichtbar, bis ein neues Bedienkonzept aufgerufen wird oder das Modell beendet wird.

Schritt 6: In diesem Schritt werden alle an den Ausgangskonnektor des nachfolgenden Bricks befindlichen Streams initialisiert (Schritt 1). Alle bereits initialisierten ausgehenden Streams des Bricks werden dabei entfernt.

Schritt 7: Nun wird der Eingangskonnektor des nachfolgenden Bricks aufgerufen. Dies geschieht durch Übergabe der im Modelle festgelegten Informationen des Streams an die entsprechenden Parameter des Eingangskonnektors.

Schritt 8: Der Stream wird nun beendet und das nachfolgende Brick wird ausgeführt.

Entlang dieser acht Schritte für die Ausführung von Streams können beliebig komplexe Automotive Service Modelle ausgeführt werden. Einzige Sonderfälle die dabei beachtet werden müssen, sind Start und Stop Elemente. Bei Start Element wird, wie bereits in den acht Schritten beschrieben, der erste Stream mit leeren Informationen initialisiert und ohne Warten auf ein Signal direkt ausgeführt. Beim Stop Element entfällt das Initialisieren der

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

nachfolgende Streams, da das Modell an dieser Stelle beendet wird und somit keine weiteren Streams mehr möglich sind.

Eine weitere Besonderheit die beachtet werden muss sind Composites. Da es sich aus funktionaler Sicht bei Composites um Platzhalter für Teilmodelle handelt, können diese ebenfalls mit Hilfe der acht Schritte zur Ausführung von Streams realisiert werden. Allerdings müssen an Composites die richtigen Eingangs- und Ausgangskonnektoren verknüpft werden. Anstelle des Eingangskonnektors am Composite ruft ein Stream direkt den Eingangskonnektor des Bricks auf, auf den der Konnektor am Composite verweist. Die zweite Ausnahme sind analog dazu die Ausgangskonnektoren des Composites. An diesen Stellen dürfen nicht die ausgehenden Streams des, das Composite beschreibenden, Modells sondern die am Konnektor des Composites ausgehenden Streams initialisiert werden.

Neben den beiden Anzeigen für den Warte- und den Home-Bildschirm verfügt die ASML Runtime noch über eine Simulation des Bedienteils im Fahrzeug. Durch diese, mit der Maus bedienbare, grafische Repräsentation können Automotive Services auch ohne spezielle Hardware oder eine Integration ins Fahrzeug am Rechner simuliert werden. Das simulierte Bedienteil löst dabei in der Runtime die gleichen Aktivitäten aus wie die reale Variante im Fahrzeug. Für die Realisierung des Simulators wird auf das HIMEPP FrontControllerFrame zurückgegriffen (Hoffmann 2010, 225). Abbildung 6-52 zeigt die Bedienteilsimulation.

Abbildung 6-52 Simulation Bedienteil (Audi MMI 2G)
Quelle: Eigene Darstellung

Durch ihre dynamische Architektur verfügt die ASML Runtime über die Möglichkeit, neue Modelle zur Laufzeit in die Liste der verfügbaren Automotive Services zu übernehmen und diese so dem Nutzer verfügbar zu machen. Ein neues Modell wird einfach auf dem Home Bildschirm an der entsprechenden Stelle in die Liste der Services eingefügt. Des Weiteren können auch aktualisierte und erweiterte Versionen bereits in der Runtime verfügbarer Automotive Services zur Laufzeit nachgeladen werden. Wird also ein Modell geändert, so erkennt die ASML Runtime dies, und führt bei zukünftiger Aktivierung des entsprechenden Automotive Services durch den Nutzer das neue Modell aus. Sollte dieser Service während der Aktualisierung des Modells bereits aktiv sein, so wird das alte Modell bis zu dessen Ende weiterhin ausgeführt. Erst bei einem erneuten Start des Automotive Services wird das entsprechende neue Modell genutzt.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

6.2.4.3 Architektur

Für die Umsetzung der ASML Runtime wurde mit HIMEPP die gleiche Funktionalität wie auch für das realisieren der einzelnen domänenspezifischen Konzepte genutzt. Die Runtime ist selber als HIMEPP Komponente realisiert und stellt im Sinne der in Abbildung 6-1 beschriebenen HIMEPP Architektur eine Client Komponente dar. Für die Umsetzung der Komponente wurden die in Abbildung 6-53 dargestellten drei Pakete Runtime, Worker und Tools realisiert. Das Runtime Paket übernimmt das Starten und Stoppen von Modellen, die Steuerung der zwei verfügbaren Anzeigen, das Initialisieren des Bedienteilsimulators sowie die Überprüfung auf neue und geänderte Modelle. Das Worker Paket beinhaltet die Funktionalität zum eigentlichen Ausführen der Automotive Service Modelle. Das Tools Paket stellt eine Erweiterung der OSGi Declarative Services dar und kümmert sich um das Laden und Initialisieren aller notwendigen Komponenten in Abhängigkeit vom jeweils realisierten Modell.

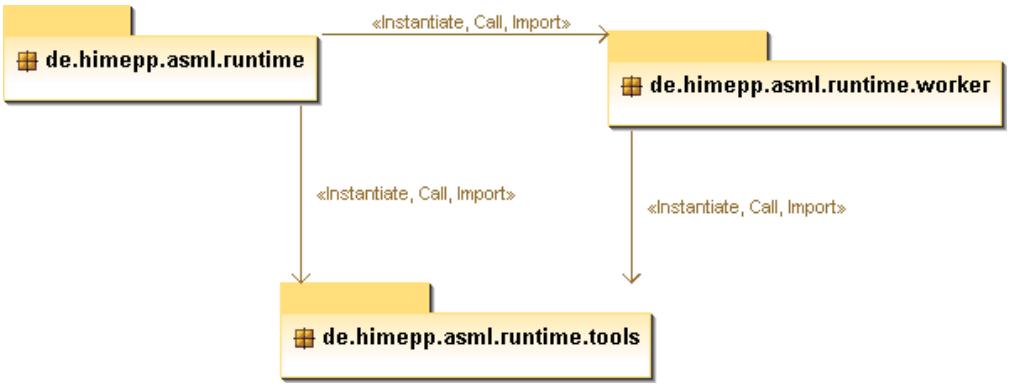


Abbildung 6-53 Übersicht der Bestandteile der Runtime
Quelle: Eigene Darstellung

6.2.4.3.1 Runtime

Das Runtime Paket besteht, wie in Abbildung 6-54 zu sehen, aus zwei Klassen. Die Klasse Runtime wird von HIMEPP beim Laden der Runtime initialisiert und kümmert sich anschließend um die weiteren verwaltenden Aufgaben derselben. Zunächst wird die Klasse SplashScreen instanziiert, welche den Warte-Bildschirm realisiert. Anschließend werden alle verfügbaren Modelle geladen und auf Ausführbarkeit überprüft. Schließlich erfolgt die Anzeige des Home-Bildschirms. Zur Umsetzung des Home Bildschirms wird das Bedienkonzept Liste benutzt. Desweiteren steuert die Klasse das Starten, Stoppen und Pausieren von Automotive Service Modellen und überwacht den Ordner mit den verfügbaren Modellen auf Änderungen.

Abbildung 6-54 Übersicht der Bestandteile des Pakets Runtime
Quelle: Eigene Darstellung

6.2.4.3.2 Worker

Für die Ausführung der Modelle sind die beiden Klasse WorkerThread und StreamWorker im Worker Paket zuständig. Für jedes aktive Modell wird von der Runtime ein neuer

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

WorkerThread in einem neuen Prozess initialisiert. Dieser ist für die Ausführung des Modells zuständig und implementiert die Runloop, die für das Starten, Stoppen und Pausieren von Services zuständig ist. Darüber hinaus verfügt die Klasse noch über Methoden um Komponenten entsprechend der Properties im Modell zu initialisieren.

Die Klasse StreamWorker realisiert schließlich die in Abschnitt 6.2.4.2 beschriebenen acht Schritte für die Ausführung von Streams. Es ist also bei der Ausführung immer abwechselnd ein Brick (HIMEPP Komponente) und ein Stream aktiv, wobei der WorkerThread über den jeweiligen Übergang der Aktivität wacht und diesen bei Bedarf pausieren oder beenden kann.

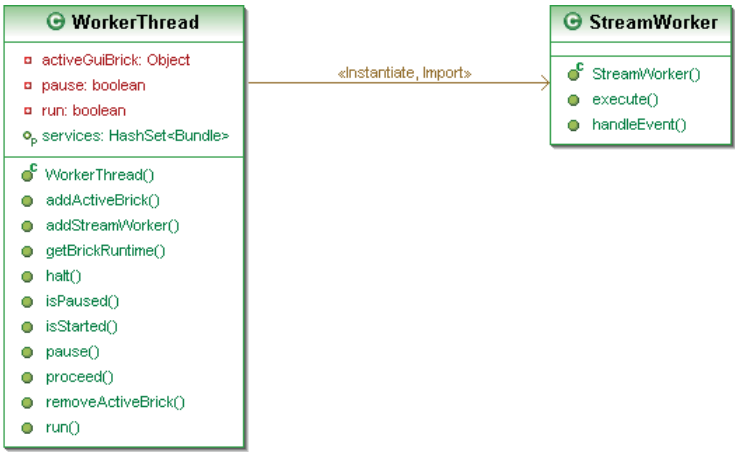


Abbildung 6-55 Übersicht der Bestandteile des Pakets Worker
Quelle: Eigene Darstellung

6.2.4.3.3 Tools

Das dritte Paket der Runtime beinhaltet die Klasse DependencyLoader. Diese erweitert die Funktionalität der OSGi Declarative Services, indem ausgehend von den in einem Modell verwendeten Bestandteilen die entsprechenden Realisierungen in HIMEPP gesucht und entsprechend geladen werden. Diese Klasse kommt also zum einen beim Ausführen von Modellen und zum anderen bei Validieren von Modellen durch die Runtime zum Einsatz.

Abbildung 6-56 Übersicht der Bestandteile des Pakets Tools (Runtime)
Quelle: Eigene Darstellung

6.3 Zusammenfassung

Mit der Gestaltung einer durchgängigen Werkzeugunterstützung konnte die in Abschnitt 5.1 entwickelnde domainspezifischen Modellierungssprache ASML zu einer vollständigen Modellierung ausgebaut werden. Erst diese Ergänzung des Modellierungskonzepts um die, für die Gestaltung notwendigen Werkzeuge und die zur Ausführung dieser modellierten Automotive Services genutzten Bestandteile, macht die Modellierung zu einem wertschöpfenden Artefakt. Die Architektur dieser Umsetzung orientiert sich an einen zweistufigen Prozess, der von der Produktidee über das Produktmodell zum funktionierten Produktcode führt.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Der erste Schritt dieses Vorgehens besteht also darin Ideen, die einzelne Stakeholder zu Automotive Services haben, unter Zuhilfenahme der entwickelten Modellierungssprache in einem Modellierungswerkzeug zu einem Automotive Service, also einem Produktmodell, zu entwickeln. Dieser Schritt ist im Sinne des Design Thinking an mehreren Stellen notwendig, um zum Beispiel Erfahrungswissen für zukünftige Iterationen verfügbar zu machen oder einzelne Ideen zu detaillieren und zu kommunizieren.

Das Modellierungswerkzeug setzt sich dabei aus zwei Bestandteilen zusammen. Der erste Bestandteil ist die ASML Workbench, welche aus Sicht des Nutzers das Modellierungswerkzeug darstellt und für die Verwaltung von Modellen und diese unterstützende Funktionen zuständig ist. Der zweite Bestandteil ist der ASML-Editor. Dieser ist eng in die ASML Workbench integriert und stellt im engeren Sinne das eigentliche Modellierungswerkzeug dar. Durch einfache grafische Funktionen wie Drag'n'Drop oder das Editieren von Properties direkt im Modell versetzte der Editor Stakeholder in die Lage, bestehende domänenspezifische Konzepte, also Bedienkonzepte und Funktionskonzepte, in Form von Bricks zu neuen Automotive Services zu kombinieren. Darüber hinaus können mit Hilfe eines Wizards auch neue oder geänderte Konzepte in Form von Stubs spezifiziert und die Modellierungssprache auf diese Weise erweitert werden.

Das mit dem Modellierungswerkzeug entwickelte Produktmodell setzt sich aus einer Reihe an Bestandteilen zusammen. Zunächst müssen die domänenspezifischen Konzepte der Modellierungssprache unabhängig von deren Realisierung und unabhängig von der Notation und den Regeln der Modellierungssprache gespeichert werden. Dies geschieht in Form spezieller XML Dokumente. Des Weiteren müssen die eigentlichen Produktmodelle gespeichert werden. Dies geschieht ebenfalls in speziell formatierten XML Dokumenten, die durch die Dateiendung *„asm"* markiert werden. Erst die Kombination aus Modellen und Konzepten, also das Zusammenspiel verschiedener Dateien, fügt sich somit zu einem kompletten Produktmodell zusammen.

Der zweite Schritt hat nun die Transformation der beschriebenen und gespeicherten Produktmodelle in ausführbare Automotive Services zur Aufgabe. An dieser Stelle wurde nicht das klassische Vorgehen (Tolvanen/Kelly 2004), welches eine Transformation des domänenspezifischen Produktmodells in ausführbaren Code vorsieht, sondern ein direktes Ausführen des Produktmodells durch eine Laufzeitumgebung realisiert.

Die resultierenden Automotive Service setzten sich also wiederum aus zwei Bestandteilen zusammen. Zum einen werden die funktionalen Aspekte der domänenspezifischen Konzepte in Form eines domänenspezifischen Frameworks auf konventionelle Weise realisiert, also von Entwicklern implementiert. Das bedeutet, dass die von den Stakeholdern definierten Stubs durch eine Implementierung in Form von Komponenten zu vollwertigen Bricks erweitert werden. Diese werden wiederum im Produktcode verwendet, um die Funktionen, die einzelne Bricks darstellen, auszuführen. Neben den beteiligten Bricks und deren Konfiguration über Properties beinhaltet das Modell des Weiteren noch den Kontroll- und Informationsfluss zwischen den einzelnen Bestandteilen. Diese Informationen werden von der ASML Runtime interpretiert und mit Hilfe eines, aus acht Schritten bestehenden, Regelwerks ausgeführt. Die Kombination aus Komponenten des domänenspezifischen Frameworks und der Interpretation des Informations- und Kontrollflusses durch die ASML Runtime realisiert somit den angestrebten Produktcode, also den Automotive Service.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Mit den vorgestellten Werkzeugen und der entwickelten Modellierungssprache stehen nun alle Bestandteile zur Verfügung, um Stakeholder bei der Gestaltung von Automotiv Services zu unterstützen. Der nachfolgende Abschnitt beschäftigt sich daher mit der Evaluation der in Abschnitt 5.1.1 aufgestellten Hypothesen, die aus den entwickelten Designprinzipien für die Modellierungssprache resultieren.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

7 Evaluation der Modellierung für Automotive Services

Das Ziel einer gestaltungsorientierten Forschung, wie sie diese Arbeit darstellt, ist es Problemstellungen, die sich in der praktischen Anwendung finden, durch die Gestaltung von Artefakten zu lösen (Briggs 2006). Um dies zu erreichen wurde im Rahmen der Unterstützung der Gestaltung von Automotive Services ein domänenspezifisches Modellieren eben dieser Services entwickelt, um den dabei auftretenden Herausforderungen (vgl. Abschnitt 3) zu begegnen.

In Abschnitt 4 werden auf der Grundlage des Vorgehens des Design Thinking die Anforderungen an eine solchen Modellierung spezifiziert. Ausgehend von den Phasen des Design Thinking (vgl. Abschnitt 4.2.1), unter Berücksichtigung der beteiligten Stakeholdergruppen (vgl. Abschnitt 3.1), wurden drei zentrale Herausforderungen an eine Modellierung von Automotive Services abgeleitet. So müssen die beiden Gruppen der nichttechnischen Stakeholder, insbesondere die der sonstigen Stakeholder, in die Lage versetzt werden ihre Visionen und Anforderungen an Automotive Services zu explizieren.

Darüber hinaus sollen diese Explikationen der modellierten Anforderungen in einer implementierbaren Repräsentation des entsprechenden Automotive Services resultieren. Schließlich sollen bereits gestaltete Lösungsaspekte im Rahmen eines Erfahrungsmanagements nutzbar gemacht werden. Unter Berücksichtigung bestehender Erkenntnisse über, diesen Herausforderungen analoge, Theorien, wurden in Abschnitt 5.1.1 drei entsprechende Designprinzipien in der Modellierungssprache realisiert (van Aken 2004; Schermann/Böhmann/Krcmar 2007). Die nachfolgenden Abschnitte befassen sich daher mit der Evaluation der drei gestaltenden Designprinzipien.

7.1 Methoden der Artefaktevaluation

Die Aufgabe von Evaluation im Kontext der gestaltungorientierten Forschung ist die Bewertung von Artefakten, also von Konstrukten, Methoden, Modellen und Instanziierungen (March/Smith 1995b), hinsichtlich ihrer Nützlichkeit, Qualität und Effizienz bei der Lösung eines spezifischen Problems (Hevner et al. 2004, 17). Um dieser Aufgabe nachkommen zu können, stellen Hevner et al. (2004, 18) eine Reihe an Methoden zur Artefaktevaluation zusammen. Diese können, wie in Abbildung 7-1 zu sehen, in beobachtende, analytische, experimentelle, testende und beschreibende Methoden gegliedert werden.

Art	Methode	Beschreibung
Beobachtend	Fallstudie	Untersuchung des Artefaktes im jeweiligen Geschäftsumfeld
	Feldstudie	Beobachtung der Nutzung des Artefakts in Projekten
Analytisch	Statische Analyse	Untersuchung der Struktur des Artefaktes
	Architekturanalyse	Untersuchung der Integrationsfähigkeit des Artefaktes in die technische Infrastruktur

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

	Optimierung	Demonstration der Optimalität des Artefaktes für den Zweck oder Aufzeigen der Grenzen der Optimierung
	Dynamische Analyse	Untersuchung des Laufzeitverhaltens des Artefaktes
Experimentell	Kontrolliertes Experiment	Untersuchung von Artefakteigenschaften unter kontrollierten Bedingungen
	Simulation	Ausführen des Artefakts unter Nutzung nichtrealer Daten
Testend	Funktionale Tests	Verwenden der Schnittstellen des Artefakts um Fehler zu finden (Black Box Tests)
	Strukturelle Tests	Testen der inneren Funktionsweise des Artefakts um Fehler zu finden (White Box Tests)
Beschreibend	Expertenwissen	Begründen der Nützlichkeit des Artefakts durch Informationen aus der Wissensbasis
	Szenarios	Erstellung von Verwendungsszenarios um die Nützlichkeit des Artefakts zu zeigen

Tabelle 7-1 Methoden der Artefaktevaluation
Quelle: Eigene Darstellung nach (Hevner et al. 2004, 18) übersetzt in (Hoffmann 2010, 245)

7.2 Evaluation der Designprinzipien der Modellierungssprache

Da sich die Herausforderungen jeweils auf ein empirisch überprüfbares Wirkungskonstrukt beziehen, bilden sie zusammen mit den Designprinzipien Ursache-Wirkungs-Beziehungen die im Rahmen der Evaluation überprüft werden können. Dies wird in der vorliegenden Arbeit in Anlehnung an (Weiss/Küchler 1988, 63) durch eine Evaluation der in Abschnitt 5.1.1 aufgestellten Hypothesen realisiert:

Hypothese 1: Stakeholder mit geringen bis gar keinen technischen Kenntnissen der Automotive Domäne und ihrer IT-Komponenten sind in der Lage, realisierbare Automotive Services zu gestalten.

Hypothese 2: Stakeholder mit geringen bis gar keinen technischen Kenntnissen der Automotive Domäne und ihrer IT-Komponenten sind in der Lage, umsetzbare Anforderungen an eine Implementierung einzelner Komponenten zu definieren.

Hypothese 3: Stakeholder sind in der Lage, existierende Elemente wieder zu verwenden um neue Funktionalitäten für künftige Nutzer zu definieren und so die Fähigkeiten der Automotive Services Modeling Language zu erweitern.

Jede dieser Hypothesen beschreibt dabei einen angestrebten Effekt, der durch den Einsatz des Artefakts erreicht werden kann. Die Aufgabe der Evaluation ist es nun zu zeigen, dass unter Einhaltung der in den Hypothesen gestellten Bedingungen ein Einsatz der Modellierungsprache zusammen mit der durchgängigen Werkzeugunterstützung die durch die Hypothesen beschriebenen Effekte hervorruft. Gehlert et al. (2009) beschreiben dieses

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Vorgehen als eine „outcome-oriented" Evaluation. Ziel ist es, „to show whether the IT artifact attains its goals" (Gehlert et al. 2009).

Dieser Aufgabe kann im Rahmen einer beobachtenden Evaluation erfüllt werden. Stakeholder aus den unterschiedlichen Kompetenzprofilen, also Automotive Service Experten, Experten der Anwendungsdomäne, Experten der Umsetzungsdomäne sowie sonstige Stakeholder werden bei der Durchführung einer speziellen Aufgabe beobachtet. Diese Aufgaben richten sich an der, durch die Modellierung ermöglichten, Kommunikation und Zusammenarbeit der einzelnen Gruppen aus und beziehen sich jeweils auf eine oder mehrere der zu überprüfenden Hypothesen. Da eine klare Trennung der Überprüfung der ersten und zweiten Hypothese im Kontext von realen Aufgabestellungen kaum möglich ist, wurden diese beiden Aspekte daher zusammen untersucht.

Nach Hevner et al. (2004, 18) kommen für eine solche, beobachtende Evaluation zwei unterschiedliche Methoden in Frage: Fallstudien und Feldstudien. Beide Methoden kann man einsetzen Aussagen über ein Artefakt zu erlangen, indem Probanden bei der Durchführung einer Aufgabe unter Zuhilfenahme des zu untersuchenden Artefakts beobachte werden. Darüber hinaus können auch Aussagen über das entstandene Ergebnis dieses Bild vervollständigen. Fallstudien untersuchen einen bestimmten, speziell gestalteten, Fall, wohingegen Feldstudien den Einsatz in der natürlichen Umgebung beschreiben. Für die vorliegende Arbeit wurde die Durchführung von drei Fallstudien gewählt, da die durchzuführenden Aufgaben speziell an die zu untersuchenden Hypothesen angepasst und die Studien ausschließlich zum Zweck der Evaluation durchgeführt wurden.

7.2.1 Fallstudie I: Lange Nacht der Wissenschaften

Das Ziel der ersten Fallstudie ist die Überprüfung der ersten beiden Hypothesen. Es soll also überprüft werden, ob Stakeholder, unter Nutzung bestehender domänenspezifischer Konzepte, durch die Modellierung in die Lage versetzt werden eigen Ideen als Automotive Services zu spezifizieren. Darüber hinaus wird des Weiteren der Aspekt fehlender Konzepte betrachte, d.h. ob Stakeholder in der Lage sind neue, noch nicht definierte domänenspezifische Konzepte zu explizieren, die anschließend auf der Grundlage dieser Explikation implementiert werden können.

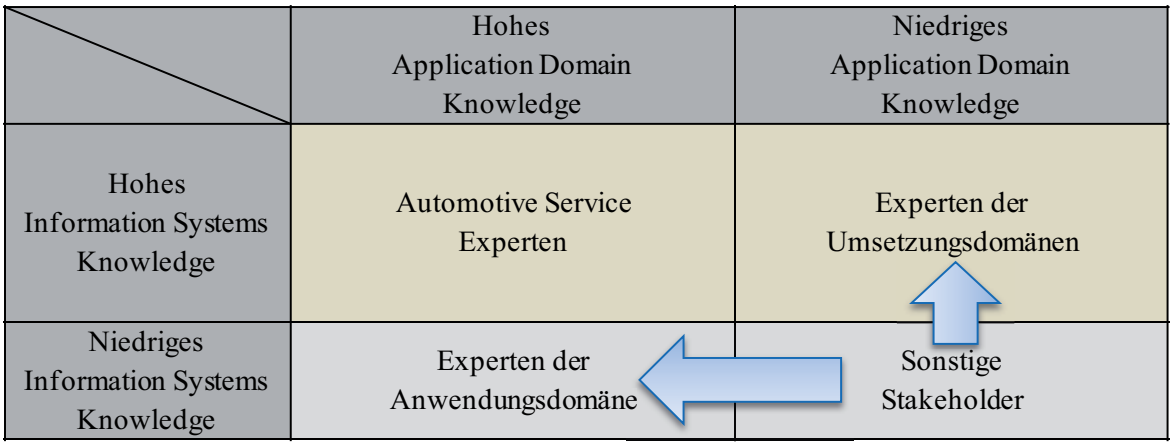


Abbildung 7-1 Beteiligte Stakeholder der ersten Fallstudie
Quelle: Eigene Darstellung

Diese Überprüfung der Hypothesen wird im Kontext der Kommunikation und Kooperation der Gruppe der sonstigen Stakeholder mit den Experten der Anwendungsdomäne und den

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Experten der Umsetzungsdomäne durchgeführt. Abbildung 7-1 stellte diesen Zusammenhang graphisch dar. Die Frage lautet in diesem Kontext also, ob einerseits Experten der Umsetzungsdomäne in der Lage sind, mit den von den sonstigen Stakeholdern definierten Stubs neue Bricks zu realisieren und andererseits Experten der Anwendungsdomäne das explizierte Wissen der sonstigen Stakeholder als Grundlage für neue, domänenspezifische Automotive Services nutzen können.

7.2.1.1 Vorgehen

Die Fallstudie wurde im Rahmen der Langen Nacht der Wissenschaften am 15. Mai 2010 am Campus Garching der Technischen Universität München durchgeführt. An einem Messestand konnten sich Besucher über das Projekt informieren und direkt vor Ort eigene Automotive Services unter Zuhilfenahme der Modellierung gestalten. Diese Modelle wurden anschließend an einen Vertreter der Gruppe der Experten der Anwendungsdomäne und einen Vertreter der Gruppe der Umsetzungsexperten zur weiteren Verwendung übergeben.

7.2.1.1.1 Probanden

Aufgrund der Tatsache, dass die Veranstaltung im Rahmenprogramm des in München zu dieser Zeit stattfinden ökumenischen Kirchentages stattfand, konnte sichergestellt werden, dass die an der Studie beteiligten Probanden keine besondere technische Affinität aufwiesen und somit dem durchschnittlichen Autofahrer entsprechen. Neben einer kurzen Präsentation der Modellierungssprache und der dazugehörigen Werkzeuge haben die Probanden keine explizite Schulung hinsichtlich ASML oder HIMEPP erhalten. Die insgesamt sieben Besucher, die einen eigenen Dienst modelliert haben, können somit als Vertreter der Gruppe der sonstigen Stakeholder angesehen werden.

Als Vertreter der Gruppe der Experten der Anwendungsdomäne wurde der Projektbearbeiter des, zu diesem Projekt parallel stattfinden, Projekts „Service Innovationsprozess am Beispiel der Elektromobilität“ (eIT) gewählt (vgl. Abschnitt 4.3.1). Die Betrachtung der Innovationsprozesse für Services rund um die Fragestellungen der Elektromobilität ermöglicht dem Probanden, ein eingehendes Wissen über diese Anwendungsdomäne zu erlangen. Neben der Untersuchung konzeptioneller Fragestellungen hinsichtlich Einsatzszenarios und Geschäftsmodellen für Elektromobilität ist es auch die Aufgabe dieses Projekts für diese Anwendungsdomäne geeignete und unterstützende Automotive Services zu finden. Hinsichtlich der Modellierung von Automotive Services wurde der Proband in der Nutzung von ASML und den dazugehörigen Werkzeugen geschult, verfügt jedoch über keine Expertise hinsichtlich der Umsetzung domänenspezifischer Konzepte in HIMEPP.

Der Projektbearbeiter des ebenfalls parallel stattfinden Projekts „Ubiquitous Automotive Services Environment“ (uBASE) (vgl. Abschnitt 4.3.1) ist entsprechend der Experte der Umsetzungsdomäne in dieser Fallstudie. Da es die Aufgabe von uBASE ist, die Möglichkeiten von Automotive Services hinsichtlich einer Integration von Backendkomponente und mobilen Geräten sowohl konzeptionell als auch in der Umsetzung zu betrachten, stellt der Projektbearbeiter einen geeigneten Vertreter dieser Gruppe dar. Der Proband hat, analog dem Vertreter der Anwendungsdomäne, einen Schulung in ASML und den dazugehörigen Werkzeugen erhalten. Zusätzlich nahm der Proband auch an einer HIMEPP Schulung teil, in der sowohl die Anwendung als auch die Entwicklung von HIMEPP Komponenten vermittelt wurde. Über ein spezielles Wissen hinsichtlich einer bestimmten Anwendungsdomäne verfügt der Proband jedoch nicht.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

7.2.1.1.2 Durchführung der Fallstudie

Wie bereits erwähnt wurde die Fallstudie auf der Langen Nacht der Wissenschaften mit zufälligen Probanden durchgeführt. Insgesamt wurden sieben Automotive Service Modelle erstellt von denen eines im Nachgang für die Fallstudie ausgewählt wurde. Für das Erstellen der Modelle standen den Probanden ein Flipchart, Stifte und vorgefertigte Bricks zum Aufkleben zur Verfügung. Für das Modellieren auf der Messe wurde der Einsatz von Flipcharts dem Einsatz des softwarebasierten Werkzeugs vorgezogen. Dies hatte einerseits den Grund die Hemmschwelle der Besucher zum Teilnehmen zu senken und andererseits auch Besucher mit nur geringer bis gar keiner technischen Affinität, die teils Schwierigkeiten mit dem Umgang mit Computern haben, ein Mitmachen zu ermöglichen. Ein in ASML geschulter Standmitarbeiter stand den Probanden zusätzlich als Hilfestellung zur Verfügung.

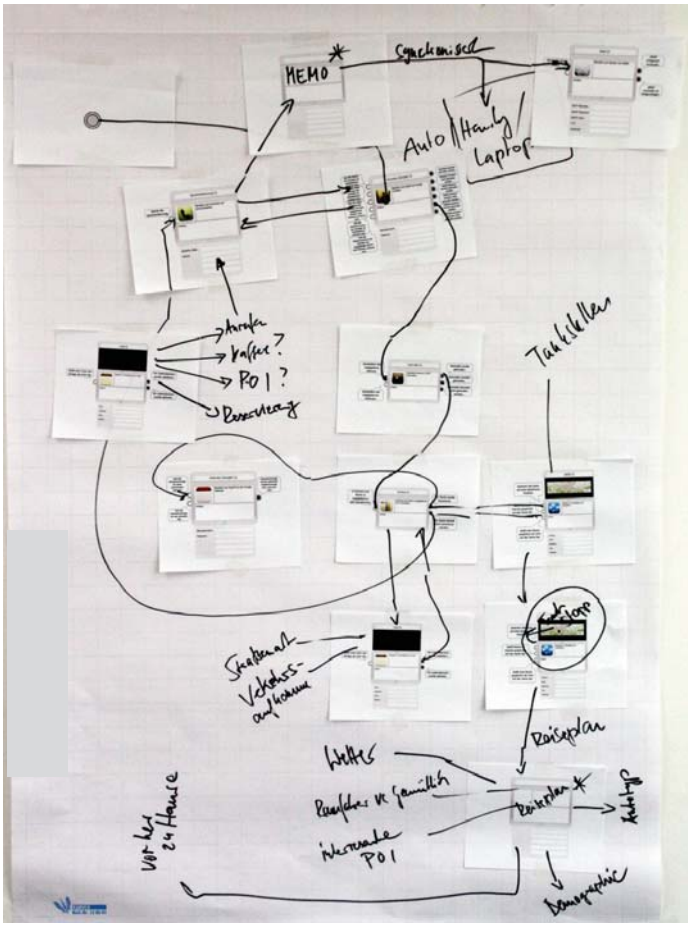


Abbildung 7-2 Automotive Service der ersten Fallstudie: Reiseplanung
Quelle: Eigene Darstellung (Fotographie des Flipcharts)

Ein interessanter, entstandener Automotive Service ist in Abbildung 7-2 zu sehende. Unter dem Einsatz einer Reihe von bestehenden Funktions- und Bedienkonzepten wurde ein komplexer Automotive Service entwickelt, mit dessen Hilfe Nutzer im Fahrzeug, abhängig von ihrem Terminplan und externen Faktoren, wie verfügbarer Tankstellen, eine detaillierte Reiseplanung zur Verfügung gestellt bekommen. An bestehenden Funktionskonzepten wurde die Spracherkennung, Kontakte, GeoCoder, Mail und Routing verwendet (vgl. Abschnitt 5.1.2.3). Als Bedienkonzept kam der Liste Brick zum Einsatz. Neben der Wiederverwendung dieser Konzepte wurden auch neue, domänenspezifische Konzepte vorgeschlagen. So wird ein Memo Stub in das Modell integriert, mit dessen Hilfe der Nutzer Sprachnachrichten

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

aufzeichnen kann. Des Weiteren wurde noch ein Stub zur Reiseplanung erstellt, der sich um die Berechnung des eigentlichen Reiseplans kümmert.

Nach Übergabe dieses Modells and den Umsetzungsexperten konnte dieser einen der beiden neuen Stubs in Form eines Bricks im domänenspezifischen Framework HIMEPP realisieren. Aus dem Memo Stub ist der Baustein „Audio Recorder“ entstanden. Der Brick verfügt über zwei Eingangskonnektoren zum Starten und Stoppen der Aufnahme sowie über zwei Ausgangskonnektoren, die den Beginn und das Ende der Aufnahme signalisieren. Für die Umsetzung des Stubs wurde dieser vom Umsetzungsexperten hinsichtlich der für die Realisierung notwendigen Details erweitert. Beispielsweise wurden die exakten Bezeichnungen der Konnektoren erst jetzt festgelegt. Der in Abbildung 7-3 dargestellte entstandene Brick wurde in der folgenden dritten Fallstudie eingesetzt.

Abbildung 7-3 Darstellung des Audio Recorder Brick
Quelle: Eigene Darstellung

Aus Sicht des Domänenexperten diente das übergebene Automotive Service Modell als Grundlage für die Ausgestaltung eines darin befindlichen Teilaspekts in einen, in Fallstudie zwei umgesetzten, Automotive Service. Ein in der Reiseplanung enthaltener Teilaspekt ist die verbleibende Reichweite des Fahrzeugs, welche für einen Planung notwendiger Tankstellen entscheidend ist. Aufgrund kapazitativer Beschränkung der Probanden wurde nicht der gesamte Reiseplanungsdienst übernommen, sondern nur der beschriebene Teilaspekt weiter betrachtet. Um dies durchzuführen wurde das Modell vom Experten der Anwendungsdomäne hinsichtlich fachlicher Aspekte erweitert und bezüglich des genauen Ablaufs des Automotive Services ausgestaltet. Das in Abbildung 7-5 dargestellte Ergebnis, die „my Reichweite App“, wird in Fallstudie zwei vorgestellt.

7.2.1.2 Beobachtung

Während der Durchführung der Fallstudie konnten eine Reihe von Beobachtungen hinsichtlich des Einsatzes und der Nützlichkeit der Automotive Services Modelling Language und der dazugehörigen Werkzeugunterstützung gemacht werden.

Aus Sicht der sonstigen Stakeholder konnte der Modellierungsansatz genutzt werden, eigene Ideen und Vorstellungen in Form von Automotive Service Modellen zu explizieren. Besonders ist dabei anzumerken, dass alle Probanden in der Lage waren die entstandenen Modelle zu verstehen und diese sukzessiv zum gewünschten Service auszubauen. Allerdings wurden die Probanden bei der Erstellung der Modelle von einem Standmitarbeiter unterstützt. Dies kam besonders bei den formalen Aspekten der Modellierung zum Einsatz. Zwar hatten alle Probanden keine Probleme damit ihre Ideen und Vorstellungen anzugeben, bei der Ausgestaltung dieser als Modelle waren sie hingegen des Öfteren auf die Hilfe des Mitarbeiters angewiesen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Hier viel insbesondere auf, dass das strukturierte Beschreiben von Abläufen der Services, welche die Grundlage für die ASML Modelle bilden, offensichtlich nicht dem Denkmuster vieler Probanden entsprach. Zwar konnten alle Stakeholder dieser Art der Darstellung inhaltlich folgen, hatten aber Schwierigkeiten ihre eigenen Vorstellungen auf diese Weise zu explizieren. Auch hatten einzelne Probanden eine Hemmschwelle etwas falsch machen zu können, der durch die Unterstützung des Standmitarbeiters entgegengewirkt werden konnte. Hinsichtlich der entstandenen Modelle viel auf, das oftmals sehr stark von den formalen Vorgaben der Modellierung, also der Notation und den dazugehörigen Regeln, abgewichen wurde. Desweiteren wurden neue domänenspezifische Konzepte nur grob skizziert und nicht mit allen, für die Umsetzung notwendigen, Informationen spezifiziert.

Der Umsetzungsexperte konnte das entstandene Modell zur Identifikation eines neuen zu realisierenden Bricks heranziehen. Jedoch verfügte das Modell nur über sehr allgemeine Informationen hinsichtlich der detaillierten Ausgestaltung des Stubs. Diese mussten vom Probanden, auf der Grundlage der im Modell befindlichen Kontextinformationen und der technischen Grundlagen für die Realisierung ergänzt werden. Als Ergebnis konnte ein neuer funktionaler Baustein erstellt werden, der anschließend in Fallstudie drei zum Einsatz kommt.

Schließlich wurde das Modell noch genutzt die explizierten Vorstellungen und Visionen des sonstigen Stakeholders an den Anwendungsexperten zu übertragen. Für diesen war nicht das gesamte Modell von Interesse, sonder lediglich ein spezieller, dort modellierter, Teilaspekt, welcher anschließend auf die Anwendungsdomäne des Probanden übertragen wurde. Das allgemeine Problem der Reiseplanung und der damit verbunden Bedarf an einer Reichweitenplanung für mögliche Tankstellen wurde auf die Domäne der Elektromobilität übertragen. Da dieses Problem dort, aufgrund beschränkter Batteriekapazitäten, als besonders Wichtig erkannt wurde, wurde der im Modell des sonstigen Stakeholders skizzierte Automotive Service für diese Domäne im Detail ausgestaltet. Es wurde also nicht die modellierte Version eins zu eins übernommen, sondern diese als Grundlage für eine an die Domäne angepasst Version genutzt.

7.2.1.3 Schlussfolgerungen

Als Ergebnis der Fallstudie lässt sich feststellen, dass die erste Hypothese, dass Stakeholder ohne Kenntnisse des domänenspezifischen Frameworks und der darin umgesetzten Komponenten auf der Basis existierender Konzepte neue Dienste modellieren können, bestätigt werden kann. Allerdings wurde dabei eine Reihe von Besonderheiten festgestellt. So sind Bausteine der Modellierung nur als grobes Konzept und nicht in ihrer konkreten Ausgestaltung eingesetzt worden. Darüber hinaus wurde auch die Bedeutung einzelner Bricks sehr frei interpretiert. Beide Beobachtungen resultierten in einem Modell, das zwar den Zweck der Dokumentation und Kommunikation von Services erfüllt, jedoch für eine Ausführbarkeit durch die Experten der Anwendungsdomäne und Umsetzungsdomäne ergänzt werden muss. Ein weiteres Problem war die notwendige Fähigkeit, Services in Form von logischen Abläufen zu beschreiben. Dieses Vorgehen entsprach nicht allen Probanden, so dass diese auf eine Unterstützung durch den Standmitarbeiter angewiesen waren.

Auch die zweite Hypothese konnte im Rahmen der ersten Fallstudie bestätigt werden. Ein Stakeholder ohne Kenntnisse des domänenspezifischen Frameworks und der darin umgesetzten Komponenten konnte neue domänenspezifische Konzept, wie die Sprachaufzeichnung und die Reiseplanung, spezifizieren und ohne die Hilfe des Umsetzungsexperten einen neuen Dienst mit diesen Konzepten modellieren. Jedoch wurden

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

auch hinsichtlich dieser Hypothese Einschränkungen beobachtet. So wurden alle neu spezifizierten domänenspezifischen Konzepte nur als grober Umriss im Modell beschrieben. Die bereitgestellten Informationen waren zwar ausreichend den Umsetzungsexperten zur Realisierung des entsprechenden Bricks zu befähigen, dieser musste allerdings für die Umsetzung wesentliche Details, wie die genauen Konnektoren, ergänzen.

Zusammenfassend lässt sich feststellen, dass für die Kommunikation der sonstigen Stakeholder mit den Experten der Anwendungsdomäne und der Umsetzungsdomäne die ersten beiden Hypothesen bestätigt werden können. Allerdings wurden auch einige Beobachtungen gemacht die eine Ergänzung und Erweiterung der Modellierungssprache und der dazugehörigen Werkzeuge für diesen Anwendungsfall implizieren. So ist für eine Realisierbarkeit der spezifizierten Automotive Services einerseits eine Überarbeitung der Modelle durch den Anwendungsexperten als auch eine detaillierte Ausgestaltung von Stubs durch den Umsetzungsexperten notwendig.

Diese Beobachtungen richten sich weniger an die Modellierung oder deren Werkzeuge als mehr an das Vorgehen zur Gestaltung neuer Automotive Services. Allerdings müssen sowohl Modell als auch Werkzeug die freie Nutzung von ASML, wie sie an den Flipcharts stattgefunden hat, unterstützen. So muss einerseits ein impliziertes Anlegen von Stubs als auch andererseits ein impliziertes Erweitern von Bricks unterstützt werden. Will ein Nutzer also ein Brick über einen Konnektor verbinden der dort noch nicht existiert, so muss dieser automatisch angelegt und somit implizite bei der Modellierung des neuen Services erstellt werden. Gleiches gilt auch für das Anlegen neuer Stubs. Die Nutzer haben diese nicht erst spezifiziert und dann verwendet, sondern eine Art Template eines leeren Stubs genutzt, welches dann im Verlauf der Modellierung implizit um dessen Eigenschaften erweitert wurde. Dieses „implizite“ Stub kann dann vom Umsetzungsexperten detailliert und umgesetzt werden.

7.2.2 Fallstudie II: My Reichweite App

Nachdem im Rahmen der ersten Fallstudie sowohl bei der Bestätigung der ersten Hypothese als auch bei der der zweiten Hypothese einige Einschränkungen beobachtet werden konnten, ist es das Ziel dieser zweiten Fallstudie diese Beobachtungen näher zu beleuchten. Es stellen sich also wiederrum die Fragen ob Stakeholder, unter Nutzung bestehender domänenspezifischer Konzepte, durch die Modellierung in die Lage versetzt werden eigene Ideen als Automotive Services zu spezifizieren sowie wie ob Stakeholder in der Lage sind neue, noch nicht definierte, domänenspezifische Konzepte zu explizieren, die anschließend auf der Grundlage dieser Explikation implementiert werden können.

Stand in der ersten Fallstudie die Kommunikation und Kooperation der sonstigen Stakeholder mit den Experten der Umsetzungsdomäne und der Anwendungsdomäne im Vordergrund, so wird nun die direkte Zusammenarbeit der beiden Expertengruppen betrachtet. Der Fokus liegt nun also darauf, ob die beiden ersten Hypothesen auch für diese Zusammenarbeit gültig sind und ob wiederum Einschränkungen beobachtet werden können. Die Ergebnisse der ersten Fallstudie legen hier die Vermutung nahe, das in ASML geschulte Experten der Anwendungsdomäne deutlich vollständigere Modelle explizieren können als die sonstigen Stakeholder, deren Stubs dann vom Umsetzungsexperte ohne Überarbeitung umgesetzt und ein funktionierender Dienst an den Anwendungsexperten übergeben werden kann. Abbildung 7-4 stellt dieses Szenario graphisch dar.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

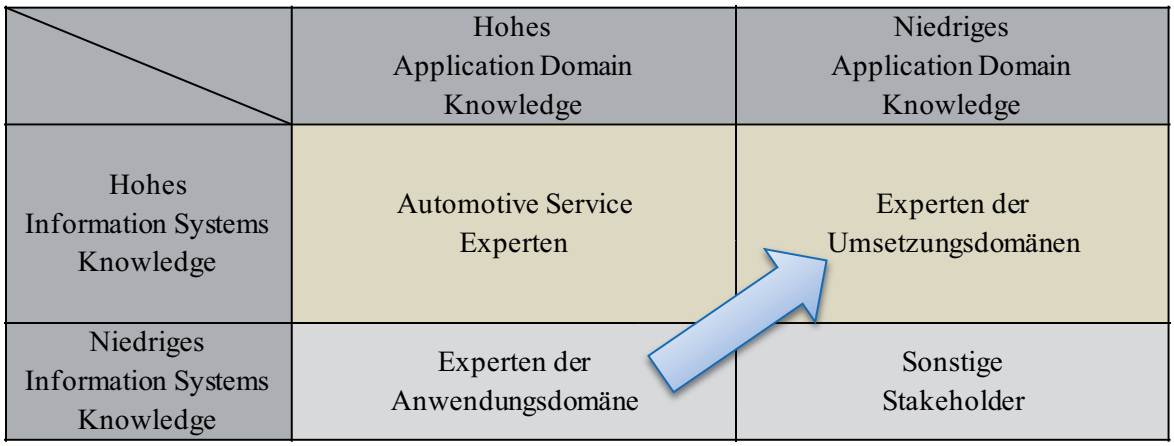


Abbildung 7-4 Beteiligte Stakeholder der zweiten Fallstudie
Quelle: Eigene Darstellung

7.2.2.1 Vorgehen

Die Fallstudie wurde im Nachgang der Langen Nacht der Wissenschaften durchgeführt. Als Grundlage diente der in Abbildung 7-2 dargestellte Reiseplanungsdienst. Dieser sollte vom Experten der Anwendungsdomäne in einen Automotive Service übersetzt werden, der die spezifischen Anforderungen der Anwendungsdomäne Elektromobilität des Experten realisiert. Zusammen mit den im Service entstandenen Stubs wurde dieser schließlich an den Umsetzungsexperten übergeben, der wiederum nach der Realisierung der Stubs in HIMEPP einen funktionierenden Automotive Service übergeben sollte.

7.2.2.1.1.1 Probanden

Wie bereits erwähnt, waren an dieser Fallstudie zwei Probanden beteiligt. Als Vertreter der Gruppe der Anwendungsexperten wurde der Projektbearbeiter des Projekts „Service Innovationsprozess am Beispiel der Elektromobilität“ (eIT) herangezogen. Einerseits durch seine im Rahmen einer Schulung erworbenen Kenntnisse im Einsatz von ASML und der dazugehörenden Modellierungsumgebung sowie andererseits der intensiven Befassung mit der Domäne der Elektromobilität stellt der Bearbeiter einen geeigneten Probanden für diese Fallstudie dar.

Entsprechend war der Projektbearbeiter des Projekts „Ubiquitous Automotive Services Environment“ (uBASE) als Proband der Gruppe der Umsetzungsexperten beteiligt. Er hat sowohl eine Schulung in ASML und der ASML Workbench als auch im domänenspezifischen Framework HIMEPP erhalten. Diese Schulungen versetzten ihn in die Lage, gestaltete Modelle in allen Details verstehen und überarbeiten zu können und darüber hinaus neue Bricks in HIMEPP zu realisieren.

7.2.2.1.2 Durchführung der Fallstudie

Als Aufgabenstellung für die Fallstudie wurde zunächst dem Experten der Anwendungsdomäne das in der ersten Fallstudie entstandenen Automotive Service Modell der Reiseplanung übergeben (vgl. Abbildung 7-2). Ziel sollte es sein, den dort beschrieben Automotive Service unter Berücksichtigung der domänenspezifischen Herausforderungen der Elektromobilität vollständig zu erweitern um schließlich einen, den Eigenarten und Anforderungen der Domäne angepassten, Automotive Service zu erhalten. Bei dieser Aufgabenstellung sollte die Reiseplanung lediglich als Orientierung dienen, der Proband

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

konnte sowohl inhaltlich als hinsichtlich des Modells von der Vorlage entsprechend seinen eigenen Einschätzungen abweichen. Als Werkzeuge standen dabei der in die ASML Workbench integrierte Editor sowie die Workbench selber zur Verfügung.

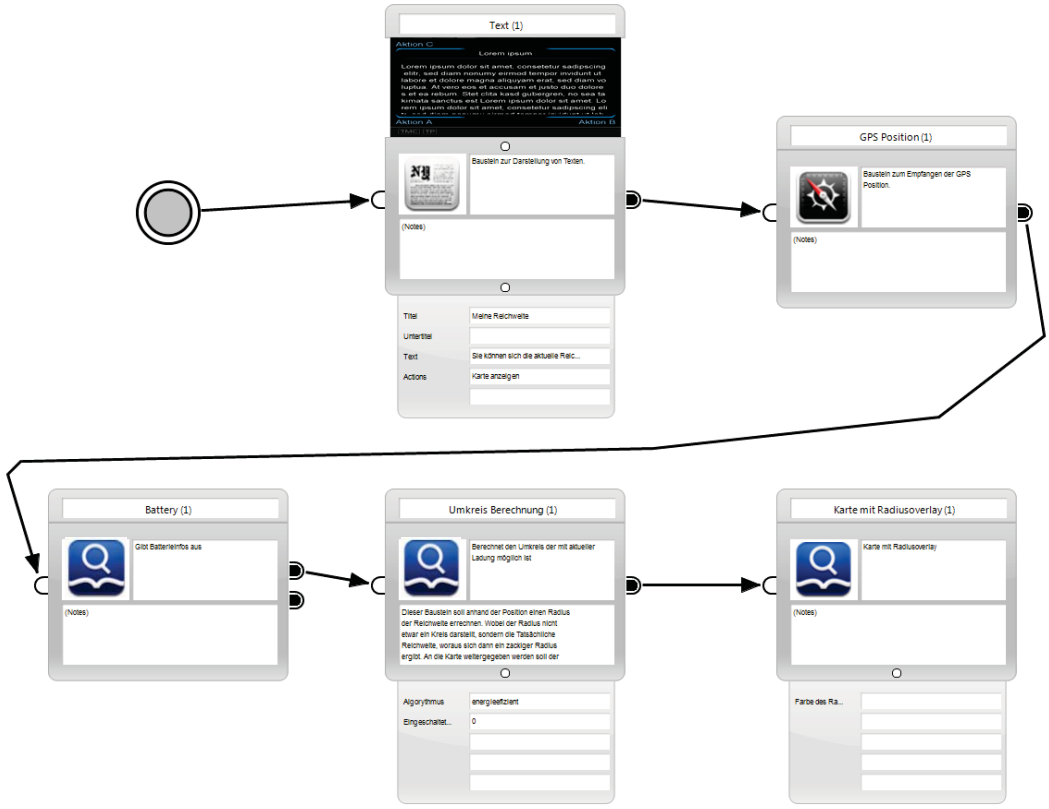


Abbildung 7-5 Modell "my Reichweiten App" des Experten der Anwendungsdomäne
Quelle: Eigene Darstellung

Zunächst wurde vom Probanden ein, aus seiner Sicht besonders relevanter, Bereich der Reiseplanung extrahiert und auf die Domäne der Elektromobilität übertragen. Ziel des neuen Automotive Services sollte es sein, auf der Grundlage der verbleibenden Batteriekapazität sowie der aktuellen Position einen maximalen noch erreichbaren Radius der Reichweite eines Elektrofahrzeugs zu berechnen und diesen auf einer Karte graphisch anzuzeigen. Abbildung 7-5 zeigt das Modell des entstandenen Automotive Services. Ausgehend von einem Begrüßungsbildschirm, auf dem die Funktionsweise des Services beschrieben wird, berechnet der Dienst unter Verwendung der GPS Position des Fahrzeugs und der verbleibenden Batteriekapazität den noch zu erreichenden Umkreis und stellt diesen schließlich auf einer Karte dar. Für das Modell griff der Proband einerseits auf existierende domänenspezifische Konzepte zurück und definierte andererseits neue noch fehlende Stubs. Für die Darstellung des Begrüßungsbildschirms wurde der Text Brick (vgl. Abschnitt 5.1.2.3.1.2) und für die aktuelle Position das GPS Position Brick (vgl. Abschnitt 5.1.2.3.2.3) verwendet. Neu hinzugekommen sind Stubs, die die Konzepte „Battery“, „Umkreis Berechnung“ und „Karte mit Radiusoverlay“ beschreiben.

