



# 1 Upgrades von Unternehmenssoftware

Anwenderunternehmen steht in regelmäßigen Abständen mit jeder neuen Version der betriebswirtschaftlichen Standardanwendungssoftware eine Vielzahl neuer Funktionen zur Verfügung, die den Unternehmen einen betriebswirtschaftlichen Mehrwert liefern können. In der Praxis beschäftigen sich die Unternehmen allerdings wenig mit diesen neuen Funktionen, da es sie zu viel Zeit kostet, alle zu bewerten und die für sie Nutzen stiftenden zu identifizieren. Oftmals werden die Unternehmen bereits von den Kosten und der Komplexität der technischen Seite von Upgrades abgeschreckt. Die Konsequenz ist somit eine geringe Upgrade- und Innovations-Bereitschaft in Anwenderunternehmen und eine relativ geringe Nutzung neuer Funktionen. Im Lebenszyklus einer Unternehmenssoftware wird in der Forschung vor allem die Seite der Implementierung betrachtet, (siehe Kapitel 2.3). Vernachlässigt wird dabei der Bereich des betriebswirtschaftlichen Upgrades, der auch enormes Innovationspotenzial beinhaltet und im Vergleich zur Implementierung den wesentlich größeren Teil des Softwarelebenszyklus ausmacht.

Das Ziel dieser Arbeit besteht darin, individuell für ein Anwenderunternehmen, Innovationen aus Upgrades zu identifizieren. Grundlage hierfür ist eine automatisierte Analyse und Bewertung der neuen Funktionen. Entwickelt wurde daher eine Strategie, das Anwenden hilft, die folgenden Fragen zu beantworten:

- Welche neuen Funktionen sind relevant im Kontext der aktuellen Nutzung?
- Welchen Nutzen kann man aus der Verwendung der neuen Funktionen ableiten?
- Welche Wirkungen haben die neuen Funktionen organisatorisch und projektbezogen für das Anwenderunternehmen?
- Welche neuen Funktionen sollten zuerst eingeführt werden, weil sie beispielsweise einen „Quick Win“ erlauben?

In diesem Kapitel werden zuerst die neuen Funktionen spezifiziert, die Gegenstand der Betrachtung sind. Anschließend wird die Problematik der Anwenderunternehmen in der Praxis noch einmal im Detail beschrieben und aufgezeigt, dass sich die Forschung mit dieser Problematik noch nicht auseinandergesetzt hat. Es folgt daraus eine Ableitung der Forschungsfragen. Abschließend wird vorgestellt, wie die Lösung des Problems ermöglicht wird.

## 1.1 Upgrade-Innovationen

Im Bereich Unternehmenssoftware wird unter einem Upgrade verstanden, dass ein existierendes System auf ein neues Release aktualisiert wird [FAUS07, S. 418].



Unter dem Begriff „Release“ werden die vom Hersteller den Anwenderunternehmen bereitgestellten neuen Funktionen verstanden. Die Neuerungen resultieren aus Änderungen und Neuentwicklungen auf technologischem, betriebswirtschaftlichem und rechtlichem Gebiet. Die Anwender von Unternehmenssoftware haben somit durch Upgrades den Vorteil, dass ihre Software betriebswirtschaftlich immer auf dem neusten Stand ist [NOTT97, S. 344f.].

Oftmals werden die Begriffe Upgrade und Release-Wechsel synonym verwendet. Sie sind dadurch charakterisiert, dass vom Hersteller betriebswirtschaftlicher Standardanwendungssoftware eine neue Version der Software bereitgestellt wird.

Ein Upgrade wird unterschieden in

- technisches („technical“) und
- betriebswirtschaftliches („functional“) Upgrade, z. B. bei [KÖSE09, S. 795] [DEMP11, S. 68] [HENN09, S. 233].

Unter technischem Upgrade wird der Prozess verstanden, die Anwendung durch eine überlegene, technische Version zu ersetzen, ohne die Funktionalität zu ändern. Dem steht das betriebswirtschaftliche Upgrade gegenüber. Dieses wird von DEMPSEY und LIAM wie folgt definiert: „A functional upgrade has greater impact as it offers key business improvements and enhanced functionality“ [DEMP11, S. 68]. Nachdem Upgrades technisch implementiert wurden, stehen der Softwarebibliothek neu entwickelte Funktionen zur Verfügung.

In der vorliegenden Arbeit wird zur besseren Abgrenzbarkeit für die in Releases bereitgestellten funktionalen Neuerungen der Begriff der Upgrade-Innovation eingeführt. Upgrade-Innovationen sind durch folgende Eigenschaften gekennzeichnet:

- Sie sind funktionale Neuerungen (nicht ausschließlich technisch)
- und werden über das Upgrade einer betriebswirtschaftlichen Standardanwendungssoftware bereitgestellt.

Der von der SAP geprägte Begriff der Release-Neuerung wurde nicht verwendet. Release-Neuerungen sind sehr fein-granular angelegt und deshalb in ihrer Gesamtheit sehr umfangreich. Die Übersichtlichkeit der Bewertung würde unter dieser Masse an zu betrachtenden Einzelementen leiden. Eine Upgrade-Innovation umfasst mehrere Release-Neuerungen, die jeweils funktional zusammenhängen, z. B. Erweiterungen einer Kampagne sind. Hinzu kommt, dass der Begriff der Release-Neuerung auch technische Neuerungen umfasst. Reine Customizing-Neuerungen oder Fehlerkorrekturen sind beispielsweise bei Release-Neuerungen, aber nicht im Begriff Upgrade-Innovation enthalten. Details zur Abgrenzung



des Begriffs Upgrade finden sich in Kapitel 3.1.3, im Hinblick auf den Begriff der Innovation in Kapitel 5.1.1.

