1 Einleitung

In modernen Automobilen spielen mechatronische Systeme wie Bremsassistenten, elektronische Lenksysteme und geregelte Allradsysteme eine immer größere Rolle. Solche Systeme ermöglichen eine Erweiterung des bisherigen Funktionsumfangs. Während rein mechanische oder rein hydraulische Systeme kaum auf die vorliegende Fahrsituation reagieren können, ermöglichen mechatronische Systeme eine an die jeweilige Situation angepasste Funktionalität. Dies wird beispielsweise genutzt, um in einer Notbremssituation die Bremswirkung zu maximieren oder in instabilen Fahrsituationen Fahrfehler durch Lenkeingriffe auszugleichen. Zudem kann das Fahrzeugverhalten optimiert werden, indem die einzelnen mechatronischen Systeme interagieren und neuartige Funktionalitäten schaffen. Daraus folgt allerdings auch, dass die Anforderungen an mechatronische Systeme immer umfangreicher und komplexer werden.

Mechatronische Systeme ermöglichen zahlreiche Innovationen, deren erfolgreiche Umsetzung insbesondere in zukünftigen Fahrzeugen für die Wettbewerbsfähigkeit eines Automobilherstellers von erheblicher Bedeutung sein wird [149]. Dabei muss allerdings sichergestellt werden, dass die bestehenden Qualitätsstandards beibehalten werden. Wegen des zunehmenden Funktionsumfangs und der damit verbundenen steigenden Komplexität stellt die Erhaltung einer gleichbleibend hohen Produktqualität eine große Herausforderung dar.

Neue Systeme wie elektronische Lenksysteme und geregelte Allradsysteme bergen bei Fehlfunktionen hohe Risiken und potentielle Gefährdungen. Daraus ergeben sich umfassende Anforderungen an die Sicherheit der Systeme. Ein sicherheitsrelevantes Fehlverhalten muss in der Entwicklung systematisch ausgeschlossen werden.

Qualität, insbesondere Sicherheit, ist eine wesentliche nicht-funktionale Anforderung bei der Entwicklung eines mechatronischen Systems (Systementwicklung). Die Einhaltung von strukturierten Entwicklungsprozessen unterstützt die Erfüllung dieser nicht-funktionalen Anforderung. Dies spiegelt sich auch in Normen wie ISO/IEC 15504 [66] oder ISO 26262 [63] wider, welche die Einhaltung eines nachweisbaren und strukturierten Vorgehens in der Entwicklung fordern.

1.1 Problemstellung

Entwicklungsprozesse in der Automobilindustrie berücksichtigen auf Grund der Historie im Wesentlichen Vorgehensweisen aus der Mechanikentwicklung, bei welcher Konstruktionszeichnungen erstellt werden, die im Fertigungsprozess umzusetzen sind. Für mechatronische Systeme ist dieses Prozedere unzureichend. Der Entwicklungsprozess muss um Elektrik/Elektronik- und Softwareentwicklung ergänzt werden. Diese Erweiterung betrifft sowohl Engineeringprozesse wie Requirements Development, Architekturdesign, Umsetzung und Test als auch Supportprozesse wie Testmanagement oder Freigabemanagement. Eine Anpassung des herkömmlichen Entwicklungsvorgehens ist somit zwingend erforderlich.

Automobilhersteller ebenso wie Lieferanten tun sich schwer mit der Entwicklung mechatronischer Systeme, da die erforderliche Anpassung der Entwicklungsprozesse bisher nur ansatzweise erfolgt ist [147]. Detaillierte Vorgaben an die Prozessabläufe sind oft nur unzureichend vorhanden. Dadurch ist das Zusammenspiel verschiedener Entwicklungsaktivitäten für die Projektbeteiligten nicht transparent. Die Abläufe in der Praxis sind in Folge dessen teilweise nicht optimal aufeinander abgestimmt, was die Effizienz der Entwicklung beeinträchtigt.

Eine weitere Folge der unzureichenden Prozessvorgaben ist eine oft lückenhafte und inkonsistente Durchführung und Dokumentation von Entwicklungsschritten. Falls Ergebnisse zu einzelnen Aktivitäten nicht dokumentiert sind, ist deren Durchführung nicht nachweisbar. Treten Fehler in der Entwicklung auf, kann deren Ursache nicht nachvollzogen werden. Im schlimmsten Fall werden Fehler überhaupt nicht entdeckt. Damit ist nicht zuverlässig gewährleistet, dass das Ziel des Entwicklungsprozesses, nämlich hohe Qualität und Sicherheit bei gleichzeitig effizienten Arbeitsabläufen, erfüllt wird.

1.2 Ziel der Arbeit

Diese Arbeit soll einen Beitrag zur Verbesserung der Qualität der Entwicklungsabläufe für mechatronische Systeme in der Automobilindustrie leisten. Auf Grund des Umfangs der gesamten Systementwicklung wird hier nur ein Teilabschnitt der Entwicklung betrachtet. Hierzu wird von einer Unterteilung der Systementwicklung in die drei Phasen *Vorentwicklung, Serienentwicklung* und *Serienbetreuung* ausgegangen (siehe Abbildung 1.1). Die Vorentwicklung umfasst die Herleitung und prototypische Umsetzung unterschiedlicher Realisierungsalternativen. In der Serienentwicklung wird eines der Konzepte aus der Vorentwicklung bis zur Serienreife weiterentwickelt. Nach Produktionsstart (engl.: start of production, SOP) erfolgen im Rahmen der Serienbetreuung die kontrollierte Durchführung notwendiger Änderungen sowie Wartungsaktivitäten. Die vorliegende Arbeit beschäftigt sich schwerpunktmäßig mit der Serienentwicklung.

Abbildung 1.1: Entwicklungsphasen einer Systementwicklung in der Automobilindustrie

Die Grundlage für die Serienentwicklung bilden Ergebnisse aus der Vorentwicklung, zu welchen auch ein Systemprototyp zählt. Zentrale Aufgabe der Serienentwicklung ist es, ausgehend von den Vorentwicklungsergebnissen ein serienfähiges Konzept zu erarbeiten und umzusetzen. Hierzu müssen insbesondere

- die bereits in einem Systemprototyp umgesetzten funktionalen Anforderungen vervollständigt und nicht-funktionale Anforderungen, beispielsweise an Systemsicherheit, Verfügbarkeit sowie Umweltverträglichkeit ergänzt werden.
- funktionale und nicht-funktionale Anforderungen für das mechatronische System und dessen Teilsysteme dokumentiert werden.
- das Systemkonzept in Bezug auf diese Anforderungen verifiziert und gegebenenfalls die Systemarchitektur angepasst werden.
- eine weitestgehend fehlerfreie Umsetzung des mechatronischen Systems und seiner Teilsysteme durch Entwicklungs- und Produktionsprozesse gewährleistet werden.

- die korrekte Umsetzung funktionaler und nicht-funktionaler Anforderungen durch Verifikations- und Validierungstätigkeiten sichergestellt werden.