Dieses Modell, zusammen mit der Definition der drei neuen Stubs, wurde anschließend an den Experten der Umsetzungsdomäne übergeben. Diese Übergabe erfolgte durch Versenden der asm-Datei sowie der drei asml-description.xml-Dateien per E-Mail und Öffnen derselben mit der ASML Workbench auf Seiten des Umsetzungsexperten.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Zunächst wurde das Modell vom Probanden hinsichtlich einer Umsetzbarkeit durch Implementation der fehlenden Stubs im domänenspezifischen Framework hin überprüft. Hier stellte sich heraus, dass zwar alle Elemente des Modells richtig verwendet wurden, die Angaben, die beim Definieren neuer Stubs angegeben waren, jedoch in dieser Form für die Implementation nicht geeignet sind. Einerseits wurden einige Datentypen nicht richtig verwendet und andererseits Funktionsnamen angegeben die im HIMEPP Framework nicht zulässig sind. So wurde beispielsweise der Funktionsname „1“ festgelegt der in HIMEPP und Java nicht zulässig ist. Nach einer Korrektur dieser formalen Fehler konnten die Stubs „Battery“ und „Umkreis Berechnung“ als Bricks implementiert werden.

Hinsichtlich des dritten neuen Stubs, der *„Karte mit Radiusoverlay“*, stellte der Proband fest, dass diese Funktionalität im existierenden Karten Brick bereits vorhanden ist. Aus diesem Grund wurde das Modell dahingehend verändert, das statt dem neuen Stub der bereits existente Brick Karten (vgl. Abschnitt 5.1.2.3.1.4) verwendet wurde. Das vollständige Modell, wie es der Experte der Umsetzungsdomäne schließlich übergeben hat, findet sich in Abbildung 7-6.

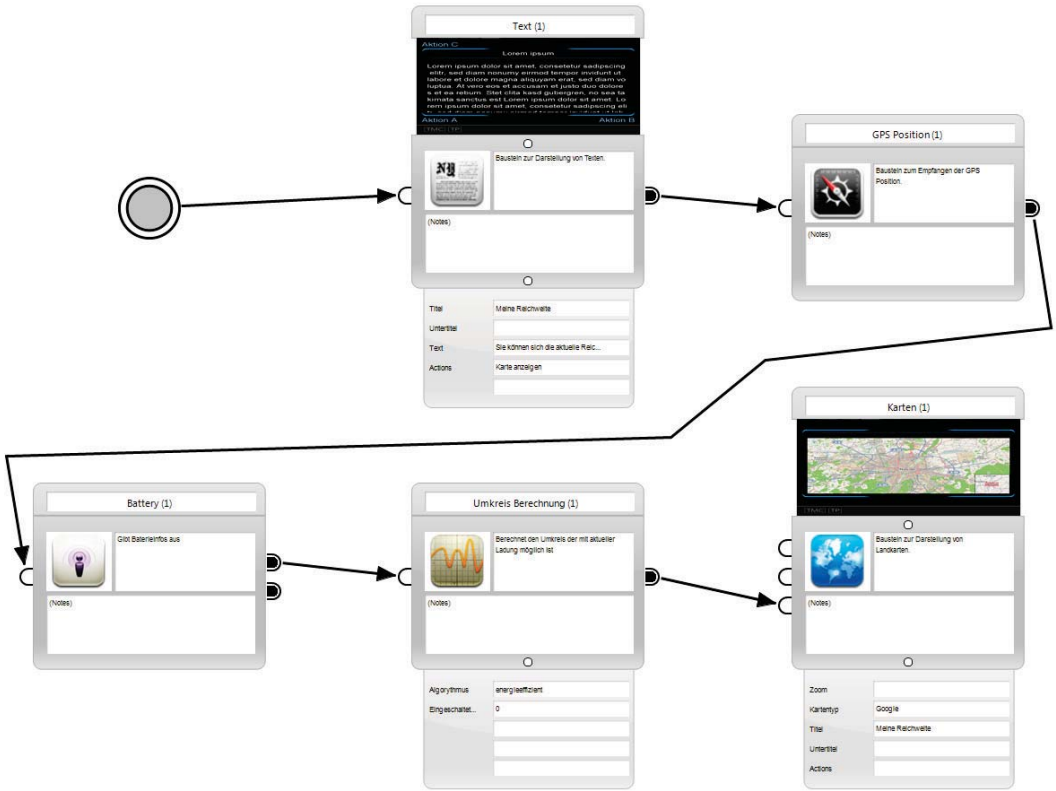


Abbildung 7-6 Modell "my Reichweiten App" des Experten der Umsetzungsdomäne
Quelle: Eigene Darstellung

In einem letzten Schritt wurden nun das überarbeitet Modell sowie die Realisierung der beiden neuen Stubs an eine Instanz der ASML Runtime übergeben und diese gestartet. Wie in Abbildung 7-7 links zu sehen, wird zunächst ein Startbildschirm angezeigt, der die Funktion des Automotive Services beschreibt. Wählt der Nutzer nun hier die Funktion *„Karte anzeigen“* aus, so werden die aktuelle Position sowie der von dieser aus noch zu erreichende Radius auf einer auf das Fahrzeug zentrierten Karte angezeigt.

Anzumerken ist dabei, dass aus Ermanglung eines Elektrofahrzeugs für die Überprüfung des Dienstes im Rahmen der Fallstudie die Angaben der aktuellen Position sowie der Ladezustand der Batterie simuliert wurden. Da die Umsetzung dieser Funktionen jedoch nicht Teil der

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

domänenspezifischen Modellierung ist, stellt dies keine Einschränkung der Ergebnisse der Fallstudie dar.

Abbildung 7-7 Automotive Service "my Reichweiten App"
Quelle: Eigene Darstellung

7.2.2.2 Beobachtungen

Nachdem im Kontext der ersten Fallstudie einige Besonderheiten und Einschränkungen beobachtet werden konnten, wurde auch im Rahmen der zweiten Fallstudie besonderes Augenmerk auf diese gerichtet. Zunächst lässt sich jedoch feststellen, dass ein vom Anwendungsexperten modellierter Dienst durch Realisierung der fehlenden Konzepte im domänenspezifischen Framework zu einem funktionierenden Automotive Service überführt werden konnte.

Aus Sicht des ersten Probanden konnten existierende Konzepte einfach und effektiv wiederverwendet werden, ohne dass sich dieser mit deren Details befassen musste. Auch die Nutzung der ASML Workbench stellte keine Probleme dar, so dass der Dienst nach der ASML Schulung eigenständig erstellt werden konnte. Jedoch wurde der Brick Wizard, welcher zum Erstellen neuer Stubs verwendet wird, vom Probanden als Hürde in der Modellierung wahrgenommen. So dauerte es zu lange bis ein neuer Baustein auf diese Weise erstellt werden kann. Dies unterbrach insbesondere den kreativen Prozess der Gestaltung. Nach Aussage des Probanden war dies zwar für den relativ kleinen Dienst „my Reichweiten App“ keine Problem, stellt aber für komplexere Automotive Services eine wesentliche Hürde dar.

Darüber hinaus werden vom Brick Wizard zu viele, teils erst für die Realisierung im domänenspezifischen Framework relevante, Informationen abgefragt. Diese sind aus Sicht des Experten der Anwendungsdomäne oft unverständlich und überflüssig. In der Fallstudie konnte dies an allen drei neuen Stubs festgestellt werden. So wurden Felder des Brick Wizard, die der Proband für unwichtig hielt oder nicht zuordnen konnte, mit „Dummy“ Informationen versehen; beispielsweise wurde für Funktionsnamen immer eine „1“ eingetragen oder Name und Beschreibung von Konnektoren jeweils mit dem gleichen Text versehen.

Auch ohne diese Detailinformationen konnte der Proband jedoch problemlos seine Vorstellung des Automotive Service als Modell spezifizieren und per E-Mail and den zweiten Probanden übergeben. Diese Übergabe wurde zwar, aufgrund der sehr kleinen Dateien, als problemlos und unkompliziert beschrieben, das Heraussuchen der entsprechenden Datei wurde aber als hinderlich bezeichnet. Der Proband merkte an, das eine Export- und Importfunktion zur Übergabe von Modellen und Stubs eine wesentliche Hilfestellung gewesen wären.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Aus Sicht des Umsetzungsexperten konnten die erhaltenen Modelle und Stubs einfach in die eigene Modellierungsumgebung integriert werden. Auch hier wurde jedoch der Wunsch nach einer einfachen Importfunktion geäußert. Zunächst viel auf, das der Umsetzungsexperte nicht einfach die fehlenden Stubs realisierte, sondern zunächst das Gesamtmodell hinsichtlich der Ausführbarkeit untersuchte. So wurde festgestellt, dass eins der neuen Stubs nicht realisiert werden musste, sondern ein Ersetzen dieses im Modell durch ein bereits existentes Brick die gleiche Aufgabe erfüllt. Neben der eigentlich gestellten Aufgabe, lediglich Stubs zu implementieren, ergänzte der Proband das Modell somit zusätzlich um die eigene Expertise hinsichtlich der Umsetzung von Automotive Services.

Auch bei der Realisierung der Stubs wurden die vom Experten der Anwendungsdomäne übergebenen Informationen überprüft und angepasst. Zwar wurden keine inhaltlichen Änderungen an den Bausteinen vorgenommen, diese jedoch um für die Realisierung wesentliche Details ergänzt. Nach diesen Anpassungen konnten die Stubs schließlich in HIMEPP realisiert werden. Zusammen mit dem geänderten Modell konnten die realisierten Stubs auf einen dritten Rechner übertragen und dort durch die ASML Runtime ausgeführt werden. Abgesehen von den beschriebenen Anpassungen konnte das Modell also durch Implementierung der neuen Stubs als Automotive Service ausgeführt werden.

7.2.2.3 Schlussfolgerungen

Analog den Schlussfolgerungen aus der ersten Fallstudie kann auch hier die erste Hypothese bestätigt werden. Ein Stakeholder ohne Kenntnisse des domänenspezifischen Frameworks und der darin umgesetzten Komponenten konnte auf der Basis existierender Konzepte einen neuen Dienst modellieren. Im Gegensatz zu den sonstigen Stakeholdern, die diese Aufgabe in der ersten Fallstudie übernommen haben, konnte der in ASML geschulte Experte der Anwendungsdomäne einen vollständigen und korrekten Automotive Service ohne fremde Hilfe modellieren. Allerdings wurden nicht alle bestehenden Bricks richtig eingesetzt und so die Verfügbarkeit des benötigten Bedienkonzepts zum Anzeigen eines Overlays auf einer Karte nicht erkannt. Der hier zusätzlich neu definierte Stub wurde erst vom Experten der Umsetzungsdomäne erkannt und wieder aus dem Modell entfernt.

Die zweite Hypothese konnte ebenfalls bestätigt werden. Ein Stakeholder ohne Kenntnisse des domänenspezifischen Frameworks und der darin umgesetzten Komponenten konnte neue domänenspezifische Konzept, wie „Battery“ und „Umkreis Berechnung“ spezifizieren und ohne die Hilfe des Umsetzungsexperten einen neuen Dienst mit diesen Konzepten modellieren. Allerdings wurden auch hier wieder einige Einschränkungen dieser Beobachtung festgestellt. So wurden viele Details eines Stubs, welche aus der Sicht der Experten der Anwendungsdomäne nicht von Wichtigkeit sind, ausgelassen oder nur unzureichend angegeben. Diese mussten vom Umsetzungsexperten ergänzt werden. Darüber hinaus wurde auch der Brick Wizard in der ASML Workbench als zu kompliziert empfunden. Das durch den Wizard erzwungene Unterbrechen der eigentlichen Modellierung wurde als starke Beeinträchtigung der Kreativität eingeschätzt.

Zusammenfassend lässt sich feststellen, dass wiederum sowohl die erste als auch die zweite Hypothese bestätigt werden konnten. Allerdings implizieren die Beobachtungen dieser Fallstudie auch einige notwendige Änderungen am Einsatz der Modellierung und den dazugehörigen Werkzeugen. Zunächst müssen die Aufgaben der beiden Expertengruppen klarer voneinander getrennt werden. Für den Experten der Anwendungsdomäne stehen die fachlichen und inhaltlichen Elemente des Modells im Vordergrund, Details, wie

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

beispielsweise der konkrete Funktionsname eines neuen Stubs, werden von ihm nicht als wichtig betrachtet und dementsprechend auch nicht richtig spezifiziert. Diese Aufgabe kommt dem Umsetzungsexperten zu, der auch die für diese Details notwendige Expertise mitbringt.

Gleiches gilt auch für die funktionale Betrachtung der Modelle. Der erste Proband wollte einen Overlay auf einer Karte anzeigen und hat auf den ersten Blick kein dazu passendes Konzept gefunden. Um mit seiner inhaltlichen Aufgabe fortfahren zu können, wurde ein neuer Stub für dieses Konzept definiert. Als Schlussfolgerung aus dieser Beobachtung sollten Umsetzungsexperten nicht nur die Aufgabe haben, neue Stubs zu implementieren, sondern auch eine funktionale Kontrolle der Modelle durchzuführen. Hierbei muss allerdings sichergestellt werden, dass die Gestaltung des Automotive Services dadurch inhaltlich nicht verändert wird.

Zuletzt muss auch das Erstellen neuer Stubs vereinfacht werden. Aus den Beobachtungen lassen sich hier zwei Schlussfolgerungen ziehen. Ersten darf der kreative Prozess des Modellierens nicht durch einen Wizard unterbrochen werden. Neue Stubs müssen durch Drag'n'Drop analog zu Bricks verwendet werden können und deren Gestaltung implizit im Verlauf des Modellierens erfolgen. Dadurch kann sich der Experte der Anwendungsdomäne auf die inhaltliche Aufgabe konzentrieren und wird nicht durch das Anlegen von Bausteinen unterbrochen. Zusätzlich muss auch sichergestellt werden, dass für die Implementierung notwendige Details erst durch den Umsetzungsexperten angegeben werden. Dies erhöht den Abstraktionsgrad für den Anwendungsexperten und stellt sicher, dass die Detailinformationen eine geeignete Implementation ermöglichen.

7.2.3 Fallstudie III: Evaluation von Automotive Services

Zuletzt steht nach der Überprüfung der ersten beiden Hypothesen in den ersten beiden Fallstudien noch die eingehende Betrachtung der dritten und letzten Hypothese aus. Dieser zufolge sollten Stakeholder in der Lage sein, existierende Elemente wieder zu verwenden um neue Funktionalitäten für künftige Nutzer zu definieren und so die Fähigkeiten der Automotive Services Modeling Language zu erweitern. Expliziertes Wissen, welches bestehende aus realisierten domänenspezifischen Konzepten in Form von Automotive Service Modelliert wurde, müsste somit in zukünftigen Automotive Services, die darüber hinaus auch aus anderen Domäne stammen, wiederverwendet werden können.

Aufgrund der Hypothese, dass sowohl realisierte Konzepte als auch domänenspezifische Automotive Service Modelle wiederverwendet werden können sollen, wurde für diese Fallstudie eine Kommunikation zwischen Automotive Service Experten und Experten der Anwendungsdomäne gewählt. Automotive Service Experten sollen zu diesem Zweck einen bestimmten domänenspezifischen Automotive Service modellieren und die darin verwendeten Konzepte entsprechend in HIMEPP realisieren. Abbildung 7-8 stellt diesen Zusammenhang graphisch dar. Ein Experte einer anderen Anwendungsdomäne sollte nun in der Lage sein diesen Service als Teil seines eigenen Services wiederzuverwenden ohne einzelne Details des übergebenen Modells oder dessen realisierter Stubs zu besitzen. Der übergebene Automotive Service soll also, analog zu einem Brick, als transparenter Baustein eingesetzt werden und somit eine Erweiterung der Möglichkeiten von ASML darstellen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

	Hohes Application Domain Knowledge	Niedriges Application Domain Knowledge
Hohes Information Systems Knowledge	Automotive Service Experten	Experten der Umsetzungsdomänen
Niedriges Information Systems Knowledge	Experten der Anwendungsdomäne	Sonstige Stakeholder

Abbildung 7-8 Beteiligte Stakeholder der dritten Fallstudie
Quelle: Eigene Darstellung

7.2.3.1 Vorgehen

Die dritten Fallstudie wurde im Rahmen des Masterpraktikum „Automotive Services“ im Sommersemester 2010 an der Technischen Universität München durchgeführt. Die Aufgabe der Praktikumsteilnehmer war es dabei eine Möglichkeit zu schaffen, beliebige Automotive Services direkt im Fahrzeug zu evaluieren. Zu diesem Zweck sollten Entwickler von zukünftigen Automotive Services mit einem Baustein für die Modellierung ausgestattet werden, mit dessen Hilfe sie Fragebögen an beliebigen Stellen ihrer eigenen Dienste integrieren können. Darüber hinaus sollte das Wissen über die Gestaltung der Fragebögen und deren Ausführung im Fahrzeug für die Modellierer transparent erfolgen, so dass diese kein explizites Wissen hinsichtlich der Evaluation benötigen.

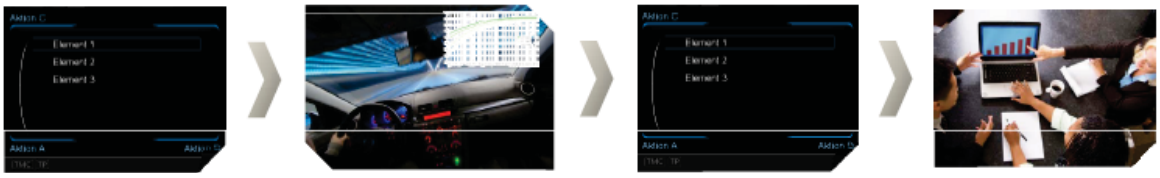


Abbildung 7-9 Vorgehen Evaluation von Automotive Services
Quelle: Eigene Darstellung

Abbildung 7-9 stellt die Grundidee der Aufgabe dar. Zunächst soll ein Fragebogen zu Beginn der Nutzung eines Services direkt im Fahrzeug möglich sein, den die Nutzer durch Spracheingabe und Drehen des Dreh-Drück-Stellers ausfüllen können. Anschließend soll ein beliebiger Automotive Service verwendet werden können, an dessen Ende wiederum ein zweiter Fragebogen die Einschätzung der Nutzer im Nachgang erfasst. Die Ergebnisse beider Fragebögen sollen anschließend ausgewertet werden können. Neben diesem einfachen Szenario sollte es darüber hinaus Möglich sein, Fragebögen auch an beliebigen Stellen im Verlauf des zu evaluierenden Services zu integrieren, beispielsweise nach der Nutzung einer bestimmten Teilfunktion.

7.2.3.1.1.1 Probanden

Wie bereits erwähnt sind für diese Fallstudie Experten einer Anwendungsdomäne sowie Automotive Service Experten notwendig. Da die Hauptaufgabe dieser Fallstudie bei den

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Automotive Service Experten liegt, wurde der Einfachheit halber wieder der Projektbearbeiter des Projekts „Service Innovationsprozess am Beispiel der Elektromobilität“ (eIT) als Proband herangezogen. Diesem kommt die Aufgabe zu, eine Evaluation in einen von ihm gestalteten Automotive Service zu integrieren. Wie in den letzten beiden Fallstudien bereits erwähnt, erhielt der Proband eine Schulung in ASML und der ASML Workbench.

Als Automotive Service Experten wurde im Rahmen dieser Fallstudie eine Gruppe an Probanden eingesetzt. Diese besteht aus den vier Teilnehmern des Automotive Services Praktikums sowie einer wissenschaftlichen Hilfskraft, die zum Zeitpunkt der Fallstudie ihre Masterarbeit über die Evaluation von Automotive Services schreibt und das Praktikum auf der fachlichen Seite unterstützt. Die Teilnehmer des Praktikums setzten sich aus zwei Studenten der Informatik und zwei Studenten der Wirtschaftsinformatik zusammen. Alle fünf Probanden erhielten eine Schulung im domänenspezifischen Framework HIMEPP sowie in der Automotive Services Modelling Language und der dazugehörigen Werkzeuge. Durch diese Zusammensetzung konnte für die Gruppe sichergestellt werden, das zum einer die notwendigen Kenntnisse zur Gestaltung und Umsetzung der Evaluation im Modell und Framework und andererseits auch die fachlichen Aspekte der Evaluation vorhanden sind.

7.2.3.1.2 Durchführung der Fallstudie

Zunächst wurde von den Automotive Experten ein Vorgehen entwickelt, wie beliebige Fragebögen erstellt, verwendet und an den Automotive Service übergeben werden können. Zu diesem Zweck wurde die Nutzung von LimeSurvey als Grundlage für die Fragebögen festgelegt. LimeSurvey ist ein OpenSource Werkzeug „zur Datenerhebung mittels Onlineumfragen“, welches die Möglichkeit eines Exports dort angelegter Fragebögen in Form von XML Dokumenten unterstützt (Schmitz 2010). Zukünftige Nutzer der Evaluationsbausteine können somit ihre Fragebögen mit der graphischen Benutzeroberfläche von LimeSurvey erstellen und diese als XML Dateien in ihren Automotive Service integrieren.

Da nun das Erstellen und Verwalten der notwendigen Fragebögen festgelegt worden ist, wurde mit der Modellierung des eigentlichen Fragebogenmodells begonnen. Aus der Sicht von ASML ist ein Fragebogen ein Automotive Service, der durch die ASML Runtime im Fahrzeug ausgeführt werden kann. In Abbildung 7-10 ist das entstandene Modell abgebildet. Ausgehend von einem Text Brick, welches den Nutzer über den Inhalt und Zweck des Fragebogens informiert, wird der QuAudiManager aufgerufen. Dieser Baustein hat die Aufgabe, die in LimeSurvey entstandenen Fragebögen zu interpretieren und den Nutzer durch den Fragebogen zu leiten. Zu diesem Zweck verfügt der Baustein über je einen Ausgangskonnektor für jeden möglichen Fragentyp.

Beim erstmaligen Aufrufen des Bausteins über seinen Eingangskonnektors wird der in den Properties angegebene Fragebogen geladen und die erste Frage aufgerufen. Abhängig von deren Typ wird der entsprechende Ausgang signalisiert und der darauffolgende Baustein angezeigt. Hier stehen entweder Fragen mit einer bestimmten Anzahl an vorgegebene Antwortmöglichkeiten oder offene Fragen zur Verfügung. Erstere werden analog einer Liste im QuAudiQuestionList Brick angezeigt und der Nutzer kann seine Antwort über den Dreh-Drück-Steller auswählen. Letztere werden durch die Anzeige des QuAudiQuestionRecord Brick realisiert. Dieser gibt dem Nutzer die Möglichkeit, seine Antwort als Sprachmemo einzugeben. Für die eigentliche Realisierung der Memo Funktionalität wird der in Fallstudie eins entstanden Audio Recorder Brick verwendet.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

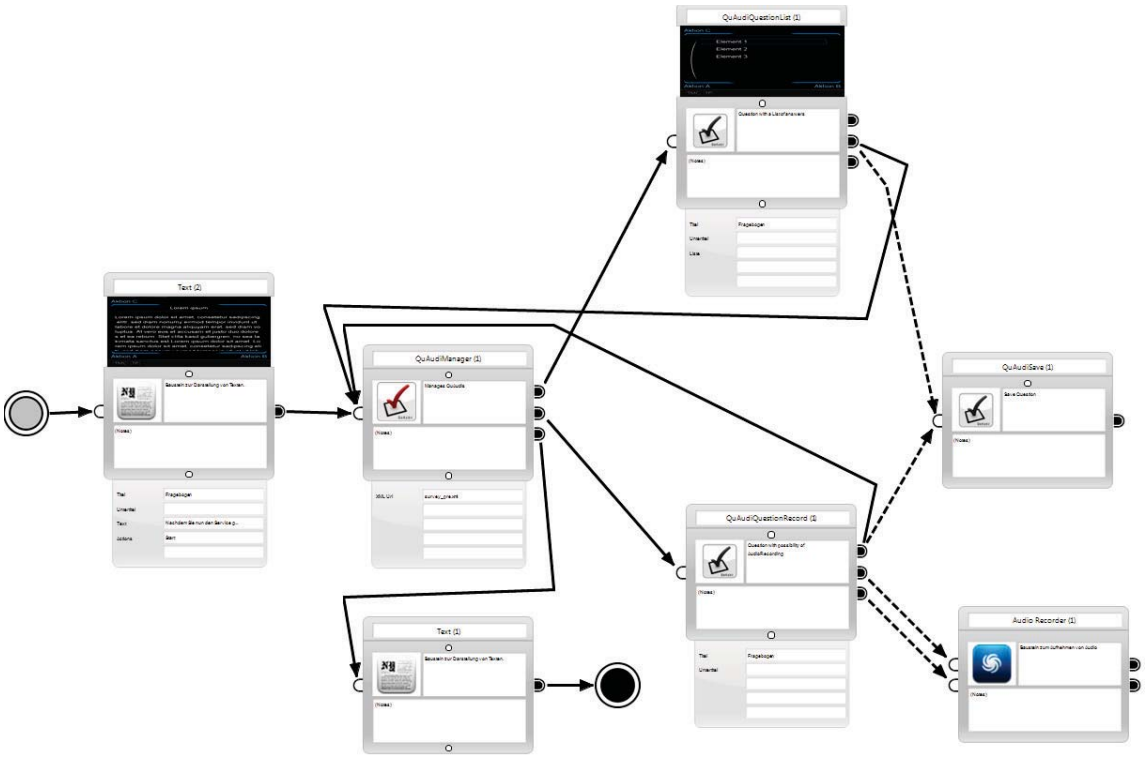


Abbildung 7-10 Modell des Fragebogen Automotive Service
Quelle: Eigene Darstellung

Nachdem ein Nutzer eine Frage beantwortet hat, wird dies wiederum an einem entsprechenden Ausgangskonnektor des jeweiligen Bausteins signalisiert. Nun wird einerseits durch einen asynchronen Stream der QuAudiSave Brick aufgerufen, der die Ergebnisse der Frage direkt in einer Datei zur späteren Auswertung ablegt. Zusätzlich wird über einen synchronen Stream der Kontrollfluss wieder an den QuAudiManager zurückgegeben. Dieser signalisiert nun entweder die nächste Frage oder ruft, sollte das Ende des Fragebogens erreicht worden sein, einen zweiten Text Brick auf. Hier bekommt der Nutzer das Ende des Fragebogens angezeigt und der Fragebogen wird beendet.

Neben der Gestaltung dieses Modells für einen Fragebogen Automotive Service wurden von den Probanden auch die vier neuen Stubs im domänenspezifischen Framework realisiert. Analog der Übergabe der „my Reichweite App“ in Fallstudie zwei wurden im Anschluss das asm-Modell sowie die Implementation der Stubs im domänenspezifischen Framework an den Experten der Anwendungsdomäne übergeben.

Dieser konnte das „Questionaire“ genannte Modell als Composite zur Evaluation eigener Automotive Services nutzen. Hierzu wurde das in Abbildung 7-11 zu sehende neue Automotive Service Modell erstellt, welches aus der Kombination von drei Composites besteht. Nach dem Start Brick wird direkt ein Fragebogen aufgerufen. Als Eingangskonnektor wurde hier der Eingangskonnektor des ersten Text Bricks in Abbildung 7-10 und als Ausgangskonnektor der des zweiten Text Bricks verwendet. Dieser wurde anschließend mit der „my Reichweite App“ verbunden, die in dieses Modell wiederum als Composite eingefügt wurde. Schließlich wird nochmal ein Fragebogen als weiteres Composite aufgerufen, bevor der Service beendet wird.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Abbildung 7-11 Integration Fragebögen in Automotive Service
Quelle: Eigene Darstellung

Der so entstandene neue Automotive Service stellt funktional ebenfalls die „my Reichweiten App“ dar, wurde jedoch mit einem Vor- und einem Nachfragebogen versehen. Aus Sicht des Nutzers wird als zunächst ein Fragebogen durchgeführt, an dessen Ende der eigentliche Service genutzt werden kann. Sobald dieser wiederum beendet wird, wird erneut ein Fragebogen aufgerufen, der eine Einschätzung des Nutzers nach der Nutzung des Services erhebt. Für diese Integration der Funktionalität des Fragebogens in einen Service aus der Domäne der Elektromobilität muss der Proband weder den Service noch den als Composite verfügbaren Fragebogen anpassen. Beide konnten einfach integriert und die von den Automotive Experten zur Verfügung gestellt Funktionalität wiederverwendet werden. Abbildung 7-12 zeigt Ausschnitte aus der Ausführung des ersten Fragebogens.

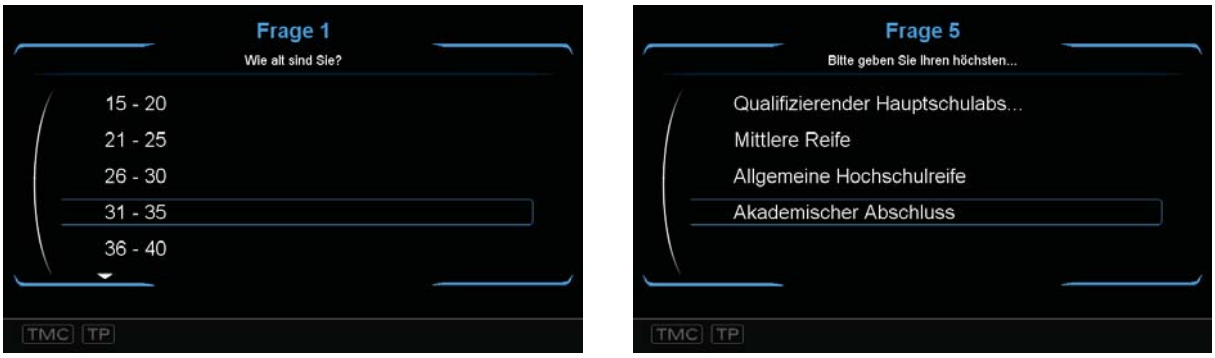


Abbildung 7-12 Ausführung eines Fragebogens im Fahrzeug
Quelle: Eigene Darstellung

7.2.3.2 Beobachtung

Bei der Durchführung der Fallstudie viel auf, das die Probanden die Möglichkeiten von ASML und HIMEPP geschickt um Funktionen externer Software ergänzt haben. So wurde das eigentliche Erstellen und Verwaltet der Fragebögen komplett aus der automotive Umgebung herausgelöst und auf ein dafür spezialisiertes Werkzeug übertragen. Auf diese Weise musste ASML nur um eine Interpretation und Nutzung der dort erstellten Informationen erweitert werden. Betrachtet man das Modell genauer, stellt man fest, dass lediglich einer der neu entstandenen Bausteine spezifische Kenntnisse über den eigentlichen Fragebogen benötigt, die restlichen drei entstandenen Bausteine auch für andere Fragebogenanbieter und Formate nutzbar sind. Allerdings wurden die neu erstellten Bausteine durch ihre Namensgebung und Realisierung eng an die Evaluationsdomäne gebunden, was eine Nutzung in anderen Kontexten erschwert. Analog zu den eingesetzten Bricks Text und Audio Recorder wäre auch eine unabhängigere Realisierung denkbar gewesen.

Hinsichtlich der ASML Workbench konnten keine besonderen Probleme oder Auffälligkeiten festgestellt werden. Zwar gab es einige konstruktive Verbesserungsvorschläge, wie beispielsweise das Vergrößern des Fensters zum Verknüpfen von Streams sowie das Verstauschen der in diesem Fenster befindlichen Inhalte, eine Umsetzung der

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Aufgabenstellung war durch die Probanden problemlos möglich. Hinsichtlich des, in der letzten Fallstudie angesprochenen, Brick Wizard gaben die Probanden an, das dieser in seiner jetzigen Form eine hilfreiche Unterstützung für das Anlegen neuer Stubs darstellt. Insgesamt wurden vier neue Stubs definiert, die schließlich auch in HIMEPP realisiert wurden.

Nach der Übergabe des neuen Modells an den Experten der Anwendungsdomäne konnte dieser dieses Modell als Composite zur Realisierung des in Abbildung 7-9 dargestellten Vorgehens zur Evaluation von Automotive Services einsetzen. Durch einfaches Kopieren des Modells und der realisierten Bricks in die eigene Umgebung, konnten das Composite wie ein gewöhnlicher Brick verwendet werden. Allerdings gab der Proband hier wiederum an, eine Importfunktion in der Workbench zu vermissen. Als zu evaluierender Service wurde der in Fallstudie zwei entwickelte Dienst „my Reichweiten App“ genutzt. Dieser wurde zu diesem Zweck selbst, zusammen mit dem Fragebogen, als Composite in ein neues Evaluationsmodell eingesetzt.

Aufgrund der Tatsache, dass Composites keine Properties der in ihnen enthaltenen Bausteine verändern können sondern diese steht im eigentlichen Modell angegeben werden, verfügen alle Instanzen eines Composites innerhalb eines Modells die gleichen Werte. Für das realisierte Szenario, in dem das Questionaire Composite mit zwei unterschiedlichen Fragebögen integriert wurde, musste der Proband dieses unter einem neuen Namen ein zweites Mal speichern, um so die unterschiedlichen Properties (z.B. Angabe des konkreten Fragebogenmodells) realisieren zu können. Dies wurde als starke Einschränkung der Nutzung von Composites bewertet.

7.2.3.3 Schlussfolgerungen

Nach der Durchführung der dritten Fallstudie lässt sich feststellen, dass auch die dritte Hypothese bestätigt werden kann. Stakeholder sind in der Lage, existierende Elemente wieder zu verwenden um neue Funktionalitäten für künftige Nutzer zu definieren und so die Fähigkeiten der Automotive Services Modeling Language zu erweitern. So wurden die existenten Bricks Text und Audio Recorder mit vier neuen Bricks ergänzt und so ein Fragebogenmodell erstellt. Dieses kann wiederum als Composite in neue Modelle integriert werden. Auf diese Weise konnte also ein neuer, komplexer Baustein zur Wiederverwendung geschaffen werden, um dessen Funktionalität ASML erweitert wurde. Darüber hinaus kann der Baustein in völlig anderen Domänen, wie der der Elektromobilität, eingesetzt werden.

Allerdings wurden auch zwei wesentliche Einschränkungen beobachtet. So wurden zentrale Aspekte des Fragebogens, nämlich das Erstellen und Verwalten desselben, auf eine externe Softwarekomponente übertragen. Dies hat zwar auf der einen Seite den Vorteil, dass der entstandenen Automotive Service, bzw. das Composite, auf seine spezielle Aufgabe, das Anzeigen und Durchführen von Fragebögen im Fahrzeug, optimiert werden konnte, andererseits muss für dessen Nutzung jedoch ein zusätzliches Werkzeug herangezogen werden. Dies hat zur Folge, dass der Einsatz des Composites von den richtigen Angaben in den Properties abhängig ist. Nur wenn dort der richtige Fragebogen angegeben wird kann dieser auch durch den Service eingesetzt werden. Dies führt jedoch zur zweiten Einschränkung. Properties können immer nur direkt im Modell und nicht am Composite angegeben werden. Wird nun ein Composite zweimal, mit unterschiedlichen Properties, im gleichen Modell verwendet, so muss das das Composite beschreibende Modell zunächst unter einem neuen Namen gespeichert werden, um dort die entsprechenden unterschiedlichen Angaben zu machen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Es lässt sich also feststellen, dass nach der Durchführung der dritten Fallstudie auch die letzte Hypothese bestätigt werden konnte. Allerdings zeigen die Beobachtungen auch Potential für Verbesserungen auf. So fehlt bislang eine Möglichkeit, Properties von Bausteinen innerhalb eines Composites durch Angaben im Composite zu überschreiben. Dies hat sich aber als eine wesentliche Anforderungen an einen praktischen Einsatz von Composites herausgestellt. Darüber hinaus muss die Übergabe von Composites und den dazugehörigen Realisierungen im domänenspezifischen Framework erleichtert werden. Analog der Übergabe von einzelnen Bricks und Automotive Services müssen auch Composites einfach aus der lokalen Modellierungsumgebung extrahiert und in eine weitere integriert werden können. Dies geschieht bislang ausschließlich von Hand, so dass nur erfahrenen Nutzer dazu ohne Hilfe in der Lage sind.

7.3 Schlussfolgerungen der Evaluation

Ziel der Evaluation ist es, die der Modellierungssprache zugrundeliegenden drei Designprinzipien der Bricks, Stubs und Composites hinsichtlich ihrer Eignung zur Begegnung der Herausforderungen an eine domänenspezifische Modellierungssprache, wie sie in Abschnitt 5.1.1 aus dem Vorgehen des Design Thinking abgeleitet wurden, zu überprüfen. Wenn also die postulierten Wirkungen beobachtet werden können, können auch die aus den Designprinzipien folgenden Hypothesen bestätigt werden. Zu diesem Zweck wurden insgesamt drei Fallstudien durchgeführt, die eben diese Hypothesen im Kontext unterschiedlicher Stakeholdergruppen adressieren.

So wurden in der ersten und in der zweiten Fallstudie die ersten beiden Hypothese untersucht. Zunächst wurde der Fokus auf das Gestalten neuer Serviceideen durch die Gruppe der sonstigen Stakeholder gerichtet und untersucht ob durch den Einsatz existierender Konzepte gepaart mit der Gestaltung neuer Konzepte Automotive Services erstellt und schließlich an die Experten der Anwendungsdomäne und Umsetzungsdomäne übergeben werden können. Auch wenn beide Hypothesen grundsätzlich bestätigt wurden, konnten einige wesentliche Beobachtungen gemacht werden. So müssen gerade sonstige Stakeholder, die weder eine speziellen Domäne vertreten noch in der Anwendung der Modellierungstechnik geschult worden sind bei der Gestaltung neuer Services unterstützt werden. Dies beinhaltet sowohl formale Aspekte der Modellierung als auch das Entgegenwirken der beobachteten Hemmschwelle Fehler zu machen. Es hat sich hier insbesondere gezeigt, dass Modelle von dieser Gruppe ein sehr frühes Stadium der Gestaltung darstellen und daher auch ein sehr großes Maß an Unschärfe beinhalten. Für die Modellierungstechnik bedeutet dies, dass Unschärfe zukünftig grundsätzlich in die Konzeption der Modellierung integriert werden muss. Hierzu zählen insbesondere zwei Aspekte.

Zu einem muss das Erstellen von Stubs in den kreativen Prozess der Modellierung integriert werden. Momentan werden neue domänenspezifische Konzepte über den Brick Wizard in ihrer Gänze modelliert und erst anschließend verwendet. Der Einsatz des Flipcharts hat jedoch gezeigt, dass hier ein impliziertes Vorgehen besser geeignet ist. Analog der Nutzung von Bricks sollten neue Stubs ebenfalls einfach via Drag'n'Drop erstellt und im Verlauf der Modellierung implizit definiert werden. Zum anderen müssen Bricks in der evaluierten Fassung der Modellierung an definierten Konnektoren verknüpft werden. Dieses Wissen steht jedoch in dieser Phase in der Regel nicht im Vordergrund, so dass ein „unscharfes" Verknüpfen von Bricks als Hilfreich empfunden wurde.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Mit der dritten Fallstudie wurde auch die dritte Hypothese, das Erfahrungsmanagement, betrachtet. Durch das verfügbar machen von definierten und erprobten Services für die Integration in neue Modelle konnte auch die Umsetzung dieser Herausforderungen durch Composites bestätigt werden. Hier vielen in der Evaluation eher technische und praktische Probleme auf. So können in der aktuellen Realisierung der Modellierung Properties von Composites nur im darunterliegenden Modell und nicht am Composite Baustein angegeben werden. Dies erschwert den multiplen Einsatz eines Composites mit unterschiedlichen Parametern, wie den Fragebogen Composite mit unterschiedlichen Fragebögen aus der dritten Fallstudie. Desweiteren wurde für die Übergabe erstellter Informationen eine mangelende Unterstützung durch die Werkzeuge beobachtet. Zwar hat ein Übertragen von Modellen, Composites und neuer Stubs in allen Fallstudien problemlos funktioniert, jedoch gaben alle Probanden an, das dieser Prozess sehr technisch und teils Aufwendig ist. Eine explizite Import- und Exportfunktion ist hier also dringend ratsam.

Insgesamt lässt sich also feststellen, dass die Anforderungen und Herausforderungen einer Modellierung von Automotive Services mit der Automotive Services Modelung Language und den dazugehörigen Werkzeugen umgesetzt werden konnten. Zwar haben sich eine Reihe an Potentialen und Chancen für eine Weiterentwicklung des Konzepts gezeigt, die zentrale Aufgabenstellung wurde jedoch erfüllt. Eine große Schwachstelle zeigt sich allerdings im begleitenden Prozess. Stakeholder aus den unterschiedlichen Gruppen nutzen die Modellierung auf unterschiedliche Art und Weise.

Betrachtet man die drei Fallstudien als Ganzes, so wurden initiale Ideen von den sonstigen Stakeholdern in einem groben, sehr unscharfen Modell definiert, welches an vielen Stellen von den formalen Anforderungen der Modellierung abweicht. Dieses Modell wurde schließlich vom Experten der Anwendungsdomäne in ein inhaltlich formales, den Anforderungen der Domäne angepasstes, Modell weiterentwickelt. Das Modell wurde also um die Expertise des Anwendungsexperten erweitert. In einem dritten Schritt konnte der Umsetzungsexperte das Modell wiederum um technische Aspekte ergänzen, so dass schließlich ein funktionierender Service zur Verfügung stand. Diese Beobachtung legt die Schlussfolgerung nahe, dass das Modellierungskonzept stark von einer dezidierteren Anpassung an diese drei Aufgaben profitieren kann.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

8 Fazit und Ausblick

Die vorliegende Arbeit stellt einen Ansatz zur interaktiven Gestaltung von Automotive Services vor. Zu diesem Zweck wird die domänenspezifische Modellierungssprache Automotive Services Modeling Language (ASML) entwickelt und gemäß der Referenzarchitektur von Tolvanen und Kelly (2004) mit einer durchgängigen Werkzeugunterstützung, bestehend aus einer Workbench, einem Editor sowie einer Laufzeitumgebung, für den praktischen Einsatz vorbereitet. Als Grundlage für die realisierte Laufzeitumgebung kommt das HIMEPP Framework (Hoffmann 2010) zum Einsatz. Schließlich wird die entwickelte Lösung im Rahmen dreier praktischer Case Studies evaluiert.

8.1 Zusammenfassung der Ergebnisse

Als Grundlage für die Gestaltung der entstandenen Lösung wird zunächst der aktuelle Stand der Technik im Bereich der Automotive Services betrachtet. Hier zeigt sich, dass Automotive Services eine logische Weiterentwicklung von Unterhaltungssystemen im Fahrzeug darstellen. Ausgehenden von einfachen Autoradios haben sich Infotainmentsysteme und Fahrerassistenzsysteme entwickelt. Während Erstere vorwiegend nicht fahrzeugbezogene Aufgabe, wie z.B. Unterhaltung, bedienen, steht bei Letzteren die Unterstützung des Fahrers bei der Steuerung und Bedienung des Fahrzeugs im Vordergrund. Zusätzlich lässt sich auch eine zunehmende Vernetzung dieser Systeme zwischen Fahrzeugen und mit speziellen Backends beobachten. Mobile Mehrwertdienste im Automobil, die über das reine Bedienen des Fahrzeugs hinaus auf die Befriedigung verschiedenster Kundenbedürfnisse abzielen, werden in diesem Sinne als Automotive Services bezeichnet und sind somit eng mit Infotainmentsysteme verknüpft.

Allerdings zeigen sich eine ganze Reihe an Innovationstreibern für Automotive Services die aus den unterschiedlichsten Branchen und Domänen entstammen. So führte die oft unter dem Begriff Web 2.0 zusammengefasste Weiterentwicklung des Internets zu verstärkter Beteiligung der Nutzer an den dort bereitgestellten Inhalten. Automotive Services müssen also nicht nur eine Bereitstellung von Informationen sondern verstärkt eine Interaktion mit den Nutzern realisieren. Eine analoge Beobachtung macht man auch, wenn man den Einfluss von Smartphones in Betracht zieht. Durch deren zunehmende Bedeutung und Verbreitung, gepaart mit einer flächendeckenden mobilen Breitbandabdeckung, hat sich die Erwartungshaltung der Nutzer an Automotive Services maßgeblich verändert. Zuletzt kommen aber auch aus der Automobilbranche selber wesentliche Innovationstreiber. Ein prominentes Beispiel ist hier die Elektromobilität. Betrachtet man mit Hinblick auf diese Erkenntnisse die aktuellen Entwicklungen der großen Automobilhersteller und Zulieferer, so werden die Themen Information, Navigation, Sicherheit, Integration externer Geräte, Integration Dritter, Vernetzung, sowie Kundenbindung als wesentliche Trends erkannt.

Hinsichtlich der Herausforderungen bei der Gestaltung und Umsetzung geeigneter Automotive Services zeigen sich eine ganze Reihe an wesentlichen Faktoren. Zunächst müssen geeignete Stakeholder identifiziert und in den Prozess der Gestaltung integriert werden. Ziel muss es hier sein, eine einfache Kommunikation über Ziele und Inhalte sowie eine geeignete Kooperation unterschiedlichster Stakeholder sicherzustellen. Grundsätzlich lassen sich zwei wesentliche Merkmale zur Klassifikation von Stakeholdern ableiten. Auf der einen Seite steht das technologische Wissen hinsichtlich der Umsetzung von Automotive

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Services und auf der anderen Seite das fachliche Wissen aus und über die Domäne, welche durch einen speziellen Service unterstützt werden soll. Aus diesen Faktoren ergibt sich eine Matrix aus vier Stakeholdergruppen, welche jeweils durch geeignete Werkzeuge und Konzepte im eigenen Beitrag sowie der Kommunikation untereinander unterstützt werden müssen: Automotive Service Experten, Experten der Anwendungsdomäne, Experten der Umsetzungsdomänen und sonstige Stakeholder.

Eine weitere Herausforderung, neben der Identifikation und Unterstützung der Stakeholder, entstammt dem Requirements Engineering. Da Automotive Services im Wesentlichen Softwaresysteme sind, können hier die Erkenntnisse, Problemstellungen und Lösungsansätze aus dieser Disziplin übertragen werden. Zunächst lässt sich feststellen, dass Stakeholder oftmals Probleme bei der Beschreibung ihrer Zielvorstellungen haben und ihnen ein eingehendes Verständnis der vorliegenden Problemstellung fehlt. Ursachen hierfür sind meist die hohe Komplexität der Problemstellung sowie der Eigenschaften der Lösungsdomäne. Zusätzlich kommen auch sprachliche Barrieren, die aus den unterschiedlichen fachlichen und kulturellen Hintergründen der Stakeholder resultieren, zum Tragen. Dies führt wiederum dazu, dass Anforderungen nicht in geeigneter Art und Weise expliziert und verstanden werden und somit die Qualität der Anforderungen und damit auch der entstehenden Lösungen darunter leidet. Darüber hinaus sind Anforderungen über den kompletten Entwicklungsprozess hinweg Veränderungen unterworfen und müssen somit stetig weiterverfolgt und nachgebessert werden. Als Lösungsansatz für dieses Problem zeichnen sich einerseits ein iteratives Vorgehen sowie andererseits die Bereitstellung einer gemeinsamen und einfachen Beschreibungsmethodik der zu gestaltenden Lösung ab. Gängige Ansätze sind hier der Einsatz von Modellierung sowie die frühe Bereitstellung von Prototypen.

Die nächste Herausforderung resultiert aus der zentralen Bedeutung der Mensch-Maschine-Interaktion. Aufgrund des speziellen Nutzungskontextes im Fahrzeug zeichnen sich hier zwei wesentliche Problemstellungen ab. Zunächst werden die Systeme zunehmend komplexer und erfordern daher eine verstärkte Integration unterschiedlichster Stakeholder aus verschiedenen Fachbereichen. Darüber hinaus kommt dem Faktor Mensch bei der Gestaltung der Interaktion mit Automotive Services eine herausragende Bedeutung zu. Die kognitiven Eigenarten des Menschen müssen bei der Gestaltung von Automotive Services daher Berücksichtigung finden. Analog den Ergebnissen des Requirements Engineering zeigt sich auch für die Gestaltung der Mensch-Maschine-Interaktion ein iteratives Vorgehen als geeignet, Stakeholder aus verschiedenen Fachbereichen zu integrieren. Hierfür ist jedoch eine gemeinsame Repräsentation der angestrebten Lösung als Kommunikations- und Kooperationsgrundlage notwendig, die darüber hinaus auch Wissen über die zu beachtenden Eigenarten des Faktors Mensch umfasst. Da dieses Wissen nicht für jeden Services individuell identifiziert werden muss, können hier existierende Erkenntnisse und Regeln bereits in der Repräsentation hinterlegt werden.

Schließlich basieren Automotive Services auf bestehenden Infotainmentsystemen. Diese bilden somit sowohl die konzeptionelle als auch die technologische Grundlage für eine Integration neuer Services ins Fahrzeug. Zu diesem Zweck werden technische Elemente, wie z.B. ein Bedienteil, ein Display in den Armaturen sowie der Zugang zu den elektronischen Systemen und Komponenten des Fahrzeugs bereitgestellt. Darüber hinaus können aus diesen Systemen auch erprobte Erkenntnisse hinsichtlich eine inhaltlichen Gestaltung und Interaktion mit den Nutzern gewonnen werden. So verfügen die betrachteten Systeme von Audi, BMW

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

und Mercedes-Benz jeweils lediglich über eine Handvoll an Bedienkonzepte für die Darstellung ihrer Funktionalität. Dieses Wissen sowie die Möglichkeiten und Limitationen der Systeme müssen Stakeholdern bei der Gestaltung neuer Automotive Services verfügbar gemacht werden.

Auf der Grundlage dieser Analysen zeigt sich der Einsatz einer domänenspezifischen Modellierung gepaart mit einem partizipativen und iterativen Vorgehen im Rahmen des Design Thinking als geeigneter Lösungsansatz. Das von Terry Winograd (1996) vorgeschlagene iterative Vorgehen fokussiert dabei auf die kreative Gestaltung von Innovationen. Hierfür werden heterogene Gruppen von Akteuren im Rahmen eines sieben Phasen umfassenden, iterativen Vorgehens durch die Verknüpfung einer bottom-up Experimentation mit einer begleitenden Anleitung von oben bei der Gestaltung von Innovationen geleitet. Hinsichtlich einer unterstützenden Beschreibungsmethodik für Automotive Service lassen sich aus den Phasen des Design Thinking sieben Anforderungen ableiten, welche wiederum zu drei grundlegenden Herausforderungen zusammengefasst werden können. Design Thinking erfordert demnach eine domänenspezifische Repräsentation der Aktivitäten und Aufgaben von Automotive Services, um Stakeholder in die Lage zu versetzen den erwarteten Wertbeitrag eines Services zu explizieren. Darüber hinaus müssen die von den Stakeholdern vorgeschlagenen Lösungsmodelle alle notwendigen Informationen vor eine Implementation des Automotive Services beinhalten. Schließlich muss eine Modellierung von Automotive Services ein effektives Erfahrungsmanagement der Ergebnisse und Erkenntnisse vorangegangener Design Thinking Zyklen unterstützen.

