



Technische Universität München

**Requirements Engineering für änderungsintensive,
marktgetriebene Softwareentwicklungsprojekte von kleinen
und mittleren Unternehmen**

Martin Fritzsche

Technische Universität München

Lehrstuhl für Software & Systems Engineering

Requirements Engineering für änderungsintensive, marktgetriebene Softwareentwicklungsprojekte von kleinen und mittleren Unternehmen

Martin Fritzsche

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. Uwe Baumgarten

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Dr. h.c. Manfred Broy
2. Univ.-Prof. Dr. Helmut Krcmar

Die Dissertation wurde am 11.10.2010 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 27.02.2011 angenommen.

Kurzzusammenfassung

Durch instabile Anforderungen erhöht sich das Risiko des Scheiterns von Softwareentwicklungsprojekten beträchtlich. Ein an Anforderungsinstabilität angepasstes Vorgehen im Requirements Engineering kann dieses Risiko verringern.

In dieser Arbeit untersuchen wir, wie das Requirements Engineering in marktgetriebenen Softwareentwicklungsprojekten von kleinen und mittleren Unternehmen, die einer hohen Anforderungsinstabilität ausgesetzt sind, geeignet gestaltet werden kann.

Wir stellen einen Katalog von Anforderungen an das Requirements Engineering in änderungsintensiven Softwareentwicklungsprojekten auf. Darin sind zum einen Anforderungen allgemeiner Natur enthalten, die in jedem Softwareentwicklungsprojekt berücksichtigt werden müssen. Darüber hinaus identifizieren wir die Hauptursachen für die Änderung von Anforderungen sowie Schwierigkeiten, die bei der Umsetzung von Anforderungsänderungen berücksichtigt werden müssen. Daraus leiten wir für änderungsintensive Softwareentwicklungsprojekte spezifische Anforderungen an das Requirements Engineering ab.

Ein zweiter Beitrag der Arbeit ist ein System Dynamics Modell, mit dem die Auswirkungen der Aufwandsverteilung im Requirements Engineering auf Änderungsrate, Projektkosten und Projektlaufzeit untersucht werden können. Das Modell wird im Zuge von zwei Fallstudien entwickelt und eingesetzt. Es wird herangezogen, um Vorgehensweisen im Requirements Engineering abzuleiten, die einen hinsichtlich Projektlaufzeit und Projektkosten geeigneten Umgang mit instabilen Anforderungen erlauben. Dabei wird geschlussfolgert, dass es sinnvoll ist zwischen stabilen und instabilen Anforderungen zu unterscheiden. Instabile Anforderungen sollen zudem dahingehend bewertet werden, wie gut, beziehungsweise einfach sie stabilisiert werden können. Die abgeleiteten Vorgehensweisen treffen Aussagen dazu, wie der Aufwand im Requirements Engineering auf die verschiedenen Anforderungstypen verteilt und wie er bei unterschiedlichen Instabilitäts- und Stabilisierbarkeitsgraden der Anforderungen angepasst werden sollte.

Schließlich stellen wir einen Requirements Engineering Prozess auf, der für änderungsintensive, marktgetriebene Softwareentwicklungsprojekte von kleinen und mittleren Unternehmen ausgelegt ist. Dieser berücksichtigt die zuvor aufgestellten Anforderungen an das Requirements Engineering in Softwareentwicklungsprojekten mit hoher Anforderungsinstabilität und integriert die mittels des System Dynamics Modells abgeleiteten Vorgehensweisen für den Umgang mit instabilen Anforderungen.

Abstract

Unstable requirements significantly increase the risk of failure of software development projects. An approach that is defined such that it accounts for requirements instability can reduce this risk.

In this thesis we investigate, how requirements engineering can be adequately organized in market driven software development projects of small and medium enterprises that face high requirements instability.

We develop a catalogue that comprises requirements on requirements engineering in change intensive software development projects. The catalogue includes general requirements, which have to be considered in every software development project. In addition we identify major causes for requirements changes and difficulties that have to be regarded during the realization of requirements changes. From these items we derive specific requirements on requirements engineering in change intensive software development projects.

A second contribution of this thesis is a system dynamics model which allows the investigation of how the effort allocation in requirements engineering influences change rate, project costs and project duration. The model is developed and applied in two case studies. Using the system dynamics model we derive approaches for requirements engineering that allow for an adequate handling of unstable requirements. We reason that it is sensible to differentiate between stable and unstable requirements. Unstable requirements should also be analyzed in regard of how well or easily they can be stabilized. The derived approaches state, how the effort for requirements engineering should be allocated to the different types of requirements and how it should be adjusted in respect to different degrees of instability and stabilizability of requirements.

Finally we develop a requirements engineering process that is fit for change intensive, market driven software development projects of small and medium enterprises. The process considers the previously compiled requirements on requirements engineering in software development projects with high requirements instability and integrates the approaches for dealing with unstable requirements that were derived using the system dynamics model.

Danksagung

Bei Prof. Manfred Broy bedanke ich mich herzlich für seine Anleitung und Unterstützung während der letzten Jahre und vor allem für die Möglichkeit mit großer Eigenverantwortlichkeit in mehreren Forschungsprojekten und Lehrveranstaltungen mitzuwirken.

Bei Prof. Helmut Krcmar möchte ich mich dafür bedanken, dass er zu meiner Arbeit das Zweitgutachten übernommen hat und mir in der Abschlussphase wertvolles Feedback gegeben hat.

Beiden Professoren danke ich für die kritische und detaillierte Durchsicht der Vorversionen meiner Arbeit. Ihre Kommentare haben mir sehr geholfen.

Besonderen Dank richte ich an Dr. Joachim Berlak und Thomas Reinhardt von der FAUSER AG sowie Jörg-Uwe Zuchold von der PSIPENTA Software Systems GmbH, deren Mithilfe mir in meiner Arbeit enorm geholfen hat. Auch Dr. Günther Boshold, Thomas Stark und Wolfgang Siebert von der Nokia Siemens Network GmbH & Co. KG möchte ich an dieser Stelle für ihre Unterstützung danken.

Ich danke allen Kollegen, mit denen ich zusammengearbeitet habe, von denen ich vieles gelernt habe und die mir die Arbeitszeit zu einer sehr schönen Zeit haben werden lassen, insbesondere Martin Feilkas, Stefan Wagner, Daniel Ratiu, Florian Deußenböck und Patrick Keil.

Ganz herzlicher Dank gilt meinen Freunden und meiner Familie, die mir viel Kraft, Unterstützung und Ermutigung gegeben haben. Ich möchte an dieser Stelle besonders Annemarie Fritzsche, Stephanie Rätzke und Ursula Leiter danken, die für mich während der Erstellung dieser Arbeit sehr große Stützen waren.

Gracias también a la rana.

Inhaltsverzeichnis

Abkürzungsverzeichnis	xii
1 Einführung	1
2 Katalog von Anforderungen an das Requirements Engineering in änderungsintensiven Projekten	5
2.1 Zielsetzungen des Requirements Engineering	5
2.1.1 Klärung des Begriffs „Zielsetzung des Requirements Engineering“	5
2.1.2 Gängiges Vorgehen im Requirements Engineering	6
2.1.3 Ableitung der Zielsetzungen des Requirements Engineering aus dem gängigen Vorgehen	14
2.2 Anforderungen an das Requirements Engineering aufgrund der Anforderungsinstabilität	19
2.2.1 Gründe für Änderungen der Anforderungen	19
2.2.2 Schwierigkeiten aufgrund von Änderungen der Anforderungen	24
2.3 Zusammenfassung	26
3 Ein System Dynamics Modell für die Analyse von RE-Vorgehensweisen zur Eingrenzung von Anforderungsinstabilität	28
3.1 Simulation im Software Engineering	28
3.2 Systems Dynamics	30
3.2.1 Einführung in System Dynamics	30
3.2.2 Vorgehen in der Entwicklung von System Dynamics Modellen	31
3.3 Verwandte Arbeiten	34
3.3.1 Abdel-Hamid und Madnick	34
3.3.2 Lin et al.	35
3.3.3 Powell et al.	35
3.3.4 Ferreira et al. und Houston et al.	36
3.3.5 Pfahl und Lebsanft	36
3.4 Fallstudie 1: FAUSER AG	36
3.4.1 Entwicklung des System Dynamics Modells	37
3.4.2 Durchführung von Experimenten	61
3.4.3 Zusammenfassung der Ergebnisse	75
3.5 Fallstudie 2: PSIPENTA Software Systems GmbH	77
3.5.1 Entwicklung des System Dynamics Modells	77
3.5.2 Durchführung von Experimenten	91
3.5.3 Zusammenfassung der Ergebnisse	103

4	REPMUS – Requirements Engineering Process for Market-Driven Projects Facing Unstable Requirements in Small and Medium Enterprises.....	106
4.1	Motivation und Beitrag.....	106
4.2	Kurzbeschreibung des RE-Ansatzes von Wiegers	107
4.3	Einordnung in den Entwicklungsprozess.....	110
4.4	Generelle Strategie.....	110
4.5	Anforderungscharakterisierung	114
4.6	Workflows	116
4.7	Aktivitäten	119
4.7.1	Anforderungen ermitteln	119
4.7.2	Anforderungen analysieren.....	126
4.7.3	Anforderungen spezifizieren	138
4.7.4	Anforderungen validieren.....	139
4.7.5	Anforderungen managen	142
4.8	Produkte.....	143
4.8.1	Liste der Anforderungen.....	143
4.8.2	Liste der Stakeholder	145
4.8.3	Liste der Stakeholderklassen	145
4.8.4	Template für die Liste der Stakeholderklassen.....	145
4.8.5	Template für die Anforderungsermittlung.....	147
4.8.6	Konflikte zwischen Anforderungen.....	150
4.8.7	Anforderungsspezifikation	150
4.8.8	Bericht der Anforderungvalidierung	150
4.8.9	Domänenmodelle.....	150
4.8.10	Analysemodelle	150
4.8.11	Systemvision.....	150
4.9	Zusammenfassung der Unterschiede zwischen REPMUS und Wiegers’ Ansatz	150
5	Bewertung von REPMUS anhand der Anforderungen an das Requirements Engineering in änderungsintensiven Projekten.....	153
5.1	Erreichung der Zielsetzungen des Requirements Engineering.....	153
5.2	Berücksichtigung der Gründe für Änderungen der Anforderungen.....	156
5.3	Berücksichtigung der Schwierigkeiten aufgrund von Änderungen der Anforderungen.....	159
5.4	Zusammenfassung	161
6	Zusammenfassung	162

Literaturverzeichnis	164
Anhang A. System Dynamics Modell FAUSER AG.....	178
Anhang B. System Dynamics Modell PSIPENTA Software Systems GmbH	187

Abkürzungsverzeichnis

RE	Requirements Engineering
KMU	Kleine und mittlere Unternehmen
TTM	Time to Market
XP	eXtreme Programming

1 Einführung

Motivation

Instabile Anforderungen erhöhen die Schwierigkeit und das Risiko des Scheiterns von Softwareentwicklungsprojekten signifikant. In bestimmten Domänen, zum Beispiel web-basierten Anwendungen können sehr hohe Änderungsraten der Anforderungen auftreten, die ein für Softwareentwickler nur sehr schwer zu bewältigendes Problem darstellen (Boehm 2000). Die Art und Weise, mit der ein Projekt mit sich ändernden Anforderungen umgeht, kann entscheidend für seinen Erfolg sein.

Es existieren Entwicklungsmethoden, zum Beispiel agile Methoden, die für den Einsatz in Projekten mit instabilen Anforderungen gedacht sind (Beck 2000, Boehm 2002, Cockburn und Higshmith 2001). Dem Problem von auftauchenden Änderungen wird in diesen jedoch nicht auf Ebene des RE begegnet sondern es wird angestrebt in den nachfolgenden Entwicklungsaktivitäten ein hohes Maß an Flexibilität zu erreichen, vor allem durch kurze Kommunikationswege, mittels derer Änderungswünsche schnell dem Entwicklungsteam bekannt werden, und die Gewährleistung einer leichten Änderbarkeit der Entwicklungsartefakte. Dabei werden die generellen sowie die für änderungsintensive Projekte spezifischen Probleme im RE selbst nur ungenügend adressiert. Wie Leite (2001) am Beispiel von XP darstellt, müssen agile Methoden angepasst, beziehungsweise erweitert werden, um RE-spezifische Aspekte abzudecken. Andererseits gibt es viele Methoden, die für das RE Richtlinien vorgeben, jedoch nicht speziell auf änderungsintensive Projekte zugeschnitten sind. Diese Methoden umfassen das Spektrum von einzelnen Techniken wie Anwendungsfällen, Szenarios, Interviews, Requirements Workshops, etc. bis hin zu umfassenden Methoden beziehungsweise Prozessen, die viele Aspekte des RE adressieren wie zum Beispiel das Requirements Engineering Reference Model von Geisberger et al. (2006) oder der Ansatz von Robertson und Robertson (1999).

Um Projektlaufzeit und Kosten bei Projekten mit einer hohen Instabilität der Anforderungen gering zu halten, werden sowohl spezifische Techniken für einzelne Aktivitäten im RE benötigt als auch ein RE-Prozess, der den Spezifika von änderungsintensiven Projekten Rechnung trägt. Der Chaos Report (Standish Group 1995) hebt die Wichtigkeit des RE für den Projekterfolg heraus und betont, dass Mängel im RE zu stark erhöhten Kosten oder dem kompletten Scheitern von Projekten führen können, insbesondere, wenn sich ändernden Anforderungen nicht entsprechend begegnet wird. Gleichzeitig weisen laut Jones (1996a) 75 Prozent der Unternehmen Schwächen im RE auf. Gerade in kleineren und mittleren Unternehmen besteht der Bedarf, angepasste RE-Prozesse einzuführen beziehungsweise zu verbessern (Nikula et al. 2000).

Ziel

Ziel dieser Arbeit ist es, zum einen die generellen und für Projekte mit hoher Anforderungsinstabilität spezifischen Anforderungen an den RE-Prozess herauszuarbeiten. Zum anderen wollen wir wirksame und anwendbare Vorgehensweisen erarbeiten, die eine hohe Anforderungsinstabilität geeignet adressieren, und diese in einem beispielhaften RE-Prozess integrieren. Der RE-Prozess soll dabei den zuvor aufgestellten Anforderungen genügen und sich auf in der Praxis bewährte Vorgehensweisen abstützen.

Dies führt zu folgenden Forschungsfragen:

Forschungsfrage 1: Welche Anforderungen an den RE-Prozess stellen sich in Projekten mit hoher Anforderungsinstabilität?

Forschungsfrage 2: Welche Vorgehensweisen bei der Verteilung des RE-Aufwands eignen sich besonders im Rahmen von Projekten mit hoher Anforderungsinstabilität?

Forschungsfrage 3: Wie kann ein RE-Prozess gestaltet werden, so dass er den Anforderungen in Projekten mit hoher Anforderungsinstabilität gerecht wird?

Betrachtete Projektklasse

Die Art von Projekten, die wir in dieser Arbeit betrachten, findet sich beispielsweise sehr häufig in der Domäne der Web- und mobilen Applikationen und zeichnet sich wie folgt aus:

Häufige Änderungen von Anforderungen: Wir betrachten Projekte, die einem hohen Wandel der Anforderungen ausgesetzt sind. Als grober Richtwert seien hier monatliche Änderungsraten der Anforderungen von 3,5% und mehr genannt. Neue Anforderungen an das Produkt kommen häufig und beständig hinzu. Bereits bestehende Anforderungen fallen oft weg oder ändern sich. Zudem sind viele Anforderungen zu Beginn des Projekts unklar und schlecht verstanden. Man kann in diesen Projekten jedoch nicht von Anforderungen ausgehen, die bezüglich der Instabilität homogen sind. Vielmehr gibt es in der Regel eine Reihe von Anforderungen, die stabil sind. Bei dem instabilen Teil der Anforderungen kann man wiederum unterscheiden zwischen Änderungen, denen man mit geeigneten Methoden entgegenwirken kann, und solchen Änderungen, die man kaum oder gar nicht beeinflussen kann. Wie an späterer Stelle ausführlich dargelegt werden soll, entstehen erstere beispielsweise dadurch, dass Anforderungen schlecht verstanden sind. Durch Maßnahmen, die das Verständnis für Anforderungen steigern, kann dieser Art von Änderungen entgegengewirkt werden. Zu den nicht beeinflussbaren Änderungen sind zum Beispiel jene zu zählen, die sich aufgrund externer Einflüsse, wie beispielsweise neuer Technologien auf dem Markt, sich ändernder Standards oder Protokolle ergeben. Wir beschäftigen uns mit Projekten, bei denen sowohl gut zu stabilisierende Anforderungen als auch kaum beziehungsweise nicht zu stabilisierende Anforderungen häufig zu finden sind.

Marktgetrieben: Es werden Produkte entwickelt, die die entwickelnde Organisation selbst vermarktet. Es existiert kein Auftraggeber. Vor dem ersten Release eines Produkts gibt es daher normalerweise keine feste bekannte Gruppe von Kunden und Benutzern, die als Stakeholder in das Projekt eingebunden werden können. Es gibt höchstens potenzielle Kunden und Benutzer (Carmel und Becker 1995; Sawyer 2000). Eine zentrale Rolle bei der Ermittlung der Anforderungen kommt daher bei der Entwicklung der ersten Version der entwickelnden Organisation selbst zu. Erst bei der Entwicklung weiterer Versionen können tatsächliche Kunden und Benutzer einbezogen werden (Lubars et al. 1993; Potts 1995). Eine weitere Auswirkung dieses Charakteristikums ist, dass die Vertragsgestaltung kein bestimmender Faktor in Hinsicht auf die Gestaltung des Vorgehens im RE ist.

Kleine und mittelgroße Projekte von KMU: Abhängig von der Teamgröße sind unterschiedliche Vorgehensweisen im RE geeignet. Größere Teams verlangen striktere Prozesse und vor allem mehr Dokumentation. In größeren Unternehmen muss zudem oft die Befolgung und Pflege von organisationsweiten Prozessen, unter Umständen geleitet durch Reifegradmole, bedacht werden. Wir beschränken uns auf kleine bis mittlere Projekte von KMU an deren Durchführung 3-8 Personen beteiligt sind.

Anstreben einer geringen Projektdauer: In der Literatur wird TTM für marktgetriebene Unternehmen als überlebenskritisch bezeichnet (Novorita und Grube 1996). Falls ein zu entwickelndes System oder eine neue Version eines Systems nicht rechtzeitig auf den Markt gebracht werden kann, so hat dies wesentliche Auswirkungen auf den Marktanteil des Unternehmens (Sawyer et al. 1999). Wir betrachten daher Projekte, bei denen man

möglichst schnell auf den Markt kommen will und die Bereitschaft besteht, dafür auch im RE Zeitoptimierungen vorzunehmen. Weitere Versionen des Systems können später folgen. Bei diesen wird aber ebenfalls eine möglichst schnelle Entwicklung angestrebt.

Knappes Budget: Das Projekt soll möglichst geringe Kosten verursachen. Kostenminimierung wird angestrebt. Dies wirkt sich auf die Vorgehensweise im RE dahingehend aus, dass Effizienz angestrebt werden muss. Diese Einschränkung trifft auf fast alle Projekte zu. Wir schließen nur die Fälle aus, in denen die Möglichkeit besteht durch eine deutliche Erhöhung des Personaleinsatzes Schwierigkeiten in der Entwicklung zu überwinden. Für KMU ist es meist nicht, beziehungsweise nur sehr schwer möglich, zusätzliche Mitarbeiter zur Entwicklung des Systems heranzuziehen (Sawyer et al. 1999).

Softwareentwicklung: Wir betrachten Projekte, bei denen nur Software entwickelt wird und Aspekte der Hardware eine untergeordnete Rolle spielen. Bei der Entwicklung eingebetteter Systeme ist es für den reibungsfreien Ablauf der Hardwareentwicklung notwendig bestimmte Schnittstellen der Softwarekomponenten schon sehr früh im Projekt festzulegen. Daher muss schon in den frühen Phasen des Projekts eine gewisse Stabilität der Anforderungen herbeigeführt werden. Derartige Einschränkungen nehmen wir für die von uns betrachteten Projekte nicht an.

Keine verteilte Entwicklung: Wir gehen davon aus, dass die Entwicklung räumlich an einem Ort geschieht. Direkte und informelle Kommunikation zwischen den Projektbeteiligten ist möglich. Reisekosten und Reisezeiten müssen nicht beachtet werden.

Keine besonders hohen Qualitätsanforderungen: Es bestehen nur niedrige bis mittlere Anforderungen an die Qualität des Produktes. Dies bedeutet, dass Qualitätsanforderungen nicht zu den besonders hervorgehobenen und die Entwicklung bestimmenden Elementen gehören. Sie müssen jedoch trotzdem beachtet werden und stellen einen wichtigen Teil der Anforderungen dar.

Keine sicherheitskritischen Systeme: Sicherheitsaspekte spielen in den von uns betrachteten Projekten keine zentrale oder kritische Rolle. Dies bedeutet nicht, dass sie völlig außer Acht gelassen werden, doch sie sind nicht einer der besonders hervorgehobenen Aspekte in der Entwicklung. Strenge Verifizierungsmaßnahmen, wie sie bei der Entwicklung sicherheitskritischer Systeme erforderlich sind, sind für die von uns betrachteten Projekte daher nicht verpflichtend.

Wissenschaftliche Beiträge

- *Entwicklung eines Katalogs von Anforderungen an das RE in änderungsintensiven Projekten.* Ausgehend von einer Literaturstudie werden die allgemeinen Zielsetzungen des RE in einem Katalog gesammelt. Darüber hinaus werden basierend auf einer Literaturstudie sowie Interviews mit Industriepartnern allgemeine Ursachen für die Änderung von Anforderungen sowie Schwierigkeiten aufgrund von Anforderungsänderungen identifiziert, denen man in der von uns betrachteten Projektklasse Rechnung tragen muss.
- *Entwicklung eines System Dynamics Modells für die Analyse von RE-Vorgehensweisen zur Eingrenzung von Anforderungsinstabilität.* Anhand zweier Fallstudien wird ein System Dynamics Modell entwickelt. Dieses kann dazu verwendet werden Softwareentwicklungsprozesse in einem mathematischen Modell zu simulieren. Anhand des Modells kann abgeschätzt werden, welche Auswirkungen unterschiedliche Vorgehensweisen hinsichtlich des RE-Aufwands auf die Projektlaufzeit und die Projektkosten haben.

- *Ableitung von RE-Vorgehensweisen, die einen hinsichtlich des Anstrebens einer geringen Projektdauer und geringer Projektkosten geeigneten Umgang mit Anforderungsinstabilität gewährleisten, aus experimentellen Untersuchungen mittels des System Dynamics Modells.* In zwei Fallstudien wird das entwickelte System Dynamics Modell angewandt. Dabei werden RE-Vorgehensweisen zum in Hinblick auf Projektlaufzeit und Projektkosten geeigneten Umgang mit Anforderungsinstabilität abgeleitet.
- *Entwicklung eines RE-Prozesses für änderungsintensive und zeitbeschränkte Projekte.* Der RE-Ansatz von Wiegers (2005) wird ausgehend von den zuvor abgeleiteten RE-Vorgehensweisen zur Zeit- und Kostenreduktion und einer Literaturstudie sowie Interviews mit Industriepartnern über die in der betrachteten Projektklasse erfolgreich eingesetzten RE-Techniken so angepasst, dass der entstehende RE-Prozess für änderungsintensive und marktgetriebene Softwareentwicklungsprojekte von KMU geeignet ist.
- *Analyse des RE-Prozesses anhand des Katalogs von Anforderungen an das RE in änderungsintensiven Projekten.* Die Eignung des entwickelten RE-Prozesses für die betrachtete Projektklasse wird anhand des zuvor aufgestellten Katalogs von Anforderungen an das RE in änderungsintensiven Projekten untersucht.

2 Katalog von Anforderungen an das Requirements Engineering in änderungsintensiven Projekten

In diesem Kapitel identifizieren wir die in der von uns betrachteten Projektklasse relevanten Anforderungen an den RE-Prozess. Dafür erarbeiten wir auf der Basis einer Literaturstudie sowie Interviews mit Industriepartnern die generellen Zielsetzungen des RE sowie die gängigen Ursachen für Anforderungsänderungen und die wesentlichen Schwierigkeiten, die sich durch Änderungen ergeben (Fritzsche und Broy 2008). Die Ergebnisse dieses Kapitels werden in dieser Arbeit dafür genutzt, RE-Vorgehensweisen zu konzipieren, die unter Beachtung der sonstigen Rahmenbedingungen aus Kapitel 1 geeignet mit instabilen Anforderungen umgehen. Darüber hinaus wird ein RE-Prozess aufgestellt bewertet, der diese Vorgehensweisen integriert.

In den folgenden Unterkapiteln stellen wir die wesentlichen von RE-Methoden zu beachtenden Kriterien in änderungsintensiven und zeitkritischen Projekten der Reihe nach vor.

2.1 Zielsetzungen des Requirements Engineering

In diesem Abschnitt diskutieren wir die Zielsetzungen des RE, d.h. die generellen Aspekte, die RE-Methoden stets berücksichtigen müssen, unabhängig von Anforderungsinstabilität.

2.1.1 Klärung des Begriffs „Zielsetzung des Requirements Engineering“

Sucht man in der Literatur nach den Zielsetzungen des RE, findet man Aussagen wie zum Beispiel die folgenden:

„... the part of development in which people attempt to discover what is desired“; (Gause und Weinberg 1989)

„Thus, from a customer’s point of view, a requirements stage is necessary because it helps to understand the new needs and to identify how they can be satisfied.“ (Macaulay 1996)

„Seldom stated, the goal of requirements engineering is not to write extensive requirement documents but to effectively transfer ideas from the customer to the developer.“ (Goetz 2002)

„The aim of RE is to help to know what to build before system development starts in order to prevent costly rework.“ (Paetsch et al. 2003)

„Developing a consistent and comprehensive specification of the ‚desired‘ system is the overall objective of RE.“ (Geisberger et al. 2006)

Die genannten Autoren identifizieren wichtige Aspekte beziehungsweise Teilziele des RE, wie zum Beispiel dass herausgefunden werden muss, was vom zu entwickelnden System gewünscht wird oder dass die Ideen des Kunden den Entwicklern kommuniziert werden sollen. Wir benötigen jedoch eine umfassende und detaillierte Liste von allen Aspekten, die eine RE-Methode berücksichtigen muss. Zudem widersprechen sich manche der aufgeführten Definitionen gegenseitig. So stellt für Geisberger et al. (2006) die Entwicklung einer Spezifikation des gewünschten Systems die grundlegende Zielsetzung dar. Goetz (2002) hingegen betont, dass nicht das Schreiben von Anforderungsdokumenten die Zielsetzung des RE ist, sondern der Ideentransfer vom Kunden zu den Entwicklern.

Es existiert keine Arbeit, die die einzelnen Zielsetzungen des RE detailliert diskutiert. Es gibt jedoch umfangreiche Literatur darüber, welche Aktivitäten dem RE zuzuordnen sind und wie sie durchgeführt werden sollten. Darüber hinaus werden Empfehlungen gegeben, welche Dokumente im RE zu verfassen sind und welche Informationen sie beinhalten sollten. Vielfach berücksichtigte Arbeiten hierzu sind die von Davis (1993), Gause und Weinberg (1989), Jackson (1995), IEEE (1998a; 1998b), Kotonya und Sommerville (1998), Macaulay (1996), Nuseibeh und Easterbrook (2000), Robertson und Robertson (1999), Sommerville und Sawyer (1997), Wiegers (2005) sowie van Lamsweerde (2009). Diese Arbeiten empfehlen, wie man vorgehen sollte, beantworten dabei aber nicht explizit die Frage nach der Zielsetzung, also warum die Aktivitäten durchgeführt werden müssen. Wir wollen jedoch die allgemeinen Anforderungen an eine RE-Methode aufstellen, die für das gesamte Spektrum an RE-Methoden gelten. Ein beispielhaftes Vorgehen anzugeben, welches nützlich ist, um eine Zielsetzung zu erreichen ist für unsere Belange nicht zielführend, wie im Folgenden dargelegt werden soll.

Meist werden als allgemeine Aktivitäten, die im RE durchgeführt werden müssen, Ermittlung, Analyse, Spezifikation, Validierung und Management von Anforderungen genannt. Wiegers schlägt zum Beispiel für die Anforderungsanalyse die folgenden Unteraktivitäten vor: Kontextdiagramm zeichnen, Prototypen erstellen, Machbarkeit analysieren, Anforderungen priorisieren, Anforderungsmodell entwickeln, usw. Diese Aktivitäten weisen zwar den von uns benötigten Detaillierungsgrad auf, stellen jedoch nur Vorschläge dar. Nicht jede RE-Methode verwendet Prototyping. Wiegers beschreibt nicht die Zielsetzungen des RE, sondern Mittel, mit denen diese erreicht werden können. Zwei wesentliche Zielsetzungen hinter Prototyping sind es, Anforderungen zu verstehen und zu überprüfen, ob die Entwicklungsergebnisse den Wünschen der Stakeholder entsprechen.

Sieht man sich eXtreme Programming an, so kann das Problem noch besser verstanden werden. Die Methode weist einige Mängel bezüglich der Anforderungsspezifikation auf. Anforderungen werden in der Form von Story Cards und Akzeptanztests dokumentiert. Details der Anforderungen müssen in Gesprächen mit den Stakeholdern geklärt werden, welche nicht dokumentiert werden. Auf diese Weise wird das Problem fehlender Dokumente durch den Einsatz direkter Kommunikation behoben. Würden wir uns nur auf Sommerville und Sawyers (1997) oder auch Wiegers (2005) Modell stützen, so würden wir diese alternative Vorgehensweise nicht berücksichtigen. Unser Katalog an Anforderungen an das RE baut auf den Zielsetzungen auf, die sich hinter den vorgeschlagenen Aktivitäten verbergen. In diesem Fall würden wir also aufnehmen, dass eine RE-Methode dafür sorgen muss, dass die Anforderungen den relevanten Projektbeteiligten kommuniziert werden. Dies ist die Hauptzielsetzung der Anforderungsspezifikation, allerdings nicht die einzige. Sie dient unter anderem auch dazu Unklarheiten aufzudecken und das Verständnis für die Anforderungen zu erhöhen.

Die einzelnen Zielsetzungen des RE bilden eine Hierarchie. In dieser stellt jedes Element ein Mittel dar, das dazu beiträgt die ihm übergeordnete Zielsetzung zu erreichen. Für die Entscheidung, ob eine Aktivität eine Zielsetzung ist, bewerten wir, ob die Aktivität ein allgemeiner Aspekt ist, den jede RE-Methode berücksichtigen muss.

2.1.2 Gängiges Vorgehen im Requirements Engineering

In diesem Kapitel wollen wir einen Überblick über die in der Literatur beschriebenen typischen high-level Aktivitäten im RE geben. Diese dienen uns im folgenden Kapitel als Ausgangspunkt für die Ermittlung der Zielsetzungen des RE. Bei der Identifikation der typischen RE-Aktivitäten stützen wir uns auf Davis (1993), Gause und Weinberg (1989),

Geisberger et al. (2006), Jackson (1995), IEEE (1998a; 1998b), Kotonya und Sommerville (1998), van Lamsweerde (2009), Macaulay (1996), Nuseibeh und Easterbrook (2000), Robertson und Robertson (1999), Sommerville und Sawyer (1997) sowie Wiegers (2005). Es soll an dieser Stelle keine detaillierte Darstellung des üblichen Vorgehens erfolgen. Eine solche kann der Literatur entnommen werden. Vielmehr sollen alle wesentlichen RE-Aktivitäten auf hohem Abstraktionsgrad beschrieben werden. Die vorgestellten Aktivitäten sind aus einer Reihe von Standardwerken zusammengetragen. Sie stellen in ihrer Gesamtheit kein einheitlich gebrauchtes Vorgehen dar. Ein Standardvorgehen im RE gibt es nicht. Die meisten Ansätze beinhalten nur einen Teil der vorgestellten Aktivitäten. Da die Aktivitätsliste jedoch als Basis für die Bestimmung der Zielsetzungen des RE dienen soll, haben wir sie möglichst umfassend gestaltet. Einige der grundlegenden RE-Aktivitäten haben wir bereits im vorigen Kapitel identifiziert. Diese wollen wir unter dem Punkt „Anforderungen entwickeln“ subsumieren. Darüber hinaus werden in der Literatur zwei Querschnittsthemen als wichtig angesehen: „Alle relevanten Stakeholder beteiligen“ und „ein ingenieurmäßiges Vorgehen anwenden“. Damit erhält unsere Aktivitätsliste die in Tabelle 2-1 dargestellte Grundstruktur.

Anforderungen entwickeln
Anforderungen ermitteln
Anforderungen analysieren
Anforderungen spezifizieren
Anforderungen validieren
Anforderungen managen
Alle relevanten Stakeholder beteiligen
Ein ingenieurmäßiges Vorgehen anwenden

Tabelle 2-1: Grundstruktur der üblichen RE-Aktivitäten

Anforderungen ermitteln

Bei der Erhebung von Anforderungen wird eine Reihe von Informationen aus verschiedenen Quellen gesammelt. Typische Quellen sind dabei Kunden und Marketing. Aber auch andere Stakeholder oder Dokumente wie Standards, Bestimmungen, Dokumentationen von ähnlichen Systemen, etc. können Informationen beisteuern. Es werden zunächst Ziele ermittelt. Darunter sind Anforderungen auf hohem Abstraktionsniveau zu verstehen. Dies können beispielsweise Geschäftsziele sein, die den geschäftlichen Nutzen betreffen, den das zu entwickelnde System bietet. Kundenziele beschreiben die Bedürfnisse von Kunden, beziehungsweise typischen Anwendern des Systems. Hierbei sind die Ziele aller relevanten Stakeholder zu berücksichtigen. Des Weiteren wird eine Systemvision aufgestellt. Sie umreißt auf hohem Abstraktionsniveau, wie das System gestaltet werden soll um die aufgestellten Ziele zu erreichen. Neben strategischen Aspekten, wie potenziellen Märkten oder Ressourcenbeschränkungen werden die Hauptfunktionen und Leistungsmerkmale des Systems beschrieben. Darüber hinaus werden die Systemgrenzen abgesteckt. Man ermittelt die für das System relevanten Ausschnitte der Anwendungsdomäne. Dazu gehören Benutzer und andere Systeme, die mit dem zu entwickelnden System interagieren. Es wird dabei festgelegt, welche Aufgaben das System übernimmt und welche nicht. Weiterhin werden die für das System zu berücksichtigenden Rahmenbedingungen ermittelt. Dazu gehören juristische, technische, soziale, organisatorische, ökonomische und politische Faktoren, die bei der Entwicklung

des Systems berücksichtigt werden müssen. Schließlich werden Anforderungen aufgestellt. Sie werden dabei von den definierten Zielen und den Rahmenbedingungen abgeleitet. Die Anforderungen stellen Verfeinerungen der Ziele und der Rahmenbedingungen dar und können ihrerseits wieder verfeinert werden. Es existiert eine Hierarchie von Anforderungen. Wichtig ist es, die wahren Anforderungen herauszufinden, also diejenigen, die den echten Zielen der Stakeholder entsprechen. Oft werden Anforderungen angeführt, die für die Erreichung der Ziele gar nicht notwendig sind. Zum Beispiel könnte der Kunde fordern, dass das System in der Programmiersprache C geschrieben wird, obwohl dies gar nicht nötig ist. Sein eigentlicher Wunsch ist, dass das System kurze Antwortzeiten hat. Er trifft in diesem Fall eine Implementierungsentscheidung und seine wahre Anforderung sowie gegebenenfalls bessere alternative Wege zur Erfüllung der Anforderung bleiben unentdeckt. Oft wird gefordert, dass die Gründe für das Aufstellen der einzelnen Anforderungen angegeben werden. Dieses Vorgehen ist unter anderem auch speziell dazu geeignet, die wahren Anforderungen herauszufinden. In Tabelle 2-2 sind die der Literatur entnommenen Unteraktivitäten zur Anforderungsermittlung zusammengefasst.

Anforderungen ermitteln	Informationen sammeln
	Ziele ermitteln
	Systemvision entwickeln
	Systemgrenzen abstecken
	Rahmenbedingungen aufnehmen
	Anforderungen sammeln
	Anforderungen verfeinern
	Wahre Anforderungen ermitteln
	Gründe für die Anforderungen identifizieren

Tabelle 2-2: Übliche Unteraktivitäten zur Anforderungsermittlung

Anforderungen analysieren

Eine der Hauptaufgaben während der Anforderungsanalyse ist es, ein höheres Verständnis für die ermittelten Anforderungen zu gewinnen. Dafür werden vielfältige Analysemethoden, wie zum Beispiel Modellierung eingesetzt. Zudem sollen Beziehungen zwischen den Anforderungen identifiziert werden. Damit können die Anforderungen weiter strukturiert werden. Des Weiteren sollen die Auswirkungen der Anforderungen auf die Geschäftsabläufe analysiert werden. Man versucht die Konsequenzen zu verstehen, die eine Verwirklichung der Anforderungen nach sich ziehen würde. Somit kann auch die Profitabilität der Anforderungen abgeschätzt werden. Während der Anforderungsanalyse werden den einzelnen Anforderungen Prioritäten zugeordnet. Ebenso werden Risiken, die mit den Anforderungen verbunden sind, ermittelt. Es wird angestrebt, alle Anforderungen vollständig zu ermitteln. Eine absolute Vollständigkeit ist aufgrund der vielfältigen Einflüsse und aufgrund von möglichen Änderungen der Anforderungen praktisch nicht erreichbar. Dieser Punkt muss daher so verstanden werden, dass man versucht eine möglichst hohe Abdeckung der Anforderungen zu erreichen. Die dabei aufgestellten Anforderungen sollen zudem auch notwendig sein. Notwendig ist eine Anforderung dann, wenn sie zur Erreichung der Geschäfts- und Kundenziele oder zur Erfüllung der Rahmenbedingungen nötig ist. Zudem soll die Korrektheit der Anforderungen gewährleistet werden. Dies bedeutet, dass die aufgestellten Anforderungen den Vorstellungen der Stakeholder entsprechen. Sie sollen außerdem eindeutig sein, also so

formuliert sein, dass sie nicht missverstanden werden. Ein weiterer zu beachtender Punkt ist die Machbarkeit der Anforderungen. Die aufgestellten Anforderungen sollen mit den vorhandenen Mitteln umsetzbar sein. Dazu gehört auch ihre Widerspruchsfreiheit. Gibt es sich gegenseitig widersprechende Anforderungen, so müssen diese Konflikte aufgelöst werden. Anforderungen sollten nachprüfbar sein. Es soll also nach der Realisierung der Anforderungen möglich sein festzustellen, ob sie erfüllt sind, oder nicht. Schließlich müssen noch die zu realisierenden Anforderungen ausgewählt werden. Zum einen soll nicht mehr entwickelt werden als notwendig, da dies unnötige Kosten verursachen würde. Zum anderen kann es aufgrund beschränkter Ressourcen nötig sein bestimmte Anforderungen fallen zu lassen. Die in der Literatur empfohlenen Unteraktivitäten zur Anforderungsanalyse sind in Tabelle 2-3 zusammengefasst.