Die Durchführung dieser Aufgaben erfordert den Einsatz von Spezialisten aus unterschiedlichen Fachbereichen und Unternehmen. Zudem müssen die Aktivitäten möglichst simultan durchgeführt werden, um die Zeit bis zur Markteinführung gering zu halten. Daraus ergibt sich ein äußerst komplexer Entwicklungsablauf. Eine detailliert festgelegte, strukturierte Vorgehensweise bringt daher in dieser Entwicklungsphase besonders großen Nutzen.

Ziel dieser Arbeit ist es, für die Phase der Serienentwicklung ein Konzept bereitzustellen, welches

- die Effizienz des Entwicklungsablaufs erhöht (*Prozessoptimierung*).
- die Einhaltung der Prozessabläufe aller Entwicklungsaktivitäten, also von Engineering- und Supportprozessen, unterstützt (*Prozesseinhaltung*).
- den Nachweis der Durchführung aller Entwicklungsaktivitäten vereinfacht (*Nachweisbarkeit*).
- möglichst allgemein angewandt werden kann (*weitreichende Anwendbarkeit*).
- die Einbindung in die bereits bestehenden Entwicklungsabläufe in der Automobilindustrie ermöglicht (*Integrierbarkeit*).
- in der Praxis einfach umgesetzt werden kann (*Umsetzbarkeit*).
- eine Steuerung der Entwicklungsabläufe durch Werkzeuge (engl.: tools) ermöglicht (*Werkzeugunterstützung*).

Wie aus dem Titel dieser Arbeit hervorgeht, soll das eben formulierte Ziel durch die Erarbeitung eines Referenzmodells für die Serienentwicklung mechatronischer Systeme in der Automobilindustrie erreicht werden.

1.3 Wissenschaftliche Beiträge der Arbeit

Durch die Bereitstellung des Referenzmodells wird der Stand der Wissenschaft um eine Darstellung der Soll-Prozessabläufe für die Serienentwicklung mechatronischer Systeme in der Automobilindustrie ergänzt. Dabei geht die vorliegende Arbeit insbesondere in den folgenden Punkten über die bestehenden Arbeiten hinaus:

- *Engineeringprozesse:*
  - *Verifikation vertikale Verfolgbarkeit:* Es wird ein Vorgehen für die Verifikation der Konsistenz von System- und Teilsystemanforderungen erarbeitet, welches die Überlegungen der bestehenden Arbeiten [94, 96] weiterführt.
  - *Sicherheitsanalysen:* Es wird die Einbindung der Sicherheitsanalysen G&R [63], FMEA [99, 146, 151] und FTA [61, 152] in den Entwicklungsablauf erarbeitet sowie ein Vorgehen für eine effiziente Erstellung der FMEA ausgehend von bereits vorliegenden Entwicklungsergebnissen bereitgestellt.
  - *Entwicklungsablauf:* Es werden die Prozessabläufe der Serienentwicklung abhängig von unterschiedlichen Reifegraden modelliert und damit gezeigt, wie bei einer iterativen Entwicklung die Entwicklungsaktivitäten schrittweise erweitert werden können.

- *Artefakte-Abhängigkeitsmodell:* Die in der Literatur vorliegenden Artefakte-Abhängigkeitsmodelle [86, 90, 108] werden um eines für die Serienentwicklung mechatronischer Systeme ergänzt.
- *Workflows:* In der vorliegenden Arbeit wird, im Gegensatz zu den bestehenden Arbeiten (siehe Kapitel 4), eine Darstellung der Soll-Prozessabläufe aus Artefaktesicht durch Workflows bereitgestellt, wodurch eine werkzeuggestützte Umsetzung des Vorgehens in der Praxis unterstützt wird.
- *Supportprozesse:* Im Referenzmodell werden zusätzlich zu den Engineeringprozessen Prozessabläufe von Supportprozessen, welche die Planung und Steuerung der Entwicklungsiterationen unterstützen, berücksichtigt und damit der Betrachtungsumfang bezüglich der Supportprozesse im Hinblick auf bestehende Arbeiten [31, 75] deutlich erweitert.

Im Einzelnen wird auf die Beiträge der Arbeit in den Zusammenfassungen der Abschnitte von Kapitel 6 bei der Erstellung des Referenzmodells eingegangen. Ein detaillierter Überblick über die Beiträge wird in Abschnitt 8.2 gegeben.

Insgesamt wird durch die vorliegende Arbeit der Stand der Wissenschaft um eine ganzheitliche Darstellung der Prozessabläufe der Serienentwicklung mechatronischer Systeme in der Automobilindustrie ergänzt. Im folgenden Abschnitt werden die einzelnen Schritte der Herleitung des Referenzmodells erläutert.

1.4 Aufbau der Arbeit

Die vorliegende Arbeit unterteilt sich in acht Kapitel (siehe Abbildung 1.2). Im aktuellen Kapitel 1 wurde die Problemstellung allgemein erläutert (Abschnitt 1.1), die Zielsetzung für diese Arbeit abgeleitet (Abschnitt 1.2) und ein Überblick über die wesentlichen wissenschaftlichen Beiträge der Arbeit gegeben (Abschnitt 1.3).

In Kapitel 2 wird ein Überblick über Begriffe zur Systementwicklung gegeben. In diesem Zusammenhang erfolgt für Aktivitäten der Systementwicklung, welche in der Literatur unterschiedlich bezeichnet werden, die Begriffsbildung für diese Arbeit. Dem Leser wird damit die in den weiteren Kapiteln verwendete Nomenklatur vermittelt.

In Kapitel 3 wird die Konzeption der Arbeit vorgestellt. Dabei wird die methodische Herangehensweise, nämlich die Anwendung einer Referenzmodellierung im Kontext der Prozessmodellierung, erläutert. Zudem wird der Betrachtungsumfang des Referenzmodells innerhalb der Fahrzeugentwicklung abgegrenzt. Der Leser erfährt in diesem Kapitel, inwiefern durch die Bereitstellung eines Referenzmodells das in Abschnitt 1.2 formulierte Ziel erreicht werden kann, wie bei der Erstellung des Referenzmodells vorgegangen wird und wie sich der Betrachtungsumfang in die Fahrzeugentwicklung einordnet.

In Kapitel 4 wird ein Überblick der für diese Arbeit relevanten wissenschaftlichen Arbeiten gegeben. Ausgehend davon wird die vorliegende Arbeit in die bestehenden wissenschaftlichen Arbeiten eingeordnet und von diesen abgegrenzt. Dem Leser wird damit gezeigt, wie sich die vorliegende Arbeit in das wissenschaftliche Umfeld eingliedert.