Der wesentliche Vorteil einer domänenspezifischen Modellierung liegt in der Erhöhung des Abstraktionsgrades. Anstatt technische oder logische Zusammenhänge der Lösungsdomäne zu beschreiben, werden Konzepte aus der Anwendungsdomäne fokussiert. Dies ermöglicht insbesondere einer Integration der Stakeholder, welche nicht über technologisches Wissen hinsichtlich der Umsetzung von Automotive Services verfügen. Darüber hinaus werden existierendes Wissen und bestehende Erfahrungen in den domänenspezifischen Konzepten der Modellierung hinterlegt. So können beispielsweise das angesprochene Wissen über den Faktor Mensch oder Limitationen und Möglichkeiten von Infotainmentsystemen verfügbar gemacht werden. Schließlich ist eine einfache prototypische Umsetzung durch den Einsatz spezieller domänenspezifischer Code Generatoren möglich.

Die explizite Ausgestaltung der domänenspezifischen Modellierungssprache wird auf der Grundlage eines theoriebasierten Designs vorgeschlagen. Theorien dienen dabei als anwendbares Wissen, welches im Sinne einer Stringenz nach Hevner et al. (2004) aus der Wissensbasis entnommen und als rechtfertigendes Wissen im Rahmen der Designentscheidungen herangezogen wird (Briggs 2006; Baskerville 2008; Schermann/ Böhmann/Krcmar 2007; Gehlert et al. 2009). Jede der aus dem Design Thinking stammenden Herausforderungen wird demnach durch ein abgeleitetes Designprinzip realisiert. Die erste Herausforderung, die domänenspezifische Repräsentation, bezieht sich auf die Fähigkeit der Stakeholder zum Verständnis der Lösung, welches im Sinne der „cognitive fit“ Theorie (Khatri et al. 2006) durch domänenspezifisches Wissen erreicht werde kann Dieses wird wiederum durch das Designprinzip Brick realisiert. Der Problemstellung des Aufwands der Problemlösung, die zweite Herausforderung, kann im Sinne der “cognitive load“ Theorie (Sweller 1988) durch die Reduktion der kognitiven Belastung begegnet werden. Durch das implementierbare Designprinzip Stub wird diese Reduktion der kognitiven Belastung erreicht. Schließlich bezieht sich die dritte Herausforderung des Erfahrungsmanagements auf die

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Fähigkeit der Stakeholder zur Problemlösung. Diese kann durch die Produktivierung von Services (Sawhney 2006) realisiert werden und wird in ASML durch das Designprinzip Composite ermöglicht. Jedes der Designprinzipien bedingt wiederum eine Hypothese, welche im Rahmen der Evaluation überprüft werden kann.

Neben der Gestaltung der eigentlichen Modellierungssprache umfasst eine domänenspezifische Modellierung jedoch nach Tolvanen und Kelly (2004) einige weitere Elemente. Grundsätzlich wird eine Produktidee durch den Einsatz der Modellierungssprache und ein diese unterstützendes Werkzeug in ein Produktmodell übersetzt, welches wiederum durch einen Code Generator und ein domänenspezifisches Framework in den ausführbaren Produktcode transformiert wird. Mit der Automotive Services Modeling Language ist bislang nur die Modellierungssprache vorhanden, die ohne die fehlenden Elemente nur eingeschränkt nutzbar ist. Daher wurde als Modellierungswerkzeug die ASML Workbench und der darin enthaltende ASML Editor entwickelt, der den Stakeholdern die Modellierung neuer Automotive Services ermöglicht. Zusätzlich wurde für das entstehende Produktmodell mit ASM eine geeignete, XML-basierte Repräsentation vorgestellt. Um eine Ausführbarkeit der Modelle zu ermöglichen und so ein einfaches Prototyping zuzulassen, wurde ein Code Generator in Form einer Laufzeitumgebung realisiert. ASML Modelle werden demnach durch die ASML Runtime interpretiert und unter Nutzung des domänenspezifischen Frameworks HIMEPP (Hoffmann 2010) ausgeführt. Somit wurde im Rahmen dieser Arbeit die gesamt Kette von der Produktidee bis zum ausführbaren Produktcode realisiert.

Im letzten Teil der Arbeit erfolgt eine Evaluation des domänenspezifischen Modellierungskonzepts im Rahmen dreier Case Studies. Hierzu werden typische Interaktionen der vier Stakeholdergruppen betrachtet und in diesem Kontext die jeweils zum Einsatz kommenden Designprinzipien untersucht. Auf diese Weise wird einerseits die Kommunikation und Kooperation der einzelnen Stakeholdergruppen und andererseits die aufgestellten Hypothesen evaluiert. Die Beobachtungen der Case Studies zeigen, dass alle drei Hypothesen grundsätzlich bestätigt werden können und somit den aus dem Vorgehen abgeleiteten Herausforderungen an ein Modellierungskonzept durch die gestalteten Designprinzipien erfolgreich begegnet werden kann. Allerdings konnten auch einige Einschränkungen und Limitationen beobachtet werden, die einen weiteren zukünftigen Forschungsbedarf implizieren.

8.2 Limitationen

Der praktische Einsatz des Modellierungskonzepts zeigt, dass Stakeholder der unterschiedlichen Gruppierungen durch ASML in die Lage versetzt werden neue Automotive Services zu explizieren und diese direkt in Form von Prototypen zu bewerten. Allerdings zeigen sich auch wesentliche Limitationen, die einerseits aus konzeptionellen Gesichtspunkten und andererseits auch aus technischen Aspekten der konkreten Umsetzung resultieren.

Zunächst lässt sich feststellen, dass gerade die Gruppe der nichttechnischen Stakeholder durch das Modellierungskonzept leicht überfordert wird. Werden Probanden ohne explizite Schulung in der Modellierungstechnik mit der Aufgabe konfrontiert zeigt sich, dass diese Unterstützung benötigen. Diese Unterstützung bezieht sich einerseits auf formale und praktische Aspekte der Sprache und Werkzeuge, andererseits lässt sich aber auch eine Hemmschwelle Fehler zu machen beobachten. Darüber hinaus zeigt sich auch, dass viele

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

sonstige Stakeholder Probleme haben Services in Form von Abläufen und Prozessen zu beschreiben.

Eine weitere Limitation ist die mangelnde Unterstützung von Unschärfe bei sehr frühen Modellen. Dies lässt sich am einfachsten an zwei konkreten Beispielen zeigen. Das Modellierungskonzept sieht mit Stubs ein Designprinzip vor, mit dessen Hilfe neue domänenspezifische Konzepte in die Modellierung integriert werden können und diese somit erweitert wird. Allerdings müssen diese bislang explizit angegeben werden (über den Brick Wizard) was sich im praktischen Einsatz als hinderlich herausgestellt hat. Insbesondere wird dadurch der kreative Prozess der Gestaltung unterbrochen und von den Nutzern teils zu viele und zu detaillierte Informationen erfragt. Ein zweites Beispiel für notwendige Unschärfe ist das Verknüpfen von Bricks mit Streams. Der Nutzer kann hier nicht einfach eine logische Verbindung zwischen zwei Bricks angeben, sondern muss diese gleich an konkrete Konnektoren knüpfen. Aus der Sicht des sonstigen Stakeholders sind diese Informationen in einem ersten Schritt jedoch oftmals unwesentlich oder noch nicht bekannt.

Die nächste Limitation kann bei Composites identifiziert werden. In ihrer momentanen Gestaltung dienen Composites als Substitute für eingebettet Automotive Services Modelle. Dies wiederum impliziert, dass Properties, über die diese Modelle angepasst werden können, auch im jeweiligen Modell selber angegeben werden müssen. Das Evaluationsbeispiel aus den Case Studies zeigt jedoch, dass ein mehrfaches Verwenden eines Composites mit unterschiedlichen Properties ein naheliegendes Einsatzszenario ist. Um dies effektiv zu realisieren müssten Composites jedoch in der Lage sein einzelne Properties des Modells zu überschreiben.

Auch die praktische Zusammenarbeit mehrere, an einem Automotive Service Modell beteiligter, Stakeholder weist noch praktische Limitationen auf. So gaben alle Probanden der Case Studies an, dass die Übergabe von erstellten oder überarbeiteten Modellen sehr technisch und aufwendig ist. Gleiches gilt auch für neu erstellte Stubs. Die ASML Workbench sollte an dieser Stelle mehr Unterstützung bereitstellen und so ein einfaches, gemeinsames Arbeiten erleichtern.

Schließlich hat der praktische Einsatz der Modellierung gezeigt, dass das Modellierungskonzept die Eigenarten der einzelnen Stakeholdergruppen stärker berücksichtigen sollte. Automotive Service Modelle wurden durchwegs in Zusammenarbeit der einzelnen Stakeholdergruppen erstellt, wobei jede dieser Gruppen ihre spezifische Aufgabe erfüllt. Zunächst wird ein grobes, sehr unscharfes Modell angelegt, welches anschließend um die Expertise des Anwendungsexperten erweitert wird. Schließlich werden technische Details ergänzen. All diese Aufgaben erfordern jedoch unterschiedliche Aspekte der Modellierung und empfinden andere wiederum als störend oder überflüssig. Gerade der ASML Editor könnte hier mehr aufgabenbezogene, praktische Hilfestellung leisten.

8.3 Ausblick und weiterer Forschungsbedarf

Auch wenn die Zielsetzung der Arbeit, die Entwicklung von Automotive Services, also von Mehrwertdiensten im Fahrzeug, durch den Einsatz einer domänenspezifischen Modellierungssprache in einem iterativen Vorgehen zu unterstützen erreicht wurde, implizieren insbesondere die Evaluation sowie die identifizierten Limitationen weiteren Forschungsbedarf. Der nachfolgende Ausblick zeigt einige dieser möglichen Entwicklungen auf und gibt erste Hinweise auf eine eventuelle Gestaltung von Lösungsansätzen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Ein erster Punkt ist das implizite Gestalten neuer domänenspezifischer Konzepte in Form von Stubs. Nutzer sollten diese in Form von graphischen Elementen direkt bei der Gestaltung des Modells erstellen ohne sie explizit, z.B. in Form eines Wizards, anzulegen. Allerdings wirft dieser Ansatz eine Reihe neuer Fragestellungen auf. Wie werden beispielsweise technisch notwendige Details spezifiziert und wie kann sichergestellt werden, dass diese Stubs auch umsetzbar sind?

Eine weitere denkbare Weiterentwicklung sind alternative Laufzeitumgebungen. Die Architektur der domänenspezifischen Modellierung sieht lediglich vor, dass es einen Code Generator mit einem domänenspezifischen Framework gibt. Dies limitiert das ausführbare Ergebnis jedoch auf eben genau diese eine Umgebung. Eine konzeptionelle Entkopplung der Laufzeitumgebe von ihrer Implementierung könnte einen Einsatz identischer Automotive Service Modelle in unterschiedlichen Umgebungen ermöglichen.

Diese Entkopplung führt schließlich auch zur nächsten möglichen Weiterentwicklung. Wenn unterschiedliche Laufzeitumgebungen für unterschiedliche Zielplattformen verfügbar gemacht werden können, ist auch die Gestaltung von Seamless Automotive Services denkbar. Konkret könnte dies bedeuten, dass die einzelnen Bestandteile eines Automotive Services auf unterschiedlichen physischen Instanzen ausgeführt werden und so ein logisch zusammenhängender, jedoch als verteilte Anwendung realisierter, Automotive Service innerhalb eines Modells beschrieben werden kann. Eine einfache Möglichkeit der Umsetzung wäre, Composites im Modell einer bestimmten Ausführungsumgebung zuzuordnen und via Streams über Systemgrenzen hinweg aufzurufen.

Geht man diesen Gedankengang noch einen Schritt weiter, können die Grundlagen von ASML auch zur Gestaltung von allgemeinen mobilen Anwendungen genutzt werden. Das Automotive in ASML zeigt sich konkret durch die domänenspezifischen Konzepte die durch einzelne Bricks repräsentiert werden. Realisiert man hier beispielsweise Smartphonekonzepte bei gleichzeitiger Verfügbarkeit einer entsprechenden Laufzeitumgebung, ist es auch denkbar Services, die nicht den Automotive Services zugeordnet werden können, zu realisieren.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Literaturverzeichnis

Aaen, I. (2008): Essence: facilitating software innovation. In: European Journal of Information Systems, Vol. 17 (2008) Nr. 5 , S. 543-553.

Abate, T. (2004): Digital pioneers / Xerox PARC scientists honored for groundbreaking work on early computers. San Francisco Chronicle, 2004.

Alby, T. (2008): Das mobile Web. Hanser Fachbuchverlag, 2008.

Alcatel Lucent (2010): Alcatel-Lucent Photo Gallery. http://www.alcatel-lucent-events.com/index.php?id=924#, zugegriffen am: 02.08.2010.

Alcatel Lucent (2010): Delivering Online Capabilities on the Road.

Alexander, P. (1992): Domain Knowledge: Evolving Themes and Emerging Concerns. In: Educational Psychologist, Vol. 27 (1992) Nr. 1 , S. 33-51.

Alexander, P.A.; Judy, J.E. (1988): The Interaction of Domain-Specific and Strategic Knowledge in Academic Performance. In: Review of Educational Research, Vol. 58 (1988) Nr. 4 , S. 375-404.

Alpine Electronics Inc. (2010): Alpine - Digital Media Station - iXA-W407BT. view-source:http://www.alpine.de/produkte/product-singleview/digital-media-station/ixa-w407bt.html, zugegriffen am: 12.07.2010.

Alpine Electronics Inc. (2010): Alpine Produkte. http://www.alpine.de/products.html, zugegriffen am: 13.07.2010.

Alpine Electronics Inc. (2010): KCE-425i - 425i iPod Video Interface. http://www.alpine.de/products/details/ipod-interface/kce-425i.html, zugegriffen am: 13.07.2010.

Andriole, S. (1994): Fast, cheap requirements prototype, or else! In: IEEE Software, Vol. 11 (1994) Nr. 2 , S. 85-87.

Antón, A.I. (2003): Successful Software Projects Need Requirements Planning. In: IEEE Software, (2003) , S. 44-46.

Apple Inc. (2010): Apple - iPhone - Mobiltelefon, iPod und Internetgerät. http://www.apple.com/de/iphone/, zugegriffen am: 06.07.2010.

Atteslander, P.; Cromm, J.; Grabow, B. (2006): Methoden der empirischen Sozialforschung. Schmidt, Berlin 2006.

Audi AG (2010): Audi A8. Audi AG, 2010.

Audi AG (2010): Audi Lexikon: MMI. http://www.audi.de/de/brand/de/tools/advice/glossary/mmi.browser.filter_i_m.html, zugegriffen am: 06.07.2010.

Audi AG (2010): Die Audi Online Dienste. http://www.audi.de/de/brand/de/neuwagen/a8/a8/ausstattung/multi_media_interface/audi_online_dienste.html, zugegriffen am: 28.09.2010.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Aurum, A.; Wohlin, C. (2005): Engineering and Managing Software Requirements. Springer, 2005.

Autosieger (2008): BMW iDrive der 2. Generation jetzt noch intuitiver bedienbar. http://www.autosieger.de/article16275.html, zugegriffen am: 05.08.2010.

BMW Group (2010): BMW ConnectedDrive. http://www.bmw.com/com/de/insights/technology/connecteddrive/overview.html, zugegriffen am: 06.07.2010.

BMW Group (2010): BMW Online. http://www.bmw.de/de/de/owners/connecteddrive/2010/online/online.html, zugegriffen am: 07.07.2010.

BMW Group (2010): Following intuition. http://www.bmw.com/com/en/newvehicles/1series/5door/2007/allfacts/ergonomics/navidrive.html, zugegriffen am: 05.08.2010.

Ballagas, R. et al. (2006): The Smart Phone: A Ubiquitous Input Device. In: IEEE Pervasive Computing, Vol. 5 (2006) Nr. 1 , S. 70-77.

Balzert, H. (1996): Lehrbuch der Software-Technik. Spektrum, 1996.

Banfield-Seguin Ltd. (2009): LTE Connected Car - The ng Connect Program. http://www.ngconnect.org/ecosystem/connected-car.htm, zugegriffen am: 13.07.2010.

Baron, J.; Swiecki, B.; Chen, Y. (2006): Vehicle Technology Trends in Electronics for the North American Market; Opportunities for the Taiwanese Automotive Industry. Center for automotive research, 2006.

Baskerville, R. (2008): What design science is not. In: European Journal of Information Systems, Vol. 17 (2008) Nr. 5 , S. 441-443.

Baskerville, R. (2008): What design science is not. In: European Journal of Information Systems, Vol. 17 (2008) Nr. 5 , S. 441-443.

Becker, W. (2002): Network of Automotive Excellence als Lösungsansatz für den Wandel in der Entwicklung/Produktion und Markenpolitik. In: 2nd European Engineering User Conference. Brüssel, 2002.

Bichler, M. (2004): Design Science in Information Systems. In: WIRTSCHAFTSINFORMATIK, (2004) , S. 1-7.

Birbaumer, N.; Schmidt, R.F. (2002): Biologische Psychologie. Springer, 2002.

Blandford, R. (2010): Alpine Electronics and Nokia to integrate smartphones into cars. http://www.allaboutsymbian.com/news/item/11226_Alpine_Electronics_and_Nokia_t.php, zugegriffen am: 13.07.2010.

Bodenstorfer, M.; Hasenauer, R. (2005): Location Based Services — Geschäftsmodelle und Einsatzfelder. Hrsg.: Schuh, A.; Holzmüller, H.H. Physica-Verlag, 2005.

Boehm, B. (1984): Verifying and Validating Software Requirements and Design Specifications. In: IEEE Software, Vol. 1 (1984) Nr. 1 , S. 75-88.

Boehm, B.W. (1988): A spiral model of software development and enhancement. In: Computer, Vol. 21 (1988) Nr. 5 , S. 61-72.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Bonobo (2010): iBoost - The iPhone App For Making Your Car Sound Turbo. http://www.iboostapp.com/, zugegriffen am: 13.07.2010

Bortz, J.; Döring, N. (2006): Forschungsmethoden und Evaluation: für Human- und Sozialwissenschaftler. Springer, Berlin 2006.

Braun, C. (2007): Modellierung der Unternehmensarchitektur. Universität St. Gallen, 2007.

Breindahl, B.C. (2008): A Scandinavian Approach to Interaction Design.

Briggs, R. (2006): On theory-driven design and deployment of collaboration systems. In: International Journal of Human-Computer Studies, Vol. 64 (2006) Nr. 7 , S. 573-582.

Briggs, R.; Gruenbacher, P. (2002): EasyWinWin : Managing Complexity in Requirements Negotiation with GSS. In: 35th Hawaii International Conference on System Sciences, 2002.

Brooks, F. (1986): No Silver Bullet: Essence and Accidents of Software Engineering. In: Proceedings of the IFIP Tenth World Computing Conference, 1986.

Brown, T. (2009): Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation. HarperBusiness, 2009.

Brügge, B.; Dutoit, A.H. (2004): Objektorientierte Softwaretechnik. Mit Entwurfsmustern, UML und Java. Pearson, 2004.

Budinsky, F. et al. (2003): Eclipse Modeling Framework. Addison-Wesley Professional, 2003.

Bundesministerium für Wirtschaft und Technologie (2009): Nationaler Entwicklungsplan Elektromobilität der Bundesregierung.

BunsenTech LLC (2010): Dynolicious. http://www.bunsentech.com/projects/dynolicious/, zugegriffen am: 13.07.2010.

Busch, F. (2007): Car-to-X im Verkehrswesen. Technische Universität München, München 2007.

Busch, V. (1945): As We May Think. The Atlantic, 1945.

Buttkereit, S. et al. (2009): Mobile broadband for the masses.

Böhm, M. et al. (2010): Cloud Computing - Outsourcing 2.0 or a new Business Model for IT Provisioning? In: Application Management Service Management und Service Creation, 2010.

CAR 2 CAR Communication Consortium (2007): C2C-CC Manifesto.

CMMI Product Team (2002): Capability Maturity Model Integration (CMMI SM), CMMI SM for Systems Engineering , Software Engineering , Integrated Product and Process Development , and Supplier Sourcing.

Card, S.K.; Moran, T.P.; Newell, A. (1986): The Psychology of Human-Computer Interaction. Lawrence Erlbaum Associates, 1986.

Carroll, J.M. (2001): Human-Computer Interaction in the New Millennium. Addison-Wesley, 2001.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Carticipate (2008): Carticipate - An Experiment in Social Transportation. http://www.carticipate.com/, zugegriffen am: 13.07.2010.

Chesbrough, H. (2003): Open Innovation: The New Imperative for Creating and Profiting from Technology. Mcgraw-Hill Professional, 2003.

Christel, M.G.; Kang, K.C. (1992): Issues in Requirements Elicitation.

Continental Automotive GmbH (2010): Continental Automotive – AutoLinQ. http://www.autolinq.de/about_autolinq.aspx, zugegriffen am: 12.07.2010.

Cunningham, W. (2007): Driving IT - Car interfaces and usability. http://reviews.cnet.com/4520-10895_7-6744922-1.html, zugegriffen am:

Cunningham, W. (2009): Sneak peek: VW's next gen infotainment system. http://reviews.cnet.com/8301-13746_7-10245227-48.html, zugegriffen am:

Daimler AG (2010): S-Klasse - Betriebsanleitung Interaktiv - [Überblick | Mittelkonsole unten]. http://www4.mercedes-benz.com/manual-cars/ba/cars/221/de/overview/konsoleunten.html, zugegriffen am:

Daimler AG (2010): smart drive kit for the iPhone: Mehr Sicherheit und Lifestyle. http://www.daimler.com/dccom/0-5-633234-49-1274927-1-0-0-0-0-0-9293-7145-0-0-0-0-0-0-0.html, zugegriffen am:

Davies, A.; Brady, T.; Hobday, M. (2007): Organizing for solutions: Systems seller vs. systems integrator. In: Industrial Marketing Management, Vol. 36 (2007) Nr. 2 , S. 183-193.

Dawson, F. (2010): Cable Pins Future Hopes Web HDTVs Not to Be Taken Lightly. In: ScreenPlay Magazine, (2010) , S. 14-15.

Deutschen Institut für Normung (1994): Beurteilen von Softwareprodukten; Qualitätsmerkmale und Leitfaden zu deren Verwendung. , zugegriffen am:

Diekmann, A. (2007): Empirische Sozialforschung: Grundlagen, Methoden, Anwendungen. rororo, 2007.

Dijkstra, E.W. (1972): The humble programmer. In: Communications of the ACM, Vol. 15 (1972) Nr. 10 , S. 859-866.

Eberlein, A.; Leite, J.C. (2002): Agile Requirements Definition : A View from Requirements Engineering Julio Cesar Sampaio do Prado Leite. In: 2002.

Ebert, C. (2005): Systematisches Requirements Management. Dpunkt Verlag, 2005.

Eclipse Foundation (2005): The Eclipse Modeling Framework (EMF) Overview. http://help.eclipse.org/ganymede/index.jsp?topic=/org.eclipse.emf.doc/references/overview/EMF.html, zugegriffen am:

Eclipse Foundation (2010): Eclipse Projects. http://www.eclipse.org/projects/listofprojects.php, zugegriffen am:

Eclipse Foundation (2010): GEF Description. http://wiki.eclipse.org/GEF_Description, zugegriffen am:

Ehmer, M. (2002): Mobile Dienste im Auto – Die Perspektive der Automobilhersteller. In: Hrsg.: Reichwald, R. Gabler, 2002.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden. Es gilt nur für den persönlichen Gebrauch.

Engel, A.K.; Gold, P. (1998): Der Mensch in der Perspektive der Kognitionswissenschaften. Suhrkamp Verlag, 1998.

Faber, M. (2009): Open Innovation. Gabler, Wiesbaden 2009.

Festag, A. et al. (2008): Safety and Security CAR-2-X Communication for Safety and Infotainment in Europe. In: Nec Technical Journal, Vol. 3 (2008) Nr. 1 , S. 21-26.

Fettke, P.; Loos, P. (2003): Multiperspective Evaluation of Reference Models – Towards a Framework. In: Springer, Berlin 2003.

Floyd, C. et al. (1989): Out of Scandinavia : Alternative Approaches to Software Design and System Development. In: Human-Computer Interaction, Vol. 4 (1989) , S. 253-350.

Ford (2010): Ford SYNC Features - Hands free phone, music, ringtones, text messaging, directions, emergency services, more. http://www.fordvehicles.com/technology/sync/features/#page=Feature8, zugegriffen am: 12.07.2010.

Forrai, J. (2010): Evaluierung , Entwurf und Implementierung einer GUI-Plattform für prototypische Dienste.

Fürtsch, T. (2010): smart bietet iPhone Integration mit Halter und App. http://iphoneforcars.wordpress.com/2010/02/26/smart-bietet-iphone-integration-mit-halter-und-app/, zugegriffen am: 07.07.2010.

Galbraith, J.R. (2002): Organizing to Deliver Solutions. In: Organizational Dynamics, Vol. 21 (2002) Nr. 2 , S. 194-207.

Galitz, W.O. (2007): The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques. John Wiley & Sons, 2007.

Gebhardt, C. (2009): Großserien-Elektroauto feiert Weltpremiere. http://www.auto-motor-und-sport.de/eco/nissan-leaf-elektroauto-1361434.html, zugegriffen am: 28.09.2010.

Gehlert, A. et al. (2009): Towards a Research Method for Theory-driven Design Research. In: Internationale Tagung Wirtschaftsinformatik, Wien 2009.

Geis, T. (2005): Willkommen im Usability Begriffszoo. http://www.fit-fuer-usability.de/archiv/willkommen-im-usability-begriffszoo, zugegriffen am: 30.07.2010.

Geisser, M. et al. (2007): Verteiltes, internetbasiertes Requirements-Engineering. In: Wirtschaftsinformatik, Vol. 49 (2007) Nr. 3 , S. 199-207.

Glaska, R. et al. (2008): Assisting Needs Driven Requirements Engineering with the ARIS Toolset. In: Software Engineering, München 2008.

Golcar, M. et al. (2010): Heuristische Ermittlung und Typisierung von Infrastrukturanforderungen mobiler Dienste im Fahrzeug. In: MKWI 2010 – Automotive Services 2010, Göttingen 2010.

Grechanik, M.; Conroy, K.M.; Probst, K.A. (2007): Finding Relevant Applications For Prototyping. In: International Conference on Software Engineering 2007.

Gregor, S. (2006): The Nature of Theory in Information Systems. In: MIS Quarterly, Vol. 30 (2006) Nr. 3 , S. 611-642.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden. Es gilt nur für den persönlichen Gebrauch.

Greiffenberg, S. (2004): Methodenentwicklung in Wirtschaft und Verwaltung. Kovac, J, 2004.

Hanrieder, G. (2004): Sprachbedienung im KFZ – Eine Erfolgsgeschichte. In: Lecture Notes in Informatics (LNI) - Proceedings, Ulm 2004.

Hansen, H.R.; Neumann, G. (2001): Wirtschaftsinformatik I. Grundlagen betrieblicher Informationsverarbeitung. Lucius & Lucius, 2001.

Harman Becker Automotive Systems GmbH (2003): Fahrerinformationssystem.

Hass, B.; Walsh, G.; Kilian, T. (2007): Web 2.0: Neue Perspektiven für Marketing und Medien. Springer, Berlin 2007.

Herczeg, M. (2006): Einführung in die Medieninformatik. Oldenburg, 2006.

Hevner, A.R. et al. (2004): Design Science in Information Systems Research. In: Information Systems Research, Vol. 28 (2004) Nr. 1 , S. 75-105.

Hoffmann, H. (2010): Ein Werkzeug zur Entwicklung nutzerorientierter Software- und Service-Prototypen im Fahrzeug.

Hofmann, H.; Lehner, F. (2001): Requirements engineering as a success factor in software projects. In: IEEE Software, Vol. 18 (2001) Nr. 4 , S. 58-66.

Hood, C. et al. (2007): Anforderungsmanagement Inhaltsverzeichnis. Heise, Hannover 2007.

Hug, T. (2001): Wie kommt Wissenschaft zu Wissen? - Einführung in die Forschungsmethodik und Forschungspraxis. Schneider, Hohengehren 2001.

Husen, C.V. (2007): Anforderungsanalyse für produktbegleitende Dienstleistungen.

IEEE Standards Board (1990): IEEE Standard Glossary of Software Engineering Terminology.

IEEE Standards Board (1998): IEEE Guide for Developing System Requirements Specifications.

IEEE Standards Board (1998): Recommended Practice for Software Requirements Specifications.

International Organization for Standardization (1998): Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability.

International Organization for Standardization (2000): Software engineering -- Product quality -- Part 1: Quality model.

International Organization for Standardization (2004): Grundsätze der Ergonomie für die Gestaltung von Arbeitssystemen.

Itten, J. (1974): Art of Color: The Subjective Experience and Objective Rationale of Color. Wiley, 1974.

Jacobsen, H. (2004): Middleware for Location-Based Services. In: Location Based Services Hrsg.: Schiller, J.; Voisard, A. Morgan Kaufmann, 2004.

Jacobson, I. (1992): Object Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, 1992.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Jennex, M.E. (2001): Research Relevance-You Get What You Reward. In: Communications of the Association for Information Systems, Vol. 6 (2001) Nr. 1.

Jirotka, M. (1994): Requirements Engineering: Social and Technical Issues (Computers and People). Academic Press, 1994.

John, B.E.; Kieras, D.E. (1996): Using GOMS for user interface design and evaluation: which technique? In: ACM Transactions on Computer-Human Interaction, Vol. 3 (1996) Nr. 4 , S. 287-319.

Kastens, U.; Büning, H.K. (2008): Modellierung. Grundlagen und formale Methoden. Hanser Fachbuch, 2008.

Kaufmann, A. (2009): Volkswagen shows off GLORIA advanced infotainment system. http://www.motorauthority.com/blog/1033235_video-volkswagen-shows-off-gloria-advanced-infotainment-system, zugegriffen am: 12.07.2010.

Kelly, S. (2005): Software-Modellierung ohne Kunstgriffe - Mit domänenspezifischer Modellierung zu hundertprozentiger Code-Generierung. In: Elektronik, 2005.

Khatri, V. et al. (2006): Understanding Conceptual Schemas: Exploring the Role of Application and IS Domain Knowledge. In: Information Systems Research, Vol. 17 (2006) Nr. 1 , S. 81-99.

Kieras, D. (1996): A Guide to GOMS Model Usability Evaluation using NGOMSL.

Knauer, R. (2002): Entwerfen und Darstellen: Die Zeichnung Als Mittel Des Architektonischen Entwurfs. Ernst & Sohn, 2002.

Koch, M.; Richter, A. (2007): Enterprise 2.0: Planung, Einführung und erfolgreicher Einsatz von Social Software in Unternehmen. Oldenburg, 2007.

Kortlüke, N.; Pieprzyk, B. (2010): Klimafreundliche Elektromobilität : Finanzielle Hürden zur Markteinführung bis 2020.

Kotonya, G.; Sommerville, I. (1998): Requirements Engineering: Processes and Techniques. John Wiley & Sons, 1998.

Krauß, T.; Versteegen, G.; Mühlbauer, S. (2006): Model Driven Requirements Engineering. In: Informatik Spektrum, Vol. 29 (2006) Nr. 6 , S. 460-464.

Krcmar, H. (2010): Informationsmanagement. Springer, Berlin 2010.

Krogstie, J. (1998): Integrating the understanding of quality in requirements specification and conceptual modeling. In: ACM SIGSOFT Software Engineering Notes, Vol. 23 (1998) Nr. 1 , S. 86-91.

Kruchten, P. (2003): The Rational Unified Process. An Introduction. Addison-Wesley Longman, Amsterdam 2003.

Lawson, H. (1999): Defining stakeholder relationships. In: Computer, Vol. 32 (1999) Nr. 3 , S. 110-112.

Leimeister, J.M. (2010): Automotive Services – auf dem Weg zu einem Wachstumsfeld für Wirtschaft und Wirtschaftsinformatik. In: Automotive Services 2010, Göttingen 2010.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Leimeister, J.M.; Glauner, C. (2008): Hybride Produkte – Einordnung und Herausforderungen für die Wirtschaftsinformatik. In: Wirtschaftsinformatik, Vol. 50 (2008) Nr. 3 , S. 248-251.

Maisch, B.; Meckel, M. (2009): Innovationskommunikation 2.0 — Das Beispiel Apple iPhone. In: Marketing Review St. Gallen, Vol. 26 (2009) Nr. 2 , S. 42-46.

March, S.T.; Smith, G.F. (1995): Design and natural science research on information technology. In: Decision Support Systems, Vol. 15 (1995) Nr. 4 , S. 251-266.

March, S.T.; Smith, G.F. (1995): Design and natural science research on information technology. In: Decision Support Systems, Vol. 15 (1995) Nr. 4 , S. 251-266.

Markus, M.L.; Majchrzak, A.; Gasser, L. (2002): Special issue a design theory für systems that that support emergent knowledge. In: MIS Quarterly, Vol. 26 (2002) Nr. 3 , S. 179-212.

Mayring, P. (2007): Qualitative Inhaltsanalyse. Grundlagen und Techniken. Utb., 2007.

McPeck, J.E. (1990): Critical Thinking and Subject Specificity: A Reply to Ennis. In: Educational Researcher, Vol. 19 (1990) Nr. 4 , S. 10-12.

Mercedes-Benz Bank (2010): iPhoneApp der Mercedes Benz Bank hilft Autofahren im Schadensfall. http://www.mercedes-benz-bank.de/intrade/cms/Presse_I_Phone_APP_PI.html;jsessionid=0000-aN259X9vrscLi23lTbg5vA:14730k44i, zugegriffen am: 13.07.2010.

Meroth, A.; Tolg, B. (2007): Infotainmentsysteme im Kraftfahrzeug. Grundlagen, Komponenten, Systeme und Anwendungen. Vieweg+Teubner, 2007.

Miller, G.A. (1994): The magical number seven, plus or minus two: some limits on our capacity for processing information. 1956. In: Psychological review, Vol. 101 (1994) Nr. 2 , S. 343-52.

Mitchell, R.K.; Agle, B.R.; Wood, D.J. (1997): Toward a Theory of Stakeholder Identification and Salience: Defining the Principle of Who and What Really Counts. In: The Academy of Management Review, Vol. 22 (1997) Nr. 4 , S. 853.

Mrkwicka, K.; Kiessling, M.; Kolbe, L.M. (2009): Potential of Web 2 . 0 Applications for Viewer. In: San Francisco, California 2009.

Müller-Bagehl, C.; Endt, P. (2004): Infotainment / Telematik im Fahrzeug. Trends für die Serienentwicklung. Expert-Verlag, 2004.

Nicolescu, V. (2009): Gestaltung avatarbasierter natürlichsprachlicher Hilfesysteme für den Einsatz in Fahrzeugen.

Niefünd, R. (2004): Fahrerassistenzsysteme Für PKWs.

Nissan (2010): NISSAN | TECHNOLOGICAL DEVELOPMENT ACTIVITIES | Overview | Robotic Interface. http://www.nissan-global.com/EN/TECHNOLOGY/INTRODUCTION/DETAILS/RI/, zugegriffen am: 12.07.2010.

Norman, D.A. (1996): Dinge des Alltags. Gutes Design und Psychologie für Gebrauchsgegenstände. Campus Verlag GmbH, 1996.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden. Es gilt nur für den persönlichen Gebrauch.

Norman, D.A.; Draper, S.W. (1986): User Centered System Design: New Perspectives on Human-computer Interaction. CRC Press, 1986.

Nuseibeh, B.; Easterbrook, S. (2000): Requirements Engineering : A Roadmap. In: Limerick, Ireland 2000.

O'Reilly, T. (2005): What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software. http://oreilly.com/web2/archive/what-is-web-20.html, zugegriffen am: 25.08.2010.

Pacheco, C.; Tovar, E. (2007): Stakeholder identification as an issue in the improvement of software requirements quality. In: Springer, Berlin 2007.

Partsch, H. (2010): Requirements-Engineering systematisch: Modellbildung für softwaregestützte Systeme. Springer, Berlin 2010.

Peters, I. (2009): Folksonomies: Indexing and Retrieval in the Web 2.0. Saur, 2009.

Pitschke, J. (2009): Was ist ein Modell ? (Und wenn ja , wieviele ?). 2009.

Pohl, K. (2008): Requirements Engineering: Grundlagen, Prinzipien,Techniken. dpunkt.Verlag GmbH, 2008.

Preiss, O.; Wegmann, A. (2001): Stakeholder discovery and classification based on systems science principles. IEEE Comput. Soc, 2001.

Pressman, R.S. (2009): Software Engineering: A Practitioner's Approach. Mcgraw-Hill Publ.Comp., 2009.

Purao, S. (2002): Design Research in the Technology of Information Systems : Truth or Dare. Atlanta 2002.

Quatrani, T. (1997): Visual Modeling With Rational Rose and Uml. Addison-Wesley, 1997.

Reichwald, R.; Krcmar, H.; Reindl, S. (2007): Mobile Dienste im Auto der Zukunft. Eul, 2007.

Reimer, J. (2005): A History of the GUI. In: ars technica, 2005.

Riedl, C. (2010): Audi - Innovation Repository. http://audi.ideaontology.org/web/guest/4?p_p_id=innovation_repository_WAR_InnovationRepository_INSTANCE_Gcrg&p_p_action=0&p_p_state=normal&p_p_mode=view&p_p_col_id=column-1&p_p_col_pos=1&p_p_col_count=2&_innovation_repository_WAR_InnovationRepository_INSTANCE_Gcrg_cmd=showDetails&_innovation_repository_WAR_InnovationRepository_INSTANCE_Gcrg_ideaID=6060, zugegriffen am: 16.08.2010.

Riedl, C. (2010): Audi - Innovation Repository. http://audi.ideaontology.org/web/guest/4?p_p_id=simple_tag_cloud_WAR_InnovationRepository_INSTANCE_6sCo&p_p_action=1&p_p_state=normal&p_p_mode=view&p_p_col_id=column-2&p_p_col_pos=1&p_p_col_count=2&_simple_tag_cloud_WAR_InnovationRepository_INSTANCE_6sCo_view=tag&_simple_tag_cloud_WAR_InnovationRepository_INSTANCE_6sCo_viewTag=+HGW, zugegriffen am: 16.08.2010.

Robertson, S.; Robertson, J. (1999): Mastering the Requirements Process. Addison-Wesley Longman, Amsterdam 1999.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Rogalski, M. (2005): The Eclipse Rich Client Platform Reasons for RCP OSGi framework. IBM, 2005.

Rosemann, M.; Vessey, I. (2008): Toward Improving the Relevance of Information Systems Research to Practice. In: MIS Quarterly, Vol. 32 (2008) Nr. 1 , S. 1-22.

Rupp, C. (2007): Requirements- Engineering und -Management. Hanser, München 2007.

Sadikov, E. (2010): Take me to my car. http://takemetomycar.anresgroup.com/, zugegriffen am: 13.07.2010.

Sawhney, M. (2006): Going Beyond the Product: Defining, Designing and Delivering Customer Solutions. In: Hrsg.: Lusch, R.F.; Vargo, S.L. Sharpe, Armonk, NY 2006.

Scheer, A. (2001): ARIS-Modellierungs-Methoden, Metamodelle, Anwendungen. Springer, Berlin 2001.

Schermann, M. (2009): Risk Service Engineering : Eine Modellierungsmethode zur wertorientierten Entwicklung von Maßnahmen der Risikosteuerung im Informationsmanagement. , zugegriffen am:

Schermann, M.; Böhmann, T.; Krcmar, H. (2007): A pattern-based approach for constructing design theories with conceptual models. In: Proceedings der European Conference on Information Systems, 2007.

Schienmann, B. (2001): Kontinuierliches Anforderungsmanagement . Prozesse - Techniken - Werkzeuge. Addison-Wesley, 2001.

Schmidt, R.F.; Lang, F.; Heckmann, M. (2007): Physiologie des Menschen: Mit Pathophysiologie. Springer, 2007.

Schmitz, C. (2010): LimeSurvey Features. http://www.limesurvey.org/de/ueber-limesurvey/features, zugegriffen am: 07.09.2010.

Schrage, M. (2004): Never go to a client meeting without a prototype. In: IEEE Software, Vol. 21 (2004) Nr. 2 , S. 42-45.

Schwabe, G.; Krcmar, H. (1996): Der Needs Driven Approach. In: Herausforderung der Telekooperation - Proceedings der DCSCW, Heidelberg 1996.

Schätzl, A. (2001): Kraft und Komfort in der Zukunft. Süddeutsche Zeitung, 2001.

Schütte, R. (1998): Grundsätze ordnungsmäßiger Referenzmodellierung. Gabler Verlag, 1998.

Sears, A.; Jacko, J.A. (2007): The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications. Lawrence Erlbaum Assoc, 2007.

Sharp, H.; Finkelstein, A.; Galal, G. (1999): Stakeholder identification in the requirements engineering process. IEEE, 1999.

Sharp, H.; Rogers, Y.; Preece, J. (2007): Interaction Design: Beyond Human-Computer Interaction. Wiley, 2007.

Shneiderman, B. (1997): Designing the User Interface. Strategies for Effective Human-Computer Interaction. 3rd. Aufl., Addison-Wesley Longman, Amsterdam 1997.

Simon, H.A. (1996): Sciences of the Artificial. The Mit Press, 1996.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Smith, C. (2009): Toyota Prius Becomes the First LTE Connected Car in the World [Toyota and ng Connect to Deliver 4G LTE Connected Car in 2010?]. http://nexus404.com/Blog/2009/11/07/toyota-prius-becomes-the-first-lte-connected-car-in-the-world-toyota-and-ng-connect-to-deliver-4g-lte-connected-car-in-2010/, zugegriffen am: 13.07.2010.

Sommerville, I.; Sawyer, P. (1997): Requirements Engineering: A Good Practice Guide. John Wiley & Sons, 1997.

Special Interest Group in Computer-Human Interaction (ACM SIGCHI) (2009): Welcome - SGIHCI. http://www.sigchi.org/, zugegriffen am: 30.07.2010.

Stachowiak, H. (1974): Allgemeine Modelltheorie. Springer, Wien 1974.

Stahl, T.; Voelter, M.; Czarnecki, K. (2006): Model-Driven Software Development: Technology, Engineering, Management. Wiley, 2006.

Strahringer, S. (1996): Metamodellierung als Instrument des Methodenvergleichs - Eine Evaluierung am Beispiel objektorientierter Analysemethoden. Shaker Verlag, 1996.

Surich Technologies Inc. (2010): iPhone 3G app - Fuel cost and driving efficiency analyzer - Trip Alyizer - Surich Technologies Inc. http://www.surichtech.com/mobile/TripAlyizer/index.htm, zugegriffen am: 13.07.2010.

Sutherland, I.E. (1964): Sketch pad a man-machine graphical communication system. In: New York, NY, USA 1964.

Sweller, J. (1988): Cognitive load during problem solving: Effects on learning. In: Cognit. Sci, Vol. 12 (1988) , S. 257-285.

Takeda, H. et al. (1990): Modeling Design Processes. In: AI Magazine, Vol. 11 (1990) Nr. 4 , S. 37-48.

Te'eni, D.; Carey, J.M.; Zhang, P. (2006): Human-Computer Interaction: Developing Effective Organizational Information Systems. Wiley, 2006.

Thomas, M. (2001): Die Vielfalt der Modelle in der Informatik. In: Informatikunterricht und Medienbildung, INFOS 2001, 9. GI-Fachtagung Informatik und Schule. 2001.

Tinham, B. (2007): Mobile IT : for real. In: Manufacturing Computer Solutions, Vol. 13 (2007) Nr. March 2007, S. 40-41.

Tolvanen, J. (2006): Domänenspezifische Modellierung für vollständige Code-Generierung. In: Javaspektrum, 2006.

Tolvanen, J.; Kelly, S. (2004): Domänenspezifische modellierung. In: Objektspektrum, 2004.

TomTom International BV (2010): TomTom LIVE Services. http://www.tomtom.com/services/, zugegriffen am: 13.07.2010.

TomTom International BV (2010): TomTom, tragbare GPS-Fahrzeugnavigationssysteme. http://www.tomtom.com/services/service.php?id=14, zugegriffen am: 13.07.2010.

Van Buskirk, R.; Moroney, B.W. (2003): Extending prototyping. In: IBM Systems Journal, Vol. 42 (2003) Nr. 4 , S. 613-623.

Versteegen, G. (2007): Neues im Anforderungsmanagement. In: Objektspekrum, 2007.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Viswanathan, S. et al. (2007): Online Infomediaries and Price Discrimination: Evidence from the Automotive Retailing Sector. In: Journal of Marketing, Vol. 71 (2007) Nr. 3 , S. 89-107.

Vogel, L. (2010): Eclipse RCP Tutorial. http://www.vogella.de/articles/EclipseRCP/article.html, zugegriffen am: 25.08.2010.

Volkswagen AG (2010): Car-to-X - Ein Kommunikationssystem für viele Anwendungen. http://www.volkswagenag.com/vwag/vwcorp/content/de/innovation/communication_and_networking/connected_world/car_to_x.html, zugegriffen am: 13.07.2010.

Volkswagen AG (2010): Volkswagen lädt Nutzer zur Mitgestaltung von Infotainmentsystemen der Zukunft ein. http://www.volkswagen.de/vwcms/master_public/virtualmaster/de3/unternehmen/mobilitaet_und_nachhaltigkeit/Aktuelles1/meldungen1/II_2010/27-04-2010.html, zugegriffen am: 13.07.2010.

Walls, J.G.; Widmeyer, G.R.; El Sawy, O.A. (1992): Building an Information System Design Theory for Vigilant EIS. In: Information Systems Research, Vol. 3 (1992) Nr. 1 , S. 36-59.

Wandke, H.; Wetzenstein, E.; Polkehn, K. (2005): Handlungsbezogene Elementarbausteine für Fahrerassistenzsysteme. In: Vdi Berichte, (2005) Nr. 1919.

Weber, R. (1987): Toward A Theory of Artifacts: A Paradigmatic Base for Information Systems Research. In: Journal of Information Systems, Vol. 1 (1987) Nr. 2 , S. 3-19.

Weiss, C.H.; Küchler, M. (1988): Evaluierungsforschung. Methoden zur Einschätzung von sozialen Reformprogrammen. VS Verlag für Sozialwissenschaften, 1988.

Wessling, R. (2010): i-app.net - iHUDisplay. http://www.i-app.net/applications/ihudisplay/, zugegriffen am: 13.07.2010.

West, J.; Mace, M. (2010): Browsing as the killer app: Explaining the rapid success of Apple's iPhone. In: Telecommunications Policy, Vol. 34 (2010) Nr. 5-6 , S. 270-286.

Wiegers, K.E. (2003): Software Requirements, Second Edition. Microsoft Press, 2003.

Wikipedia (2010): Augenbewegung.

Wikipedia (2010): Johannes Itten.

Wikipedia (2010): Microsoft Windows.

Wikipedia (2010): Xerox Star.

Winograd, T. (1996): Bringing Design to Software. ACM Press, 1996.

Young, S. et al. (2001): Quality and people in the development of situationally specific methods. IEEE Comput. Soc, 2001.

Zowghi, D.; Coulin, C. (2005): Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. 2005.

van Aken, J.E. (2004): Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-Tested and Grounded Technological Rules. In: Vol. 41 (2004) Nr. 2 , S. 219-246.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang

Anhang A Gesprächsprotokolle der Interviews

Anhang A.1 Alpha

Datum: 16.06.2008

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	2,3 Mio. €
Mitarbeiter gesamt	20

2. Produkte des Unternehmens

- ***Das Unternehmen entwickelt Individualsoftware, als auch. Open-Source-Software***

  Auch IT-Beratung

- ***Bieten Sie auch Dienstleistungen zusammen mit dem Produkt an? Falls ja, welche? Werden Dienstleistungen unabhängig vom Produkt angeboten?***

  Es werden auch Dienstleistungen unabhängig vom Produkt angeboten, zum Beispiel Schulungen.

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Software-Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell?***

  Vorwiegend V-Modell

3. Projekte des Unternehmens

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  Aus einem Forschungsprojekt (intern)

- ***Könnten Sie das Projekt kurz beschreiben?***

  Ein Produktdatenmanagementsystem, dass an Prozess(Prozessphasen) angebunden ist.

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

  2001 hat das Projekt angefangen.

- ***Wie lange dauert ein Projekt durchschnittlich?***

  Projektspezifisch. Mindestens 2 Wochen. Es gibt Projekte, die 4-5 Jahre dauern. Im Durchschnitt dauert ein Projekt 3-4 Monate.

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen***

  Momentan 4. Allgemein unterschiedlich: 1-5,6. Meistens 1-2

  Rollen: Projektleiter, Mitarbeiter: Softwareentwickler, Analytiker, Designer, Qualitätssicherer, Architekten.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

4. Anforderungsmanagement

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Projektspezifisch, aber meist am Anfang des Projekts und dann am Ende.

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Nicht festgeschrieben. Workshop(Diskussion) mit Protokoll-> Lastenheft -> Validierung mit den Kunden.

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…)***

  Der Kunde und der Anwender geben die Anforderungen. Außerdem die Rahmenbedingungen (interne Anforderungen), z. B. bezüglich der Programmierrahmen aus IT Abteilung.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

  Die externen Anforderungen werden in natürlicher Sprache verfasst und müssen nicht übersetzt werden.

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  projektspezifisch

- ***Wie wird damit umgegangen, dass viele Anforderungen erst während der Entwicklung dazukommen?***

  Hängt davon ab, wie umfangreich die Anforderungsänderungen sind. Meistens werden sie übernommen und implementiert. Der Umfang der Anforderung bestimmt den Betrag, der vom Auftraggeber verlangt wird.

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  Die meisten beziehen sich auf die Funktionalität.

- ***Welcher Aufwand entsteht in einem durchschnittlichen Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

  Projektspezifisch

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  Neues Arbeitspaket wird aufgemacht.

- ***Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  -

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  Anforderungsanalysten und Reviewer.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, usw.)?***

  Interview, Workshop, Disskusion, Brainstorming

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Konflikte zwischen Anforderungen, der Auftraggeber kann nicht genau sagen, was er vom Produkt erwartet. Verschiedene Disziplinen (Interdisziplinäre Team), unterschiedliche Sicht von verschiedenen Kundengruppen.

- ***Wie werden die Anforderungen analysiert?***

  Während der Formulierung des Lastenhefts und in Workshops.

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Ja: muss, soll, kann

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  Lastenheftvorlagen, Use-Cases, Schablonen, Satzschablonen, UML

- ***Welche Informationen werden dokumentiert? Werden die Änderungen an Anforderungen dokumentiert? Falls ja, was wird dabei genau dokumentiert?***

  Namen, Vorbedingungen, Nachbedingungen, Änderungen an Anforderungen (nicht immer im Lastenheft), von wem kommt(Ansprechspartner)

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  Es werden mehrere Werkzeuge eingesetzt.

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

  Projektspezifisch, gemeinsam demokratisch in der Gruppe anhand Prioritäten

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Während der Abnahme, formal.

8. Prototypen

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  Anforderungsanalyse – experimentell, explorativ und evolutionär. Anforderungsüberprüfung oder Prüfung von neuen Funktionalitäten.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  Endanwender, Management. Prototypen stellen ein Mittel zur Kommunikation dar.

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  Es werden Prototypen gebaut. Keine bestimmte Sprache für Prototypsentwicklung.

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  Die Sachen werden nebenbei gemacht.