Anforderungen analysieren	Anforderungen verstehen	
	Beziehungen zwischen Anforderungen identifizieren	
	Auswirkungen der Anforderungen auf die Geschäftsabläufe verstehen	
	Return on Investment ermitteln	
	Anforderungen priorisieren	
	Risiken ermitteln	
	Vollständigkeit gewährleisten	
	Notwendigkeit gewährleisten	
	Korrektheit gewährleisten	
	Eindeutigkeit gewährleisten	
	Machbarkeit gewährleisten	Widerspruchsfreiheit gewährleisten
	Nachprüfbarkeit gewährleisten	
	Zu realisierende Anforderungen auswählen	

Tabelle 2-3: Übliche Unteraktivitäten zur Anforderungsanalyse

Anforderungen spezifizieren

Hauptaufgabe in der Anforderungsspezifikation ist es, eine Spezifikation zu erstellen, die in Design und Implementierung genutzt werden kann. Die Anforderungen sollen verständlich beschrieben werden. Von einer „guten“ Anforderungsspezifikation wird vielfach verlangt folgende Kriterien zu erfüllen: Vollständigkeit, Modifizierbarkeit, Rückverfolgbarkeit und Konsistenz. Alle aufgenommenen Anforderungen sollen also zunächst vollständig spezifiziert werden. Dies bedeutet, dass sie so dokumentiert werden, dass alle für die weitere Entwicklung notwendigen Details in ausreichendem Maß beschrieben sind. Sie sollen in einer Art und Weise dokumentiert werden, die es erlaubt die Anforderungen leicht zu ändern. Daher müssen einzelne Anforderungen auffindbar und ihre Beschreibung leicht anpassbar sein. Dafür werden Anforderungen in der Regel eindeutig benannt. Vielfach gefordert wird in diesem Zusammenhang eine Struktur, die der Anforderungsspezifikation zugrunde liegt. Eine solche erleichtert das Auffinden und Modifizieren von Anforderungen. Des Weiteren sollen spezifizierte Anforderungen zurückverfolgbar sein. Es soll also klar sein, woher sie sich ableiten. Dafür wird in der Regel die Quelle der Anforderung dokumentiert. Quellen können Stakeholder, Ziele, Rahmenbedingungen oder auch Anforderungen auf höherem Abstraktionsniveau sein. Schließlich sollen die Anforderungen auf konsistente Art und Weise dokumentiert werden,

spricht die äußere Form der Spezifikation sollte über alle Anforderungen hinweg einheitlich sein. In Tabelle 2-4 sind die in der Literatur empfohlenen Unteraktivitäten zur Anforderungsspezifikation zusammengefasst.

Anforderungen spezifizieren	Eine Spezifikation erstellen, die in Design und Implementierung genutzt werden kann
	Anforderungen verständlich machen
	Vollständigkeit gewährleisten
	Modifizierbarkeit gewährleisten
	Rückverfolgbarkeit gewährleisten
Konsistenz gewährleisten	

Tabelle 2-4: Übliche Unteraktivitäten zur Anforderungsspezifikation

Anforderungen validieren

In der Anforderungvalidierung wird die Anforderungsspezifikation überprüft. Es soll festgestellt werden, ob darin alle gewünschten Anforderungen enthalten sind und ob sie auf korrekte Weise dokumentiert wurden. Des Weiteren sollen Anforderungen getestet werden. Dabei versucht man das erwartete Systemverhalten durch Testfälle nachzuvollziehen. Die Testfälle leiten sich dabei von den funktionellen Anforderungen ab. Auf diese Weise können Analysemodelle und Prototypen verifiziert werden. Außerdem können die Stakeholder überprüfen, ob die aufgestellten Anforderungen zu einem System führen, das ihren Bedürfnissen entspricht. Hierfür werden von den Stakeholdern Akzeptanzkriterien definiert. Die in der Literatur empfohlenen Unteraktivitäten zur Anforderungvalidierung sind in Tabelle 2-5 zusammengefasst.

Anforderungen validieren	Anforderungsspezifikation auf Vollständigkeit und Korrektheit überprüfen
	Anforderungen testen
	Akzeptanzkriterien definieren

Tabelle 2-5: Übliche Unteraktivitäten zur Anforderungvalidierung

Anforderungen managen

Im Laufe eines Projektes muss man auf Änderungen der Anforderungen reagieren können. Daher ist ein Verfahren zur Änderung von Anforderungen zu definieren. Dieses beschreibt, wie Änderungen an den Anforderungen eingebracht, analysiert und entschieden werden. Änderungen von Anforderungen haben Auswirkungen auf Teile des Systems, andere Anforderungen und das Projekt allgemein. Vielfach wird empfohlen die Auswirkungen einer Anforderungsänderung zu analysieren. Es wird eine Baseline der Anforderungsspezifikation erstellt. Die Erstellung weiterer Versionen wird kontrolliert. Über Änderungen der Spezifikation entscheidet das Change Control Board. Änderungen werden protokolliert. Dabei werden Zeitpunkt und Begründung der Änderung aufgenommen. Zu den Anforderungen wird ihr derzeitiger Status aufgenommen. Dieser kann zum Beispiel vorgeschlagen, bewilligt oder implementiert sein. Weiterhin soll die Verfolgbarkeit der Anforderungen gewährleistet werden. Die Anforderungen können zum

Beispiel durch eine Anforderungsrückverfolgbarkeitsmatrix mit Elementen aus dem Design, dem Code und den Tests verknüpft werden. Anforderungen können auch mit Anforderungen höheren Abstraktionsgrads oder solchen, mit denen sie in Beziehung stehen, verknüpft werden. Einer Überwachung unterliegen auch die den Anforderungen zugeordneten Risiken und Aufwände. Diese werden immer wieder neu bestimmt beziehungsweise aktualisiert. Die üblicherweise empfohlenen Unteraktivitäten zum Anforderungsmanagement sind in Tabelle 2-6 zusammengefasst.

Anforderungen managen	Prozess zur Änderung von Anforderungen definieren
	Auswirkungen von Änderungen der Anforderungen analysieren
	Änderungen an der Anforderungsspezifikation kontrollieren
	Status der Anforderungen überwachen
	Anforderungen verfolgen
	Anforderungsrisiken kontrollieren
	Anforderungsaufwand kontrollieren

Tabelle 2-6: Übliche Unteraktivitäten zum Anforderungsmanagement

Alle relevanten Stakeholder beteiligen

Es gibt stets eine Vielzahl an Parteien, die ein Interesse an dem zu entwickelnden Projekt haben. Die Interessen aller dieser Parteien sollten berücksichtigt werden. Hierfür muss mit den für das Projekt relevanten Stakeholdern kommuniziert werden. Die verschiedenen Klassen von Stakeholdern müssen identifiziert und einbezogen werden. Stakeholder sind alle Personen, die einen direkten oder indirekten Einfluss auf die Anforderungen des Systems haben. Besonders wichtige Stakeholder sind Endbenutzer. Aber auch alle anderen Personen, die von dem System betroffen sind stellen relevante Stakeholder dar. Diejenigen, die Nachbarsysteme entwickeln und warten, Geschäftsleiter, Domänenexperten, juristische Experten, die Marketingabteilung, usw. können ebenfalls Stakeholder sein. Da Anforderungen auch an den Prozess, nach dem das System entwickelt werden soll, gestellt werden können, stellen auch alle Prozessbeteiligten Stakeholder dar: Projektleiter, Qualitätssicherung, Architekten, Entwickler, etc. Die Kommunikation unter den Stakeholdern wird als wichtig angesehen. Insbesondere die Kommunikation zwischen Stakeholdern und Entwicklung ist hervorzuheben. Oft haben Stakeholder falsche Vorstellungen über die Machbarkeit ihrer Bedürfnisse oder gar über ihre Ziele selbst. Diese müssen aufgedeckt und beseitigt werden. In diesem Zusammenhang ist es wichtig die Zwänge der Realisierung und die Grenzen der Anforderungen den Stakeholdern zu vermitteln. Risiken sind den Stakeholdern nicht immer bewusst könnten ihre Wünsche aber beeinflussen. Daher müssen sie ihnen vermittelt werden. Die empfohlenen Aktivitäten bezüglich der Beteiligung der Stakeholder sind in Tabelle 2-7 zusammengefasst.

Alle relevanten Stakeholder beteiligen	Alle Interessen berücksichtigen	
	Stakeholderklassen identifizieren	
	Mit Stakeholdern kommunizieren	Kommunikation zwischen Entwicklung und Stakeholdern ermöglichen
	Falsche Vorstellungen beseitigen	Zwänge der Realisierung vermitteln
		Grenzen der Anforderungen vermitteln
Risiken kommunizieren		

Tabelle 2-7: Übliche Aktivitäten bezüglich der Beteiligung der Stakeholder

Ein ingenieurmäßiges Vorgehen anwenden

Im RE soll ein geplanter und strukturierter Ansatz angewendet werden. Dies bedeutet zum einen, dass der RE-Prozess klar definiert sein soll. Er soll im Vorhinein festlegen, welche Arbeitsschritte durchzuführen sind, welche Artefakte dabei erstellt werden müssen, wer an der Erstellung beteiligt ist und wer dafür verantwortlich ist. Es sollten Beziehungen zwischen Aktivitäten und zwischen Artefakten definiert werden, aus denen sich eine zeitliche Reihenfolge ableitet, nach der Aktivitäten durchgeführt und Artefakte erstellt werden. Der Prozess weist damit eine festgelegte Struktur auf. Zudem sollte der Prozess wiederholbar und verständlich sein. In der Regel wird daher eine strukturierte Dokumentation des Prozesses gefordert. Des Weiteren soll festgelegt werden, nach welchen konkreten Methoden die einzelnen Aktivitäten durchgeführt werden sollen und welche Notationen bei der Erstellung der Artefakte Anwendung finden. In Tabelle 2-8 ist zusammengefasst, was in der Literatur unter der Anwendung eines ingenieurmäßigen Vorgehens verstanden wird.

Ein ingenieurmäßiges Vorgehen anwenden	Einem geplanten, strukturierten Vorgehen folgen
	Verständlichkeit und Wiederholbarkeit des Prozesses sicherstellen
	Konkrete Notationen und Methoden bereitstellen

Tabelle 2-8: Übliche Aktivitäten bezüglich der Anwendung eines ingenieurmäßigen Vorgehens

Zusammenfassung

In diesem Kapitel haben wir die üblicherweise in der Literatur empfohlenen RE-Aktivitäten überblicksartig vorgestellt. Die Liste der dabei identifizierten Aktivitäten ist in Tabelle 2-9 dargestellt.

Anforderungen entwickeln		
Anforderungen ermitteln	Informationen sammeln	
	Ziele ermitteln	Geschäftsziele ermitteln
		Kundenziele ermitteln
	Systemvision entwickeln	
	Systemgrenzen abstecken	
	Rahmenbedingungen aufnehmen	
	Anforderungen sammeln	Geschäftsanforderungen sammeln
		Kundenanforderungen sammeln
	Anforderungen verfeinern	
	Wahre Anforderungen ermitteln	
	Gründe für die Anforderungen identifizieren	
Anforderungen analysieren	Anforderungen verstehen	
	Beziehungen zwischen Anforderungen identifizieren	
	Auswirkungen der Anforderungen auf die Geschäftsabläufe verstehen	
	Return on Investment ermitteln	
	Anforderungen priorisieren	
	Risiken ermitteln	
	Vollständigkeit gewährleisten	
	Notwendigkeit gewährleisten	
	Korrektheit gewährleisten	
	Eindeutigkeit gewährleisten	
	Machbarkeit gewährleisten	Widerspruchsfreiheit gewährleisten
	Nachprüfbarkeit gewährleisten	
	Zu realisierende Anforderungen auswählen	
Anforderungen spezifizieren	Eine Spezifikation erstellen, die in Design und Implementierung genutzt werden kann	
	Anforderungen verständlich machen	
	Vollständigkeit gewährleisten	
	Modifizierbarkeit gewährleisten	
	Rückverfolgbarkeit gewährleisten	
	Konsistenz gewährleisten	
Anforderungen validieren	Anforderungsspezifikation überprüfen	
	Anforderungen testen	
	Akzeptanzkriterien definieren	
Anforderungen managen	Prozess zur Änderung von Anforderungen definieren	
	Auswirkungen von Änderungen der Anforderungen analysieren	
	Änderungen an der Anforderungsspezifikation kontrollieren	
	Status der Anforderungen überwachen	
	Anforderungen verfolgen	
	Anforderungsrisiken kontrollieren	
	Anforderungsaufwand kontrollieren	
Alle relevanten Stakeholder beteiligen		
Alle Interessen berücksichtigen		
Stakeholderklassen identifizieren		
Mit Stakeholdern kommunizieren	Kommunikation zwischen Entwicklung und	

	Stakeholdern ermöglichen
Falsche Vorstellungen beseitigen	Zwänge der Realisierung vermitteln
	Grenzen der Anforderungen vermitteln
Risiken kommunizieren	
Ein ingenieurmäßiges Vorgehen anwenden	
	Einem geplanten, strukturierten Vorgehen folgen
	Verständlichkeit und Wiederholbarkeit des Prozesses sicherstellen
	Konkrete Notationen und Methoden bereitstellen

Tabelle 2-9: In der Literatur üblicherweise empfohlene high-level RE-Aktivitäten

2.1.3 Ableitung der Zielsetzungen des Requirements Engineering aus dem gängigen Vorgehen

Zu jeder der üblichen RE-Aktivitäten, die im vorangegangenen Kapitel vorgestellt wurde, wollen wir nun die primären Zielsetzungen, die mit ihr verfolgt werden identifizieren. Aktivitäten können mehrere Zielsetzungen haben. Beispielsweise ist es die primäre Zielsetzung der Anforderungsspezifikation, Anforderungen zu kommunizieren. Als Nebeneffekt wird unter anderem auch das Verständnis der Anforderungen erhöht. Wir beschränken unsere Analyse auf die primären Zielsetzungen der Aktivitäten. Es wird angenommen, dass Nebeneffekte bereits durch die primären Zielsetzungen anderer Aktivitäten abgedeckt sind. Wir ordnen die gefundenen Zielsetzungen in einer Hierarchie. Die Ableitung der Zielsetzungen basiert auf unseren eigenen Überlegungen sowie Diskussionen mit Industriepartnern.

Zunächst wollen wir die Aktivitäten, die mit der Anforderungsermittlung verbunden sind, analysieren. Informationen zu sammeln ist eine sehr allgemeine Aktivität, die nur schwerlich mit einer einzigen Zielsetzung verbunden werden kann. Abhängig von der Art der gesammelten Information werden unterschiedliche Zielsetzungen verfolgt. Im Kontext der Anforderungsermittlung können zwei spezifische Zielsetzungen identifiziert werden. Zum einen versucht man die Bedürfnisse der Stakeholder herauszufinden. Zum anderen soll ein breites Verständnis der Domäne, der Organisation und der Geschäftsprozesse gewonnen werden. Die Ermittlung der Ziele hat ebenfalls die Zielsetzung, die Bedürfnisse der Stakeholder herauszufinden. Auch die Entwicklung einer Systemvision umfasst mehrere Aspekte. Hierbei sollen Ziele, die mit der Entwicklung des Systems verfolgt werden, herausgefunden und diese allen relevanten Stakeholdern kommuniziert werden. Der Kommunikationsaspekt beinhaltet wiederum mehrere weiterführende Zielsetzungen. Diese werden wir im Kontext der Anforderungsspezifikation betrachten. Durch das Abstecken der Systemgrenzen können die Schnittstellen zu Benutzern und anderen Systemen identifiziert werden. Die damit verbundene generelle Zielsetzung ist, die Ziele herauszufinden, die mit der Entwicklung des Systems verfolgt werden. Der Fokus der Aktivität ist jedoch, klarzumachen, welche Ziele dabei vom System erfüllt werden sollen und welche durch andere Mittel außerhalb des Systems, wie beispielsweise Nachbarsysteme, erreicht werden müssen. Wir nehmen das Abstecken der Systemgrenzen als eigene Zielsetzung mit auf. Eine weitere typische RE-Aktivität ist das Aufnehmen der Rahmenbedingungen. Die damit verfolgte Zielsetzung ist es, ein breites Verständnis der Domäne, der Organisation und der Geschäftsprozesse zu gewinnen. Die Aktivität „Anforderungen sammeln“ stellt selbst eine Zielsetzung dar: Anforderungen aufstellen, die notwendig sind, um die Bedürfnisse der Stakeholder abzudecken. Sie ist der bereits

identifizierten Zielsetzung, die Bedürfnisse der Stakeholder herauszufinden, untergeordnet. Das Verfeinern von Anforderungen hat dieselbe Zielsetzung wie das Sammeln von Anforderungen, da nach unserem Verständnis der Begriff „Anforderung“ Anforderungen jeden Detaillierungsgrads umfasst. Die Identifikation der Gründe für die Anforderungen stellt wiederum selbst eine Zielsetzung dar. Sie stellt eine Detaillierung der übergeordneten Zielsetzung, die Bedürfnisse der Stakeholder herauszufinden, dar. Die Zielsetzung hinter der Aktivität, die wahren Anforderungen zu ermitteln, ist es wiederum, Anforderungen aufzustellen, die die Bedürfnisse der Stakeholder abdecken. Hierbei ist zu bemerken, dass die Anforderungen nur dann aufgestellt werden, wenn sie der Erfüllung der Bedürfnisse der Stakeholder dienen. Es müssen daher diejenigen unter den eingebrachten Anforderungen herausgefiltert werden, die nicht der Erfüllung der Bedürfnisse der Stakeholder dienlich sind. Außerdem bezeichnen wir mit Bedürfnissen der Stakeholder nur deren wahre Interessen und nicht diejenigen, die sie irrtümlicherweise zu haben glauben. Wir benutzen das Wort „herausfinden“ in dem Ziel „Die Bedürfnisse der Stakeholder herausfinden“ um zu verdeutlichen, dass die Bedürfnisse der Stakeholder diesen keinesfalls im Vorhinein bewusst sind. Es ist ein gewisser Aufwand nötig, um sie zu bestimmen.

Nun wollen wir RE-Zielsetzungen von den Aktivitäten der Anforderungsanalyse ableiten. Anforderungen zu verstehen dient neben den bereits identifizierten Zielsetzungen, die Gründe für Anforderungen zu identifizieren und die Bedürfnisse der Stakeholder herauszufinden auch dazu die Auswirkungen des Systems auf die Geschäftsabläufe zu verstehen. Beziehungen zwischen Anforderungen müssen identifiziert werden, damit Konflikte gefunden und aufgelöst werden können. Die Aktivität „Auswirkungen der Anforderungen auf die Geschäftsabläufe verstehen“ dient der bereits identifizierten Zielsetzung „Auswirkungen des Systems auf die Geschäftsabläufe verstehen“. Der Return on Investment soll ermittelt werden, um zu gewährleisten, dass das System profitabel ist. Dafür müssen die Rendite des Systems und die der einzelnen Anforderungen bestimmt werden. Der Kostenfaktor spielt auch später eine Rolle, da während des gesamten Projektverlaufs das Budget beachtet werden muss. Das Festlegen von Prioritäten ist wichtig, um die Auflösung von Konflikten zu unterstützen. Anforderungen niedriger Priorität werden zugunsten wichtigerer Anforderungen fallengelassen oder geändert. Die zuvor identifizierte Zielsetzung wird also wie folgt erweitert: Konflikte zwischen Anforderungen auflösen und dabei den wichtigeren den Vorrang geben. Prioritäten werden zudem dazu benötigt, auszuwählen welche Anforderungen realisiert werden sollen. Oft können aufgrund von Zeit- und Budgetknappheit nicht alle Ziele erreicht werden. Es muss daher bestimmt werden, welche Ziele und welche Anforderungen Vorrang haben: die wichtigsten Ziele, die mit der Entwicklung des Systems verfolgt werden, erreichen, sich klar machen, welche Anforderungen nicht realisiert werden können und die notwendigen Anforderungen zur Erreichung der wichtigsten Ziele auswählen. Risiken zu ermitteln stellt selbst eine Zielsetzung dar. Darüber hinaus sollen auch Vorkehrungsmaßnahmen getroffen werden. Wir leiten daher folgende Zielsetzung ab: Risiken in der Entwicklung identifizieren und Vorkehrungsmaßnahmen sowie Ausweichpläne bereitstellen. Vollständigkeit zu gewährleisten zielt darauf ab, dass alle Bedürfnisse der Stakeholder herausgefunden werden sollen. Die Gewährleistung der Notwendigkeit bedeutet, dass nur diejenigen Anforderungen aufgestellt werden sollen, die wirklich notwendig sind, um die Bedürfnisse der Stakeholder abzudecken. Korrektheit und Eindeutigkeit zu gewährleisten, also die Anforderungen korrekt zu erfassen, stellt eine separate Zielsetzung dar. Die Machbarkeit zu gewährleisten, stellt ebenfalls eine eigene Zielsetzung dar: Nur machbare Anforderungen berücksichtigen. Die Zielsetzung hinter der Aktivität „Nachprüfbarkeit gewährleisten“ ist es, zu überprüfen, dass die Realisierung den Anforderungen entspricht:

Einhaltung der Anforderungen gewährleisten. Auszuwählen, welche Anforderungen realisiert werden sollen, ist eine eigenständige Zielsetzung: die notwendigen Anforderungen zur Erreichung der wichtigsten Ziele auswählen.

In der Folge konzentrieren wir uns auf Aktivitäten, die auf die Anforderungsspezifikation bezogen sind. Hinter der Aktivität „Eine Spezifikation erstellen, die in Design und Implementierung genutzt werden kann“ verbirgt sich die Zielsetzung, die Anforderungen den Entwicklern zu kommunizieren. Dies geschieht in der Regel durch Dokumente. Eine dazu übergeordnete Zielsetzung ist es, die notwendigen Anforderungen zu realisieren. Die Entwickler müssen die Anforderungen richtig verstehen. Daher müssen diese auf korrekte und verständliche Weise kommuniziert werden. Dies wird durch folgende Zielsetzungen abgedeckt: „gewährleisten, dass die Anforderungen für die Entwickler verständlich sind“ und „Anforderungen korrekt erfassen“. Vollständigkeit zu gewährleisten bedeutet, dass alle Bedürfnisse aller relevanten Stakeholder herausgefunden werden und dass alle Anforderungen, die notwendig sind, um diese Bedürfnisse abzudecken, aufgestellt werden müssen. Bereits aufgestellte Anforderungen können sich ändern. Daher müssen sie modifizierbar sein. Einerseits können Änderungen auftreten, weil nicht zielführende Anforderungen aufgestellt wurden, die nicht den Bedürfnissen der Stakeholder entsprechen. Andererseits können sich Anforderungen ändern, weil sich die Bedürfnisse der Stakeholder ändern. Die damit verbundenen Zielsetzungen sind die folgenden: Anforderungen aufstellen, die notwendig sind, um die Bedürfnisse der Stakeholder abzudecken und Änderungen der Bedürfnisse der Stakeholder beachten. Mit der Gewährleistung von Anforderungsrückverfolgbarkeit versucht man einerseits, die Anforderungvalidierung zu unterstützen, indem man bewirkt, dass die aufgestellten Anforderungen mit den Bedürfnissen der Stakeholder abgeglichen werden können. Damit verbundene Zielsetzungen sind wiederum die Anforderungen aufzustellen, die notwendig sind, um die Bedürfnisse der Stakeholder abzudecken, und die Gründe für Anforderungen zu ermitteln. Zudem will man bewirken, dass andere Entwicklungsartefakte den aufgestellten Anforderungen entsprechen. Dies hat die Zielsetzung die Einhaltung der Anforderungen zu gewährleisten. Eine weitere übliche Aktivität im RE ist es Konsistenz zu gewährleisten. Die damit verbundene Zielsetzung ist auch hier, die notwendigen Anforderungen aufzustellen, die die Bedürfnisse der Stakeholder abdecken. Außerdem stellt Konsistenz der Spezifikation ein Qualitätsmerkmal dar. Man verfolgt damit also gleichzeitig die Zielsetzung, die Qualität des Entwicklungsprozesses zu fördern.

Als nächstes werden wir Aktivitäten analysieren, die mit der Anforderungvalidierung verbunden sind. Durch das Überprüfen der Anforderungsspezifikation soll gewährleistet werden, dass die Anforderungen korrekt erfasst wurden. Das Testen von Anforderungen und das Aufstellen von Akzeptanzkriterien haben zusätzlich die Zielsetzung, das Verständnis für die aufgestellten Anforderungen zu erhöhen. Eine zugehörige Zielsetzung ist, alle Bedürfnisse der Stakeholder herauszufinden und zu überprüfen, ob sie durch die aufgestellten Anforderungen vollständig abgedeckt sind. Zudem soll die Einhaltung der Anforderungen gewährleistet werden.

Nun wenden wir uns Aktivitäten aus dem Anforderungsmanagement zu. Mit der Definition eines Prozesses zur Änderung von Anforderungen werden dieselben Zielsetzungen verfolgt, wie mit der Gewährleistung der Modifizierbarkeit der Spezifikation: Anforderungen aufstellen, die notwendig sind, um die Bedürfnisse der Stakeholder abzudecken und Änderungen der Bedürfnisse der Stakeholder beachten. Auswirkungen von Änderungen der Anforderungen werden analysiert, um zu bestimmen, ob diese machbar sind. Außerdem soll für Änderungen überprüft werden, ob ihre Wichtigkeit hoch genug ist, um ihre Kosten und das eventuelle Weglassen von mit ihnen im Konflikt

stehenden Anforderungen zu rechtfertigen. Die Zielsetzung ist also zum einen zu ermitteln, welche Anforderungen in der Entwicklung Vorrang haben und zum anderen sich klar zu machen, welche Anforderungen nicht realisiert werden können. Änderungen an der Anforderungsspezifikation sollen kontrolliert werden. Damit soll unter anderem gewährleistet werden, dass Änderungen an den Anforderungen den Entwicklern kommuniziert werden. Die Überwachung des Anforderungsstatus sowie das Verfolgen von Anforderungen werden durchgeführt um zu gewährleisten, dass das System den Anforderungen entspricht. Anforderungsrisiken zu kontrollieren beinhaltet mehrere Zielsetzungen. Risiken sollen identifiziert und den Stakeholdern kommuniziert werden. Darüber hinaus sollen Vorkehrungsmaßnahmen getroffen und Ausweichpläne bereitgestellt werden. Eine weitere übliche RE-Aktivität ist die Kontrolle der mit den Anforderungen verbundenen Aufwände. Es muss gewährleistet sein, dass die Anforderungen innerhalb des Budgets sowie des Zeitplans realisiert werden können. Die zugehörigen Zielsetzungen sind „Budget beachten“ und „Zeitplan beachten“.

Das nächste Thema unserer Analyse ist die Beteiligung aller relevanten Stakeholder. Dabei sollen Stakeholderklassen identifiziert und möglichst alle Interessen berücksichtigt werden. Dies hat die Zielsetzung, alle Bedürfnisse aller relevanten Stakeholder herauszufinden. Die Kommunikation mit den Stakeholdern hat zwei Zielsetzungen. Zum einen ist sie notwendig um ihre Bedürfnisse zu ermitteln. Zum anderen ist es wichtig, ihnen Feedback aus der Entwicklung zu geben. Sie müssen über die Machbarkeit der Anforderungen und die mit ihnen verbundenen Risiken informiert werden. Ein wichtiger Nebeneffekt ist, dass das Verständnis der Stakeholder über die Anforderungen verbessert wird. Oft haben Stakeholder eine falsche Wahrnehmung hinsichtlich der Machbarkeit ihrer Anforderungen oder kennen ihre wahren Bedürfnisse nicht. Diese falschen Vorstellungen sollen beseitigt werden. Die angeführten Punkte sind in folgenden Zielsetzungen abgedeckt: „alle Bedürfnisse aller relevanten Stakeholder herausfinden“ und „den Stakeholdern die Machbarkeit der Anforderungen kommunizieren“.

Zum Abschluss unserer Analyse widmen wir uns noch den Zielsetzungen, die mit der Anwendung eines ingenieurmäßigen Vorgehens verbunden sind. Mit dem Einsatz eines geplanten, strukturierten Vorgehens versucht man die Qualität des Entwicklungsprozesses zu gewährleisten. Die Zielsetzung, die man bei der Verfolgung von Prozessqualität hat, ist letztlich die Erhöhung der Produktqualität. Dieselbe Zielsetzung steht hinter der Forderung, die Verständlichkeit und Wiederholbarkeit des Prozesses sicherzustellen. Auch die Bereitstellung von konkreten Notationen und Methoden dient letztlich diesem Zweck.

In Tabelle 2-10 haben wir alle Zielsetzungen des RE, die wir in diesem Kapitel identifiziert haben, zusammengefasst.

Die Ziele, die mit der Entwicklung des Systems verfolgt werden, herausfinden				
Alle Bedürfnisse aller relevanten Stakeholder herausfinden		Anforderungen aufstellen, die notwendig sind, um die Bedürfnisse der Stakeholder abzudecken		
		Gründe für Anforderungen ermitteln		
		Änderungen der Bedürfnisse der Stakeholder beachten		
Ein breites Verständnis der Domäne, der Organisation und der Geschäftsprozesse gewinnen				
Auswirkungen des Systems auf die Geschäftsabläufe verstehen				
Gewährleisten, dass die Entwicklung des Systems profitabel ist		Return on Investment des Systems und der einzelnen Anforderungen ermitteln		
Systemgrenzen abstecken				
Die wichtigsten Ziele, die mit der Entwicklung des Systems verfolgt werden, erreichen				
Die notwendigen Anforderungen zur Erreichung der wichtigsten Ziele auswählen				
Die notwendigen Anforderungen realisieren	Nur machbare Anforderungen berücksichtigen	Sich klar machen, welche Anforderungen nicht realisiert werden können	Konflikte zwischen Anforderungen auflösen und dabei den wichtigeren den Vorrang geben	Ermitteln, welche Anforderungen in der Entwicklung Vorrang haben
			Budget beachten	
			Zeitplan beachten	
			Den Stakeholdern die Machbarkeit der Anforderungen kommunizieren	
	Anforderungen den Entwicklern kommunizieren	Gewährleisten, dass die Anforderungen für die Entwickler verständlich sind		
		Anforderungen korrekt erfassen		
Einhaltung der Anforderungen gewährleisten	Risiken in der Entwicklung identifizieren und Vorkehrungsmaßnahmen sowie Ausweichpläne bereitstellen			
	Risiken den Stakeholdern kommunizieren			
Qualität des Produkts fördern				
Qualität des Entwicklungsprozesses gewährleisten				

Tabelle 2-10: Zielsetzungen des Requirements Engineering

Mit den ermittelten Zielsetzungen des RE verfügen wir über einen Katalog an allgemeinen Anforderungen an RE-Methoden, die in allen Projekten wichtig sind. In den von uns betrachteten Projekten ist mit der Instabilität der Anforderungen ein spezieller Aspekt vorhanden, der von der eingesetzten RE-Methode zusätzlich berücksichtigt werden sollte. Anforderungen an RE-Methoden, die sich von der Instabilität der Anforderungen herleiten, sollen im nächsten Kapitel ermittelt werden.

2.2 Anforderungen an das Requirements Engineering aufgrund der Anforderungsinstabilität

In diesem Kapitel ermitteln wir die Schwierigkeiten, die sich im Projekt aufgrund der Häufigkeit von Anforderungsänderungen stellen. Von ihnen lassen sich weitere Anforderungen an RE-Methoden in der von uns betrachteten Projektklasse ableiten. In Kapitel 2.2.1 stellen wir die wesentlichen Gründe dar, wegen derer sich Anforderungen ändern können. Die Probleme, die sich aus der Änderung von Anforderungen für das Projekt ergeben präsentieren wir in Kapitel 2.2.2. Die Ergebnisse in diesem Kapitel basieren auf einer Literaturstudie und unseren Diskussionen mit Industriepartnern, unter anderem in einem Projekt, das sich mit den Herausforderungen von KMU im Kontext der Wandlungsfähigkeit von Softwareprodukten sowie des organisatorischen Umfelds befasst (Broy et al. 2010).

2.2.1 Gründe für Änderungen der Anforderungen

In diesem Kapitel identifizieren und klassifizieren wir Gründe für Anforderungsänderungen. Der Chaos Report (Standish Group 1995) zeigt, dass instabile Anforderungen ein weitverbreitetes Problem sind. Jones (1996a) spricht im Kontext von kommerzieller Software von einer monatlichen Änderungsrate der Anforderungen von 3,5%. Wie mit diesem Problem umgegangen wird, ist einer der kritischen Erfolgsfaktoren für Softwareentwicklungsprojekte. Instabilität von Anforderungen wurde in etlichen empirischen Studien als ein häufiger Risikofaktor mit starken Auswirkungen auf das Projekt identifiziert (Boehm 1991; Curtis et al. 1988; Houston 2000; Jones 1994; Känsälä 1997; Moynihan 1997; Ropponen 1999; Ropponen und Lyytinen 2000; Schmidt et al. 2001). Uns ist jedoch keine Arbeit bekannt, in der die zentralen Gründe für Änderungen von Anforderungen vollständig zusammengetragen werden. Es existieren nur einige Arbeiten, die sich mit Teilaspekten auseinandersetzen, beispielsweise Ambler (2002), Davis (1993), Jones (1996b), Larman (2004), Strens und Sugden (1995) sowie Wan-Kadir und Loucopoulos (2004). Es ist schwierig, empirische Daten über das Problem von Anforderungsänderungen zu sammeln, da die Gründe für die Instabilität von Anforderungen sich je nach Domäne, Organisation und Projekt stark unterscheiden können. Dementsprechend werden in der Literatur keine quantitativen Aussagen darüber getroffen, in welchem Maße Anforderungsänderungen auf die einzelnen Ursachen zurückzuführen sind. Wir wollen uns in diesem Kapitel dem Problem qualitativ nähern, indem wir eine umfangreiche kategorisierte Liste der wesentlichen Gründe für Anforderungsänderungen erstellen. In Kapitel 3.4 werden für eine konkrete Organisation Aussagen über die quantitative Verteilung der Anforderungsänderungen auf die verschiedenen Änderungsgründe gemacht.

Als generelle Kategorisierung unterscheiden wir zwischen Faktoren, die beeinflusst werden können, sprich denen mittels eines geeigneten RE-Ansatzes begegnet werden kann, und denjenigen, die nicht beeinflussbar sind.

Zunächst konzentrieren wir uns auf Gründe für Anforderungsänderungen, die beeinflusst werden können.

Oft werden Anforderungen in den frühen Phasen des Projekts nicht vollständig verstanden. Das kann zum einen bedeuten, dass bestimmte Anforderungen nicht bekannt sind beziehungsweise übersehen werden (Sawyer et al. 1999). Zum anderen kann es sein, dass den Stakeholdern ihre wahren Bedürfnisse nicht bewusst sind oder dass sie die von ihnen

definierten Anforderungen und ihre Konsequenzen nicht wirklich verstehen. Während der Entwicklung verbessert sich ihr Verständnis, was dann Änderungen der Anforderungen nach sich zieht (Hall et al. 2002). Laut Jones (1996b) ist die Hauptursache für Anforderungsänderungen, dass die Domäne, in der das zu entwickelnde System eingesetzt werden soll nicht vollständig verstanden wird. Er führt weiter an, dass sich das Verständnis der Domäne erst im Laufe des Projektes verbessert und somit Anforderungsänderungen in vielen Projekten eine technische Notwendigkeit sind. Walia et al. (2006) sowie Walia und Carver (2009) heben mangelndes Wissen über die Domäne ebenfalls als einen der wesentlichen Faktoren, die zu Fehlern in den Anforderungen führen, heraus. Auch Nurmuliani et al. (2004) führen wachsendes Verständnis während der Entwicklung als eine der Hauptursachen für Anforderungsänderungen auf. Weiterhin kann mangelndes Verständnis die Konsequenz von Kommunikationsschwierigkeiten sein (Hall et al. 2002; Walia et al. 2006). Es kommt vor, dass Anforderungen fehlerhaft oder missverständlich beschrieben werden, so dass sie von anderen Stakeholdern und Entwicklern falsch verstanden werden. Strens und Sugden (1995) konstatieren, dass Spezifikationen, die zu offen und ungenau sind, einen Mangel an Details aufweisen und zu viel Interpretationsspielraum lassen, Missverständnisse und damit zusätzliche Anforderungsänderungen bewirken.

Im Folgenden werden einige Beispiele für Anforderungsänderungen gegeben. Dabei wollen wir unter anderem verdeutlichen auf welche Granularitätsebene der Anforderungen wir uns beziehen. Die Beispiele stellen Anforderungen an einen Webshop dar.