In Kapitel 5 wird beispielhaft ein bestehender Prozessablauf der Serienentwicklung für die Fahrwerkentwicklung dargestellt. Im Rahmen der Ist-Prozess-Analyse werden Optimierungsmöglichkeiten des bestehenden Prozessablaufs aufgezeigt und somit die in Kapitel 1.1 beschriebene Problemstellung in-

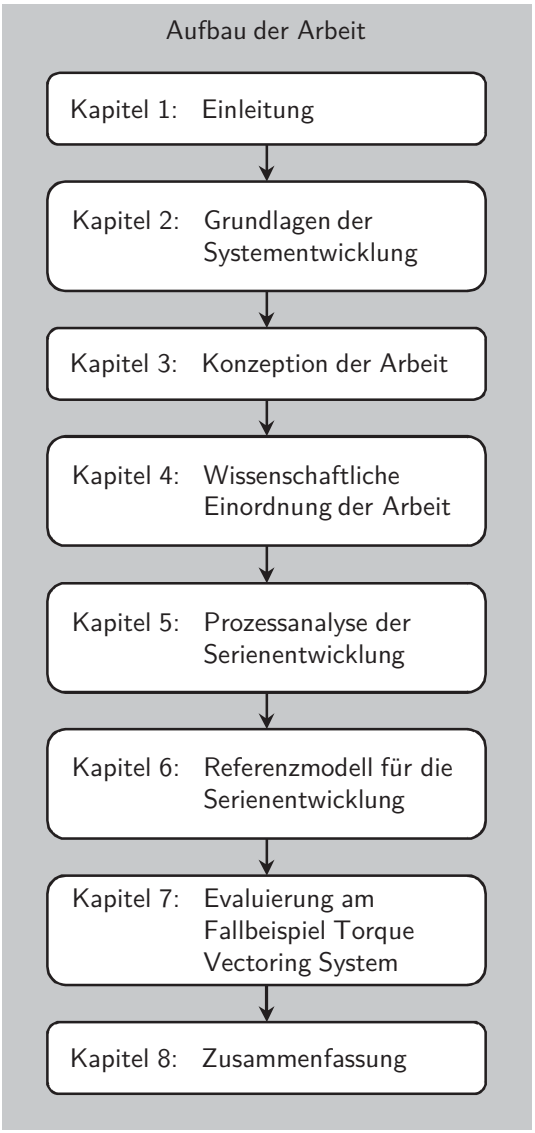

Abbildung 1.2: Aufbau der Arbeit

haltlich bezüglich der Serienentwicklung detailliert. Dem Leser wird damit die Problemstellung aus Praxis-Sicht näher gebracht.

In Kapitel 6 wird das Referenzmodell für die Serienentwicklung mechatronischer Systeme in der Automobilindustrie erarbeitet. Entsprechend der in Kapitel 3 vorgestellten Methodik wird der Soll-Prozess zuerst für die Engineeringprozesse modelliert. Anschließend werden schrittweise die Supportprozesse in das Referenzmodell integriert. Die Erstellung des Referenzmodells erfolgt dabei unter Berücksichtigung der in Kapitel 3 zugrunde gelegten Rahmenbedingungen mit dem Ziel der Umsetzung der in Kapitel 5 identifizierten Optimierungsmöglichkeiten. Dem Leser wird in diesem Kapitel ein Überblick über das Konzept des Referenzmodells für die Serienentwicklung aus theoretischer Sicht gegeben.

In Kapitel 7 wird das Referenzmodell am Beispiel der Entwicklung eines Torque Vectoring Systems, also an Hand eines Fahrwerkregelsystems, durch welches das Antriebsmoment an der Hinterachse situationsabhängig variabel zwischen rechtem und linkem Rad verlagert werden kann [92, 125], evaluiert. Die hergeleiteten Entwicklungsartefakte sowie die zugehörigen Workflows werden dabei im Werkzeug Integrity[1] [106] umgesetzt und ausgeführt. Dem Leser wird damit gezeigt, wie das erarbeitete Referenzmodell in der Praxis angewendet werden kann.

In Kapitel 8 erfolgt eine Zusammenfassung dieser Arbeit. Dabei werden die wissenschaftlichen Beiträge dieser Arbeit herausgestellt. Zudem wird ein Ausblick auf weiterführende Fragestellungen gegeben.

Vorabveröffentlichungen: Einzelne Inhalte dieser Arbeit wurden bereits in [112–115] veröffentlicht.

[1] Integrity ist ein Werkzeug der Firma PTC zur Unterstützung der Entwicklung unter anderem in den Bereichen Requirements Management, Konfigurationsmanagement und Testmanagement.

2 Grundlagen der Systementwicklung

Die vorliegende Arbeit befasst sich mit der Entwicklung mechatronischer Systeme in der Automobilindustrie. Ziel dieses Kapitels ist es, einen Überblick über Grundlagen der Systementwicklung zu geben, soweit sie für diese Arbeit benötigt werden. Da Begriffe im Kontext der Systementwicklung hinsichtlich Nomenklatur und Bedeutung unterschiedlich verwendet werden, wird in diesem Kapitel zugleich die Begriffsbildung für diese Arbeit durchgeführt.

In Abschnitt 2.1 werden grundlegende Begriffe der Systementwicklung eingeführt. In den Abschnitten 2.2 und 2.3 wird auf die in dieser Arbeit verwendete Definition der Engineering- und Supportprozesse eingegangen. Abschnitt 2.4 befasst sich mit der Verfolgbarkeit von Anforderungen entlang ihres Lebenszyklus (Traceability). In Abschnitt 2.5 werden die wesentlichen Aussagen dieses Kapitels zusammengefasst.

2.1 Begriffe zur Systementwicklung

2.1.1 System und Systemarchitektur

Ein *System* wird durch seine (syntaktische) *Schnittstelle*, welche sich aus *Eingangsgrößen* und *Ausgangsgrößen* zusammensetzt (siehe Abbildung 2.1), zur Umgebung abgegrenzt. Abhängig von den Werten der Eingangsgrößen wirkt das System über die Ausgangsgrößen auf die Umgebung. Der funktionale Zusammenhang zwischen Eingangs- und Ausgangsgrößen wird als *Systemverhalten* (semantische Schnittstelle [21]) bezeichnet. Ein System kann beispielsweise ein Automobil, das Fahrwerk eines Automobils oder ein einzelnes mechatronisches System wie das Torque Vectoring System sein. Diese Arbeit befasst sich mit *mechatronischen Systemen*, also mit Systemen, deren Verhalten durch die Interaktion von Mechanik, Elektronik und Informatik bestimmt wird [58].