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  Dagegen: nur konzeptionell.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.2 Beta

Datum: 08.09.2008

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	19.000 Mio €
Mitarbeiter gesamt	180.000

2. Produkte des Unternehmens

- ***Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  Nur Dienstleistungen; Outsourcing Dienstleistung z.B. Buchhaltung für ein Unternehmen

  Individualsoftware: Als Dienstleistung von einem externen Unternehmen

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren:***

  Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware

  Kommerzielle Software vs. Open-Source-Software

  Individualsoftware: siehe oben

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  Wir verwenden ein internes Vorgehensmodell, das auf dem V-Modell basiert. Es umfasst alle Lebenszyklen: Von Strategie bis Konzeption und Application-Management und Outsourcing.

  RE ist eine Phase im Vorgehensmodell.

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  Auftrag eines Kunden

- ***Könnten Sie das Projekt kurz beschreiben?***

  Es handelt sich um den Bereich von Banken, im speziellen Investmentbanken: Es sollte ein Handelssystem für strukturierte Produkte entwickelt werden. In solchen Produkten werden Aktien und Anleihen gemischt. Dieser Bereich ist hochkomplex in der Abwicklung. Das Gesamtprodukt wird auf einzelne Produkte aufgebaut.

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

  keine

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

3 Phasen. Die 3te läuft gerade. Das System ist bereits seit 2 Jahren im Produktivbetrieb. Das Projekt ist in 2005 angefangen.

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

  - Beta: 20-25 Personen (nicht alle Vollzeit).
    Rollen: Projektleiter und 2 Teams:
    Ein fachliches Team (4-5 Personen): Fachkonzepte, Testanforderungen, Abstimmungen mit dem Kunden.
    Technisches Team: Architektur, Implementierung, technischer Test.
  - Kunde: 10-15 (nicht alle Vollzeit).

4. Requirements Engineering

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Mischung aus beiden.RE ist eine Phase im Projekt.

  Am Anfang wurde eine fachliche Anforderungssammlungsphase gemacht. Dort wurde gesammelt und Priorisiert und dann festgelegt, welche Anforderungen realisiert werden.

  Während der Entwicklung des Release 1 wurde teilweise neu priorisiert. Am Ende des Release 1 wurde aus dem gelernten Wissen und den nicht realisierten Anforderungen eine neue Liste erstellt und nochmals Priorisiert. Diese neue Liste wurde dann im Release 2 realisiert. Derzeit entwickeln wir das Release 3; schon am Anfang war geplant mehrere Releases zu entwickeln.

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Alle diese Phasen, außer dem Change-Management, werden am Anfang in Analysephase durchgeführt. Dann wird das Angebot mit dem Preis erstellt.

  Alle späteren Änderungen wurden über Change-Requests eingebracht. Für jede Änderung wird geschätzt, welche Auswirkung sie auf Zeit, Kosten, Aufwand. Der Kunde konnte dann entscheiden, ob er die Änderung will oder nicht. Je nach Kosten der Änderung muss der Kunde dann mehr bezahlen.

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  In der ersten Phase hauptsächlich vom Kunden.

  Im Laufe des Projektes: 70% vom Kunden, 30% von uns. Bei der Realisierung haben wir gemerkt, dass sich nicht alle Features so umsetzen lassen, wie es der Kunde anfangs gedacht hat.

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  Das wurde von Ausführungsteam mit Hilfe von einfacher Exel-Liste verwaltet.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Wir. Die Anforderungen wurden Aufgenommen, dann inhaltlich beschrieben. Dann wurde diese Beschreibung vom Kunden abgenommen. Dann haben wir eine technische Beschreibung gemacht. Diese technische Beschreibung wurde dann an die Programmierer gegeben.

Das fachliche Team hat das gemacht. Für die technische Übersetzung hat das das technische Team gemacht.

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  Am Anfang wurde eine fachliche Anforderungssammlungsphase gemacht. Dort wurde gesammelt und Priorisiert und dann festgelegt, welche Anforderungen realisiert werden.

  Aus der Sicht des Realisierungsaufwandes haben sich ca. 15% bis 20% der Anforderungen geändert.

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  Bei den Features die nicht so umgesetzt werden konnten, ging es oft um Tradeoffs: Wir haben gesamt, wir können es so machen, aber dann dauert die Analysefunktion im System z.B. 2 Minuten. Da diese 2 Minuten den Performance-Anforderungen nicht genügen, mussten wir das mit dem Kunden besprechen und das Feature dann ändern.

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  -

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

  Der Aufwand wird über das Vorgehensmodell geschätzt. Es wird die Anzahl der Eingabemasken, Datenbankzugriffe usw. gezählt. Dann wird ein Testaufwand usw. hinzugegeben und so ergibt sich der Aufwand.

  Die Abhängigkeiten werden auch berücksichtigt. Eine bestimmte Anforderung kann nur realisiert werden, wenn auch bestimmte andere Anforderungen realisiert werden.

  Es werden dann Pakete angeboten. Das sind so eine Art „Ausstattungspakete“.

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  -

- ***Von wem kommen die Änderungen?***

  -

- ***Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?***

  Am Anfang wurde eine fachliche Dokumentation erstellt, in Word. Um dann wurden die Anforderungen so gesammelt, wie ich schon beschrieben hatte.

- ***Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?***

  -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Pool von Techniken. Für ein Projekt werden bestimmte Techniken vorgegeben.

  Oder der Kunde gibt die Methoden vor.

  Vom Kunden kamen 3 bis 4 Leute zu uns, die eine sehr genaue Vorstellung hatten, wie die zu entwickelnde Software aussehen soll. Mit denen Zusammen haben wir das System entwickelt.

  Es waren also: Workshops, strukturierte Interviews. Zuerst wurde das Problem in mehrere Teile strukturiert, und dann für die einzelnen Teile Workshops und Interviews durchgeführt. Die Ergebnisse wurden aufgeschrieben und dann nochmals mit dem Kunden durchgesprochen, um zu sehen, ob wir es richtig verstanden und aufgeschrieben haben.

  Bei größeren Projekten: strukturiertes Interview für einen ersten Eindruck und Grundverständnis. Dann detaillierte Workshops, um die Details zu verstehen.

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Unterschiedliche Terminologie zwischen fachlichen und technischen Bereichen.

  Es werden Leute benötigt, die die fachlichen Dinge gut genug verstehen und gleichzeitig auch technische Lösungen vorschlagen können.

  Diese Probleme kommen sehr oft vor. Im Umgang mit diesen Problemen sind wir als Beta aber besonders gut. Bei uns sind die fachlichen Leute gewohnt mit Technikern zusammenzuarbeiten. Auch die Techniker wissen wir man mit fachlichen Leuten sprechen muss, damit die es verstehen.

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  Die Anforderungen werden beschrieben: WAS soll passieren und WIE soll es passieren. Und die Zusammenhänge zwischen den Anforderungen werden beschrieben. Der Ablauf wird erstellt. Das Datenmodell wird erstellt, mit genauen Angaben von Wertebereichen usw. Das wird aber alles fachlich beschrieben.

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Bei der Definition der Anforderungen werden der Aufwand und der Nutzen hinzugegeben.

  Es wird auch immer die Zeit berücksichtigt. Beispiel: Euro-Einführung. Das System muss am 1.1.2002 funktionieren, weil sonst war's das mit der Bank.

  Rechtliche Anforderungen müssen immer eingehalten werden. Sonst wird das System nicht abgenommen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Die Realisierung wird normalerweise nach Aufwand gemacht. Das System wird zum Festpreis bestellt und dann werden so viele Anforderungen realisiert, wie es für diesen Preis möglich ist.

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  In einfachen Excel-Listen verwaltet.

  Dort wurde festgehalten: Von wem, was betrifft es, welcher Aufwand

  Es gibt auch eine Toolunterstützung für Anforderungen, so wie Projektmanagement-Tools. Bei dem Projekt haben wir das aber noch nicht eingesetzt.

- ***Welche Informationen werden dokumentiert?***

  Nummer, Name, Beschreibung, Kosten, Nutzen, Woher kommt sie(Quelle)

  Der Nutzen wird bei konsolidierten Paketen ermittelt. Für feingranulare Anforderungen kann der Nutzen nicht einzeln genannt werden

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  RequisitePro ist das Standardtool bei Beta. Das wurde erst 2007 eingesetzt.

  Doors wird oft von Kunden verwendet.

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

  Letztendlich immer der Kunde. Wenn der Kunde mehr Anforderungen haben will, als zum Festpreis möglich, dann muss er dafür auch bezahlen.

  Wenn eine Anforderung technisch nicht möglich ist bzw. nur mit besserer Hardware möglich ist, so muss der Kunde auch entscheiden, wie viel er zu zahlen bereit ist.

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Wenn die Implementierung fertig ist, dann müssen die Testfälle für den Abnahmetest alle laufen. Theoretisch müssen erst die Testfälle erstellt werden, dann starten das Programmierung, in Realität wird das parallel gemacht.

  Anhängig von Vertrag. Die Endabnahme(Fachliche Abnahme) macht der Kunde. Der Kunde testet und unterschreibt dann, dass wir das Geld bekommen. Die Abnahme Testfälle kommen von Kunden.

8. Prototypen

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  Schon vorher muss man genau festlegen, was man mit dem Prototypen erreichen will. Wichtig ist: schwierig zu erklärende Sachen visualisieren; Sachen mit großen Aufwand

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

zu besprechen; Sachen mit großen Risiko zu besprechen. Nicht Ziel ist: 100% der Anforderungen umzusetzen, denn dann habe ich schon die Anwendung.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  Hauptaufgabe von Prototypen ist Kommunikation mit Kunden.

  Bei fast allen Projekten werden Prototypen eingesetzt. Nur bei Systemen, bei denen es praktisch keine Endanwender gibt, werden ohne Prototyp gemacht; z.B. Zahlungsverkehrsabwicklung, regelbasierte Systeme usw.

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  Früher mit Basic, heute eher mit HTML, weil es schnell geht.

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  grob geschätzt: 20-25% vom Analyseaufwand. Dieser selbst ist so 15% bis 20%. Insgesamt sind's dann so 5%.

  Prototyping ist in den letzten 10 Jahren immer schneller geworden. Der Aufwand einen Prototyp zu erstellen, ist fast kein Aufwand mehr. Der Aufwand den Prototypen mit dem Kunden zu besprechen ist auch kleiner geworden.

  Die richtige Technologie für den Prototypen ist wichtig.

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  Mitte der 1990er haben die Kunden oft nicht verstanden, dass der Prototyp nur ein Prototyp ist. Die Kunden dachten dann, dass das Produkt schon fertig ist und sie dachten in 3 Tagen wäre das endgültige System fertig. In Wirklichkeit braucht man dann noch Jahre um das System fertig zu machen. Das dem Kunden klar zu machen war sehr schwierig.

  Wenn man mit jemandem Arbeitet, der das noch nie gemacht hat, dann gibt es das Problem auch noch. Aber heutzutage haben schon fast alle Leute mal mit einem Prototypen gearbeitet und wissen wie damit umzugehen ist.

  Gefahr zu viele Iterationen zu machen. Beispiel: Farbe rot => grün => nein, doch lieber gelb => doch lieber rosa.

  Weil man den Prototypen schnell ändern kann, beginnt man rumzuspielen. Und nachdem man viel Zeit verloren hat, macht man erst die Entscheidung.

  Man muss von vornherein sagen: Es gibt 2 Iterationen und dann trifft man die Entscheidung.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.3 Gamma

Datum: 24.06.2008, 29.09.2008

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	-
Mitarbeiter gesamt (weltweit)	-

2. Produkte des Unternehmens

- ***Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  IT im firmeninternen Bereich (betriebsintern Level 2)

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren:***

  - Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware
  - Kommerzielle Software vs. Open-Source-Software
  - Individualsoftware

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  Nur grobes Vorgehensmodell: Projektidee -> Konzeptphase -> Lastenheftphase -> Verifizierungsphase

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  Intern, Kunde sind hier die Fachbereiche (eigentlich zentrale konzernweite IT, momentan jedoch markenbezogen)

- ***Könnten Sie das Projekt kurz beschreiben?***

  Weiterentwicklungsprojekt im Bereich softwareseitiger Fahrzeugdiebstahlschutz für mehrere Automarken des VW Konzerns

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

  Elektronikentwicklung, Produktion, Handel, Abteilung für Aufgaben des Diebstahlschutzes, andere Automarken

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

  Ende des Initialprojektes war vor zwei Jahren, seitdem andauerndes Weiterentwicklungsprojekt.

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

  Auf der IT Seite etwa 20 Personen, etwa 76 Mann stark, jedoch nicht alle ins Projekt eingebunden

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

4. Requirements Engineering

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Es wird ein initiales Lastenheft (Baseline) erstellt, Hauptaugenmerk liegt jedoch auf Change Requests bedingt durch Verfeinerung der Anforderungen und somit auch bedingt durch ungenaue Erfassung von Prozessen/Anforderungen. Der gesamte RE Prozess ist auf Change Management ausgelegt (Projekt wird als Fleckenteppich betrachtet).

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Prozessmodell: Projektidee -> Konzeptphase -> Lastenheftphase -> Verifizierungsphase

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  Zum einen aus der IT (nicht funktionale Anforderungen, Schnittstellen, etc.), zum anderen aus den Fachbereichen (funktionale).

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  Zwei Produktmanager als Schnittstelle zum „Kunden“ und das Gremium Projektteam.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

  Entwickler (relativ spät im Prozess -> fehlerbehaftet),Fachabteilungen nur bedingt beteiligt.

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  Etwa 60% des Gesamtumfangs sind durch Weiterentwicklung entstanden, 40% standen initial fest.

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  Je nach Aufwand und damit Budget, Priorisierung der Anforderungen.

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  Funktionale Anforderungen sind recht früh bekannt, nicht funktionale Anforderungen entstehen meist im Entwicklungsprozess.

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

  Nur Antwort zu Frage 4.6 (Anfangsmenge der Anforderungen) genannt.

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Es gibt Change Requests, aber speziell im späteren Projektverlauf ist die Dokumentierung absolut mangelhaft bzw. nicht mehr vorhanden. Das Team entscheidet, ob ein Change ein eigenes Lastenheft bekommt und somit als Art eigenes Projekt behandelt wird.

- ***Von wem kommen die Änderungen?***

  Zu etwa 80% aus den Fachbereichen, zu 20% aus der IT.

- ***Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?***

  Ja, in den entsprechenden Prozessphasen (siehe Prozessbeschreibung), allerdings besteht hier enormes Verbesserungspotenzial

- ***Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?***

  Teilweise eigenes Lastenheft für einzelne Anforderungen (je nach Mächtigkeit). In der Vergangenheit wurde diese dann zu einem einzigen Lastenheft gebündelt, was sich allerdings als sehr unglücklich herausstellte, da die Inhalte teils sehr konträr waren (andere Perspektiven auf Anforderungen). Zusammengehörige Anforderungen waren ohne Reihenfolge eingebunden -> Verbesserung zu integriertem Systemlastenheft angestrebt.

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Workshops, Besprechung von Änderungen/Neuanforderungen in Projektteamsitzung

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Siehe Konflikte: Hauptsächlich schlechte und unqualifizierte Kommunikation

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  Verfeinerung durch Entwickler, teilweise mit Produktmanagern (wenn speziell und komplex).

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Priorisierung der komplexen Anforderungen im Projektteam (zu welchem Release, Kosten) mit Nummerierung

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  Es gibt Templates vom VW Konzern. Anforderungen werden hier beschrieben und falls nötig soweit nachformalisiert, dass sie von beiden Seiten eindeutig verstanden werden.

  Bedingt durch die Projektstruktur („Fleckenteppich“-Vergleich) gibt es jedoch keine durchgängige Gesamtdokumentation, sondern viele Einzeldokumentationen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Welche Informationen werden dokumentiert?***

  Nur Verweis auf Lastenheft (Realisierungskonzepte).

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  Word, Excel, Powerpoint (Präsentationen vor Gremium), Mercury TestDirector, HP Quality Center, eigenes Budgetierungstool. Unterschiedliche Fachbereiche nutzen unterschiedliche Tools (Elektronikentwicklung beispielsweise Doors, das aus Sicherheitsgründen nicht von der Projekt-IT genutzt werden kann).

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

  Abhängig von den Kosten: Unter 20.000 Euro pro Einzelmaßnahme ist das operative Team autark (interne Priorisierung und Entscheidung), darüber liegt die Verantwortlichkeit bei einem Entscheiderkreis.

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Interne Entwicklung überprüft, ob die mit externen Dienstleistern abgeschlossenen Werkverträge eingehalten wurden; Live-Vortests mit den Kunden; TestDirector Tool; eventuell (bei Bedarf) noch Testfallgenerierung durch Testcenter

8. Prototypen

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  Keine Nutzung von Prototypen für Weiterentwicklungen, sondern nur für Neuentwicklungen (Mockups), prototypartige Livetests

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  Fachabteilung.

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  Falls funktionaler Prototyp, dann so wie die Applikation selbst. Der Prototyp dient als Basis zur Weiterentwicklung.

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  Prototypen nur bei nicht-komplexen Gebieten sinnvoll, da sonst kein Mehrwert gegenüber der normalen Entwicklung (zu aufwendig). Hier reichen PPT-Präsentationen zur Übersicht.

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.4 Delta

Datum: 23.06.2008

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	56.018 Mio.€
Mitarbeiter gesamt (weltweit)	107.539

2. Produkte des Unternehmens

- ***Welche Software-Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  Keine Software-Produkte, nur internes SAP Anforderungsmanagement (Einbindung von SAP Anwendungen): Delta- bzw. Automobilspezifische Prozesse werden von Delta bei SAP eingebracht, sodass SAP diese Prozesse in seine Standardsoftware integrieren kann (zentral für alle Projekte von Delta)

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren:***

  - Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware
  - Kommerzielle Software vs. Open-Source-Software

  Keine Produkte, aber Anforderungsmanagement bzgl. kommerzieller Standardsoftware bzw. bzgl. nötiger Eigenentwicklungen

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  Bisheriges Modell:

  3 Stufen: Initialphase, Grobkonzept, Feinkonzept (Fachkonzept und IT-Konzept mit Lösung)

  Ablauf: Ermittlung der Anforderungen durch Deltas COC Bereich, Übergabe an SAP (Einsteuerung in SAP Clearing House Prozess), Entscheidung über Umsetzung bei SAP, dann Einordnung in eine der vier folgenden Kategorien:

  - SAP wird diese Anforderung übernehmen
  - Anforderung ist interessant für SAP, bisher nicht in Entwicklung, eventuell bald
  - nicht von Interesse für SAP, eigene Umsetzung erforderlich, SAP behält sich vor die von Unternehmen vorgeschlagene Lösung weg zu werfen oder zu übernehmen.
  - Delta stuft diese Anforderungen und Prozesse als vertraulich (internes Know-How) ein, somit kommt nur eine Eigenentwicklung in Frage

  Danach eventuelle Vereinbarung mit SAP (Kosten, Verantwortlichkeiten etc.), dem folgend Verfeinerung der Anforderungen

  Der gesamte Prozess ist stark iterativ, da Prozesse in den Feinplanung eventuell anders ablaufen, als in der Grobplanung angenommen

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anforderungen sind stets an einen Business Case geknüpft, der diese Anforderung rechtfertigt (vor allem in Kategorie 3 muss ein Business Case den Nutzen einer Investition für das Unternehmen untermauern)

Diese Model ist zu aufwändig vor allem zeitlich, verlangt viele Kapazitäten der Delta Seite.

Neues Modell:

Modularisiertes Anforderungsmanagement für einzelne Produkte (der Einteilung von SAP folgend)

Projekte/Fachbereiche senden an zentrale Stelle (COC), diese an SAP. Dann wird pragmatisch ohne Iterationsprozess entschieden, was zu tun ist.

Früher deutlich aufwendiger und schwerer, dafür aber genauere Übersicht über umgesetzte Anforderungen

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  Intern

- ***Könnten Sie das Projekt kurz beschreiben?***

  Anforderungsmanagement wie oben beschrieben als interner Prozess

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

  Nur Delta (Fachabteilung, Fachbereichs-IT und zentrale IT) und SAP.

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

  Anfang des Jahres bis Juli, also 6 Monate für etwas kleineres Projekt (Umsetzung folgt und ist nicht mehr Teil des Projekts); etwa 200 Anforderungen

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

  Etwa 15 Leute.

4. Requirements Engineering

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Initiale Erfassung, aber kontinuierlich Verwaltung mit bedarfsweiser Neuerfassung.

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Erhebung, Interpretation, Strukturierung: Siehe Prozessbeschreibung des Vorgehensmodells von Frage 2.3

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Jede Fachabteilung hat einen eigenen kleinen COC-Bereich, dieser nimmt Anforderungen entgegen und leitet sie an SAP weiter. Oder Anforderung von IT-Mitarbeiter der Fachabteilung per OSS-Ticket direkt an SAP.

Anforderungen aus den drei Bereichen Projekte/Betrieb/Wartung.

Workshops nur bei Projekten.

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  Früher Delta zentral durch Excel-Tracking, jetzt dezentral auch durch Excel. Ohne zusätzliche Tools.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

Erfolgt meist in Kooperation: Fachbereich, Fachbereichs-IT, SAP.

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  Durch die Verfeinerung von Anforderungen ergeben sich oft Inkonsistenzen, die Änderungen und Anpassungen bedingen. Bei den Kategorien 1&2 ist der Teil der angepassten Anforderungen am Projektende höher als der Initialteil der Anforderungen, aber auch bei 3&4 (Delta-spezifische Prozesse, genauer bekannt, daher hier weniger Änderungsbedarf).

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  Durch Verfeinerung bedingte Anforderungen werden neu erfasst bzw. angepasst (iterativer Prozess, obwohl das neue Vorgehensmodell Iterationen vermeidet).

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  Kategorisierung erfolgt von Projekt zu Projekt, aber keine feste Katalogisierung.

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

  Erneute Iteration, keine festen Werte genannt.

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  Siehe Vorfragen.

- ***Von wem kommen die Änderungen?***

  Meist durch Verfeinerung der Anforderungen und Erkenntnis über ungenau/falsch erfasste Prozesse, die diese Anforderungen bedingen.

- ***Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?***

  -

- ***Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?***

  -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Keine proaktive Suche nach Anforderungen.

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Aufwand: Viele Stellen und Absprachen liegen im Prozess, viel Zeit vergeht, abhängig von vielen Unbekannten, geringe Erfolgsquote bei Standardisierungsprozess der Anforderungen

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  Geht nicht ohne Iterationen. Grobkonzept erfolgt möglichst am konkreten System, nicht nur rohe Beschreibungen. IT-Konzept wird ohne Fachbereich von der IT geschrieben.

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Stufen 1 (MUST have!) bis 6 (nice to have); kein standardisiertes Vorgehen; individuelle Kategorisierung.

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  Es gibt Templates, aber Dokumentation den Projekten freigestellt.

- ***Welche Informationen werden dokumentiert?***

  Thema, Ansprechpartner, Release, Autor, Beschreibung, Reifegrad der Anforderung, Standardfähigkeit, Termin, Nutzen (Business Case), Priorität

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  Auf Dauer lässt sich nicht alles mit Excel tracken, deshalb gibt es Anstrebenden, ein zentrales Tool zu schaffen.

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

  Höherer IT-Manager entscheidet bei Delta über Umsetzungen in den Anforderungskategorien 3&4, da so viele unnötige Anforderungen so gefiltert werden (durch den Manager selbst, durch das Hemmnis des langen Entscheidungswegs sowie durch die Notwendigkeit eines Business Cases). Bei 1&2 entscheidet SAP.

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Zusammenkunft beider Parteien, Abhaken der Anforderungen (Validierung). Bei altem System war die zahlmäßige Überprüfung der Anforderungsumsetzung durch SAP

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

leichter. Generell liegt die Validierung in der Verantwortlichkeit der Fachbereiche. Leider werden einige Eigenentwicklungen nicht genutzt (Veralterung von Anforderungen, lange Prozessdauer), genaue Zahlen liegen jedoch Delta-intern nicht vor.

8. Prototypen

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  „Arbeit am System“ wichtig für Fachabteilungen, zur besseren Einschätzung der genauen Anforderungen als Entscheidungsgrundlage. Meist Standard SAP Mockups und Demos, keine eigens entwickelten Prototypen.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  Anforderungssteller (Fachabteilungen)

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  Bei echte Prototypen meist vertikale Implementierung, Prototyp als funktionaler Teil der Lösung.

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  Wenn Prototyping für Anforderungen der Priorität 1, dann sind Kosten kein Entscheidungsfaktor. Prototypen sind für Prio1 Anforderungen meist elementar. Um Kosten zu sparen arbeitet man inkrementell die Prototypen zu Lösungen aus.

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.5 Epsilon

Datum: 30.09.2008

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	56 .000
Mitarbeiter gesamt (weltweit)	107.000

2. Produkte des Unternehmens

- ***Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  Allgemein: Services ums Auto, Bankservices

  Abteilung: firmeninternes Software Engineering („IT für IT"); Erstellung und Anpassung von Individuallösungen

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren:***

  - Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware
  - Kommerzielle Software vs. Open-Source-Software

  Kommerzielle Individualsoftware, bedingt auch Verwaltung von Standardsoftware

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  Vor Projektstart werden die Anforderungen durch die Fachabteilung festgelegt; der Entwicklungsprozess umfasst nur die reine Implementierung; nach dieser wird die Software als Leistungsstufe 1 an die Abteilung übergeben

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  immer intern

- ***Könnten Sie das Projekt kurz beschreiben?***

  kein bestimmtes Projekt besprochen

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

  oft externe Dienstleister

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

  kein bestimmtes Projekt besprochen

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

  kein bestimmtes Projekt besprochen

4. Requirements Engineering

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Anforderungen werden vor Projektstart in der Fachabteilung erfasst und danach übergeben; Änderungen werden nach Leistungsstufe 1 über automatisches Änderungsmanagement verwaltet; davor werden Änderungen projektabhängig verwaltet

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Anforderungserhebung und –Übersetzung sind Aufgabe der Fachabteilung. Die Anforderungen werden in Fachabteilungen mittels Excel-Listen verwaltet.

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  Wichtige Stakeholder, z.B. Fachabteilung (sowohl initial, als auch Änderungen)

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  Das Projekt.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

  der Fachprozess muss Anforderungen so formulieren, dass die IT sie umsetzen kann (nötigenfalls wird die Anforderungen zu Neu-Formulierung zurückgeschickt)

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  nicht abschätzbar, aber tendenziell hohe Zahlen

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  Sie werden gesammelt und in der nächsten Releasestufe umgesetzt.

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  -

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

  nicht abschätzbar, aber tendenziell hohe Zahlen

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  Automatische Abwicklung über Tools für Änderungsmanagement nach Leistungsstufe 1 (nach erster Einführung), bis dahin projektinterne Änderungsverwaltung (nicht unbedingt toolbasiert)

- ***Von wem kommen die Änderungen?***

  Aus der Fachabteilung.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?***

  Ja, Fachkonzept und DV-Konzept

- ***Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?***

  standardmäßig, Grobkonzept und verfeinerte IT-bezogene Ausarbeitung

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Meetings Manchmal Moderation durch erfahrene Projektleiter

  Erfassung abteilungsabhängig in Excel bzw. Word oder mit Tools

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Mangelnde Präzisierung von Anforderungen, nicht klare Zusammenhänge also mangelnde Informationsintegrität durch schlechte Kommunikation

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  Durch die Fachabteilungen, gegebenenfalls im Dialog mit dem Projektteam

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Priorisierung entlang der Leistungsstufe (welche Features müssen in nächster Leistungsstufe enthalten sein?)

  Bewertung durch Fachabteilung und IT (schwerpunktmäßig) nach standardmäßigen Kriterien wie Kosten, Komplexität etc.

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  Fachkonzept und DV-Konzept (letzteres nicht immer, da oft externer Einkauf der Lösung)

  strenger Leitfaden und Templates als Vorlage

- ***Welche Informationen werden dokumentiert?***

  in anderem Zusammenhang konkret nur Ersteller und Priorisierung genannt, ansonsten getreu der Fachkonzept- und DV-Konzept-Leitfäden

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  Dokumentation nur über Word

  Allgemeine Verwaltung über Tools ist den Projektleitern freigestellt (nur ITPM ist BMW-weite Vorgabe)

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

  Solange die Finanzierung stimmt, ist das Sache der Fachabteilungen

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Idealtypischerweise werden Validierungskriterien mit den Anforderungen von der Fachabteilung spezifiziert, in der Praxis jedoch selten

  Abnahme durch Fachabteilung nach eigenen und projektindividuellen Kriterien

8. Prototypen

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  Für die Kundenkommunikation oder Testen von Systemen, allgemein kaum Prototyping.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  Kunden

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  -

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  Leider meist zu hoher Aufwand. Oft nur bei Wartungsprojekten möglich, da hier bereits eine Basis gegeben ist, auf der man leicht aufsetzen kann.

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  Negativ ist die Gefahr der funktionalen Weiterentwicklung eines (nicht perfekten) Prototyps als Ausgangsbasis für das Endprodukt

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.6 Zeta

Datum: 20.08.2008

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	180 Mio €
Mitarbeiter gesamt	1000

2. Produkte des Unternehmens

- ***Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  Das Unternehmen bietet Produkte aus dem Bereich der bildunterstützten Chirurgie an. Die Dienstleistungen sind im Wesentlichen die Service-Verträge, zum Beispiel, die Wartung der Geräte. Die Dienstleistungen werden zusammen mit dem Produkt angeboten. Der Kunden kann entscheiden, in wie fern er den Service-Vertrag dazu nimmt.

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren:***

  - Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware
  - Kommerzielle Software vs. Open-Source-Software

  Die Produkte bestehen aus Produkt (Hardware), Software und Dienstleistungen.

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  Der Entwicklungsprozess basiert auf dem V-Modell. Es gibt ein Project-Support-Office Team, das eine Unterstützung des Projektmanagements im Hause leistet. Die Projektleiter werden durch das Team unterstützt. Das RE (Tools, Vorgehensweise) wird von diesem Team definiert und im weiteren Verlauf unterstützt. Es gibt standardisierte Tools und Vorgehensweisen, die abhängig vom Projekt eingesetzt werden. Die Komplexität, die Schnittstellen und weitere Aspekte des Projektes bestimmen, welche Konzepte eingesetzt werden.

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  Das Projekt ist intern entstanden.

- ***Könnten Sie das Projekt kurz beschreiben?***

  Es geht um die Navigation am Hüft-Gelenk. Dabei wird die Hüft-Chirurgie unterstützt. Der Chirurg kann die Hüft-Implantate in die richtige Position bringen. Die Hardware-Komponente ist bei vielen Applikationen gleich, die Software wird vom zu erstellenden Produkt bestimmt. Es gibt viele Arten der Navigation. Die Software ist entscheiden für das Produkt.

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Es gibt keine weiteren Kooperationspartner. Es gibt weitere Schnittstellen im Marketing-Bereich. Es gab auch Partner, die die Vorgehensweise bei der Spezifikation von Anforderungen aus dem Fachbereich unterstützt haben. Die Entwicklung wurde aber nicht outgesourced.

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

  Das Projekt hat 1,5 Jahre gedauert.

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

  Es sind Software-Entwickler und Projekt-Ingenieure. Die an dem Projekt beteiligten Teams hatten Software-Entwickler, Projektengeniere, Hardware-Entwickler, Softwaremanager. Projektengeniere sind sehr nah an der Entwicklung der Applikation. 8 Personen waren am Projekt beteiligt. Es gab einen obersten Projektleiter, weitere Teilprojektleiter, Softwareprojektmanager usw. Und es gab einen Gesamtverantwortlichen.

4. Requirements Engineering

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Die Anforderungen werden kontinuierlich verwaltet. Es gibt einen Lenkungsausschuss, der die Verwaltung von Anforderungen koordiniert.

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Der Prozess der Verwaltung von Anforderungen liegt nah an der Beschreibung des Prozesses des Anforderungsmanagements in der Literatur. Die Anforderungen werden erst von verschiedenen Stakeholdergruppen erhoben, in Spezifikationsdokument erstellt dann dem Lenkungsausschuss vorgelegt, falls die Änderungen im Laufe des Projektes ergeben dann wird Lenkungsausschuss darüber entscheiden.

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  In der Regel kommen die Anforderungen aus den Bereichen Service, Vertrieb, Kunden, interne Angelegenheiten. der Projektleiter definiert bzw. sammelt die Anforderungen. Die Herkunft der Anforderung spielt im Moment keine Rolle für die Behandlung von dieser Anforderung.

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  Der Projektleiter ist dafür zuständig, die Anforderungen zu definieren, sie zu bewerten usw. Er macht die Verwaltung. Die Größe des Projektes bestimmt, wer bzw. wie viele Personen an der Verwaltung von Anforderungen beteiligt sind.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

  Der Projektleiter ist dafür zuständig, die Vorstellungen in die Anforderungen zu übersetzen. Es gibt ein Tool (R-Tool), in das die Anforderung eingegeben wird. Wenn

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

diese Anforderung weitergeleitet wird, zum Beispiel, an den Softwareentwickler, dann wird diese Anforderung wieder toolunterstützt übersetzt.

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  Von den Initialanforderungen bleiben ca. 40% der Anforderungen übrig.

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  Diese Anforderungen werden aufgenommen. Der Projektleiter soll aber die Konsequenzen davon abwägen. Der Projektleiter soll dann entscheiden, ob die Anforderungen umgesetzt werden.

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  Funktionalität.

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

  Projektspezifisch. In dem betrachteten Projekt war der Aufwand sehr groß.

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  Der Projektleiter ist dafür zuständig, die Änderungen an Anforderungen entsprechend zu dokumentieren. Der Lenkungsausschuss entscheidet über die Änderungen. Der Projektleiter ist dafür zuständig, die Änderungen zur Realisierung freizugeben. Der Lenkungsausschuss entscheidet über die Änderungen.

  Der Lenkungsausschuss tagt regelmäßig. Die geänderten und dazugekommenen Anforderungen werden gesammelt und dem Lenkungsausschuss präsentiert. Die Mitglieder des Lenkungsausschusses sind nicht vordefiniert, sonder es wird eine Entscheidung projektabhängig getroffen. Der Projektleister und die Geschäftsleitung entscheiden, wer zum Lenkungsausschuss gehören soll, es können auch Externe dabei sein.

- ***Von wem kommen die Änderungen?***

  Die Änderungen in dem Projekt sind von extern gekommen. Allgemein auch von der Geschäftsführung. Viele Projekte laufen parallel und haben enge Zusammenhänge. Die Änderung an einem Projekt führt zu den Änderungen in weiteren Projekten.

- ***Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?***

  Es wird nicht genau unterschieden.

  Es gibt aber ein Word-Dokument, in dem die Anforderungen in Fließtext beschrieben sind.

  Im R-Tool ist auch schon ein Teil der Realisierung beschrieben. Dort werden die Anforderungen formaler beschrieben.

- ***Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?***

  -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Die Anforderungen werden erhoben und in einem Spezifikationsdokument aufgeschrieben. Der Projektleiter initiiert Workshops und befragt zu den Anforderungen. Dabei werden die relevanten Stakeholder (meist andere Abteilungen) und die Geschäftsleitung gefragt. Zum Beispiel, man sieht den Bedarf für eine Folgeentwicklung. Dann werden die aktuellen Stakeholder interviewt und befragt wie Service-Abteilungen. Der Medizinproduktbereich ist eine wichtige Quelle für die Anforderungen. Das sind offizielle Anforderungen, die auf jeden Fall berücksichtigt werden müssen. Zum Beispiel werden die Kundenreklamationen im Rahmen des Qualitätsmanagements zentral verwaltet. Die gehen als Anforderungen in die Folgeversion des Produktes ein.

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Bewertung der Priorität anhand der Projektzeit, -kosten usw.

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  Nachdem die Anforderungen aufgenommen werden, werden sie in das R-Tool eingetragen. Das ist so zu sagen eine Datenbank, die die ausformulierten Anforderungen enthält. So eine Anforderung wird in natürlicher Sprache beschrieben. Es gibt aber keine genauen Vorgaben, was sie enthalten soll. Das Tool gibt schon einiges vor, welche Angaben gemacht werden sollen.

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Die Priorisierung kann auch im Tool vorgenommen werden. Der Projektleiter gibt an, welche Priorisierung er für nötig hält. Die Priorisierung erfolgt nach klassischem Schema: muss / soll / kann.

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  Nachdem die Anforderungen von verschiedenen Stakeholdern erhoben werden, werden sie in einem Spezifikationsdokument festgelegt. Dieses Dokument wird dem Lenkungsausschuss vorgelegt. Das ist auch der Projektstart.

  Das R-Tools kann auch als Dokumentation angesehen werden.

- ***Welche Informationen werden dokumentiert?***

  Es gibt verschiedene Arten von Spezifikationen: z.B. GUI-Spezifikation, oder Schulungsdokumentation.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

In der Dokumentation gibt es einerseits die reinen Anforderungen, andererseits auch technischere Spezifikationen, wo auch auf die Komponenten des Systems eingegangen wird.

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  Es gibt das R-Tool in den die Anforderungen festgehalten werden. Das Tool ist gleichzeitig die Dokumentation des Produkts. Denn in der Medizintechnik gibt es genaue Vorschriften, wie die Dokumentation inklusive Unterschriften und Freigaben aussehen muss. Es wird natürlichsprachlich Dokumentiert.

  Es gibt auch das X-Tool, das entwicklungsnäher ist. In diesem Tool kommen auch natürlichsprachliche Dokumentationen vor. Einzelne Projektleiter können auch UML, MindMap usw. verwenden, es gibt aber keine Vorschrift.

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

  Der Projektleiter entscheidet dies.

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Das sind sehr wichtige Aspekte. Es geht um medizinische Geräte, deswegen ist es besonders wichtig, zu überprüfen, ob die entwickelten Konzepte mit den Anforderungen übereinstimmen. Das erfolgt in einer Testphase.

  Im R-Tool wird zu jeder Anforderung ein Testfall spezifiziert. Dieser muss dann überprüft werden. Der Testfall wird abhängig von der Anforderung auch natürlichsprachlich definiert.

8. Prototypen

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  Prototypen werden eingesetzt, um Konflikte zu lösen. Sie dienen zur Kommunikation und um den Leuten was Konkretes zu zeigen. Es wird eine spätere Funktion des Projektes plastisch dargestellt.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  -

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  Es wird ein Softwareprototyp erstellt. Der wird in der Sprache programmiert, in der auch normal Programmiert wird. Es sind teilweise inkrementelle Prototypen, teilweise nicht. Für Hardwareentwicklung werden Kunststoffmodelle benutzt.

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  Software: 30% vom Projektaufwand

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Hardware: Dort ist es etwas schwieriger, weil für Prototypen mit externen Firmen gearbeitet werden muss.

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  Dagegen: Es verbraucht Ressourcen.

  Dafür: Man plastisch darstellen, was man macht. Dadurch ist Diskussion besser möglich. Verifikation.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.7 Eta

Datum: 05.09.2008

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	286 Mio €
Mitarbeiter gesamt	1800

2. Produkte des Unternehmens

- ***Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  Nur Dienstleistungen: Individualsoftware und Anpassung von Standardsoftware (vor allem SAP).

  Auch Businessprozessberatung im Automotive Bereich.

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren: Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware; Kommerzielle Software vs. Open-Source-Software***

  Nur Individualsoftware.

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  -

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  Seit 2001 beraten wir ein Produkt bei einem Kunden, das in verschiedenen Märkten ausgebaut wird. Das Produkt wird in einem internationalen Umfeld eingesetzt. Das Projekt ist auf Wunsch der Kunden entstanden.

- ***Könnten Sie das Projekt kurz beschreiben?***

  Die Hauptentwicklung lief zwischen 2001 und 2003. Seither läuft nur mehr die Wartung. Es handelt sich um ein Verkaufssystem.

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

  Kunde: Internationales Unternehmen mit Niederlassungen in vielen Ländern (genannt Märkte)

  Lieferant: Umsetzung und Implementierung

  Märkte: Niederlassungen des Kunden in verschiedenen Ländern. 18

[18] Wichtig für das Verständnis des restlichen Interviews.
Als Markt werden alle Niederlassungen des Kunden in einem Land bezeichnet.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Ich bin die Qualitätsmanagerin dieses Projekts.

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

  Die Hauptentwicklung hat 2 Jahre gedauert. Zurzeit läuft die Wartung und Weiterentwicklung und das Rollout in neuen Märkten.

  Phasen des Projektes:

  - Entwicklung Leistungsstufe 1
  - Probephase
  - Rollout an die einzelnen Märkte

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

  Anzahl Personen:

  - Kunde: 6 Mitarbeiter
  - Lieferanten(Softwareentwicklung): 6 bis 7 Personen
  - Märkte (ein Markt ist immer ein Land): pro Markt 4 Personen. Zurzeit gibt es bereits 17 Märkte.

  Rollen:

  - Teamleiter
  - Release-Manager für jedes Release (genannt Releaseowner)
  - Projektleiter für jedes Rollout an eine Filiale
  - Anforderungsmanager (genannt Markt-Manager): Macht das Anforderungsmanagement für einen Markt

4. Requirements Engineering

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Beim ursprünglichen Projekt wurden die Anforderungen einmal erhoben. Dort wurde ein Grobkonzept geschrieben, das die Business-Anforderungen enthält. Anschließend wurde ein Fachkonzept und ein IT-Konzept erstellt. Aufgrund des IT-Konzepts wird entwickelt.

  Dazu gibt es einen Change-Management-Prozess mit Change-Requests. Es gibt zwei Change-Requests:

  - Neue Features
  - Fehler im Fachkonzept ausbessern

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Gesamtprozess der Entwicklung: Grobkonzept, Fachkonzept, IT-Konzept, Realisierung & Implementierung, Integration, Abnahme, Probephase in zwei Pilotmärkten

  Erste Leistungsstufe: Standalone bei jedem Händler

  Zweite Leistungsstufe: Hauptschnittstelle zum Hauptsystem. Jeder Händler sieht dieselben Kundendaten

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Der RE-Prozess wird wie folgt durchgeführt:

Bei neuer Anforderung wird die Anforderung aufgeschrieben. Sie wird dann von der Zentrale interpretiert und umformuliert. Dann wird der Aufwand evaluiert, den die Anforderung verursacht. Dann wird entschieden, ob die Anforderung umgesetzt wird. Dann wird eine Deadline für das nächste Release festgelegt. Bis zu dieser Deadline müssen alle neuen Anforderungen verständlich aufgeschrieben werden. Dann werden alle neuen Anforderungen priorisiert. Die Anforderungen die mit gesetzlichen Anforderungen zu tun haben müssen auf jeden Fall umgesetzt werden. Die anderen Anforderungen werden je nach Wichtigkeit und Kosten umgesetzt oder nicht. In einem Abstimmmeeting wird das alles entschieden. Eine Grundabstimmung mit den Märkten wird aber schon vorher durchgeführt.

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  Die Änderungen können von überall kommen: alle Anwender, Management, Lieferanten, strategische Entscheidungen, Wartung.

  Bei der Weiterentwicklung bekommen alle Märkte jährlich ein neues weiterentwickeltes Release.

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  Zentralstelle von jeweiligen Ländern.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

  -

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  Die ursprünglichen fachlichen Komponenten haben sich nach der ersten Festlegung nicht mehr verändert. Auch die Architektur hat sich nicht mehr verändert. Es war keine prototypische und keine inkrementelle Entwicklung.

  Während der Wartung und des Rollouts in neuen Märkten werden für jeden Markt Anforderungen gesammelt. Sehr große Änderungen am System können jedoch nicht für einen Markt gemacht werden, weil bei einem neuen Release alle Märkte dasselbe System bekommen.

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  Die Märkte führen eine Vorpriorisierung der Anforderungen durch. Jeder Markt muss seine Änderungen, die er will selbst bezahlen, deshalb wird auch gut priorisiert.

  Die vorpriorisierten Anforderungen werden zentral gesammelt und in einer Runde wird entschieden, welche umgesetzt werden und welche nicht.

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  Vor allem Funktionale Anforderungen kommen neu dazu.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Große Probleme haben aber auch technische Ursachen gehabt. Es wird dann nicht genau zwischen technischen und Qualitätsanforderungen unterschieden.

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

  Während der Entwicklung: Relativ viele Change-Requests. Projektphase abhängig, in früheren Phasen viel Änderungen und Aufwand.

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  Alle Änderungen von Anforderungen werden Dokumentiert. Es gibt eine vollständige Nachvollziehbarkeit zwischen Anforderungen.

- ***Von wem kommen die Änderungen?***

  Von Überall: Kunden, Lieferanten usw.

  Da bei diesem System in den letzten 6 Jahren nur mehr Wartung und Weiterentwicklung gemacht wird, sind Anforderungen und Änderungen größtenteils das selbe.

- ***Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?***

  Am Anfang der Entwicklung wurden ein Grobkonzept, dann ein Feinkonzept und dann ein IT-Konzept erstellt.

- ***Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?***

  -

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Bei der Einführung in neuen Märkten: Workshop wo das bestehende Produkt diskutiert wird. Es werden auch Interviews durchgeführt. Da es aber bereits 17 Märkte gibt, kann ein Markt nicht zu große Änderungen fordern, denn alle Märkte bekommen dann dasselbe geänderte Produkt.

  Wir haben aber keinen direkten Zugriff auf die Endanwender. Die Endanwender kommunizieren mit der Hauptniederlassung im jeweiligen Markt. Wir kommunizieren dann nur mit den RE-Verantwortlichen der Hauptniederlassung.

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Schlampige Anforderung, die man nicht richtig versteht. Bei so einer Anforderung kennt man den Hintergrund nicht und man versteht nicht was sie wollen.

  Ein großes Problem ist, dass ein Marktverantwortlicher eine Anforderung unbedingt durchsetzten will, welche sich aber auf Grund der Architektur nicht umsetzten lässt. So eine Anforderung zu erkennen ist sehr schwer.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  Persönliche Kommunikationsfähigkeit der Person die das durchführt. In der Zentrale und den Märkten müssen Personen ausgewählt werden, die sich gut verstehen. Dabei treten auch kulturelle Probleme auf.

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Die Märkte führen eine Vorpriorisierung durch. Jeder Markt muss seine Änderungen, die er will selbst bezahlen, deshalb wird auch gut priorisiert.

  Anforderungen, die auf geänderten rechtlichen Rahmenbedingungen zurückgehen, werden immer umgesetzt. Bei den Anderen wird diskutiert.

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  QualityCenter als durchgängiges Werkzeug zur Dokumentation der Anforderungen. Alle Märkte müssen dieses System einsetzen.

  Alle Änderungen an Anforderungen werden dokumentiert. Die Anforderungen sind komplett Nachvollziehbar.

  Das Grob- und Fachkonzept wurde noch in Word geschrieben. Die Verwaltung geschieht jetzt in QualityCenter. Das Fachkonzept wird aber in Word aktuell gehalten. In nächster Zeit werden wir das Fachkonzept komplett in QualityCenter übernehmen. Dann werden wir kein Word mehr verwenden. Am Anfang wird jedoch ein Fachkonzept geschrieben werden müssen, für die Gestaltung des Vertrags.

- ***Welche Informationen werden dokumentiert?***

  Nummer, Text, Priorität, zugeordnetes Release, Testfälle, Quelle – Ansprechpartner (wer bestellt hat), Fehler, falls in Abnahmetest sowie Änderungen an der Anforderung vorgenommen wurden.

  Zurzeit ist im System noch eine flache Liste von Anforderungen. Wir werden jedoch jetzt eine hierarchische Liste von Anforderungen einführen. Dabei sollen einzelnen Komponenten mehrere Anforderungen zugeordnet werden.

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  QualityCenter als durchgängiges Werkzeug zur Dokumentation der Anforderungen. Alle Märkte müssen dieses System einsetzen und Word.

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

  Aufgrund der Vorpriorisierung der Anforderungssteller entscheidet folgende Runde: Releaseowner (hat den Überblick über alle Anforderungen), Team-Leiter (Budget), Kundenverantwortlicher, Lieferant (was ist Realisierbar). Die Entscheidung wird protokolliert und an die jeweilige Länder zurückgegebne, wobei die Gelegenheit haben diese abdiskutieren.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Testfälle werden in den jeweiligen Märkten erstellt. Der Releaseowner ist für das Schreiben der Testfälle verantwortlich. Er muss sich darum kümmern, dass die Testfälle geschrieben werden. Marktverantwortlicher ist eher für Test der technischen Anforderungen verantwortlich.

  Die Überprüfung wird in einem Gateway-Meeting durchgeführt: Ich als Qualitätsmanagerin des Projekts sitze diesem Gateway vor. Im Gateway-Meeting wird eine Checkliste eingesetzt, welche ich zusammen mit den Releaseownern vorbereite. Diese Checkliste wird an alle Märkte, alle Lieferanten, alle Schnittstellenbetroffenen geschickt.

  Beispiele für Fragen in der Checkliste:

  - Sind alle Anforderungen, so wie sie gedacht waren, umgesetzt?
  - Wurden alle Anforderungen ausreichend getestet?

  Das eigentliche Testen des Systems wird von den einzelnen Märkten durchgeführt. Das Gateway-Meeting fasst dann die Ergebnisse zusammen.

  Im Meeting sind folgende Rollen vertreten: alle Markt-Manager. Im Meeting wird die Checkliste Schritt für Schritt durchgegangen und über Probleme diskutiert. Für jeden Punkt gibt es eine Ampelwertung (grün = keine großen Probleme, gelb = kritische Maßnahmen, die Fehler müssen behoben werden, bevor das System eingesetzt werden kann, rot = große Änderungen, es wird ein weiteres Gateway-Meeting angesetzt). Wenn nicht entschieden werden kann, gibt es eine Eskalation. Dann müssen der Vorgesetzte des Marktowners und der Vorgesetzte des Teamleiters entscheiden.

8. Prototypen

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  In der Weiterentwicklung werden keine Prototypen eingesetzt. Auch bei neuen Rollouts an neue Märkte gibt es keine Prototypen.

  Am Anfang der Entwicklung wurde ein Prototyp gemacht. Der Prototyp war nur dazu da um die UI zu gestalten. Um die Funktionalität des Systems festzulegen haben wir uns an dem Vorgängersystem orientiert. Beim Prototypen ging es vor allem darum CI-Konform zu sein und das Look-and-Feel richtig umzusetzen.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  -

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  Es wurde dieselbe Technologie wie im Produktiv-System eingesetzt.

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  Es gab 2 Prototypen. Den ersten hat man verworfen, weil man die Technologie verworfen hat. Daher war es Aufwändig die Prototypen zu erstellen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Kostenteil ungefähr 20% der Gesamtentwicklung.

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  Teilweise wäre es effizienter inkrementell vorzugehen. Aber das hängt von der Kritikalität ab. Und das hängt davon ab, was man mit den Prototypen erreichen will. Um die Machbarkeit zu überprüfen ist ein Prototyp sicher sinnvoll.

  Bei jeder Entscheidung für oder gegen Prototypen muss der Kosten-Nutzen abgeschätzt werden.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.8 Theta

Datum: 27.08.2008

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	---
Mitarbeiter gesamt	---

2. Produkte des Unternehmens

- ***Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  Abteilung: Forschung und Vorentwicklung, Team Requirements und Usability-Engineering.

  Wir gestalten die Spezifikationsprozesse für die Steuergerätentwicklung. Wir schreiben die Lastenhefte und helfen anderen Abteilungen ihre Lastenhefte zu verbessern.

  Wir machen vor allem: Kombinistrumente, Innenraumkomponenten.

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren: Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware, Kommerzielle Software vs. Open-Source-Software***

  -

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  Ja, es gibt ein Vorgehensmodell. Eingebettet im Entwicklungsprozess für Fahrzeuge, gibt es eine eigene Definition für die Entwicklung von Steuergeräten, innerhalb der Definition gibt es eine genauere Beschreibung, wie Lastenhefte erstellt werden und auch weitere Themen wie Prüfung, Testen.