Anforderung A1: Das System schlägt den Benutzern die in den letzten sechs Monaten beliebtesten Artikel vor.

Anforderung A1.1: Alle Suchanfragen, die an den Webshop geschickt werden, werden für sechs Monate gespeichert.

Anforderung A1': Das System schlägt dem Benutzer Artikel vor, die denen ähnlich sind, für die er sich in den letzten sechs Monaten interessiert hat.

Anforderung A1.1': Das System merkt sich für jeden Benutzer alle Suchanfragen, die er in den vergangenen sechs Monaten an den Webshop geschickt hat.

Ein mögliches Szenario ist, dass die Anforderungen A1 und A1.1 aufgestellt wurden, die Stakeholder später jedoch merken, dass ihr eigentliches Ziel ein anderes ist. So wollen sie in Wirklichkeit nicht den Benutzern die beliebtesten Artikel aller Benutzer anzeigen, sondern gezielt solche, die den letzten Artikeln, für die sich der Benutzer selbst interessiert hat, ähnlich sind. Dies führt zur Änderungen der Anforderungen A1 und A1.1 zu den neuen Anforderungen A1' und A1.1'. An diesem Beispiel sehen wir zudem, dass Änderungen sich über mehrere Ebenen in der Anforderungshierarchie erstrecken können. Durch eine frühe Diskussion über die Anforderung und ihrer Verknüpfung zu den eigentlichen Zielen der Stakeholder könnte der Änderungsbedarf schon früh erkannt und die Anforderung angepasst werden. Es handelt sich also um eine stabilisierbare Anforderung.

Anforderung A2: Das System ordnet die Suchergebnisse und zeigt sie dem Benutzer an.

Anforderung A2': Das System ordnet die Suchergebnisse nach absteigender Ähnlichkeit zur Sucheingabe und zeigt sie dem Benutzer an.

Anforderung A2 kann von den Entwicklern missverstanden werden. So könnten sie beispielsweise das System so entwerfen, dass die Suchergebnisse alphabetisch geordnet werden, obwohl die Stakeholder eine Ordnung anhand der Ähnlichkeit zur Sucheingabe wünschen. Wird dieses Problem identifiziert, könnte dies beispielsweise zur geänderten Anforderung A2' führen. Die missverständliche Formulierung der Anforderung könnte schon früh beispielsweise durch ein Review oder Diskussionen identifiziert und ausgeräumt werden. Daher ist auch diese Anforderung stabilisierbar.

Ungenauere, beziehungsweise falsche Schätzungen der Kosten von Anforderungen zu Beginn des Projekts sind ein weiterer Grund für Anforderungsänderungen. In den meisten Projekten ist es sehr schwierig präzise Abschätzungen zu treffen. Während der Entwicklung kann klar werden, dass eine größere oder auch nur eine kleinere Anzahl an Anforderungen innerhalb des Budgets und des Zeitplans realisiert werden kann als zuvor angenommen.

Auch durch sich ändernde Prioritäten können sich Anforderungen ändern. Beispielsweise kann es sein, dass einige Anforderungen als unwichtig eingestuft wurden und zugunsten wichtigerer und mit ihnen im Konflikt stehender Anforderungen fallen gelassen wurden. Falls sie später eine höhere Priorität erhalten, kann es sein, dass sie wieder aufgenommen werden oder Änderungen an ihnen, die zur Konfliktauflösung vorgenommen wurden, rückgängig gemacht werden. Normalerweise ändern sich Prioritäten allein deswegen, weil sie zu Beginn falsch eingeschätzt wurden. Es kann jedoch auch sein, dass sie sich wegen externer Gründe, die nicht beeinflusst werden können, ändern, zum Beispiel wegen organisatorischer Änderungen oder Änderungen in der Umgebung des Systems.

Technische Probleme während der Entwicklung sind ein weiterer Grund für Änderungen der Anforderungen. Es kann sich herausstellen, dass Anforderungen nicht implementiert werden können. Entweder sind sie aus technischen Gründen überhaupt nicht realisierbar oder nur nicht innerhalb des verfügbaren Budgets und Zeitrahmens. Diese Situation tritt in der Regel deshalb auf, weil die Machbarkeit der Anforderungen falsch eingeschätzt wurde.

Ein weiterer Grund für Änderungen sind Konflikte zwischen Anforderungen, die zu spät entdeckt werden (Sawyer et al. 1999). Konflikte müssen aufgelöst werden. Dies geschieht indem man einzelne Anforderungen zugunsten anderer fallen lässt oder indem man Kompromisse findet, sprich Anforderungen abändert. Falls dies nicht in den frühen Phasen des Projektes geschieht, so muss dies zu einem späteren Zeitpunkt nachgeholt werden.

Änderungen können auch auftreten wenn neue Stakeholder in das Projekt eingeführt werden. Dies tritt in der Regel deswegen auf, weil versäumt wurde alle relevanten Stakeholder zu Beginn des Projektes zu identifizieren und in das Projekt einzubeziehen (Walia et al. 2006). Es ist jedoch auch möglich, dass aufgrund von externen Faktoren neue Stakeholder hinzukommen ohne, dass diese schon früher hätten gefunden werden können.

Werden Anforderungen falsch verstanden, so kann dies unmittelbar auch zu einer falschen Abschätzung ihrer Kosten, Priorität und Machbarkeit sowie das Übersehen von Konflikten führen. Der Grund für entsprechende Fehler muss aber nicht im mangelnden Verständnis liegen. Diese können jeweils auch für richtig verstandene Anforderungen, z.B. aufgrund von Fehlern in der Kostenschätzung selbst, auftreten.

Nun wollen wir Faktoren betrachten, die nicht durch eine geeignete Vorgehensweise im RE beeinflusst werden können.

Geschäftsprozesse können sich ändern (Wan-Kadir und Loucopoulos 2004). Sie können sich zum einen wegen der Einführung des Systems ändern (Lehman 1980). Falls diese

Änderungen nicht antizipiert wurden und die aufgestellten Anforderungen diese Änderungen außer Acht lassen, führt dies zu zusätzlichen Anforderungsänderungen. Oft ist es jedoch sehr schwierig, diese Änderungen vorherzusehen, so dass eine Vorhersage nicht praktikabel ist. Es kann von Vorteil sein durch ein geeignetes Vorgehen zu versuchen Änderungen vorherzusagen, bei denen dies ohne großen Aufwand möglich ist. Geschäftsprozesse können sich jedoch auch unabhängig von der Einführung des Systems durch externe Einflüsse, die nicht vorhergesehen werden können, ändern. Zudem können sich Schnittstellen mit anderen Systemen ändern.

Neben Änderungen in der unmittelbaren Umgebung des Systems müssen auch externe Faktoren berücksichtigt werden. Die Einführung neuer Technologien auf dem Markt ist eine der grundlegenden Ursachen für Änderungen (Nurmuliani et al. 2004). In bestimmten Domänen kann dies sehr häufig der Fall sein. Wegen neuer Technologien kann zusätzliche Funktionalität gefordert werden oder es können sich neue Möglichkeiten ergeben, wie das System realisiert werden kann oder soll.

Aufgrund wirtschaftlicher Änderungen oder neuer Konkurrenten auf dem Markt können sich Geschäftsstrategie und –prioritäten der Organisation, die das System einsetzen will, ändern. Damit können Änderungen der Anforderungen verbunden sein (Nurmuliani et al. 2004).

Anforderungsänderungen können nötig werden, wenn sich das Budget ändert. Wird das Budget gekürzt, so müssen möglicherweise Anforderungen gestrichen oder abgeändert werden. Wird das Budget erhöht, können Anforderungen Beachtung finden, die zuvor aufgrund von Budgetknappheit außer Acht gelassen wurden. Ähnliches gilt für die zeitliche Planung. Analog zur Budgetplanung kann auch hier der Zeitplan verkürzt oder verlängert werden. Während Änderungen des Budgets und des Zeitplans sehr wohl ein Grund für Anforderungsänderungen sein können, sind sie jedoch in der Regel nicht besonders häufig. Daher stellen sie zwar einen Faktor dar, der zu der Häufigkeit der Änderungen beiträgt, spielen aber eine vergleichsweise geringe Rolle.

Des Weiteren können sich Gesetze, Regelungen oder Standards, die das System betreffen, ändern. Daraus resultieren möglicherweise Anforderungsänderungen.

In Tabelle 2-11 sind die identifizierten Gründe für die Instabilität von Anforderungen zusammengefasst.

Ursachen für Anforderungsänderungen, die vom Vorgehen im RE unmittelbar beeinflusst werden können	
Faktoren bezüglich des Verständnisses der Anforderungen	Anforderungen wurden in den frühen Phasen des Projekts nicht verstanden
	Stakeholder entwickeln im Laufe des Projektes ein besseres Verständnis für das gewünschte System
	Anforderungen wurden aufgrund von Kommunikationsproblemen falsch interpretiert
Falsche Schätzungen der Kosten von Anforderungen	Das Budget genügt nicht für die zur Realisierung ausgewählten Anforderungen
	Der Zeitplan genügt nicht für die zur Realisierung ausgewählten Anforderungen
Prioritäten der Anforderungen ändern sich	
Machbarkeit der Anforderungen wurde falsch eingeschätzt	Anforderungen können aufgrund technischer Probleme nicht realisiert werden
	Anforderungen können nicht innerhalb des Zeitplans und des Budgets realisiert werden
Konflikte zwischen Anforderungen werden gefunden	
Neue Stakeholder werden am Projekt beteiligt	
Ursachen für Anforderungsänderungen, die vom Vorgehen im RE nicht unmittelbar beeinflusst werden können	
Änderungen an den Geschäftsprozessen in der unmittelbaren Umgebung des Systems	Geschäftsprozesse im Umfeld des Systems ändern sich
	Schnittstellen mit Nachbarsystemen ändern sich
Einführung neuer Technologien	Neue Technologien, mit denen das System auf andere Weise realisiert werden kann
	Neue Technologien, die in das Produkt integriert werden sollen
Faktoren, die die Organisation betreffen	Geschäftsstrategie oder -prioritäten ändern sich
	Änderung des Budgets
	Änderung des Zeitplans
Gesetze, Regelungen oder Standards ändern sich	

Tabelle 2-11: Gründe für die Instabilität von Anforderungen

Es ist vorteilhaft, wenn eine RE-Methode zwischen Änderungsgründen unterscheidet, die beeinflusst werden können und jenen, bei denen dem nicht der Fall ist. Der übliche Ansatz um Anforderungsänderungen zu vermeiden ist eine lange und intensive RE-Phase zu Beginn des Projekts, in der man versucht Anforderungen möglichst gut zu verstehen, ihre Prioritäten, Kosten und Machbarkeit richtig einzuschätzen und Konflikte zu beseitigen. Der damit verbundene zeitliche Aufwand muss jedoch durch den Nutzen gerechtfertigt sein. Es ist somit das Rechte Maß des RE-Aufwands beziehungsweise der Maßnahmen zur Reduzierung der Anforderungsinstabilität zu finden.

2.2.2 Schwierigkeiten aufgrund von Änderungen der Anforderungen

Nachdem wir die Gründe für Änderungen der Anforderungen identifiziert haben, wollen wir in diesem Kapitel näher auf die aus Änderungen resultierenden Probleme eingehen. Wir stellen die Schwierigkeiten vor, die es zu bewältigen gilt, sowie dabei notwendig werdende Maßnahmen.

Während es keine Arbeit gibt, die die Schwierigkeiten aufgrund von Änderungen von Anforderungen umfassend bespricht, so gibt es einige Arbeiten die Einzelaspekte besprechen. Tomayko (2002) sowie Thayer und Dorfman (1997) konzentrieren sich auf das Problem der Kostenschätzung in änderungsintensiven Projekten. Boehm (1981), Boehm und Papaccio (1988) sowie Brooks (1995) diskutieren ausführlich die Kosten von Änderungen. Zu diesem Thema existieren auch einige Studien, zum Beispiel Fagan (1974), Boehm et al. (1975), Boehm (1976) und Daly (1977). In unserer Arbeit wollen wir uns jedoch nicht auf Einzelaspekte beschränken, sondern eine umfangreiche Liste der wesentlichen Schwierigkeiten im RE erarbeiten, die durch Änderungen der Anforderungen entstehen.

Notwendige Änderungen an Anforderungen müssen erkannt werden. Da sich Anforderungen während des gesamten Projekts ändern können, müssen für die gesamte Projektlaufzeit Möglichkeiten gegeben sein, notwendige Änderungen zu erkennen. Werden Änderungsnotwendigkeiten wahrgenommen so müssen Änderungsvorschläge eingebracht werden. Schließlich muss über die Annahme der Änderungsvorschläge entschieden werden.

Sich ändernde Anforderungen erzeugen zusätzlichen Aufwand. Zunächst muss darüber entschieden werden, ob Änderungsvorschläge angenommen oder abgelehnt werden. Dafür müssen die Entscheidungsträger versammelt werden, normalerweise in Form des sogenannten Change Control Boards. Um jedoch die gesamte Entscheidungskompetenz zu berücksichtigen müssten alle relevanten Stakeholdereinbezogen werden. Theoretisch könnte es genügen nur diejenigen Stakeholder zu versammeln, deren Interessen von den Änderungen betroffen sind. Welche das sind, ist im Vornherein jedoch oft nicht klar. Zudem könnten einzelne Projektteilnehmer sich Vorteile davon versprechen, bestimmte Personen aus Entscheidungen herauszuhalten, selbst wenn oder gerade weil deren Interessen dabei berührt werden. Wenn der Aufwand, alle relevanten Stakeholder bei jeder Entscheidung zu versammeln gemieden werden soll, müssen Mechanismen gefunden werden, die es erlauben die betroffenen Stakeholder zu identifizieren und ihnen angemessenen Einfluss auf die Entscheidung zu gewährleisten. Wurde eine Änderung angenommen, müssen davon betroffene Artefakte angepasst werden. Der erste Schritt bei der Anpassung ist es, die betroffenen Artefakte sowie die Stellen, an denen Änderungen vorzunehmen sind, zu identifizieren. Trivialerweise ist zunächst die Anforderungsspezifikation betroffen. Die neuen beziehungsweise geänderten Anforderungen müssen darin aufgenommen werden. Doch auch Anpassungen von Entwurfsdokumenten, Code, Testfällen und anderer Artefakte können notwendig sein. Ist eine Anforderung zum Zeitpunkt ihrer Änderung schon fertig implementiert und getestet, so ist eine Vielzahl von Artefakten von der Änderung betroffen. Hilfreich bei der Identifikation betroffener Artefakte ist es, wenn Anforderungen zu sämtlichen Dokumenten, die von ihnen abhängen, verfolgt werden können. Da das Bereitstellen einer solchen Verfolgbarkeit ebenfalls einen Zusatzaufwand darstellt, muss überlegt werden, ob dieser durch den Nutzen gerechtfertigt ist. Abhängig von der Art der Änderungen müssen Artefakte geändert, verworfen oder hinzugefügt werden. Um die Kosten gering zu halten, ist es vorteilhaft Artefakte erweiterbar und änderbar zu halten. Da die Gewährleistung von

Modifizierbarkeit selbst auch Kosten verursacht, muss abgeschätzt werden, ob sich diese rechtfertigen lassen. Software an neue Anforderungen anzupassen ist oft kompliziert und fehleranfällig (Brooks 1995). Belady und Lehman untersuchen diesen Aspekt ausführlich in ihren Arbeiten (1976 und 1985). Sie stellen dabei fest, dass sich die Struktur von Programmen im Laufe der Zeit durch beständige Änderungen verschlechtert. Es wird daher immer schwieriger weitere Änderungen vorzunehmen, ohne dass zusätzliche Fehler eingeführt werden. Da Anforderungen sich im Laufe des Projektes ändern oder gar wegfallen können, kann der Aufwand, diese zu analysieren und zu realisieren vergeudet sein. Daher ist es nicht immer für alle Anforderungen sinnvoll zu Beginn eines Projektes viel Aufwand für ihre Analyse und Dokumentation zu betreiben. Es sollte überprüft werden, ob dieser Aufwand sich trotz des Risikos, dass er umsonst sein könnte, rechtfertigen lässt. Da neue Anforderungen stets Zusatzaufwand bedeuten, existiert eine Hemmschwelle diese aufzunehmen (Schwaber 2002). Es kann zu einer verringerten Qualität des Systems führen, wenn Anforderungen aus diesem Grund abgelehnt werden.

Ein weiteres Problem bei instabilen Anforderungen ist, dass nicht alle Konflikte zwischen Anforderungen in den frühen Phasen des Projekts erkannt werden können, da zu Beginn nicht alle Anforderungen vorliegen. Die Konfliktanalyse muss für geänderte Anforderungen erneut durchgeführt werden. Anforderungen, die mit neuen Anforderungen höherer Priorität im Konflikt stehen, müssen unter Umständen fallengelassen werden. Der Aufwand, der bislang bei ihrer Realisierung betrieben wurde, ist in diesem Fall vergeudet. Es gibt eine Hemmschwelle neue Anforderungen aufzunehmen, wenn klar ist, dass sie im Konflikt mit bereits bestehenden sind. Werden sie deswegen nicht aufgenommen so kann dies zu Einbußen bei der Akzeptanz des Systems führen.

Gedanken zu den Anforderungen, die zu Beginn der Entwicklung gefasst wurden, sind später nicht mehr präsent. Problematisch ist dies vor allem dann, wenn eine erneute Konfliktanalyse durchgeführt werden muss, oder wenn die Prioritäten der Anforderungen aktualisiert werden sollen. Zusätzliche Schwierigkeiten können entstehen, wenn relevante Stakeholder nicht mehr verfügbar sind. Wichtige Gedanken sind vor allem die Begründung für die Priorität der Anforderung oder der Grund für das Verwerfen einer Anforderung. Man kann diesem Problem entgegenwirken, indem man versucht alle Gedanken zu dokumentieren. In der Praxis ist es jedoch quasi unmöglich sie vollständig zu dokumentieren. Zudem bedeutet dies zusätzlichen Aufwand, welcher wiederum gerechtfertigt werden muss.

Ändern sich Anforderungen so muss der Projektplan in der Regel umgestellt werden. Aufwands- und Risikoabschätzungen sich ändernder Anforderungen oder zu diesen abhängigen Anforderungen müssen aktualisiert werden. Des Weiteren kann sich der Inhalt von Releases oder Iterationen ändern. Zeit- und Kostenplanung müssen überprüft werden. Die Untersuchungen von Nurmaliani et al. (2006) zeigen an, dass Änderungen aufgrund von Konflikten in der Regel nicht besonders teuer sind. Daher ist die frühe Auflösung von Konflikten für die Begrenzung der Kosten nicht von essenzieller Bedeutung.

Die Kosten des Projekts stellen eine weitere Schwierigkeit dar. Da nicht alle Anforderungen in den frühen Phasen des Projekts bekannt sind, ist nicht klar, wie viel das System kosten wird (Tomayko 2002). Änderungen können womöglich nicht im Rahmen des Budgets realisiert werden. Daher muss entweder das Budget erweitert oder Anforderungen fallengelassen werden. Die Einführung neuer oder Änderung bestehender Anforderungen erst spät im Projekt ist oft besonders teuer (Boehm 1981; Nurmaliani et al. 2006).

In Tabelle 2-12 haben wir die Schwierigkeiten, die sich im Projekt durch die Änderung von Anforderungen ergeben, zusammengefasst.

Notwendige Änderungen müssen identifiziert werden	
Änderungsvorschläge müssen eingebracht werden	
Es muss über die Annahme von Änderungsvorschlägen entschieden werden	
Zusätzlicher Aufwand entsteht	
Stakeholder müssen erneut versammelt werden, um über die Annahme von Änderungsvorschlägen zu entscheiden	
Artefakte müssen angepasst werden	Anzupassende Artefakte müssen identifiziert werden
	Artefakte müssen modifiziert, entfernt oder hinzugefügt werden
	Artefakte sollten erweiterbar und modifizierbar sein
	Anpassungen sind oft kompliziert und fehleranfällig
Anfänglicher Aufwand kann vergeudet sein, wenn Anforderungen sich ändern	
Anforderungen werden möglicherweise abgelehnt, da man den Zusatzaufwand scheut	
Nicht alle Konflikte können zu Projektbeginn erkannt werden	
Konfliktanalyse muss mehrfach im Laufe des Projekts durchgeführt werden	
Entwicklungsergebnisse müssen teilweise verworfen werden, weil sie im Konflikt mit neuen Anforderungen stehen	
Wichtige Anforderungen werden möglicherweise abgelehnt, da sie im Konflikt mit alten Anforderungen stehen	
Nicht dokumentierte Überlegungen zu den Anforderungen sind später nicht mehr präsent, werden jedoch bei Anforderungsänderungen wieder benötigt	
Projektplan muss angepasst werden	
Schätzungen müssen aktualisiert werden	
Inhalt von Releases und Iterationen kann sich ändern	
Zeitplan muss angepasst werden	
Budget muss angepasst werden	
Kostenprobleme entstehen	
Kosten des Systems sind zu Beginn des Projektes nicht bekannt	
Änderungen lassen sich möglicherweise nicht innerhalb des Budgets realisieren	
Späte Änderungen können besonders teuer sein	

Tabelle 2-12: Schwierigkeiten aufgrund von Anforderungsänderungen

2.3 Zusammenfassung

In diesem Kapitel haben wir wichtige Anforderungen, die RE-Methoden in änderungsintensiven Projekten berücksichtigen müssen, aufgenommen. Wir sind dabei detailliert darauf eingegangen, welche Zielsetzungen man im RE generell verfolgt. Zudem haben wir analysiert wodurch es zu Änderungen der Anforderungen kommen kann.

Schließlich haben wir Probleme aufgrund von Anforderungsänderungen identifiziert, denen eine RE-Methode begegnen muss.

Die Ergebnisse aus diesem Kapitel können zum einen dazu verwendet werden um bestehende RE-Methoden auf Tauglichkeit für änderungsintensive Projekte zu analysieren. Zum anderen stellen sie einen Leitfaden, bei der Konzeption von auf diese Projektsituation zugeschnittenen RE-Methoden, dar.

1. **Zielsetzungen des RE:** RE-Methoden müssen effektive Mittel zur Erreichung der beschriebenen Zielsetzungen bereitstellen.
2. **Gründe für Änderungen der Anforderungen:** In der von uns betrachteten Projektklasse müssen RE-Methoden die aufgeführten Gründe für Änderungen berücksichtigen beziehungsweise, falls praktikabel, geeignete Mittel einsetzen, die den Änderungsursachen entgegenwirken.
3. **Schwierigkeiten aufgrund von Änderungen von Anforderungen:** RE-Methoden, die für Projekte mit besonders instabilen Anforderungen eingesetzt werden, müssen geeignete Vorgehensweisen bereitstellen um mit den aufgeführten Schwierigkeiten umzugehen, beziehungsweise diese zu lindern.

Die Ergebnisse dieses Kapitels nutzen wir in dieser Arbeit dafür RE-Vorgehensweisen zum geeigneten Umgang mit Anforderungsinstabilität sowie einen RE-Prozess, der diese Vorgehensweisen integriert, zu entwerfen und zu bewerten.

Der Anforderungskatalog wurde zudem dazu verwendet einige der bekanntesten agilen Methoden zu untersuchen (Fritzsche 2008): XP, Scrum (Schwaber 1995; Schwaber und Beedle 2002; Schwaber 2004), die Crystal Methodologies (Cockburn 2002), die Dynamic Systems Development Method (Stapleton, 1997) und Adaptive Software Development (Highsmith 2000). Es zeigte sich, dass agile Methoden weitgehend dazu geeignet sind, die grundlegenden Zielsetzungen des RE zu erreichen. Zum gleichen Ergebnis kamen wir schon in einer früheren Untersuchung von XP und Scrum (Fritzsche und Keil 2000a, 2000b). Wir konnten jedoch auch einige Schwächen der untersuchten Methoden im Kontext der von uns betrachteten Projektklasse identifizieren. Hervorzuheben ist, dass die untersuchten agilen Methoden zu Beginn des Projektes nur wenig Aufwand in ein besseres Verständnis der Anforderungen investieren. Dies wiederum zieht eine höhere Änderungsrate der Anforderungen und damit höhere Änderungskosten nach sich. Die Vertreter von agilen Methoden führen oft an, dass Anforderungen sich ohnehin ändern werden. Wie wir in diesem Kapitel bereits diskutiert haben, gibt es jedoch vielfältige Ursachen für die Instabilität von Anforderungen. Änderungen können auf externe Gründe zurückzuführen sein, die nicht beeinflussbar sind. Es gibt jedoch auch Instabilitätsfaktoren, die vom gewählten Vorgehen im RE herrühren oder durch dieses beeinflussbar sind. Wie wir in Kapitel 3 zeigen werden, ist es daher nicht immer und für alle Anforderungen sinnvoll aus Gründen der Zeitersparnis den Aufwand im RE möglichst gering zu halten.

3 Ein System Dynamics Modell für die Analyse von RE-Vorgehensweisen zur Eingrenzung von Anforderungsinstabilität

In diesem Kapitel wird ein System Dynamics Modell aufgestellt und in zwei Fallstudien angewendet. Mittels des System Dynamics Modells simulieren wir Softwareentwicklungsprozesse in einem mathematischen Modell und schätzen die Auswirkungen unterschiedlicher Vorgehensweisen hinsichtlich des RE-Aufwands auf die Änderungsrate der Anforderungen sowie Projektlaufzeit und –kosten ab. Im Anschluss leiten wir RE-Vorgehensweisen ab, die hinsichtlich Projektlaufzeit und –kosten geeignet mit instabilen Anforderungen umgehen.

System Dynamics wurde als Simulationsmethode gewählt, da damit das dynamische Verhalten eines Softwareentwicklungsprozesses mitsamt den positiven und negativen Wirkungsketten unter den betrachteten Parametern (siehe hierzu Kapitel 3.2.2) beschrieben werden kann. System Dynamics erlaubt es uns zudem Verbesserungen im Systemverhalten durch Parameter- und Strukturänderungen zu untersuchen.

Zunächst geben wir eine Einführung in die Simulation im Software Engineering und gehen dabei genauer auf System Dynamics Modelle ein. Wir geben eine Übersicht über bestehende Arbeiten, die ähnliche Themen wie die unsere untersuchen, und grenzen uns von diesen ab. Schließlich stellen wir unser System Dynamics Modell auf und wenden es in zwei Fallstudien an.

3.1 Simulation im Software Engineering

Simulation ist eine in den Ingenieursdisziplinen vielfach eingesetzte Methode, um das Verhalten von Prozessen, Risiken oder komplexen Systemen zu analysieren. Sie erlaubt es, Analysen durchzuführen bevor umfangreiche Zeit- und Kosteninvestitionen getätigt werden (Müller und Pfahl 2008). Insbesondere System Dynamics Modelle wurden bereits vielfach erfolgreich im Bereich Software Engineering eingesetzt (Pfahl 2001). Für die Erstellung und Ausführung von Simulationsmodellen existiert eine Reihe von Werkzeugen. Es gibt zudem eine Reihe gut dokumentierter Anwendungen von Simulationsmethoden in den für unsere Arbeit besonders relevanten Bereichen Projektmanagement (Lee und Miller 2004, Lin et al. 1997, Padberg 2006, Pfahl und Lebsanft 2000), Risikomanagement (Houston et al. 2001, Neu et al. 2002, Pfahl 2005), Requirements Engineering (Christie und Staley 2000, Ferreira et al. 2003, Höst et al. 2001, Lerch et al. 1997, Stallinger und Grünbacher 2001) sowie Softwareentwicklungsprozesse (Kuppaswami et al. 2003, Mišic et al. 2004, Powell et al. 1999).

Es herrscht in der Literatur über Softwareentwicklungsprozesse der Konsens, dass es trotz Kenntnissen über den Prozess in Form von Prozessaktivitäten, Artefakten, Rollen und ihren Beziehungen schwierig ist, das tatsächliche Verhalten des Prozesses im Einsatz vorherzusagen (Müller und Pfahl 2008). Eine Möglichkeit, den Prozess zu analysieren ist es diesen in einer Fallstudie anzuwenden. Dies bedeutet, dass man ein konkretes Softwareentwicklungsprojekt nach dem zu untersuchenden Prozess ablaufen lässt und den Einfluss des Prozesses auf den Projekterfolg misst und bewertet. Diese Variante ist jedoch sehr aufwändig. Zudem können die Resultate einer bestimmten Fallstudie nicht unbedingt auf andere Kontexte übertragen werden (Müller und Pfahl 2008). In diesem Kontext haben

Simulationsmodelle ihre eigene Berechtigung, da sie zum einen sehr billig in der Anwendung sind und zum anderen ebenfalls helfen können mentale Modelle, die dem Prozess zugrunde liegen, zu visualisieren, zu quantifizieren und zu überprüfen. Zudem ist es sehr leicht möglich Probleme im Prozess zu untersuchen und die Wirksamkeit korrigierender Anpassungen experimentell zu überprüfen, ehe sie real umgesetzt werden (Müller und Pfahl 2008). Simulationsmodelle stellen dabei meist nur Teilprozesse dar. Dies können aktuell implementierte oder auch nur geplante Prozesse sein. Es wird dabei jedoch stets vom Prozess abstrahiert, es werden also nur ausgewählte Aspekte des Prozesses abgebildet (Kellner et al. 1999).

Simulationsmodelle sind gut geeignet um bestimmte Komplexitätsfaktoren bei der Analyse von Softwareentwicklungsprozessen zu beherrschen (Kellner et al. 1999):

- *Unsicherheiten im System sowie stochastische Elemente:* Risiken müssen in der Softwareentwicklung bedacht werden. Sie stellen Unsicherheitsfaktoren dar, deren mögliche Auswirkungen verstanden und evaluiert werden sollten. Simulationsmodelle ermöglichen eine geeignete Abbildung von Unsicherheiten in komplexen Systemen.
- *Dynamisches Verhalten:* Das Verhalten eines Entwicklungsprozesses kann sich im Laufe der Zeit ändern. So können sich beispielsweise bestimmte Variablen wie Produktivität und Fehlerentdeckungsrate ändern. Mittels dynamischer Simulationsmodelle kann diese Variabilität ausgedrückt werden.
- *Feedbackmechanismen:* Veränderungen bestimmter Variablen an einem Punkt im System können komplexe Auswirkungen auf das restliche System haben, die wiederum in einer Wirkungskette die Ausgangsvariablen beeinflussen. Simulationsmodelle erlauben es solche Feedbackmechanismen darzustellen.

Simuliert man physikalische Prozesse, so kann das Modell auf physikalische Gesetze abgestützt werden. Im Falle der Simulation von Softwareentwicklungsprozessen müssen jedoch menschliche Faktoren und andere schwer greifbare Einflussgrößen berücksichtigt werden. Dies erschwert die Datensammlung. Simulationen erzeugen jedoch nur verlässliche Ergebnisse wenn die Eingabedaten für das Simulationsmodell die Wirklichkeit näherungsweise korrekt wiedergeben. Unter Eingabedaten sind nicht nur einzelne Parameter sondern auch Beziehungen zwischen diesen zu verstehen. Um eine möglichst getreue Wiedergabe der Wirklichkeit zu erreichen werden Expertenschätzungen der Eingabedaten benötigt. Verifikation und Validierung sind bei der Erstellung von Simulationsmodellen zudem besonders wichtige Aspekte (Robertson 1997).

Simulationsmethoden lassen sich in vier Dimensionen kategorisieren:

1. Deterministische vs. stochastische Simulation
2. Statische vs. dynamische Simulation
3. Kontinuierliche vs. ereignisgesteuerte Simulation
4. Quantitative vs. qualitative Simulation

Simulationsmodelle, die probabilistische Elemente enthalten, nennt man stochastische Simulationsmodelle. Im Gegenzug dazu enthalten deterministische Simulationsmodelle keine probabilistischen Elemente. Bei deterministischen Simulationsmodellen ergeben sich für ein bestimmtes Set an Eingangsparametern immer dieselben Ergebnisse, stochastische Simulationsmodelle hingegen produzieren bei verschiedenen Simulationsläufen unterschiedliche Ergebnisse.

Statische Simulationsmodelle berechnen die Modellparameter für einen bestimmten Zeitpunkt während Dynamische Simulationsmodelle die zeitliche Veränderung der Modellparameter über eine spezifizierte Zeitspanne darstellen.

Kontinuierliche Simulationsmodelle berechnen die Modellparameter in äquidistanten Zeitschritten neu. Bei ereignisgesteuerten Simulationsmodellen erfolgt die Neuberechnung der Modellparameter beim Eintreten bestimmter Ereignisse.

Bei quantitativen Simulationsmodellen werden die Modellparameter als Zahlwerte erfasst während dies bei qualitativen Simulationsmodellen nicht der Fall ist.

Es gibt auch hybride Simulationsmodelle, die deterministische mit stochastischen Elementen oder kontinuierliche mit ereignisgesteuerten Elementen kombinieren. Ein Beispiel dafür ist das Modell von Martin und Raffo (2001).

3.2 Systems Dynamics

3.2.1 Einführung in System Dynamics

Es existieren mehrere Definitionen zum Begriff System Dynamics. Wir verwenden die Definition von Coyle (1996):

„System dynamics deals with the time-dependent behaviour of managed systems with the aim of describing the system and understanding, through qualitative and quantitative models, how information feedback governs its behaviour, and designing robust information feedback structures and control policies through simulation and optimization.“

System Dynamics ist sowohl ein qualitativer als auch quantitativer kontinuierlicher und dynamischer Simulationsansatz. System Dynamics Modelle bestehen aus mathematischen Gleichungen, die Zeitverhalten in Differentialgleichungen erster Ordnung ausdrücken. Zu den mathematischen Gleichungen existieren graphische Repräsentationen, sogenannte Flussgraphen. System Dynamics Modelle bestehen aus Netzwerken von quantitativ modellierten Wirkbeziehungen. In diesen Netzwerken lassen sich direkte und indirekte Wirkungsketten darstellen. Eine Einführung in die mathematischen Grundlagen von System Dynamics findet sich bei Forrester (1971).

System Dynamics wurde in den späten 1950ern am MIT entwickelt und diente zunächst der Lösung industrieller Probleme. Zu diesem Zeitpunkt war die Methode noch unter dem Namen „Industrial Dynamics“ (Forrester 1961) bekannt. Die Methode wurde schließlich in weiteren Domänen eingesetzt und gelangte so zu ihrem heutigen Namen „System Dynamics“. Roberts (1964) beschreibt Studien in den Domänen Fertigung, Forschung und Entwicklung, Management und Finanzen, Marketing und Vertrieb, Ökologie, Ökonomie und Soziologie. In den 1990ern fand System Dynamics schließlich auch Anwendung im Software Engineering. Pfahl (2001) identifiziert folgende Gebiete im Software Engineering, in denen System Dynamics eingesetzt wurde:

- Projektmanagement
- Parallele Softwareentwicklung
- Requirements Engineering
- Auswirkungen von Prozessverbesserungen auf die Entwicklungszeit
- Auswirkungen von Qualitätssicherungsmaßnahmen

- Reliability Management
- Wartung
- Evolution
- Outsourcing
- Training

Im Folgenden stellen wir dar, wie man bei der Entwicklung von System Dynamics Modellen vorgeht.

3.2.2 Vorgehen in der Entwicklung von System Dynamics Modellen

Das Vorgehen zur Entwicklung von System Dynamics Modellen wurde von Forrester (1961) aufgestellt. Dieses wurde im Laufe der Zeit von verschiedenen Autoren weiterentwickelt. Eine Einführung in System Dynamics findet sich in Müller und Pfahl (2008). Die grundlegenden Schritte bei der Entwicklung von System Dynamics Modellen sind die folgenden (Pfahl 2001):

1. Problembeschreibung
2. Definition des Referenzverhaltens
3. Identifikation der Grundmechanismen
4. Erstellung des Ursache-Wirkungs-Netzwerks
5. Aufstellen der mathematischen Gleichungen sowie Erstellung des Flussgraphen
6. Kalibrierung des Modells
7. Verifikation und Validierung des Modells
8. Strategieanalyse basierend auf der Simulation

In der **Problembeschreibung** wird formuliert welche Fragen mit dem zu entwickelnden Simulationsmodell beantwortet werden sollen.

Die **Definition des Referenzverhaltens** beschreibt explizit das in der Realität beobachtete dynamische, d.h. zeitabhängige, Verhalten eines oder mehrerer Parameter des untersuchten Systems. Das Referenzverhalten stellt bereits ein initiales Modell dar. Das Referenzverhalten kann entweder ein beobachtetes, unter Umständen problematisches Verhalten, das untersucht und verbessert werden soll oder auch ein angestrebtes Verhalten sein. Die Definition des Referenzverhaltens hilft zum einen dabei, wichtige Ausgabeparameter des Simulationsmodells zu bestimmen und zum anderen ist es von essenzieller Bedeutung für die Validierung des Modells.