Die komplexe Funktionalität mechatronischer Systeme sowie das Zusammenwirken verschiedener Fachdisziplinen machen eine weitere Unterteilung des Systems in *Teilsysteme* erforderlich. Häufig werden

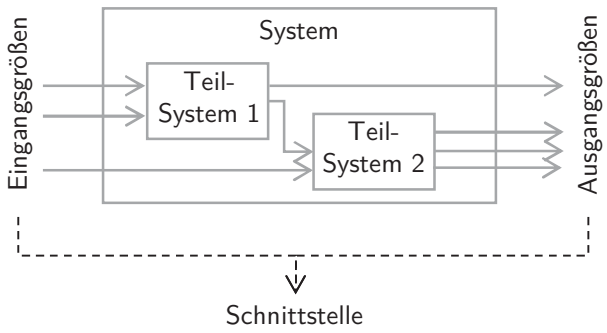

Abbildung 2.1: Schematische Darstellung grundlegender Begriffe eines Systems

mechatronische Systeme in Sensorik, Logik und Aktuatorik unterteilt. Die einzelnen Teilsysteme bilden dabei ihrerseits jeweils ein System und können in weitere Teilsysteme zerlegt werden. Durch die Anordnung der Teilsysteme zueinander sowie deren Zusammenspiel wird die *Architektur* des Systems festgelegt.

2.1.2 Klassifizierungen von Anforderungen

Ein System soll ein bestimmtes Verhalten aufweisen und über Eigenschaften wie Zuverlässigkeit oder Bedienbarkeit verfügen. Das geforderte Verhalten sowie die erwarteten Eigenschaften werden als Anforderungen an das System bezeichnet. In der IEEE 610 [134] wird der Begriff der Anforderung folgendermaßen definiert[2]:

Eine Anforderung ist

1. eine Bedingung oder Eigenschaft, die der Nutzer eines Systems benötigt, um ein Problem zu lösen oder ein Ziel zu erreichen.
2. eine Bedingung oder Eigenschaft, die ein System aufweisen muss, um einen Vertrag, einen Standard, eine Spezifikation oder andere formale Vorgaben zu erfüllen.
3. eine dokumentierte Repräsentation einer Bedingung oder Eigenschaft wie in 1 oder 2 definiert.

Nach dieser Definition muss eine Anforderung nicht eindeutig als solche definiert und abgegrenzt sein, auch muss sie nicht in dokumentierter Form vorliegen. Es ist ausreichend, wenn der Nutzer des Systems die geforderte Eigenschaft oder Bedingung kennt, oder diese in einem Dokument enthalten ist. In dieser Arbeit wird der Begriff der Anforderung enger gefasst und wie folgt verwendet:

Eine *Anforderung* ist eine dokumentierte, eindeutig von weiteren Informationen abgegrenzte Repräsentation eines vom System erwarteten Verhaltens oder einer vom System geforderten nicht verhaltensprägenden Eigenschaft.

Anforderungen durchlaufen in der Entwicklung verschiedene Arbeitsschritte. Die Abfolge dieser Arbeitsschritte, die sich aus dem Entwicklungsprozess ergibt, wird als *Lebenszyklus einer Anforderung* bezeichnet.

Die Anforderungen an ein System sowie weitere für das Systemverständnis erforderliche Informationen wie die Systembeschreibung bilden die *Anforderungsspezifikation* des Systems. Vorlagen für Anforderungsspezifikationen [84, 117, 136, 150] greifen auf verschiedene Möglichkeiten der Klassifizierung von Anforderungen [102, 119] zurück. In dieser Arbeit werden funktionale und nicht-funktionale Anforderungen unterschieden.

Eine *funktionale Anforderung* ist eine Anforderung, die das Systemverhalten (teilweise) festlegt. Wird durch die Anforderung kein Verhalten beschrieben, so wird die Anforderung als *nicht-funktional* bezeichnet. Funktionale Anforderungen legen für Definitionsbereiche von Eingangsgrößen die erwarteten Werte(bereiche) sowie das Zeitverhalten[3] für Ausgangsgrößen des Systems fest. Nicht-funktionale Anforderungen wie Verfügbarkeit oder Sicherheit können systemübergreifend sein und haben im Allgemeinen keinen direkten Bezug zu einzelnen Schnittstellengrößen.

[2] Die Definition in [134] erfolgt auf Englisch. Die hier verwendete Definition der Anforderung entspricht der Übersetzung der Formulierung aus [134].

[3] Die zeitlichen Anforderungen hinsichtlich der Systemreaktion werden als Performanzanforderungen teilweise auch den nicht-funktionalen Anforderungen zugeordnet. Im Kontext mechatronischer Systeme im Fahrzeug wird das Systemverhalten durch Systemreaktion und Systemreaktionszeit festgelegt. Daher wird in dieser Arbeit das Zeitverhalten als Teil der funktionalen Anforderungen gesehen.

Eine weitere wichtige Klassifizierung von Anforderungen ergibt sich aus deren Sicherheitsbezug. Unter *Sicherheit* wird dabei verstanden, dass ein System frei von unvertretbaren Risiken ist [36], was dem englischen Begriff „safety“[4] entspricht. Eine Anforderung, deren fehlerhafte Umsetzung oder fehlerhafte Ausführung der Umsetzung zu einer Gefährdung für Mensch oder Umwelt führen kann, wird als *sicherheitsrelevante Anforderung* bezeichnet. *Sicherheitsanforderungen* werden Anforderungen genannt, welche die Umsetzung von Sicherheitsfunktionen fordern [36]. Eine *Sicherheitsfunktion* ist eine Funktion, die zur Risikominderung ausgeführt wird, um bei Eintreten eines festgelegten gefährlichen Vorfalls einen sicheren Systemzustand zu erreichen oder aufrecht zu erhalten [36].

2.1.3 Konkretisierung funktionaler Anforderungen und Maßnahmen zu nicht-funktionalen Anforderungen

Wird ein System in Teilsysteme unterteilt, so muss die Erfüllung der Systemanforderungen durch die interagierenden Teilsysteme sichergestellt werden. Daher werden bei der Festlegung der Systemarchitektur funktionale Anforderungen gegebenenfalls in mehrere Teilsystemanforderungen unterteilt (*Anforderungsdekomposition*) und den Teilsystemen zugeordnet (*Anforderungszuordnung*). Zugleich kann hierbei auch eine inhaltliche *Detaillierung* der Anforderungen erfolgen. Die Anforderungsdekomposition, Anforderungszuordnung und Detaillierung funktionaler Anforderungen im Rahmen der Festlegung der Systemarchitektur wird im Weiteren als *Konkretisierung* funktionaler Anforderungen bezeichnet und ausführlich in Abschnitt 6.1.2 analysiert. Abbildung 2.2 visualisiert die Konkretisierung funktionaler Anforderungen. Im linken grau hinterlegten Bereich ist die Dekomposition der funktionalen Anforderungen, im rechten grau hinterlegten Bereich die Unterteilung des Systems in Teilsysteme dargestellt. Die zunehmende Detaillierung der funktionalen Anforderungen wird durch den Detaillierungsgrad links im Bild angedeutet. Die Zuordnung der Anforderungen zum System und dessen Teilsystemen wird durch die gestrichelten Pfeile gezeigt.