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  Kombiinstrument.

- ***Könnten Sie das Projekt kurz beschreiben?***

  Eine Gruppe von Leuten ist immer für eine ganze Reihe von ähnlichen Kombiinstrumenten zuständig.

  Wir erstellen, unter aber nur die Lastenhefte für die. Die gesamte andere Entwicklung machen Lieferanten.

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

  Für 1 Steuergerät 1 Lieferant

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Die gesamte Entwicklung eines Steuergerätes dauert 4 bis 6 Jahre.

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

  8 Leute. Diese entwickeln aber für mehrere Baureihen gleichzeitig.

  Rollen:

  - Gesamtprojektleiter (auch Komponentenverantwortlicher genannt)
  - Softwareverantwortlicher

  Funktionalitäten Verantwortlicher

  - IMI-Gruppe
  - Testen

  Es gibt aber sehr viele Stakeholder, weil sehr viele andere Sachen, Meldungen auf dem Kombigerät ausgeben müssen.

4. Requirements Engineering

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Die Anforderungen werden kontinuierlich weitergepflegt. Während ein Steuergerät bereits in Autos verbaut wurden, kann sich immer noch was ändern. Wir haben ca. 2 große Änderungen pro Jahr. Das Lastenheft wird also 5 bis 10 Jahre lang weitergepflegt.

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Aus Sicht der Entwickler: Der Prozess ist im Grunde eine Abwechslung zwischen Meetings und Dokumentation.

  Der Prozess über längere Zeit: Ein Änderungsmanagement gibt es schon. Validierung und Verifikation gibt es aber schon. Im Verhältnis zum Lieferanten wird dann aber auch ein intensives Review gemacht. Dort werden alle Anforderungen nochmals genau angeschaut. Während der Umsetzung durch den Lieferanten verwalten wird eine Liste-offener-Punkte.

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  Vorgängersysteme: Wir entwickeln nicht zum ersten Mal ein Kombigerät.

  Gesamtproduktentwicklung: Die grundlegenden Funktionen, die realisiert werden müssen, sind schon dort bekannt.

  Fachabteilungen: Jeder Funktion wird von einer Fachabteilung entwickelt. Mit denen müssen wir die detaillierten Anforderungen abstimmen.

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  Das machen wir. Auf unsere Doors-Datenbank haben aber auch andere Kollegen Zugang.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

  -

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  Neue Anforderungen und geänderte Anforderungen sind sozusagen das selbe.

  Echte Änderungen an Anforderungen (inhaltliche Änderungen): 10%

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  - Änderungswunsch kommt bei uns an
  - Wir analysieren ihn. Oft verursacht eine Änderung Folgeänderungen
  - Projektgremien müssen entscheiden: Faktoren: Entwicklungskosten, Stückkosten, Zeit, Risiken.
  - Parallel dazu: Der Lieferant bewertet, wie viel die Änderung kostet.
  - Gremium fällt Entscheidung.
  - Das aktualisierte Lastenheft wird fürs nächste Release an den Lieferanten gegeben.

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  Alle ändern sich. Vor allem:

  Technische Anforderungen, Nutzeroberfläche.

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

  Schwer zu sagen. Viele Änderungen werden schon von vornherein eingeplant.

  Viele Änderungen an Anforderungen kommen schon dann, wenn der Lieferant noch nicht mit der Implementierung begonnen hat.

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  Das Lastenheft wird immer aktuell gehalten.

- ***Von wem kommen die Änderungen?***

  Änderungen können von Intern kommen: Andere Abteilung ändert was, wodurch wir die Änderung übernehmen müssen. Das wird auch alles im Lastenheft dokumentiert.

  Beim Testen treten Fehler auf, die Änderungen hervorrufen.

  Auch vom Lieferanten kommen Änderungen.

- ***Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?***

  Ja, Lastenhefte werden erstellt. Zuständig: der Zuständige fürs jeweilige Steuergerät, der Systemverantwortliche. Pflichtenheft: Wird vom Lieferanten als Antwort auf unser Pflichtenheft erstellt.

- ***Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Lastenheft: gemäß DIN: geschrieben von Auftraggeber. Exakte Menüabläufe. Buskommunikation wird beschrieben. Pflichtenheft: gemäß DIN: geschrieben von Auftragnehmen. Noch eine Stufe feiner detailliert. Ist schon praktisch eine Modulspezifikation. Die Realisierung wird genau beschrieben.

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Besprechungen: d.h. Workshops.

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Identifizieren der richtigen Ansprechpartner.

  Die Kommunikation ist etwas schwierig. Es gibt zwei Sichtweisen:

  - Steuergeräte
  - Funktionen. Um eine Fahrzeugfunktion (z.B. Licht) zu realisieren, spielen viele Steuergeräte mit (beim Fortscheinwerfer sind es 19 Steuergeräte). Alle beteiligten Steuergerätentwickler müssen sich abstimmen.

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  -

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Nur: Ja/Nein.

  D.h. Anforderungen können abgelehnt/angenommen werden.

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  Es gibt ein Template für Lastenhefte. Das ist aber mehr als eine Vorlage. Es ist eine Anleitung von 500 Seiten, wo schon viele Standardanforderungen festgelegt sind. (Standardanforderungen sind z.B. Umweltanforderungen, wie dass das Steuergerät 15 Jahre lange hält, nicht rostet, usw.) Die nicht benötigten Standardanforderungen müssen dann aber aus der Vorlage entfernt werden.

  Wir haben nicht einzelne unabhängige Anforderungen. Wir beschreiben die Anforderungen in Fließtext in Doors. Teilweise gibt es eigene Dateien, die von Doors aus referenziert werden.

  Teilweise wird Statemate und Simulink/Stateflow verwendet. Diese Modellierungen werden dann aber auch wie oben übernommen.

- ***Welche Informationen werden dokumentiert?***

  Doors.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  Doors.

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

  Die Baureihe. D.h. der Endkunde. Der Endkunde schaut sich die Anforderungen aber nicht im Detail an. Aber den großen Rahmen gibt er vor.

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Wenn die Teile vom Lieferanten kommen, gibt es einen Komponententest.

  Dann gibt es einen Gesamtfahrzeugtest.

  Die Testfälle kommen aber nicht alle aus dem Lastenheft. Teilweise sind die Anforderungen auch gleichzeitig die Testfälle.

8. Prototypen

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  Nein, bei der Ermittlung werden keine Prototypen gemacht.

  Erst wenn der Lieferant feststeht, werden Prototypen gemacht. Der Lieferant erstellt Muster:

  - a-Muster: Nur Aussehen muss übereinstimmen. Es darf noch andere Hardware drinnen sein.
  - b-Muster: Seriennah, noch vieles darf anders sein.
  - c-Muster:
  - d-Muster: Die Herstellungsmaschinen müssen dieselben wie bei der Serienfertigung sein.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  -

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  -

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  -

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.9 Iota

Datum: 21.07.2008

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	99,4 Mrd. €
Mitarbeiter gesamt	273.000

2. Produkte des Unternehmens

- ***Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  Forschung und Vorentwicklung, Bereich: Softwareprozessgestaltung.

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren: Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware, Kommerzielle Software vs. Open-Source-Software***

  -

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  Bzgl. RE Eingebettet in Gesamten Entwicklungsprozess gibt es ein Abschnitt zu Lastenhefterstellung, Prüfung usw. und enthält Infos auch wer für die Phasen zuständig ist.

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  Kompetenzgruppe RE. Unternehmensübergreifend.

- ***Könnten Sie das Projekt kurz beschreiben?***

  Die Kompetenzgruppe pflegt das RE-Wissen von Daimler TSS. Daimler TSS macht den Lifecycle von Software-Systemen, Projektmanagement-Unterstützung. Macht viel Beratung und teilweise auch selbst Softwareentwicklung.

  Wie entwickeln Software-Informationssysteme, keine eingebetteten fürs Auto. Wir machen interne Business-Applikationen.

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

  Lieferanten, für eine Komponente 1 Lieferant.

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

  -

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

  8 Leute sind in dieser Kompetenzgruppe.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Rollen:

- RE-Ingenieur
- Projektleiter- Komponentenverantwortlicher
- Technikleiter

4. Requirements Engineering

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Die Anforderungen werden kontinuierlich weiter gepflegt(für Projekte die 6-7 Jahre laufen)

  Es gibt verschiedene Projekte:

  - reine Entwicklungsprojekte
  - Wartungsprojekte

  Normalerweise wird es wasserfallartig gemacht. Am Anfang werden die Anforderungen erhoben. Es gibt keine iterative Vorgehensweise. Alle späteren Änderungen werden über Change-Requests gemacht.

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Projektspezifisch wird das klassische Vorgehen in RE vorgenommen: Erheben, Änderung, Validation und Verifikation, Review und auch Dokumentiert.

  Es ist ein ständiger Diskussionsprozess.

  Ein unternehmensweites RE gibt es bei Daimler nicht. Wir sind gerade dabei eine Standardisierung zu erstellen. Es wird aber Jahre dauern, bis ein einheitliches RE sich durchsetzen wird.

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  Der RE-Ingenieur muss moderieren. Die Kunden und die Lieferanten verhandeln miteinander. Wenn der Kunde viel Know-How hat, dann stellt er die Anforderungen. Ansonsten muss der Kunde viel mit dem Lieferanten diskutieren, um auch zu verstehen, welche Anforderungen umsetzbar sind.

  Kunden sind der Fachbereich oder der IT-Bereich.

  Es wird auch unterschieden zwischen Customer und Stakeholder: Customer ist der der bezahlt. Die Stakeholder sind jene, die wirklich vom entwickelten Produkt betroffen sind.

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  Wir verwalten die Anforderungen.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

  -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  In der Realisierungsphase anhand eines Pflichtenhefts: projektabhängig.

  Zum Aufwand ist keine allgemeine Aussage möglich.

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  keine allgemeine Aussage möglich.

  Wir haben einen Issue-Change-Prozess. Dort ist definiert, wie mit Change-Requests umgegangen wird.

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  Kann man nicht einem Bereich einordnen: technisches Bereich, UI, Funktionalität.

  Es gibt auch die Kategorie Nicht-funktionale Anforderungen: Diese werden aber nicht sehr gut behandelt.

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

  Unterschiedlich.

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  Es wird ermittelt wie viel die Änderung Zeit und Geld kostet. Es wird dann mit dem Kunden kommuniziert. Der Projektleiter entscheidet letztendlich was gemacht wird.

  Die Änderungen werden je nach Art der Änderung unterschiedlich behandelt. Teilweise sind es technische Änderungen (z.B. neues Feld in der Datenbank). Manchmal ändert sich auch eine Anforderung.

- ***Von wem kommen die Änderungen?***

  Von Kunden oder Lieferanten. Der Kunde kann Issues ins Change-Management-System ein gepflegt.

- ***Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?***

  Ja. Der Schwerpunkt ist das Pflichtenheft. Die Ideen werden in einem Pflichtenheft festgehalten, um dann eine Ausschreibung für die Realisierung zu machen.

  Die Begriffe sind etwas verschwommen. Manchmal wird ein Dokument als Pflichtenheft bezeichnet, obwohl es aber eher ein Lastenheft ist. Ein Lastenheft gibt es eigentlich nicht. Die Projektidee stellt das Lastenheft dar.

  Erstellung des Pflichtenhefts:

  - Teilweise eigenes Projekt, das das Pflichtenheft erstellt.
  - Ansonsten wird das oft vom RE-Ingenieur erstellt.
  - Das hängt aber stark von der Größe des Projektes ab.

- ***Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?***

  -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Zuerst wird eine Prozess-Analyse gemacht. Da wird geschaut, wie das System aussehen muss und wie Vorgänger System ausgesehen hat um die Prozesse zu unterstützen. Die Prozesse sind meistens schon dokumentiert. Aus der ISO-Standardisierung gibt es schon definierte Prozesse. Manchmal müssen wir auch erst zusammen mit den Fachabteilungen die Prozesse analysieren. Das ist eine Kostenfrage.

  Manchmal hat der Kunde schon genaue Vorstellungen, und er hat schon ein Pflichtenheft geschrieben.

  Ansonsten werden Workshops (Besprechungen) gemacht. In den Workshops wird mit allen Stakeholdern diskutiert.

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Richtigen Gesprächspartner identifizieren und von ihm Informationen bekommen. Das Hauptproblem ist, dass die Kunden glauben, dass wir die Anforderungen liefern müssen, weil wir der Kunde sind. Den Kunden ist nicht klar, dass sie sehr viel Arbeit in die Erhebung der Anforderungen stecken müssen.

  Der Kunde kann sich schlecht vorstellen, wie die Lösung aussieht. Die wertvollen Feedbacks des Kunden kommen erst, wenn schon ein Prototyp vorliegt.

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  Strukturiert wird anhand von Templates. Es gibt z.B. ein Template für Use-Cases. Auch die Struktur von Pflichtenheften ist vorgegeben. Es ist dann auch vorgegeben, wie die UI definiert wird.

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Wir versuchen die Priorität (nach Datum) herauszufinden. Der Kunde möchte aber immer alle seine Anforderungen umgesetzt haben. Oft werden nur die Anforderungen mit der Priorität 1 in das Pflichtenheft aufgenommen.

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  Seit 1. Juni gibt es ein Template für Pflichtenhefte. Vorher haben wir es auch schon ähnlich gemacht. Aber jetzt haben wir einen Standard.

- ***Welche Informationen werden dokumentiert?***

  Nummer (Identifier), Owner, Priorität, Fließtext, Referenz auf Testfälle

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  Office. RequisitePro. Doors. Die Tendenz geht in Richtung Doors.

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

  Der Kunde.

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Es werden Testfälle erstellt. Diese werden getestet.

  Der Kunde führt eine Abnahme durch und überprüft, ob das Produkt seinen Erwartungen entspricht. Oft delegiert der Kunde diese Abnahme aber wieder an andere. Oft führen auch wie System- und Lasttests durch. Der Kunde spielt einige wichtige Szenarien durch, und dann ist die Abnahme fertig.

  Bei sehr großen Projekten gibt es eine eigene Abteilung, die Abnahmetests durchführt. Dort sitzt dann die Testabteilung mit den Kunden zusammen und die testen gemeinsam.

8. Prototypen

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  Vor allem GUI-Prototypen.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  Diskussionen mit dem Kunden.

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  HTML, Powerpoint

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  Man setzt Prototypen nur für die zentralen Dialoge ein, weil der Aufwand groß ist.

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  Man muss deutlichen sagen, wozu der Prototyp dient. Ansonsten stehen nicht sinnvolle Diskussionen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.10 Kappa

Datum: 10.09.08

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, in Mio. €	186 Mio €
Mitarbeiter gesamt	1200

2. Produkte des Unternehmens

- ***Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  Kappa hat bis auf einige Ausnahmen keine eigenen Produkte. Sie bieten SW-Lösungen an, die individuell für den Kunden entwickelt werden. Dabei handelt es sich dann um Spezialsoftware und Embedded Systems. Darüber hinaus werden Dienstleistungen zusammen mit dem Produkt (Software) und auch ohne eine Produktentwicklung angeboten. Es werden sowohl individuelle als auch standardisierte Dienstleistungen angeboten.

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren:***

  - Individualsoftware
  - Kommerzielle Software

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  Im Unternehmen wird das V-Modell XT eingesetzt. Der RE-Prozess basiert auf einer standardisierten Vorgehensweise, die vom Unternehmen entwickelt wurde.

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  Meistens entstehen die Projekte durch Ausschreibungen, die von Kunden gemacht werden. Das Unternehmen bewirbt sich auf eine Ausschreibung.

- ***Könnten Sie das Projekt kurz beschreiben?***

  Im Rahmen des Interviews wurde kein Projekt betrachtet, sondern es ging im Allgemeinen um die Aspekte des Requirements Engineering im Unternehmen.

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

  -

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

  Ein durchschnittliches Projekt dauert ca. 2 – 5 Jahre.

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

  Ein durchschnittliches Projekt benötigt ca. 8.000 bis 9.000 MT.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

4. Requirements Engineering

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Die Anforderungen werden kontinuierlich verwaltet.

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Die Phasen der Anforderungsverwaltung entsprechen den angegebenen. In einigen Fällen wird auf die Verifikation und Validierung verzichtet. Der Prozess des Requirements Engineerings ist durch eine enge Kopplung mit dem Risikomanagement gekennzeichnet. Das ist kein agiler Prozess.

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  Die wichtigste Quelle für die Anforderungen ist der Kunde. Interne Anforderungen spielen bei der Entwicklung auch eine wichtige Rolle. Die Anforderungen vom Kunden werden im Vergleich zu den internen Anforderungen priorisiert.

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  An der Verwaltung von Anforderungen ist ein Anforderungsmanagement-Team beteiligt. Projektleiter und Systemtechniker spielen dabei auch eine wichtige Rolle.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

  Der Bedarf des Kunden wird vom Anforderungsmanagements-Team in die Anforderungen übersetzt. Die Anforderungen werden genau aufgeschrieben.

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  Bei den Projekten, die vom Bund kommen, sind alle Anforderungen vorgegeben. Ansonsten ist es projektspezifisch.

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  Ein wichtiger Aspekt des Anforderungsmanagements ist sein modularer Aufbau. Falls eine Anforderung während der Entwicklung dazukommt, soll sie im Zusammenhang mit dem schon entwickelten System betrachtet werden. D. h. die Auswirkungen der dazugekommenen Anforderung auf das zu dem Zeitpunkt entwickelte System soll analysiert und überprüft werden.

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  Die meisten Anforderungen, die später hinzukommen, sind funktionale Anforderungen.

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Je umfangreicher das Projekt ist, desto größer ist der Aufwand für die Änderungen an Anforderungen.

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  Falls sich eine Anforderung ändert, wird ihr Zusammenhang mit dem gesamten Kontext genau überprüft. Erst dann kann man die Änderungen, die sich durch die Anforderung ergibt, umsetzen.

- ***Von wem kommen die Änderungen?***

  Es ist immer unterschiedlich. Im Wesentlichen ist der Kunde für die Änderungen zuständig. Auch interne Änderungen sind möglich. Ein Change Request im Rahmen des V-Modell XT ruft immer eine Iteration im Vorgehensmodell auf.

- ***Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?***

  Das Lastenheft entspricht der Beschreibung der Leistungen, die das Unternehmen dem Kunden erbringen muss. Im Lastenheft werden die initialen Anforderungen festgehalten. Im Pflichtenheft werden die Anforderungen ausformuliert. Das wird zusammen mit dem Kunden gemacht.

- ***Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?***

  Es hängt vom Projekt ab. Sowohl das Lastenheft als auch das Pflichtenheft werden vom Unternehmen geschrieben. Das Lastenheft enthält Beschreibungen, die vom Kunden kommen. Das Pflichtenheft enthält genauere Informationen über die Anforderungen. Dafür werden die Informationen im Lastenheft ausformuliert.

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Die Techniken, die für die Ermittlung von Anforderungen eingesetzt werden, sind meistens Interviews und Workshops. Den Interviews liegen vorstrukturierte Fragekataloge zu Grunde. Die Analyse der Ist-Situation und der Rahmen für die Entwicklung des Projektes liefern weitere Anforderungen.

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Zu den häufigsten Problemen im Zusammenhang mit der Anforderungsermittlung gehört der Aspekt, dass sich die Kunden oft über die Ziele des zu entwickelnden Systems nicht im Klaren sind. Deswegen können sie auch die Anforderungen nicht genau richtig formulieren.

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  Die Anforderungsanalyse basiert auf den Geschäftsmodellen und -prozessen. Ein wichtiger Aspekt dabei ist, dass Geschäftsprozesse von Geschäftsmodellen gesteuert

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

werden. Geschäftsprozesse steuern dabei die Produkte. Komplexe Anforderungen sind immer zu verfeinern. Ein wichtiger Aspekt ist die Anforderungsgruppierung. Dabei werden die Anforderungen mit demselben Hintergrund zusammengruppiert. Dadurch wird die Handhabung von Anforderungen erleichtert. Die Analyse von Anforderungen erfolgt durch ihre ausführliche Beschreibung, es wird eine Spezifikation erstellt.

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Die Anforderungen werden priorisiert. Dazu werden die Anforderungen nach verschiedenen Typen sortiert. Es entstehen sog. Container, die eine Strukturierung von Anforderungen angeben.

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  Die Anforderungen werden textuell beschrieben. Dazu gibt es vom Unternehmen entworfene Muster.

- ***Welche Informationen werden dokumentiert?***

  Jede Anforderung hat eine eindeutige Nummer. Darüber hinaus werden die Beschreibung der Anforderung, die Änderung, der für die Anforderung zuständige Bearbeiter, die Quelle der Anforderung etc. dokumentiert. Im Falle einer Änderung bekommt die Anforderung eine Versionsnummer, die mit dem entsprechenden Change Request in Verbindung gebracht werden kann.

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  Die Verwaltung von Anforderungen wird mit Microsoft Office-Tools unterstützt. Außerdem gibt es Tools, die vom Unternehmen entworfen wurden.

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

  Der Auftraggeber soll das entscheiden. Die Umsetzung der Anforderungen ist mit dem Risikomanagement eng verbunden. D. h. die Zusammenhänge zwischen den zu realisierenden Anforderungen und dem System sollen genau überprüft werden.

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Das ist die Aufgabe der Auftraggeber. Dazu werden vom Unternehmen und den Auftraggebern Tests entwickelt. Die Abnahme des Systems kann erst dann erfolgen, wenn das System vom Auftraggeber ausführlich getestet wurde.

8. Prototypen

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Prototypen dienen der Kommunikation zwischen dem Auftraggeber und dem Unternehmen. Sie werden für die Überprüfung der Umsetzbarkeit der Anforderungen und der Erwartungen der Auftraggeber eingesetzt.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  Auftraggeber

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  Projektspezifisch (UML, Papierprototype usw.)

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  Im Durchschnitt sind das ca. 10% vom gesamten Aufwand für die Entwicklung

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  Der Einsatz von Prototypen ist sehr komplex.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.11 Lambda

Datum: 29.10.2008

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	-
Mitarbeiter gesamt (weltweit)	-

2. Produkte des Unternehmens

- ***Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  Produkte: Medizinische Geräte, Kontrastmittel, Arzneimittel, IT im Umfeld Gesundheitswesen

  Dienstleistungen: Planung, Implementierung, Schulung, Beratung

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren:***

  - Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware
  - Kommerzielle Software vs. Open-Source-Software

  Kommerzielle Standardsoftware

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  Komplexe Vorgehensmodelle die speziellen Regularien im medizinischen Bereich entsprechen

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  aufgegriffenes Kundenbedürfnis (ein Krankenhaus, aber interessant für viele Krankenhäuser, d.h. oft wird im Bereich der Grundfunktionalität der Bedarf des ganzes Marktes erfasst, nicht nur der eines einzelnen Kunden)

- ***Könnten Sie das Projekt kurz beschreiben?***

  Entwicklung einer Schnittstellensoftware, die zwischen diversen Krankenhaussystemen und bildgebenden Systemen vermittelt, damit Patientendaten den erfassten Bilddaten zugeordnet werden können. Die Schnittstelle seitens der bildgebenden Systeme ist standardisiert (DICOM).

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

  Gesetzgeber

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

  Start 2007, momentan erweiterter Testbetrieb

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

  Etwa 7-8 Leute im Kernteam, das Krankenhaus nur temporär.

4. Requirements Engineering

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  kontinuierlich

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Entsprechend den Standardprozessen der Literatur, aber situationsbedingte Anpassungen der Phasen (z.B. Verkürzung)

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  Entsprechend der Herkunft gleiche Behandlung, jedoch quantitative Erfassung einer einzelnen Nachfrage (Wie oft kommt die gleiche Anforderung? -> wichtiger oder unwichtiger?). Generell Anforderungen vom Kunden, aber auch gesetzliche Vorgaben.

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  Marketing Team, das Kundenanforderungen aufnimmt und mit dem Engineering-Bereich diskutiert.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

  Das Upstream-Marketing, das mit der Entwicklung verzahnt ist.

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  Projektspezifisch, aber nicht nur hinzukommende Anforderungen, sondern auch wegfallende oder aufgeschobene

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  Nein, Upstream-Marketing entscheidet individuell; eventuell auch Umsetzung zu späterem Zeitpunkt

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  Funktionalität und Nutzeroberfläche, Qualitätsanforderungen werden nie als zusätzliche Anforderungen betrachtet, sondern als konstant gefordert

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Sehr hoher (aber projektspezifischer) Aufwand, da zusätzliche Iterationen.

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  Bewertungsschema bzgl. Relevanz einer neuen Anforderung (bspw. Sicherheitsrelevanz).

- ***Von wem kommen die Änderungen?***

  Kunde, Vertrieb, Technik, Marketing, Entwicklung, Gesetzgeber.

- ***Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?***

  Interne Pflichtenhefte, die Entwicklungsabschnitte (Milestones) definieren und ihre zeitliche Umsetzung festhalten

- ***Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?***

  -

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Interne Workshops, Kundenevents, Vorstellung einer geplanten Umsetzung mit anschließender Einholung von Feedback

  Marketing-Abteilung als zentrale Stelle zur Erfassung und Sammlung von Anforderungen, anschließendes Meeting mit dem Kunden

  Teils proaktive Suche nach Anforderungen, teils simple Annahme von Anforderungen

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Erstellen des Rankings von Anforderungen; Beispiel: Anforderungen, die sich speziell auf den deutschen Markt beziehen, besitzen keinerlei Relevanz für den französischen Markt

  Oberflächendesign

  Wirtschaftliche Abwägung von Anforderungen

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  Aufgabe des Produktmanagers des jeweiligen Bereiches nach eigener Erfahrung und einem Gusto. Wesentlich ist die Zerlegung der Anforderungen, damit diese atomar diskutiert werden können. Hierfür auch Einbindung von Schlüsselkunden.

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Zwingende Anforderungen haben höchste Priorität, hierbei gibt es auch zeitliche Einstufungen.

Andere Bemessungsfaktoren: Kundennutzen, Umsatzpotenzial, Realisierbarkeit mit vorhanden Entwicklungsressourcen

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  -

- ***Welche Informationen werden dokumentiert?***

  -

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  Nur Excel, aber weitere nicht ausgeschlossen.

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

  Upstream-Marketing

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Tests durch Upstream-Marketing (vor allem auch genaue Überprüfung der gesetzlich vorgegebenen Anforderungen) und durch Kundenabnahme (Endabnahme oder Nutzung des Release Candidate bzw. Prototyp)

  Testbedingungen bzw. gegen zu testende Anforderungen werden früh in der Entwicklung definiert

8. Prototypen

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  Kaum Prototyping im Requirements Engineering; Wenn dann zum Aufzeigen bestimmter Funktionalitäten.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  wichtige oder potenzielle Kunden

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  Produkt in unfertigem Zustand

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  zu hoher Aufwand für sinnvollen Einsatz

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  siehe Vorfrage.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.12 My

Datum: 26.08.2008

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	11.300 Mio €
Mitarbeiter gesamt	My: 160 Rho: 49.000

2. Produkte des Unternehmens

- ***Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  Produkte: Anpassung der SAP-Standardlösung für ein Westberliner Krankenhaus.

  Dienstleistungen: für das Hauptprodukt i.s.h.med und auch für SAP-Systeme allgemein(von Finanzbuchhaltung bis Materialwirtschaft in Controling Rahmen).

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren: Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware, Kommerzielle Software vs. Open-Source-Software***

  Standardsoftware, die für jeden Kunden konfiguriert und angepasst wird.

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  Es gibt ein modulbasiertes Vorgehensmodel für Basisfunktionalitäten. Es gibt eine standardisierte Anforderungsmaske um Kundenwünsche zu berücksichtigen.

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  i.s.h.med ist ein Standardprodukt, das wir für unsere Kunden konfigurieren und anpassen.

- ***Könnten Sie das Projekt kurz beschreiben?***

  i.s.h.med ist Basismodul für Grundfunktionalität von Leistungskommunikation im Krankenhaus; Stationssichten, Leistungsstellensichten und Dokumentation. Später sind weitere Module hinzugekommen.

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

  Der Kunde ist das Krankenhaus.

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Bei einfacher Anpassung 3 bis 6 Monate für die Einrichtung in einem Krankenhaus. Bei Speziallösungen braucht es länger.

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

  Anpassung des Hauptprodukts für ein bestimmtes Krankenhaus: ca. 10 Mitarbeiter

4. Requirements Engineering

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Nur am Anfang des Projekts.

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Die initialen Anforderungen legt der Kunde bereits im Pflichtenheft fest. Es wird eine Analyse der Ist- und Soll-Situation gemacht.

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  Die Anforderungen kommen vom Kunden. Zusätzlich machen wir eine Analyse beim Kunden.

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  -

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

  -

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  -

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  Es muss zusammen mit dem Kunden entschieden werden, welche zusätzlichen Funktionen bezahlt werden müssen, und welche schon innerhalb des ursprünglichen Rahmens realisiert werden können.

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  -

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

  -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  -

- ***Von wem kommen die Änderungen?***

  -

- ***Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?***

  Pflichtenheft kommt in einem vorgelagerten Schritt vor. In diesem Schritt wird analysiert, ob die Standardsoftware überhaupt im jeweiligen Krankenhaus eingesetzt werden kann.

- ***Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?***

  Im Pflichtenheft legt der Kunde seine Anforderungen an das System fest.

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Interviews, Workshops, Begehungen, bestehende Dokumente, Checklisten

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Verantwortliche, die sich gut auskennen, stehen nicht zur Verfügung. Man erhält die nötigen Informationen nicht.

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  -

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Ja, es wird priorisiert. Aus Kostengründen werden dann auch Anforderungen weggelassen. Die anfänglichen Wünsche des Kunden sind oft nicht so umsetzbar. Man einigt sich dann mit dem Kunden auf einen Katalog von Anforderungen.

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage)? Anforderungen zu dokumentieren?***

  Es wird als durchgängiger Fließtext formuliert. Dieser Text hat jedoch eine Standard-Kapitelstruktur, die immer eingehalten wird.

- ***Welche Informationen werden dokumentiert?***

  -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  -

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

  Nach der Ist-Soll-Analyse wird ein Konzept erstellt, das der Kunde abnehmen muss.

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Es werden technische Dokumentationen erstellt, damit der Kunde genau nachvollziehen kann, wie das installierte System aussieht. So kann der Kunde auch genau prüfen, ob das System sein Pflichtenheft realisiert.

8. Prototypen

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  Es gibt ein Demo- und Schulungssystem um den Kunden das System zu präsentieren.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  Kunden

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  -

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  -

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.13 Ny

Datum: 22.08.2008

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	98.786 Mio €
Mitarbeiter gesamt (weltweit)	386.000

2. Produkte des Unternehmens

- ***Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  - Applikationssoftware
  - Individualsoftware
  - Kommerzielle Software

  Die angebotenen Dienstleistungen sind produktunabhängig und neutral.

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren:***

  - Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware
  - Kommerzielle Software vs. Open-Source-Software
  - Individualsoftware und kommerzielle Software.

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  Es gibt standardisierte Prozesse. Das Vorgehen wird abhängig vom Projekt gewählt.

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  Beide Projekte sind auf Veranlassung von Kunden entstanden.

- ***Könnten Sie das Projekt kurz beschreiben?***

  2 Projekte. Die Herkunft des Projektes in Bezug auf die Branche spielt eine wichtige Rolle für die gesamte Projektentwicklung.

  Projekt A: Implementierung eines Master-Data-Management-Systems. Das ist ein sehr Infrastruktur- und IT-lastiges Projekt. Volumen: 11 – 12 Mio. € (für 18 Monate).

  Projekt B: Volumen: ca. 35 Mio. € nur für Ny.

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

  Im Rahmen des Projektes A geht es um Abgeltungssteuer. Dadurch sind die Anforderungen klar vorhanden. Das Projekt B wird für die Medienindustrie abgewickelt. Das Projekt A hat außer dem Kunden einen weiteren Kooperationspartner, der für die steuerrechtlichen Aspekte zuständig ist.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

  Projekt A: begonnen im Herbst 2007. Es läuft zurzeit die erste Phase. Das Projekt befindet sich jetzt in der Test-Phase.

  Projekt B: hat ca. 3 Jahre gedauert. Die Einführung soll am 01.01.09 stattfinden.

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

  Projekt A: 100, davon Ny: ca. 40, anderer Kooperationspartner: ca. 20, Rest der Kunde (in Auftrag der Kunde)

4. Requirements Engineering

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Projekt A: Es wird vertraglich kein Gewerk abgeliefert. Die Anforderungen werden kontinuierlich verwaltet.

  Projekt B: Die Anforderungen werden kontinuierlich verwaltet.

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Der Prozess der Anforderungsverwaltung enthält diese Phasen: Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validierung, Änderungsmanagement und Anforderungsverfolgung.

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  Während der Durchführung des Projektes ist die IT-Abteilung des Kunden der primäre Ansprechpartner.

  Beim Projekt A wurden regulatorische Anforderungen gegeben. Die Initialanforderungen werden vom Unternehmen zusammen mit dem Kunden festgelegt. Es ging um eine komplett neue Entwicklung.

  Beim Projekt B wurden regulatorische Anforderungen gegeben. Es ging um eine komplett neue Entwicklung. Das Unternehmen hat eine Vorstudie gemacht, die ca. 4 Monate gedauert hat. Die Vorstudie hat die Basis für die Ausschreibung geliefert. Ny hat darauf ein Gewerk in Form einer Leistungsbeschreibung gemacht, die von Ny zu erfüllen ist. Die Anforderungen wurden zusammen mit dem Kunden erfasst und aufgestellt.

  Die Kunden haben oft eine Fachabteilung und eine IT-Abteilung. Die Prozesse für die Verwaltung von Anforderungen, die hausintern ablaufen, sind gut strukturiert. Die Änderungen kommen oft von den Fachabteilungen. Es ist aber wichtig, die gewünschte Änderung auf den Zusammenhang mit dem bestehenden System genau zu überprüfen und zu analysieren, bevor sie umgesetzt werden soll. Viele Projekte haben eine starke

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Offshore-Komponente. Dadurch wird das Thema des Anforderungsmanagements sehr wichtig.

Die primäre Quelle ist der Kunde (70-80 %). Der Rest kommt vom Unternehmen.

Projekt A: Am Anfang wurde ca. 60-70 % der Anforderungen bekannt. Der Rest ist unklar, was im Prozessverlauf zu Mehrkosten, Mehraufwand, Verzögerungen und schließlich zu Qualitätsproblemen führt. Das Problem ist auch, dass der Kunde oft keine klaren Vorstellungen hat, wodurch das Erfassen von Anforderungen erschwert wird.

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  Der Projektleiter hat die Aufgabe, den Vertrag mit dem Kunden zu verwalten. Er muss die Kosten, das Budget, die Qualität, das Change Management verwalten. Es wird zwischen dem Vertrag und der Kundenbeziehung auf einer Seite und der Leistungserbringung auf der anderen Seite. Die Leistungserbringung ist der Aspekt, mit dem sich der Projektleiter beschäftigt, nicht so der kommerzielle Aspekt.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

  Der Projektleiter und sein Team übersetzen den Bedarf des Kunden in die Anforderungen. Das Team besteht aus Software-Architekten, Software-Beratern usw. In beiden betrachteten Projekten gibt es ein Architektur-Team, das vom Anfang des Projektes dafür zuständig ist, die Lösung aufzubauen. Das Team soll auch die Erfüllung der Anforderungen während der Implementierungsphase überwachen. Bei der Wartung ist das Team für die Fehlerbereinigung, Change Management zuständig.

  Zu Prozess-Kick-Off werden alle Stakeholder eingeladen. Dabei werden die Prozesse, die Gouvernement -Struktur, die Verantwortlichkeiten festgelegt. Die Kundenanforderungen sind Annahmen für das Projekt.

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  Projektspezifisch.

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  Die Leistungsbeschreibung des Projektes legt fest, wie der Änderungsprozess aussieht. Bei einem Gewerk erfolgt die Leistungsbeschreibung sehr detailliert. Das ist gleichzeitig der Vertrag zwischen dem Kunden und dem Unternehmen. Gleichzeitig wird der Prozess für mögliche Änderungen in der Zukunft festgelegt.

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  Primär sind das funktionale Anforderungen. Die meisten Änderungen kommen von der Fachabteilung. Die Änderungen im funktionalen Bereich betragen ca. 2/3 aller Änderungen. Der Rest sind die Änderungen im nicht-funktionalen Bereich.

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Projektspezifisch. Es wird aber schon beim Schließen des Vertrags genaue Kostenrahmen festgelegt.

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  Projekt A: Falls sich eine Anforderung ändert (kommt vom Gesetzgeber – die Abgeltungssteuer soll berücksichtigt werden), wird vom Kunden (in diesem Fall sind das die Banken, die vom Gesetzgeber informiert werden) diese Änderung übersetzt und an die Entwickler weitergeleitet.

  Projekt B. Nachdem die Leistungen von der Seite von Ny festgelegt wurden, wurden die Kosten für mögliche Änderungen festgelegt. Das Änderungsmanagement war unbefriedigend.

  In jedem Projekt erfolgt eine Leistungsbeschreibung. Es wird festgelegt, was Ny im Rahmen des Projektes zu leisten hat. Es wird auch in der Leistungsbeschreibung festgelegt, wie der Änderungsprozess aussehen soll. Es werden auch die Rahmen für die Kosten im Falle einer Änderung erfasst. Falls sich eine Anforderung ändert, wird sie bezüglich der Termine und der Kosten überprüft.

  Der Change Management-Prozess hat eine vorgegebene Struktur. Das Unternehmen hat eine spezielle Change Management-Methode. Das ist so zu sagen das Framework für das Änderungsmanagement. Am Anfang der Entwicklung werden mit dem Kunden im Rahmen eines Methodenworkshops die Prozesse für das Projekt ausgewählt. Dabei wird versucht die Wünsche der Kunden zu berücksichtigen und das Beste zu nehmen. Die Prozesse sollen eine Tool-Unterstützung haben. Falls der Kunde bestimmte Tools will, soll versucht werden, diese Wünsche zu unterstützen. Es wird im Allgemeinen festgelegt, wie das Projekt durchzuführen ist.

- ***Von wem kommen die Änderungen?***

  Die Änderungen kommen entweder vom Kunden oder das Unternehmen weist dem Kunden darauf hin, dass eine Anforderung geändert werden soll.

- ***Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?***

  -

- ***Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?***

  -

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Oft werden die Anforderungen zusammen mit dem Kunden definiert. Es werden Workshops durchgeführt. Die initialen Anforderungen kommen aus der Fachabteilung.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Die Vorstellungen der Fachabteilung sind oft nicht in den Zeitrahmen und nicht in den Kostenrahmen zu realisieren.

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  Das initiale Anforderungsdokument wird in eine DV-Spezifikation übersetzt. Oft werden dafür Use Cases verwendet. Das Ergebnis ist eine Programmiervorgabe, falls im Rahmen des Projektes eine Programmierung erfolgen soll.

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Primär ist die Priorisierung von der Fachabteilung gesteuert. Falls die Implementierung aufwendig ist, soll mit der Fachabteilung diskutiert werden. Ansonsten wird das Ampel-System verwendet.

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  Die Anforderungen werden in entsprechenden unstrukturierten Templates dokumentiert. Die Templates werden in den Gesprächen mit dem Kunden eingesetzt. Die Anforderungen werden in Fließtext in Word aufgeschrieben. Toolunterstützung findet in seltenen Fällen statt. Es gibt eine Repository, die angeben, wie Checklisten, Protokolle, Change Management etc. zu gestalten sind.

- ***Welche Informationen werden dokumentiert?***

  Es wird angegeben, wer die Anforderung initiiert hat, der Inhalt der Anforderung (scope), an welchen Stellen die Anforderung eine Auswirkung auslöst, eine Terminplanung und die Implementierungsgrobbeschreibung aber nur auf der Architekturebene. Die Implementierungsgrobbeschreibung aber nur auf der Architekturebene wird dann im Laufe der Entwicklung ergänzt. Ergänzend wird auch eine Aufwandsschätzung gemacht (zum Beispiel Mann / Monat). Die Änderungen der Anforderungen werden auch dokumentiert. Der Kunde soll bestätigen, dass die aufgefassten Anforderungen seinen Erwartungen entsprechen.

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  Die Anforderungsdokumentation ist eine reine Beschreibung. Use Cases und die Datenverarbeitungskonzepte unterstützen die Dokumentation. Die Tools werden selten eingesetzt. Das liegt an dem Aufwand, der dadurch für den Kunden entsteht. Es geht dabei nicht um finanzielle Aspekte, aber auch um Schulungen der Mitarbeiter usw. Wenige Kunden wollen in die IT für die Unterstützung des Anforderungsmanagements investieren.

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Der Kunde.

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Es wurden am Anfang des Projektes mit dem Kunden die Abnahmerahmen vereinbart. Der Kunde soll im Rahmen der Abnahme entsprechende Tests durchführen. Die Abnahme soll der Kunde machen. Die Erstellung der Testfälle und der Qualitätsüberprüfung kann auch vom Unternehmen zusammen mit dem Kunden durchgeführt werden. Die Teilabnahmen finden auch im Projektverlauf statt.

8. Prototypen

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  Prototyping ist ein sehr wichtiger Aspekt für die Entwicklung. Prototypen bieten Mittel für die Kommunikation mit dem Kunden an. Damit kann man zeigen, was realisiert werden kann und was nicht. Der Zweck des Prototypen ist die Demonstration und die Pilotierung.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  Stakeholder, der Kunde.

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  Projektspezifisch.

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  Fast kein Aufwand, denn es wird dadurch das Risiko für die Entwicklung reduziert, es wird aus den Fehlern gelernt.

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  Wenn aber das Projekt sehr kurz ist oder ein Bestandsprojekt ist, dann macht Prototyping keinen Sinn.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.14 Xi

Datum: 26.06.2008

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	3 Mio €
Mitarbeiter gesamt (weltweit)	30

2. Produkte des Unternehmens

- ***Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  Keine Produkte, nur Dienstleistungen (reine Individualsoftwareentwicklung): Industrielle Forschung und Entwicklung, E-Goverment, und E-Bussines Mobility and Connectivity.

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren:***

  - Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware
  - Kommerzielle Software vs. Open-Source-Software

  Kommerzielle Individualsoftware

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  Für jedes Tätigkeitsfeld und für jeden Auftrag andere Anforderungen, daher projektindividuelles Vorgehen. Meist agile Methoden (z.B. Scrum) oder V-Model XT.

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  Kontakt entstand durch Ausschreibung des Kunden (o2). Xi bewarb sich und gewann den Auftrag.

- ***Könnten Sie das Projekt kurz beschreiben?***

  Billing und Tarifverwaltung im Prepaidtarif von o2 sind neu zu gestalten. Es soll ein Wechsel von Excel-basierter Verwaltung auf vernünftige Softwaresysteme erfolgen.

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

  Keine Dritten, außer drei externen Mitarbeitern von o2, die jedoch wie Festmitarbeiter gehandhabt werden.

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

  Das Projekt läuft seit dem 1. September 2007 und hat eine Dauer von 13 Monaten. Danach sind Folgeprojekte geplant, da sich branchenbedingt Änderungen ergeben werden, die eine Anpassung des Systems erfordern.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

  Sechs Xi Entwickler, davon ein Projektleiter (Koordination, Spezifikation) und ein Softwarearchitekt (Konzeption, Mitarbeiterführung).

4. Requirements Engineering

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Die Anforderungen werden über die Dauer des gesamten Projekts verwaltet. Das erste halbe Jahr ging es dabei vorwiegend um die Festsetzung und Verfeinerung der Anforderungen, da die von o2 geleisteten Angaben teils widersprüchlich waren, danach mussten und müssen neue und geänderte Anforderungen gemanaged werden.

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  In diesem Projekt sehr iterativ, eine Reihenfolge ist nur schwer zu setzen. Dies ist durch die Komplexität der Aufgaben bedingt.

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  Anforderungen kommen komplett vom Kunden und dessen Fachabteilungen. Anfangs ein einwöchiger Workshop zur Einarbeitung, danach regelmäßige Treffen.

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  Projektleiter macht die Grobplanung, die Entwickler die Konkretisierung.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

  Ausschließlich Aufgabe des Projektleiters.

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  Etwa 30-40% zusätzliche Anforderungen, aber auch steigende Komplexität der bekannten Anforderungen (etwa um den Faktor 2), daraus resultierend: stark steigendes Volumen.

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  Es wurde eine Liste (Excel) aller Änderungen geführt, die mit dem Kunden durchgesprochen werden. Je nach Änderung wurde hierbei entschieden, ob eine Änderung durchgeführt werden soll (entsprechender Change Request) und zu wessen Lasten sie gehen soll.

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Hauptsächlich nicht-funktionale Anforderungen. Problem war Unkenntnis des Auftraggebers über die Komplexität der Daten.

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

  Sämtliche Änderungen in diesem Projekt haben nur unbedeutenden Mehraufwand verursacht, die Anforderungen waren relativ stabil.

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  Bei neutralem Aufwand trifft der Projektleiter die Entscheidung, bei höherem Aufwand muss der Kunde das Delta verantworten und akzeptieren. Sollte der Aufwand durch eine Änderung sinken, so wird das Delta als Puffer im Gesamtprozess genutzt.

- ***Von wem kommen die Änderungen?***

  Vom Auftraggeber o2. Nur bei Umsetzungsproblemen, die auf Inkonsistenzen der Anforderungen beruhten, musste Xi die Initiative ergreifen und sich an o2 wenden. Die Verantwortung für das Projekt lag bei zwei verschiedenen Fachabteilungen von o2.

- ***Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?***

  -

- ***Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?^***

  -

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Zu Beginn des Projekts findet ein einwöchiger Workshop statt. In diesem kommen beispielsweise Use Cases zum Einsatz. Alle Mitarbeiter sind durch Fortbildungen in den Bereichen agile Softwarearchitektur, agiles Projektmanagement sowie Kommunikation & Moderation geschult, und setzen die hier gelernten Techniken in Workshops und allgemein im weiteren Projektverlauf ein.

  Für exaktere Auskünfte wurde auf ein anderweitiges Treffen mit dem zuständigen Projektleiter verwiesen.

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Das häufigste Problem ist das Budget, sprich das Aufkommen für Änderungen. Hierbei ist weniger die letztendliche Genehmigung der zentrale Punkt, als vielmehr die Zeitspanne bis zu dieser. Der Kunde muss flexibel und beweglich sein, schnelle Entscheidungen zu treffen, was nur den wenigsten Unternehmen gelingt. Idealerweise

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

sollte eine Entscheidung innerhalb einer Woche getroffen werden, meist erfolgt sie jedoch nach etwa 8 Wochen.

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  Dies geschieht in Zusammenarbeit mit dem Kunden, jedoch nicht mehr auf Projektleiterebene, sondern meist auf Entwicklerebene im Laufe des Prozesses (mit Fachabteilung des Kunden). Etwa 80% der Anforderungen werden durch den Projektleiter geklärt, die Klärung der Detailfragen obliegt den Entwicklern.

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Sollte der Kunde Priorisierungen vorgeben, so werden diese umgesetzt. Wo nötig setzt Xi Prioritäten basierend auf technischen Anforderungen und Anforderungen an den Entwicklungsprozess. Es gibt dabei drei Prioritätsblöcke: 1. „sofort zu erledigen“ 2. „pending, auf Nachfrage“ 3. „unsicher, on hold“

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  Ja, es gibt ein Muster. Die Dokumentation erfolgt über Excel.

- ***Welche Informationen werden dokumentiert?***

  Datum, Ursprungsaufwand, Zusatzaufwand, Ansprechpartner bei o2, Verantwortlicher bei Xi, Deadline der Entscheidungsfindung, Zeitpunkt der Fertigstellung, Priorisierung der Anforderungen

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  Excel

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

  Der Kunde trifft die endgültige Entscheidung, der Projektleiter berät und ist in den Entscheidungsprozess involviert.

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Xi ist verpflichtet, ein Testprotokoll zu führen. Der Kunde testet gegen dieses Protokoll und gegen die Anforderungen. In der abschließenden achtwöchigen Testphase ist ein Xi-Entwickler beteiligt, der dem Kunden vor Ort zur Verfügung steht. Bei einem Aufwand von sechs Personenjahren für das Gesamtprojekt sind diese acht Personenwochen ein verhältnismäßig sehr geringer Aufwand.

8. Prototypen

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  Prototypen werden in zwei Bereichen eingesetzt: für Visuelles (Mockups) und für Technisches, also horizontal und vertikal. Ersteres soll dem Kunden frühestmöglich einen Eindruck vermitteln, wie die Software am Ende aussieht. Sollte dies nicht den Wünschen des Kunden entsprechen, kann dieser früh in den Entwicklungsprozess eingreifen. Letzteres hingegen dient der Verifikation des Einsatzes von Techniken. Deshalb wird ein Use Case durch alle Schichten implementiert.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  Für den Kunden.

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  Abhängig von den Mitarbeitern: Powerpoint, Photoshop, GUI-Tools. Obliegt den zuständigen Entwicklern.

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  Sehr geringer Aufwand bei hoher Sicherheit.

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  Letztgenannte Tatsache macht Prototypen unverzichtbar.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.15 Omikron

Datum: 24.09.2008

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	-
Mitarbeiter gesamt (weltweit)	-

2. Produkte des Unternehmens

- ***Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  Fast ausschließlich SAP Lösungen; Dienstleistung im Vordergrund (Beratung, Implementierung und Betrieb), aber auch in geringem Bereich SAP Addons im Automotivebereich und Individualentwicklungen zur Kundenanpassung

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren:***

  - Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware
  - Kommerzielle Software vs. Open-Source-Software

  Kommerzielle Standardsoftware

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  Eigenes Omikron-Vorgehensmodell, basiert teilweise auf V-Modell97 und der SAP ASAP Methode; RE ist hierin integriert.

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  Auf Veranlassung des Kunden.

- ***Könnten Sie das Projekt kurz beschreiben?***

  Einführung von SAP im Bereich der kompletten Materialflusssteuerung in einem Automobil-Montagewerk.

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

  Der Automobilhersteller Porsche, sprich die Konzernmutter.

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

  Das Projekt begann im August 2006 und endete im März 2008. Es folgen jedoch Optimierungen und Erweiterungen.

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Bis zu 25 Omikron-Mitarbeiter waren zeitweise beteiligt, zu Spitzenzeiten vorwiegend im Entwicklungsbereich (zehn Omikron-Mitarbeiter), seitens von Porsche kamen zehn weitere Mitarbeiter hinzu. Es gab einen Projektleiter, drei Teilprojektleiter, je Teilprojekt (drei Stück) zwei Applikationsberater, zusätzlich Systemarchitekten und Entwickler.

4. Requirements Engineering

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Definitionsphase (Anforderungsanalyse und Entwurf): Anforderungen werden aus Prozessbeschreibungen abgeleitet, Katalogisierung der Anforderungen nach Prozessen, hier 500 an der Zahl)

  Entwurfsphase (Beschreibung der Software zur Lösung): Lösungen sollen Anforderungen matchen. Der Fachbereich erstellt für jede Anforderung 1-n Anwendungsfälle und für jeden Anwendungsfall 1-n Prüfungsfälle. Eventuell werden Anforderungen geändert oder verworfen, hierfür gib es Change Request Verfahren. Diese Prozesse werden vollständig in Excel dokumentiert.

  In diesem Projekt als Sonderfall noch besonders ausgeprägte Phase der Umsetzungsverifikation: Wurden alle Anforderungen umgesetzt? Generell Standardvorgehen, hier jedoch besonders akribisch. Auch hier erfolgt eine vollständige Dokumentation.

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Die Anforderungen ergeben sich aus den Prozessen und sind relativ einfach festgehalten (Liste). Unterschieden werden nur funktionale und nicht-funktionale Anforderungen (Anforderungen an Datensicherheit beispielsweise). Es gibt es einen Kriterienkatalog zur Kategorisierung. Bedingt durch den Geschäftsbereich (SAP Umfeld) ergeben sich für jedes Unternehmen auch gewisse Grundanforderungen, die somit in jedem Projekt identisch sind.