## 1.2 Defizite in der Praxis

Für Unternehmen gehören die in regelmäßigen Abständen bereitgestellten Upgrades zum Softwarelebenszyklus. Die Frage ist nicht ob, sondern wann ein Upgrade durchgeführt wird [KREM00, S. 55].

Informationsdefizite bezüglich der Release-Neuerungen im Allgemeinen sowie die impliziten Kosten sind Argumente, die Upgrades hinauszögern [HUGF04, S. 23]. Die größten Kostenblöcke eines Upgrades lassen sich folgendermaßen benennen [KÖSE09, S. 794] [SAPL02, S. 759]:

- Zeit und Ressourcen für die Durchführung des Upgrades
- Beeinträchtigung des Betriebs durch Ausfallzeit des Produktivsystems oder Schulung der Mitarbeiter
- Zeitaufwand für Tests oder Fehlerbehebung

Die Laufzeiten von technischen Upgrades sind schwer vorherzusagen und werden beispielsweise von Hardware, Umfang von Eigenentwicklungen oder noch benötigten Support Packages beeinflusst [FAUS07, S. 420]. Je mehr Modifikationen in Form von Ergänzungsprogrammierungen ein Unternehmen durchgeführt hat, umso mehr steigen die Komplexität und der Aufwand für ein Upgrade. Wenn Upgrades nicht mehr oder nur schleppend vorgenommen werden, geht einer der großen Vorteile von betrieblicher Standardanwendungssoftware verloren, da sich das Unternehmen vom „technologischen und betriebswirtschaftlichen Fortschritt“ abkoppelt [NOTT97, S. 344f.].

Durch eine frühzeitige Durchführung eines Upgrades sowie die Nutzung der funktionalen Neuerungen können Unternehmen auch den oft genannten Nachteil von Standardanwendungssoftware, nämlich der nicht möglichen Differenzierung von der Konkurrenz, durch Schnelligkeit wettmachen [KREM00, S. 55].

In der Projektrealität beschäftigen sich die Unternehmen allerdings wenig mit betriebswirtschaftlichen Innovationspotenzialen, da sie das technische Upgrade zu viel Zeit kostet, siehe hierzu die Ergebnisse aus Kapitel 3. Auch fehlen oft das Wissen und die Zeit im Unternehmen, alle neuen Funktionen zu bewerten und den individuellen Nutzen zu identifizieren. Es besteht für das Unternehmen hierbei, ähnlich wie bei der Implementierung, die betriebswirtschaftliche Herausforderung, aus der Vielzahl an neu zur Verfügung gestellten Funktionen



eines ERP-Systems, die für sie relevanten zu identifizieren. Nicht jede Neuerung bringt für Unternehmen automatisch eine Verbesserung des Geschäftsablaufs. Upgrades haben einen kleineren Umfang als eine Implementierung, treten dafür aber häufiger auf und bedürfen daher auch einer intensiveren Auseinandersetzung.

Was insgesamt fehlt, ist ein systematischer Ansatz zur Entscheidungsunterstützung, wie die für ein Unternehmen relevanten Innovationen, die ein Upgrade bereitstellt, identifiziert und bewertet werden können. Damit würde es ermöglicht, dass Unternehmen, sobald ein neues Release auf den Markt kommt, noch vor dem eigentlichen Upgrade beurteilen können, welchen Nutzen es für sie stiftet.

### 1.3 Forschungslücke

Die Entwicklung von Unternehmenssoftware oder ERP-Systemen (zur Abgrenzung dieser Begriffe siehe Kapitel 3.1.1) wird in der Literatur oft in zwei Phasen bzw. „waves“ aufgeteilt, z. B. bei [KR/EM02, o. S.] [SHAN03, S. 5]. Die Erste besteht aus der Akquisition, Konfiguration und Implementierung von Unternehmenssoftware. Die zweite Phase, auch als „post-implementation phase“ bezeichnet [BAJW04, o. S.] [BREA04, o. S.] [YU05, o. S.], fokussiert auf den Support und die Wartung sowie die kontinuierliche Verbesserung. Es geht darum, den Nutzen der IT-Investitionen zu erhöhen.

Bekannt ist, dass die zweite Phase in Bezug auf die gesamte Lebensdauer einer Unternehmenssoftware den größten Teil der Kosten ausmacht [SCHE90a, S. 126]. Trotzdem beschäftigen sich die Veröffentlichungen zu Unternehmenssoftware in erster Linie mit der Implementierung, wie verschiedene Literaturanalysen zeigen [ESTE00, o. S.] [DONG02, o. S.] [JACO03, o. S.] [NG03, o. S.] [MOON07, o. S.] [MAED12, o. S.]. Hierzu existieren viele Veröffentlichungen, beispielsweise zu strategischen Optionen, Vermeidung von Fehlern, Identifikation von Anknüpfungspunkten zum Business-IT-Alignment oder verbesserten Einführungsmethoden [ALMA03, o. S.].