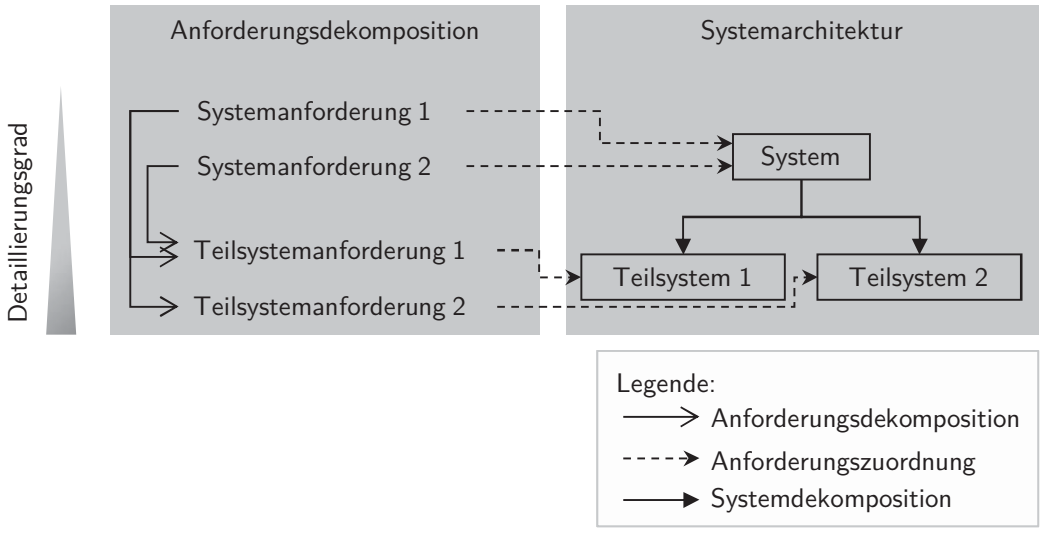

Abbildung 2.2: Konkretisierung funktionaler Anforderungen

[4] Der deutsche Begriff Sicherheit wird für die englischen Begriffe safety und security verwendet. Safety bedeutet, dass vom Produkt keine Gefährdung ausgeht, während security den Schutz des Systems vor äußeren Angriffen bezeichnet.

Im Gegensatz zu funktionalen Anforderungen müssen aus nicht-funktionalen Systemanforderungen nicht zwingend nicht-funktionale Teilsystemanforderungen resultieren. Beispielsweise muss die nicht-funktionale Anforderung der Modularität eines Systems direkt in der Systemarchitektur berücksichtigt werden. Auch die häufig in der Automobilindustrie vorzufindende nicht-funktionale Anforderung, dass die Entwicklungsprozesse für die im Automotive SPICE Prozessreferenzmodell [144] definierten Prozesse den Reifegrad Level 2 entsprechend dem Automotive SPICE Prozess Assessment Modell [143] erfüllen müssen, führt nicht zu Teilsystemanforderungen, sondern muss bei der Definition und Durchführung des Entwicklungsprozesses beachtet werden.

Die Erfüllung nicht-funktionaler Systemanforderungen kann auf verschiedene Arten erreicht werden, worauf in Abschnitt 6.1.2 ausführlich eingegangen wird. So können neben den eben aufgeführten Beispielen aus nicht-funktionalen Anforderungen unter anderem auch funktionale Anforderungen resultieren. Häufig werden dabei mehrere *Maßnahmen* aus einer nicht-funktionalen Anforderung abgeleitet, um deren Erfüllung sicherzustellen.

Bei der Festlegung von Maßnahmen ist es hilfreich, auf Erfahrungen aus vorangegangenen Entwicklungen (engl.: best practices) sowie Normen und Richtlinien zurückzugreifen. So kann die Erfüllung der nicht-funktionalen Anforderungen mit einer hohen Wahrscheinlichkeit sichergestellt werden.

2.1.4 Systemstrukturbaum und Funktionsnetz

Die Unterteilung des Systems in Teilsysteme, wie auch die Dekomposition von Anforderungen, lässt sich in Form von Graphen übersichtlich darstellen. Die Aufteilung des Systems in Teilsysteme kann in Form eines *Systemstrukturbaums* [99] dargestellt werden (siehe Abbildung 2.3). Dabei bildet das *System* die Wurzel, die *Teilsysteme* die inneren Knoten, und die atomaren Teilsysteme (*Module*) die Blätter des Baums. Keine Berücksichtigung findet im Systemstrukturbaum das Zusammenspiel der Teilsysteme durch den Signalfluss.

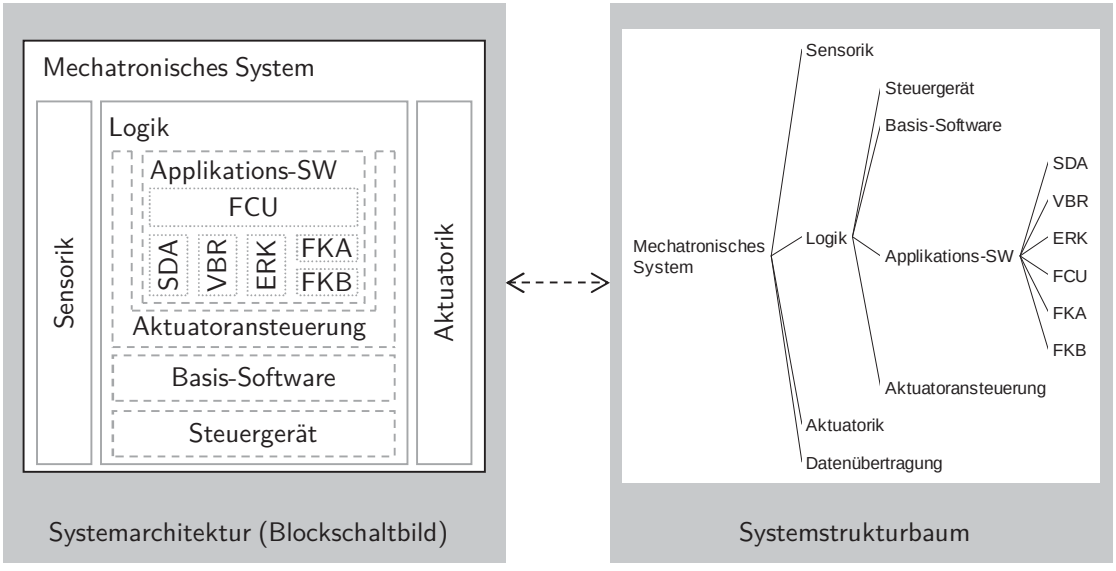

Abbildung 2.3: Systemarchitektur und Systemstrukturbaum eines mechatronischen Systems

Beispiel 2.1: Ein mechatronisches System setzt sich aus den Teilsystemen Sensorik, Logik, Aktuatorik sowie der Datenübertragung zwischen diesen Teilsystemen zusammen. Die Logik lässt sich wei-

ter unterteilen in Steuergerät (Hardware)[5], Basis-Software, Aktuatoransteuerung und Applikations-Software. Die Applikations-Software setzt sich beispielsweise aus Signaldatenaufbereitung (SDA), Vorberechnung (VBR), Erkennung von Fahrzuständen (ERK), Überwachung (FCU), sowie Reglerfunktion A (FKA) und Reglerfunktion B (FKB) zusammen. Weitere Unterteilungen der Teilsysteme des mechatronischen Systems werden hier aus Gründen der Übersichtlichkeit nicht näher beschrieben. Das Blockdiagramm zu der eben erläuterten Architektur ist in Abbildung 2.3 links skizziert. Der zugehörige Systemstrukturbaum ist auf der rechten Seite der Abbildung dargestellt.