Die sogenannten **Grundmechanismen** bezeichnen die grundlegenden Ursache-Wirkungs-Beziehungen, die das Verhalten der zu untersuchenden Parameter beeinflussen. Die Ursache-Wirkungs-Beziehungen müssen normalerweise in Interviews von Experten in Erfahrung gebracht werden. Die Grundmechanismen werden in Form binärer Beziehungen dokumentiert. Es gibt zwei Arten von Relationen:

- Positive Auswirkung: Ursache (+) → Wirkung (+) oder Ursache (-) → Wirkung (-)
- Negative Auswirkung: Ursache (+) → Wirkung (-) oder Ursache (-) → Wirkung (+)

Bei der positiven Auswirkung führt eine Zunahme der Variable „Ursache“ zu einer Zunahme der Variable „Wirkung“. Entsprechend führt eine Abnahme der Variable „Ursache“ zu einer Abnahme der Variable „Wirkung“. Im Falle der negativen Auswirkung bringt eine Zunahme der Variable „Ursache“ zu einer Abnahme der Variable „Wirkung“ und eine Abnahme der Variable „Ursache“ zu einer Zunahme der Variable „Wirkung“

In einem **Ursache-Wirkungs-Netzwerk** werden alle Grundmechanismen zusammengeführt und als Graph dargestellt. Über die Kombination von Grundmechanismen können sich Wirkungsketten, die sich selbst verstärken, und Wirkungsketten, die eine Dämpfung bewirken, entstehen. Man spricht in diesem Zusammenhang von positiver Polarität beziehungsweise negativer Polarität der Wirkungsketten. Ein einfaches Beispiel für ein Ursache-Wirkungs-Netzwerk zeigt Abbildung 3-1. Darin finden sich eine positive und eine negative Wirkungskette:

- Positive Wirkungskette: Aufwand für Inspektionen (+) → Fehler im System (-) → Änderungen am System (-) → Benötigte Zeit (-) → Aufwand für Inspektionen (+)
- Negative Wirkungskette: Aufwand für Inspektionen (+) → Benötigte Zeit (+) → Zeitdruck (+) → Aufwand für Inspektionen (-)

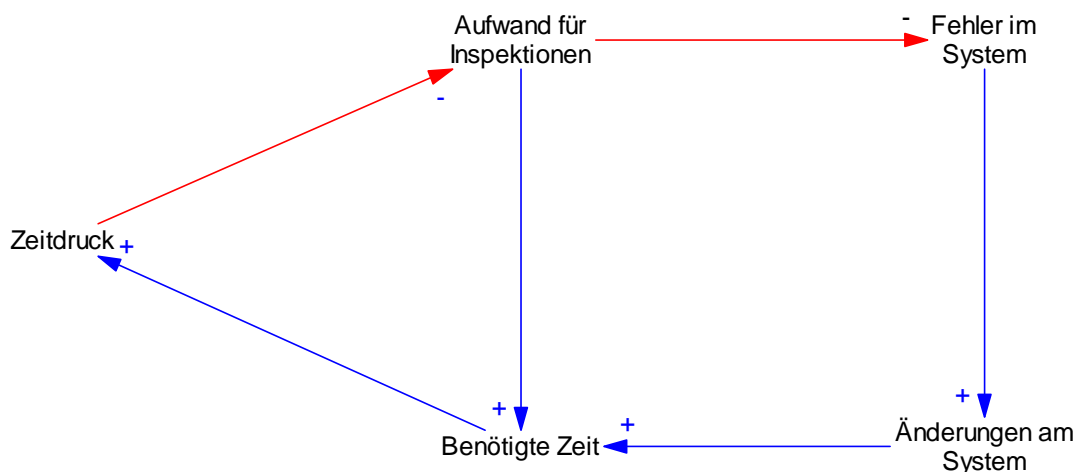


Abbildung 3-1: Beispiel für ein Ursache-Wirkungs-Netzwerk

Das Ursache-Wirkungs-Netzwerk wird in ein formales Modell bestehend aus mathematischen Gleichungen transformiert. Ein System Dynamics Modell wird durch eine Menge von Differenzgleichungen beschrieben. Variablen, die den Zustand des Modells wiedergeben, werden Levels genannt. Sie werden durch Gleichungen der folgenden Form beschrieben:

$$Level(t + dt) = Level(t) + \sum_{k=1}^m \text{Eingangsrate}_k(t) - \sum_{k=1}^n \text{Ausgangsrate}_k(t)$$

Beschrieben ist ein Level mit m Eingangsraten ($\text{Eingangsrate}_1, \dots, \text{Eingangsrate}_m$) und n Ausgangsraten ($\text{Ausgangsrate}_1, \dots, \text{Ausgangsrate}_n$); dt bezeichnet einen Zeitschritt von einem diskreten Zeitpunkt zum nächsten.

Die Initialisierung der Level erfolgt zu Beginn der Simulation. Raten geben die Änderung von Levels an. Wie Levels werden sie ebenfalls durch Gleichungen dargestellt. Raten können von Levels, Konstanten oder Hilfsvariablen abhängen. Letztere stellen Zwischenrechnungen dar, die komplexere Berechnungen in Einzelschritte aufbrechen, um das Modell verständlich zu halten. Eine Gleichung, die eine Rate beschreibt, sieht beispielsweise wie folgt aus:

$$\text{Arbeitsrate}(t) = \text{IF THEN ELSE} (\text{UnerledigteArbeit}(t) > 0, \text{Entwicklerzahl}(t) * \text{ProduktivitätProPerson}, 0) \\ \sim \text{Person}$$

Die verwendete Notation entspricht der des System Dynamics Modellierungstools Vensim 3.0 (Ventana Systems 1997). Die Gleichung beschreibt die Rate *Arbeitsrate*, die in der Einheit *Person*Tag/Tag* als *Person* gemessen wird. Diese ist abhängig vom Level *UnerledigteArbeit*, dem Level *Entwicklerzahl* und der Konstante *ProduktivitätProPerson*. Solange noch Arbeit zu erledigen ist (Level *UnerledigteArbeit* > 0) entspricht die pro Zeitschritt vollführte Arbeit dem Produkt aus der Zahl der zum jeweiligen Zeitpunkt verfügbaren Entwickler (Level *Entwicklerzahl*) und der durchschnittlichen Produktivität pro Person (Konstante *ProduktivitätProPerson*). Ist keine unerledigte Arbeit mehr vorhanden, so ist *Arbeitsrate* gleich Null. Bei der Gleichung handelt sich um ein fiktives und vereinfachtes Beispiel. So wird beispielsweise der Fall nicht berücksichtigt, dass die Arbeitsrate niedriger sein muss, wenn nicht mehr genügend unerledigte Arbeit vorhanden ist, um das ganze Entwicklerteam für den nächsten Zeitschritt auszulasten. Die Gleichung entstammt nicht dem in Kapitel 3.4 und 3.5 entwickelten System Dynamics Modell.

System Dynamics Modelle werden mittels Flussgraphen dargestellt. Abbildung 3-2 zeigt ein Beispiel für einen Flussgraphen. In diesem ist ein einfacher Prozess dargestellt, in dem ein Level, das eine Menge verbleibender Arbeit (Work Remaining) darstellt, gemäß einer Rate (work flow) reduziert wird und ein Level, das eine Menge erledigter Arbeit (Work Accomplished) darstellt, gemäß der selben Rate erhöht wird. Die Rate „work flow“ stellt hierbei die Ausgangsrate für „Work Remaining“ und die Eingangsrate für „Work Accomplished“ dar. Die Konstante „initial project definition“ gibt den Anfangswert von „Work Remaining“ vor. Der Anfangswert von „Work Accomplished“ ist Null.

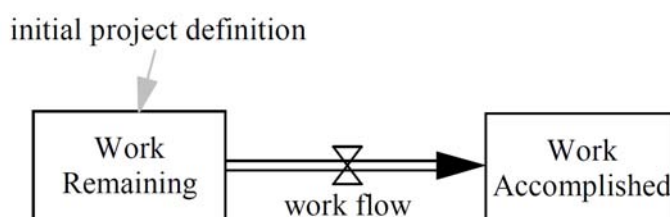


Abbildung 3-2: Beispiel für einen Flussgraph (aus Ventana Systems 1997)

In der **Modellkalibrierung** werden initiale Werte für die Levels bestimmt, sowie Konstanten und Parameter festgelegt.

Die **Verifikation** des Modells geschieht durch direkte und indirekte verhaltensorientierte strukturelle Tests. Direkte strukturelle Tests überprüfen die korrekte Übersetzung der kausalen Beziehungen aus dem Ursache-Wirkungs-Netzwerk in den Flussgraphen. Sie

überprüfen zudem die Konsistenz der Einheiten der Modellgrößen sowie die Bedeutung der Modellgleichungen für Grenzwerte. Indirekte strukturelle Tests überprüfen das Verhalten des Modells bei Änderung der Parameter insbesondere auch für Extremwerte. Mittels einer Sensitivitätsanalyse (Tank-Nielsen 1980) kann festgestellt werden, wie groß der Einfluss der einzelnen Parameter auf ein bestimmtes Verhalten ist. Sie ist insbesondere dafür geeignet, um die Stellen zu identifizieren, bei denen alternative Strategien am besten greifen. Für die **Validierung** des Modells werden direkte strukturelle Tests und Verhaltenstests verwendet. Direkte strukturelle Tests werden eingesetzt um zu überprüfen ob alle wesentlichen kausalen Zusammenhänge im Modell aufgenommen wurden, die Modellvariablen Größen aus der Realität entsprechen und die Modellparameter korrekt eingestellt wurden (Pfahl 2001). Verhaltenstests werden angewendet, um zu überprüfen ob das Referenzverhalten ausreichend gut mit dem Modell reproduziert werden kann, ob unerwartete Verhaltensmuster auftreten, ob es zu Verhaltensanomalien kommt wenn einzelne Wirkbeziehungen ausgenommen werden und ob Modellverbesserungen unerwartete Verhaltensmuster bewirken (Pfahl 2001). Zudem versucht man die generelle Nützlichkeit des Modells zu bewerten. Darunter fällt, die angemessene Granularität und Modularität des Modells einzuschätzen. Es wird weiterhin bewertet, ob das Modell neue Einsichten bringt, ob es dazu geeignet ist unerwartetes Verhalten zu erklären und ob es generalisiert werden kann.

Basierend auf der Simulation kann schließlich eine **Strategieanalyse** durchgeführt werden. Dabei wird untersucht, warum bestimmte Strategien zu dem simulierten Verhalten führen. Es werden zudem Strategieänderungen untersucht, die eine Verbesserung des Verhaltens hinsichtlich der in der Problembeschreibung festgehaltenen Probleme bewirken. Strategieänderungen können entweder durch quantitative Änderung der Modellparameter bewirkt werden oder durch Manipulation der Modellstruktur.

Strategieanalyse, Validierung und Modellkalibrierung gehen in der Regel Hand in Hand (Müller und Pfahl 2008). Detaillierte Vorgehensbeschreibungen zur Entwicklung von System Dynamics Modellen finden sich bei Forrester (1961, 1971), Roberts (1964), Randers (1973, 1980), Richardson und Pugh (1981), Bossel (1992), Coyle (1996), Pfahl (2001), Pfahl und Ruhe (2003) sowie Müller und Pfahl (2008).

3.3 Verwandte Arbeiten

In diesem Kapitel stellen wir im Überblick einige System Dynamics Modelle vor, die von besonderer Bedeutung für die Themenstellung dieser Arbeit sind. Zudem grenzen wir unsere Arbeit von den vorgestellten Modellen ab.

3.3.1 Abdel-Hamid und Madnick

Abdel-Hamid und Madnick (1991) entwickelten als erste ein System Dynamics Modell, das ein Softwareentwicklungsprojekt simuliert. Ihre Arbeit basiert auf einer umfassenden Analyse von kleinen und mittleren Softwareentwicklungsprojekten der NASA in den 1980ern. Ihr Modell umfasst vier verschiedene Bereiche: Human Resource Management, Software Production, Controlling und Planning. Betrachtet wird nur die Entwicklungsphase, d.h. RE und Wartung werden ausgeblendet. Zudem nehmen die Autoren stabile Anforderungen an. Die Ausgabeparameter des Modells zeigen das Verhalten zentraler Größen für das Management an: Projektabschlusstermin, geschätzte Kosten und Personaleinsatz. Abdel-Hamid und Madnick untersuchten mit ihrem Modell

die Auswirkungen verschiedener Managementstrategien zum Beispiel in der Kostenschätzung.

Das Modell von Abdel-Hamid und Madnick ist als erstes System Dynamics Modell eines Softwareentwicklungsprojekts der Wegbereiter für nachfolgende Arbeiten. Es ist jedoch nicht geeignet um unsere Forschungsfragen zu untersuchen, da es RE völlig ausspart und von stabilen Anforderungen ausgeht.

3.3.2 Lin et al.

Das von Lin et al. (1997) entwickelte Modell bricht den Softwareentwicklungsprozess in die Einzelaktivitäten Requirements, Preliminary Design, Detailed Design, Build-Integration-Test und System Test auf, was eine Analyse des Zeit- und Kostenplans auf einer detaillierteren Ebene ermöglicht. Das Modell ist als Planungswerkzeug gedacht, mit dem Kosten, Zeit und Funktionalität gegeneinander abgewogen und die Auswirkungen verschiedener Managementstrategien analysiert werden können. Es bezieht dabei die Möglichkeit ein, dass sich Anforderungen während des Projekts ändern. Lin et al. untersuchen eine Reihe von Parametern. Zum einen analysieren sie die Auswirkung von Zeitdruck auf die Produktivität und Projektkosten. Zum anderen simulieren sie welche Bedeutung die Teamzusammensetzung für die Fehlerrate, die Projektlaufzeit, und die Projektkosten hat. Darüber hinaus analysieren sie die Auswirkungen von Projektgröße und Umfang der Qualitätssicherungsmaßnahmen auf die Projektlaufzeit und die Projektkosten.

Lin et al. beziehen die Instabilität von Anforderungen mit ein. Die entsprechenden Modellelemente finden sich auch in unserem Modell wieder. Das Modell von Lin et al. lässt sich jedoch nicht direkt auf unsere Forschungsfragen anwenden, da eine Variierung des Aufwands im RE sowie deren Auswirkungen auf die Änderungsrate nicht abgebildet werden kann. Auch eine gesonderte Betrachtung von unterschiedlichen Änderungsraten der Anforderungen ist nicht einfach möglich.

3.3.3 Powell et al.

Powell et al. (1999) untersuchten bei Rolls Royce plc mittels eines System Dynamics Modells im Kontext von TTM die Möglichkeit, die Entwicklung von eingebetteten Systemen mittels Parallelisierung und inkrementellem Vorgehen zu beschleunigen. Sie kamen zu dem Schluss, dass durch diese Strategien sowohl der Zeitplan verkürzt als auch die Produktqualität erhöht und potenziell die Entwicklungskosten gesenkt werden können. Der kürzere Zeitplan entsteht laut Powell et al. durch eine Beschleunigung der Entwicklung und Kompression des kritischen Pfades in der Entwicklung. Die reduzierten Kosten führen sie auf die frühe Entdeckung und Behebung von Problemen in der Entwicklung zurück. Qualitätsgewinne schließlich werden durch frühere, längere und dadurch gründlichere Tests möglich. Powell et al. verwenden ihr Modell dazu, festzustellen welcher Grad von Parallelität und welche Zahl von Iterationen nötig ist, um verfügbare Ressourcen effektiv einzusetzen und realistische Überlappzeiten zu gewährleisten, so dass Arbeitsergebnisse reifen können ehe sie als Input für nachfolgende Aktivitäten gebraucht werden. Die Autoren gehen davon aus, dass durch einen stärkeren Überlapp der einzelnen Aktivitäten die notwendig werdenden Änderungen zunehmen. Sie bestimmen mittels ihres Modells eine optimale Größe des Überlapps.

Das Modell von Powell et al. lässt sich nicht direkt auf unsere Forschungsfragen anwenden. RE liegt nicht spezifisch im Fokus des Modells. Es lässt nicht die getrennte Betrachtung verschiedener Anforderungstypen sowie die Aufwandsreduktion im RE zu.

Da wir jedoch gerade diese Strategien sowie deren Auswirkungen auf Änderungsrate und Projektlaufzeit untersuchen, lässt sich das Modell für unsere Zwecke nicht einsetzen.

3.3.4 Ferreira et al. und Houston et al.

Das Modell von Ferreira et al. (2003) baut auf dem von Houston et al. (2001) auf. Beide Modelle untersuchen die Auswirkungen von instabilen Anforderungen in Softwareentwicklungsprojekten. Die Modelle basieren auf Daten, die aus einer Umfrage bei Projektmanagern gewonnen wurden. Die Untersuchungen zeigen, dass eine höhere Instabilität der Anforderungen zu zusätzlichem Gesamtaufwand und zusätzlichen Änderungen führt. Ferreira et al. zeigen, dass die Instabilität der Anforderungen außerdem zu einer geringeren Motivation des Personals führt.

Die Ergebnisse von Ferreira et al. sowie die von Houston et al. stützen unsere Annahmen bezüglich des zusätzlich nötigen Aufwands bei gesteigerter Instabilität der Anforderungen. Mit ihren Modellen kann allerdings nicht untersucht werden inwieweit geänderter Aufwand für RE sich auf Änderungsrate beziehungsweise Projektlaufzeit und –kosten auswirken.

3.3.5 Pfahl und Lebsanft

Das Modell von Pfahl und Lebsanft (2000) untersucht wie sich die Instabilität von Anforderungen auf die Projektkosten auswirkt. Darüber hinaus analysieren die Autoren die Kosteneffizienz von Maßnahmen zur Stabilisierung der Anforderungen. Das Modell basiert auf Daten, die in einer Entwicklungsabteilung für eingebettete Systeme in der Siemens AG erhoben wurden. Es wurde angenommen, dass die Instabilität der Anforderungen in der Softwareentwicklung vom Umfang der Qualitätssicherungsmaßnahmen im Systems Engineering abhängt. In mehreren Experimenten wurde der Aufwand für Qualitätssicherung im Systems Engineering variiert und die Auswirkungen auf die Gesamtprojektkosten beobachtet.

Pfahl und Lebsanft leisten für unseren Ansatz wichtige Vorarbeiten indem sie die Auswirkungen von Stabilisierungsmaßnahmen für Anforderungen auf die Projektkosten analysieren. Die entsprechenden Modellelemente dienen als Grundlage für die Entwicklung unseres Modells. Unser Modell führt zusätzlich die getrennte Betrachtung verschiedener Anforderungstypen ein. Zudem betrachten wir keine Systementwicklung sondern reine Softwareentwicklung. Die Stabilisierung der Anforderungen erfolgt in unserem Modell durch den Aufwand für RE.

3.4 Fallstudie 1: FAUSER AG

In Zusammenarbeit mit der FAUSER AG entwickelten wir die erste Version unseres System Dynamics Modells.

„Die FAUSER AG ist als Softwarehersteller und Lösungsanbieter für kleine und mittelständische Industrieunternehmen seit mehr als 10 Jahren am Softwaremarkt vertreten und entwickelt, vertreibt und wartet an drei Standorten in Deutschland Softwarelösungen für die integrierte Auftragsabwicklung, echtzeitfähige Fertigungsfeinplanung sowie effiziente Betriebsdatenerfassung.“ (Broy et al. 2010)

3.4.1 Entwicklung des System Dynamics Modells

Bei der Entwicklung des Simulationsmodells folgten wir dem in Kapitel 3.2.2 beschriebenen Vorgehen.

Die Datenerhebung erfolgte in Workshops mit Mitarbeitern der FAUSER AG. Teilnahmen ein Chefentwickler sowie ein Projektmanager der zudem für Kundenkontakte und Vertrieb zuständig ist. Dieselben Experten wurden an der Verifikation und Validierung des Modells beteiligt.

Die erhobenen Daten beziehen sich auf ein konkretes Beispielprojekt. Die meisten Werte stellen Einschätzungen im Nachgang des Projekts dar und basieren nicht auf dokumentierten Zahlen. An dem Projekt waren vier Personen beteiligt. Das Entwicklerteam in dem Beispielprojekt bestand aus einem einzigen Entwickler. Am RE wurden neben dem Entwickler drei weitere Personen beteiligt. Für die Simulation betrachten wir den Projektverlauf beginnend ab dem Projekt-Kick-Off. Der Verlauf gliederte sich in drei Phasen. Zunächst erfolgte eine initiale RE-Phase in der die Anforderungen ermittelt und analysiert wurden. In der folgenden Entwicklungsphase wurden die Anforderungen umgesetzt. Dabei existierte keine klare zeitliche Trennung von Entwurf, Implementierung und Test. Zudem wird die Integration kontinuierlich betrieben. Am Ende des Projekts erfolgte eine abschließende RE-Phase, in der Reviews durchgeführt und noch einmal viele Änderungswünsche eingebracht und umgesetzt wurden.

Für die Simulation wurde angenommen, dass die Kosten für die Durchführung von Änderungen an bereits realisierten Anforderungen fünfmal höher sind, als für deren Erstrealisierung. Dies beruht auf der Einschätzung der Experten der FAUSER AG und entspricht zudem genau dem Wert den Boehm und Basili (2001) bei der Entwicklung von einfachen nicht kritischen Systemen annehmen.

Für die Erstellung des Simulationsmodells und dessen Ausführung wurde das Werkzeug Vensim PLE von Ventana Systems (Ventana Systems 1997) eingesetzt.

Problembeschreibung

Folgende Fragen wollen wir mittels des System Dynamics Modells beantworten:

1. Welche Auswirkungen hat eine Variierung des Aufwands für RE auf die Änderungsrate?
2. Können durch eine geeignete Konzentration des RE-Aufwands auf bestimmte Anforderungstypen Vorteile hinsichtlich Projektlaufzeit und –kosten erlangt werden? Wir unterscheiden bei dieser Frage zwischen den idealisierten Anforderungstypen stabile, stabilisierbare und nicht stabilisierbare Anforderungen.
3. Können durch Variierung des Aufwands für RE Optima in Projektlaufzeit und Projektkosten erreicht werden?
4. Welche Auswirkungen hat die Stabilisierbarkeit von Anforderungen auf die, sich durch Variierung des RE-Aufwands einstellenden, Optima in Projektlaufzeit und Projektkosten? Wir betrachten hierbei die Stabilisierbarkeit der Anforderungen durch den Verständnisgewinn in der Anforderungsanalyse.
5. Welche Auswirkungen hat die Änderungsrate von Anforderungen auf die, sich durch Variierung des RE-Aufwands einstellenden, Optima in Projektlaufzeit und Projektkosten?

Definition des Referenzverhaltens

Als Referenzverhalten wurden die im Beispielprojekt beobachtete Menge der realisierten Anforderungen sowie die Menge der hinzukommenden oder geänderten Anforderungen pro Zeitspanne jeweils im Projektverlauf ausgewählt. Mit dem Referenzverhalten wurden zum einen zwei wichtige Ausgabeparameter des Simulationsmodells bestimmt und zum anderen ist es von essenzieller Bedeutung für die Validierung des Modells.

Unter der Menge der realisierten Anforderungen verstehen wir nicht die Anzahl der realisierten Anforderungen, sondern den mit ihrer Erstrealisierung verbundenen Aufwand in Personentagen. Die bloße Anzahl an Anforderungen ist kein geeignetes Maß, da sich unmittelbar die Frage nach der Granularität der Anforderungen stellen würde und mit verschiedenen Anforderungen ganz unterschiedlich große Aufwände verbunden sein können. Wir orientieren uns an dem System Dynamics Modell von Pfahl und Lebsaft (2000). In diesem wurden Mengen von Anforderungen und Anforderungsänderungen durch den mit diesen verbundenen Entwicklungsaufwand gemessen. Anhand dieses Maßes lässt sich gut darstellen, welcher Anteil des Systems bereits realisiert ist und welcher Anteil des Systems von Änderungen betroffen ist. Da bei der FAUSER AG keine zeitliche Trennung zwischen Entwurf, Entwicklung und Test besteht und die Integration kontinuierlich durchgeführt wird, wird eine Anforderung genau dann als realisiert verstanden, wenn sie implementiert, integriert und getestet wurde.

Die Menge der realisierten Anforderungen ist Null während der initialen RE-Phase, die ca. ein Achtel der gesamten Projektlaufzeit umfasst, steigt bei Entwicklungsbeginn an und flacht zunehmend ab, verläuft während der Entwicklung also logarithmisch (Abbildung 3-3). Der logarithmische Verlauf ergibt sich daraus, dass der Aufwand für Änderungen im Laufe des Projektes immer höher wird und somit die Entwicklung langsamer voranschreitet. Die Kurve stellt nur eine grobe Näherung dar. Entscheidend ist der qualitative Verlauf der Kurve. Die Referenzkurve stellt einen Input seitens der Experten der FAUSER AG dar. Diese schätzten ihren Verlauf im Nachgang des Projektes ein. Sie basiert nicht auf dokumentierten Zahlen sondern stellt eine intuitive Einschätzung dar. Über ihren qualitativen Verlauf herrscht große Sicherheit bei den Experten der FAUSER AG. In der Validierung des Simulationsmodells ist zu überprüfen, ob der qualitative Verlauf der Referenzkurve korrekt wiedergegeben wird.

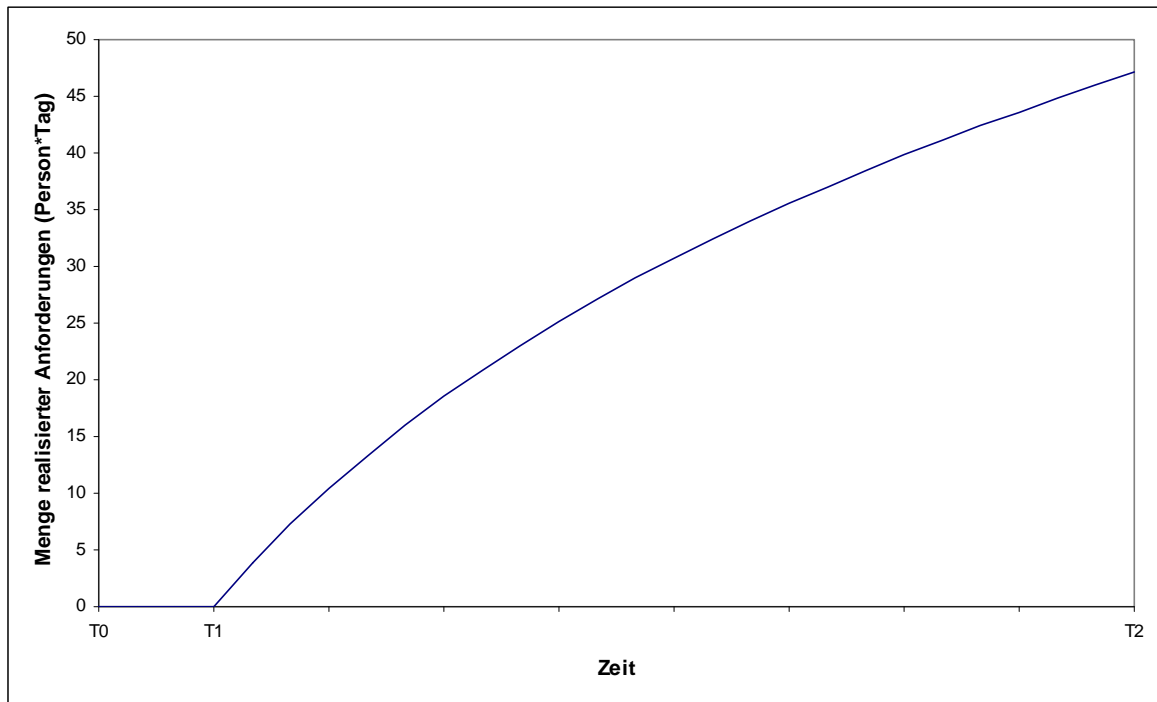


Abbildung 3-3: Referenzkurve für die Menge realisierter Anforderungen im Beispielprojekt bei der FAUSER AG (die Menge der Anforderungen wird in Form des für sie tatsächlich notwendigen Erstrealisierungsaufwands gemessen)

T0: Projekt-Kick-Off und Beginn der initialen RE-Phase

T1: Ende der initialen RE-Phase und Entwicklungsbeginn (ca. nach Ablauf von einem Achtel der betrachteten Zeitspanne)

T2: Entwicklungsende

Die zweite Referenzkurve gibt die Menge der hinzukommenden oder geänderten Anforderungen pro Zeitspanne wieder. Die von uns verwendete Maßzahl für eine Menge von Anforderungen ist, wie bereits bei der ersten Referenzkurve erläutert, der mit ihrer Erstrealisierung verbundene Aufwand in Personentagen. Dementsprechend ergibt sich im Modell für die Menge der hinzukommenden oder geänderten Anforderungen pro Zeitspanne die Einheit *Personentag/Tag*, also *Person*.

Der Menge der hinzukommenden oder geänderten Anforderungen ist während der Anforderungsanalyse, die ca. ein Achtel der gesamten Projektlaufzeit umfasst, sehr hoch, fällt dann ab, durchläuft ein Minimum und steigt schließlich wieder an. Am Ende des Projekts erfolgt im Beispielprojekt eine abschließende Phase der Anforderungsanalyse. Diese umfasst ebenfalls in etwa ein Achtel der gesamten Projektlaufzeit. Während dieser Phase ist die Menge der hinzukommenden oder geänderten Anforderungen noch einmal relativ hoch. Nach einer groben Schätzung treten 80-90% der hinzukommenden oder geänderten Anforderungen in der ersten Analysephase auf und 10-20% in der abschließenden (Abbildung 3-4).

Im Beispielprojekt der FAUSER AG wird die Annahme neuer Anforderungen oder Anforderungsänderungen nur beschlossen, während RE-Workshops mit den Stakeholdern abgehalten werden. Die hohe Anzahl der hinzukommenden oder geänderten Anforderungen zu Beginn und am Ende des Projekts erklärt sich dadurch, dass zu diesen Zeitpunkten Analysephasen stattfinden. Da zu Beginn des Projekts die Anforderungen entwickelt werden und zudem noch große Unsicherheit bezüglich der Anforderungen besteht, ist zu diesem Zeitpunkt die Menge der hinzukommenden oder geänderten

Anforderungen sehr groß. Der Verlauf der Kurve der hinzukommenden oder geänderten Anforderungen zwischen den Analysephasen leitet sich unmittelbar von der Frequenz von RE-Workshops ab, die zu Beginn und am Ende der Realisierungsphase relativ hoch dazwischen jedoch sehr niedrig ist. Wie auch die erste Referenzkurve stellt diese nur eine grobe Näherung dar, die auf der Abschätzung der Projektbeteiligten im Nachgang des Projektes basiert. Über den qualitativen Verlauf der Kurve besteht jedoch bei den Experten große Sicherheit. Verwunderlich ist der Verlauf der Referenzkurve in der abschließenden Analysephase, da bis zum Ende des Projektes noch Änderungen beschlossen werden und diese nicht mehr umgesetzt werden können. Die Kurve kann jedoch so interpretiert werden, dass die letzten Änderungen im nächsten Release umgesetzt werden. In der Validierung des Simulationsmodells ist zu überprüfen, ob der qualitative Verlauf der Referenzkurve korrekt wiedergegeben wird.

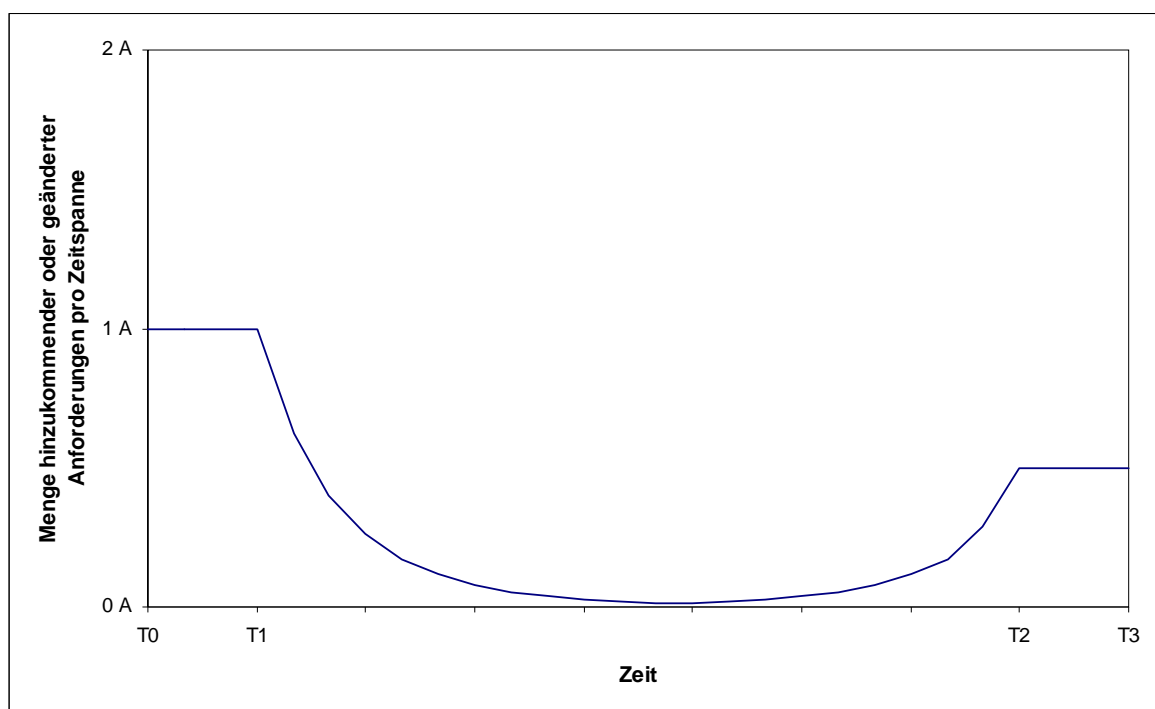


Abbildung 3-4: Referenzkurve für die Menge der hinzukommenden oder geänderten Anforderungen pro Zeitspanne im Beispielprojekt bei der FAUSER AG.

A: Menge der in der initialen RE-Phase hinzukommenden oder geänderten Anforderungen pro Zeitspanne (in der initialen RE-Phase treten in Summe 80-90% der hinzukommenden oder geänderten Anforderungen auf, in der abschließenden 10-20%)

T0: Projekt-Kick-Off und Beginn der initialen RE-Phase

T1: Ende der initialen RE-Phase und Entwicklungsbeginn (ca. nach Ablauf von einem Achtel der betrachteten Zeitspanne)

T2: Beginn der abschließenden Analysephase (ca. nach Ablauf von sieben Achtel der betrachteten Zeitspanne)

T3: Entwicklungsende

Identifikation der Grundmechanismen

Für die Erstellung des System Dynamics Modells mussten die wichtigsten Grundmechanismen identifiziert werden, die das typische Verhalten im Beispielprojekt erzeugen. Gemeinsam mit den Experten der FAUSER AG wurden folgende Grundmechanismen identifiziert:

1. Aufwand für die Analyse der stabilisierbaren Anforderungen (+) → Verständnis der stabilisierbaren Anforderungen (+)
2. Aufwand für die Analyse der stabilisierbaren Anforderungen (+) → Verständnis des Gesamtsystems (+)
3. Aufwand für die Analyse der stabilisierbaren Anforderungen (+) → Aufwand für RE (+)
4. Aufwand für die Analyse der stabilen Anforderungen (+) → Verständnis des Gesamtsystems (+)
5. Aufwand für die Analyse der stabilen Anforderungen (+) → Aufwand für RE (+)
6. Aufwand für die Analyse der nicht stabilisierbaren Anforderungen (+) → Verständnis des Gesamtsystems (+)
7. Aufwand für die Analyse der nicht stabilisierbaren Anforderungen (+) → Aufwand für RE (+)
8. Verständnis des Gesamtsystems (+) → Verständnis der stabilisierbaren Anforderungen (+)
9. Verständnis der stabilisierbaren Anforderungen (+) → Änderungsrate der stabilisierbaren Anforderungen (-)
10. Änderungsrate der stabilisierbaren Anforderungen (+) → Änderungen der stabilisierbaren Anforderungen (+)
11. Aufwand für RE (+) → Gesamtprojektkosten (+)
12. Aufwand für RE (+) → Projektdauer (+)
13. Aufwand für Entwicklung (+) → Gesamtprojektkosten (+)
14. Aufwand für Entwicklung (+) → Projektdauer (+)
15. Projektdauer (+) → Änderungen der stabilisierbaren Anforderungen (+)
16. Projektdauer (+) → Änderungen der nicht stabilisierbaren Anforderungen (+)
17. Änderungen der stabilisierbaren Anforderungen (+) → Aufwand für die Analyse der stabilisierbaren Anforderungen (+)
18. Änderungen der stabilisierbaren Anforderungen (+) → Aufwand für die Entwicklung (+)
19. Änderungen der nicht stabilisierbaren Anforderungen (+) → Aufwand für die Analyse der nicht stabilisierbaren Anforderungen (+)
20. Änderungen der nicht stabilisierbaren Anforderungen (+) → Aufwand für die Entwicklung (+)

Erstellung des Ursache-Wirkungs-Netzwerks

Die Grundmechanismen wurden in einem Ursache-Wirkungs-Netzwerks zusammengeführt (Abbildung 3-5).