Die Veröffentlichungen zur „post-implementation phase“ haben in letzter Zeit etwas zugenommen und beschäftigen sich mit den in Tabelle 1 aufgelisteten Themenschwerpunkten.

Die Veröffentlichungen der letzten Jahre beschäftigten sich zusammengefasst vor allem mit den Auswirkungen der Implementierung auf die Mitarbeiter und die Organisation. Einzelne hatten auch die Wartung bzw. den Ablauf von Upgrades zum Thema, wie beispielsweise NG et al., die ein ERP-Wartungsmodell entwickelt haben [NG03, o. S.] oder COLLINS, der ein ERP-Upgrade Phasen-Modell definiert hat [COLL99, S. 46]



Tabelle 1: Themenschwerpunkte der „post-implementation phase“

| Autoren                                                           | Kategorie                           | Themenschwerpunkte                                                                                                                       |
|-------------------------------------------------------------------|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Rikhardsson, P.;<br>Kræmmergaard, P. [RIKH06,<br>o. S.]           | Auswirkungen Im-<br>plementierung   | Allgemein                                                                                                                                |
| Morris, M. G.; Venkatesh, V.<br>[MORR10, o. S.]                   | Auswirkungen Im-<br>plementierung   | Aufgaben sowie Zufriedenheit der Mitarbeiter                                                                                             |
| Locke, J.; Lowe, A. [LOCK05,<br>o. S.],                           | Auswirkungen Im-<br>plementierung   | Vertrauen                                                                                                                                |
| Liang, H.; Saraf, N.; Hu, Q.;<br>Yajiong, X. [LIAN07, o. S.]      | Auswirkungen Im-<br>plementierung   | Belastungen und Spannungen sowie die Rolle des<br>Top Managements                                                                        |
| Volkoff, O.; Strong, D. M.;<br>Elmes, M. B. [VOLK05, o. S.]       | Auswirkungen Im-<br>plementierung   | Integration                                                                                                                              |
| Poston, R.; Grabinski, S.<br>[POST01, o. S.]                      | Auswirkungen Im-<br>plementierung   | Finanzielle Situation des Unternehmens                                                                                                   |
| Hunton, J.; Lippincott, B.;<br>Reck, J. [HUNT03, o. S.],          | Auswirkungen Im-<br>plementierung   | Leistung von Unternehmen, die eine Unterneh-<br>menssoftware implementiert haben im Vergleich zu<br>denen, die keine implementiert haben |
| Dillard, J.; Ruchala, L.;<br>Yuthas, K. [DILL05, o. S.],          | Auswirkungen Im-<br>plementierung   | Administration bzw. Legitimation von Handlungs-<br>weisen                                                                                |
| Shaw, N. [SHAW01, o. S.]                                          | Auswirkungen Im-<br>plementierung   | Mitarbeiter und die Organisation in Bezug auf Up-<br>grades                                                                              |
| Sasidharan, S.; Santhanam,<br>R.; Brass, D. J. [SASI11,<br>o. S.] | Auswirkungen Im-<br>plementierung   | Einfluss von sozialen Netzwerkstrukturen der Mit-<br>arbeiter auf den Erfolg von Unternehmenssoftware                                    |
| Ng, Celeste See Pui; Gable,<br>Guy; Chan, Taizan [NG03,<br>o. S.] | Wartung                             | ERP-Wartungsmodell                                                                                                                       |
| Zhao, Fan [ZHAO07, o. S.]                                         | Erfolgsfaktoren<br>Upgrade          | Erfolgsfaktoren in Bezug auf Mitarbeiter und Orga-<br>nisation                                                                           |
| Somers, T.; Nelson, K.<br>[SOME04, o. S.]                         | Erfolgsfaktoren<br>ERP-Lebenszyklus | Kritische Aktivitäten und Personen im ERP-<br>Projektlebenszyklus                                                                        |
| Boogaard, Peter [BOOG11,<br>o. S.]                                | Erfolgsfaktoren<br>Upgrade          | Schlüsselaspekte zur Ausführung von erfolgrei-<br>chen Upgrades                                                                          |
| Collins, K. [COLL99, o. S.]                                       | Erfolgsfaktoren<br>Upgrade          | Strategien und Ausführung von Upgrades                                                                                                   |



Zum Thema Upgrade wird zudem noch nach Faktoren gesucht, die die erfolgreiche Ausführung von Upgrades beeinflussen. ZHAO legt hier seinen Schwerpunkt auf Erfolgsfaktoren eines Upgrades in Bezug auf Mitarbeiter und Organisation [ZHAO07, o. S.], SOMERS und NELSON auf die kritischen Aktivitäten im gesamten Lebenszyklus [SOME04, o. S.] und BOOGARD auf die Aspekte, die vor einem Upgrade zu beachten sind [BOOG11, o. S.].

BOOGARD weist darauf hin, dass ein Upgrade-Projekt ökonomisch nur dann sinnvoll ist, wenn die Möglichkeit genutzt wird, Geschäftsprozesse zu reevaluieren, neue Funktionalität zu nutzen und Veränderungen der Organisationen umzusetzen. „Otherwise you are going to end up with outdated processes and the latest technology“ [BOOG11, S. 10]. Auch COLLINS stellt fest, dass Unternehmen durch das Nutzen neuer Funktionalität mit der aktuellen Geschäftsentwicklung mithalten, ihre Prozesse verbessern und effizientere Geschäftsmodelle aufbauen können [COLL99, S. 43].