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  Anforderungen ergeben sich aus Problemen im aktuellen Ist-Zustand, aus Soll-Prozessen und aus Rahmenbedingungen heraus. Oft stellen die Fachbereiche Anforderungen, die sie aus Altsystemen kennen. Omikron hat nur eine beratende Funktion.

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  In der Anforderungsdefinition (Lastenheft) ist der Kunde der Hauptverantwortliche (und somit Prozessverwalter), Omikron unterstützt ihn hierbei.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

  In Zusammenarbeit mit dem Kunden (im Rahmen von Workshops etc.).

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  500 waren bekannt. Etwa 10-20% kamen hinzu.

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  Abhängig von der Phase sind die damit verbundenen Kosten- und Zeitfragen. Mit ihnen stehen und fallen Entscheidungen über die Umsetzung einer Änderung. Ideal sind Streichungen von Anforderungen oder aufwandsneutrales Tauschen.

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  Hauptsächlich funktionale Anforderungen. In diesem Projekt kamen aber eher neue Anforderungen dazu, als dass sich bestehende Anforderungen geändert haben.

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

  -

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  Change Requests bei Änderungen, die Dokumentierung erfolgt über die Versionierung in Excel Tabellen.

- ***Von wem kommen die Änderungen?***

  Bedingt durch Prozessveränderungen beim Kunden.

- ***Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?***

  Anforderungsdefinition (Lastenheft), Softwarespezifikation (fachlich und technisch), Entwurf (Pflichtenheft) → Alles Bestandteil der Konzeptionsphase.

- ***Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?***

  Das Pflichtenheft enthält technische Lösungsbeschreibung: Welche Standard SAP Komponenten sollen genutzt werden, welche Einstellungen müssen getätigt werden, welche individuellen Modifikationen und Addons werden benötigt?

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Zu Beginn des Projekts findet ein Workshop mit IT Fachbereichen statt. Omikron stellt hierfür nur Applikationsberater, keine Entwickler. Der Soll-Prozess wird mit Flipcharts beschrieben, parallel sollen bereits Anforderungen erfasst und dokumentiert werden (zügiges Vorgehen). Gerade im Automotive Bereich werden zusätzlich Termine

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

vereinbart, um die Ist-Prozesse vor Ort zu betrachten und zu analysieren. Dabei werden die Mitarbeiter befragt und hieraus zu lösende Probleme abgeleitet. Dieser Prozess ist sehr informell gehalten.

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Verständnisprobleme, Begrifflichkeiten (Verständnis der Fachsprache), falsches Verständnis, Runterbrechen auf Anforderungen (schlechte Formulierung)

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  Siehe Vorfragen.

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Priorisierung erfolgt nur bei optionalen oder konkurrierenden Anforderungen zur Schmälerung des Katalogs auf den Teil der Anforderungen, der letztendlich umgesetzt wird.

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  Die Dokumentation erfolgt verbal in Textdokumenten und tabellarisch mit Nummerierung über Excel. Obsolete und geänderte Anforderungen werden nicht oder in veränderter Form in höher versionierte Dokumente übernommen, bleiben jedoch in den alten so erhalten wie zuvor.

- ***Welche Informationen werden dokumentiert?***

  Beschreibung, Nummer, Verantwortlichkeit und Herkunft

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  Nur Office Programme.

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

  Anforderungen müssen auf Geschäfts- und/oder Projektziele abgestimmt sein. Sollte eine Anforderung diese Kriterium nicht erfüllen, so wird priorisiert und damit selektiert.

  Der Kunde entscheidet über die Umsetzung der Anforderungen. Omikron hat nur beratende Funktion.

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Größere Kunden besitzen meist ein eigenes Quality Management (Testcenter), das die Prüfung durchführt. Allgemein werden anhand von Prüffällen und –varianten

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Negativtests, Modultests, Integrationstests sowie Last- und Performancetests durchgeführt.

8. Prototypen

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  Bestimmte Prozesse werden in einem Demosystem eingerichtet, damit der Kunde sich ein Bild vom späteren Endsystem machen kann. So werden Anforderungen besser verstanden und können somit besser beschrieben werden. Sonderwünsche des Kunden werden früh erkannt und mit der Standardsoftware umgesetzt oder idealerweisen vermieden. Nötigenfalls Addons.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  Projektmitarbeiter aus Fachbereich oder Mitarbeiter aus späteren Nutzteams.

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  Modifikation von Standard SAP Demosystemen oder von Referenzsystemen alter Kunden.

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  Geringer Aufwand (geringerer als „den Kunden aus seinen Erwartungen herauszuholen“). Durch das Vorhandensein von Demosystemen, die nur geringfügig für den Prototype modifiziert werden müssen, ist der Aufwand gemessen am Gesamtprojekt verschwindend gering.

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  Erwartungsmanagement, Kunde äußert ohne Prototyping meist sehr viele Wünsche, die nicht umgesetzt werden können, was bei ihm Enttäuschung hervorruft.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.16 Pi

Datum: *08.07.08*

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	198 Mio €
Mitarbeiter gesamt	1400

2. Produkte des Unternehmens

- ***Welche Software-Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  Pi entwickelt keine Produkte. Pi ist ein Projekt- und Beratungshaus. Die Projekte entstehen ausschließlich auf Veranlassung der Kunden. Die Software wird im Zusammenhang mit dem Dienstleistungen angeboten. Zu den Kernaktivitäten des Unternehmens gehört auch das Begleiten des Kunden in den Phasen von der Anforderungsspezifikation bis zur Systemeinführung.

- ***Wenn die Dienstleistungen angeboten werden, welche Anforderungen werden an die Dienstleistungen gestellt? Wie werden sie ermittelt?***

  Es werden Beratungsdienste angeboten, wie zum Beispiel die Anforderungsanalyse. Es werden Project Services (Engineering/Integration, IT-Beratung) angeboten.

  - Befähigung des Kunden durch Beratung und Coaching (Methodenkompetenz)
  - Mitwirkung beim Anforderungsmanagement (Durchführungskompetenz)

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren:***

  Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware

  - Kommerzielle Software vs. Open-Source-Software
  - Individualsoftware, speziell auf den Kunden zugeschnitten.

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  Es gibt ein iteratives Vorgehensmodell, das sich an das Wasserfall-Modell orientiert, für die Entwicklung. Die genaue Vorgehensweise hängt aber vom Projekt ab.

  Im Requirements Engineering hat Pi eine Spezifikation, die beschreibt, wie das RE gestaltet werden soll und in den gesamten Entwicklungsprozess integriert werden soll. Die Anforderungsanalyse ist ein Teil der Konzeption. Das Anforderungsmanagement basiert auf dem typischen Projektmodell, von der Anforderungsspezifikation bis zur Realisierung. Die Anforderungsspezifikation ist dann in das Projektmodell eingebettet. Das ist dann die erste Phase bei der Durchführung eines Projektes. Die Anforderungsspezifikation beschreibt, wie die Anforderungen erhoben werden, wie sind zu dokumentieren sind etc. Es werden methodische Bausteine in Bezug auf die

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Verwaltung von Anforderungen vorgegeben. Auch wie die Anforderungen erhoben und beschrieben werden sollen.

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  Das Projekt ist 2007 entstanden. Das Projekt ist auf Veranlassung des Kunden entstanden.

- ***Könnten Sie das Projekt kurz beschreiben?***

  Es geht um Verkaufssysteme für Automobilhersteller. Im Rahmen des Projektes soll ein Verkaufssystem für die europäischen Automobilhändler entwickelte werden.

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

  Es gibt weitere Kooperationspartner. Pi ist der Ansprechpartner für das Gesamtprojekt. Es sind mehrere Teilprojekte im Rahmen dieses Projektes.

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

  Das Projekt dauert noch. Es wird ca. 3.5 Jahre dauern. Nach ca. 4 Jahren soll die Entwicklung abgeschlossen werden.

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

  50 insgesamt, 25 bei Pi. Es gibt einen Projektleiter, einen fachlichen und einen technischen Chef-Designer. Unter Projektleitern wird es zwischen dem Projektmanager, der sich um den Kontakt zu den Kunden kümmert, und dem Projektleiter, der sich um das Projekt kümmert, unterschieden.

4. Requirements Engineering

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Die Anforderungen werden in einer der Projektphasen erhoben. Die Verwaltung von Anforderungen findet im Laufe des ganzen Projektes statt. In der Phase „Anforderungsspezifikation“ (oder „Anforderungsanalyse“) werden die Anforderungen erhoben und analysiert. Das Ergebnis dieser Phase ist ein Anforderungsdokument. Nach der Abnahme des Anforderungsdokuments wird das Fachkonzept entworfen. Darauf basierend werden die Realisierungskonzepte erarbeitet. Dabei werden fachliche (funktional und nichtfunktional), gesetzliche Anforderungen, IT-Anforderungen aus dem Unternehmen (zum Beispiel, welche Plattform eingesetzt werden soll usw.) unterschieden.

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Die Anforderungen werden erhoben, dann analysiert und dokumentiert. Das Anforderungsdokument stellt die Grundlage für die Aufstellung des Fachkonzeptes dar. Im Rahmen der fachlichen Konzeption wird die Lösung für die Anforderungen

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

beschrieben. Genau in dieser Phase erfolgt die Verifikation von Anforderungen. Dabei geht es darum festzustellen, ob die Lösung den Anforderungen entspricht. Es geht dabei darum, die Anforderungen und die Lösungen in Einklang zu bringen.

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…) Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  Die Anforderungen kommen von Kunden. Die Fachabteilungen übermitteln die Anforderungen an das Unternehmen.

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  Die Verwaltung von Anforderungen ist das ganze Entwicklungsteam beteiligt. Der Projektleiter ist für den gesamten Prozess der Anforderungsverwaltung zuständig. Das Team teilt die Arbeit nach fachlichen Gesichtspunkten.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

  Die Anforderungen sind in natürlicher Sprache beschrieben. Die Anforderungen werden in spezielle dafür vorgesehene Formen eingetragen. Dadurch erfolgt die Übersetzung der Anforderungen. Die Anforderungen werden konkretisiert.

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  Ca. 90% der Anforderungen sind schon zu Beginn der Entwicklung bekannt. Das hängt auch davon ab, ob man den Kunden und seine Geschäftsabläufe schon kennt. Falls es um einen neuen Kunden geht, dann werden ca. 70% der Anforderungen am Anfang ermittelt.

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  Sowohl wenn neue Anforderungen kommen als auch wenn sich die Anforderungen ändern, soll ein Change Request - Verfahren erfolgen. Im Rahmen dieses Verfahrens wird ein Gremium einberufen, das entscheiden soll, ob die Änderungen umgesetzt werden soll. Es wird unterschieden, ob es um eine komplett neue oder eine geänderte Anforderung geht.

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  Es kommen oft fachliche Details. Es sind oft auch nichtfunktionale Anforderungen. Zu den nichtfunktionalen Anforderungen gehören Projektanforderungen, Anforderungen, die angeben, wie die Altsysteme abgelöst werden sollen, nichtfunktionale Systemanforderungen, wie zum Beispiel, Antwortzeiten, Betriebsanforderungen. Es ist wichtig, die nichtfunktionalen Anforderungen zu erheben. Nützlichkeit und Akzeptanz des Systems basiert auf der Erfüllung von nichtfunktionalen Anforderungen.

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Je später im Laufe der Entwicklung eine Anforderung dazu kommt, desto schwieriger ist es, sie zu integrieren. Besonders schwierig ist es, wenn das IT-Konzept abgeschlossen ist und dann noch nichtfunktionale Anforderungen kommen. Zum Beispiel die Antwortzeiten sind für das IT-Konzept entscheidend.

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  Falls sich eine Anforderung ändert, wird sie in ein Gremium eingebracht uns das Gremium soll entscheiden, ob die Anforderung umgesetzt wird. Dabei sollen auch die Entscheidungen bezüglich der Kosten getroffen werden, die durch die Änderung entstehen. Der andere Aspekt ist die Zeit. Die Frage ist, ob die Durchführung des Projektes durch die Änderung bzw. das Hinzufügen einer Anforderung beeinflusst wird. In diesem Fall soll entschieden werden, wie sich die Realisierungszeit dadurch ändert. Falls es eine Anforderung kommt, soll entscheiden werden, ob es sich um eine funktionale oder nichtfunktionale Anforderung und ob es sich um eine komplett neue Anforderung oder eine Änderung einer Anforderung handelt. Anschließend kommt die Anforderung in das Change Request Verfahren. Das Gremium soll dann entscheiden, wie die Anforderung zu behandeln ist.

- ***Von wem kommen die Änderungen?***

  Das Unternehmen sammelt die Anforderungen in „beschreibender Form“. Der Kunde hat eine Fachabteilung, die für das Aufstellen der Anforderungen zuständig ist. Dabei wird es zwischen zwei Arten der Fachabteilung unterscheiden: das sind zum Einen die Endnutzer und die Prozessabteilungen, die sich mit dem Konzept der Entwicklung der Lösung beschäftigen. Im Rahmen des Projektes beschäftigt sich diese Abteilung mit dem Aspekt, wie soll das Verkaufssystem aussehen. Diese zwei Abteilungen entscheiden dann über die Änderungen in Anforderungen.

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Die Anforderungsermittlung erfolgt durch Workshops, Interviews mit dem Kunden, im Rahmen des Projektes- mit den Fachabteilungen, Prototypen. Dabei werden sowohl die Endbenutzer als auch die Prozessgestalter gefragt. Oft ist auch die IT-Abteilung an den Interviews oder den Workshops beteiligt. Eine andere Möglichkeit für die Ermittlung von Anforderungen bietet die Analyse der Dokumentation von schon bestehenden Systemen. Die Fachabteilungen weisen auf die Aspekte in der Dokumentation schon bestehender Systeme hin, die sie gerne im neuen System haben wollen. Dabei stellen Workshops, so zu sagen, große Interviews dar. Im Rahmen eines Workshops werden die Stakeholder zu den Anforderungen gefragt. Während der Interviews werden die Anforderungen in großen Runden von 2-3 Leuten überlegt. Entscheidend für die Auswahl zwischen dem Workshop und dem Interview ist die Größe der Runde der Gefragten.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Das Problem besteht oft darin, unterschiedliche Ziele verschiedener Stakeholder unter einen Hut zu bringen.

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  Das Anforderungsteam hat die Aufgabe, die ermittelten Anforderungen zu konkretisieren. Die Anforderungen sind in natürlicher Sprache beschrieben. Die Anforderungen werden in spezielle dafür vorgesehene Formen eingetragen. Dadurch erfolgt die Übersetzung der Anforderungen. Die Anforderungen werden konkretisiert.

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Die Priorisierung ist kundenabhängig. Dabei können die Anforderungen nach muss / soll / kann aufgeteilt werden. Eine andere Möglichkeit besteht darin, dass die Priorisierung auf der Basis gesetzlicher Vorschriften oder der finanziellen Aspekte erfolgt.

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  Es gibt ein Anforderungsdokument, das die Anforderungen beschreibt, eine Spezifikation der Anforderungen (das Fachkonzept), die die fachlichen Aspekte beschreiben. Die genaue Umsetzung ist nicht der Teil der fachlichen Spezifikation. Dabei werden Geschäftsabläufe, elementare Funktionen, Daten, Benutzeroberflächen etc. beschrieben. Das IT-Konzept beschreibt, welche Aspekte wie umgesetzt werden sollen, zum Beispiel, welche Datenbanken oder Entwicklungsumgebungen eingesetzt werden müssen. Das IT-Konzept beschreibt die IT-Komponenten, die benötigt werden, um die Anforderungen zu realisieren. Das Fachkonzept stellt so zu sagen den Vertrag zwischen dem Unternehmen und dem Kunden dar.

- ***Welche Informationen werden dokumentiert?***

  Folgende Informationen werden aufgeschrieben. Das Anforderungsteam soll die Anforderungen dokumentieren. Dabei wird die Motivation für die Anforderung, der Titel der Anforderung, die Nummer, eine detaillierte Beschreibung der Anforderung, Abhängigkeiten zu den anderen Anforderungen, die Release Nummer, die Klassifizierung (soll / muss / kann). Die Anforderungen werden nach Prozessen dokumentiert.

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  Die Dokumentation von Anforderungen wird mit Office-Werkzeugen unterstützt. Andere Werkzeuge werden projektspezifisch eingesetzt.

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Der Kunde kann sich überlegen, welche Anforderungen für ihn besonders wichtig sind. Das teilt er dann mit und genau diese Anforderungen sind dann umzusetzen. Das wird im Rahmen von Workshops gemacht.

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Das erfolgt im Rahmen des Entwicklungsprozesses. Das kann im Rahmen der Überprüfung der Anforderungsdokumentation stattfinden. Eine andere Möglichkeit bieten Prototypen. In der Testspezifikation wird angegeben, welche Tests durchgeführt werden sollen, damit die Anforderungen als erfüllt gelten. Die Anforderungsanalyse ist das erste Element der Fachspezifikation. Die technischen Aspekte sind nicht in der fachlichen Konzeption enthalten. Es gibt auch eine Vorlage für das Fachkonzept. Die Prozesse werden mit EPK modelliert. Da es für den Kunden problematisch sein kann, die Prozesse zu verstehen, wird es versucht, keine Tools (zum Beispiel, ARIS) im Fachkonzept einzusetzen.

8. Prototypen

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  Prototypen werden vorwiegend für die Kommunikation mit dem Kunden eingesetzt. Am Anfang werden Prototypen in Gesprächen mit Kunden eingesetzt, um das Projekt zu gewinnen. Zum Verständnis des Fachbereichs werden die Prototypen in Form von Mock-Up eingesetzt, die oft in Papierform oder mit Office-Werkzeugen erstellt werden. Prototypen werden auch verwendet, um die Konzepte der späteren Entwicklungen darzustellen.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  Für den Fachbereich, End-User und weitere Kunden.

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  Projektspezifisch. Papierform, Office-Tools, leichtgewichtige IT-Werkzeuge (z. B. Visual Basic) oder die schwergewichtige Technik, die auch beim System später eingesetzt werden.

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  Screen Mock-Up sind einfach zu erstellen, die Prototypen in schwergewichtiger Technik sollen entwickelt werden, wenn die Technik noch nie ausprobiert wurde.

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  Der Einsatz von Prototypen bringt viele Vorteile. Wenn die Technik nicht bekannt ist, kann man dadurch die Technik ausprobieren. Falls die Technik bekannt ist, kann der Prototyp dazu genutzt werden, um dem Kunden das system zu präsentieren.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang A.17 Rho

Datum: *17.07.2008*

1. Allgemeine Informationen zum Unternehmen

	2007
Umsatz gesamt, Deutschland, in Mio. €	11.300 Mio €
Mitarbeiter gesamt	49.000

2. Produkte des Unternehmens

- ***Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten?***

  Software- und Hardwareprodukte. Diese gehören immer zusammen. Die Software wird nie alleine verkauft.

  Zu den Produkten werden auch Dienstleistungen angeboten.

- ***Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren: Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware, Kommerzielle Software vs. Open-Source-Software***

  -

- ***Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das RE?***

  -

3. Projekte des Unternehmens (ein bestimmtes Projekt)

- ***Wie ist das Projekt entstanden? Auf Veranlassung von Kunden oder intern?***

  Die Entscheidung für das Produkt wurde bei uns gefallen. Aber die Anforderungen der Kunden spielen dabei eine Rolle. Es wurde eine Marktforschung durchgeführt, um zu wissen, welche Produkte sich die Kunden wünschen.

  Es handelt sich um Innovationen. Die Kunden regen die Innovation an. Die Kunden beschreiben ihr Problem, und wir schauen dann, ob wir das realisieren können oder nicht. Es geht aber nicht ausschließlich um eine technische Innovation, denn der Kundennutzen steht im Vordergrund.

- ***Könnten Sie das Projekt kurz beschreiben?***

  System im Operationssaal. Es geht um Bildgebung bei Operationen im Zusammenhang mit Knochen.

- ***Wer ist am Projekt beteiligt? Gemeint sind weitere Kooperationspartner***

  Rho Medial und Zulieferer. Rho ist vor allem Integrator. Die einzelnen Teile liefern Zulieferer. Wie der Kundennutzen erzeugt wird, weiß jedoch nur Rho.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Wie lange hat das Projekt gedauert? Oder dauert das Projekt noch?***

  Ende 2006; es dauert voraussichtlich 3 Jahre

- ***Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen.***

  Rho: 10

  Zulieferer: 60-70

4. Requirements Engineering

- ***Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?***

  Das Produktmanagement schreibt das Lastenheft. Dieses wird zum Start des Projekts geschrieben. Dort steht die initiale Erwartung an das Projekt drinnen.

  Im Laufe des ganzen Projektes ist RE ein kontinuierlicher Prozess, der immer wieder Feedback während der Entwicklung bekommt.

  Das Ziel des Projektes steht im Business-Plan.

- ***Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?***

  Hanore-Methode. Quantifizieren von Anforderungen.

  Dann werden die Anforderungen formuliert. Dort ist genau vorgegeben, wie sie formuliert werden müssen.

  Dann werden die Anforderungen bewertet und in eine Datenbank eingetragen.

  Wenn der Projektleiter das Pflichtenheft formuliert, wird auch nach dieser Methode vorgegangen.

- ***Woher kommen die Anforderungen? (Kunde, intern, Berater…). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?***

  Die Kunden werden eingebunden.

- ***Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?***

  Der Produktmanager muss dieses Thema betreuen.

- ***Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?***

  Die Anforderungen des Lastenhefts wird von uns ins Pflichtenheft übersetzt.

- ***Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?***

  Nachdem Lasten- und Pflichtenheft festgeschrieben worden sind, sind die Anforderungen sehr stabil. Es gibt dann nur mehr sehr wenige Änderungen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?***

  Die neuen Anforderungen werden auch teilweise umgesetzt. Im Prozess wird entschieden welche Anforderungen umgesetzt werden.

  Zurzeit stellen wir den Prozess auf Scrum um, um flexibler zu sein.

- ***Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)***

  Es kommen vor allem funktionale Anforderungen dazu. Bei nicht-funktionalen Anforderungen kommen vor allem Anforderungen bezüglich der Ergonomie.

- ***Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?***

  Nicht mehr als 10%. Das ist eine grobe Schätzung anhand unserer Meilensteine.

  Wenn wesentlich mehr Anforderungen neu kommen, dann wird für die neuen Anforderungen ein neues Projekt eröffnet. Das neue Projekt ist dann oft eine Weiterentwicklung des anderen. Dass das alte Projekt sofort abgebrochen wird, kommt so gut wie nie vor.

- ***Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?***

  Es gibt ein Tracing.

- ***Von wem kommen die Änderungen?***

  Normative Anforderungen kommen von der Qualitätsabteilung.

  Andere Anforderungen kommen vom Produktmanagement. Auch die Änderungswünsche kommen vom Produktmanagement.

- ***Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?***

  Lastenheft vom Produktmanagement. Pflichtenheft von uns, dort halten wir die technischen Anforderungen fest.

- ***Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?***

  Pflichtenheft werden die Konzepte festgehalten. Im Lastenheft stehen die initialen Anforderungen an das Produkt.

5. Anforderungsermittlung und Anforderungsanalyse

- ***Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?***

  Wir haben einen bestimmten Kundenkreis von dem wir Anforderungen bekommen.

  Feldbeobachtung, Workshops, Befragungen, Prototypen.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten?***

  Die Qualität der Anforderungen ist ein großes Problem. Die Anforderungen so aufzuschreiben, dass sie gut verständlich sind, ist schwer. Die Kunden liefern die Anforderungen nicht in der Form, in der wir sie brauchen.

  Die Internationalität ist auch ein Problem. Mit unserem Produkt gehen Leute mit verschiedenem Bildungshintergrund um.

- ***Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen)***

  Manchmal werden Kunden eingeladen und mit denen wird das Lasten- und Pflichtenheft diskutiert.

- ***Werden die Anforderungen priorisiert? Falls ja, wie?***

  Die Anforderungen werden priorisiert nach: muss, soll, kann

6. Anforderungsdokumentation

- ***Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren?***

  Es gibt eine Vorgabe, wie die Sätze formuliert werden müssen. Diese Vorgaben werden immer eingehalten.

  Die Testfälle werden gleich bei der Anforderung angegeben.

  Teilweise wird auch UML verwendet. Aber das wird eher schon auf der Seite der Realisierung gemacht.

- ***Welche Informationen werden dokumentiert?***

  Anforderung als Text, Kommentar,

- ***Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?***

  RequisitePro (?), CaliberRM.

7. Verifikation und Validierung

- ***Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?***

  Das Produktmanagement trifft diese Entscheidungen.

- ***Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?***

  Durch das Tracing ist klar, welche Anforderungen umgesetzt worden sind. Das entscheidende ist, dass alle Traces umgesetzt worden sind.

  Letztendlich gibt es einen Integrations- und Abnahmetest. Dort wird das überprüft.

8. Prototypen

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- ***Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, …)***

  1) Prototypen im Rahmen eines Workshops: In einem Workshop wird ein Prototyp eingesetzt um besseres Feedback zu bekommen.

  2) Prototypen um die Machbarkeit zu demonstrieren.

  3) Prototypen werden auch verwendet, um den Stand des Projektes zu demonstrieren: Zwischendurch werden alle Komponenten integriert, um zu zeigen, dass Teilintegrationen schon funktionieren. So treten beim letztendlichen Integrationstest nicht mehr so schlimme Fehler auf.

- ***Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt?***

  Zum Teil der Endanwender.

  Zum Teil wird der Prototyp für die Integration erstellt.

- ***Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, …)***

  Es wird die Technologie eingesetzt, die auch im Endprodukt eingesetzt wird. Bei uns handelt es sich aber auch nicht um Software-Prototypen.

  Nur in seltenen Fällen wird eine Software-Benutzeroberfläche in Flash realisiert.

- ***Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?***

  -

- ***Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?***

  -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang B Thematische Zusammenfassung der Interviews

Anhang B.1 Requirements Engineering

1. Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes?
   - Automotive
     - *Gamma:* Es wird ein initiales Lastenheft (Baseline) erstellt, Hauptaugenmerk liegt jedoch auf Change Requests bedingt durch Verfeinerung der Anforderungen und somit auch bedingt durch ungenaue Erfassung von Prozessen/Anforderungen. Der gesamte RE Prozess ist auf Change Management ausgelegt (Projekt wird als Fleckenteppich betrachtet).
     - *Delta:* Kontinuierlich, Ableitung der Kennzahlen am Ende des Jahres bzgl. Anzahl der umgesetzten Anforderungen
     - *Epsilon:* -
     - *Iota:* Die Anforderungen werden kontinuierlich weiter gepflegt
     - *Theta:* Die Anforderungen werden kontinuierlich weitergepflegt
   - Software
     - *Alpha:* Projektspezifisch, aber meist am Anfang des Projekts und dann am Ende.
     - *Beta:* Mischung aus beiden. RE ist eine Phase im Projekt. Am Anfang wurde eine fachliche Anforderungssammlungsphase gemacht. Dort wurde gesammelt und Priorisiert und dann festgelegt, welche Anforderungen realisiert werden. Während der Entwicklung des Release 1 wurde teilweise neu priorisiert. Am Ende des Release 1 wurde aus dem gelernten Wissen und den nicht realisierten Anforderungen eine neue Liste erstellt und nochmals Priorisiert. Diese neue Liste wurde dann im Release 2 realisiert. Derzeit entwickeln wir das Release 3; schon am Anfang war geplant mehrere Releases zu entwickeln.
     - *Eta:* Beim ursprünglichen Projekt wurden die Anforderungen einmal erhoben. Dort wurde ein Grobkonzept geschrieben, das die Business-Anforderungen enthält. Anschließend wurde ein Fachkonzept und ein IT-Konzept erstellt. Aufgrund des IT-Konzepts wird entwickelt. Dazu gibt es einen Change-Management-Prozess mit Change-Requests. Es gibt zwei Change-Requests: Neue Features, Fehler im Fachkonzept ausbessern
     - *Kappa:* Die Anforderungen werden kontinuierlich verwaltet
     - *Ny:* Die Anforderungen werden kontinuierlich verwaltet
     - *Xi:* Die Anforderungen werden über die Dauer des gesamten Projekts verwaltet
     - *Omikron:* nur in der Anfangsphase
     - *Pi:* Die Verwaltung von Anforderungen findet im Laufe des ganzen Projektes statt
   - Medizintechnik
     - *Rho:* Im Laufe des ganzen Projektes ist RE ein kontinuierlicher Prozess, der immer wieder Feedback während der Entwicklung bekommt
     - *My:* Nur am Anfang des Projekts
     - *Lambda:* kontinuierlich
     - *Zeta:* Die Anforderungen werden kontinuierlich verwaltet. Es gibt einen Lenkungsausschuss, der die Verwaltung von Anforderungen koordiniert.
2. Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung?
   - Automotive

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *Gamma:* Change Request wird priorisiert, grob beschrieben in Projektteam bewertet, Budgetierung geklärt, dann geht in IT, wird ausformuliert und in Lastenheft eingetragen und nach Release implementiert.
- *Delta:* Anforderungserhebung mittels Ticketingssystem.
- *Epsilon:* Excel Verwaltung der Anforderungen
- *Iota:* Projektspezifisch wird das klassische Vorgehen in RE vorgenommen: Erheben, Änderung, Validation und Verifikation, Review und auch Dokumentiert.
- *Theta:* Validierung und Verifikation gibt es aber schon. Im Verhältnis zum Lieferanten wird dann aber auch ein intensives Review gemacht. Dort werden alle Anforderungen nochmals genau angeschaut.

- Software
  - *Alpha:* Nicht festgeschrieben. Workshop(Diskussion) mit Protokoll-> Lastenheft -> Validierung mit den Kunden.
  - *Beta:* Alle diese Phasen, außer dem Change-Management, werden am Anfang in Analysephase durchgeführt. Dann wird das Angebot mit dem Preis erstellt. Alle späteren Änderungen wurden über Change-Requests eingebracht. Für jede Änderung wird geschätzt, welche Auswirkung sie auf Zeit, Kosten, Aufwand. Der Kunde konnte dann entscheiden, ob er die Änderung will oder nicht. Je nach Kosten der Änderung muss der Kunde dann mehr bezahlen.
  - *Eta:* Der RE-Prozess wird wie folgt durchgeführt: Bei neuer Anforderung wird die Anforderung aufgeschrieben. Sie wird dann von der Zentrale interpretiert und umformuliert. Dann wird der Aufwand evaluiert, den die Anforderung verursacht. Dann wird entschieden, ob die Anforderung umgesetzt wird. Dann wird eine Deadline für das nächste Release festgelegt. Bis zu dieser Deadline müssen alle neuen Anforderungen verständlich aufgeschrieben werden. Dann werden alle neuen Anforderungen priorisiert. Die Anforderungen die mit gesetzlichen Anforderungen zu tun haben müssen auf jeden Fall umgesetzt werden. Die anderen Anforderungen werden je nach Wichtigkeit und Kosten umgesetzt oder nicht. In einem Abstimmmeeting wird das alles entschieden. Eine Grundabstimmung mit den Märkten wird aber schon vorher durchgeführt.
  - *Kappa:* Die Phasen der Anforderungsverwaltung entsprechen den angegebenen. In einigen Fällen wird auf die Verifikation und Validierung verzichtet. Der Prozess des Requirements Engineerings ist durch eine enge Kopplung mit dem Risikomanagement gekennzeichnet. Das ist kein agiler Prozess.
  - *Ny:* Der Prozess der Anforderungsverwaltung enthält diese Phasen: Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validierung, Änderungsmanagement und Anforderungsverfolgung.
  - *Xi:* sehr iterativ, eine Reihenfolge ist nur schwer zu setzen. Dies ist durch die Komplexität der Aufgaben bedingt
  - *Omikron:* Die Anforderungen ergeben sich aus den Prozessen und sind relativ einfach festgehalten (Liste)
  - *Pi:* Die Anforderungen werden erhoben, dann analysiert und dokumentiert. Das Anforderungsdokument stellt die Grundlage für die Aufstellung des Fachkonzeptes dar. Im Rahmen der fachlichen Konzeption wird die Lösung für die Anforderungen beschrieben. Genau in dieser Phase erfolgt die Verifikation von Anforderungen. Dabei geht es darum festzustellen, ob die Lösung den Anforderungen entspricht. Es geht dabei darum, die Anforderungen und die Lösungen in Einklang zu bringen
- Medizintechnik
  - *Rho:* Hanore-Methode. Quantifizieren von Anforderungen. Dann werden die Anforderungen formuliert. Dort ist genau vorgegeben, wie sie formuliert werden

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

müssen. Dann werden die Anforderungen bewertet und in eine Datenbank eingetragen.
  - *My:* Die initialen Anforderungen legt der Kunde bereits im Pflichtenheft fest. Es wird eine Analyse der Ist- und Soll-Situation gemacht
  - *Lambda:* Entsprechend den Standardprozessen der Literatur, aber situationsbedingte Anpassungen der Phasen (z.B. Verkürzung)
  - *Zeta:* Der Prozess der Verwaltung von Anforderungen liegt nah an der Beschreibung des Prozesses des Anforderungsmanagements in der Literatur. Die Anforderungen werden erst von verschiedenen Stakeholdergruppen erhoben, in Spezifikationsdokument erstellt dann dem Lenkungsausschuss vorgelegt, falls die Änderungen im Laufe des Projektes ergeben dann wird Lenkungsausschuss darüber entscheiden.

3. Woher kommen die Anforderungen? (Kunde, intern, Berater...). Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht?
   - Automotive
     - *Gamma:* Zum einen aus der IT (nicht funktionale Anforderungen, Schnittstellen, etc.), zum anderen aus den Fachbereichen (funktionale Anforderungen).
     - *Delta:* Es macht einen großen Unterschied, ob Anforderungen aus einem Projekt kommen, oder ob einzelne Anwender was beisteuern. Bei Anforderungen aus Projekten werden Workshops gemacht, sonst nicht.
     - *Epsilon: Stakeholder.* Bei neuen Anforderungen können das die Endanwender, oder aber auch andere sein
     - *Iota:* Die Kunden und die Lieferanten verhandeln miteinander
     - *Theta:* Fachabteilungen
   - Software
     - *Alpha:* Der Kunde und der Anwender geben die Anforderungen. Außerdem die Rahmenbedingungen (interne Anforderungen), z. B. bezüglich der Programmierrahmen aus IT Abteilung.
     - *Beta:* In der ersten Phase hauptsächlich vom Kunden. Im Laufe des Projektes: 70% vom Kunden, 30% von uns. Bei der Realisierung haben wir gemerkt, dass sich nicht alle Features so umsetzen lassen, wie es der Kunde anfangs gedacht hat.
     - *Eta:* Die Änderungen können von überall kommen: alle Anwender, Management, Lieferanten, strategische Entscheidungen, Wartung.
     - *Kappa:* Die wichtigste Quelle für die Anforderungen ist der Kunde. Interne Anforderungen spielen bei der Entwicklung auch eine wichtige Rolle. Die Anforderungen vom Kunden werden im Vergleich zu den internen Anforderungen priorisiert.
     - *Ny:* Die primäre Quelle ist der Kunde (70-80 %). Der Rest kommt vom Unternehmen
     - *Xi:* Anforderungen kommen komplett vom Kunden und dessen Fachabteilungen
     - *Omikron:* Anforderungen ergeben sich aus Problemen im aktuellen Ist-Zustand, aus Soll-Prozessen und aus Rahmenbedingungen heraus. Oft stellen die Fachbereiche Anforderungen, die sie aus Altsystemen kennen. Omikron hat nur eine beratende Funktion
     - *Pi:* Die Anforderungen kommen von Kunden
   - Medizintechnik
     - *Rho:* Die Kunden werden eingebunden
     - *My:* Die Anforderungen kommen vom Kunden. Zusätzlich machen wir eine Analyse beim Kunden
     - *Lambda:* Entsprechend der Herkunft gleiche Behandlung, jedoch quantitative Erfassung einer einzelnen Nachfrage (Wie oft kommt die gleiche Anforderung? ->

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

wichtiger oder unwichtiger?). Generell Anforderungen vom Kunden, aber auch gesetzliche Vorgaben.
  - *Zeta:* In der Regel kommen die Anforderungen aus den Bereichen Service, Vertrieb, Kunden, interne Angelegenheiten. Der Projektleiter definiert bzw. sammelt die Anforderungen. Die Herkunft der Anforderung spielt im Moment keine Rolle für die Behandlung von dieser Anforderung

4. Wer ist am Prozess der Verwaltung von Anforderungen beteiligt?
   - Automotive
     - *Gamma:* Zwei Produktmanager als Schnittstelle zum „Kunden“ (Fachbereich) und das Gremium Projektteam.
     - *Delta:* Beim alten Prozess: Wir (zentrale Stelle) haben das Tracking der Anforderungen übernommen, in Excel. Beim neuen Prozess: Jeder Bereich (dezentral) trackt die Anforderungen in Excel selbst. Für SAP Anforderungsmanagement wird kein Tool(außer Excel) benutzt.
     - *Epsilon:* Das macht das Projekt, d.h. die Fachabteilung
     - *Iota:* Wir verwalten die Anforderungen.
     - *Theta:* Das machen wir. Auf unsere Doors-Datenbank haben aber auch andere Kollegen Zugang
   - Software
     - *Alpha:* Anforderungsanalysten und Reviewer.
     - *Beta:* Das wurde von Ausführungsteam mit Hilfe von einfacher Exel-Liste verwaltet.
     - *Eta:* Zentralstelle von jeweiligen Ländern
     - *Kappa:* An der Verwaltung von Anforderungen ist ein Anforderungsmanagement-Team beteiligt. Projektleiter und Systemtechniker spielen dabei auch eine wichtige Rolle.
     - *Ny:* Der Projektleiter hat die Aufgabe, den Vertrag mit dem Kunden zu verwalten. Er muss die Kosten, das Budget, die Qualität, das Change Management verwalten.
     - *Xi:* Projektleiter macht die Grobplanung, die Entwickler die Konkretisierung
     - *Omikron:* In der Anforderungsdefinition (Lastenheft) ist der Kunde der Hauptverantwortliche (und somit Prozessverwalter), Omikron unterstützt ihn hierbei.
     - *Pi:* Die Verwaltung von Anforderungen ist das ganze Entwicklungsteam beteiligt. Der Projektleiter ist für den gesamten Prozess der Anforderungsverwaltung zuständig. Das Team teilt die Arbeit nach fachlichen Gesichtspunkten.
   - Medizintechnik
     - *Rho:* Der Produktmanager muss dieses Thema betreuen
     - *My:* -
     - *Lambda:* Marketing Team, das Kundenanforderungen aufnimmt und mit dem Engineering-Bereich diskutiert.
     - *Zeta:* Der Projektleiter ist dafür zuständig, die Anforderungen zu definieren, sie zu bewerten usw. Er macht die Verwaltung. Die Größe des Projektes bestimmt, wer bzw. wie viele Personen an der Verwaltung von Anforderungen beteiligt sind

5. Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht?
   - Automotive
     - *Gamma:* Zu größtem Teil von Entwicklern (relativ spät im Prozess -> fehlerbehaftet),Fachabteilungen sind nur bedingt beteiligt.
     - *Delta:* In Kooperation: Fachbereich arbeitet mit, um die Anforderung genau zu verstehen. Es gibt ein Template der ausgefühlt werden soll zu Anforderungsdefinition.
     - *Epsilon:* Es wird in Kooperation zwischen Fachbereich und IT-Bereich gemacht. Bei einer Anforderung sind immer zwei Bewerter eingetragen, ein Fachbewerter und ein

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

IT-Bewerter. Wenn der IT-Bewerter die Anforderung nicht versteht, so geht sie zurück zum Fachbewerter, der sie umformulierten muss.
  - *Iota:* -
  - *Theta:* -
- Software
  - *Alpha:* Die externen Anforderungen werden in natürlicher Sprache verfasst und müssen nicht übersetzt werden.
  - *Beta:* Wir. Die Anforderungen wurden Aufgenommen, dann inhaltlich beschrieben. Dann wurde diese Beschreibung vom Kunden abgenommen. Dann haben wir eine technische Beschreibung gemacht. Diese technische Beschreibung wurde dann an die Programmierer gegeben. Das fachliche Team hat das gemacht. Für die technische Übersetzung hat das das technische Team gemacht.
  - *Eta:* -
  - *Kappa:* Der Bedarf des Kunden wird vom Anforderungsmanagements-Team in die Anforderungen übersetzt. Die Anforderungen werden genau aufgeschrieben.
  - *Ny:* Der Projektleiter und sein Team übersetzen den Bedarf des Kunden in die Anforderungen. Das team besteht aus Software-Architekten, Software-Beratern
  - *Xi:* Ausschließlich Aufgabe des Projektleiters.
  - *Omikron:* In Zusammenarbeit mit dem Kunden
  - *Pi:* Die Anforderungen sind in natürlicher Sprache beschrieben. Die Anforderungen werden in spezielle dafür vorgesehene Formen eingetragen. Dadurch erfolgt die Übersetzung der Anforderungen. Die Anforderungen werden konkretisiert.
- Medizintechnik
  - *Rho:* Die Anforderungen des Lastenhefts wird von uns ins Pflichtenheft übersetzt
  - *My:* -
  - *Lambda:* Das Upstream-Marketing, das mit der Entwicklung verzahnt ist.
  - *Zeta:* Der Projektleiter ist dafür zuständig, die Vorstellungen in die Anforderungen zu übersetzen. Es gibt ein Tool (R-Tool), in das die Anforderung eingegeben wird. Wenn diese Anforderung weitergeleitet wird, zum Beispiel, an den Softwareentwickler, dann wird diese Anforderung wieder toolunterstützt übersetzt.

6. Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu?
   - Automotive
     - *Gamma:* Etwa 60% des Gesamtumfangs sind durch Weiterentwicklung entstanden, 40% standen initial fest.
     - *Delta:* Auch nach der initialen Inbetriebnahme eines SAP-System kommen noch viele Anforderungen dazu. Vor allem feingranulare Anforderungen kommen später noch dazu.
     - *Epsilon:* -
     - *Iota:* projektabhängig
     - *Theta:* Echte Änderungen an Anforderungen (inhaltliche Änderungen): 10%
   - Software
     - *Alpha:* projektspezifisch
     - *Beta:* Am Anfang wurde eine fachliche Anforderungssammlungsphase gemacht. Dort wurde gesammelt und Priorisiert und dann festgelegt, welche Anforderungen realisiert werden. Aus der Sicht des Realisierungsaufwandes haben sich ca. 15% bis 20% der Anforderungen geändert.
     - *Eta:* Die ursprünglichen fachlichen Komponenten haben sich nach der ersten Festlegung nicht mehr verändert. Auch die Architektur hat sich nicht mehr verändert.
     - *Kappa:* Bei den Projekten, die vom Bund kommen, sind alle Anforderungen vorgegeben. Ansonsten ist es projektspezifisch.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *Ny:* Projektspezifisch. (z.B. in betrachtet in Umfrage Projekt: am Anfang wurde ca. 60-70 % der Anforderungen bekannt)
- *Xi:* Etwa 30-40% zusätzliche Anforderungen, aber auch steigende Komplexität der bekannten Anforderungen (etwa um den Faktor 2), daraus resultierend: stark steigendes Volumen.
- *Omikron:* Etwa 10-20% kamen hinzu
- *Pi:* Ca. 90% der Anforderungen sind schon zu Beginn der Entwicklung bekannt

- Medizintechnik
  - *Rho:* Nachdem Lasten- und Pflichtenheft festgeschrieben worden sind, sind die Anforderungen sehr stabil. Es gibt dann nur mehr sehr wenige Änderungen.
  - *My:* -
  - *Lambda:* Projektspezifisch, aber nicht nur hinzukommende Anforderungen, sondern auch wegfallende oder aufgeschobene
  - *Zeta:* Von den Initialanforderungen bleiben ca. 40% der Anforderungen übrig

7. Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt?
   - Automotive
     - Gamma: Je nach Aufwand, Ressourcen und damit Budget, Priorisierung der Anforderungen.
     - *Delta:* -
     - *Epsilon:* -
     - *Iota:* keine allgemeine Aussage möglich. Wir haben einen Issue-Change-Prozess. Dort ist definiert, wie mit Change-Requests umgegangen wird.
     - *Theta:* Projektgremien müssen entscheiden: Faktoren: Entwicklungskosten, Stückkosten, Zeit, Risiken. Lieferanten bewertet Kosten. Lastenheft wird falls nötig aktualisiert.
   - Software
     - *Alpha:* Hängt davon ab, wie umfangreich die Anforderungsänderungen sind. Meistens werden sie übernommen und implementiert. Der Umfang der Anforderung bestimmt den Betrag, der vom Auftraggeber verlangt wird.
     - *Beta:* Bei den Features die nicht so umgesetzt werden konnten, ging es oft um Tradeoffs: Wir haben gesamt, wir können es so machen, aber dann dauert die Analysefunktion im System z.B. 2 Minuten. Da diese 2 Minuten den Performance-Anforderungen nicht genügen, mussten wir das mit dem Kunden besprechen und das Feature dann ändern.
     - *Eta:* Die Märkte führen eine Vorpriorisierung der Anforderungen durch. Jeder Markt muss seine Änderungen, die er will selbst bezahlen, deshalb wird auch gut priorisiert.Die vorpriorisierten Anforderungen werden zentral gesammelt und in einer Runde wird entschieden, welche umgesetzt werden und welche nicht.
     - *Kappa:* Ein wichtiger Aspekt des Anforderungsmanagements ist sein modularer Aufbau. Falls eine Anforderung während der Entwicklung dazukommt, soll sie im Zusammenhang mit dem schon entwickelten System betrachtet werden. D. h. die Auswirkungen der dazugekommenen Anforderung auf das zu dem Zeitpunkt entwickelte System soll analysiert und überprüft werden.
     - *Ny:* Die Leistungsbeschreibung des Projektes legt fest, wie der Änderungsprozess aussieht
     - *Xi:* Es wurde eine Liste (Excel) aller Änderungen geführt, die mit dem Kunden durchgesprochen werden. Je nach Änderung wurde hierbei entschieden, ob eine Änderung durchgeführt werden soll (entsprechender Change Request) und zu wessen Lasten sie gehen soll.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

  - *Omikron:* Abhängig von der Phase sind die damit verbundenen Kosten- und Zeitfragen. Mit ihnen stehen und fallen Entscheidungen über die Umsetzung einer Änderung. Ideal sind Streichungen von Anforderungen oder aufwandsneutrales Tauschen
  - *Pi:* Sowohl wenn neue Anforderungen kommen als auch wenn sich die Anforderungen ändern, soll ein Change Request - Verfahren erfolgen. Im Rahmen dieses Verfahrens wird ein Gremium einberufen, das entscheiden soll, ob die Änderungen umgesetzt werden soll. Es wird unterschieden, ob es um eine komplett neue oder eine geänderte Anforderung geht.
- Medizintechnik
  - *Rho:* Die neuen Anforderungen werden auch teilweise umgesetzt. Im Prozess wird entschieden welche Anforderungen umgesetzt werden. Zurzeit stellen wir den Prozess auf Scrum um, um flexibler zu sein.
  - *My:* Es muss zusammen mit dem Kunden entschieden werden, welche zusätzlichen Funktionen bezahlt werden müssen, und welche schon innerhalb des ursprünglichen Rahmens realisiert werden können.
  - *Lambda:* Nein, Upstream-Marketing entscheidet individuell; eventuell auch Umsetzung zu späterem Zeitpunkt
  - *Zeta:* Diese Anforderungen werden aufgenommen. Der Projektleiter soll aber die Konsequenzen davon abwägen. Der Projektleiter soll dann entscheiden, ob die Anforderungen umgesetzt werden.

8. Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...)
   - Automotive
     - *Gamma:* Funktionale Anforderungen sind recht früh bekannt, nicht funktionale Anforderungen entstehen meist im Entwicklungsprozess.
     - *Delta:* Es kommen alle Arten von Anforderungen hinzu. Es wird ein Kategorisiert um den Überblick zu behalten: z.B. Fachlichkeit, GUI, Nicht-funktional, usw. oder Kategorisierung nach Unternehmensprozessen. Eine einheitliche unternehmensweite Kategorisierung(projektspezifisch) gibt es nicht.
     - *Epsilon:* -
     - *Iota:* Kann man nicht einem Bereich einordnen: technisches Bereich, UI, Funktionalität.
     - *Theta:* Technische Anforderungen, Nutzeroberfläche
   - Software
     - *Alpha:* Die meisten beziehen sich auf die Funktionalität.
     - *Beta:* -
     - *Eta:* Vor allem Funktionale Anforderungen kommen neu dazu
     - *Kappa:* Die meisten Anforderungen, die später hinzukommen, sind funktionale Anforderungen
     - *Ny:* Primär sind das funktionale Anforderungen. Die meisten Änderungen kommen von der Fachabteilung. Die Änderungen im funktionalen Bereich betragen ca. 2/3 aller Änderungen. Der Rest sind die Änderungen im nicht-funktionalen Bereich.
     - *Xi:* Hauptsächlich nicht-funktionale Anforderungen
     - *Omikron:* Hauptsächlich funktionale Anforderungen
     - *Pi:* Es kommen oft fachliche Details. Es sind oft auch nichtfunktionale Anforderungen
   - Medizintechnik
     - *Rho:* Es kommen vor allem funktionale Anforderungen dazu. Bei nicht-funktionalen Anforderungen kommen vor allem Anforderungen bezüglich der Ergonomie.
     - *My:* -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *Lambda:* Funktionalität und Nutzeroberfläche, Qualitätsanforderungen werden nie als zusätzliche Anforderungen betrachtet, sondern als konstant gefordert
- *Zeta:* Funktionalität.

9. Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen?
   - Automotive
     - *Gamma:* -
     - *Delta:* Ist sehr schwer zu sagen. Wir sind dafür verantwortlich kontinuierlich die Anforderungen an die SAP-Systeme zu verwalten.
     - *Epsilon:* -
     - *Iota:* Unterschiedlich
     - *Theta:* Schwer zu sagen
   - Software
     - *Alpha:* Projektspezifisch
     - *Beta:* Der Aufwand wird über das Vorgehensmodell geschätzt. Es wird die Anzahl der Eingabemasken, Datenbankzugriffe usw. gezählt. Dann wird ein Testaufwand usw. hinzugegeben und so ergibt sich der Aufwand. Die Abhängigkeiten werden auch berücksichtigt. Eine bestimmte Anforderung kann nur realisiert werden, wenn auch bestimmte andere Anforderungen realisiert werden. Es werden dann Pakete angeboten. Das sind so eine Art „Ausstattungspakete".
     - *Eta:* Während der Entwicklung: Relativ viele Change-Requests. Projektphase abhängig, in früheren Phasen viel Änderungen und Aufwand.
     - *Kappa:* Je umfangreicher das Projekt ist, desto größer ist der Aufwand für die Änderungen an Anforderungen.
     - *Ny:* Projektspezifisch. Es wird aber schon beim Schließen des Vertrags genaue Kostenrahmen festgelegt.
     - *Xi:* Sämtliche Änderungen in diesem Projekt haben nur unbedeutenden Mehraufwand verursacht, die Anforderungen waren relativ stabil.
     - *Omikron:* -
     - *Pi:* Je später im Laufe der Entwicklung eine Anforderung dazu kommt, desto schwieriger ist es, sie zu integrieren. Besonders schwierig ist es, wenn das IT-Konzept abgeschlossen ist und dann noch nichtfunktionale Anforderungen kommen
   - Medizintechnik
     - *Rho:* Nicht mehr als 10%. Das ist eine grobe Schätzung anhand unserer Meilensteine
     - *My:* -
     - *Lambda:* Sehr hoher (aber projektspezifischer) Aufwand, da zusätzliche Iterationen
     - *Zeta:* Projektspezifisch. In dem betrachteten Projekt war der Aufwand sehr groß.

10. Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?
    - Automotive
      - *Gamma:* Template (ein DIN A4). für Änderungsanträge mit wichtigsten Informationen: von wenn angereicht, wann und so weiter bis voraussichtlicher Kostenaufwand. Es gibt Change Requests, aber speziell im späteren Projektverlauf ist die Dokumentierung absolut mangelhaft bzw. nicht mehr vorhanden. Das Team entscheidet, ob ein Change ein eigenes Lastenheft bekommt und somit als Art eigenes Projekt behandelt wird. Was ist Ziel: alle Änderungen zu einem Antrag bündeln und in ein Lastenheft überführen der in System Lastenheft eingeht. Damit Nachvollziehbarkeit geschaffen wird.
      - *Delta:* -
      - *Epsilon:* Spezielle Werkzeug für Änderungsmanagement: Bewertung, Entscheidung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *Iota:* Es wird ermittelt wie viel die Änderung Zeit und Geld kostet. Es wird dann mit dem Kunden kommuniziert. Der Projektleiter entscheidet letztendlich was gemacht wird
- *Theta:* Das Lastenheft wird immer aktuell gehalten

- Software
  - *Alpha:* Neues Arbeitspaket wird aufgemacht.
  - *Beta:* -
  - *Eta:* Alle Änderungen von Anforderungen werden Dokumentiert. Es gibt eine vollständige Nachvollziehbarkeit zwischen Anforderungen.
  - *Kappa:* Falls sich eine Anforderung ändert, wird ihr Zusammenhang mit dem gesamten Kontext genau überprüft. Erst dann kann man die Änderungen, die sich durch die Anforderung ergibt, umsetzen.
  - *Ny:* Falls sich eine Anforderung ändert wird vom Kunden diese Änderung übersetzt und an die Entwickler weitergeleitet. Der Change Management-Prozess hat eine vorgegebene Struktur. Das Unternehmen hat eine spezielle Change Management-Methode. Die Prozesse sollen eine Tool-Unterstützung haben.
  - *Xi:* Bei neutralem Aufwand trifft der Projektleiter die Entscheidung, bei höherem Aufwand muss der Kunde das Delta verantworten und akzeptieren. Sollte der Aufwand durch eine Änderung sinken, so wird das Delta als Puffer im Gesamtprozess genutzt
  - *Omikron:* Change Requests bei Änderungen, die Dokumentierung erfolgt über die Versionierung in Excel Tabellen.
  - *Pi:* Falls sich eine Anforderung ändert, wird sie in ein Gremium eingebracht uns das Gremium soll entscheiden, ob die Anforderung umgesetzt wird. Dabei sollen auch die Entscheidungen bezüglich der Kosten getroffen werden, die durch die Änderung entstehen. Der andere Aspekt ist die Zeit. Die Frage ist, ob die Durchführung des Projektes durch die Änderung bzw. das Hinzufügen einer Anforderung beeinflusst wird. In diesem Fall soll entschieden werden, wie sich die Realisierungszeit dadurch ändert. Falls es eine Anforderung kommt, soll entscheiden werden, ob es sich um eine funktionale oder nichtfunktionale Anforderung und ob es sich um eine komplett neue Anforderung oder eine Änderung einer Anforderung handelt. Anschließend kommt die Anforderung in das Change Request Verfahren. Das Gremium soll dann entscheiden, wie die Anforderung zu behandeln ist
- Medizintechnik
  - *Rho: Es gibt ein Tracing*
  - *My:* -
  - *Lambda:* Bewertungsschema bzgl. Relevanz einer neuen Anforderung (bspw. Sicherheitsrelevanz).
  - *Zeta:* Der Projektleiter ist dafür zuständig, die Änderungen an Anforderungen entsprechend zu dokumentieren. Der Lenkungsausschuss entscheidet über die Änderungen. Der Projektleiter ist dafür zuständig, die Änderungen zur Realisierung freizugeben. Der Lenkungsausschuss entscheidet über die Änderungen. Der Lenkungsausschuss tagt regelmäßig. Die geänderten und dazugekommenen Anforderungen werden gesammelt und dem Lenkungsausschuss präsentiert. Die Mitglieder des Lenkungsausschusses sind nicht vordefiniert, sonder es wird eine Entscheidung projektabhängig getroffen. Der Projektleister und die Geschäftsleitung entscheiden, wer zum Lenkungsausschuss gehören soll, es können auch Externe dabei sein.

11. Von wem kommen die Änderungen?
    - Automotive

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *Gamma:* Zu etwa 80% aus den Fachbereichen, zu 20% aus der IT oder Schnittstellenthematik
- *Delta:* -
- *Epsilon:* -
- *Iota:* Von Kunden oder Lieferanten.
- *Theta:* Änderungen können von Intern und auch Lieferanten kommen

- Software
  - *Alpha:* -
  - *Beta:* -
  - *Eta:* Von Überall: Kunden, Lieferanten usw
  - *Kappa:* Es ist immer unterschiedlich. Im Wesentlichen ist der Kunde für die Änderungen zuständig. Auch interne Änderungen sind möglich. Ein Change Request im Rahmen des V-Modell XT ruft immer eine Iteration im Vorgehensmodell auf.
  - *Ny:* Die Änderungen kommen entweder vom Kunden oder das Unternehmen weist dem Kunden darauf hin, dass eine Anforderung geändert werden soll
  - *Xi:* Vom Auftraggeber, Nur bei Umsetzungsproblemen, die auf Inkonsistenzen der Anforderungen beruhten, musste Xi die Initiative ergreifen
  - *Omikron:* Bedingt durch Prozessveränderungen beim Kunden
  - *Pi:* Der Kunde hat eine Fachabteilung, die für das Aufstellen der Anforderungen zuständig ist. Dabei wird es zwischen zwei Arten der Fachabteilung unterscheiden: das sind zum Einen die Endnutzer und die Prozessabteilungen, die sich mit dem Konzept der Entwicklung der Lösung beschäftigen
- Medizintechnik
  - *Rho:* Normative Anforderungen kommen von der Qualitätsabteilung. Andere Anforderungen kommen vom Produktmanagement.
  - *My:* -
  - *Lambda:* Kunde, Vertrieb, Technik, Marketing, Entwicklung, Gesetzgeber
  - *Zeta:* Die Änderungen in dem Projekt sind von extern gekommen. Allgemein auch von der Geschäftsführung. Viele Projekte laufen parallel und haben enge Zusammenhänge. Die Änderung an einem Projekt führt zu den Änderungen in weiteren Projekten.

12. Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem?

- Automotive
  - *Gamma:* Ja, in den entsprechenden Prozessphasen (siehe Prozessbeschreibung), allerdings besteht hier enormes Verbesserungspotenzial, z.B. wird Lastenheft erst durch IT erstellt
  - *Delta:* -
  - *Epsilon:* ja
  - *Iota:* Die Begriffe sind etwas verschwommen. Manchmal wird ein Dokument als Pflichtenheft bezeichnet, obwohl es aber ehr ein Lastenheft ist. Ein Lastenheft gibt es eigentlich nicht. Die Projektidee stellt das Lastenheft dar
  - *Theta:* Ja, Lastenhefte werden erstellt. Zuständig: der Zuständige fürs jeweilige Steuergerät, der Systemverantwortliche. Pflichtenheft: Wird vom Lieferanten als Antwort auf unser Pflichtenheft erstellt
- Software
  - *Alpha:* -
  - *Beta:* Am Anfang wurde eine fachliche Dokumentation erstellt, in Word. Um dann wurden die Anforderungen so gesammelt, wie ich schon beschrieben hatte.
  - *Eta:* Am Anfang der Entwicklung wurden ein Grobkonzept, dann ein Feinkonzept und dann ein IT-Konzept erstellt

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *Kappa:* Das Lastenheft entspricht der Beschreibung der Leistungen, die das Unternehmen dem Kunden erbringen muss. Im Lastenheft werden die initialen Anforderungen festgehalten. Im Pflichtenheft werden die Anforderungen ausformuliert. Das wird zusammen mit dem Kunden gemacht
- *Ny:* -
- *Xi:* -
- *Omikron:* Anforderungsdefinition (Lastenheft), Softwarespezifikation (fachlich und technisch), Entwurf (Pflichtenheft) → Alles Bestandteil der Konzeptionsphase.
- *Pi:* -

- Medizintechnik
  - *Rho:* Lastenheft vom Produktmanagement. Pflichtenheft von uns, dort halten wir die technischen Anforderungen fest.
  - *My:* Pflichtenheft kommt in einem vorgelagerten Schritt vor. In diesem Schritt wird analysiert, ob die Standardsoftware überhaupt im jeweiligen Krankenhaus eingesetzt werden kann.
  - *Lambda:* Interne Pflichtenhefte, die Entwicklungsabschnitte (Milestones) definieren und ihre zeitliche Umsetzung festhalten
  - *Zeta:* Es wird nicht genau unterschieden. Es gibt aber ein Word-Dokument, in dem die Anforderungen in Fließtext beschrieben sind. Im R-Tool ist auch schon ein Teil der Realisierung beschrieben. Dort werden die Anforderungen formaler beschrieben.

13. Welche Informationen sind im Pflichtheft enthalten? Welche Informationen sind im Lastenheft enthalten?

- Automotive
  - *Gamma:* Teilweise eigenes Lastenheft für einzelne Anforderungen (je nach Mächtigkeit). In der Vergangenheit wurde diese dann zu einem einzigen Lastenheft gebündelt, was sich allerdings als sehr unglücklich herausstellte, da die Inhalte teils sehr konträr waren (andere Perspektiven auf Anforderungen). Zusammengehörige Anforderungen waren ohne Reihenfolge eingebunden -> Verbesserung zu integriertem Systemlastenheft angestrebt. Pflichtenheft enthält konkretisierte Anforderungsdokumentation.
  - *Delta:* -
  - *Epsilon:* Lastenheft = Fachkonzept, Pflichtenheft = DV-Konzept
  - *Iota:* Der Schwerpunkt ist das Pflichtenheft. Die Ideen werden in einem Pflichtenheft festgehalten, um dann eine Ausschreibung für die Realisierung zu machen. Ein Lastenheft gibt es eigentlich nicht. Die Projektidee stellt das Lastenheft dar
  - *Theta:* Lastenheft: gemäß DIN: geschrieben von Auftraggeber. Exakte Menüabläufe. Buskommunikation wird beschrieben. Pflichtenheft: gemäß DIN: geschrieben von Auftragnehmen. Noch eine Stufe feiner detailliert. Ist schon praktisch eine Modulspezifikation. Die Realisierung wird genau beschrieben
- Software
  - *Alpha:* -
  - *Beta:* -
  - *Eta:* -
  - *Kappa:* Es hängt vom Projekt ab. Sowohl das Lastenheft als auch das Pflichtenheft werden vom Unternehmen geschrieben. Das Lastenheft enthält Beschreibungen, die vom Kunden kommen. Das Pflichtenheft enthält genauere Informationen über die Anforderungen. Dafür werden die Informationen im Lastenheft ausformuliert.
  - *Ny:* -
  - *Xi:* -
  - *Omikron:* Das Pflichtenheft enthält technische Lösungsbeschreibung
  - *Pi:* -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- Medizintechnik
  - *Rho:* Pflichtenheft werden die Konzepte festgehalten. Im Lastenheft stehen die initialen Anforderungen an das Produkt.
  - *My:* Im Pflichtenheft legt der Kunde seine Anforderungen an das System fest.
  - *Lambda:* -
  - *Zeta:* -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang B.2 Anforderungsermittlung und Anforderungsanalyse

1. Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z.B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)?
   - Automotive
     - *Gamma:* Workshops, Besprechung von Änderungen/Neuanforderungen in Projektteamsitzung
     - *Delta:* Keine proaktive Suche nach Anforderungen
     - *Epsilon* Meetings Manchmal Moderation durch erfahrene Projektleiter Erfassung abteilungsabhängig in Excel bzw. Word oder mit Tools
     - *Iota:* Prozess-Analyse, Workshops (Besprechungen*)*
     - *Theta:* Besprechungen: d.h. Workshops
   - Software
     - *Alpha:* Interview, Workshop, Diskussion, Brainstorming
     - *Beta:* Workshops, strukturierte Interviews
     - *Eta:* Workshop wo das bestehende Produkt diskutiert wird. Es werden auch Interviews durchgeführt
     - *Kappa:* Interviews und Workshops
     - *Ny:* Oft werden die Anforderungen zusammen mit dem Kunden definiert. Es werden Workshops durchgeführt
     - *Xi:* einwöchiger Workshop
     - *Omikron:* Workshop mit IT Fachbereichen
     - *Pi:* Workshops, Interviews mit dem Kunden, im Rahmen des Projektes- mit den Fachabteilungen, Prototypen, Eine andere Möglichkeit für die Ermittlung von Anforderungen bietet die Analyse der Dokumentation von schon bestehenden Systemen
   - Medizintechnik
     - *Rho:* Feldbeobachtung, Workshops, Befragungen, Prototypen
     - *My:* Interviews, Workshops, Begehungen, bestehende Dokumente, Checklisten
     - *Lambda:* Interne Workshops, Kundenevents, Vorstellung einer geplanten Umsetzung mit anschließender Einholung von Feedback
     - *Zeta:* Die Anforderungen werden erhoben und in einem Spezifikationsdokument aufgeschrieben. Der Projektleiter initiiert Workshops und befragt zu den Anforderungen
2. Was sind die häufigsten Probleme, die bei Ermittlung von Anforderungen auftreten?
   - Automotive
     - *Gamma:* Hauptsächlich schlechte und unqualifizierte Kommunikation
     - *Delta:* Aufwand: Viele Stellen und Absprachen liegen im Prozess, viel Zeit vergeht, abhängig von vielen Unbekannten, geringe Erfolgsquote bei Standardisierungsprozess der Anforderungen
     - *Epsilon:* Mangelnde Präzisierung von Anforderungen, nicht klare Zusammenhänge also mangelnde Informationsintegrität durch schlechte Kommunikation
     - *Iota:* Richtigen Gesprächspartner identifizieren und von ihm Informationen bekommen. Feedback von Kunden schwer zu bekommen erst nach erstem Prototyp
     - *Theta:* Identifizieren der richtigen Ansprechpartner, Die Kommunikation ist etwas schwierig
   - Software

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *Alpha:* Konflikte zwischen Anforderungen, der Auftraggeber kann nicht genau sagen, was er vom Produkt erwartet. Verschiedene Disziplinen (Interdisziplinäre Team), unterschiedliche Sicht von verschiedenen Kundengruppen
- *Beta:* Unterschiedliche Terminologie zwischen fachlichen und technischen Bereichen. Es werden Leute benötigt, die die fachlichen Dinge gut genug verstehen und gleichzeitig auch technische Lösungen vorschlagen können.
- *Eta:* Schlampige Anforderung, die man nicht richtig versteht. Ein Marktverantwortlicher eine Anforderung unbedingt durchsetzten will, welche sich aber auf Grund der Architektur nicht umsetzten lässt
- *Kappa:* Die Kunden oft über die Ziele des zu entwickelnden Systems nicht im Klaren sind
- *Ny:* Die Vorstellungen der Fachabteilung sind oft nicht in den Zeitrahmen und nicht in den Kostenrahmen zu realisieren
- *Xi:* Das häufigste Problem ist das Budget, sprich das Aufkommen für Änderungen
- *Omikron:* Verständnisprobleme, Begrifflichkeiten (Verständnis der Fachsprache), falsches Verständnis, Runter brechen auf Anforderungen (schlechte Formulierung)
- *Pi:* Das Problem besteht oft darin, unterschiedliche Ziele verschiedener Stakeholder unter einen Hut zu bringen

- Medizintechnik
  - *Rho:* Die Qualität der Anforderungen ist ein großes Problem. Die Anforderungen so aufzuschreiben, dass sie gut verständlich sind, ist schwer. Die Kunden liefern die Anforderungen nicht in der Form, in der wir sie brauchen, Die Internationalität ist auch ein Problem. Mit unserem Produkt gehen Leute mit verschiedenem Bildungshintergrund um
  - *My:* Verantwortliche, die sich gut auskennen, stehen nicht zur Verfügung
  - *Lambda:* Erstellen des Rankings von Anforderungen, Wirtschaftliche Abwägung von Anforderungen
  - *Zeta:* Bewertung der Priorität anhand der Projektzeit, -kosten

3. Wie werden die Anforderungen analysiert?(Unter Analyse versteht man die Konkretisierung von Anforderungen)
   - Automotive
     - *Gamma:* Verfeinerung durch Entwickler, teilweise mit Produktmanagern
     - *Delta:* Geht nicht ohne Iterationen. Grobkonzept erfolgt möglichst am konkreten System, nicht nur rohe Beschreibungen. IT-Konzept wird ohne Fachbereich von der IT geschrieben
     - *Epsilon:* Durch die Fachabteilungen, gegebenenfalls im Dialog mit dem Projektteam
     - *Iota:* Strukturiert wird anhand von Templates. Es gibt z.B. ein Template für Use-Cases
     - *Theta:* -
   - Software
     - *Alpha:* Während der Formulierung des Lastenhefts und in Workshops
     - *Beta:* Die Anforderungen werden beschrieben: WAS soll passieren und WIE soll es passieren. Und die Zusammenhänge zwischen den Anforderungen werden beschrieben. Der Ablauf wird erstellt. Das Datenmodell wird erstellt, mit genauen Angaben von Wertebereichen usw.
     - *Eta:* Persönliche Kommunikationsfähigkeit der Person die das durchführt
     - *Kappa:* Komplexe Anforderungen sind immer zu verfeinern. Ein wichtiger Aspekt ist die Anforderungsgruppierung. Die Analyse von Anforderungen erfolgt durch ihre ausführliche Beschreibung, es wird eine Spezifikation erstellt
     - *Ny:* Das initiale Anforderungsdokument wird in eine DV-Spezifikation übersetzt. Oft werden dafür Use Cases verwendet

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

  - *Xi:* Dies geschieht in Zusammenarbeit mit dem Kunden, Etwa 80% der Anforderungen werden durch den Projektleiter geklärt, die Klärung der Detailfragen obliegt den Entwicklern
  - *Omikron:* zusätzlich Termine vereinbart, um die Ist-Prozesse vor Ort zu betrachten und zu analysieren. Dabei werden die Mitarbeiter befragt und hieraus zu lösende Probleme abgeleitet
  - *Pi:* Das Anforderungsteam hat die Aufgabe, die ermittelten Anforderungen zu konkretisieren. Die Anforderungen sind in natürlicher Sprache beschrieben. Die Anforderungen werden in spezielle dafür vorgesehene Formen eingetragen. Dadurch erfolgt die Übersetzung der Anforderungen. Die Anforderungen werden konkretisiert
- Medizintechnik
  - *Rho:* Manchmal werden Kunden eingeladen und mit denen wird das Lasten- und Pflichtenheft diskutiert
  - *My:* -
  - *Lambda:* Wesentlich ist die Zerlegung der Anforderungen, damit diese atomar diskutiert werden können. Hierfür auch Einbindung von Schlüsselkunden.
  - *Zeta:* Das ist so zu sagen eine Datenbank, die die ausformulierten Anforderungen enthält. So eine Anforderung wird in natürlicher Sprache beschrieben. Es gibt aber keine genauen Vorgaben, was sie enthalten soll. Das Tool gibt schon einiges vor, welche Angaben gemacht werden sollen.

4. Werden die Anforderungen priorisiert? Falls ja, wie?
- Automotive
  - *Gamma:* Priorisierung der komplexen Anforderungen im Projektteam (zu welchem Release, Kosten) mit Nummerierung
  - *Delta:* Stufen 1 (MUST have!) bis 6 (nice to have); kein standardisiertes Vorgehen; individuelle Kategorisierung
  - *Epsilon:* Priorisierung entlang der Leistungsstufe (welche Features müssen in nächster Leistungsstufe enthalten sein?) Bewertung durch Fachabteilung und IT (schwerpunktmäßig) nach standardmäßigen Kriterien wie Kosten, Komplexität etc.
  - *Iota:* Wir versuchen die Priorität (nach Datum) herauszufinden
  - *Theta:* Nur: Ja/Nein, D.h. Anforderungen können abgelehnt/angenommen werden
- Software
  - *Alpha: Ja:* muss, soll, kann
  - *Beta:* Bei der Definition der Anforderungen werden der Aufwand und der Nutzen hinzugegeben. Rechtliche Anforderungen müssen immer eingehalten werden
  - *Eta:* Ja, Jeder Markt muss seine Änderungen, die er will selbst bezahlen, deshalb wird auch gut priorisiert
  - *Kappa:* Die Anforderungen werden priorisiert. Dazu werden die Anforderungen nach verschiedenen Typen sortiert
  - *Ny:* das Ampel-System verwendet.
  - *Xi:* Prioritätsblöcke: 1. „sofort zu erledigen“ 2. „pending, auf Nachfrage“ 3. „unsicher, on hold“ oder Kundensystem
  - *Omikron:* Priorisierung erfolgt nur bei optionalen oder konkurrierenden Anforderungen zur Schmälerung des Katalogs auf den Teil der Anforderungen, der letztendlich umgesetzt wird
  - *Pi:* Die Priorisierung ist kundenabhängig. Dabei können die Anforderungen nach muss / soll / kann aufgeteilt werden. Eine andere Möglichkeit besteht darin, dass die Priorisierung auf der Basis gesetzlicher Vorschriften oder der finanziellen Aspekte erfolgt
- Medizintechnik
  - *Rho:* Die Anforderungen werden priorisiert nach: muss, soll, kann

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *My:* Ja, es wird priorisiert. Aus Kostengründen werden dann auch Anforderungen weggelassen. Man einigt sich dann mit dem Kunden auf einen Katalog von Anforderungen
- *Lambda:* Bemessungsfaktoren: Kundennutzen, Umsatzpotenzial, Realisierbarkeit mit vorhanden Entwicklungsressourcen, auch zeitliche Einstufungen
- *Zeta:* Die Priorisierung kann auch im Tool vorgenommen werden. Der Projektleiter gibt an, welche Priorisierung er für nötig hält. Die Priorisierung erfolgt nach klassischem Schema: muss / soll / kann

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang B.3 Anforderungsdokumentation

1. Wie werden die Anforderungen dokumentiert? Gibt es ein Muster(Vorlage), um Anforderungen zu dokumentieren?
   - Automotive
     - *Gamma:* Es gibt Templates vom VW Konzern. Anforderungen werden hier beschrieben und falls nötig soweit nachformalisiert, dass sie von beiden Seiten eindeutig verstanden werden
     - *Delta:* Es gibt Templates, aber Dokumentation den Projekten freigestellt
     - *Epsilon:* Fachkonzept und DV-Konzept (letzteres nicht immer, da oft externer Einkauf der Lösung) strenger Leitfaden und Templates als Vorlage
     - *Iota:* Seit 1. Juni gibt es ein Template für Pflichtenhefte. Vorher haben wir es auch schon ähnlich gemacht. Aber jetzt haben wir einen Standard
     - *Theta:* Es gibt ein Template für Lastenhefte. Das ist aber mehr als eine Vorlage. Es ist eine Anleitung von 500 Seiten, wo schon viele Standardanforderungen festgelegt sind
   - Software
     - *Alpha:* Lastenheftvorlagen, Use-Cases, Schablonen, Satzschablonen, UML
     - *Beta*: In einfachen Excel-Listen verwaltet. Dort wurde festgehalten: Von wem, was betrifft es, welcher Aufwand Es gibt auch eine Toolunterstützung für Anforderungen, so wie Projektmanagement-Tools. Bei dem Projekt haben wir das aber noch nicht eingesetzt
     - *Eta:* QualityCenter als durchgängiges Werkzeug zur Dokumentation der Anforderungen. Alle Märkte müssen dieses System einsetzen. Alle Änderungen an Anforderungen werden dokumentiert. Die Anforderungen sind komplett Nachvollziehbar.
     - *Kappa:* Die Anforderungen werden textuell beschrieben. Dazu gibt es vom Unternehmen entworfene Muster
     - *Ny:* Die Anforderungen werden in entsprechenden unstrukturierten Templates dokumentiert. Die Templates werden in den Gesprächen mit dem Kunden eingesetzt. Die Anforderungen werden in Fließtext in Word aufgeschrieben. Toolunterstützung findet in seltenen Fällen statt. Es gibt eine Repository, die angeben, wie Checklisten, Protokolle, Change Management etc. zu gestalten sind.
     - *Xi:* Ja, es gibt ein Muster. Die Dokumentation erfolgt über Excel.
     - *Omikron*: Die Dokumentation erfolgt verbal in Textdokumenten und tabellarisch mit Nummerierung über Excel. Obsolete und geänderte Anforderungen werden nicht oder in veränderter Form in höher versionierte Dokumente übernommen, bleiben jedoch in den alten so erhalten wie zuvor
     - *Pi*: Es gibt ein Anforderungsdokument, das die Anforderungen beschreibt, eine Spezifikation der Anforderungen (das Fachkonzept), die die fachlichen Aspekte beschreiben. Die genaue Umsetzung ist nicht der Teil der fachlichen Spezifikation. Dabei werden Geschäftsabläufe, elementare Funktionen, Daten, Benutzeroberflächen etc. beschrieben. Das IT-Konzept beschreibt, welche Aspekte wie umgesetzt werden sollen, zum Beispiel, welche Datenbanken oder Entwicklungsumgebungen eingesetzt werden müssen. Das IT-Konzept beschreibt die IT-Komponenten, die benötigt werden, um die Anforderungen zu realisieren. Das Fachkonzept stellt so zu sagen den Vertrag zwischen dem Unternehmen und dem Kunden dar
   - Medizintechnik
     - *Rho:* Es gibt eine Vorgabe, wie die Sätze formuliert werden müssen. Diese Vorgaben werden immer eingehalten. Die Testfälle werden gleich bei der Anforderung

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

angegeben. Teilweise wird auch UML verwendet. Aber das wird ehr schon auf der Seite der Realisierung gemacht
  - *My:* Es wird als durchgängiger Fließtext formuliert. Dieser Text hat jedoch eine Standard-Kapitelstruktur, die immer eingehalten wird
  - *Lambda:* -
  - *Zeta:* sie in einem Spezifikationsdokument festgelegt

2. Welche Informationen werden dokumentiert?
- Automotive
  - *Gamma:* Nur Verweis auf Lastenheft (Realisierungskonzepte)
  - *Delta:* Thema, Ansprechpartner, Release, Autor, Beschreibung, Reifegrad der Anforderung, Standardfähigkeit, Termin, Nutzen (Business Case), Priorität
  - *Epsilon:* in anderem Zusammenhang konkret nur Ersteller und Priorisierung genannt, ansonsten getreu der Fachkonzept- und DV-Konzept-Leitfäden
  - *Iota:* Nummer (Identifier), Owner, Priorität, Fließtext, Referenz auf Testfälle
  - *Theta:* Wir beschreiben die Anforderungen in Fließtext in Doors. Teilweise gibt es eigene Dateien, die von Doors aus referenziert werden. Teilweise wird Statemate und Simulink/Stateflow verwendet. Diese Modellierungen werden dann aber auch wie oben übernommen
- Software
  - *Alpha:* Namen, Vorbedingungen, Nachbedingungen, Änderungen an Anforderungen (nicht immer im Lastenheft), von wem kommt(Ansprechpartner)
  - *Beta:* Nummer, Name, Beschreibung, Kosten, Nutzen, Woher kommt sie(Quelle)
  - *Eta:* Nummer, Text, Priorität, zugeordnetes Release, Testfälle, Quelle – Ansprechpartner (wer bestellt hat), Fehler, falls in Abnahmetest sowie Änderungen an der Anforderung vorgenommen wurden
  - *Kappa:* Jede Anforderung hat eine eindeutige Nummer. Darüber hinaus werden die Beschreibung der Anforderung, die Änderung, der für die Anforderung zuständige Bearbeiter, die Quelle der Anforderung etc. dokumentiert. Im Falle einer Änderung bekommt die Anforderung eine Versionsnummer, die mit dem entsprechenden Change Request in Verbindung gebracht werden kann
  - *Ny:* Es wird angegeben, wer die Anforderung initiiert hat, der Inhalt der Anforderung (scope), an welchen Stellen die Anforderung eine Auswirkung auslöst, eine Terminplanung und die Implementierungsgrobbeschreibung aber nur auf der Architekturebene. Die Implementierungsgrobbeschreibung aber nur auf der Architekturebene wird dann im Laufe der Entwicklung ergänzt. Ergänzend wird auch eine Aufwandsschätzung gemacht (zum Beispiel Mann / Monat). Die Änderungen der Anforderungen werden auch dokumentiert. Der Kunde soll bestätigen, dass die aufgefassten Anforderungen seinen Erwartungen entsprechen
  - *Xi:* Datum, Ursprungsaufwand, Zusatzaufwand, Ansprechpartner bei o2, Verantwortlicher bei Xi, Deadline der Entscheidungsfindung, Zeitpunkt der Fertigstellung, Priorisierung der Anforderungen
  - *Omikron:* Beschreibung, Nummer, Verantwortlichkeit und Herkunft
  - *Pi:* Folgende Informationen werden aufgeschrieben. Das Anforderungsteam soll die Anforderungen dokumentieren. Dabei wird die Motivation für die Anforderung, der Titel der Anforderung, die Nummer, eine detaillierte Beschreibung der Anforderung, Abhängigkeiten zu den anderen Anforderungen, die Release Nummer, die Klassifizierung (soll / muss / kann). Die Anforderungen werden nach Prozessen dokumentiert
- Medizintechnik
  - *Rho:* Anforderung als Text, Kommentar
  - *My:* -

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *Lambda:* -
- *Zeta:* In der Dokumentation gibt es einerseits die reinen Anforderungen, andererseits auch technischere Spezifikationen, wo auch auf die Komponenten des Systems eingegangen wird

3. Wie wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nicht, warum nicht?

- Automotive
  - *Gamma:* Word, Excel, Powerpoint (Präsentationen vor Gremium), Eclipse für Entwickler, Mercury TestDirector, HP Quality Center, eigenes Budgetierungstool. Unterschiedliche Fachbereiche nutzen unterschiedliche Tools (Elektronikentwicklung beispielsweise Doors, das aus Sicherheitsgründen nicht von der Projekt-IT genutzt werden kann). Generell kann man sagen: Unterschiedliche Abteilungen benutzen unterschiedliche Tools für RE
  - *Delta:* Auf Dauer lässt sich nicht alles mit Excel tracken, deshalb gibt es Anstrebungen, ein zentrales Tool zu schaffen
  - *Epsilon:* Dokumentation nur über Word Allgemeine Verwaltung über Tools ist den Projektleitern freigestellt (nur ITPM ist BMW-weite Vorgabe)
  - *Iota:* Office. RequisitePro. Doors. Die Tendenz geht in Richtung Doors
  - *Theta:* Doors.
- Software
  - *Alpha:* Es werden mehrere Werkzeuge eingesetzt
  - *Beta:* RequisitePro ist das Standardtool bei Beta. Das wurde erst 2007 eingesetzt. Doors wird oft von Kunden verwendet
  - *Eta:* QualityCenter als durchgängiges Werkzeug zur Dokumentation der Anforderungen. Alle Märkte müssen dieses System einsetzen und Word
  - *Kappa:* Die Verwaltung von Anforderungen wird mit Microsoft Office-Tools unterstützt. Außerdem gibt es Tools, die vom Unternehmen entworfen wurden
  - *Ny:* Die Anforderungsdokumentation ist eine reine Beschreibung. Use Cases und die Datenverarbeitungskonzepte unterstützen die Dokumentation. Die Tools werden selten eingesetzt. Das liegt an dem Aufwand, der dadurch für den Kunden entsteht. Es geht dabei nicht um finanzielle Aspekte, aber auch um Schulungen der Mitarbeiter usw. Wenige Kunden wollen in die IT für die Unterstützung des Anforderungsmanagements investieren
  - *Xi:* Excel
  - *Omikron:* Nur Office Programme
  - *Pi:* Die Dokumentation von Anforderungen wird mit Office-Werkzeugen unterstützt. Andere Werkzeuge werden projektspezifisch eingesetzt
- Medizintechnik
  - *Rho:* RequisitePro (?), CaliberRM
  - *My:* -
  - *Lambda:* Nur Excel, aber weitere nicht ausgeschlossen
  - *Zeta:* Es gibt das R-Tool in den die Anforderungen festgehalten werden. Das Tool ist gleichzeitig die Dokumentation des Produkts. Denn in der Medizintechnik gibt es genaue Vorschriften, wie die Dokumentation inklusive Unterschriften und Freigaben aussehen muss. Es wird natürlich sprachlich Dokumentiert. Einzelne Projektleiter können auch UML, MindMap usw. verwenden, es gibt aber keine Vorschrift

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang B.4 Verifikation und Validierung

1. Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden?
   - Automotive
     - *Gamma:* Abhängig von den Kosten: Unter 20.000 Euro pro Einzelmaßnahme ist das operative Team autark (interne Priorisierung und Entscheidung), darüber liegt die Verantwortlichkeit bei einem Entscheiderkreis. Entwicklerteam entscheidet über Umsetzbarkeit (ob es umsetzbar ist
     - *Delta:* Höherer IT-Manager entscheidet bei BMW über Umsetzungen in den Anforderungskategorien 3&4, da so viele unnötige Anforderungen so gefiltert werden (durch den Manager selbst, durch das Hemmnis des langen Entscheidungswegs sowie durch die Notwendigkeit eines Business Cases). Bei 1&2 entscheidet SAP
     - *Epsilon:* Solange die Finanzierung stimmt, ist das Sache der Fachabteilungen
     - *Iota:* Der Kunde
     - *Theta:* Die Baureihe. D.h. der Endkunde. Der Endkunde schaut sich die Anforderungen aber nicht im Detail an. Aber den großen Rahmen gibt er vor
   - Software
     - *Alpha:* Projektspezifisch, gemeinsam demokratisch in der Gruppe anhand Prioritäten
     - *Beta:* Letztendlich immer der Kunde. Wenn der Kunde mehr Anforderungen haben will, als zum Festpreis möglich, dann muss er dafür auch bezahlen. Wenn eine Anforderung technisch nicht möglich ist bzw. nur mit besserer Hardware möglich ist, so muss der Kunde auch entscheiden, wie viel er zu zahlen bereit ist.
     - *Eta:* Aufgrund der Vorpriorisierung der Anforderungssteller entscheidet folgende Runde: Releaseowner (hat den Überblick über alle Anforderungen), Team-Leiter (Budget), Kundenverantwortlicher, Lieferant (was ist Realisierbar)
     - *Kappa:* Der Auftraggeber soll das entscheiden. Die Umsetzung der Anforderungen ist mit dem Risikomanagement eng verbunden. D. h. die Zusammenhänge zwischen den zu realisierenden Anforderungen und dem System sollen genau überprüft werden
     - *Ny:* Der Kunde
     - *Xi:* Der Kunde trifft die endgültige Entscheidung, der Projektleiter berät und ist in den Entscheidungsprozess involviert
     - *Omikron:* Anforderungen müssen auf Geschäfts- und/oder Projektziele abgestimmt sein. Sollte eine Anforderung diese Kriterium nicht erfüllen, so wird priorisiert und damit selektiert. Der Kunde entscheidet über die Umsetzung der Anforderungen. Omikron hat nur beratende Funktion
     - *Pi:* Der Kunde kann sich überlegen, welche Anforderungen für ihn besonders wichtig sind. Das teilt er dann mit und genau diese Anforderungen sind dann umzusetzen. Das wird im Rahmen von Workshops gemacht
   - Medizintechnik
     - *Rho:* Das Produktmanagement trifft diese Entscheidungen
     - *My:* Nach der Ist-Soll-Analyse wird ein Konzept erstellt, das der Kunde abnehmen muss
     - *Lambda:* Upstream-Marketing
     - *Zeta:* Der Projektleiter entscheidet dies
2. Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?
   - Automotive
     - *Gamma:* Leistungsscheine sind Grundlage für Überprüfung. Interne Entwicklung überprüft, ob die mit externen Dienstleistern abgeschlossenen Werkverträge

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

eingehalten wurden; Live-Vortests mit den Kunden; TestDirector Tool; eventuell (bei Bedarf) noch Testfallgenerierung durch Testcenter
  - *Delta:* Generell liegt die Validierung in der Verantwortlichkeit der Fachbereiche. Leider werden einige Eigenentwicklungen nicht genutzt (Veralterung von Anforderungen, lange Prozessdauer), genaue Zahlen liegen jedoch BMW-intern nicht vor.
  - *Epsilon:* Idealtypischer weise werden Validierungskriterien mit den Anforderungen von der Fachabteilung spezifiziert, in der Praxis jedoch selten
  - *Iota:* Es werden Testfälle erstellt. Diese werden getestet. Der Kunde führt eine Abnahme durch und überprüft, ob das Produkt seinen Erwartungen entspricht. Oft delegiert der Kunde diese Abnahme aber wieder an andere. Oft führen auch wie System- und Lasttests durch. Der Kunde spielt einige wichtige Szenarien durch, und dann ist die Abnahme fertig. Bei sehr großen Projekten gibt es eine eigene Abteilung, die Abnahmetests durchführt. Dort sitzt dann die Testabteilung mit den Kunden zusammen und die testen gemeinsam
  - *Theta:* Wenn die Teile vom Lieferanten kommen, gibt es einen Komponententest. Dann gibt es einen Gesamtfahrzeugtest. Die Testfälle kommen aber nicht alle aus dem Lastenheft. Teilweise sind die Anforderungen auch gleichzeitig die Testfälle
- Software
  - *Alpha:* Während der Abnahme, formal
  - *Beta:* Erst internes Testen. Anhängig von Vertrag. Die Endabnahme(Fachliche Abnahme) macht der Kunde. Der Kunde testet und unterschreibt dann, dass wir das Geld bekommen. Die Abnahme Testfälle kommen von Kunden.
  - *Eta:* Testfälle werden in den jeweiligen Märkten erstellt. Der Releaseowner ist für das Schreiben der Testfälle verantwortlich. Er muss sich darum kümmern, dass die Testfälle geschrieben werden. Marktverantwortlicher ist eher für Test der technischen Anforderungen verantwortlich. Die Überprüfung wird in einem Gateway-Meeting durchgeführt. Das eigentliche Testen des Systems wird von den einzelnen Märkten durchgeführt
  - *Kappa:* Das ist die Aufgabe der Auftraggeber. Dazu werden vom Unternehmen und den Auftraggebern Tests entwickelt. Die Abnahme des Systems kann erst dann erfolgen, wenn das System vom Auftraggeber ausführlich getestet wurde
  - *Ny:* Es wurden am Anfang des Projektes mit dem Kunden die Abnahmerahmen vereinbart. Der Kunde soll im Rahmen der Abnahme entsprechende Tests durchführen. Die Abnahme soll der Kunde machen. Die Erstellung der Testfälle und der Qualitätsüberprüfung kann auch vom Unternehmen zusammen mit dem Kunden durchgeführt werden. Die Teilabnahmen finden auch im Projektverlauf statt
  - *Xi:* Jambit ist verpflichtet, ein Testprotokoll zu führen. Der Kunde testet gegen dieses Protokoll und gegen die Anforderungen. In der abschließenden achtwöchigen Testphase ist ein Jambit-Entwickler beteiligt, der dem Kunden vor Ort zur Verfügung steht. Bei einem Aufwand von sechs Personenjahren für das Gesamtprojekt sind diese acht Personenwochen ein verhältnismäßig sehr geringer Aufwand.
  - *Omikron:* Größere Kunden besitzen meist ein eigenes Quality Management (Testcenter), das die Prüfung durchführt. Allgemein werden anhand von Prüffällen und –varianten Negativtests, Modultests, Integrationstests sowie Last- und Performancetests durchgeführt
  - *Pi:* Das erfolgt im Rahmen des Entwicklungsprozesses. Das kann im Rahmen der Überprüfung der Anforderungsdokumentation stattfinden. Eine andere Möglichkeit bieten Prototypen. In der Testspezifikation wird angegeben, welche Tests durchgeführt werden sollen, damit die Anforderungen als erfüllt gelten. Die

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anforderungsanalyse ist das erste Element der Fachspezifikation. Die technischen Aspekte sind nicht in der fachlichen Konzeption enthalten. Es gibt auch eine Vorlage für das Fachkonzept. Die Prozesse werden mit EPK modelliert. Da es für den Kunden problematisch sein kann, die Prozesse zu verstehen, wird es versucht, keine Tools (zum Beispiel, ARIS) im Fachkonzept einzusetzen

- Medizintechnik
  - *Rho:* Durch das Tracing ist klar, welche Anforderungen umgesetzt worden sind. Das entscheidende ist, dass alle Traces umgesetzt worden sind. Letztendlich gibt es einen Integrations- und Abnahmetest. Dort wird das überprüft
  - *My:* Es werden technische Dokumentationen erstellt, damit der Kunde genau nachvollziehen kann, wie das installierte System aussieht. So kann der Kunde auch genau prüfen, ob das System sein Pflichtenheft realisiert
  - *Lambda:* Tests durch Upstream-Marketing (vor allem auch genaue Überprüfung der gesetzlich vorgegebenen Anforderungen) und durch Kundenabnahme (Endabnahme oder Nutzung des Release Candidate bzw. Prototyp) Testbedingungen bzw. gegen zu testende Anforderungen werden früh in der Entwicklung definiert
  - *Zeta:* Das sind sehr wichtige Aspekte. Es geht um medizinische Geräte, deswegen ist es besonders wichtig, zu überprüfen, ob die entwickelten Konzepte mit den Anforderungen übereinstimmen. Das erfolgt in einer Testphase. Im R-Tool wird zu jeder Anforderung ein Testfall spezifiziert. Dieser muss dann überprüft werden. Der Testfall wird abhängig von der Anforderung auch natürlich sprachlich definiert

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang B.5 Prototyping

1. Welche Zweck oder welche Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung,...)
   - Automotive
     - *Gamma:* Keine Nutzung von Prototypen für Weiterentwicklungen
     - *Delta:* Prototypen in Kommunikationsbereich sind super wichtig.
     - *Epsilon:* Für die Kundenkommunikation oder Testen von Systemen, allgemein kaum Prototyping.
     - *Iota:* Vor allem GUI-Prototypen.
     - *Theta:* Nein, bei der Ermittlung werden keine Prototypen gemacht
   - Software
     - *Alpha:* Anforderungsüberprüfung oder Prüfung von neuen Funktionalitäten
     - *Beta:* schwierig zu erklärende Sachen visualisieren; Sachen mit großen Aufwand zu besprechen
     - *Eta:* In der Weiterentwicklung werden keine Prototypen eingesetzt. Am Anfang der Entwicklung wurde ein Prototyp gemacht. Der Prototyp war nur dazu da um die UI zu gestalten.
     - *Kappa:* Prototypen dienen der Kommunikation zwischen dem Auftraggeber und dem Unternehmen. Sie werden für die Überprüfung der Umsetzbarkeit der Anforderungen und der Erwartungen der Auftraggeber eingesetzt.
     - *Ny:* Der Zweck des Prototypen ist die Demonstration und die Pilotierung.
     - *Xi:* Prototypen werden in zwei Bereichen eingesetzt: für Visuelles (Mockups) und für Technisches, also horizontal und vertikal. Ersteres soll dem Kunden frühestmöglich einen Eindruck vermitteln, wie die Software am Ende aussieht. Sollte dies nicht den Wünschen des Kunden entsprechen, kann dieser früh in den Entwicklungsprozess eingreifen. Letzteres hingegen dient der Verifikation des Einsatzes von Techniken.
     - *Omikron:* Kunde sich ein Bild vom späteren Endsystem machen kann. So werden Anforderungen besser verstanden und können somit besser beschrieben werden. Sonderwünsche des Kunden werden früh erkannt
     - *Pi:* Prototypen werden vorwiegend für die Kommunikation mit dem Kunden eingesetzt. Am Anfang werden Prototypen in Gesprächen mit Kunden eingesetzt, um das Projekt zu gewinnen.
   - Medizintechnik
     - *Rho:* Prototypen im Rahmen eines Workshops: In einem Workshop wird ein Prototyp eingesetzt um besseres Feedback zu bekommen, Prototypen um die Machbarkeit zu demonstrieren, um den Stand des Projektes zu demonstrieren
     - *My:* den Kunden das System zu präsentieren
     - *Lambda:* Kaum Prototyping im Requirements Engineering; Wenn dann zum Aufzeigen bestimmter Funktionalitäten.
     - *Zeta:* Prototypen werden eingesetzt, um Konflikte zu lösen. Sie dienen zur Kommunikation und um den Leuten was Konkretes zu zeigen.
2. Für welche Zielgruppen (Stakeholder) werden Prototypen entwickelt?
   - Automotive
     - *Gamma:* Fachabteilung(Kunden)
     - *Delta:* Anforderungssteller (Fachabteilungen)
     - *Epsilon:* Kunden
     - *Iota:* Diskussionen mit dem Kunden.
     - *Theta:* -
   - Software

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *Alpha:* Endanwender, Management
- *Beta:* Kunden
- *Eta:* -
- *Kappa:* Auftraggeber (Kunde)
- *Ny:* Stakeholder, der Kunde
- *Xi:* Für den Kunden
- *Omikron:* Projektmitarbeiter aus Fachbereich oder Mitarbeiter aus späteren Nutzteams.
- *Pi:* Für den Fachbereich, End-User und weitere Kunden

- Medizintechnik
  - *Rho:* Zum Teil der Endanwender. Zum Teil wird der Prototyp für die Integration erstellt.
  - *My:* Kunden
  - *Lambda:* wichtige oder potenzielle Kunden
  - *Zeta:* -

3. Wie werden diese Prototypen entwickelt?(Werkzeuge, Sprachen, Frameworks, ...)
   - Automotive
     - *Gamma:* Die üblichen Tools für Java-Entwicklung, PowerPoint. Weiterentwicklung des Prototypen bis fertigen Produkt.
     - *Delta:* Bei echte Prototypen meist vertikale Implementierung, Prototyp als funktionaler Teil der Lösung
     - *Epsilon:* -
     - *Iota:* HTML, Powerpoint
     - *Theta:* -
   - Software
     - *Alpha:* Keine bestimmte Sprache für Prototypenentwicklung
     - *Beta:* Früher mit Basic, heute eher mit HTML
     - *Eta:* Es wurde dieselbe Technologie wie im Produktiv-System eingesetzt
     - *Kappa:* Projektspezifisch (UML, Papierprototype usw.)
     - *Ny:* Projektspezifisch.
     - *Xi:* Powerpoint, Photoshop, GUI-Tools
     - *Omikron:* Modifikation von Standard SAP Demosystem
     - *Pi:* Projektspezifisch. Papierform, Office-Tools, leichtgewichtige IT-Werkzeuge (z. B. Visual Basic) oder die schwergewichtige Technik, die auch beim System später eingesetzt werden.
   - Medizintechnik
     - *Rho:* Es wird die Technologie eingesetzt, die auch im Endprodukt eingesetzt wird
     - *My:* -
     - *Lambda:* Produkt in unfertigem Zustand
     - *Zeta:* Es wird ein Softwareprototyp erstellt. Der wird in der Sprache programmiert, in der auch normal Programmiert wird. Es sind teilweise inkrementelle Prototypen, teilweise nicht

4. Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein?
   - Automotive
     - *Gamma:* Prototypen nur bei nicht-komplexen Gebieten sinnvoll, da sonst kein Mehrwert gegenüber der normalen Entwicklung (zu aufwendig)

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

- *Delta:* Wenn Prototyping für Anforderungen der Priorität 1, dann sind Kosten kein Entscheidungsfaktor. Prototypen sind für Prio1 Anforderungen meist elementar. Um Kosten zu sparen arbeitet man inkrementell die Prototypen zu Lösungen aus.
- *Epsilon:* Leider meist zu hoher Aufwand. Oft nur bei Wartungsprojekten möglich, da hier bereits eine Basis gegeben ist, auf der man leicht aufsetzen kann.
- *Iota:* Man setzt Prototypen nur für die zentralen Dialoge ein, weil der Aufwand groß ist
- *Theta:* -

- Software
  - *Alpha:* -
  - *Beta:* grob geschätzt: 20-25% vom Analyseaufwand. Dieser selbst ist so 15% bis 20%. Insgesamt sind's dann so 5%. Prototyping ist in den letzten 10 Jahren immer schneller geworden. Der Aufwand einen Prototyp zu erstellen, ist fast kein Aufwand mehr.
  - *Eta:* Kostenteil ungefähr 20% der Gesamtentwicklung.
  - *Kappa:* Im Durchschnitt sind das ca. 10% vom gesamten Aufwand für die Entwicklung
  - *Ny:* Fast kein Aufwand, denn es wird dadurch das Risiko für die Entwicklung reduziert, es wird aus den Fehlern gelernt.
  - *Xi:* Sehr geringer Aufwand bei hoher Sicherheit.
  - *Omikron:* Geringer Aufwand
  - *Pi:* Screen Mock-Up sind einfach zu erstellen, die Prototypen in schwergewichtiger Technik sollen entwickelt werden, wenn die Technik noch nie ausprobiert wurde
- Medizintechnik
  - *Rho:* -
  - *My:* -
  - *Lambda:* zu hoher Aufwand für sinnvollen Einsatz
  - *Zeta:* Software: 30% vom Projektaufwand

5. Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?
   - Automotive
     - *Gamma:* Zu viel Aufwand für komplexe Gebieten.
     - *Delta:* Arbeit am System" wichtig für Fachabteilungen, zur besseren Einschätzung der genauen Anforderungen als Entscheidungsgrundlage
     - *Epsilon:* Negativ ist die Gefahr der funktionalen Weiterentwicklung eines (nicht perfekten) Prototyps als Ausgangsbasis für das Endprodukt
     - *Iota:* Man muss deutlichen sagen, wozu der Prototyp dient. Ansonsten stehen nicht sinnvolle Diskussionen.
     - *Theta:* -
   - Software
     - *Alpha:* Dagegen: nur konzeptionell.
     - *Beta:* Mitte der 1990er haben die Kunden oft nicht verstanden, dass der Prototyp nur ein Prototyp ist. Die Kunden dachten dann, dass das Produkt schon fertig ist und sie dachten in 3 Tagen wäre das endgültige System fertig. Wenn man mit jemandem Arbeitet, der das noch nie gemacht hat, dann gibt es das Problem auch noch. Aber heutzutage haben schon fast alle Leute mal mit einem Prototypen gearbeitet und wissen wie damit umzugehen ist. Gefahr zu viele Iterationen zu machen.
     - *Eta:* Teilweise wäre es effizienter inkrementell vorzugehen. Aber das hängt von der Kritikalität ab. Und das hängt davon ab, was man mit den Prototypen erreichen will. Um die Machbarkeit zu überprüfen ist ein Prototyp sicher sinnvoll. Bei jeder

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Entscheidung für oder gegen Prototypen muss der Kosten-Nutzen abgeschätzt werden.
  - *Kappa:* Der Einsatz von Prototypen ist sehr komplex.
  - *Ny:* Wenn aber das Projekt sehr kurz ist oder ein Bestandsprojekt ist, dann macht Prototyping keinen Sinn.
  - *Xi:* Letztgenannte Tatsache macht Prototypen unverzichtbar.
  - *Omikron:* Erwartungsmanagement, Kunde äußert ohne Prototyping meist sehr viele Wünsche, die nicht umgesetzt werden können, was bei ihm Enttäuschung hervorruft
  - *Pi:* Der Einsatz von Prototypen bringt viele Vorteile. Wenn die Technik nicht bekannt ist, kann man dadurch die Technik ausprobieren. Falls die Technik bekannt ist, kann der Prototyp dazu genutzt werden, um dem Kunden das System zu präsentieren.
- Medizintechnik
  - *Rho:* -
  - *My:* -
  - *Lambda:* zu hoher Aufwand für sinnvollen Einsatz
  - *Zeta:* Dagegen: Es verbraucht Ressourcen. Dafür: Man plastisch darstellen, was man macht. Dadurch ist Diskussion besser möglich. Verifikation.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang C Unterlagen Innovationsworkshop

Technische Universität München

Projekt Kick-Off

22. Januar 2010

Maximilian Pühler

Chair for Information Systems

Technische Universität München

Agenda

- Motivation
- Herausforderungen
- Vorgehen – *Der Weg zum neuen Service*
- Modellierung – *Modellieren statt Implementieren*
- Architektur
- Meilensteine 2010
- Workshop
- Ausblick: Ideenportal

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Technische Universität München

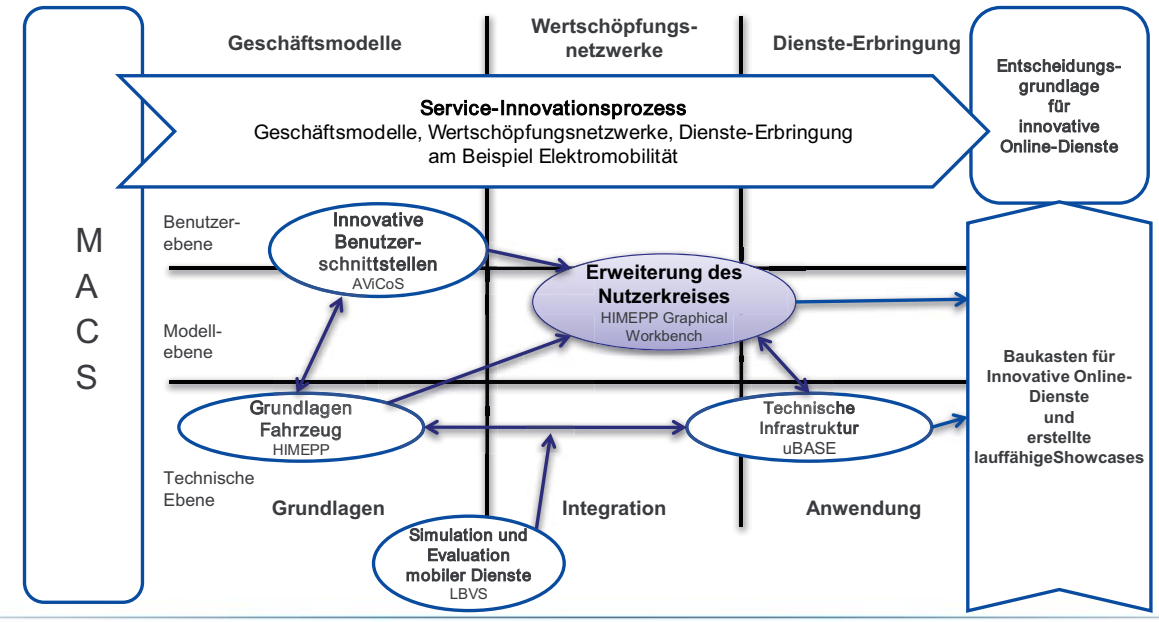
Agenda

- Motivation
- Herausforderungen
- Vorgehen – *Der Weg zum neuen Service*
- Modellierung – *Modellieren statt Implementieren*
- Architektur
- Meilensteine 2010
- Workshop
- Ausblick: Ideenportal

Technische Universität München

Motivation

Roadmap mobiler Dienste im Fahrzeug



Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Motivation

Grafisches Prototyping vernetzter Dienste im Fahrzeug

Problemstellung

- **Prototypische Darstellung** von neuartigen Online-Diensten im Fahrzeug erfordern **hohen Realisierungsaufwand**
- Für **Experten** macht **HIMEPP** (INI.TUM Projekt 2006-2009) Schnittstellen zu Fahrzeug und Fahrer zugänglich

Zielsetzung

- Tool für **visuelle prototypische Umsetzung** von Online-Diensten zur Beurteilung der Kundenrelevanz im frühen Ideenstadium
- **HIMEPP als Bestandteil** einer durchgängigen Architektur von Fahrzeug und IT-Backend
- Erweiterung der HIMEPP Prototypenplattform als Tool zur **günstigen und schnellen Realisierung** neuer Dienste vor Kunde zu Zwecken der **Ideengenerierung und -tests**

Durchführung

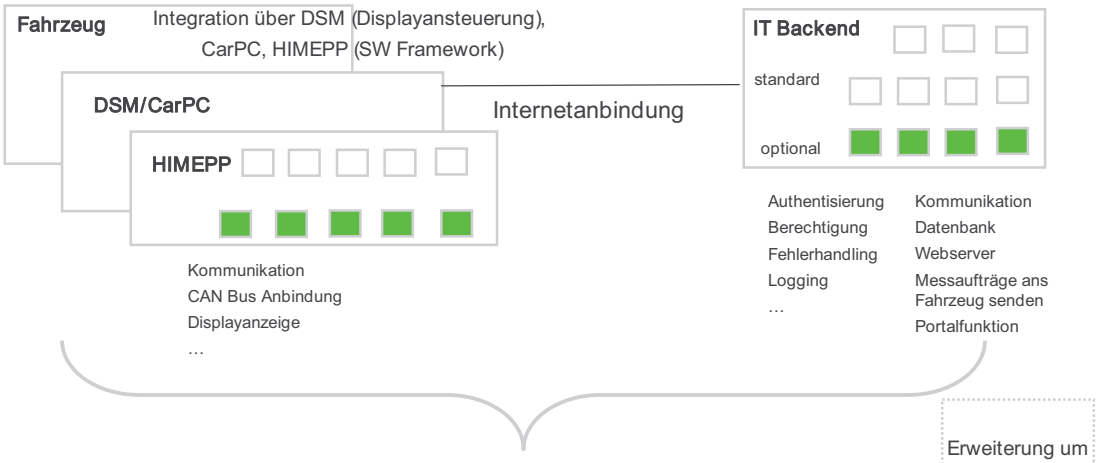
- **Erweiterung** der im **HIMEPP** Projekt realisierten Hardware und Software Plattform (CarPC) um **Modell- und Interaktionsebene**
- Verfahren zur programmgestützten **Überführung von Anwendungsfällen** auf grafische Oberflächen
- Abstimmung der Arbeitspakete und Schnittstellen mit Fachbereichen

Ausblick

- **Visuelles Konzept** ermöglicht in Zukunft **allen Anspruchsgruppen** das Prototyping von neuartigen Online-Diensten
- **Anbindung an bestehende Simulationstools** fürs Fahrzeug und Serienendgeräte (MMI3G)
- Modellebene ermöglicht später die einfache Überführung ausgewählter Prototypen in den (modellgetriebenen) Produktprozess (z.B. AutoSAR)

Motivation

Schnell neue vernetzte Funktionsprototypen erlebbar machen



Erweiterung um **Mobile Device** vorgesehen

- ▶ Beim Erstellen eines neuen Prototypen wird dem Anwender durch graphische Auswahl von optionalen Software Modulen ein Projekt generiert.
- ▶ Dafür stehen logische Komponenten für das Fahrzeug (HIMEPP) und IT Backend (uBase, in Beantragung) zur Verfügung.
- ▶ Anwender ist schnell arbeitsfähig und konzentriert sich auf seine Funktionsentwicklung.