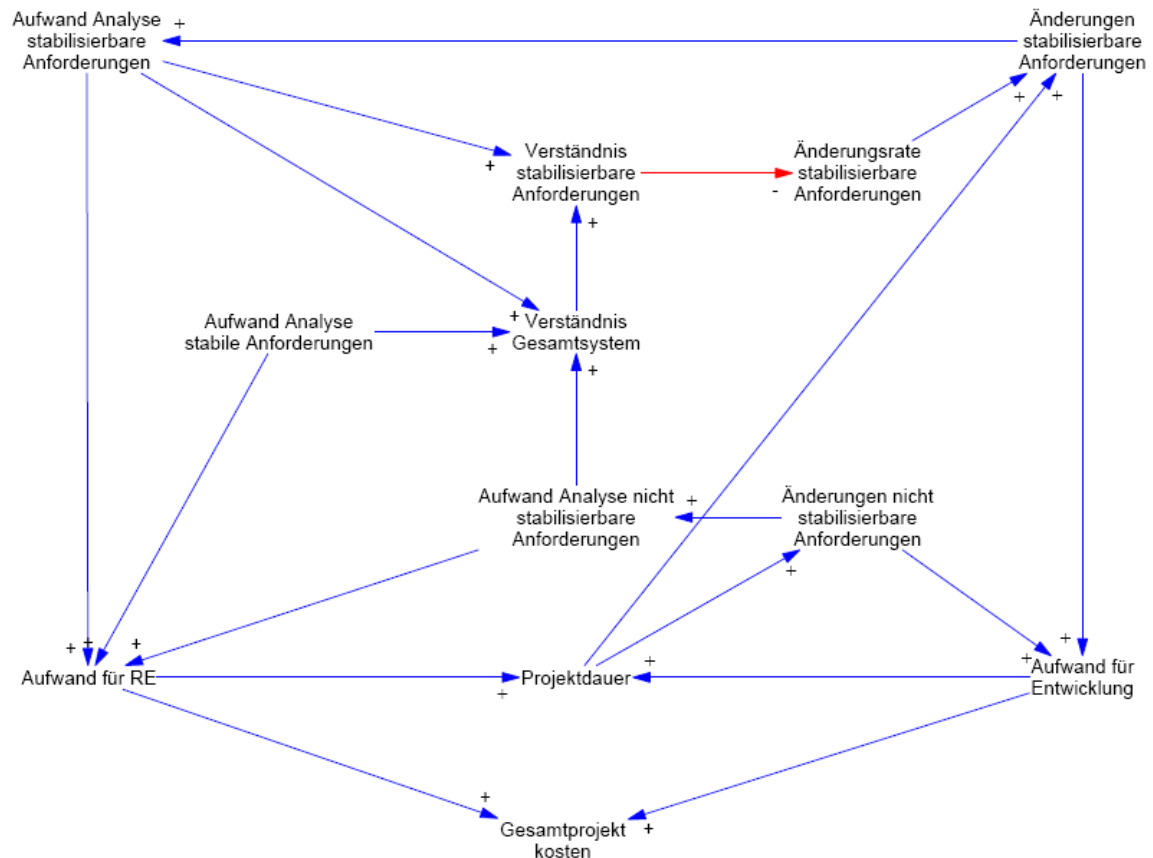


Abbildung 3-5: Ursache-Wirkungs-Netzwerk für die Fallstudie bei der FAUSER AG

Über die Kombination der Grundmechanismen ergeben sich folgende Wirkungsketten positiver Polarität:

1. Aufwand für die Analyse der stabilisierbaren Anforderungen (+) → Aufwand für RE (+) → Projektdauer (+) → Änderungen der stabilisierbaren Anforderungen (+) → Aufwand für die Analyse der stabilisierbaren Anforderungen (+)
2. Änderungen der nicht stabilisierbaren Anforderungen (+) → Aufwand für die Entwicklung (+) → Projektdauer (+) → Änderungen der nicht stabilisierbaren Anforderungen (+)
3. Änderungen der stabilisierbaren Anforderungen (+) → Aufwand für die Entwicklung (+) → Projektdauer (+) → Änderungen der stabilisierbaren Anforderungen (+)
4. Änderungen der nicht stabilisierbaren Anforderungen (+) → Aufwand für die Analyse der nicht stabilisierbaren Anforderungen (+) → Aufwand für RE (+) → Projektdauer (+) → Änderungen der nicht stabilisierbaren Anforderungen (+)

Zudem ergeben sich folgende Wirkungsketten negativer Polarität:

1. Aufwand für die Analyse der stabilisierbaren Anforderungen (+) → Verständnis der stabilisierbaren Anforderungen (+) → Änderungsrate der stabilisierbaren Anforderungen (-) → Änderungen der stabilisierbaren Anforderungen (-) → Aufwand für die Analyse der stabilisierbaren Anforderungen (-)

2. Aufwand für die Analyse der stabilisierbaren Anforderungen (+) → Verständnis des Gesamtsystems (+) → Verständnis der stabilisierbaren Anforderungen (+) → Änderungsrate der stabilisierbaren Anforderungen (-) → Änderungen der stabilisierbaren Anforderungen (-) → Aufwand für die Analyse der stabilisierbaren Anforderungen (-)
3. Aufwand für die Analyse der stabilisierbaren Anforderungen (+) → Aufwand für RE (+) → Projektdauer (+) → Änderungen der nicht stabilisierbaren Anforderungen (+) → Aufwand für die Analyse der nicht stabilisierbaren Anforderungen (+) → Verständnis des Gesamtsystems (+) → Verständnis der stabilisierbaren Anforderungen (+) → Änderungsrate der stabilisierbaren Anforderungen (-) → Änderungen der stabilisierbaren Anforderungen (-) → Aufwand für die Analyse der stabilisierbaren Anforderungen (-)
4. Änderungen der nicht stabilisierbaren Anforderungen (+) → Aufwand für die Analyse der nicht stabilisierbaren Anforderungen (+) → Verständnis des Gesamtsystems (+) → Verständnis der stabilisierbaren Anforderungen (+) → Änderungsrate der stabilisierbaren Anforderungen (-) → Änderungen der stabilisierbaren Anforderungen (-) → Aufwand für die Entwicklung (-) → Projektdauer (-) → Änderungen der nicht stabilisierbaren Anforderungen (-)

Aufstellen der mathematischen Gleichungen sowie Erstellung des Flussgraphen und Kalibrierung des Modells

Die Gleichungen, die das Modell vollständig beschreiben, finden sich in Anhang A.

Das Modell besteht aus sieben Modulen:

1. Entwicklung der stabilisierbaren Anforderungen (Abbildung 3-6)
2. Entwicklung der nicht stabilisierbaren Anforderungen (Abbildung 3-7)
3. Entwicklung der stabilen Anforderungen (Abbildung 3-8)
4. Beeinflussung der Änderungsrate (Abbildung 3-9)
5. Aufwandsverteilung (Abbildung 3-10 und Abbildung 3-11)
6. Kostenrechnung (Abbildung 3-12)
7. Referenzgrößen (Abbildung 3-13)

Die Module 1- 3 beschreiben die Anforderungsanalyse und Realisierung der einzelnen Anforderungstypen. Gleichzeitig bilden sie mögliche Änderungen im Projektverlauf ab. Wie auch Ferreira et al. (2003) berücksichtigen wir nur Änderungen, die an bereits analysierten Anforderungen stattfinden. Änderungskosten an noch nicht analysierten Anforderungen sind so gering, dass sie vernachlässigt werden können. Es wird jeweils zu Beginn die Menge der Anforderungen des jeweiligen Anforderungstyps in Personentagen berechnet. Gemäß der in Modul 5 bestimmten Analyserate für den jeweiligen Anforderungstyp werden die Anforderungen analysiert. Bei nicht stabilen Anforderungen muss berücksichtigt werden, dass bereits analysierte Anforderungen zurück in den Pool der nicht analysierten Anforderungen wandern können. Die Rate nach der dies geschieht ändert sich im Projektverlauf und wird in Modul 4 ermittelt. In einem zweiten Schritt werden gemäß der in Modul 5 bestimmten Realisierungsrate analysierte Anforderungen zu realisierten Anforderungen. Bei instabilen Anforderungen ergibt sich ein zweiter Analyse- und Realisierungszyklus für Änderungen an bereits realisierten Anforderungen. Dieser zweite Zyklus ist notwendig, da diese Anforderungen anderen Realisierungskosten und im

Fälle der stabilisierbaren Anforderungen auch einer anderen Änderungsrate unterliegen. Die geänderten Realisierungskosten ergeben sich aus der Annahme, dass Änderungen an Anforderungen, die bereits realisiert sind, besonders teuer sind (vgl. Boehm 1981 und Numuliani et al. 2006). Je nach Anforderung können diese zusätzlichen Kosten sehr unterschiedlich ausfallen, da beispielsweise die Anzahl der Komponenten, die von der Anforderung betroffen ist, sich unmittelbar in den Änderungskosten widerspiegelt. Als grobe Abschätzung gaben die Experten der FAUSER AG an, dass die Änderung einer bereits realisierten Anforderung im Schnitt fünfmal so teuer ist wie ihre Erstrealisierung. Daher ergibt sich eine deutlich langsamere Realisierungsrate für Änderungen an bereits realisierten Anforderungen. Die Änderungsrate bereits realisierter stabilisierbarer Anforderungen wird als geringer angenommen, da davon ausgegangen werden kann, dass im Mittel aufgrund von Lerneffekten während der Realisierung Anforderungen besser verstanden wurden und somit weniger instabil sind.

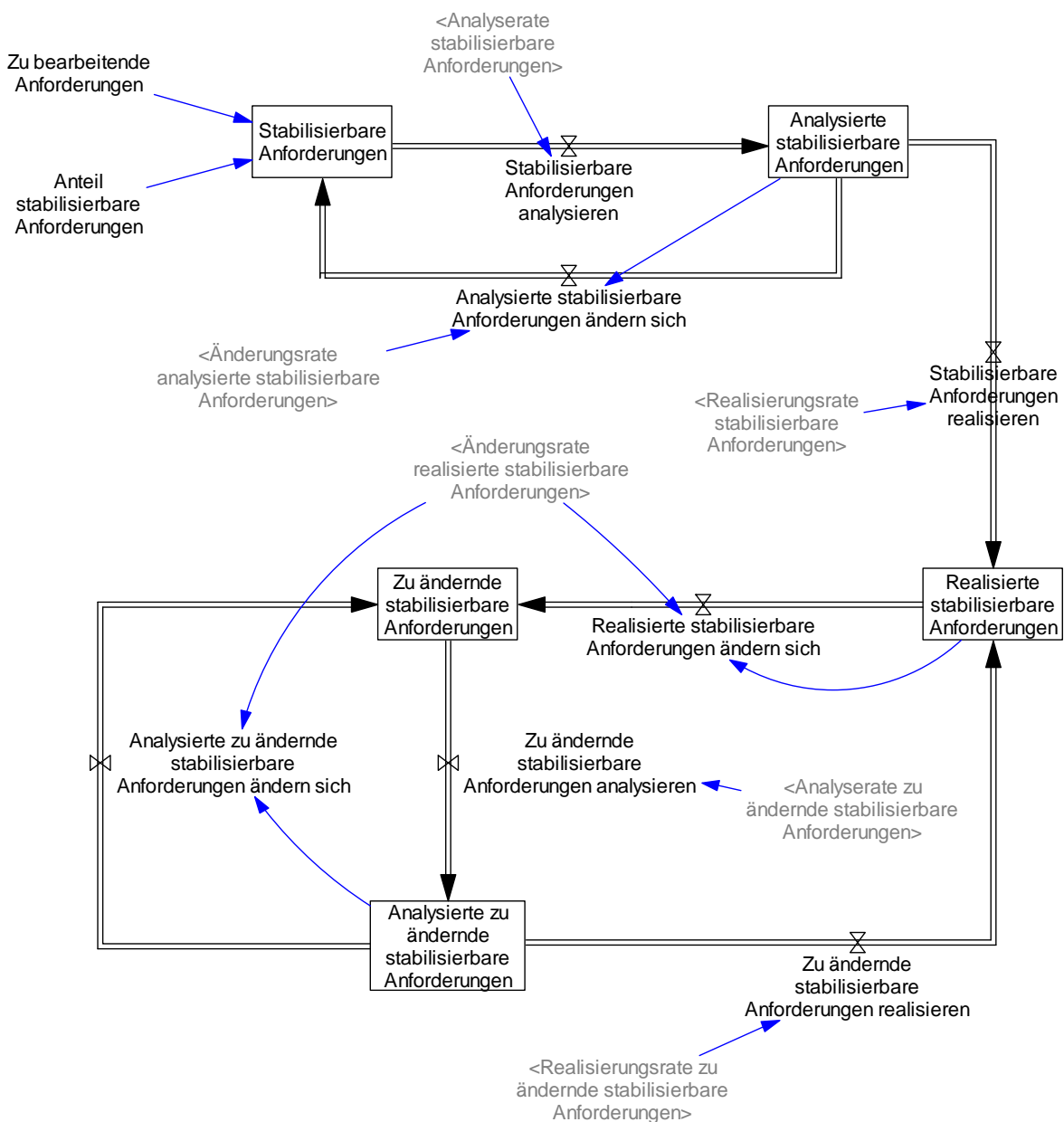


Abbildung 3-6: Flussgraph des Moduls „Entwicklung der stabilisierbaren Anforderungen“

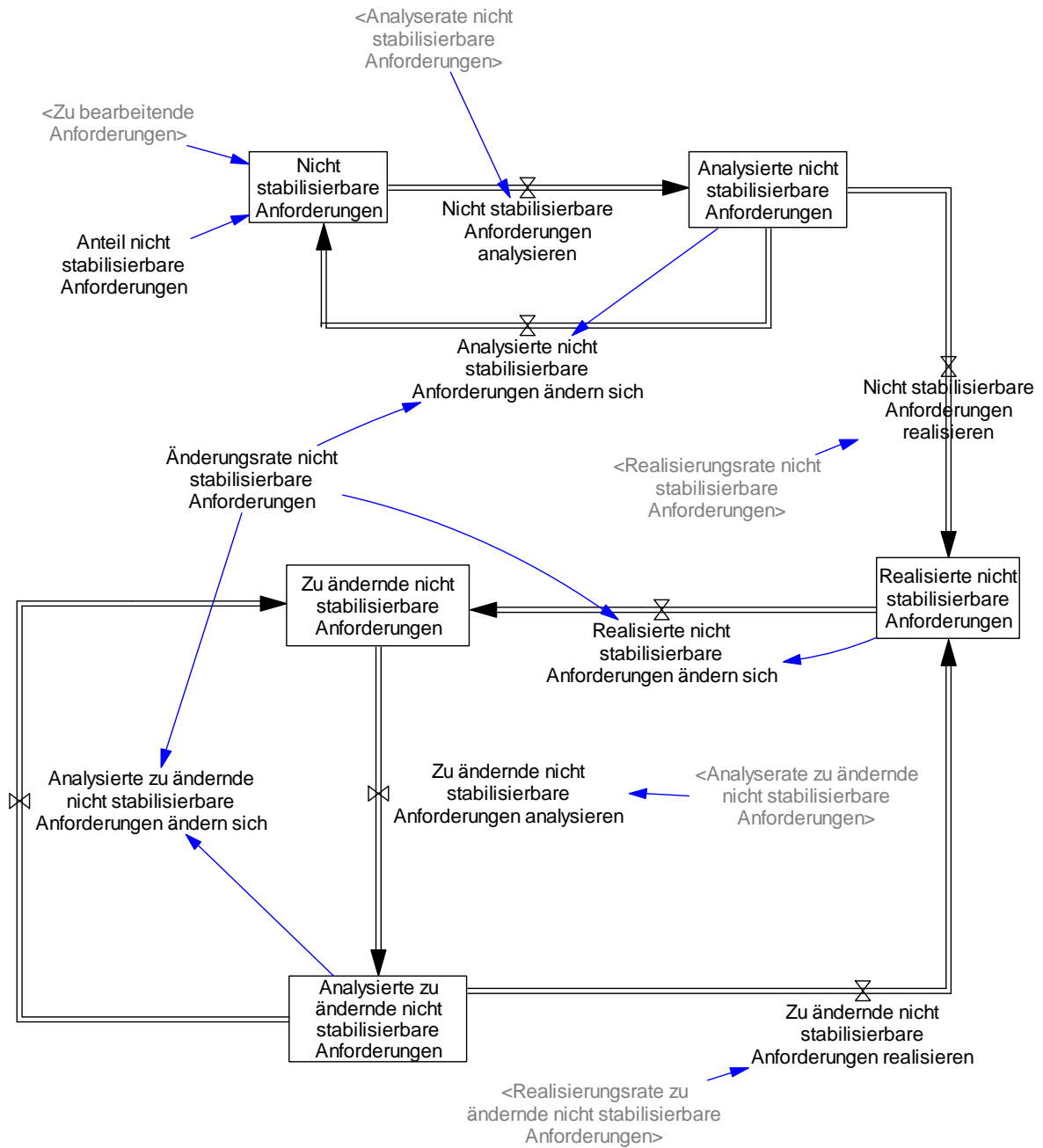


Abbildung 3-7: Flussgraph des Moduls „Entwicklung der nicht stabilisierbaren Anforderungen“

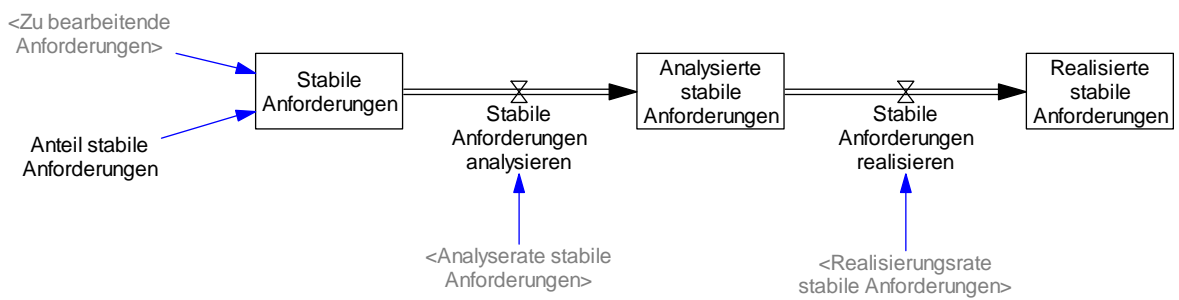


Abbildung 3-8: Flussgraph des Moduls „Entwicklung der stabilen Anforderungen“

Im Modul „Beeinflussung der Änderungsrate“ wird die Änderungsrate der stabilisierbaren Anforderungen errechnet. Für diese Berechnung werden mehrere Hilfsvariable eingeführt, die das Verständnis der stabilisierbaren Anforderungen darstellen. Dieses ist eine abstrakte Größe, die die Ermittlung der Änderungsrate für stabilisierbare Anforderungen aus dem Aufwand für die Analyse der stabilisierbaren Anforderungen, dem Anteil bereits analysierter Anforderungen und dem Entwicklungsstatus der Anforderungen in zwei Schritten ermöglicht. Für die Berechnung des Verständnisses werden der Verständniskennwert über die Anforderungsanalyse sowie über Lerneffekte während der Realisierung und der indirekte Verständniskennwert über die Kenntnis des Gesamtsystems einbezogen. Präzise Werte für den jeweiligen Verständniskennwert und die Ableitung der Änderungsrate aus dem Verständnis sind sehr schwer zu finden. Die Experten der FAUSER AG konnten hier nur grobe Abschätzungen treffen, so dass sich eine Unsicherheit bezüglich der absoluten Werte im Modell ergibt. In Bezug auf den qualitativen Verlauf des Verhältnisses zwischen Analyseaufwand, Anzahl der analysierten Anforderungen und gewonnenem Verständnis waren die Experten sich jedoch sicher. Gleiches gilt für das Verhältnis zwischen gewonnenem Verständnis und resultierender Änderungsrate. Die Experten der FAUSER AG gaben an, dass bei zunehmendem Aufwand für die Analyse von stabilisierbaren Anforderungen das Verständnis für diese logarithmisch zunimmt. Sie nahmen zudem an, dass bei steigendem Anteil der bereits analysierten Anforderungen sich das Verständnis für die stabilisierbaren Anforderungen ebenfalls logarithmisch erhöht. Den größten Beitrag zum Verständnis für die stabilisierbaren Anforderungen wird von den Experten der FAUSER AG ihrer direkten Analyse zugeschrieben. Das Verhältnis zwischen Verständnis der stabilisierbaren Anforderungen und ihrer Änderungsrate wurde als linear angenommen. Bei steigendem Verständnis sinkt die Änderungsrate. Dabei wurde davon ausgegangen, dass ein gewisser Level an Verständnis immer gegeben ist. Wird dieser unterschritten, kann die Änderungsrate enorm abweichen. In der Simulation ist darauf zu achten, dass dieses Minimum an angenommenem Verständnis nicht unterschritten wird. Insgesamt ergibt sich das Bild, dass mit steigendem Aufwand für die Analyse stabilisierbarer Anforderungen und auch mit steigendem Anteil der bereits analysierten Anforderungen die Änderungsrate zunächst stark abfällt, diese Absenkung aber zunehmend abflacht.

Aufgrund der Unsicherheit bzgl. des genauen Verhältnisses zwischen Analyseaufwand für stabilisierbare Anforderungen, Anteil der analysierten Anforderungen, Verständnis und Änderungsrate wurden mehrere Tests durchgeführt, in denen der quantitative Verlauf des Verhältnisses variiert, der qualitative Verlauf jedoch beibehalten wurde. Die Tests ergaben, dass die Aussagen, die wir aus der Simulation ziehen, trotz der Unsicherheit in diesem Punkt aufrechterhalten werden können.

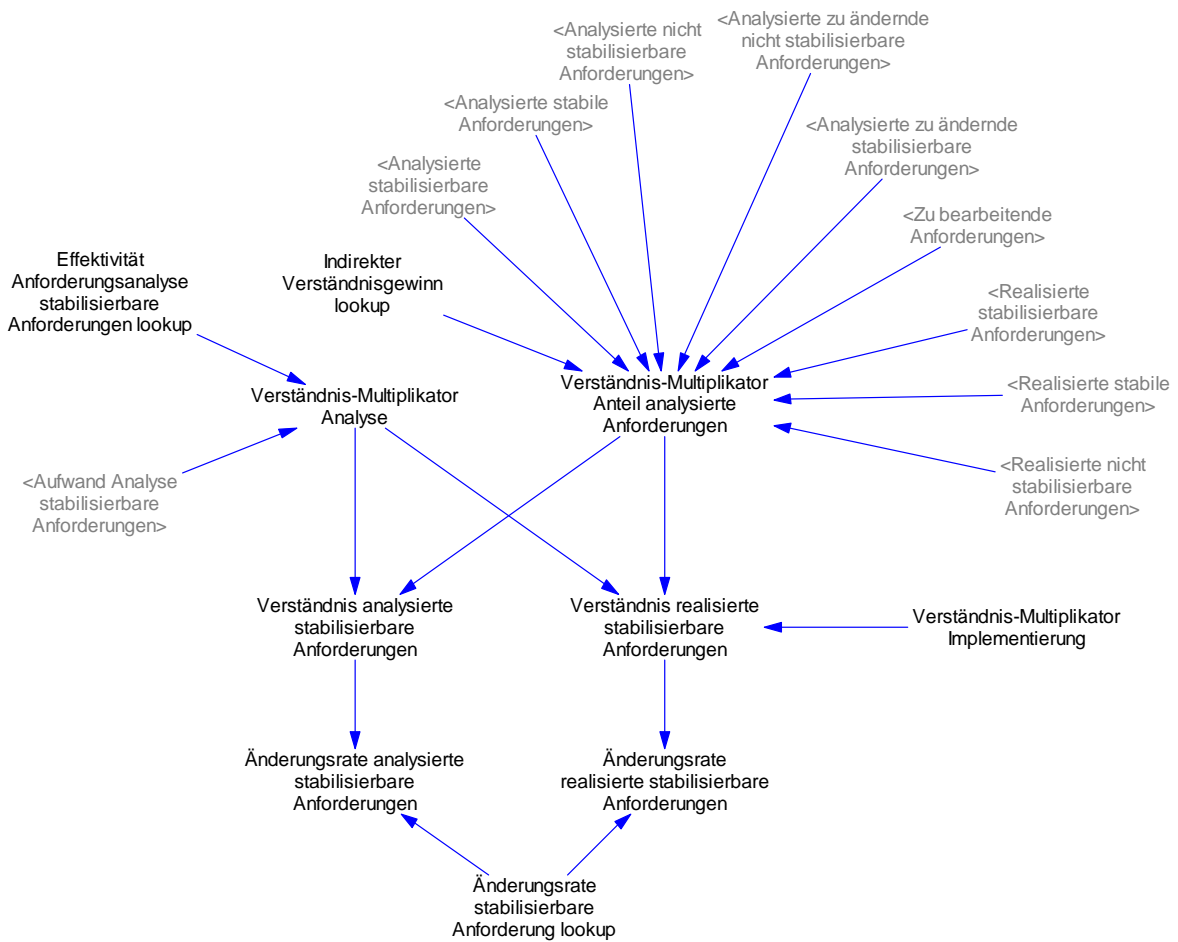


Abbildung 3-9: Flussgraph des Moduls „Beeinflussung der Änderungsrate“

Im Modul „Aufwandsverteilung“ werden die Analyse- und Realisierungsraten der einzelnen Anforderungstypen bestimmt. Zudem wird in diesem Modul kontrolliert zu welchem Zeitpunkt Anforderungsanalyse und Realisierung erfolgen. Der Übersichtlichkeit halber zeigen wir in Abbildung 3-11 einen Ausschnitt aus dem Modul, der die Berechnung der Analyse- und Realisierungsrate der stabilisierbaren Anforderungen betrifft. Bei der Bestimmung der Analyse- und Realisierungsraten wurde angenommen, dass bei der Reihenfolge, in der Anforderungen analysiert und realisiert werden, nicht unterschieden wird, ob eine Anforderung stabil, stabilisierbar oder nicht stabilisierbar ist. Die verfügbare Arbeitskraft wird also gleichmäßig der Analyse beziehungsweise Realisierung der unterschiedlichen Anforderungstypen zugewiesen. Hervorzuheben ist in diesem Modul, dass Änderungen an bereits realisierten Anforderungen im Gegensatz zu ihrer Erstrealisierung als deutlich teurer angenommen werden. Dies spiegelt sich darin wider, dass die entsprechenden Realisierungsraten geringer ausfallen.

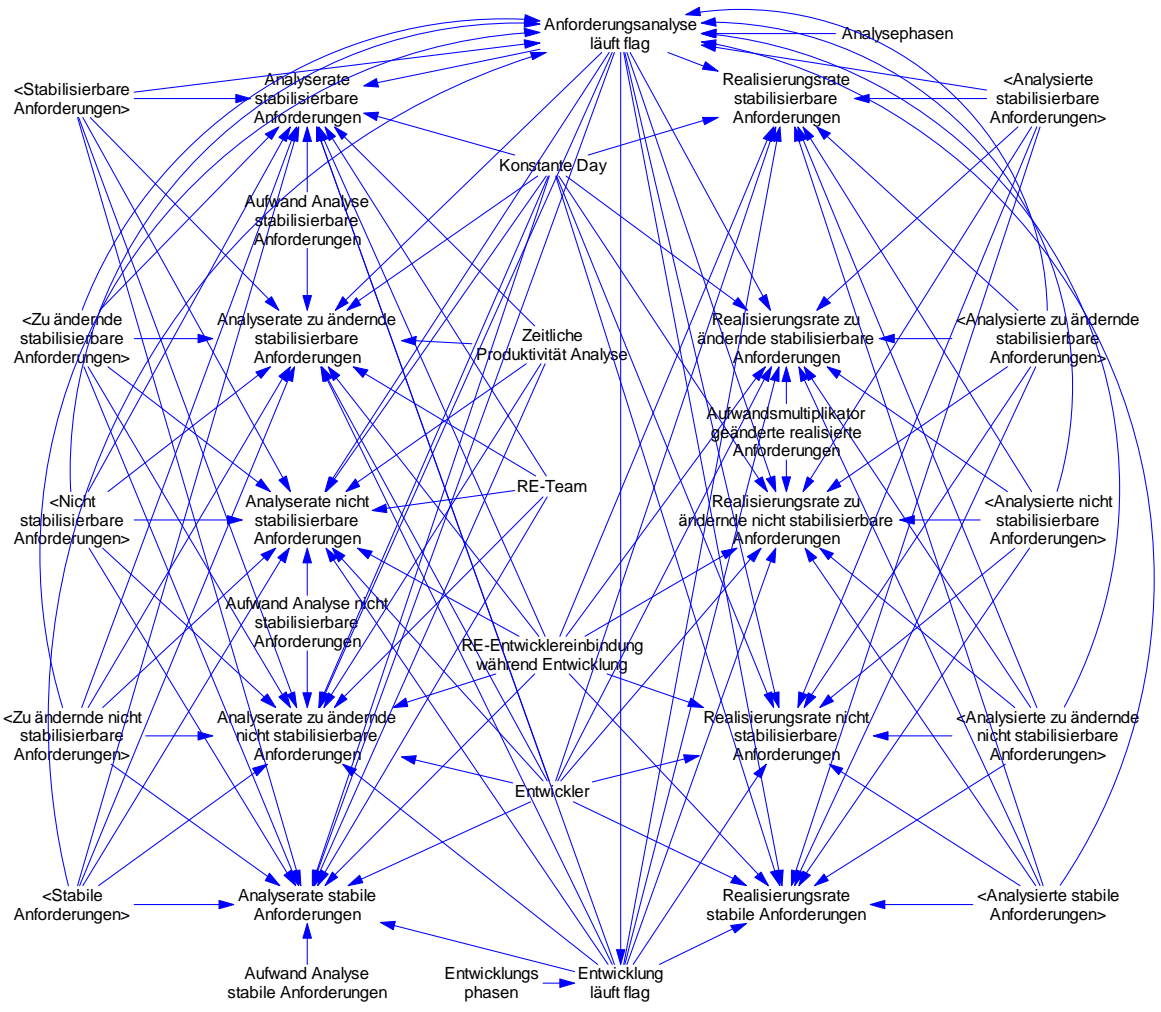


Abbildung 3-10: Flussgraph des Moduls „Aufwandsverteilung“

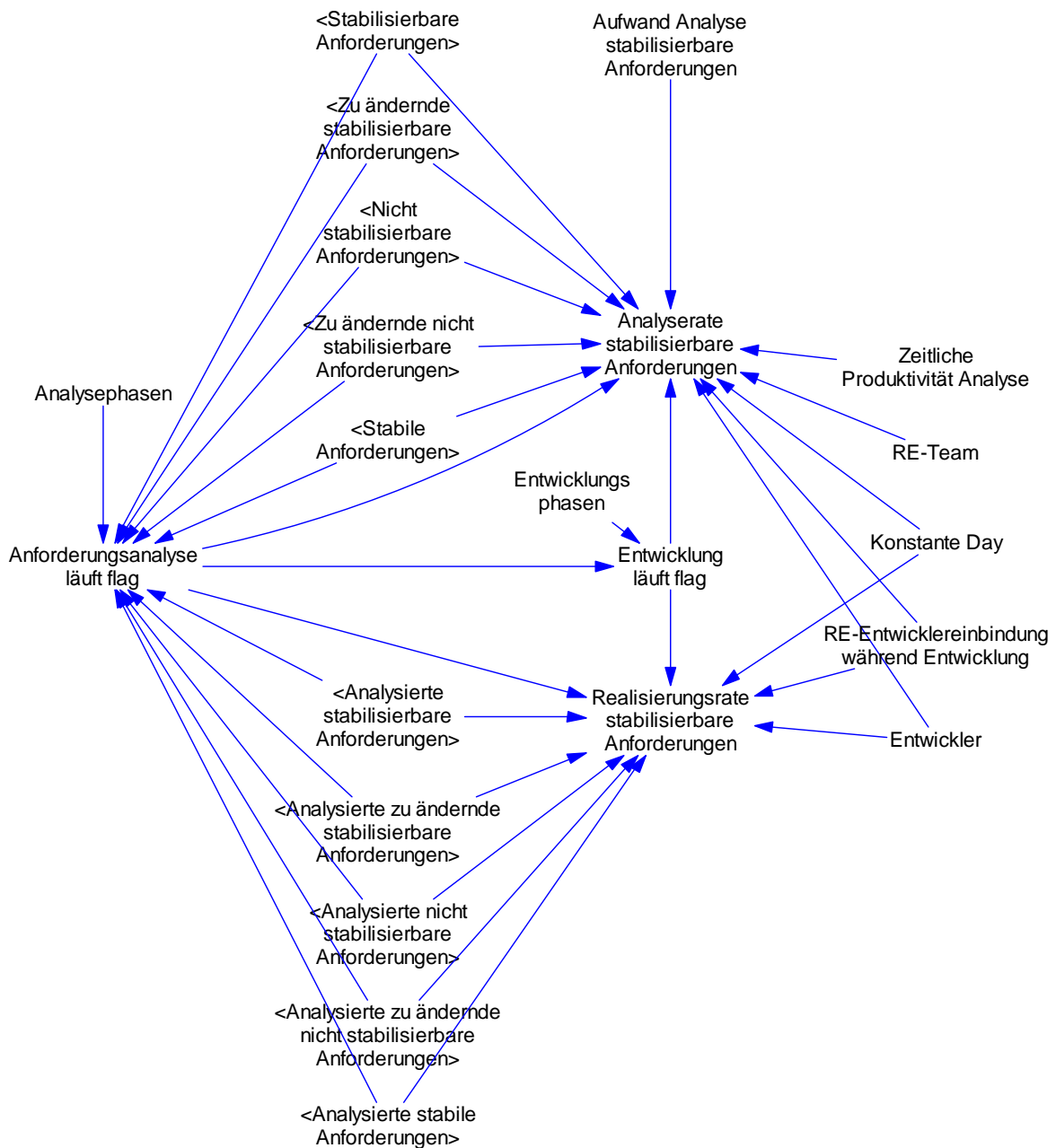


Abbildung 3-11: Ausschnitt aus dem Flussgraphen des Moduls „Aufwandsverteilung“

In den Modulen „Kostenrechnung“ und „Referenzgrößen“ werden die Gesamtkosten des Projekts, das durchschnittliche Verständnis und die durchschnittliche Änderungsrate für die Gesamtheit der stabilisierbaren Anforderungen berechnet. Zudem werden die Gesamtzahl der realisierten Anforderungen und die Menge der zu einem bestimmten Zeitpunkt einer Analyse unterzogenen Anforderungen ermittelt. Diese Größen werden lediglich für die Auswertung der Simulation benötigt.

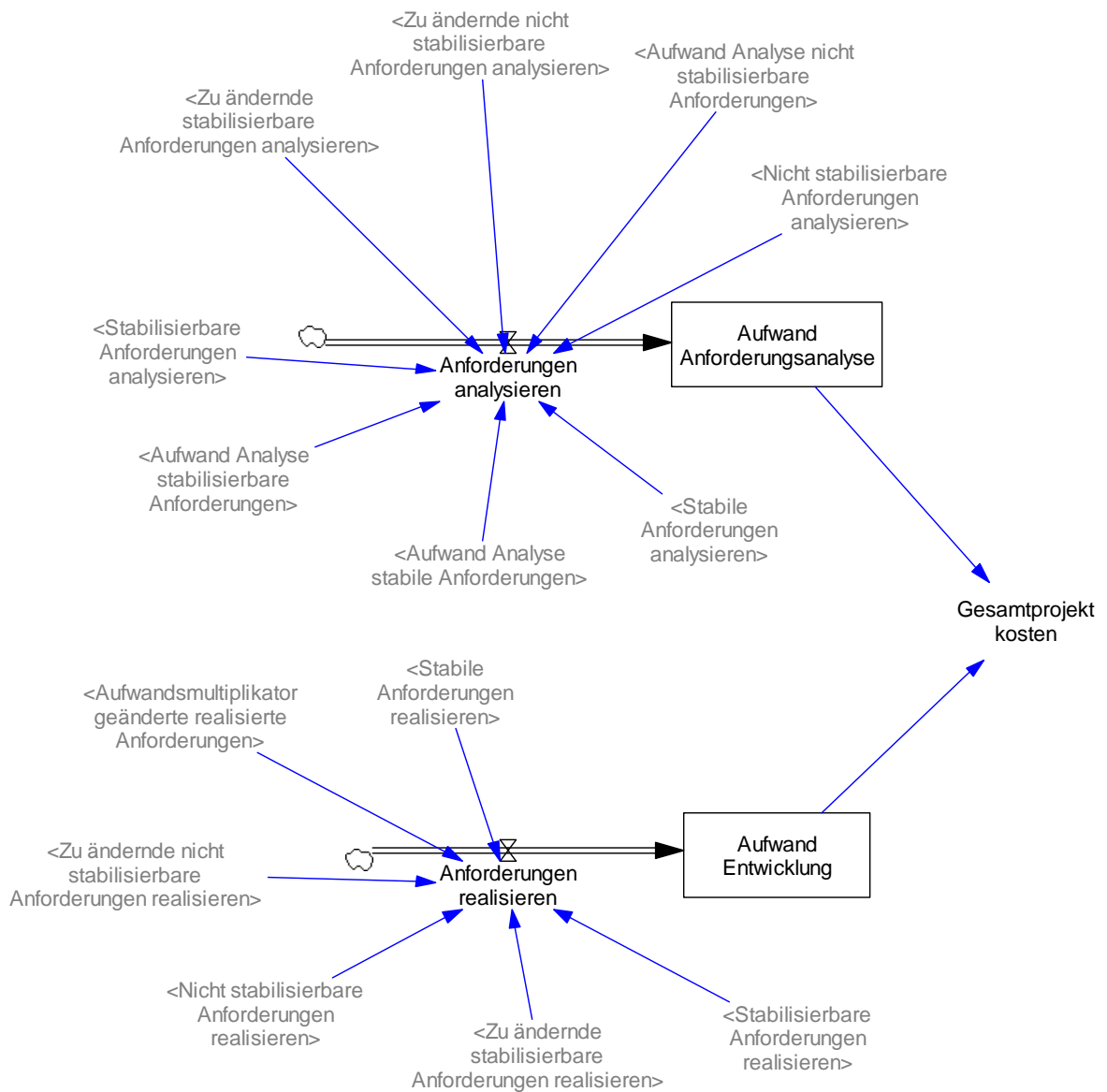


Abbildung 3-12: Flussgraph des Moduls „Kostenrechnung“

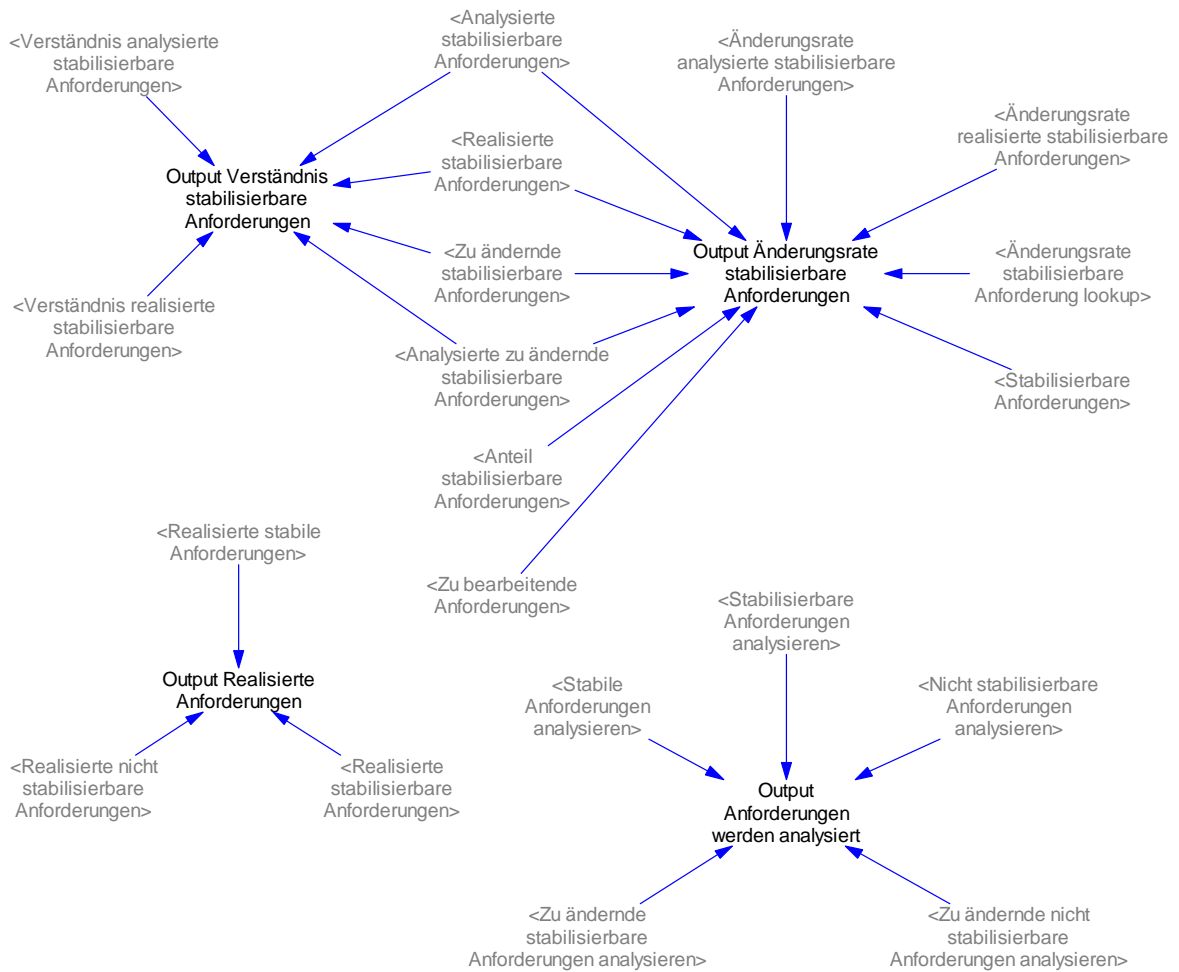


Abbildung 3-13: Flussgraph des Moduls „Referenzgrößen“

Wir wollen anhand des in Abbildung 3-14 gezeigten Ausschnittes aus dem Modul „Entwicklung der nicht stabilisierbaren Anforderungen“ exemplarisch eine detaillierte Beschreibung einiger Gleichungen des Modells geben. Die verwendete Notation in den Gleichungen entspricht der des System Dynamics Modellierungstools Vensim 3.0 (Ventana Systems 1997).