Es fehlt aber ein übergreifender Ansatz, wie alle für ein Unternehmen relevanten Nutzenpotenziale aus einem Upgrade identifiziert werden können. Denn in einem neuen Release sind neue Funktionalitäten für verschiedenste Bereiche der Standardanwendungssoftware enthalten und nicht alle sind für jedes Anwenderunternehmen interessant.

Einordnungen und Details zu den Modellen zum Upgrade, z. B. von ZHAO und NG, finden sich in Kapitel 3.1.

Auch der eher praktisch orientierte Bereich (siehe Kapitel 3.2) beschäftigt sich nicht mit dieser Thematik. Hier wird vor allem das technische Upgrade für bestimmte Unternehmenssoftwareprodukte, wie die SAP Business Suite, fokussiert, z. B. welche Werkzeuge unterstützend herangezogen werden können [SCHÄ11, o. S.] [SCHÄ09, o. S.] [HENN09, o. S.] oder wie Upgrade- und Support-Packages eingespielt werden [VANS07, o. S.]. Es wird davon ausgegangen, dass bereits die Entscheidung für ein Upgrade gefallen ist und dieses jetzt noch mit möglichst geringen Belastungen des täglichen Betriebs umgesetzt werden muss. Weitere Autoren beschäftigen sich mit der Seite der Funktionalität von betriebswirtschaftlicher Standardanwendungssoftware und mit den wichtigsten Innovationen neuer Releases für einen bestimmten Bereich, zum Beispiel im Finanzwesen dem neuen Hauptbuch [ARIF08, S. 121-162] oder im Customer Relationship Management dem Customer Interaction Center [BURT09, S. 82-364]. Diese bieten durchaus für einzelne Bereiche eine gute Informationsbasis bezüglich Nutzen und Aufwand der neuen Funktionalität. Die Informationen sind jedoch jeweils immer nur punktuell und nicht gleichartig strukturiert.



## 1.4 Forschungsfragen

Es werden daher folgende Forschungsfragen formuliert:

**F1:** Worin liegen aus Sicht von Anwendungsunternehmen die Herausforderungen bei der Identifikation von Innovationspotenzialen aus Upgrades für Unternehmenssoftware?

**F2:** Welche Aspekte sollten bei der Bewertung von Innovationspotenzialen herangezogen werden?

**F3:** Welche Werkzeugart eignet sich als Realisierungsplattform für die automatisierte Unterstützung zur Bewertung von Upgrade-Innovationen?

**F4:** Wie und mit welchem Werkzeug kann eine derartige Analyse umgesetzt werden?

Um Antworten auf die ersten beiden Forschungsfragen zu bekommen, hat die Autorin Studien zum Thema Upgrade analysiert und in Experteninterviews Vertreter von Anwenderunternehmen befragt. So konnten Ursachen der Probleme und Wirkungen aufgezeigt werden. Darauf aufbauend wurde das Modell der Reverse Business Innovations (RBI) definiert. Die Konzeption basiert neben den identifizierten Herausforderungen aus den Experteninterviews auch auf Ergebnissen der Innovations- und Entscheidungstheorie. Es wurde ein regelbasiertes Expertensystem als Realisierungsplattform ausgewählt. Die Umsetzung erfolgte am Beispiel der SAP Business Suite, dem Marktführer im Bereich Unternehmenssoftware. Anschließend erfolgte eine Validierung mit einem Pilotanwenderunternehmen. Ergebnis ist, dass es möglich wurde, mittels RBI eine nachvollziehbare Identifikation von Innovationspotenzialen individuell für ein Anwenderunternehmen vorzunehmen.

Als Basis dient eine Nutzungsanalyse der produktiven Unternehmenssoftware. Die Entscheidungsunterstützung berücksichtigt also zunächst die bisherige „Reverse“-Systemnutzung und gleicht sie mit einem Innovationskatalog ab. Im Innovationskatalog sind alle Release-Neuerungen enthalten, angereichert um zusätzliche Informationen, wie Ausschlusskriterien, Relevanzfaktoren oder Nutzen.

Somit übernimmt das Werkzeug die Informationsbeschaffung und -auswertung und gibt pro Unternehmen die interessanten Upgrade-Innovationen aus. Hierfür wurde nach Kriterien gesucht, die einen Rückschluss auf die Relevanz einer Upgrade-Innovation zulassen. D. h., es wurde die Frage gestellt, unter welchen Bedingungen diese Funktionalität für ein Unternehmen interessant ist. Ähnlich wie beim ‚pattern-based approach‘ [DELF09, o. S.] bzw. Mustervergleich wurden so Kriterien formalisiert. Für jede Upgrade-Innovation wird eine



Bewertung vorgenommen und anschließend das Ergebnis entscheidungsorientiert präsentiert.