Im Gegensatz zur Systemstruktur können die Abhängigkeiten zwischen funktionalen System- und Teilsystemanforderungen nicht als Baum dargestellt werden. Eine funktionale Teilsystemanforderung kann die zentrale Berechnung einer Größe beschreiben, die zur Erfüllung mehrerer funktionaler Systemanforderungen notwendig ist. Eine Teilsystemanforderung kann also für die Erfüllung mehrerer funktionaler Systemanforderungen notwendig sein. Die Abhängigkeiten zwischen funktionalen System- und Teilsystemanforderungen bilden daher ein Netz, welches als *Funktionsnetz (Anforderungsnetz)* (siehe [99], S. 26/27) bezeichnet wird. Durch das Funktionsnetz wird also dargestellt, welche Teilsystemanforderungen aus Systemanforderungen abgeleitet werden und damit die Dekomposition der Anforderungen abgebildet.

Die Verwendung des Begriffs Funktionsnetz in dieser Form ist im Zusammenhang mit der Erstellung einer *Fehlermöglichkeits- und Einflussanalyse (FMEA, engl.: Failure Modes and Effects Analysis)* [99, 146, 151] üblich. Dagegen muss das *Fahrzeugfunktionsnetz* abgegrenzt werden. In diesem werden die Funktionen auf Fahrzeugebene sowie deren Teilfunktionen zueinander in Bezug gesetzt. Das Fahrzeugfunktionsnetz liefert einen Überblick über funktionale Abhängigkeiten auf Fahrzeugebene, während das Funktionsnetz die Abhängigkeiten der Anforderungen[6] an ein System und seine Teilsysteme darstellt.

Die Konkretisierung funktionaler Anforderungen erfolgt im Rahmen der Festlegung der Systemarchitektur. Damit ist eine eindeutige Zuordnung der Anforderungen zum System und zu Teilsystemen gegeben, was zugleich bedeutet, dass das Funktionsnetz in den Systemstrukturbaum integriert werden kann (siehe Abbildung 2.4). Da im Funktionsnetz ausschließlich die Anforderungsdekomposition abgebildet wird und das Systemverhalten durch das Zusammenspiel der Teilsysteme realisiert wird, sind als Nachfolger einer funktionalen Anforderungen eines Knotens des Systemstrukturbaums nur Anforderungen der Kinder des Systemstrukturbaumknotens zulässig. Damit sind die in Abbildung 2.4 gestrichelten Abhängigkeiten in einem Funktionsnetz unzulässig. Insgesamt bildet das Funktionsnetz somit ein Netz innerhalb des Systemstrukturbaums, bei welchem ausschließlich Abhängigkeiten zwischen Anforderungen eines Systemstrukturbaumknotens und Anforderungen von dessen Vorgänger- bzw. Nachfolgerknoten bestehen.

2.1.5 Entwicklungsprozess, Vorgehensmodelle und Prozessreifegradmodelle

Bei der Entwicklung eines Systems werden unterschiedliche *Aktivitäten*, wie beispielsweise der Entwurf der Systemarchitektur, durchgeführt. Die Ergebnisse der Aktivitäten werden im Weiteren als *Artefakte*[7] bezeichnet. Für eine erfolgreiche Entwicklung ist es erforderlich, dass die Abfolge der Entwicklungs-

[5] In diesem Beispiel wird vereinfachend davon ausgegangen, dass die Logik vollständig auf einem Steuergerät realisiert wird.

[6] Durch die Anforderungen an die Systeme werden die auf Fahrzeugebene abstrakt beschriebenen Funktionen inhaltlich detailliert.

[7] Unter dem Begriff Artefakt werden Arbeitsergebnisse in dokumentierter und undokumentierter Form verstanden.

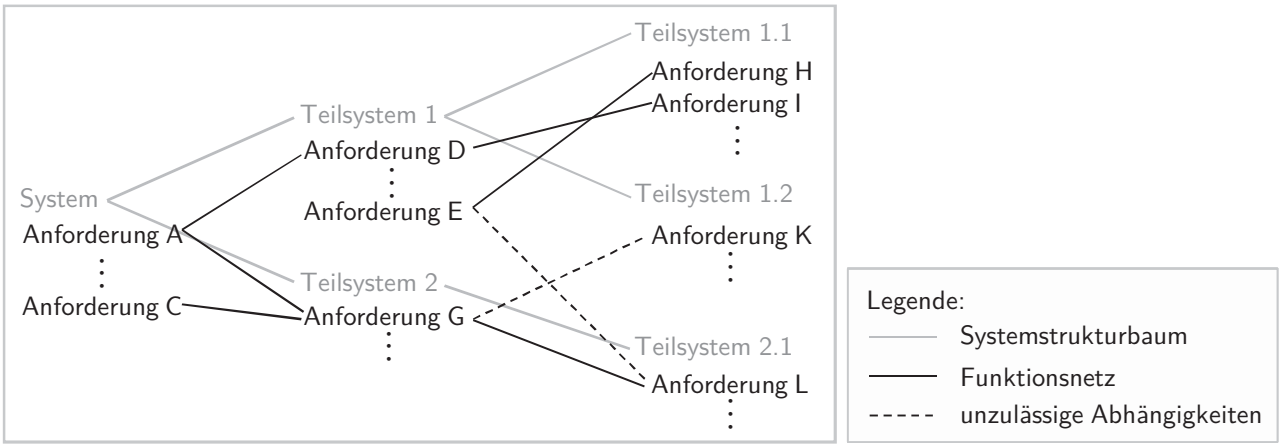

Abbildung 2.4: Funktionsnetz im Systemstrukturbaum

aktivitäten klar festgelegt ist. Die Abfolge der Entwicklungsaktivitäten wird als *Entwicklungsprozess* bezeichnet.

Bei der Definition des Entwicklungsprozesses kann auf *Vorgehensmodelle* zurückgegriffen werden. Durch solche Modelle werden Möglichkeiten der logischen Anordnung der Entwicklungsaktivitäten vorgeschlagen. Beispiele für Vorgehensmodelle sind das Wasserfallmodell [118], das Spiralmodell [14], der Rational Unified Process [153], das V-Modell 97 [33] sowie das V-Modell XT [22].