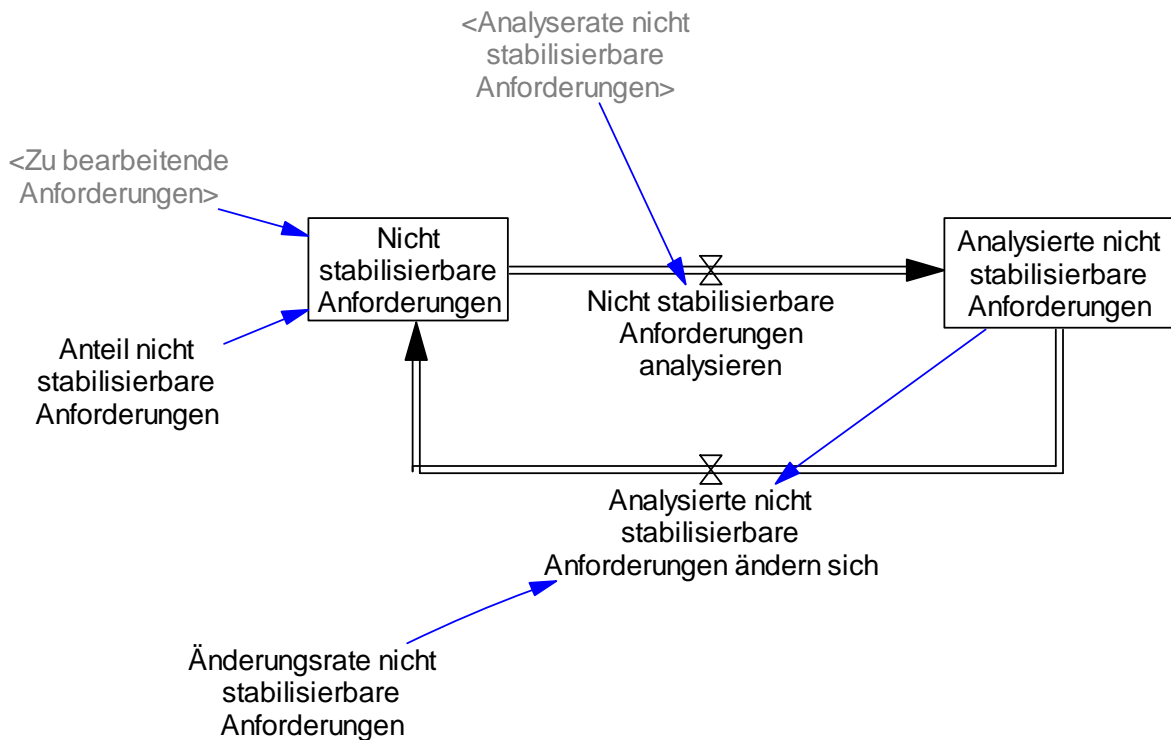


Abbildung 3-14: Ausschnitt aus dem Flussgraph des Moduls „Entwicklung der nicht stabilisierbaren Anforderungen“

Der Level „Nicht stabilisierbare Anforderungen“ stellt die Menge der noch nicht analysierten nicht stabilisierbaren Anforderungen dar, seien dies neue oder geänderte Anforderungen. Der Level errechnet sich nach folgender Formel:

$$\begin{aligned} \text{Nicht stabilisierbare Anforderungen} = & \\ & \text{INTEG} (\text{Analysierte nicht stabilisierbare Anforderungen ändern sich} - \text{Nicht} \\ & \text{stabilisierbare Anforderungen analysieren, Zu bearbeitende Anforderungen} \\ & * \text{Anteil nicht stabilisierbare Anforderungen}) \\ & \sim \text{Person} * \text{Day} \end{aligned}$$

Die Menge der nicht stabilisierbaren Anforderungen wird in Personentagen gemessen und entspricht den tatsächlichen Kosten für die Erstrealisierung der betreffenden Anforderungen. Der Initialwert des Levels errechnet sich aus dem Produkt der beiden Konstanten „Zu bearbeitende Anforderungen“ und „Anteil nicht stabilisierbare Anforderungen“. Die Gesamtmenge der in dem Beispielprojekt zu realisierenden Anforderungen wurde seitens der Experten der FAUSER AG mit 50 Personentagen beziffert, der Anteil der nicht stabilisierbaren Anforderungen mit 20%. Die Einheit, in der der Anteil der nicht stabilisierbaren Anforderungen gemessen wird ist dimensionslos, was in der verwendeten Notation durch „Dmnl“ ausgedrückt wird:

$$\begin{aligned} \text{Zu bearbeitende Anforderungen} = & 50 \\ & \sim \text{Person} * \text{Day} \end{aligned}$$

$$\begin{aligned} \text{Anteil nicht stabilisierbare Anforderungen} = & 0.2 \\ & \sim \text{Dmnl} \end{aligned}$$

Beide Werte stellen eine intuitive Einschätzung seitens der Experten der FAUSER AG im Nachgang des Projektes dar. Die Experten gaben an, dass für Anforderungen schon während des Projekts mit in etwa 95% Genauigkeit bestimmt werden kann, ob sie als stabil, stabilisierbar oder nicht stabilisierbar einzuordnen sind. Der Level „Nicht stabilisierbare Anforderungen“ verändert sich gemäß der Rate „Nicht stabilisierbare Anforderungen analysieren“, welche angibt in welchem Umfang Anforderungen pro Zeitschritt analysiert werden, und der Rate „Analysierte nicht stabilisierbare Anforderungen ändern sich“, die wiedergibt in welchem Umfang bereits analysierte Anforderungen pro Zeitschritt geändert werden und somit neu analysiert werden müssen:

Nicht stabilisierbare Anforderungen analysieren =
 Analyserate nicht stabilisierbare Anforderungen
 ~ Person

Analysierte nicht stabilisierbare Anforderungen ändern sich =
 Analysierte nicht stabilisierbare Anforderungen * Änderungsrate nicht stabilisierbare Anforderungen
 ~ Person

Beide Raten haben die Einheit Person, da sie zeitliche Ableitungen von Levels darstellen, die als Einheit Personentage haben. Anders ausgedrückt stellt jede Rate dar, wie viele Personentage an Anforderungen pro Tag analysiert werden. Die Rate nach der nicht stabilisierbare Anforderungen analysiert werden ergibt sich aus der im Modul „Aufwandsverteilung“ ermittelten Hilfsvariable „Analyserate nicht stabilisierbare Anforderungen“:

Analyserate nicht stabilisierbare Anforderungen =
 IF THEN ELSE (Anforderungsanalyse läuft flag, MIN (Nicht stabilisierbare Anforderungen / Konstante Day, ZIDZ (Nicht stabilisierbare Anforderungen, Nicht stabilisierbare Anforderungen + Stabile Anforderungen + Stabilisierbare Anforderungen + Zu ändernde nicht stabilisierbare Anforderungen + Zu ändernde stabilisierbare Anforderungen) * („RE-Team“ + IF THEN ELSE (Entwicklung läuft flag, „RE-Entwicklereinbindung während Entwicklung“ * Entwickler, Entwickler)) * Zeitliche Produktivität Analyse / Aufwand Analyse nicht stabilisierbare Anforderungen), 0)
 ~ Person

Analog zur Rate „Analysierte nicht stabilisierbare Anforderungen ändern sich“ hat auch diese Hilfsvariable die Einheit Person. Als Bedingung wird für die Ermittlung der Analyserate geprüft, ob das Projekt sich in einer Analysephase befindet („Anforderungsanalyse läuft flag“). Ist dem nicht der Fall, so ist die Rate gleich Null. Im anderen Fall wird zunächst bestimmt wie viele Personen für die Anforderungsanalyse verfügbar sind:

(„RE-Team“ + IF THEN ELSE (Entwicklung läuft flag, „RE-Entwicklereinbindung während Entwicklung“ * Entwickler, Entwickler))

Personen, die an der Anforderungsanalyse beteiligt werden sind sowohl Entwickler, also auch Personen, die nur in der Anforderungsanalyse mitwirken (Konstante „RE-Team“).

Falls aktuell eine Entwicklungsphase stattfindet (Hilfsvariable „*Entwicklung läuft flag*“) so ist die Zahl der an der Anforderungsanalyse beteiligten Entwickler gleich dem Produkt aus der Gesamtzahl der Entwickler (Konstante „*Entwickler*“) und dem Prozentsatz zu dem die Entwickler während einer Entwicklungsphase zeitlich für RE zur Verfügung stehen (Konstante „*RE-Entwicklereinbindung während Entwicklung*“). Findet im betreffenden Zeitschritt keine Entwicklungsphase statt, so werden alle Entwickler an der Anforderungsanalyse beteiligt.

Weiterhin wird berechnet welcher Teil der aktuell für die Anforderungsanalyse aufgewendeten Arbeitskraft für die Analyse nicht stabilisierbarer Anforderungen aufgewendet wird:

ZIDZ (Nicht stabilisierbare Anforderungen, Nicht stabilisierbare Anforderungen + Stabile Anforderungen + Stabilisierbare Anforderungen + Zu ändernde nicht stabilisierbare Anforderungen + Zu ändernde stabilisierbare Anforderungen)

Bei der FAUSER AG wird bei der Auswahl der als nächstes zu analysierenden Anforderungen nicht nach dem Typ der Anforderungen unterschieden. Für die Simulation gehen wir daher davon aus, dass wenn zu einem bestimmten Zeitpunkt beispielsweise ein Fünftel der Anforderungen der Kategorie „nicht stabilisierbare Anforderungen“ angehört im nächsten Zeitschritt ein Fünftel des Analyseaufwands auf Anforderungen des entsprechenden Typs aufgewendet wird, was im Schnitt einer zufälligen Auswahl der Anforderungen entspricht. Entsprechend wird ermittelt wie groß der Anteil der zu analysierenden nicht stabilisierbaren Anforderungen an der Gesamtzahl der zu analysierenden Anforderungen ist. ZIDZ bezeichnet einen Quotient, der für den Fall, dass der Divisor gleich Null ist, als Null interpretiert wird (Zero If Divided by Zero).

Die für nicht stabilisierbare Anforderungen in einem Zeitschritt aufgewendete Arbeitskraft ergibt sich aus dem Produkt aus der Zahl der daran beteiligten Personen und dem Anteil ihrer Arbeitskraft, den sie für die Analyse nicht stabilisierbare Anforderungen aufwenden. Weiterhin muss ihre Produktivität in der Analyse (Konstante „*Zeitliche Produktivität Analyse*“) berücksichtigt werden. Dieser Faktor spiegelt wider, dass eine am RE beteiligte Person aufgrund von beispielsweise Reisezeiten pro Kalendertag nicht einen ganzen Personentag Arbeit verrichtet. Schließlich muss noch einberechnet werden wie viel Personentage Analyseaufwand für eine nicht stabilisierbare Anforderung mit dem Umfang von einem Personentag aufgewendet wird (Konstante „*Aufwand Analyse nicht stabilisierbare Anforderungen*“). Falls damit mehr nicht stabilisierbare Anforderungen analysiert werden würden, als überhaupt vorhanden sind, so wird die Analyserate gleich der Zahl der noch vorhandenen nicht stabilisierbaren Anforderungen gesetzt.

Die Menge der bereits analysierte Anforderungen, die sich in einem Zeitschritt ändern (Rate „*Analysierte nicht stabilisierbare Anforderungen ändern sich*“) ergibt sich als Produkt aus der zum betreffenden Zeitpunkt vorliegenden analysierten Anforderungen (Level „*Analysierte nicht stabilisierbare Anforderungen*“) und der für nicht stabilisierbare Anforderungen angenommenen Änderungsrate „*Änderungsrate nicht stabilisierbare Anforderungen*“:

$$\text{Änderungsrate nicht stabilisierbare Anforderungen} = 0.002 \\ \sim 1 / \text{Day}$$

Die Experten der FAUSER AG gaben an, dass sich rund 10% der nicht stabilisierbaren Anforderungen im gesamten Projektverlauf ändern. Umgerechnet ergibt sich damit pro Tag eine ungefähre Änderungsrate von 0,2%.

Der Level „Analyalisierte nicht stabilisierbare Anforderungen“ gibt die Anforderungen wieder, die bereits analysiert wurden und auf Realisierung warten. Er ist durch die folgende Gleichung beschrieben:

$$\begin{aligned} \text{Analyalisierte nicht stabilisierbare Anforderungen} = & \\ & \text{INTEG (Nicht stabilisierbare Anforderungen analysieren - Analyalisierte nicht} \\ & \text{stabilisierbare Anforderungen ändern sich - Nicht stabilisierbare} \\ & \text{Anforderungen realisieren, 0)} \\ & \sim \text{Person} * \text{Day} \end{aligned}$$

Der Level hat analog zum Level „Nicht stabilisierbare Anforderungen“ die Einheit Personentag. Sein Initialwert ist Null, da zu Beginn des Projekts noch keine Anforderungen analysiert wurden. Werden Anforderungen analysiert (Rate „Nicht stabilisierbare Anforderungen analysieren“) nimmt der Wert des Levels zu, werden sie realisiert (Rate „Nicht stabilisierbare Anforderungen realisieren“) oder ändern sie sich (Rate „Analyalisierte nicht stabilisierbare Anforderungen ändern sich“) nimmt er ab.

Der vollständige Satz an Gleichungen der das Modell beschreibt findet sich in Anhang A.

Tabelle 3-1 zeigt einige der für die Simulation zentralen Eingabeparameter. Die Experten der FAUSER AG gaben an, dass in der Regel für 95% der Anforderungen korrekt angegeben werden kann, ob sie stabil, stabilisierbar oder nicht stabilisierbar sind. Die dargestellten Werte stellen Einschätzungen der Experten im Nachgang des Projektes dar und basieren nicht auf dokumentierten Zahlen.

Parameter	Wert
Anteil stabiler Anforderungen	40%
Anteil stabilisierbarer Anforderungen	40%
Anteil nicht stabilisierbarer Anforderungen	20%
Anteil der stabilisierbaren Anforderungen, der im Projektverlauf von Änderungen betroffen ist	30%
Anteil der nicht stabilisierbaren Anforderungen, der im Projektverlauf von Änderungen betroffen ist	10%
Anteil des RE-Aufwands am Gesamtentwicklungsaufwand für eine Anforderung	1/6
Durchschnittlicher Faktor, um den die Durchführung einer Änderung einer bereits realisierten Anforderungen teurer ist als deren Erstrealisierung	5

Tabelle 3-1: Zentrale Eingabeparameter für das Simulationsmodell bei der FAUSER AG

Bei der FAUSER AG sind Änderungen an stabilisierbaren Anforderungen auf eine Reihe von Gründen zurückzuführen. (Tabelle 3-2). Die häufigste Ursache ist, dass der Realisierungsaufwand für die Anforderungen falsch eingeschätzt wurde, was zum Teil auch indirekt auf ein mangelndes Verständnis der Anforderungen zurückzuführen ist. Bei der FAUSER AG kommt es sehr häufig vor, dass die Zeit nicht ausreicht, um alle für ein Release eingeplanten Anforderungen zu realisieren. So müssen viele Anforderungen auf spätere Releases verschoben werden. Die zweithäufigste Ursache ist, dass Anforderungen zunächst falsch verstanden werden und später abgeändert werden müssen. Letztlich auch auf mangelndes Verständnis zurückzuführen ist die falsche Einschätzung der Priorität der Anforderungen. So kommt es bei der FAUSER AG häufig vor, dass Anforderungen zunächst eine so niedrige Priorität zugeordnet bekommen, dass sie fallen gelassen werden.

Im späteren Projektverlauf werden sie jedoch neu eingebracht, da ihnen schließlich doch eine höhere Priorität beigemessen wird. Anforderungsänderungen aufgrund einer falschen Einschätzung der Machbarkeit der Anforderungen kommen bei der FAUSER AG quasi nicht vor, da in der Regel klar ist, ob eine Anforderung technisch machbar ist. Dieser Faktor spielt nur in wenigen Projekten, in denen neue Technologien zum Einsatz kommen, eine Rolle. Anforderungsänderungen treten bei der FAUSER AG auch häufig aufgrund dessen auf, dass Konflikte zwischen Anforderungen entdeckt werden. Ein wesentlicher Faktor für Anforderungsänderungen bei der FAUSER AG ist zudem, dass manche Stakeholder sich nicht von Beginn an in das Projekt einbringen.

Änderungsgrund	Geschätzte Häufigkeit
Faktoren bezüglich des Verständnisses der Anforderungen	10-20%
Falsche Schätzungen der Kosten von Anforderungen	60%
Prioritäten der Anforderungen ändern sich	10%
Machbarkeit der Anforderungen wurde falsch eingeschätzt	<1%
Konflikte zwischen Anforderungen werden gefunden	10%
Neue Stakeholder werden am Projekt beteiligt	10%

Tabelle 3-2: Geschätzte Häufigkeit der Ursachen für Anforderungsänderungen bei der FAUSER AG

Anforderungsänderungen, die nicht vermeidbar sind, sind quasi ausschließlich auf die Änderung von Geschäftsprozessen in der unmittelbaren Umgebung des Systems zurückzuführen. Weitere in Kapitel 2.2.1 identifizierte Faktoren spielen nur eine untergeordnete Rolle. So fallen Anforderungsänderungen aufgrund von neuen Technologien, organisatorischen Faktoren sowie Änderungen in Gesetzen, Regelungen und Standards bei der FAUSER AG kaum ins Gewicht.

Verifikation und Validierung des Modells

Zur Verifikation des Modells wurde zunächst mittels einer Reihe von direkten strukturellen Tests belegt, dass die Grundmechanismen sich korrekt im Modell wiederfinden. Für jeden Grundmechanismus wurde dabei der betreffende Ausschnitt des Modells eines Reviews unterzogen. So musste beispielsweise für den Grundmechanismus „Änderungsrate der stabilisierbaren Anforderungen (+) → Änderungen der stabilisierbaren Anforderungen (+)“ der in Abbildung 3-15 dargestellte Modellausschnitt untersucht werden. Änderungen der stabilisierbaren Anforderungen werden im Simulationsmodell durch drei Raten wiedergegeben, die jeweils Anforderungen in unterschiedlichen Entwicklungsständen betreffen: „Analysierte stabilisierbare Anforderungen ändern sich“, „Realisierte stabilisierbare Anforderungen ändern sich“ und „Analysierte zu ändernde stabilisierbare Anforderungen ändern sich“. Die Raten errechnen sich jeweils wie folgt:

Analysierte stabilisierbare Anforderungen ändern sich =
 Analysierte stabilisierbare Anforderungen * Änderungsrate analysierte stabilisierbare Anforderungen

Realisierte stabilisierbare Anforderungen ändern sich =
 Realisierte stabilisierbare Anforderungen * Änderungsrate realisierte stabilisierbare Anforderungen

Analysierte zu ändernde stabilisierbare Anforderungen ändern sich =
 Analysierte zu ändernde stabilisierbare Anforderungen * Änderungsrate
 realisierte stabilisierbare Anforderungen

Die Levels „Analysierte stabilisierbare Anforderungen“, „Realisierte stabilisierbare Anforderungen“ und „Analysierte zu ändernde stabilisierbare Anforderungen“ nehmen nur positive Werte an, was in einem eigenen Test gezeigt wurde. Man sieht daher unmittelbar, dass eine höhere Änderungsrate der stabilisierbaren Anforderungen im Simulationsmodell jeweils zu einer Steigerung der Änderungen an stabilisierbaren Anforderungen führt. Der untersuchte Grundmechanismus ist also korrekt wiedergegeben.

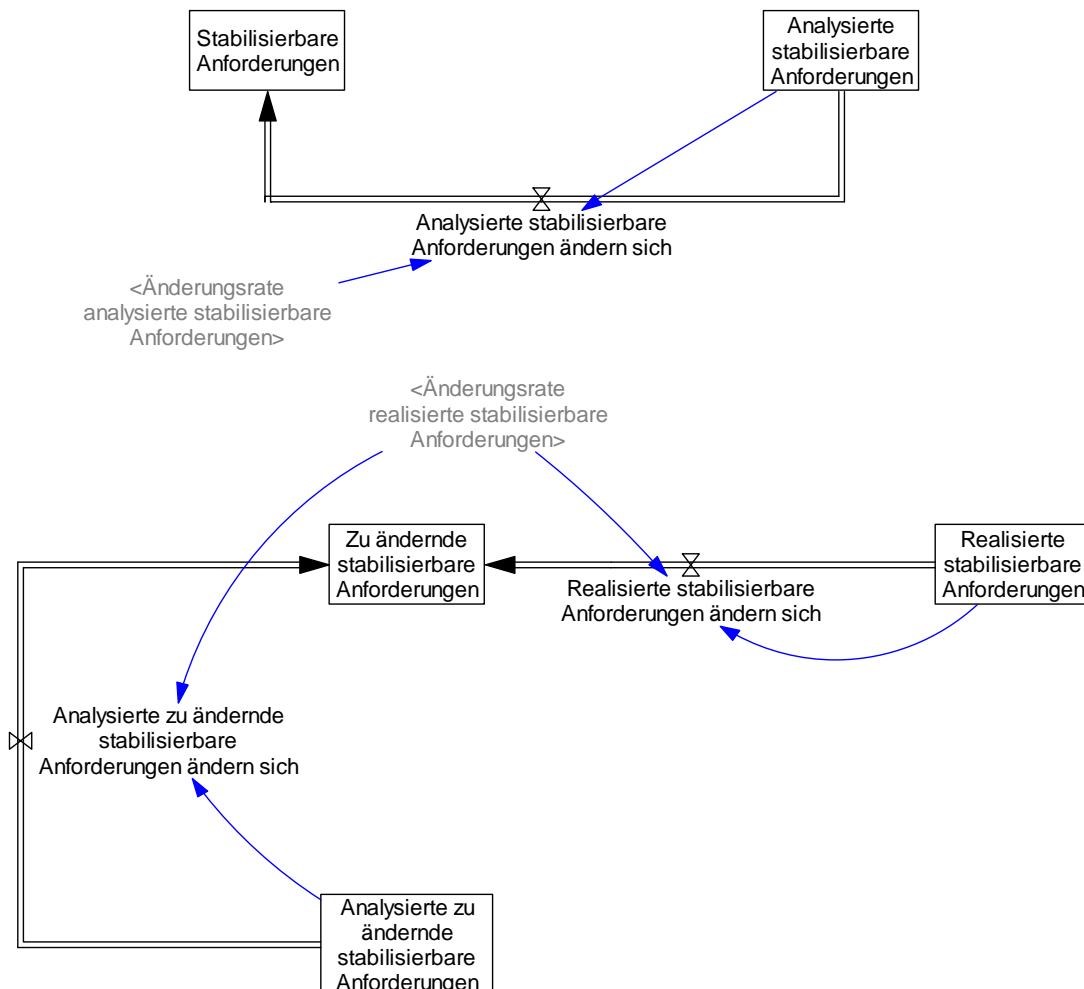


Abbildung 3-15: Für den Grundmechanismus „Änderungsrate der stabilisierbaren Anforderungen (+) → Änderungen der stabilisierbaren Anforderungen (+)“ relevanter Ausschnitt aus dem Simulationsmodell

Die Konsistenz der Einheiten der Modellgrößen wurde werkzeuggestützt sichergestellt.

Unter Einbeziehung der Experten der FAUSER AG wurde für eine Reihe von Parametern eine Sensitivitätsanalyse durchgeführt. Dabei wurde das Verhalten des Modells bei

Änderung dieser Parameter, insbesondere auch für Extremwerte bewertet. Dabei wurden Parameter untersucht, die

- die Menge der zu bearbeitenden Anforderungen festlegen,
- die Unterteilung der Anforderungen in stabilisierbare, nicht stabilisierbare und stabile Anforderungen festlegen,
- die Änderungsrate festlegen,
- den Aufwand für die Anforderungsanalyse festlegen und
- die Kosten von Änderungen an bereits realisierten Anforderungen festlegen

Es wurde festgestellt, dass das Simulationsmodell bei einer extremen Reduzierung des RE-Aufwands keine sinnvollen Ergebnisse ausgibt. So zeigt das Simulationsmodell an, dass eine Entwicklung auch mit einer extremen Reduzierung des RE-Aufwands auf 0,01% des Gesamtaufwands erfolgreich durchgeführt werden kann. Dies ist, wie auch die Experten der FAUSER AG bestätigen, nicht realistisch, da stets ein gewisser Mindestaufwand im RE nötig ist, um ein für die Realisierung notwendiges Minimalverständnis der Anforderungen zu erlangen. Weiterhin ist zu bedenken, dass bei der Unterschreitung eines gewissen Mindestmaßes an RE-Aufwand für stabile und nicht stabilisierbare Anforderungen unsere idealisierte Betrachtung dieser Anforderungstypen nicht aufrechterhalten werden kann. Wird kaum Analyseaufwand für diese Anforderungstypen aufgewendet, können aufgrund eines völlig falschen Verständnisses dieser Anforderungen auch von uns als stabil angenommene Anforderungen instabil werden beziehungsweise nicht stabilisierbare Anforderungen eine höhere und beeinflussbare Änderungsrate aufweisen. Die Experten der FAUSER AG teilen diesen Standpunkt. Ihnen zufolge ist dieses Mindestmaß von Projekt zu Projekt verschieden und kann nicht allgemein angegeben werden. Bei der Durchführung von Experimenten mittels des Simulationsmodells ist daher fallweise zu entscheiden, ob das Mindestmaß für den RE-Aufwand eingehalten wurde. Das Verhalten des Simulationsmodells bei der Variierung der übrigens Parameter wurde jeweils von den Experten der FAUSER AG als realistisch eingestuft.

Zur Validierung des Modells wurden wiederum direkte strukturelle Tests und Verhaltenstests herangezogen.

Gemeinsam mit den Experten der FAUSER AG wurde das Simulationsmodell dahingehend untersucht, ob alle wesentlichen kausalen Zusammenhänge im Modell aufgenommen wurden. Dabei wurde einerseits untersucht, ob die wesentlichen in der Praxis beobachteten und angenommen kausalen Zusammenhänge sich im Modell wiederfinden, und andererseits, ob sich alle im Modell dargestellten kausalen Zusammenhänge aus der Realität ableiten.

Unter Einbeziehung der Experten der FAUSER AG wurde überprüft, ob die Modellparameter korrekt eingestellt wurden. Hierbei wurde insbesondere überprüft inwieweit sich Unsicherheiten in den Eingangsgrößen im Simulationsmodell auswirken. Darüber, wie sich der Aufwand für die Analyse stabilisierbarer Anforderungen, der Anteil der bereits analysierten Anforderungen und Lerneffekte bei der Realisierung auf das Verständnis der stabilisierbaren Anforderungen beziehungsweise ihre Änderungsrate quantitativ auswirken herrschte Unsicherheit seitens der Experten der FAUSER AG. Qualitative Aussagen konnten sie mit großer Sicherheit treffen. So waren sich die Experten beispielsweise sicher, dass bei steigendem Aufwand für die Analyse stabilisierbarer Anforderungen die Änderungsrate zunächst stark abnimmt, diese Abnahme aber

zunehmend abflacht. Der genaue quantitative Verlauf konnte jedoch nicht mit Sicherheit angegeben werden. Es wurden Tests durchgeführt, in denen ein breites Spektrum denkbarer Werte für unsichere Parameter durchlaufen wurde. Es wurde jeweils untersucht, ob das Simulationsmodell trotzdem sinnvolle Ausgaben produziert, insbesondere, ob das Referenzverhalten noch erreicht werden kann. Das Modell erwies sich als robust gegenüber diesen Parameterungenauigkeiten, produzierte also immer noch sinnvolle Ausgaben. Die von uns in dieser Fallstudie anhand des Simulationsmodells getroffenen Aussagen sind auch bei Abweichungen in den unsicheren Parametern haltbar. Bestimmte quantitative Aussagen sind jedoch nicht möglich. So kann beispielsweise geschlussfolgert werden, dass es bei einer Variierung des Aufwands für die Analyse stabilisierbarer Anforderungen hinsichtlich der Projektlaufzeit ein Minimum gibt und sich dieses abhängig von der Änderungsrate verschiebt. Der präzise Wert, für den sich das Minimum in der Realität ergibt, kann jedoch durch das Simulationsmodell aufgrund der Ungenauigkeiten in den Eingabeparametern nicht ermittelt werden.

Es folgte eine Reihe von Verhaltenstests. Dabei wurden die Experten der FAUSER AG hinzugezogen, um zu bewerten, ob das Modell ein sinnvolles und realitätsnahes Verhalten wiedergibt. Dabei wurde insbesondere die Erreichung des zuvor definierten Referenzverhaltens überprüft. Das Referenzverhalten „Menge der realisierten Anforderungen“ konnte sehr genau getroffen werden (Abbildung 3-16). Die Referenzkurve „Menge der hinzukommenden oder geänderten Anforderungen pro Zeitspanne“ konnte mit Einschränkungen reproduziert werden (Abbildung 3-17). Das erwartete hohe Plateau zu Beginn des Projektes wurde korrekt wiedergegeben. Erwartet wurde in der Folge zunächst ein stetiger Abfall der Anzahl der Änderungen bei Beginn der Realisierung mit darauffolgendem Durchlaufen eines Minimums und einem erneutem Anstieg. In der abschließenden Phase der Anforderungsanalyse gegen Ende des Projektes wurde ein relativ hohes Plateau an Änderungen erwartet. Das Modell nähert diese Zeiträume niedriger RE-Aktivität durch mehrere kurze Phasen hoher RE-Aktivität unterbrochen durch Phasen mit gar keiner RE-Aktivität an. Auch mit diesen geringfügigen Einschränkungen kann die Wiedergabe des Referenzverhaltens durch das Simulationsmodell als ausreichend angenommen werden. Das Modell stimmt mit den Ergebnissen der Studien von Stark et al. (1999) sowie Zowghi und Nurmuliani (2002) darin überein, dass Änderungen immer erst in Phasen erhöhter RE-Aktivität zu Tage treten.

Die Entwicklung des Modells erfolgte in mehreren Iterationen. Dabei wurde insbesondere seine Granularität und Modularität angepasst.

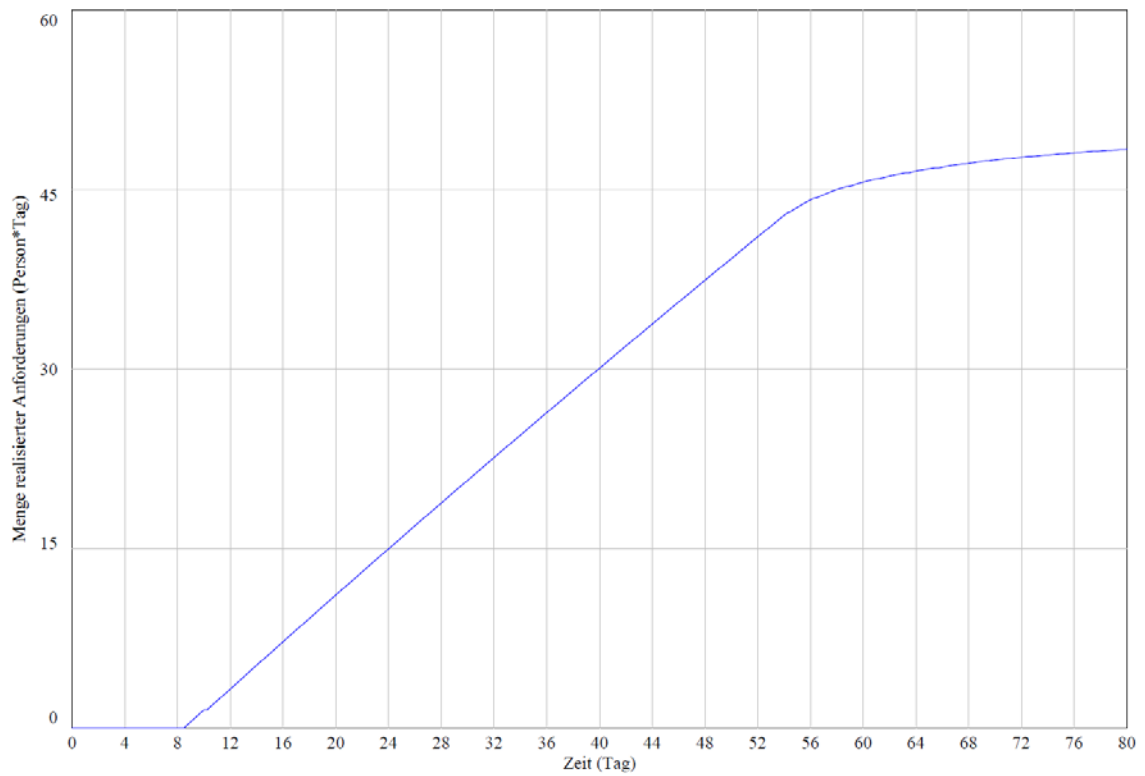


Abbildung 3-16: Reproduktion des Referenzverhaltens „Menge der realisierten Anforderungen“ mittels des Simulationsmodells (die Menge der Anforderungen wird in Form des für sie tatsächlich notwendigen Erstrealisierungsaufwands gemessen)

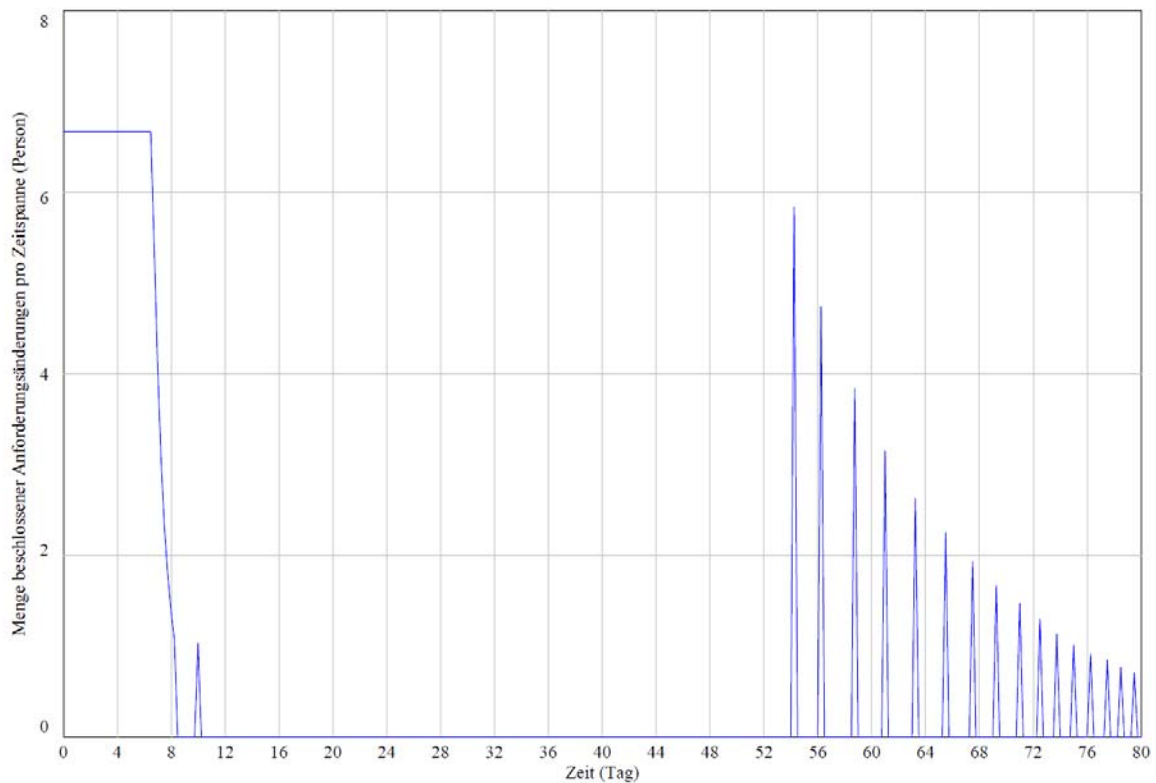


Abbildung 3-17: Reproduktion des Referenzverhaltens „Menge der hinzukommenden oder geänderten Anforderungen pro Zeitspanne“ mittels des Simulationsmodells (die Menge der Anforderungsänderungen wird in Form des für sie tatsächlich notwendigen Erstrealisierungsaufwands gemessen, die Menge pro Zeitspanne demzufolge in Personentag/Tag)

3.4.2 Durchführung von Experimenten

Wir führten mit dem Simulationsmodell eine Reihe von Experimenten durch, um Antworten auf die Fragen in der Problembeschreibung zu finden. Dabei wurde eine Sensitivitätsanalyse verschiedener Modellparameter durchgeführt. Variiert wurden dabei der RE-Aufwand für alle Anforderungstypen, der RE-Aufwand für stabilisierbare Anforderungen, die Verteilung des RE-Aufwands auf die verschiedenen Anforderungstypen, die Änderungsrate der stabilisierbaren Anforderungen sowie die Stabilisierbarkeit der Anforderungen.