Dies hat folgende Vorteile für die Unternehmen:

- Das Ergebnis ist eine unternehmensindividuelle Entscheidungsgrundlage, die entsprechend aufbereitet wurde, sodass schnell und einfach eine Übersicht gewonnen werden kann, welche Upgrade-Innovationen generell einen Nutzen stiften und welche davon mit Priorität eingeführt werden sollten, da sie beispielsweise einen „Quick Win“ bieten.
- Die Potenziale von Upgrades einer Unternehmenssoftware können so schneller zugänglich und implementiert werden. Unternehmen profitieren damit jederzeit vom Fortschritt der Unternehmenssoftware.
- Möglicherweise können Eigenentwicklungen durch neue Funktionalität abgelöst werden. Je mehr Eigenentwicklungen vorhanden sind, umso mehr Aufwand verursachen diese im Unternehmen. Jede Eigenentwicklung, die abgelöst werden kann, bedeutet also bei künftigen Release-Wechsels oder im Support weniger Aufwand.
- Wartungsgebühren können sinken, wenn Unternehmen dazu bewegt werden, ein Upgrade in ihrem System durchzuführen. Die meisten Hersteller verlangen nach Ablauf der Standardwartung Aufschläge auf die Wartungsgebühren bei veralteten Release-Ständen, siehe hierzu Kapitel 0.

Zusammengefasst hilft das Modell Unternehmen, die Nachhaltigkeit der häufig nicht unerheblichen IT-Investitionen sicherzustellen.

## 1.5 Aufbau der Arbeit

Das Forschungsvorhaben begann mit der Problemerkennung. Es wurde die automatisierte Bewertung von neuen Funktionalitäten, die über ein Upgrade zur Verfügung gestellt werden, für die individuelle Unternehmenssituation als Problem und der systemunterstützte Anforderungsabgleich bei Upgrades als Forschungslücke erkannt. Es wurde die bestehende Literatur zu Upgrades geprüft, um Hinweise zur möglichen Lösung zu erhalten und die Innovationen einordnen zu können (Kapitel 1). Aktuell beschäftigt sich die Forschung in erster Linie mit der Implementierung und nur wenige Autoren mit der Phase nach der Implementierung. In dieser Phase nach der Implementierung liegt der Schwerpunkt auf einer Analyse der Auswirkungen der Implementierung auf die Mitarbeiter. Zum Upgrade selbst finden sich nur sehr wenige Veröffentlichungen und sehr wenig Theorie. Es wurde der Begriff der Upgrade-Innovation definiert, um den Bewertungsgegenstand fassbar zu machen, da es bisher



noch keine Bezeichnung für funktionale Neuerungen, die über ein Upgrade bereitgestellt werden, gibt.

Kapitel 2 zeigt detaillierter die theoretischen Modelle auf, in denen das Thema Wartung bzw. Upgrade aufgegriffen wird. Ebenso wird die Upgrade-Strategie des Marktführers im Bereich Unternehmenssoftware vorgestellt und darauf aufbauend die Relevanz des Themas in Theorie und Praxis beurteilt.

Da zu diesem Thema bisher sehr wenige Veröffentlichungen existieren, wurden Experteninterviews mit Vertretern aus der Praxis durchgeführt (Kapitel 3). Als Experten wurden Mitarbeiter von Unternehmen ausgewählt, die in der Vergangenheit Upgrade-Projekte in ihrem Unternehmen verantwortet haben. Diese bestätigten die Bewertung von Upgrade-Innovationen als Defizit und nannten Anforderungen an eine Entscheidungsunterstützung.

Die Konzeption zur Lösung des Problems wird in Kapitel 4 vorgestellt. Für die Upgrade-Innovationen werden zum einen aus den Experteninterviews, zum anderen aus der Entscheidungs- und Innovationsforschung Bewertungskriterien abgeleitet. Anschließend wird bewertet, ob durch die Konzeption auch die definierten Anforderungen abgedeckt werden.

In Kapitel 5 wird geprüft, welche Art von Plattform sich zur Realisierung eignet. Benötigt wird ein System, das strukturiert Informationen speichert, eine Analyse individuell für die Unternehmen durchführt und eine Schlussfolgerung für die Upgrade-Innovationen selbstständig ziehen kann.

In Kapitel 6 wird dann basierend auf den Anforderungen ein Produkt ausgewählt und das Modell in diesem umgesetzt. Hierzu wird vorgestellt, wie bei der Entwicklung eines Prototyps am Beispiel von SAP CRM der SAP AG vorgegangen wurde. Prinzipiell ist das Modell aber auch auf Unternehmenssoftware anderer Hersteller übertragbar.

Es folgt die Validierung des Modells anhand von Pilotanwenderunternehmen (Kapitel 7). Bei diesen wurde im SAP CRM-System eine Nutzungsanalyse ausgeführt und darauf aufbauend die Upgrade-Innovationen empfohlen. Es werden Ergebnisse vorgestellt und bewertet. Dabei wird auch auf Grenzen eingegangen.

Die Arbeit endet mit einem Ausblick in Kapitel 8, in dem die Bedeutung von regelbasierten Systemen in der Vergangenheit, heute und in der Zukunft beurteilt wird. Hierfür wird auf bestehende regelbasierte Systeme verwiesen, es werden Erweiterungsmöglichkeiten des Modells vorgestellt sowie weitere zukünftige Anwendungsfelder identifiziert, die sich bedingt durch neue technologische Entwicklungen ergeben.



---

Im letzten Kapitel werden die zentralen Ergebnisse dieser Arbeit noch einmal zusammengefasst.