Das Vorgehen in der Systementwicklung in der Automobilindustrie ist nicht standardisiert, allerdings wird häufig das V-Modell als Vorgehensmodell herangezogen (siehe beispielsweise [41, 123]). Grund hierfür ist die dem V-Modell zugrundeliegende Idee der stufenweisen Absicherung des Produktes sowie die graphische Darstellung, die eine einfache Erläuterung der durchzuführenden Aktivitäten ermöglicht. Zusätzlich zum V-Modell fließt bei der Festlegung der Entwicklungsprozesse in der Automobilindustrie auch der dem Spiralmodell zugrundeliegende Gedanke der iterativen Entwicklung ein. Auch in dieser Arbeit wird sowohl die Idee des V-Modells, das Produkt stufenweise zu validieren, als auch die aus dem Spiralmodell bekannte iterative Entwicklung berücksichtigt.

Die Festlegung des Entwicklungsprozesses für eine Systementwicklung, ausgehend von einem Vorgehensmodell, ist eine wichtige Aktivität zu Beginn der Entwicklung. Während der Entwicklung muss überprüft werden, ob Anpassungen im Entwicklungsprozess, beispielsweise auf Grund nicht berücksichtigter Rahmenbedingungen, erforderlich sind. Die Identifikation von Prozessverbesserungsmaßnahmen kann strukturiert unter Verwendung von *Prozessreifegradmodellen* wie SPICE (Software Process Improvement and Capability Determination, [66]), CMMI (Capability Maturity Model Integration, [27]) oder TPI (Test Process Improvement, [37]), welches speziell auf die Optimierung des Testprozesses abzielt, erfolgen.

In Vorgehensmodellen, Prozessdefinitionen in der Praxis und Prozessreifegradmodellen werden unterschiedliche Terminologien für die durchzuführenden Entwicklungsaktivitäten verwendet. Aus diesem Grund erfolgt im Weiteren die Begriffsbildung für diese Arbeit. Dabei wird eine Unterteilung der Prozesse in Engineeringprozesse und Supportprozesse vorgenommen. Als *Engineeringprozesse* werden die Teilprozesse bezeichnet, die sich direkt mit der Entwicklung des Systems befassen. Die Teilprozesse, die der Planung, Strukturierung und methodischen Unterstützung der Systementwicklung dienen, werden *Supportprozesse* genannt.

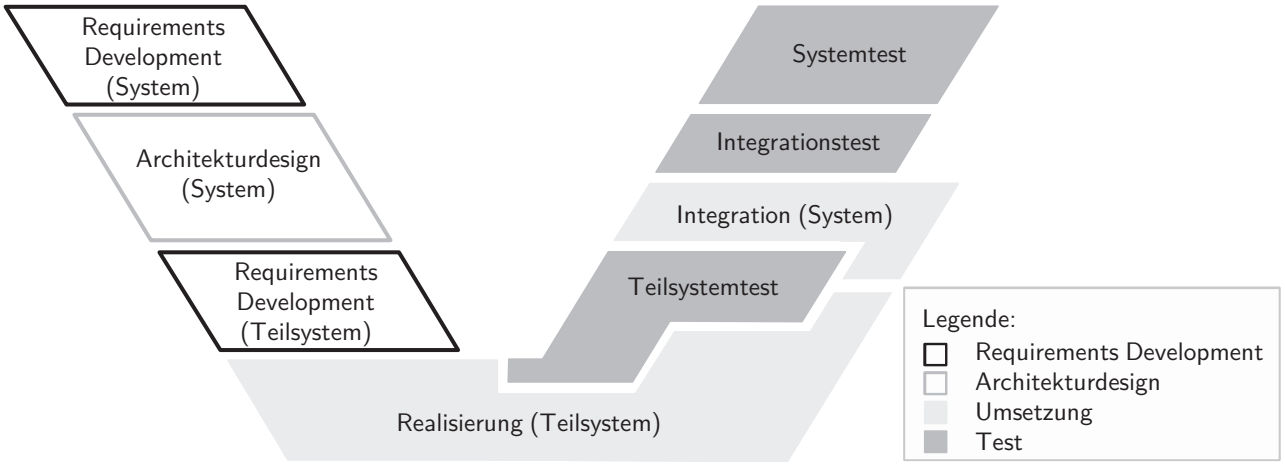

Abbildung 2.5: Überblick über die Engineeringprozesse

2.2 Engineeringprozesse

Zu den Engineeringprozessen werden in dieser Arbeit Requirements Development, Architekturdesign, Umsetzung und Test gezählt. Diese Prozesse sind in Abbildung 2.5 entsprechend dem V-Modell dargestellt. Das Requirements Development ist durch schwarz umrandete Felder, das Architekturdesign durch grau umrandete Felder, die Umsetzung (Realisierung und Integration) durch die hellgrauen Felder sowie der Test durch dunkelgraue Felder dargestellt.

Ziele und wesentliche Aufgaben zu den einzelnen Engineeringprozessen werden in den folgenden Abschnitten für diese Arbeit festgelegt. Die einzelnen Aufgaben sind dabei im Wesentlichen aus den oben erwähnten Vorgehensmodellen [14, 22, 33, 118, 153] und Prozessreifegradmodellen [27, 66] bereits bekannt.

Bei der Erläuterung zu den Engineeringprozessen werden speziell Aktivitäten aufgeführt, die bei der Entwicklung sicherheitsrelevanter Systeme durchzuführen sind, um die Sicherheit des Systems zu gewährleisten. Als *sicherheitsrelevantes System* wird hierbei ein System bezeichnet, wenn es bei fehlerhafter Umsetzung oder fehlerhafter Ausführung zu einer Gefährdung für Mensch oder Umwelt führen kann [36]. Weiterführende Erläuterungen zu den sicherheitsbezogenen Aktivitäten können beispielsweise [12, 36, 63, 87] entnommen werden.

2.2.1 Requirements Development

Das Requirements Development wird in dieser Arbeit als ein Bestandteil des Teilprozesses Requirements Engineering verstanden.

Der Begriff *Requirements Engineering* wird in der umfangreichen wissenschaftlichen Literatur unterschiedlich verwendet [95]. So wird in [101, 155] unter Requirements Engineering der Prozess der Erarbeitung einer widerspruchsfreien Spezifikation verstanden. In [89, 128] wird dem Requirements Engineering zudem die Verwaltung von Anforderungen zugeordnet. In [154] wird schließlich eine Unterteilung des Requirements Engineering in die Teilprozesse *Requirements Development* und *Requirements Management* vorgenommen (siehe Abbildung 2.6). Dabei werden dem Requirements Development alle Aktivitäten zugeordnet, welche zur Herleitung und Dokumentation einer widerspruchsfreien Spezifika-

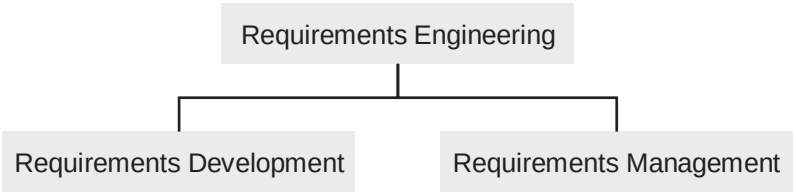

Abbildung 2.6: Unterteilung des Requirements Engineering

tion erforderlich sind. Zum Requirements Management zählen die Verwaltung und Bereitstellung von Anforderungen sowie Informationen zu diesen. In dieser Arbeit wird das Requirements Engineering wie in [154] in Requirements Development und Requirements Management unterteilt.