Auswirkung des RE-Aufwands auf die Änderungsrate der stabilisierbaren Anforderungen

Zunächst untersuchten wir die Auswirkung der Variierung des RE-Aufwands auf die Änderungsrate der Anforderungen. Dafür wurden die Modellvariablen für den Analyseaufwand der einzelnen Anforderungstypen modifiziert. Abbildung 3-18 zeigt den zeitlichen Verlauf der Änderungsrate für drei verschiedene Simulationsläufe.

Man kann klar erkennen, dass die Änderungsrate durch die Anforderungsanalyse sowohl zu Beginn als auch gegen Ende des Projekts stark gedrückt wird. Zwischenzeitlich steigt die Änderungsrate an, da Anforderungen durch Änderungen wieder in den Pool der nicht analysierten Anforderungen wandern und somit eine höhere Änderungsrate aufweisen. Der Anstieg der Änderungsrate wird im Projektverlauf deutlich flacher. Dies führen wir auf

Lerneffekte durch die Realisierung zurück, die eine geringere Änderungsrate zur Folge haben. Vergleicht man die Kurven der einzelnen Simulationsläufe, so ist festzustellen, dass ein erhöhter Aufwand im RE eine stärkere Absenkung der Änderungsrate bewirkt. Diese Absenkung erfolgt jedoch zeitlich nach hinten versetzt, da die Anforderungsanalyse länger dauert. Im beobachteten Bereich ergibt sich je nach eingesetztem Aufwand für RE eine signifikante Schwankung der Änderungsrate von bis zu 16,0%.

Die Richtigkeit des qualitativen Verlaufs der Kurve ist einerseits dadurch abgesichert, dass er direkt an der Erreichung der Referenzkurve für durchgeführte Änderungen an stabilisierbaren Anforderungen (Abbildung 3-17) beteiligt ist. Die besonders hohe Änderungsrate zu Beginn des Projektes erklärt sich einfach dadurch, dass in dieser Phase noch eine sehr hohe Unsicherheit bezüglich der Anforderungen herrscht und zunächst viele falsche und gleich wieder verworfene Anforderungen aufgestellt werden. Diese Überlegung deckt sich mit dem Verhalten, das gemeinhin in der Praxis beobachtet wird. Mittels der Anforderungsanalyse wird das Verständnis für die Anforderungen immens gesteigert, was zu einer deutlich niedrigeren Änderungsrate führt. Je mehr Aufwand dabei in die Analyse der Anforderungen gesteckt wird, desto stärker fällt dieser Effekt aus. Abhängig von der Art der Anforderungen zahlt sich der Analyseaufwand unterschiedlich aus. So kann die Änderungsrate von nicht stabilisierbaren Anforderungen mit einem gesteigerten RE-Aufwand nicht oder im nicht idealisierten Fall nur geringfügig beeinflusst werden. Auch für die Experten der FAUSER AG steht es außer Zweifel, dass zusätzlicher Aufwand im RE die Änderungsrate drückt, insbesondere wenn dieser Aufwand für stabilisierbare beziehungsweise allgemein risikobehaftete Anforderungen aufgebracht wird. Die Unterschiede zwischen den einzelnen Kurven erklären sich dementsprechend durch den höheren Verständniserfolg bei gesteigertem RE-Aufwand. Analog kann der Kurvenverlauf in der zweiten Analysephase am Ende des Projekts erklärt werden. Während der Realisierungsphase wirken zwei entgegen gesetzte Effekte. Zum einen steigern Lerneffekte während der Realisierung das Verständnis der Anforderungen und senken somit die Änderungsrate. Zum anderen kumulieren sich Änderungen an Anforderungen, die im Beispielprojekt jedoch zunächst nicht bearbeitet werden, sprich es wird keine Analyse der anfallenden Change Requests durchgeführt. Dies erhöht die Unsicherheit der Stakeholder bezüglich der Anforderungen, steigert also die Änderungsrate. Welcher Effekt stärker ist kann allgemein nicht entschieden werden. Es sollte sich jedoch nur eine sehr geringe Variierung der Änderungsrate ergeben. Dies ist in unserem Modell der Fall.

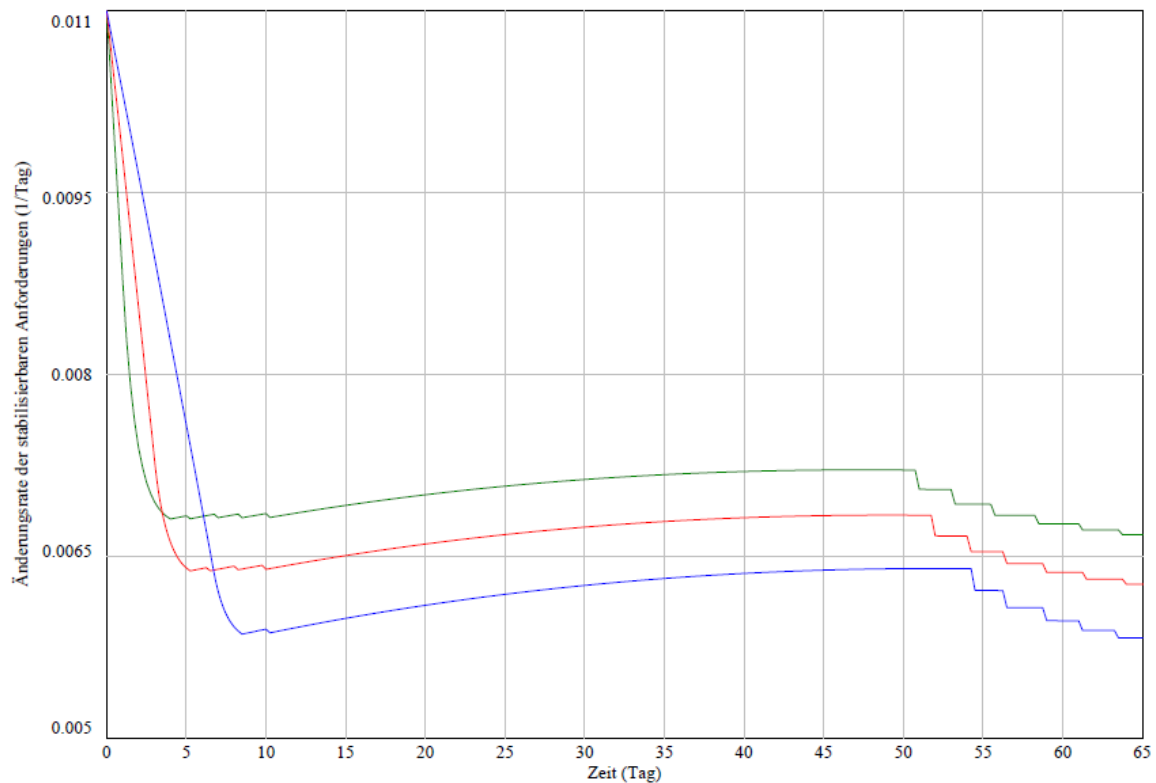


Abbildung 3-18: Simulation des zeitlichen Verlaufs der Änderungsrate der stabilisierbaren Anforderungen (die Änderungsrate wird im Prozentsatz der Anforderungen, der sich pro Tag ändert, gemessen)

Blau: Unveränderter RE-Aufwand

Rot: um 50% reduzierter RE-Aufwand für alle Anforderungstypen

Grün: um 75% reduzierter RE-Aufwand für alle Anforderungstypen

Auswirkung von Schwerpunktsetzungen im RE auf die Projektlaufzeit und die Projektkosten

Die nächste Frage, die wir untersuchten, ist, ob durch die Konzentrierung des RE-Aufwands auf stabilisierbare Anforderungen Vorteile hinsichtlich der Projektlaufzeit beziehungsweise der Projektkosten erlangt werden können. Dafür führten wir eine Reihe weiterer Simulationsläufe durch, bei denen wir den RE-Aufwand unterschiedlich auf die einzelnen Anforderungstypen verteilten. Der Gesamtaufwand für RE wurde dabei konstant gehalten. Der für stabilisierbare Anforderungen aufgebrauchte Anteil des Aufwandes wurde variiert. Der Aufwand für nicht stabilisierbare Anforderungen war dabei stets der gleiche wie der für stabile Anforderungen. Wir beobachteten zu welchem Zeitpunkt 90% der Anforderungen realisiert wurden und die bis dahin aufgelaufenen Kosten in Personentagen.

Bei diesem Experiment wurden wir auf zweierlei Weise eingeschränkt. Zum einen ist der Aufwand für die Analyse stabilisierbarer Anforderungen bei der FAUSER AG ohnehin schon sehr hoch, und zwar, wie wir später sehen werden, bereits höher als empfehlenswert. Bei solch hohen Werten für den Analyseaufwand resultiert eine Variierung des Aufwands nur in geringen Schwankungen in der Änderungsrate. Nur eine starke Abweichung hin zu niedrigeren Aufwänden macht sich deutlich bemerkbar. Zum anderen stehen keine Daten zur Verfügung, die eine Berechnung der Änderungsrate für einen noch wesentlich höheren Analyseaufwand erlauben. Um bei der Variierung im Bereich zu bleiben, in dem die Änderungsrate errechnet werden kann und zudem die Variierung einen deutlichen

Ausschlag in der Änderungsrate bewirkt, gingen wir für dieses Experiment von einem um 35% geringeren Gesamtaufwand für RE aus.

Die Ergebnisse werden in Tabelle 3-3 dargestellt. Man kann erkennen, dass durch eine Konzentration des RE-Aufwands auf stabilisierbare Anforderungen der Zeitaufwand und die Kosten für das Projekt reduziert werden können. Dies ist auch unmittelbar klar, wenn man bedenkt, dass die Analyse von nicht stabilisierbaren und stabilen Anforderungen nur geringe Auswirkungen auf die Änderungsrate der stabilisierbaren Anforderungen haben. Sie tragen zum Verständnis des Gesamtsystems bei, das wiederum einen indirekten Verständnissgewinn für stabilisierbare Anforderungen bedeutet. Man ist dabei hinsichtlich der Steigerung des Verständnisses jedoch wesentlich weniger effizient als bei einer direkten Analyse der stabilisierbaren Anforderungen. Die Schwankung der Gesamtprojektkosten abhängig von der Verteilung des RE-Aufwands beträgt im beobachteten Bereich bis zu 12%. Bedenkt man, dass die Änderungskosten im Normalfall bei der FAUSER AG nur 9,3% der Gesamtkosten ausmachen, zeigt sich, dass die Änderungskosten im beobachteten Bereich um bis zu 230% schwanken. Es ergibt sich also ein signifikantes Optimierungspotenzial.

Zum Realisierungsaufwand relativer Aufwand für die Analyse der einzelnen Anforderungstypen [dimensionslos]			Tag an dem 90% der Anforderungen realisiert wurden	Kosten für die Realisierung von 90% der Anforderungen [Personentage]
Aufwand stabilisierbare Anforderungen	Aufwand nicht stabilisierbare Anforderungen	Aufwand stabile Anforderungen		
0,0124	0,2084	0,2084	62,50	62,00
0,0250	0,2000	0,2000	59,25	58,82
0,1300	0,1300	0,1300	56,50	56,38
0,1750	0,1000	0,1000	56,00	55,97
0,2875	0,0250	0,0250	55,25	55,50

Tabelle 3-3: Vergleich der Projektlaufzeit bei unterschiedlicher Konzentration des RE-Aufwands auf verschiedene Anforderungstypen und gleichbleibenden Gesamtaufwand (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

Auswirkung der Variierung des RE-Aufwands auf die Projektlaufzeit und die Projektkosten

Wir untersuchten, wie sich eine Variierung des RE-Aufwands für stabilisierbare Anforderungen auf die Projektlaufzeit und die Projektkosten auswirkt. Dabei erwarteten wir bei steigendem RE-Aufwand eine Einsparung im Zeit- und Kostenaufwand durch eine niedrigere Änderungsrate. Schon die Studien von Stark et al. (1999) sowie die von Zowghi und Nurmuliani (2002) zeigten, dass die Volatilität von Anforderungen einen negativen Einfluss auf Projektlaufzeit- und Kosten hat. Zum gleichen Ergebnis kommen die Simulationsmodelle von Houston et al. (2001) und Ferreira et al. (2003). Zusätzlich muss jedoch bedacht werden, dass der Zeit- und Kostenersparnis bezüglich des geringeren Änderungsaufwands die Zunahme von Zeit und Kosten durch die längere Dauer der Anforderungsanalyse entgegenwirkt. Bei bestimmten RE-Aufwänden für stabilisierbare Anforderungen erwarteten wir daher lokale Minima der Projektlaufzeit sowie der Projektkosten. Den Aufwand für die Analyse von stabilen und nicht stabilisierbaren

Anforderungen hielten wir in der Simulation konstant. Die vorangegangenen Experimente zeigten bereits, dass dieser nur eine geringe Auswirkung auf die Änderungsrate der stabilisierbaren Anforderungen hat. Wir führten eine Reihe von Simulationsläufen mit jeweils unterschiedlichem RE-Aufwand für stabilisierbare Anforderungen durch.

Abbildung 3-19 und Abbildung 3-20 zeigen den Verlauf von Projektlaufzeit und Projektkosten in der Simulation abhängig vom eingesetzten RE-Aufwand für stabilisierbare Anforderungen. Man erkennt, dass bei einer Erhöhung des RE-Aufwands für stabilisierbare Anforderungen über den für das Beispielprojekt angegebenen Wert von 0,2 Projektlaufzeit und –kosten ansteigen. Bei einer Senkung fallen sie zunächst ab, steigen dann aber wieder an. Wir können aus den Ergebnissen schließen, dass sich lokale Minima in der Projektlaufzeit und den Projektkosten bei einem bestimmten RE-Aufwand für stabilisierbare Anforderungen einstellen. Aufgrund der Granularität des Simulationsmodells, das eine Unsicherheit in der Berechnung der Projektlaufzeit von ca. einem viertel Tag aufweist, lassen sich diese Minima jedoch nur grob eingrenzen. Gerade in der Projektlaufzeit ergibt sich aufgrund der Ungenauigkeit ein relativ großes Intervall für das Minimum.

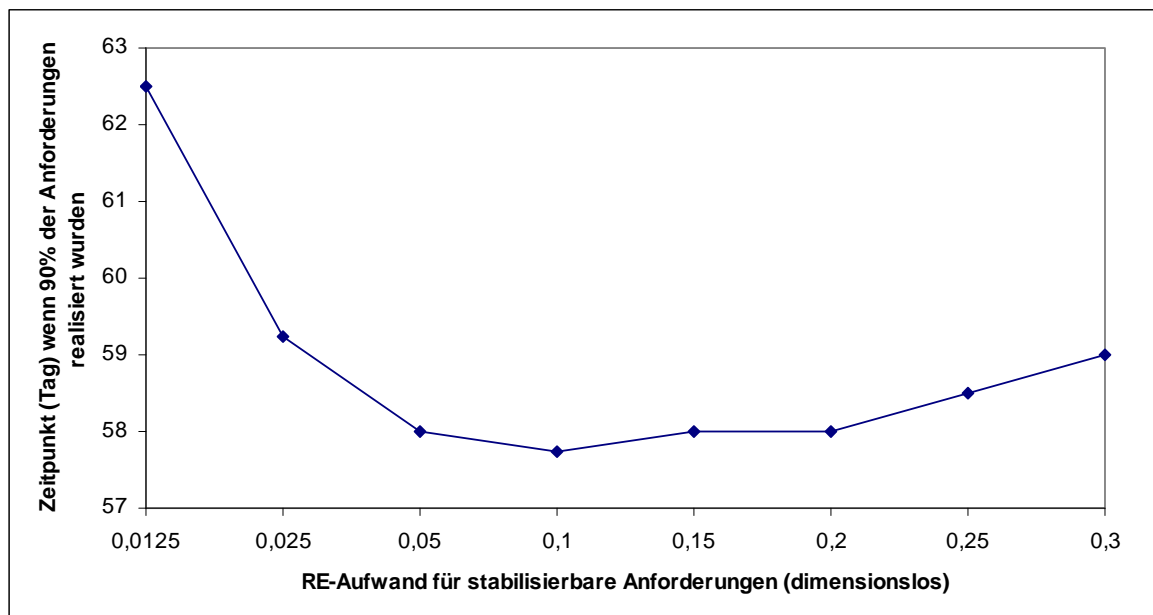


Abbildung 3-19: Simulation der Projektlaufzeit bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

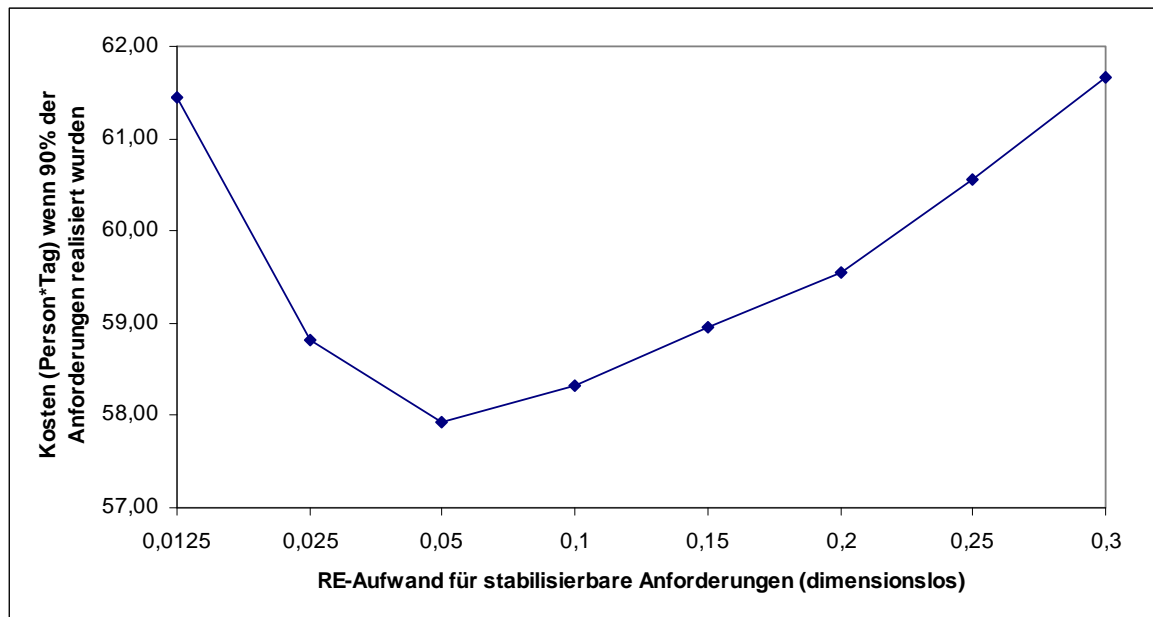


Abbildung 3-20: Simulation der Projektkosten bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

Abbildung 3-21 und Tabelle 3-4 zeigen die durch Simulation bestimmten Kosten für Anforderungsänderungen abhängig vom jeweils eingesetzten RE-Aufwand für stabilisierbare Anforderungen. Man erkennt deutlich, dass die Änderungskosten umso geringer sind, je mehr Aufwand in das RE investiert wird. Im Referenzfall von 0,2 Personentagen RE-Aufwand für eine Anforderung, deren Realisierung mit einem Personentag veranschlagt ist, ergeben sich insgesamt Änderungskosten von 5,5 Personentagen, was in etwa 9,3% des Gesamtaufwands entspricht. Erhöht man den RE-Aufwand für stabilisierbare Anforderungen um 25% so reduzieren sich die Änderungskosten auf 7,1% des Gesamtaufwands. Bei einer Reduzierung des RE-Aufwands für stabilisierbare Anforderungen um 50% steigen die Änderungskosten hingegen auf 15,1% des Gesamtaufwands. Die Gesamtprojektkosten schwanken abhängig vom Aufwand für stabilisierbare Anforderungen im beobachteten Bereich um bis zu 6,5%. Die Änderungskosten schwanken dabei sogar um den Faktor 5. Eine alleinige Variierung des Aufwands für stabilisierbare Anforderungen bietet demzufolge ein spürbares Optimierungspotenzial. Dies liegt vor allem daran, dass der im Simulationsmodell als optimal angezeigte Aufwand für stabilisierbare Anforderungen deutlich unter dem aktuell eingesetzten liegt. Reduziert man den Aufwand für stabilisierbare Anforderungen, so macht dieser im Vergleich zum Aufwand für die anderen Anforderungstypen nur noch einen relativ geringen Teil aus. Dementsprechend kann nur ein ebenso kleiner Teil der Kosten eingespart werden. In einem weiteren Experiment wollen wir untersuchen wie signifikant die Schwankung in den Gesamtprojektkosten ist, wenn der Aufwand für sämtliche Anforderungstypen variiert wird. Den Beobachtungen von Boehm und Basili (2001) zufolge entspricht der Aufwand für Änderungen in der Praxis im Schnitt 40-50% der Gesamtkosten, ist also deutlich höher als bei der FAUSER AG. Würde man diese Zahl zugrunde legen, so ergäbe sich bei der Variierung des Aufwands für stabilisierbare Anforderungen eine noch deutlich stärkere Schwankung in den Gesamtkosten. Es ist dabei jedoch zu beachten, dass die Zahl von Boehm und Basili sich nicht nur auf kleinere Projekte bezieht, sondern auch auf größere, in denen Änderungen an bereits realisierten Anforderungen deutlich teurer anzunehmen sind (Boehm und Basili 2001).

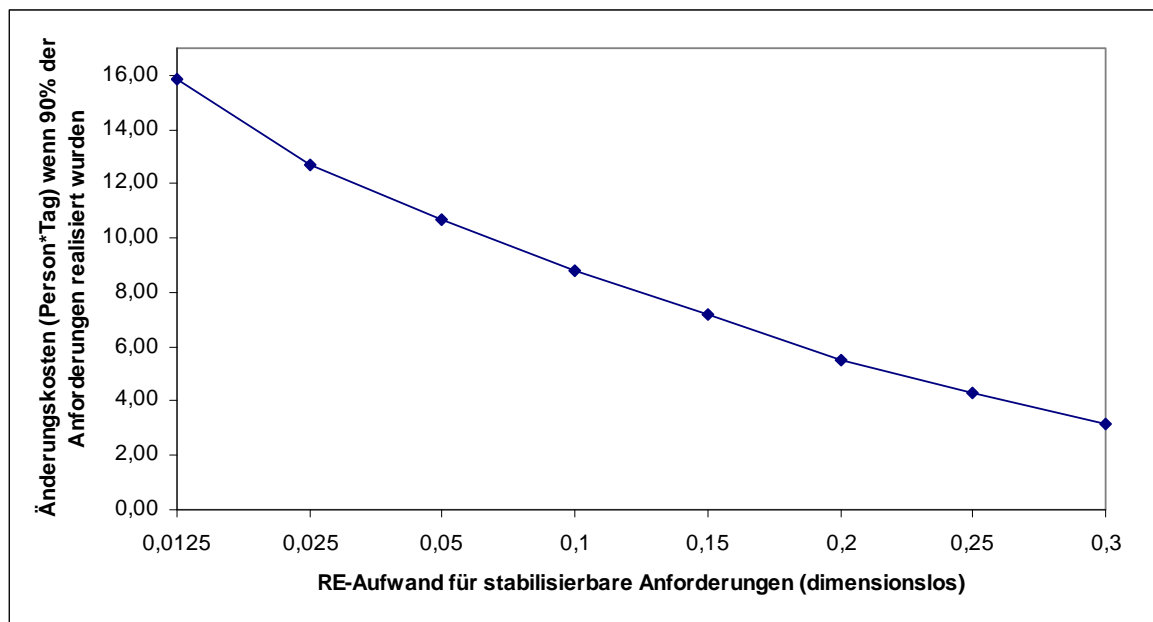


Abbildung 3-21: Simulation der Änderungskosten bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

Zum Realisierungsaufwand relativer Aufwand für die Analyse der stabilisierbaren Anforderungen	Anteil der Änderungskosten am Gesamtaufwand
0,0125	25,9%
0,025	21,6%
0,05	18,4%
0,1	15,1%
0,15	12,2%
0,2	9,3%
0,25	7,1%
0,3	5,1%

Tabelle 3-4: Simulation des Anteils der Änderungskosten am Gesamtaufwand bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

In einem weiteren Experiment variierten wir nicht nur den Aufwand für stabilisierbare Anforderungen, sondern auch den für stabile und nicht stabilisierbare Anforderungen. Die Ergebnisse sind in Abbildung 3-22 und Abbildung 3-23 dargestellt. Wiederum stellen sich Minima in Projektlaufzeit und –kosten für bestimmte RE-Aufwände ein. Die Schwankungsbreite in den Gesamtprojektkosten ist in diesem Fall wesentlich höher als bei einer alleinigen Variierung des Aufwands für stabilisierbare Anforderungen und liegt im beobachteten Bereich bei bis zu 23,7%. Die Änderungskosten schwanken im beobachteten Bereich nahezu um den Faktor 1,8. Das Simulationsmodell zeigt demzufolge ein signifikantes Optimierungspotenzial an.

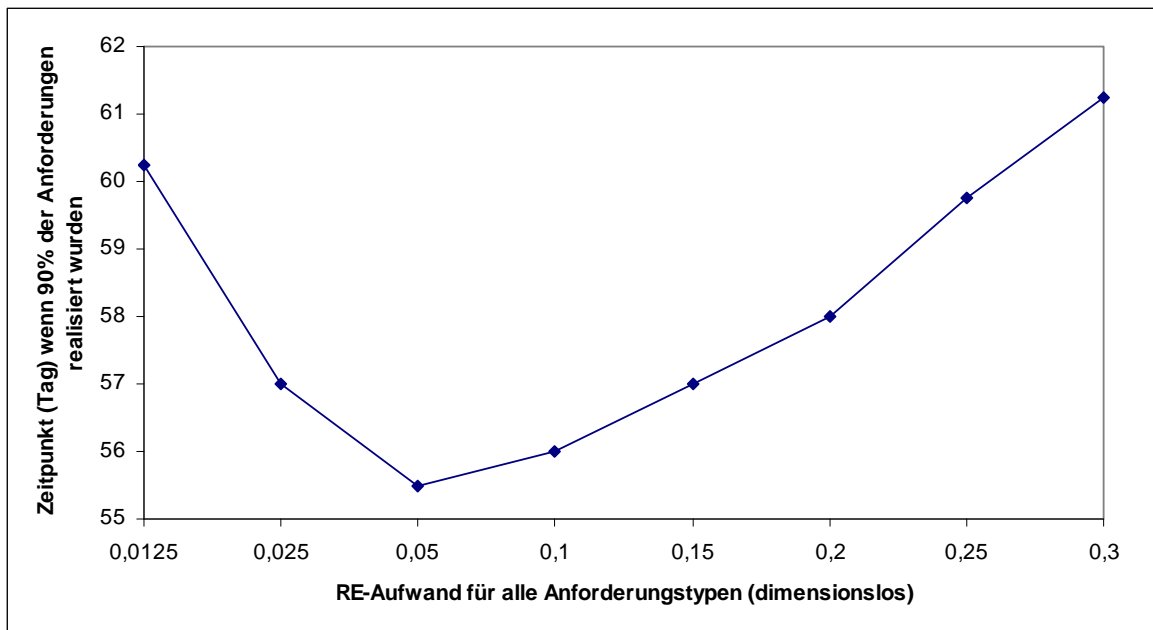


Abbildung 3-22: Simulation der Projektlaufzeit bei unterschiedlich hohem RE-Aufwand für alle Anforderungstypen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

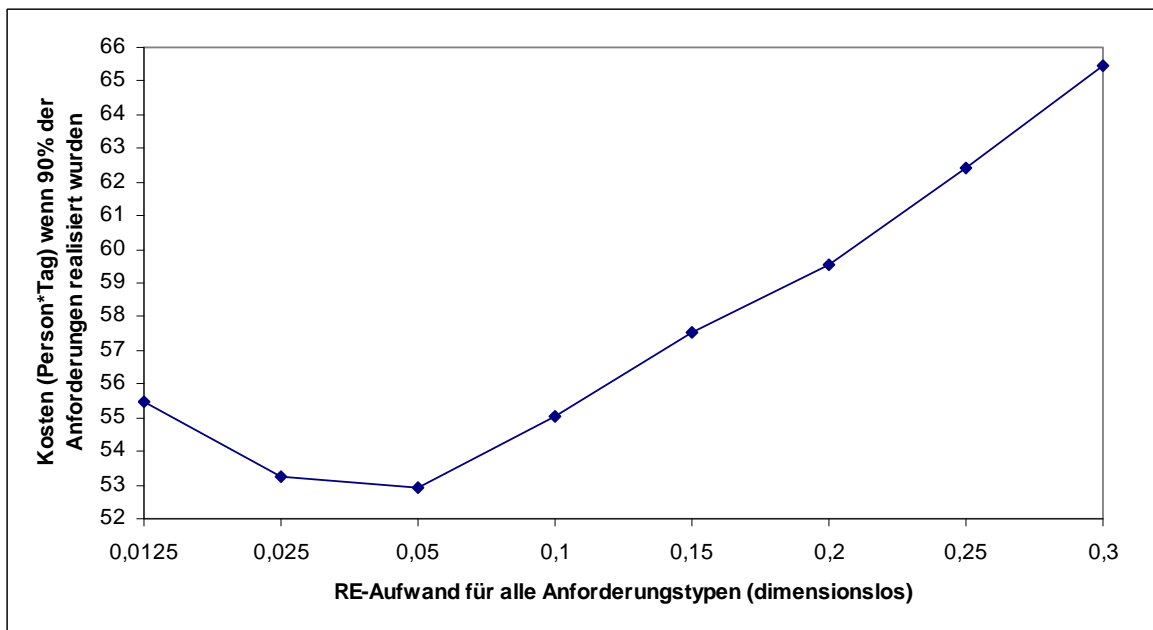


Abbildung 3-23: Simulation der Projektkosten bei unterschiedlich hohem RE-Aufwand für alle Anforderungstypen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

Auswirkung der Stabilisierbarkeit der Anforderungen auf den in Bezug auf Projektlaufzeit und Projektkosten optimalen RE-Aufwand für stabilisierbare Anforderungen

Weiterhin untersuchten wir die Auswirkung einer geringeren Stabilisierbarkeit der stabilisierbaren Anforderungen. Wir schwächten die Auswirkung des Verständnisses für

die Anforderungen auf die Absenkung der Änderungsrate ab, wie in Abbildung 3-24 gezeigt.

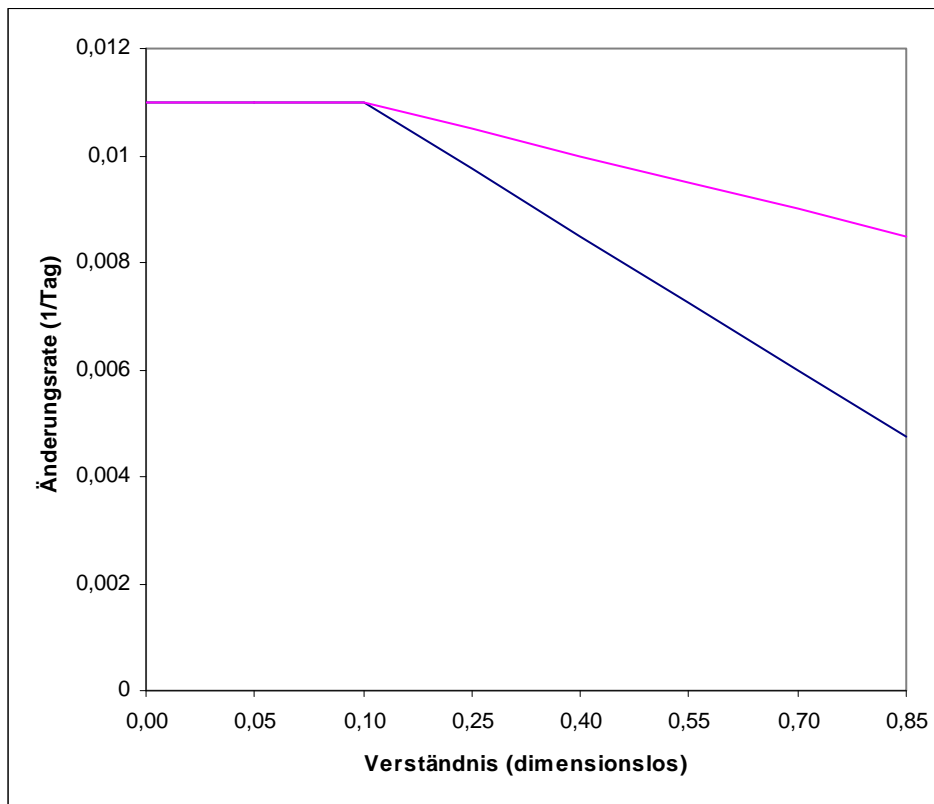


Abbildung 3-24: Änderungsrate der stabilisierbaren Anforderungen abhängig von deren Verständnis in der aktuellen Situation des betrachteten Projekts (schwarz) und bei geringerer Stabilisierbarkeit (rosa) (die Änderungsrate wird im Prozentsatz der Anforderungen, der sich pro Tag ändert, gemessen; das Verständnis ist eine Hilfsvariable, die bei der Umrechnung unter anderem von Analyseaufwand in resultierende Änderungsrate herangezogen wird, und ist dimensionslos)

Zunächst untersuchten wir welche Auswirkung die geringere Stabilisierbarkeit der stabilisierbaren Anforderungen auf die Projektlaufzeit hat. Die Simulation ergab einen um 2,25 Tage erhöhten Zeitbedarf bis zum Zeitpunkt an dem 85% der Anforderungen realisiert wurden. Es ist unmittelbar klar, dass eine geringere Stabilisierbarkeit der Anforderungen zu einer höheren Änderungsrate führt, damit mehr Änderungskosten anfallen und die Projektlaufzeit ansteigt. Dies wird durch das Modell richtig wiedergegeben.

Wir variierten wie zuvor den RE-Aufwand für stabilisierbare Anforderungen und untersuchten in welchem Bereich sich die lokalen Minima in der Projektlaufzeit beziehungsweise den Projektkosten einstellen.

Vergleicht man Abbildung 3-19 und Abbildung 3-20 mit Abbildung 3-25 und Abbildung 3-26 so lässt sich herauslesen, dass bei einer geringeren Stabilisierbarkeit der Anforderungen sich die Minima zu niedrigeren RE-Aufwänden hin verschieben. Je geringer der stabilisierende Effekt der Anforderungsanalyse, desto weniger lohnt es sich für diesen einen hohen Aufwand zu betreiben.

Die Beobachtung im Modell deckt sich mit unseren Erwartungen. Das Verhältnis zwischen Analyseaufwand und durch die Analyse gewonnenem Verständnis verläuft logarithmisch. Das beobachtete Minimum in der Projektlaufzeit entspricht dem Punkt, an dem eine

Steigerung des RE-Aufwands höher ist als die dadurch eingesparten Änderungskosten. Bei einer höheren Stabilisierbarkeit der Anforderungen bringt höheres Verständnis eine stärkere Einsparung in den Änderungskosten mit sich (vgl. Abbildung 3-24). Der Punkt, an dem sich zusätzlicher Analyseaufwand nicht mehr bezahlt macht, verschiebt sich also hin zu höheren Aufwänden. Aufgrund des logarithmischen Verlaufs des Verhältnisses zwischen Analyseaufwand und gewonnenem Verständnis gibt es jedoch stets ein Optimum.