## 2 Upgrade von Unternehmenssoftware

In diesem Kapitel werden die Ergebnisse der Literaturanalyse zum Thema Upgrade von Unternehmenssoftware vorgestellt. Hierzu wird eine umfassende Betrachtung der theoretischen Modelle und der praktischen Seite am Beispiel der am weitesten verbreiteten Unternehmenssoftware, der SAP Business Suite durchgeführt.

### 2.1 Theoretische Modelle

In diesem Kapitel werden der Begriff Upgrade abgegrenzt, Modelle aufgezeigt, die Bedeutung des Upgrades für verschiedene Softwarekategorien diskutiert und erläutert, warum Standardsoftware ein Innovationstreiber ist.

#### 2.1.1 Softwarekategorien

Software wird nach der Nähe zur Anwendung in Systemsoftware und Anwendungssoftware unterschieden, siehe beispielsweise [MERT10, S. 17] [THOM96, S. 34] [ZAHN01, S. 393] [SCHE90a, S. 10]. Systemsoftware bildet zusammen mit der Hardware die Voraussetzung für den Betrieb der Anwendungssoftware [SCHE90a, S. 11]. Anwendungssoftware bezieht sich auf ein konkretes betriebswirtschaftliches Anwendungsgebiet. Anwendungssysteme werden für die primären Prozesse der Wertschöpfungskette wie Vertrieb, Beschaffung und Produktion, aber auch für die unterstützenden Prozesse, wie Personalwesen, Finanzbuchhaltung und Controlling eingesetzt [STAH05, S. 326].

Abbildung 1 zeigt die Softwarekategorien nach SCHEER und Abbildung 2 nach THOME. Bei beiden sind die zentralen Kriterien zur Einteilung in Softwarekategorien die Unterscheidung nach System- und Anwendungssoftware sowie Standard- und Individualentwicklungen.

Als Standardsoftware werden Softwarepakete bezeichnet, die in vielen Organisationen ohne Veränderungen, nur mittels Parametrierung, einsetzbar sind. Parametrierung bedeutet, dass die Software an die spezifischen Bedürfnisse der Organisation angepasst werden kann, der Quellcode dabei aber nicht verändert wird [ÖSTE97, S. 379]. Den Gegensatz zur Standardsoftware bildet die Individualsoftware, die extra für ein einzelnes Unternehmen entwickelt wird [SCHE90a, S. 68].

Zu Beginn der 70er Jahre begann neben der Entwicklung von Individualsoftware auch der Handel mit ablauffähigen, fertigen Programmen [SCHM90, S. 170]. In den 70er und 80er Jahren entschieden sich vor allem Kleinbetriebe für Standardsoftware, weil sie sich Individualentwicklungen nicht leisten konnten [ÖSTE97, S. 379].

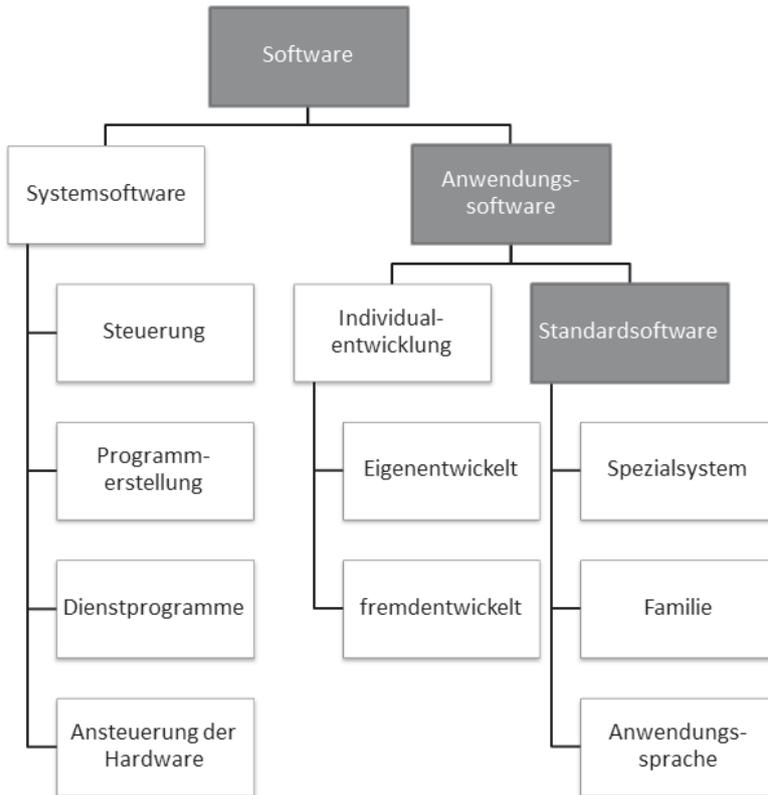


Abbildung 1: Klassifikation von Software nach Scheer in Anlehnung an [SCHE90a, S. 10 u. S. 139]