Ziel des Requirements Development ist die Erstellung der Spezifikation für ein System[8]. Im Zuge des Requirements Development müssen folgende Aktivitäten durchgeführt werden:

- *Erhebung* der Informationsquellen (Personen und Dokumente), durch die Forderungen an das System gestellt werden.
- *Analyse* der erhobenen Informationen hinsichtlich funktionaler und nicht-funktionaler Anforderungen.
- *Dokumentation* der erhobenen und analysierten Anforderungen sowie der Information über deren Herkunft in einer strukturierten Form.

Bei der Entwicklung sicherheitsrelevanter Systeme müssen insbesondere folgende Aktivitäten durchgeführt werden:

- *Sicherheitsrelevante Anforderungen ermitteln:* Es muss ermittelt werden, welche Anforderungen als sicherheitsrelevant einzustufen sind. Hierzu muss für die Funktionen des mechatronischen Systems eine *Gefährdungsanalyse und Risikobewertung* (G&R, engl.: hazard analysis and risk assessment) [12, 36, 63] durchgeführt werden. Hierbei werden potentielle Gefährdungen, die bei fehlerhafter Umsetzung oder Ausführung einer Funktion auftreten können, analysiert und bewertet. Aus der G&R lässt sich die Sicherheitseinstufung, der sogenannte *Automotive Sicherheitsintegritäts-Level*[9] (ASIL, engl.: automotive safety integrity level) für Anforderungen an das mechatronische System ableiten. Für Anforderungen an Teilsysteme des mechatronischen Systems kann der ASIL der Anforderungen aus der Einstufung der Anforderungen des übergeordneten Systems ermittelt werden, siehe ISO 26262[10] [63], Teil 9, Abschnitt 5.
- *Sicherheitsanforderungen ableiten:* Zu den als sicherheitsrelevant eingestuften Anforderungen müssen Sicherheitsanforderungen abgeleitet werden, in denen Sicherheitsfunktionen festgelegt werden, welche das Eintreten einer Gefährdung im Fehlerfall verhindern. Hierzu muss zu den sicherheitsrelevanten Anforderungen das *Sicherheitskriterium*, also der Grenzwert ab dem ein Fehler zu einer Gefährdung führt, ermittelt werden.
- *Sicherheitsintegritäts-Level des Systems ableiten:* Aus den ASIL's der Anforderungen muss der ASIL des Systems abgeleitet werden. Dieser entspricht der maximalen Sicherheitseinstufung der

[8] Als System kann hier auch ein Teilsystem eines mechatronischen Systems betrachtet werden.
[9] Der Sicherheitsintegritäts-Level ist eine diskrete Stufe, die die maximal zulässige Ausfallswahrscheinlichkeit einer Sicherheitsfunktion eines Systems festlegt [36].
[10] Die ISO 26262 [63] ist die domänenspezifische Ausprägung der DIN EN 61508 [36] für die Automobilindustrie.

Anforderungen. Entsprechend des ASIL's für das System müssen die Vorgaben der ISO 26262 [63] bei der Entwicklung berücksichtigt werden.

Die Durchführung des Requirements Development wird in der Fachliteratur [57, 119, 154] ausführlich diskutiert. Durch die Bereitstellung von Vorlagen (engl.: Templates) für Anforderungsspezifikationen [84, 117, 136, 150] soll deren Erstellung unterstützt werden. Zudem soll durch die Vorlagen die Erfüllung von Qualitätskriterien für Spezifikationen [110, 119, 136] erleichtert werden[11].

2.2.2 Architekturdesign

Ziel des Engineeringprozesses *Architekturdesign* ist es, ausgehend von funktionalen und nicht-funktionalen Anforderungen eine Systemarchitektur festzulegen. Hierzu werden die folgenden Aktivitäten durchgeführt:

- *Bewertungskriterien ermitteln:* Ausgehend von den nicht-funktionalen Systemanforderungen müssen Kriterien für die Bewertung der Architekturalternativen abgeleitet werden.
- *Architekturalternativen herleiten:* Es müssen verschiedene Alternativen für die Systemarchitektur erarbeitet werden, welche die Erfüllung funktionaler und nicht-funktionaler Anforderungen ermöglichen.
- *Architekturentscheidung treffen:* An Hand der Bewertungskriterien muss eine Entscheidung getroffen werden, welche Architekturalternative umgesetzt wird.
- *Systemarchitektur dokumentieren:* Die Struktur des Systems, welche die Zerlegung des Systems in Teilsysteme sowie das Zusammenspiel der Teilsysteme darstellt, muss dokumentiert werden. Dabei müssen zu den Teilsystemen der Teilsystemtyp (SW, Mechanik, Elektrik/Elektronik), die Aufgabe des Teilsystems und dessen Schnittstellen festgehalten werden. Ebenso muss das dynamische Zusammenspiel der Teilsysteme dokumentiert werden.

Um die Schritte des Architekturdesign einfach nachvollziehen zu können, ist es sinnvoll, Methoden des Entscheidungsmanagements [34, 70] anzuwenden.

Bei der Entwicklung sicherheitsrelevanter Systeme müssen im Architekturdesign insbesondere folgende Aktivitäten durchgeführt werden:

- *Sicherheitsarchitektur erstellen:* Es muss eine *Sicherheitsarchitektur* erstellt werden, durch die gewährleistet wird, dass die Ausfallswahrscheinlichkeit der Sicherheitsfunktionen geringer ist als die dem ASIL der Anforderung entsprechende maximal zulässige Ausfallswahrscheinlichkeit.
- *Sicherheitsanalysen durchführen:* An Hand von Sicherheitsanalysen wie *Fehlermöglichkeits- und Einflussanalyse (FMEA, engl.: Failure Modes and Effects Analysis)* [99, 146, 151] oder *Fehlerbaumanalyse (engl.: Fault Tree Analysis, FTA)* [61, 152] muss bewertet werden, ob die Systemarchitektur sowie die zugehörige Anforderungsdekomposition geeignet sind, potentielle Fehler zu beherrschen. Zudem muss der quantitative Nachweis erbracht werden, dass für die Sicherheitsfunktionen die jeweiligen Anforderungen an die *Sicherheitsintegrität* erfüllt werden. Dabei bezeichnet die Sicherheitsintegrität die Wahrscheinlichkeit, mit der ein System die Sicherheitsfunktionen unter allen festgelegten Bedingungen innerhalb eines festgelegten Zeitraums anforderungsgemäß ausführt [36].

[11] Durch die Bereitstellung einer klar strukturierten Vorlage soll beispielsweise vermieden werden, dass Anforderungen unvollständig, mehrfach oder widersprüchlich spezifiziert werden.