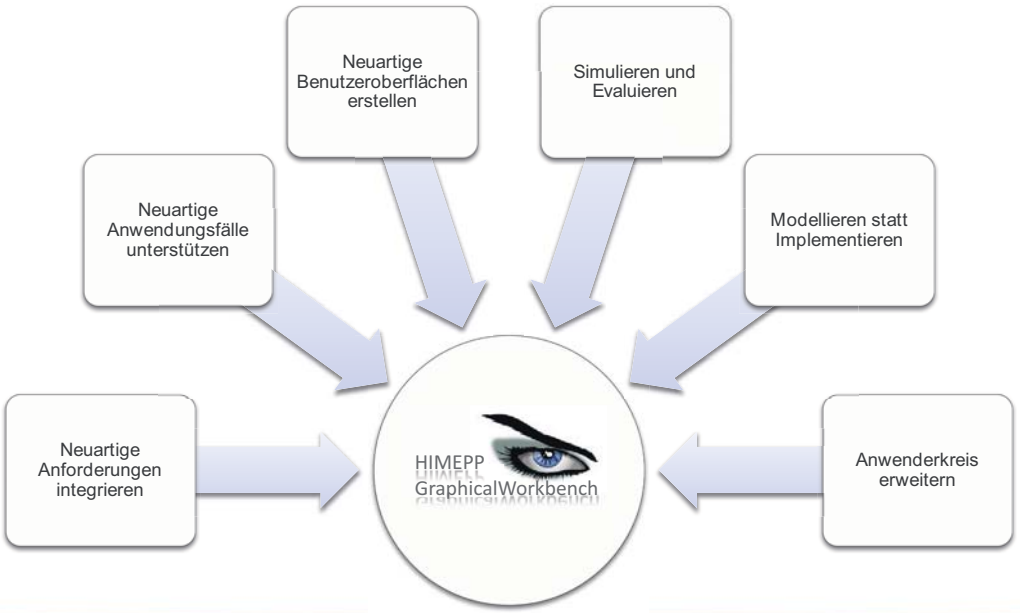
Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden. Es gilt nur für den persönlichen Gebrauch.

Agenda

- Motivation
- Herausforderungen
- Vorgehen – *Der Weg zum neuen Service*
- Modellierung – *Modellieren statt Implementieren*
- Architektur
- Meilensteine 2010
- Workshop
- Ausblick: Ideenportal

Herausforderungen

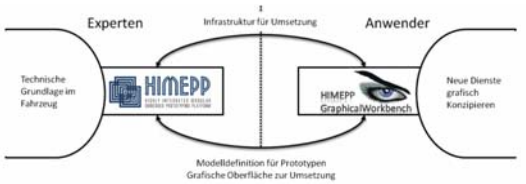
Ziele

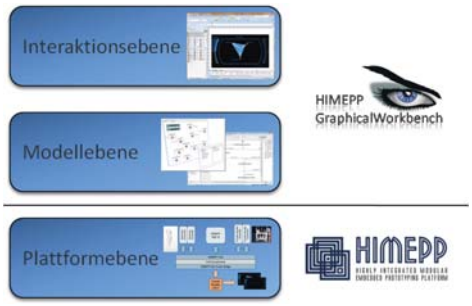


Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Herausforderungen

Vision





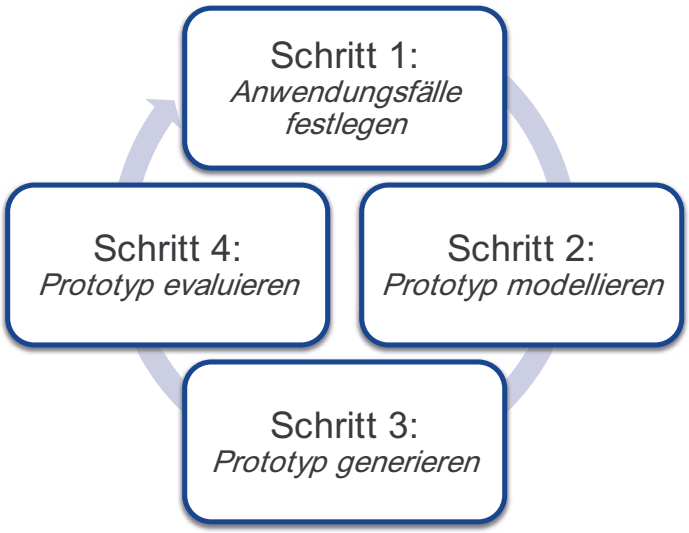
- Herausforderungen:
  - Modellieren statt Implementieren
  - Benutzeroberfläche automatisiert erstellen
  - Erweiterung des Anwenderkreises
  - Vernetzung der Prototypen mit Mobilgeräten
  - Simulieren und Evaluieren innovativer Prototypen
- Ergebnis:
  - Grafische Modellierungsumgebung für Automotive Services
  - Demonstrator „Outlook im Fahrzeug“

Agenda

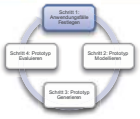
- Motivation
- Herausforderungen
- Vorgehen – *Der Weg zum neuen Service*
- Modellierung – *Modellieren statt Implementieren*
- Architektur
- Meilensteine 2010
- Workshop
- Ausblick: Ideenportal

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Vorgehen
Vorgehensmodell



Vorgehen
Schritt 1: Anwendungsfälle festlegen

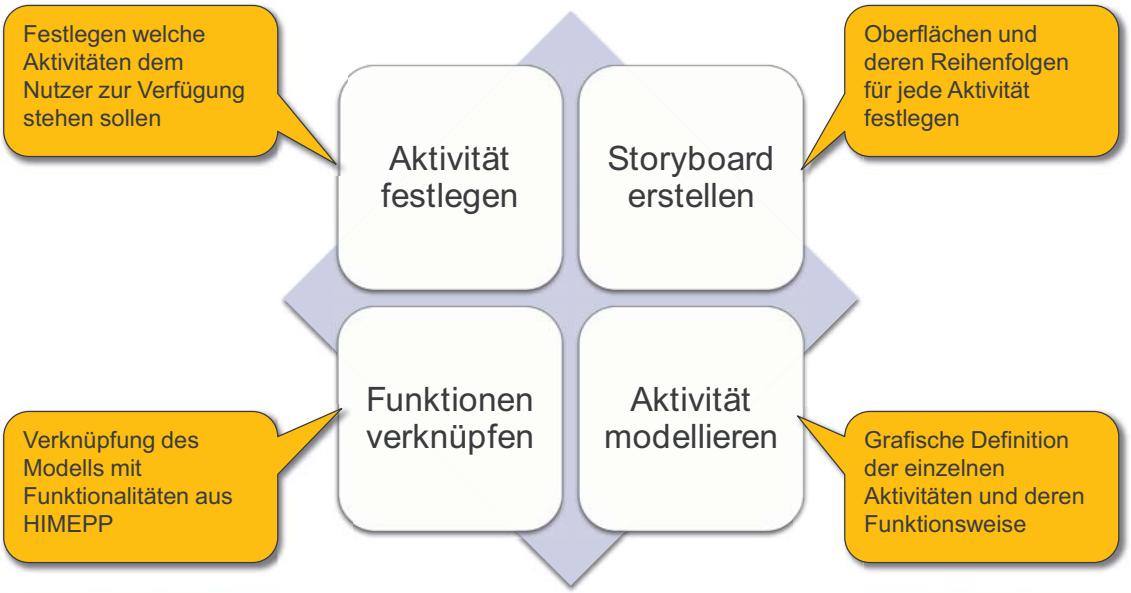


- „Klassisches“ Requirements Engineering anwendbar
  - Workshops
  - Brainstorming
  - Kundenkliniken
- Innovationsmanagement
  - Open Innovation
  - Befragungen
  - Beobachtungen
- Methoden der Markforschung
  - Markt
  - Konkurrenz
  - …

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

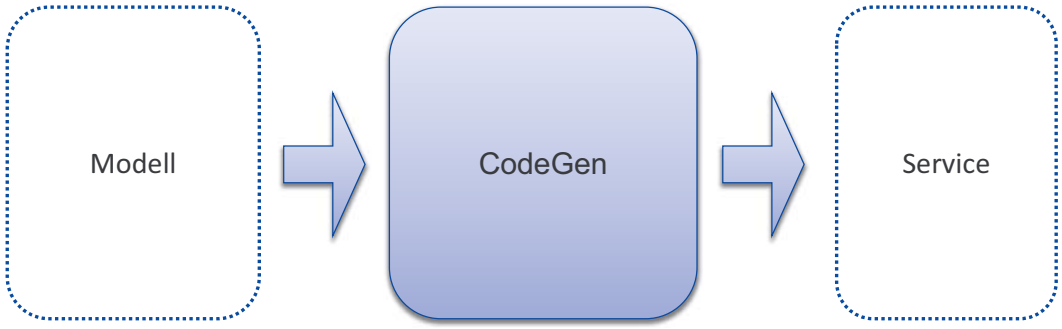
Vorgehen

Schritt 2: Prototyp Modellieren



Vorgehen

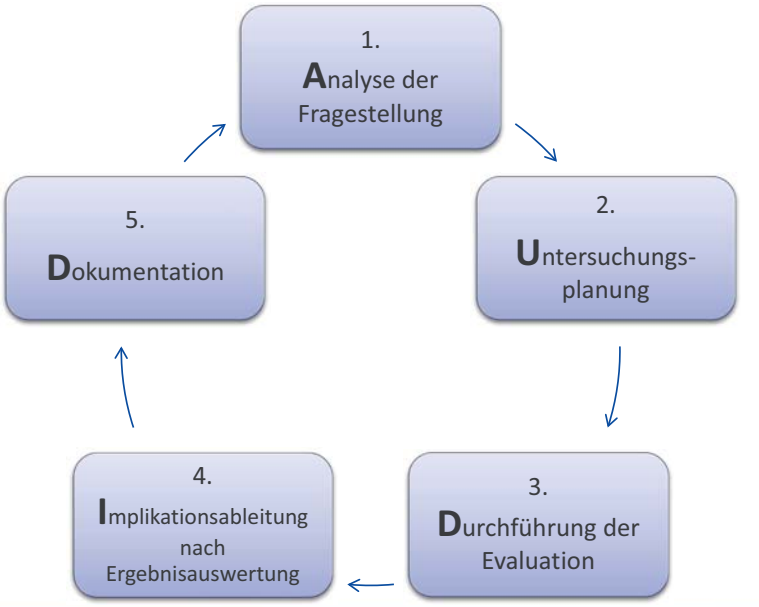
Schritt 3: Prototyp Generieren



Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Vorgehen

Schritt 4: Prototyp Evaluieren



Systematische Evaluation neuer Automotive Services in Anlehnung an das AUDID Evaluations-model

Quelle: Sharafi 2009

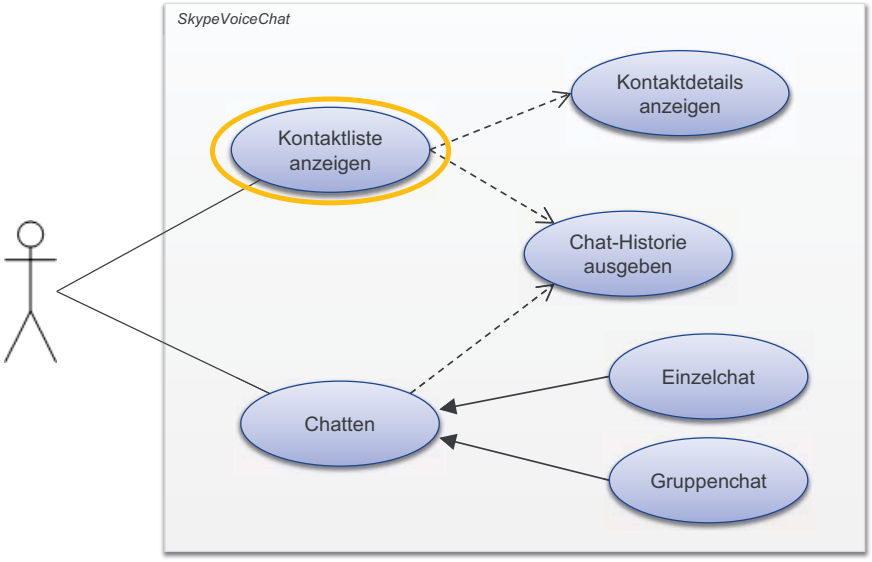
Agenda

- Motivation
- Herausforderungen
- Vorgehen – *Der Weg zum neuen Service*
- Modellierung – *Modellieren statt Implementieren*
- Architektur
- Meilensteine 2010
- Workshop
- Ausblick: Ideenportal

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

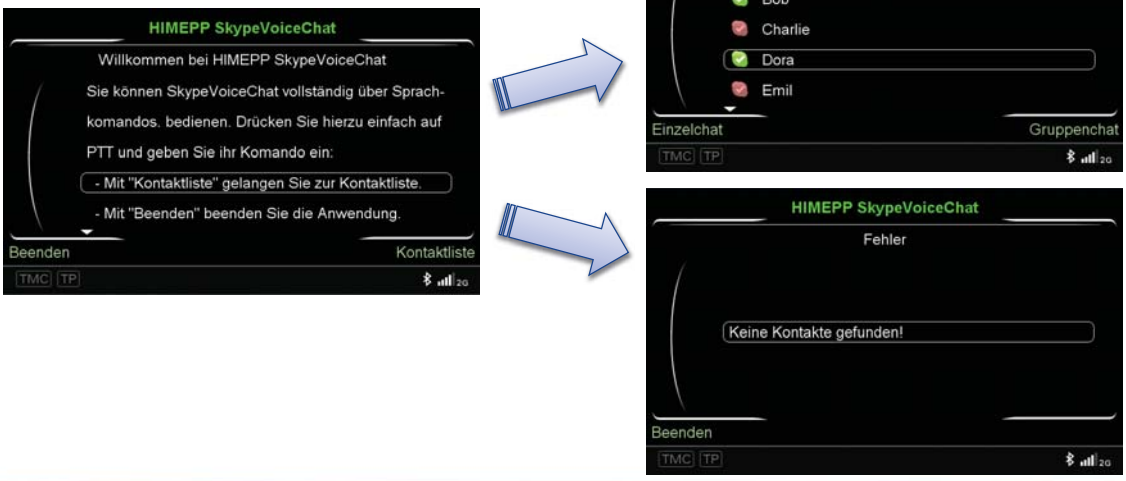
Technische Universität München

Modellierung
Aktivität festlegen



Technische Universität München

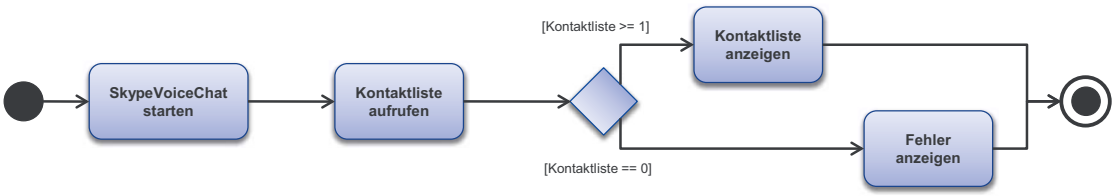
Modellierung
Storyboard erstellen



Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

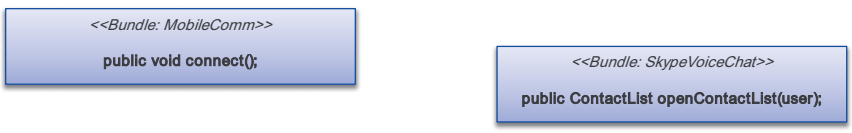
Technische Universität München

Modellierung
Aktivität modellieren



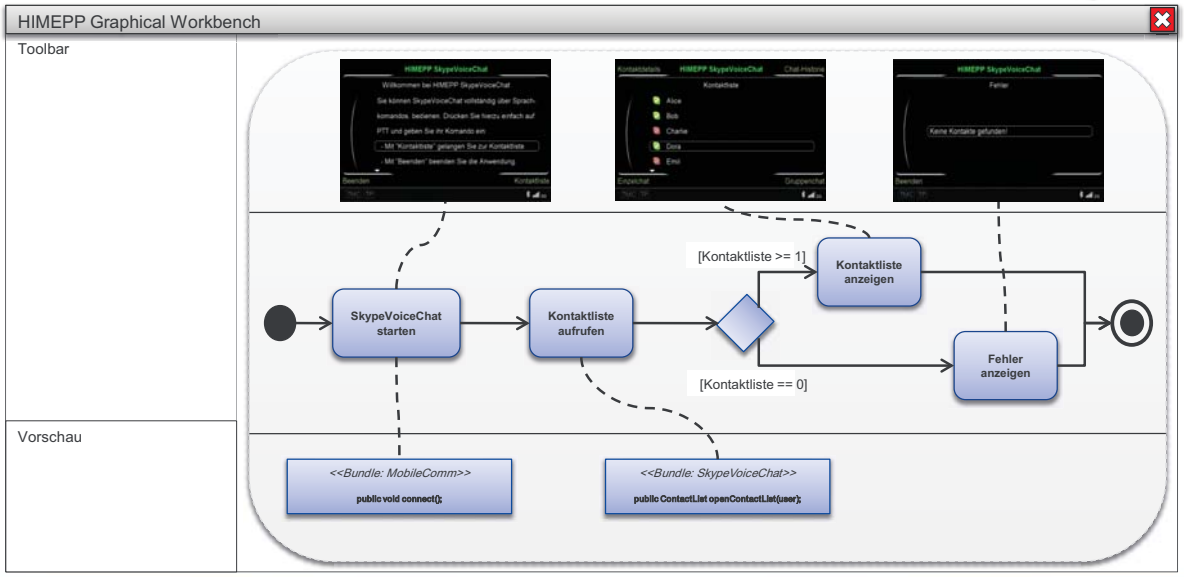
Technische Universität München

Modellierung
Funktionalität verknüpfen



Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Modellierung
Automotive Service Model

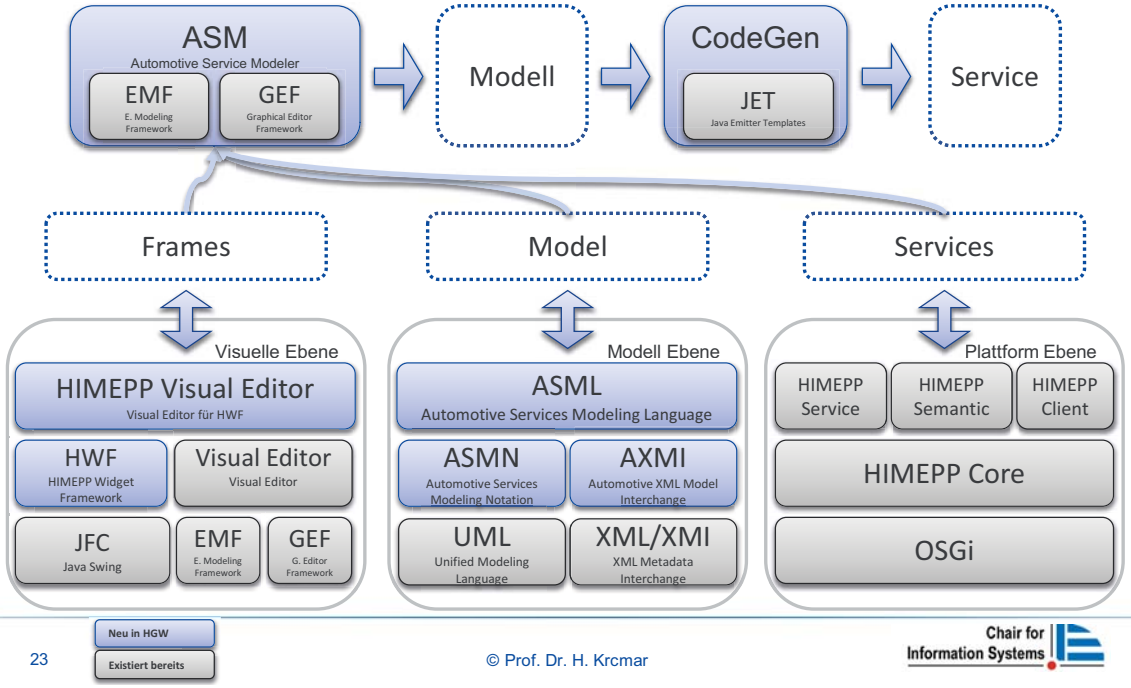


Agenda

- Motivation
- Herausforderungen
- Vorgehen – *Der Weg zum neuen Service*
- Modellierung – *Modellieren statt Implementieren*
- Architektur
- Meilensteine 2010
- Workshop
- Ausblick: Ideenportal

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

HGW Architekturübersicht



© Prof. Dr. H. Krcmar

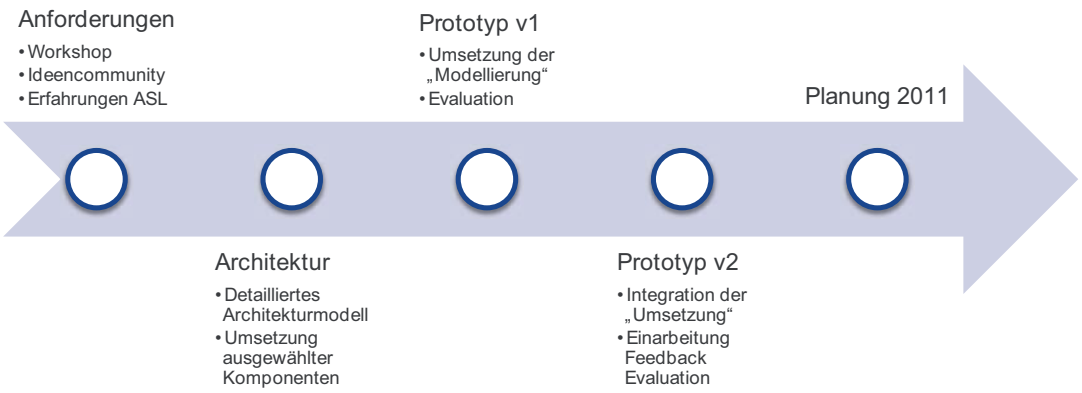
Agenda

- Motivation
- Herausforderungen
- Vorgehen – *Der Weg zum neuen Service*
- Modellierung – *Modellieren statt Implementieren*
- Architektur
- Meilensteine 2010
- Workshop
- Ausblick: Ideenportal

© Prof. Dr. H. Krcmar

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Meilensteine 2010



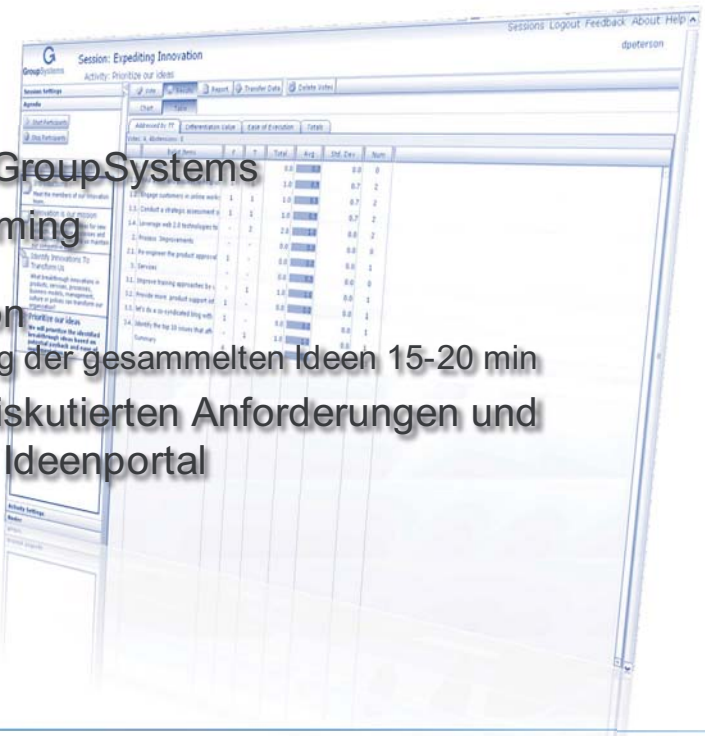
Agenda

- Motivation
- Herausforderungen
- Vorgehen – *Der Weg zum neuen Service*
- Modellierung – *Modellieren statt Implementieren*
- Architektur
- Meilensteine 2010
- Workshop
- Ausblick: Ideenportal

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Workshop

Vorgehen



- Brainstorming mit GroupSystems
  - Teil 1: Brainstorming
    15-20 min
  - Teil 2: Diskussion
    und Kategorisierung der gesammelten Ideen 15-20 min
- Aufbereitung der diskutierten Anforderungen und Überführung in ein Ideenportal

Agenda

- Motivation
- Herausforderungen
- Vorgehen – *Der Weg zum neuen Service*
- Modellierung – *Modellieren statt Implementieren*
- Architektur
- Meilensteine 2010
- Workshop
- Ausblick: Ideenportal

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Technische Universität München

Ausblick

Ideenportal

1. Workshopergebnisse werden als Anforderungen auf der Ideenplattform zusammengefasst
2. Ein Link auf die Plattform wird an alle "Interessierten" verteilen
3. Auf der Plattform können Anforderungen
   1. kommentiert,
   2. priorisiert und
   3. ergänzt

   werden.
4. Ergebnis: Abgestimmte Anforderungen an HGW

→ Link: http://audi.ideaontology.org

Technische Universität München

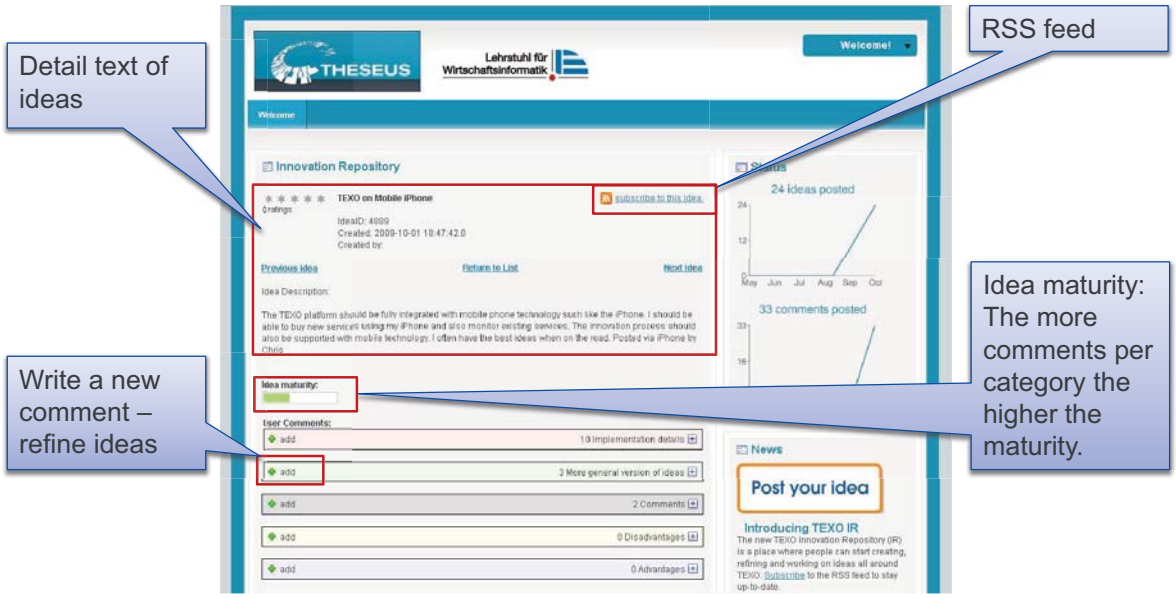
Ausblick

Ideenportal – TEXO Innovation Repository

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Ausblick

Ideenportal – TEXO Idea Details



HIMEPP GraphicalWorkbench

Vielen Dank!

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Anhang D Protokoll Innovationsworkshop

GroupSystems
info@groupsystems.com

HGW Kick-Off

Session Details:

Start:	22.01.2010
End:	22.01.2010
Location:	01.13.010

Session Leader:

Login	Name	Email
labmaster	l labmaster	n@n.com

Session Participants:

Login	Name	Email
mobile6		
user04		
user05		
user06		
user07		
user08		
user10		
user11		
user12		
user13		
user14		
user15		
user4		
user5		
user7		
user8		

Session Documents:

Name	Description	File Name/URL
ThinkTank Generated Report	This report was generated by ThinkTank and is a report on this session.	SessionReport.doc

Agenda:

Brainstorming

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

1. Brainstorming

1. **Brainstorming**

2. **Wiederverwendung**
   - 2.1. methoden des Freischaltens berücksichtigen
   - 2.2. HGW sollte klare Konventionen fuer ein methodisches Modellieren durch die Anwender und die Wiederverwendbarkeit sicherstellen
   - 2.3. es sollte die möglichkeit geben, bewährte dienste als neue modellkomponenten abzulegen und wiederverwendbar zu machen
   - 2.4. wird es ein standardset an funktionen (login, logout, session, etc) geben, die immer d.h. ohne interaktion bei einem dienst dabei sind?
   - *2.4.1. Authentifizierung? Authorisierung*
   - 2.5. schnelle erweiterbarkeit der erstellten dienste
   - *2.5.1. Auch um noch nicht vorhandene Komponenten wie Backend etc.*
   - 2.6. eine nachträgliche erweiterbarkeit um Mobile devices/Fahrzeug sollte jederzeit möglich sein

3. **Usability**
   - 3.1. HGW sollte auch nicht überladen werden um damit sein ziel genau zu erfüllen
   - 3.2. "entwickler die folgende komponente verwendet haben, nutzten auch diese Komponente"
   - 3.3. Für bestehende Module evtl. immer ein Beispiel zeigen wie diese Komponte funktioniert und eingebunden werden kann
   - 3.4. Bei der Erstellung der Prozessmodelle den Benutzer darauf hinweisen, wenn bestimmte Komponenten so nicht zusammenpassen bzw. evtl. nicht miteinander komptibel sind.
   - 3.5. wie kann man sicherstellen, dass der nutzer aus einem generischen ziel heraus zu seinem dienst geführt wird
   - 3.6. wie könnte ein Tutorial integriert werden? Ist es nötig?
   - 3.7. es sollte eine art vorschau geben, die in echtzeit den dienst anzeigt
   - 3.8. Denkbar wäre auch, dass ein Entwickler seinen Use Case in Worten beschreibt und HGW die passenden Module vorschlägt.
   - 3.9. der Modellierer sollte möglichst keine IT Kenntnisse benötigen, sondern seine Prozessvorstellungen einfach abbilden können
   - 3.10. HGW sollte eine Funktion beinhalten zu den bestehenden Services eine Beschreibung abzugeben und Stichwörter, die die Funktionalität beschreiben. Entwickler könnte die passenden Module finden indem er paar Stichpunkte eingibt.
   - 3.11. HGW sollte ein Repository für verschiedene Funktionalitäten haben

4. **Anbindung HIMEPP**
   - 4.1. Fahrzeugbezug verstärken
   - *4.1.1. Cargate anbindung -> CAN BUS*
   - *4.1.2. im Fahrzeug verbaute, bereits vorhandene Komponen wie Kamera*
   - *4.1.3. Simulationskomponenten der Fahrzeugkomponenten ->GPS, CAN Bus Signale etc.*
   - *4.1.4. schnelle integration neuer sensoren und deren apis*

5. **GUI**
   - 5.1. Verteilte Anwendung (mehrere gleichzeitig Eingabe und Ausgabesysteme)
   - 5.2. Eingabemechanismen über GUI designbar? Frontcontroller über TOuchscreen?

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

5.3. kann man auch andere Displays ansprechen (rearseat)?
5.4. FIS auch designbar (verteilte Anwendung zwischen FIS und MMI Screen)
5.5. unterschiedliche ausgabeformate berücksichtigen, gerade wegen augmented reality
5.6. golden rules für die gestaltung der screens
5.7. gibt es vordefinierte Standardframes innerhalb derer man dann eine neuen Dienst modellieren kann (z.b. Funktionen, die zb. architektinosch durch ein Portal fest vorgegeben sind; Sicherheitsmechanismen die zwingend vorgeschrieben sind, ...).
5.8. sollte man das Look and feel/Bedienkonzept/Layout im laufenden Betrieb ändern können um Unterschiede zu demonstrieren?
5.9. sollte man beim modellieren berücksichtigen können in welchem Fahrzeugtyp/modell/Marke die Applikation laufen können soll?
5.10. Skindesigner?
5.11. inwieweit sind funktionen "skinnable"?
5.12. mehrsprachigkeit von Applikationen im Fahrzeug vorsehen

6. Verteiltes Arbeiten

6.1. es sollte eine möglichkeit geben, gemeinsam an modellen/prototypen zu arbeiten
6.2. mit dem Innovationsprozess verknüpfen und dessen Anforderungen berücksichtigen

7. Anforderungen Modellierung

7.1. Die Steuerung des Prototypen schon standardmäßig per Spracheingabe im HGW machen, dabei auch sofort bei der Modellierung den Sprachwortschatz festlegen
7.2. mehrsprachigkeit von Applikationen im Fahrzeug vorsehen
7.3. HGW sollte neben der Anwendung und Funktionen auch die Daten (IN/OUT) beschreiben
7.4. kann das mobile gerät/das fahrzeug mit dynamisch auswählbaren Backends kommunizieren?
7.5. wie kann die interaktion zwischen mobilgerät und auto modelliert werden?
7.6. was sind wichtige konfigurationsoptionen für komponenten in hgw
7.7. gibt es zu beachtende rahmenbedingungen, wenn hgw in unterschiedlichen Szenarios einsetzt werden soll (Mobil, Desktop, Auto)
7.8. Es sollte keinen service ohne eine miminalausstattung an funktionen z.B. für Sicherheit geben
7.9. Gesamtarchitektur berücksichtigen um die Schnittstellen nach außen vorzusehen
7.9.1. Backend -> Auto -> Service Provider beispielsweise
7.10. wer kann neue modellierungskomponenten einstellen?
7.11. gibt es zentrale entscheidungspunkte in solchen anwendungen (z.B. fehlerscreens)
7.12. es sollte möglich sein zu einem User Case verschiedene Alternativen abbilden zu können
7.13. mit dem Innovationsprozess verknüpfen und dessen Anforderungen berücksichtigen
7.14. kann man von vorneherein einen Dienst für verschiedene Bedienkonzepte modellieren?
7.15. gibt es eine art qualitätssicherung, d.h. überprüfung des dienstes auf fahrsicherheit, logische inkonsistenzen etc.
7.16. Grundsätze ordnungsemäßer Modellierung (GOM) einhalten
7.17. eine nachträgliche erweiterbarkeit um Mobile devices/Fahrzeug sollte jederzeit möglich sein

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

7.18. funktionen mit Fahrzuegbezug sollten leicht aus Backend konfigurierbar sein (bei Modellierung)
7.18.1. es sollte keinen unterschied machen, wo der jeweilige Dienst läuft
7.19. können die Bedienschritte/Daten die vom Dienst/Fahrzeug generiert werden dem Admin zentral bereitgestellt werden, so dass dieser die Nutzbarkeit/Anwendungsfälle/Fehler auswerten kann
7.20. FIS auch designbar (verteilte Anwendung zwischen FIS und MMI Screen)
7.21. kann man auch andere Displays ansprechen (rearseat)?
7.22. Verteilte Anwendung (mehrere gleichzeitig Eingabe und Ausgabesysteme)

8. Plattform
8.1. ist eine Touchpad/Touchscreen Bedienung auf der Roadmap?
8.2. hgw im auto bedienen?
8.2.1. über tablet-pc
8.3. Vielleicht eine LIVE-Erprobung im Auto sogar schon beim Modellieren und Zusammenbauen von einem Service (die Daten werden sofort mitm Auto synchronisiert und der Service kann sofortt getestet werden)
8.4. wird HGW OSGi 4.0 unterstützt und was passiert bei Versionsänderungen?
8.5. wird es zu HGW ein Diskussionsportal geben für inhaltliche aber auch für toolbezogene Ideen?
8.6. Gibt es Bundles um das Auto aus dem Backend zu steuern (z.B. Standheizung)?
8.7. sind in HGW immer die aktuellen Komponenten enthalten? hat es eine Anbindung an das SVN?
8.8. Organisationsübergreifende DB mit den bestehenden Services?
8.9. HGW sollte von aehnlichen Plattformen profitieren koennen (Developer Groups, Standards,...)
8.10. kann HGW als Webbased Tool funktionieren?
8.11. jede komponente soll einen abgesicherten mechanismus vorhalten dynamisch im betrieb angepasst zu werden
8.11.1. Einstellparameter einer Messung neu CAN Botschaften etc
8.12. neben dem fahrzeug, dem Mobile und dem Server auch den WebServer als Zielplattform für die Applikation berücksichtigen
8.13. analogien zu bestehenden ähnlichen tools aufzeigen (yahoo pipes, etc...)
8.14. modellierungstool sollte mehrsprachig (englisch) sein
8.15. moddelierung von HGW im WWW
8.16. Nicht nur Bundles HGW Bundles einbinden können, sondern auch z.B. Web Services aus dem Internet (z.B. Wetterdienst über Web Services). Im Allgemeinen in HGW auch andere Komponten einbinden können.
8.17. Verteilte Anwendung (mehrere gleichzeitig Eingabe und Ausgabesysteme)

9. Modellierungssprache
9.1. was ist die "zentrale" Ebene, die GUI, das Modell, oder die verfügbaren Services
9.2. wie werden neue modellierungselemente konstruiert und wie wird deren korrektheit sichergestellt
9.3. Um die Komplexität der Prozessmodelle in der Modellierungsumgebung zu vereinfachen wäre denkbar, dass man in die einzelnen Prozessschritte reinklicken kann (Rein- und Rauszoomen).
9.4. kann HGW im normalen und Experten modus genutzt werden? Beim experten ist alles flexibel, beim normalen Nutzer sind grundsätzliche Festlegungen (Bedienkonzept, Funktionslogik der Bedienung) bereits festgelegt
9.5. wird es eine art referenzmodell für einen automotive service geben?
9.6. Regeln für die formale Beschreibung der Komponenten sollten rückwärtskompatibel erweiterbar sein

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

10. Dokumentation von Services

10.1. direkte Beschreibung/Dokumentation von modellierten Diensten ermöglichen/einfordern/einforderbar machen
10.2. es sollte eine minimale Versionierungsmöglichkeit der Modell geben
10.3. ein Export des MOdells für ein Lastenheft (Nachbau in anderem System) sollte möglich sein (Dienst ist lebendes Lastenheft)
10.4. HGW soll die Dokumentation der umgesetzen Dienste erleichtern/automatisieren
10.5. HGW soll Bestandteil des Innovationsprozesses werden
10.6. Gesamtarchitektur berücksichtigen um die Schnittstellen nach außen vorzusehen
10.6.1. Backend -> Auto -> Service Provider beispielsweise
10.7. wird es zu HGW ein Diskussionsportal geben für inhaltliche aber auch für toolbezogene Ideen?

11. Funktionalität

11.1. sollte man HGW/uBase Last/Stresstesten wenn da viele Nutzer drauf zugreifen?! (E-Flotte)
11.1.1. modelle aus hgw als visualisierungsgrundlage für die stress/last-darstellung
11.2. codegenerierung auch für mobile platformen (iphone, blackberry) als vision?
11.3. automatische property generierung bzw. einstellungsscreens der anwendung
11.4. einbindung neuer apis (sensoren etc.) klickbar machen
11.5. generiert selbst einen gesamtübersichtsscreen (angelehnt an den audi wizard) der alle erstellten prototypen bzw. alle im projekt vorhandenen prototypen darstellt
11.5.1. und aktualisiert
11.6. gibt es eine simulationsumgebung im modell?
11.7. sollte man die Funktion/den Dienst auch im graphischen Modus ablaufen lassen können, um Zustände, Variableninhalte und Methoden überprüfen zu können?!
11.8. ggf. die möglichkeit verschiedene tarifmodelle für einen dienst zu hinterlegen
11.9. nachladen von Funktionen/Applikationen als Funktion vorhalten
11.10. Importfunktion mim Web (mit Prüfmechanismus, ob es eine valide Funtkion ist) für von Nutzern generierte Dienste für HGW WEttbewerbe
11.11. feedback Mechanismen für HIMEPP Dienste Wettbewerbe als Webtool nutzbar machen
11.12. Nicht nur Bundles HGW Bundles einbinden können, sondern auch z.B. Web Services aus dem Internet (z.B. Wetterdienst über Web Services). Im Allgemeinen in HGW auch andere Komponten einbinden können.
11.13. wie kann man einen Feedbackloop der Nutzer einbauen?
11.14. wie ist sichergestellt, dass änderungen an den services (neue funktionalitäten) in hgw aufgenommen werden
11.15. Der generierte Code von HGW ist platformunabhängig und kann mit den platformspezifischen Modulen (Mobile Device, Backend, Auto) gekoppelt werden. In HGW wird jedoch nicht zwischen den Platformen unterschieden.
11.16. Parserkomponente für Anbindung verschiedener Services bzw. Dienstleister
11.16.1. auch ohne Backend (parsen live im Fahrzeug)
11.17. wie ist der prozess neue funktionen/bausteine hgw hinzuzufügen
11.18. Um die Komplexität der Prozessmodelle in der Modellierungsumgebung zu vereinfachen wäre denkbar, dass man in die einzelnen Prozessschritte reinklicken kann (Rein- und Rauszoomen).
11.19. wie ist es mit der generierung von dummy-daten für neue, noch nicht implementierte Komponenten

12. Simulation

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

12.1. Simulationskomponenten der einzelnen Komponenten wie GPS (auch für Vorschau der Modellierung)
12.2. Evtl. Integration in ein Brainstorming-Tool, zu den erbrachten Schlagwörtern werden von HGW passende Module vorgeschlagen bzw. vielleicht komplette bereits bestehende Prozessteile
12.3. Qualitätsmetriken für die Usability neuer Prototypen (Klicks, Blickwechsel, Menues)
12.4. es sollten Unterstützungsmechanismen eingebaut werden, um die modellierte Benutzerführung (im gedachten Dienst) zu überprüfen (z.b. Anzahl Klicks bis zum gewünschten Ergebnis, etc)
12.5. man sollte Dummyschnittstellen für nicht verfügbare anzuschliessende IT Systeme berücksichtigen (Carport, SWAP...)
12.6. wie ist es mit der generierung von dummy-daten für neue, noch nicht implementierte Komponenten
12.7. können die Bedienschritte/Daten die vom Dienst/Fahrzeug generiert werden dem Admin zentral bereitgestellt werden, so dass dieser die Nutzbarkeit/Anwendungsfälle/Fehler auswerten kann
12.8. kann man automatisch für Dienste berücksichtigen, dass die Nutzerbedienschritte ausgewertet werden
12.9. Kann ein kompletter Simulator eingebaut werden der Seriendienste und Prototypen transparent auf anderen Laufzeitumgebungen darstellt?
12.10. sollte man die Funktion/den Dienst auch im graphischen Modus ablaufen lassen können, um Zustände, Variableninhalte und Methoden überprüfen zu können?!
12.11. gibt es eine simulationsumgebung im modell?

13.Vorgehensmodell

13.1. könnte man guidelines/Hilfen erstellen, wie iPhone Apps auf HGW portiert werden können? oder angebunden werden?
13.2. elektromobilität als szenario für die erprobung von hgw
13.3. erweiterung von HGW für unterschiedliche perspektiven denkbar: z.B. was passiert beim benutzer und was passiert bei einem anderen beteiligten (z.b. Dienstanbieter oder Intermediär)
13.4. HGW sollte den Schritt/Weg vom Protoypen zum Serieneinsatz beruecksichtigen/vorsehen
13.4.1. Dokumentation der "Modellierung" (aktivitätsdiagramme etc)
13.5. HGW soll Bestandteil des Innovationsprozesses werden
13.6. wie kann man sicherstellen, dass der nutzer aus einem generischen ziel heraus zu seinem dienst geführt wird

14.Nutzer

14.1. könnte man definiertes, mit festlegungen eingeschränktes, HGW für einen Wettbewerb rausgeben? muss man es schützen?
14.2. was ist der unterschied, wenn "endkunden" oder interne anspruchsgruppen modellieren sollen.
14.3. an wen richtet sich HGW genau? Soll der endkunde selber mobile dienste erstellen oder ein mitarbeiter der geschult werden könnte? => komplexität der bendienung

15.Sprachsteuerung

15.1. ist es notwendig zu bestimmten anwendungen die sprachsteuerung mit zu designen?

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.