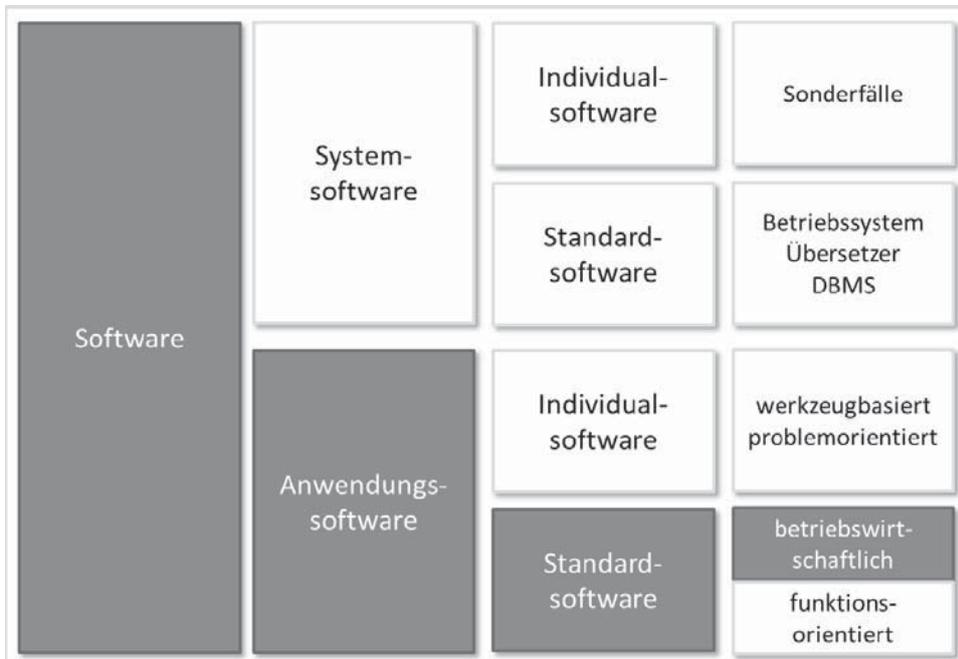


Abbildung 2: Softwarekategorien [THOM96, S. 34]

Mitte der 80er Jahre hat sich bei den Anbietern von Standardanwendungssoftware eine starke Erweiterung des funktionalen Umfangs und der Integration der Arbeitsgebiete vollzogen



[RAUB90, S. 60]. Dies wurde zu einem der wichtigsten Argumente für Standardsoftware [ÖSTE90a, S. 24]. Standardsoftware begann schließlich von Unternehmen aller Größe als strategische Option wahrgenommen zu werden [ÖSTE90b, S. 9].

Ob Standard- oder Individualsoftware die bessere Lösung für Unternehmen bildet, wurde lange Zeit diskutiert. Zu den Befürwortern von Standardanwendungssoftware gehören beispielsweise ÖSTERLE [ÖSTE90a, S. 26], SCHMID [SCHM90, S. 171], SCHEER et al. [SCHE90b, S. 97] oder THOME und HUGFARD [THOM96, S. 15].

Individualsoftware wird noch für Bereiche benötigt, in denen es keine Standardsoftware gibt und sich die Individualentwicklung somit auf besonders wichtige Problemstellungen oder Sonderwünsche konzentrieren kann [HUGF94, S. 35].

Die wichtigsten **Vorteile beim Kauf von Standardsoftware** sind folgende [STAH05, S. 296f.] [HANS09, S. 666] [ÖSTE90a, S. 25-27] [FÜGL90, S. 155]:

- In der Regel kostengünstiger bei Entwicklung, Einführung und Wartung als Individualsoftware
- Sofortige Verfügbarkeit
- Bessere Qualität der Software aufgrund von höherer Erfahrung der Programmierer und intensiverer Tests
- Herausbildung von Marktstandards
- Höhere Sicherheit (kein Datenverlust, kein unerlaubter Zugriff nicht autorisierter Personen etc.)
- Bessere Dokumentation
- Hersteller entwickelt die Software ständig weiter hinsichtlich Betriebs-, Datenbank- oder Monitorsystemen

Anfangs waren betriebswirtschaftliche Standardanwendungssysteme sehr inflexibel, wodurch sie zu hemmenden Faktoren in Bezug auf Innovationen wurden. Inzwischen kann Standardanwendungssoftware aber als Innovationsförderer bezeichnet werden, da die Hersteller immer bestrebt sind, ihre Lösungen auf den aktuellen Stand der Softwaretechnologie zu bringen und diese somit auch von Unternehmen schneller genutzt werden können. Mit der Einführung von Standardsoftware holen sich die Unternehmen eine „State-of-the-Art-Plattform“, mit der sie technologisch, betriebswirtschaftlich, organisatorisch, logistisch und im Hinblick auf die Produktfunktionalität aufschließen. Standardsoftware fungiert nur noch selten als Innovationsbremse, wenn die Systeme in ihrer Konzeption und ihrem Kern auf



veralteten Technologien basieren oder wenn sich Branchen im Umbruch befinden [HUGF94, S. 41] [ADLE90, S. 163, 164 u. 170].

Ein Modell, wie es in dieser Arbeit für das betriebswirtschaftliche Upgrade vorgestellt wird, wäre undenkbar für Individualentwicklungen. Denn bei Individuallösungen werden Erweiterungen nur auf expliziten Wunsch des Unternehmens entwickelt. „Upgrades“ liefern bei diesen also nur fehlende Funktionalität, die von dem einen Anwenderunternehmen explizit spezifiziert wurde. Bei Standardanwendungssoftware hingegen werden neue Funktionen ständig entwickelt, beispielsweise aufgrund von Wünschen verschiedener Kunden, neuen Technologien, neuen Standards etc. Der Vielzahl von Anwenderunternehmen der betriebswirtschaftlichen Standardanwendungssoftware stehen somit in regelmäßigen Abständen neue Funktionen zur Verfügung, ohne dass diese explizit nachgefragt wurden.

Bei Betriebssystemen, wie beispielsweise Windows, werden nach deren Installation in regelmäßigen Abständen Fehlerkorrekturen bereitgestellt. Um umfangreiche neue Funktionalität nutzen zu können, muss die aktuelle Version des Betriebssystems deinstalliert und eine Nachfolgerversion neu installiert werden. Bei dieser Softwarekategorie ist aber auch die Komplexität einer Implementierung überschaubar, da für Mitarbeiter eines Unternehmens zum einen das Betriebssystem jeweils ähnlich ausgeprägt ist und zum anderen der Zugriff auf weitere Softwareprogramme über Server im Unternehmen ausgeführt wird und das Betriebssystem damit als vorgelagert und austauschbar betrachtet werden kann. Mit einer Neuinstallation verbundene Ausfallzeiten betreffen daher nur einzelne Arbeitsplätze. Hinzu kommt, dass eine Neuinstallation einer Systemsoftware weniger Zeit beansprucht, als die Neuinstallation einer betriebswirtschaftlichen Standardanwendungssoftware und somit auf Urlaube der betroffenen Mitarbeiter, Wochenenden o. ä. gelegt werden kann.

Bei der funktionsorientierten Standardsoftware sind auch Neuinstallationen die gängige Praxis. Dies wird möglich, da der betroffene Bereich und die betroffenen Mitarbeiter abgegrenzt werden können. Zudem werden die Daten, beispielsweise bei MS Office-Produkten, außerhalb der Softwarelösung gespeichert, sodass diese anschließend einfach mit einer neuen Softwareversion geöffnet werden können. Da derartige Dateien normalerweise auch in höheren Softwareversionen verwendbar sind, können übergangsweise alte und neue Versionen parallel im Unternehmen eingesetzt werden, ohne die Mitarbeiter in ihrer täglichen Arbeit einzuschränken.

Den größten Aufwand bei einer Neuinstallation weist der Bereich der betriebswirtschaftlichen Standardsoftware auf. Hier sind zum einen verschiedenste, je nach Verbreitungsgrad



unter Umständen auch alle Unternehmensbereiche und Mitarbeiter betroffen. Die Mitarbeiter können nicht alle die gleichen Schulungen besuchen, da sie mit verschiedenen Modulen und folglich nicht mit den gleichen Benutzungsoberflächen und inhaltlich mit ganz anderen Funktionen arbeiten. Ein weiterer Treiber der Komplexität sind oftmals in Eigenregie durchgeführte Programmänderungen des Unternehmens (siehe hierzu Kapitel 3.1.3). Bei betriebswirtschaftlicher Standardanwendungssoftware wird von einem auf den anderen Tag die komplette neue Lösung für das ganze Unternehmen umgestellt, wobei ein konsistenter Übergang sichergestellt werden muss.

In der Softwarekategorie der betriebswirtschaftlichen Standardanwendungssoftware, in die beispielsweise die Unternehmenssoftware SAP Business Suite fällt, sind Upgrades im Vergleich zu einer Neuinstallation somit eine große Erleichterung. Eine Ablösung durch eine neue Softwarelösung würde noch viel mehr Zeit und Ressourcen kosten. Zum einen muss die Datenqualität gewahrt werden, zum anderen haben die zu migrierenden Daten einen sehr großen Umfang. Hinzu kommt, dass der Betrieb ohne diese Software oftmals stillsteht und Ausfallzeiten daher absolut zu vermeiden sind.

Durch die kontinuierliche Weiterentwicklung der Unternehmenssoftware und das zur Verfügung Stellen dieser neuen Funktionen mittels Releases bzw. Upgrades haben die Hersteller von Unternehmenssoftware eine größere Investitionssicherheit für die Unternehmen geschaffen. In der Praxis ist daher die Einführung einer neuen Unternehmenssoftware vor allem für große Unternehmen, die eine dieser Unternehmenssoftwaresysteme, wie die SAP Business Suite, bereits implementiert haben, keine Option.

Das Konzept des kontinuierlichen Upgrades anstelle wiederholter Neuinstallationen ist für betriebswirtschaftliche Standardanwendungssoftware enorm wichtig. Diese Softwarekategorie wird im Folgenden näher betrachtet.

### **2.1.2 Unternehmenssoftware**

Die ersten betrieblichen Informationssysteme gab es in den 1960er und 1970er Jahren. Sie zeichneten sich dadurch aus, dass einzelne Funktionen, wie beispielsweise die Lagerbestandsführung, unterstützt werden konnten. In den darauffolgenden Jahren ging der Trend dahin, dass immer mehr Funktionen und Prozesse abgebildet wurden und die Software nicht mehr individuell programmiert wurde [HANS09, S. 660f.].

Enterprise Resource Planning (ERP)-Systeme sind integrierte Informationssysteme mit dem Anspruch, die gesamten Unternehmensprozesse abzudecken. Derartige Systeme sind mitt-