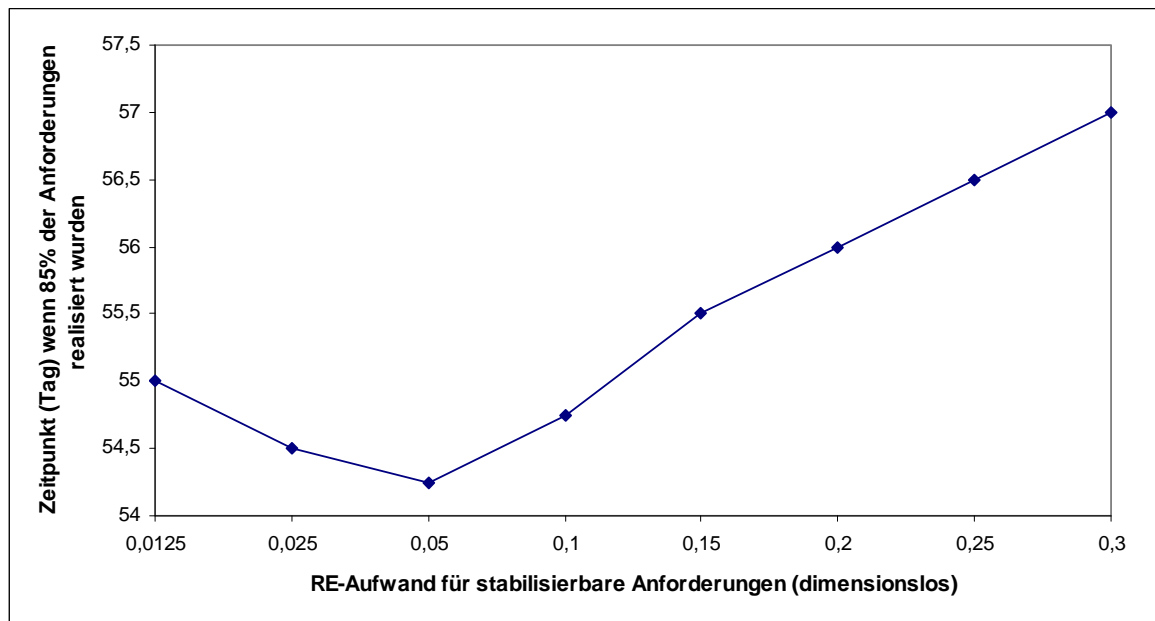


Abbildung 3-25: Simulation der Projektlaufzeit bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei geringerer Stabilisierbarkeit (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

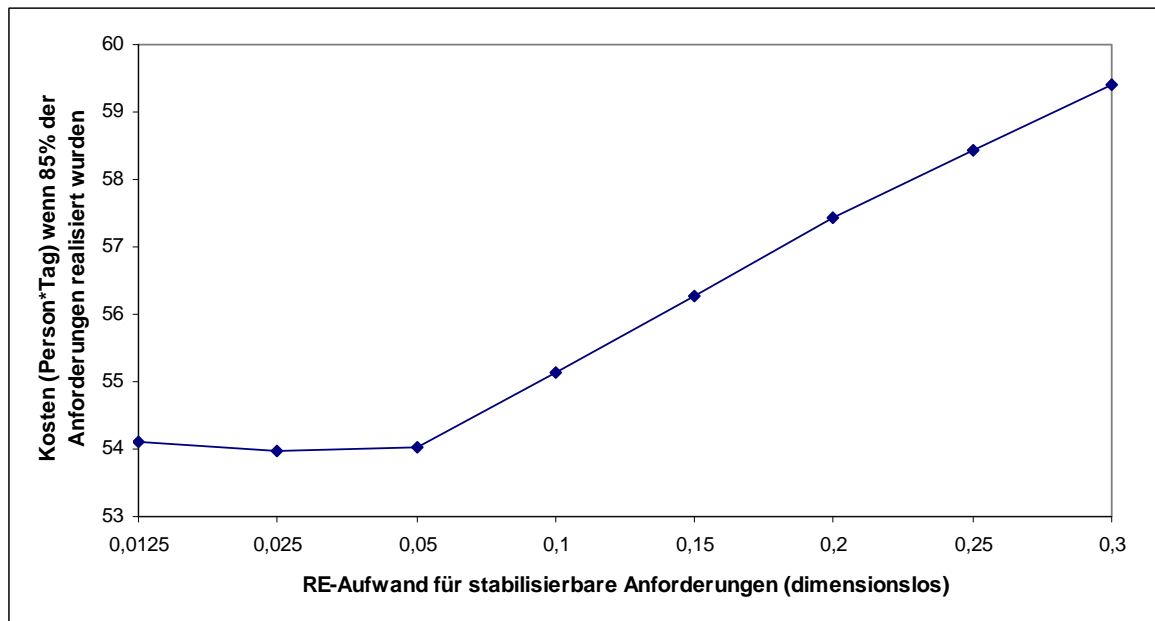


Abbildung 3-26: Simulation der Projektkosten bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei geringerer Stabilisierbarkeit (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

Auswirkung der Änderungsrate der stabilisierbaren Anforderungen auf den in Bezug auf Projektlaufzeit und Projektkosten optimalen RE-Aufwand für stabilisierbare Anforderungen

In weiteren Experimenten erhöhten wir die Änderungsrate der stabilisierbaren Anforderungen um 100% (Abbildung 3-27 und Abbildung 3-28) und 200% (Abbildung 3-29 und Abbildung 3-30) beziehungsweise senkten sie um 50% (Abbildung 3-31 und Abbildung 3-32). Wir führten jeweils eine Reihe von Simulationsläufen mit unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen durch.

Man kann erkennen, dass bei steigender Änderungsrate die beobachteten lokalen Minima in Projektlaufzeit und -kosten sich hin zu höheren RE-Aufwänden für stabilisierbare Anforderungen verschieben.

Dabei muss jedoch bedacht werden, dass eine Erhöhung der Änderungsrate in der Art und Weise, wie wir sie vorgenommen haben, gleichzeitig eine Erhöhung der Stabilisierbarkeit darstellt. Betrachtet man Abbildung 3-33 so kann man sich dies unmittelbar klarmachen. Erhöht man den Aufwand für die Anforderungsanalyse derart, dass das Verständnis für stabilisierbare Anforderungen von 0,1 auf 0,7 steigt, so wird normalerweise die Änderungsrate der stabilisierbaren Anforderungen von 1,1% auf 0,6% gesenkt. Im Fall der um 100% erhöhten Änderungsrate ergibt sich beim gleichen Mehraufwand für die Anforderungsanalyse eine Senkung der Änderungsrate von 2,2% auf 1,2%. Im ersten Fall entspricht die Senkung 0,5% im zweiten Fall 1%. Daher ist die Einsparung im Änderungsaufwand wesentlich höher. Entscheidend sind die absoluten Einsparungen in den Änderungskosten, die ein bestimmter RE-Aufwand mit sich führt. Sind die Änderungskosten sehr hoch so zieht wie in unserem Beispiel eine prozentuale Stabilisierung der Anforderungen eine höhere Einsparung nach sich als bei niedrigen Änderungskosten. Dieser Effekt wird in unserem Modell korrekt wiedergegeben. Laut den Experten der FAUSER AG führt im Regelfall eine Erhöhung der Instabilität stets zu dem

Effekt der gleichzeitigen Erhöhung der Stabilisierbarkeit. Wenn dem der Fall ist, gilt, dass bei zunehmend instabilen Anforderungen mehr Aufwand in deren Analyse investiert werden sollte, sofern sie stabilisierbar sind und eine Minimierung der Projektlaufzeit oder der Projektkosten angestrebt wird.

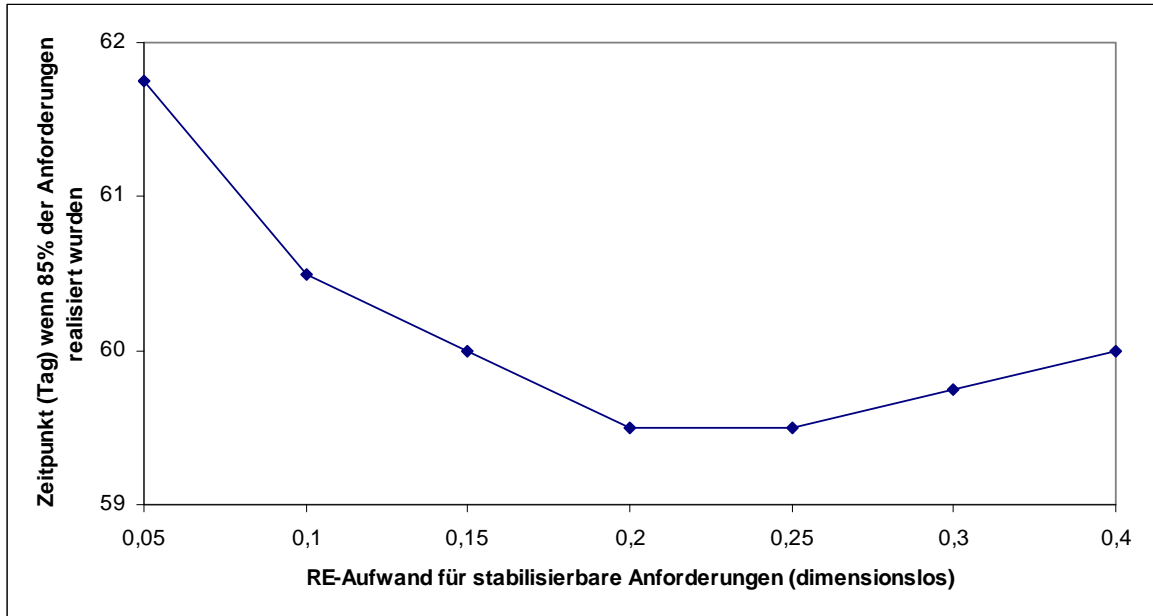


Abbildung 3-27: Simulation der Projektlaufzeit bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei 100% höherer Änderungsrate der stabilisierbaren Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

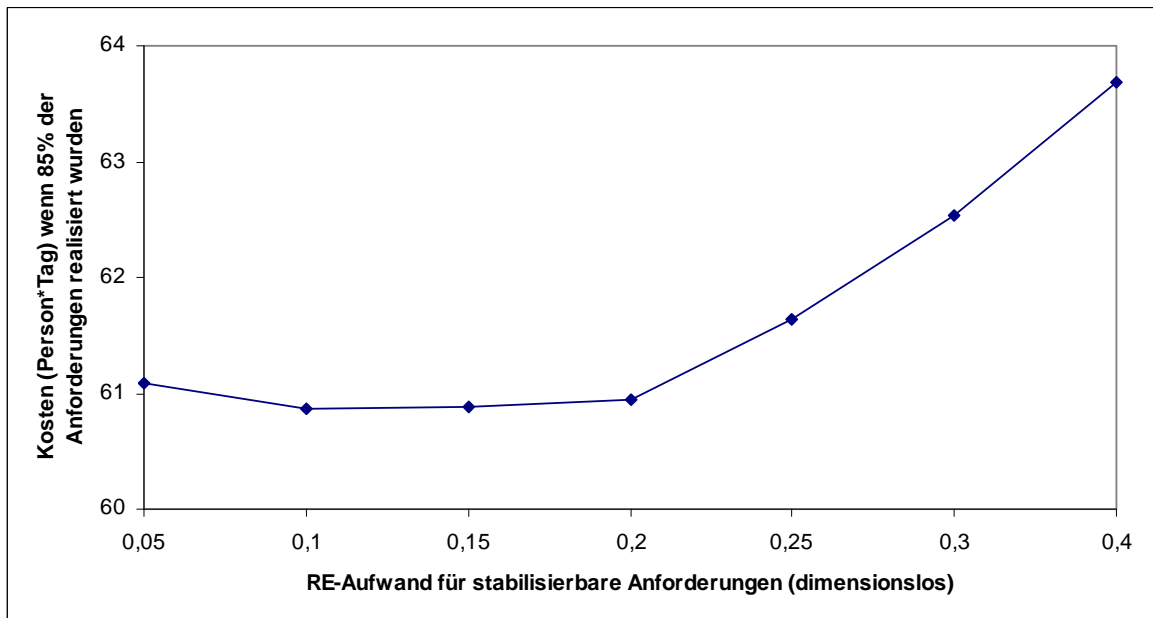


Abbildung 3-28: Simulation der Projektkosten bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei 100% höherer Änderungsrate der stabilisierbaren Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

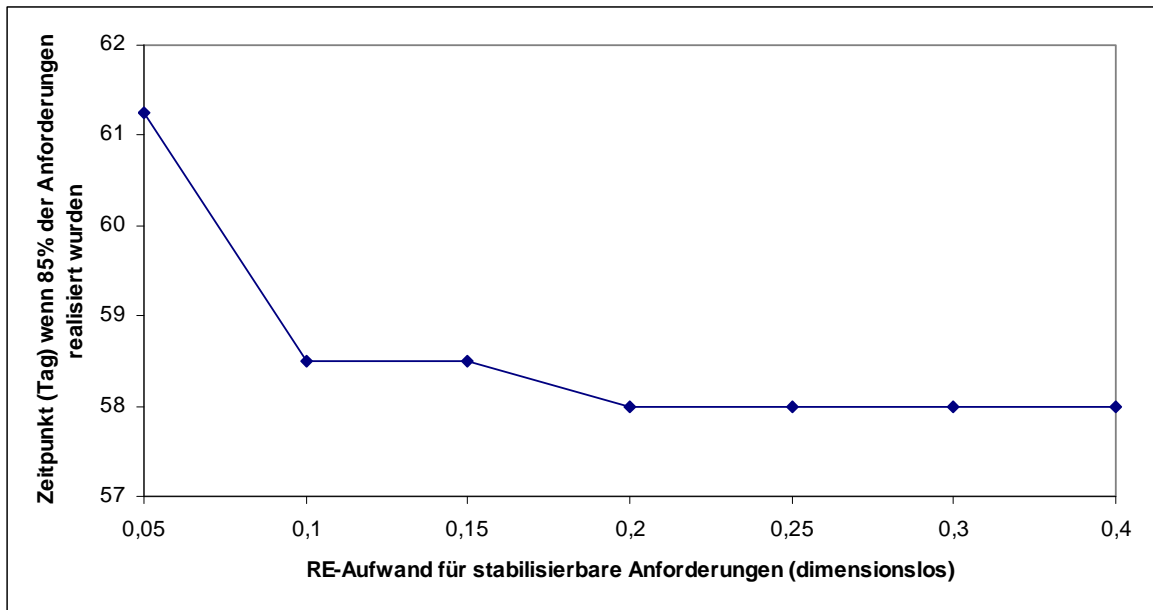


Abbildung 3-29: Simulation der Projektlaufzeit bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei 200% höherer Änderungsrate der stabilisierbaren Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

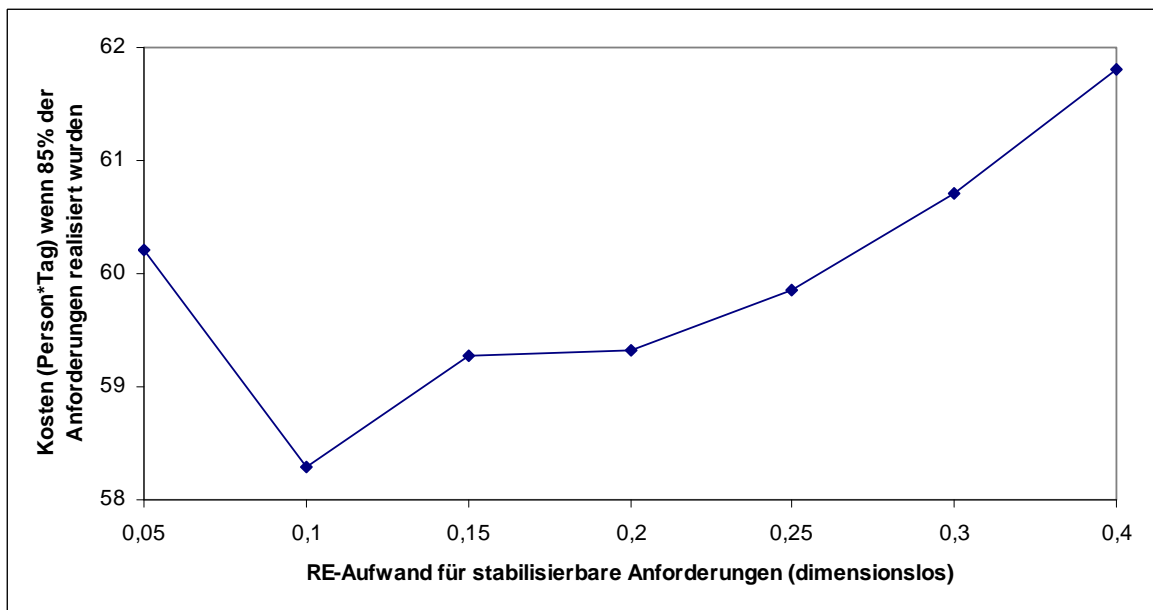


Abbildung 3-30: Simulation der Projektkosten bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei 200% höherer Änderungsrate der stabilisierbaren Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

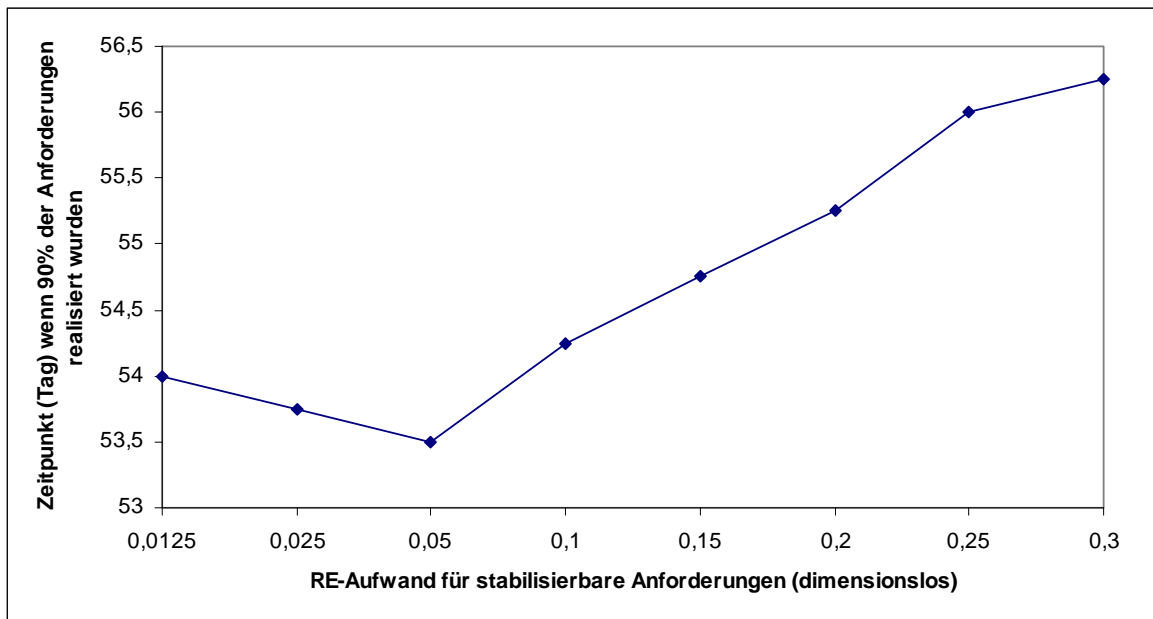


Abbildung 3-31: Simulation der Projektlaufzeit bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei 50% niedrigerer Änderungsrate der stabilisierbaren Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

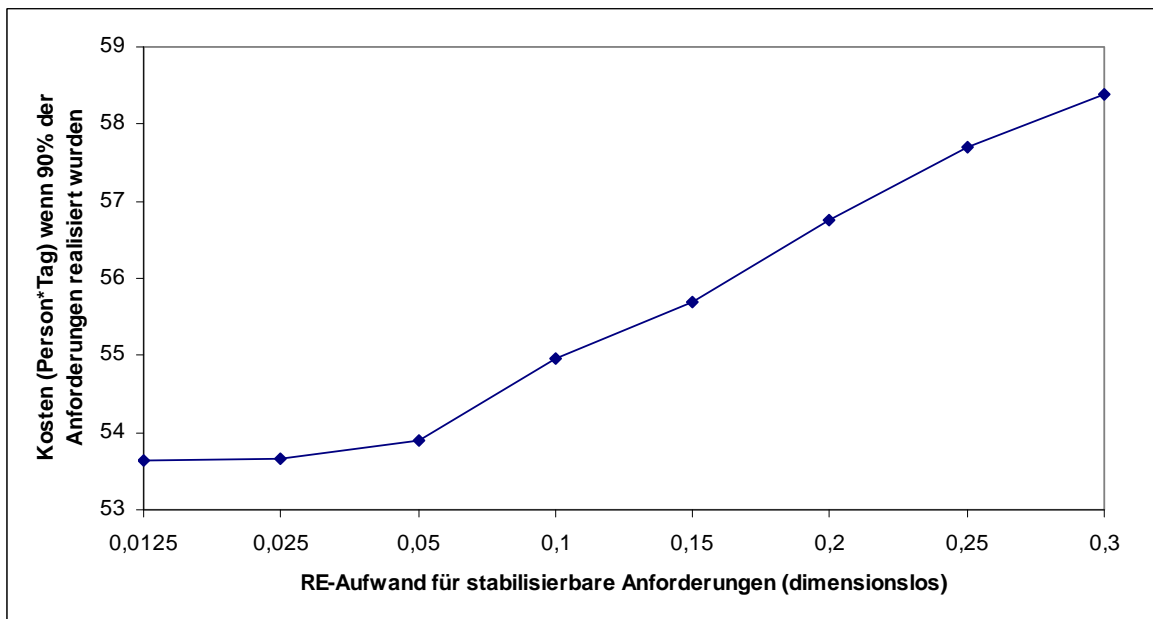


Abbildung 3-32: Simulation der Projektkosten bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei 50% niedrigerer Änderungsrate der stabilisierbaren Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

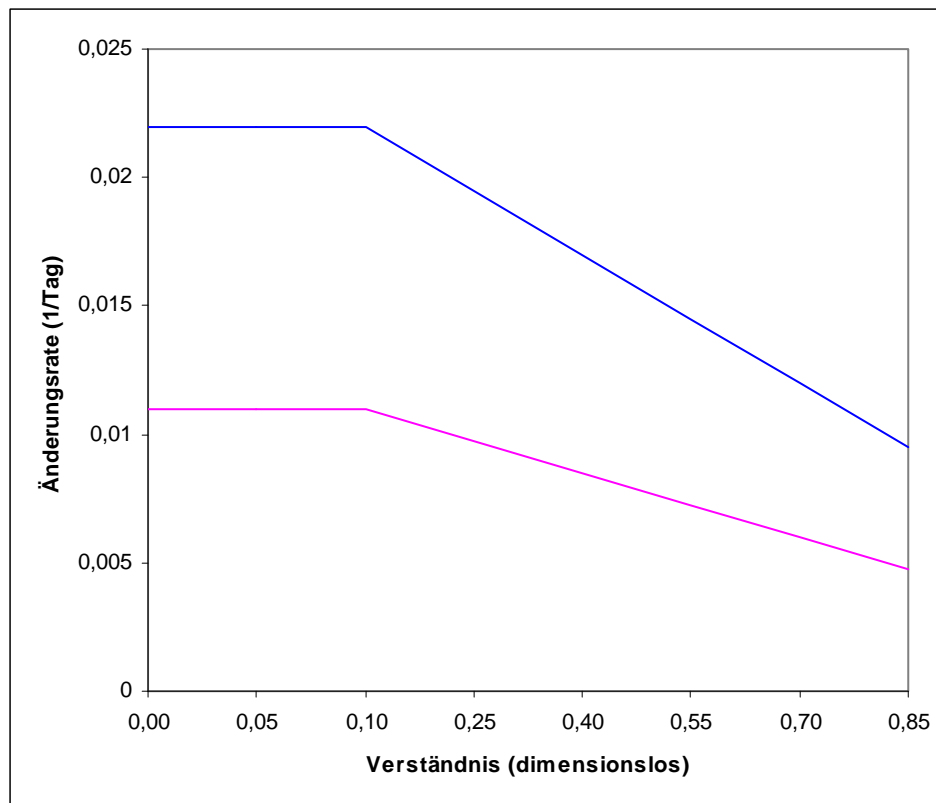


Abbildung 3-33: Änderungsrate abhängig vom Verständnis für die unveränderte Änderungsrate (rosa) und eine um 100% erhöhte Änderungsrate (blau) (die Änderungsrate wird im Prozentsatz der Anforderungen, der sich pro Tag ändert, gemessen; das Verständnis ist eine Hilfsvariable, die bei der Umrechnung unter anderem von Analyseaufwand in resultierende Änderungsrate herangezogen wird, und ist dimensionslos)

3.4.3 Zusammenfassung der Ergebnisse

Durch die Fallstudie bei der FAUSER AG gewinnen wir Erkenntnisse in Bezug auf die Fragen aus der Problembeschreibung.

Zum einen ist hervorzuheben, dass eine Variierung des Aufwands für RE sich in der Änderungsrate der Anforderungen niederschlägt. So kann durch eine Erhöhung des Analyseaufwands die Änderungsrate gesenkt werden. Dies ergibt sich nahezu unmittelbar aus den Eingaben seitens der Experten der FAUSER AG.

Eine geringere Änderungsrate bewirkt durch Einsparung von Änderungsaufwand eine Verkürzung der Projektlaufzeit und geringere Kosten. Da die Anforderungsanalyse selbst jedoch Zeit und Kosten in Anspruch nimmt, führt eine Erhöhung des RE-Aufwands nicht automatisch zu einem Zeit- und Kostengewinn. Wir konnten in unserer Simulation vielmehr zeigen, dass sich für bestimmte RE-Aufwände Minima in der Projektlaufzeit und den Projektkosten ergeben. Während wir bezüglich der für die Simulation aufgenommenen Parameter im Simulationsmodell diese Minima präzise bestimmen könnten, darf man jedoch nicht davon ausgehen, dass diese quantitativ mit der Wirklichkeit übereinstimmen. Gerade in Bezug auf die Beziehung zwischen RE-Aufwand und Verständnisgewinn sowie zwischen gewonnenem Verständnis für die Anforderungen und Änderungsrate konnten von den Experten der FAUSER AG nur ungefähre Abschätzungen vorgenommen werden. Darüber hinaus können die Beziehungen sich je nach Projekt quantitativ unterscheiden.

Daher können die genauen, mit dem Simulationsmodell festgestellten Minima weder verallgemeinert noch für das untersuchte Projekt als exakt angenommen werden. Bezüglich der im Simulationsmodell abgebildeten Grundmechanismen sowie des qualitativen Verlaufs der Beziehungen zwischen RE-Aufwand, Verständnis und Änderungsrate besteht jedoch Sicherheit. Daher können wir also schlussfolgern, dass sich im Allgemeinen stets Minima in Projektlaufzeit und Projektkosten bei bestimmten RE-Aufwänden einstellen, die exakten Parameter, für die sich die Minima einstellen, können jedoch nicht genau bestimmt werden. Aufgrund von TTM sollte angestrebt werden das Minimum in der Projektlaufzeit zu erreichen. Das Minimum in den Projektkosten ist im gewählten Beispiel zwar stark korreliert aber nicht immer deckungsgleich mit dem Minimum in der Projektlaufzeit. Dies führt dazu, dass zwischen Zeit und Kosten abgewogen werden muss. In der Praxis, ist es dennoch nicht immer sinnvoll sich direkt auf die so ermittelten Werte festzulegen. Es gibt weitere Parameter und Ziele die bedacht werden müssen. Insbesondere wegen der Qualität des Produkts müssen eventuell Abstriche gemacht werden. Der Aufwand für RE hat einen Einfluss auf die Abweichung des Produkts von den Bedürfnissen der Stakeholder. Ein Indikator für diese Abweichung ist die Änderungsrate am Ende des Projekts. Je höher die Änderungsrate zu diesem Zeitpunkt, desto höher ist erwartungsgemäß die Abweichung. Ein höherer RE-Aufwand führt also dazu, dass das System genauer den Bedürfnissen der Stakeholder entspricht. Es kann daher nötig sein vom Minimum der Projektlaufzeit abzuweichen. Es verbleibt also die Aufgabe je nach Projekt andere Interessen gegen TTM und Kostenminimierung abzuwägen. Ein weiterer zu beachtender Faktor ist, dass, wie bereits in Kapitel 3.4.1 erwähnt, ein gewisses Mindestmaß an RE-Aufwand für alle Anforderungen nötig ist. Wird dieses unterschritten, so ist die Validität des Simulationsmodells nicht mehr gewährleistet.

Wir haben gezeigt, dass bei einer geringeren Stabilisierbarkeit sich das Minimum von Projektlaufzeit und –kosten hin zu niedrigeren RE-Aufwänden verschiebt. Bei einer höheren Stabilisierbarkeit hingegen lohnt es sich im Hinblick auf TTM und Kostenminimierung mehr Aufwand in die Anforderungsanalyse zu investieren.

Weiterhin konnten wir feststellen, dass die Änderungsrate die beobachteten Minima von Projektlaufzeit und -kosten beeinflusst. Je höher die Änderungsrate, desto weiter verschieben sich die Minima hin zu höheren RE-Aufwänden. Dabei muss jedoch bedacht werden, dass wir davon ausgehen, dass, wie in Kapitel 3.4.2 erklärt, mit der Erhöhung der Änderungsrate eine höhere Stabilisierbarkeit der Anforderungen einhergeht. Letztendlich kann der hier beobachtete Effekt auf die zuvor beschriebene Abhängigkeit der Minima von der Stabilisierbarkeit der Anforderungen zurückgeführt werden.

Schließlich ist noch anzumerken, dass die aufgeführten Vorgehensweisen zur Minimierung von Projektlaufzeit und –kosten nur dann anwendbar sind, wenn der Anforderungstyp für den Großteil der Anforderungen richtig eingeschätzt werden kann, sowie deren Stabilisierbarkeit und die durchschnittliche Änderungsrate. Laut der FAUSER AG ist die Abschätzung des Anforderungstyps in ihren Projekten für die meisten Anforderungen kein Problem. Es ist nicht leicht Änderungsrate und Stabilisierbarkeit direkt anzugeben, doch kann in den meisten Projekten relativ leicht abgeschätzt werden, in welche Richtung Änderungsrate und Stabilisierbarkeit im Gegensatz zu bisherigen Projekten abweichen.

Die FAUSER AG stufte die qualitativen Ergebnisse des Simulationsmodells als richtig ein. Die Beteiligten konnten in der Fallstudie ein tieferes Verständnis für die Prozesse der FAUSER AG und die darunterliegenden Grundmechanismen gewinnen. Zudem wurde den Beteiligten bewusst, wie das Vorgehen im RE auf unterschiedliche Projektsituationen angepasst werden sollte.

Für die FAUSER AG ergeben sich aus der Simulation folgende Erkenntnisse:

1. Für Anforderungen, die als stabilisierbar eingestuft werden, sollten umfangreichere Analysen in der RE-Phase angestellt werden als für andere Anforderungen. Da als Hauptursache für Anforderungsänderungen die falsche Einschätzung ihrer Kosten genannt wurde, sollte entsprechend vor allem der Aufwand für Kostenschätzungen erhöht bzw. allgemein das Vorgehen in der Schätzung der Kosten von Anforderungen angepasst werden.
2. Wird die *Stabilisierbarkeit* der Anforderungen höher als in einem „durchschnittlichen“ Projekt eingeschätzt, so sind umfangreichere Analysen als üblich in der RE-Phase durchzuführen. Ist die Stabilisierbarkeit niedriger, sollte entsprechend weniger Analyseaufwand betrieben werden.
3. Ist der *Anteil der stabilisierbaren Anforderungen* in einem Projekt höher als in einem „durchschnittlichen“ Projekt, so stellt dies eine insgesamt höhere Stabilisierbarkeit der Anforderungen dar und es sollte entsprechend mehr Analyseaufwand in der RE-Phase aufgewendet werden. Analog wird weniger Analyseaufwand betrieben, wenn der Anteil der stabilisierbaren Anforderungen vergleichsweise niedrig ist.
4. Wird die *Änderungsrate der Anforderungen* in einem Projekt höher als in einem „durchschnittlichen“ Projekt eingeschätzt, so weist dies auf eine insgesamt höhere Stabilisierbarkeit der Anforderungen hin. In diesem Fall sollten umfangreichere Analysen als üblich in der RE-Phase durchgeführt werden. Wird die Änderungsrate hingegen niedriger als gewöhnlich eingeschätzt so sollte entsprechend weniger Analyseaufwand betrieben werden.

3.5 Fallstudie 2: PSIPENTA Software Systems GmbH

Das in Kapitel 3.4 entwickelte System Dynamics Modell wendeten wir in einer zweiten Fallstudie bei der PSIPENTA Software Systems GmbH (nachfolgend PSIPENTA genannt) an.

„Die PSIPENTA Software Systems GmbH ist 1997 als hundertprozentige Tochter aus der seit 1969 bestehenden PSI AG hervorgegangen. Sie ist wichtigster Produkt- und Technologie-Lieferant im Konzernsegment Produktionsmanagement. Als größte eigenständige Gesellschaft der PSI AG liefert sie Komplettlösungen für die Produktion (ERP), Logistik, den Anlagenbetrieb sowie das Supply Chain Management.“ (Broy et al. 2010)

3.5.1 Entwicklung des System Dynamics Modells

Für die Fallstudie bei PSIPENTA wurde das in Kapitel 3.4 entwickelte Simulationsmodell angepasst. Dies umfasste einerseits eine strukturelle Anpassung des Modells, um den betrachteten iterativen Softwareentwicklungsprozess abbilden zu können. Andererseits wurde eine Neukalibrierung durchgeführt. Bei der Anpassung des Simulationsmodells folgten wir den in Kapitel 3.2.2 beschriebenen Schritten zur Entwicklung von System Dynamics Modellen.

Die Datenerhebung erfolgte anhand mehrerer Interviews mit drei Softwareentwicklungsexperten und Projektmanagern von PSIPENTA. An der Verifikation und Validierung des Modells wurden dieselben Experten beteiligt.

Die erhobenen Daten beziehen sich nicht auf ein konkretes Projekt sondern geben wieder, was die befragten Experten unter einem typischen kleineren Projekt ihres Unternehmens verstehen. Sie stellen größtenteils intuitive Einschätzungen dar und basieren nicht auf dokumentierten Zahlen. In einem typischen kleineren Projekt bei PSIPENTA werden im Mittel zwei Personen am RE beteiligt. Das Entwicklerteam besteht im Mittel ebenfalls aus zwei Personen. Wir betrachten für die Simulation den Projektverlauf ab dem Projekt-Kick-Off. Ein Projekt durchläuft zwei ungefähr gleich lange Iterationen. Zu Beginn jeder Iteration findet eine RE-Phase gefolgt von einer Entwicklungsphase statt. In den Entwicklungsphasen besteht keine klare zeitliche Trennung von Entwurf, Implementierung und Test. Die Integration wird kontinuierlich durchgeführt. In der Entwicklungsphase der ersten Iteration werden Änderungen an Anforderungen zwar registriert jedoch nur vereinzelt beschlossen und umgesetzt. Während der Entwicklungsphase der zweiten Iteration wird über Anforderungsänderungen unmittelbar entschieden. Ihre Umsetzung erfolgt ebenfalls direkt in dieser Iteration.

Wir gehen für die Simulation davon aus, dass die Änderung bereits realisierter Anforderungen um 50% teurer ist als die Erstrealisierung der entsprechenden Anforderungen. Diese Zahl beruht auf der Einschätzung der Experten von PSIPENTA und liegt deutlich unter dem von der FAUSER AG sowie von Boehm und Basili (2001) für die Entwicklung von einfachen nicht kritischen Systemen angenommenen Wert.

Für die Erstellung und Ausführung des Simulationsmodells wurde das Werkzeug Vensim PLE von Ventana Systems (Ventana Systems 1997) eingesetzt.

Problembeschreibung

Die Fragen, die wir in der Fallstudie bei PSIPENTA mit dem System Dynamics Modell beantworten wollten, entsprachen denen aus der Fallstudie bei der FAUSER AG.

1. Welche Auswirkungen hat eine Variierung des Aufwands für RE auf die Änderungsrate?
2. Können durch eine geeignete Konzentration des RE-Aufwands auf bestimmte Anforderungstypen Vorteile hinsichtlich Projektlaufzeit und –kosten erlangt werden? Wir unterscheiden bei dieser Frage zwischen den idealisierten Anforderungstypen stabile, stabilisierbare und nicht stabilisierbare Anforderungen.
3. Können durch Variierung des Aufwands für RE Optima in Projektlaufzeit und Projektkosten erreicht werden?
4. Welche Auswirkungen hat die Stabilisierbarkeit von Anforderungen auf die, sich durch Variierung des RE-Aufwands einstellenden, Optima in Projektlaufzeit und Projektkosten? Wir betrachten hierbei die Stabilisierbarkeit der Anforderungen durch den Verständnisgewinn in der Anforderungsanalyse.
5. Welche Auswirkungen hat die Änderungsrate von Anforderungen auf die, sich durch Variierung des RE-Aufwands einstellenden, Optima in Projektlaufzeit und Projektkosten?

Definition des Referenzverhaltens

Die Experten von PSIPENTA gaben Referenzkurven für die Menge der realisierten Anforderungen und für die Menge der hinzukommenden oder geänderten Anforderungen pro Zeitspanne jeweils im Projektverlauf vor. Wie bereits in Kapitel 3.4.1 beschrieben, spielen diese Referenzkurven eine zentrale Rolle in der Validierung des Simulationsmodells.

Analog zur Fallstudie bei der FAUSER AG verstehen wir unter der Menge der realisierten Anforderungen nicht die Anzahl realisierter Anforderungen, sondern den mit ihrer Erstrealisierung assoziierten Aufwand in Personentagen. Wir folgen dabei dem Vorgehen von Pfahl und Lebsanft (2000), die bei der Entwicklung ihres Systems Dynamics Modells Mengen von Anforderungen ebenfalls in Form von damit verbundenem Realisierungsaufwand messen. Die Anzahl von Anforderungen ist kein geeignetes Maß, da je nach Granularität der Anforderungen sich stark unterschiedliche Aufwände für ihre Realisierung ergeben. Messen wir Mengen von Anforderungen hingegen in Form von damit verbundenem Realisierungsaufwand, so lässt sich gut darstellen, welcher Anteil des Systems bereits realisiert ist und welches Ausmaß Änderungen annehmen. Bei der betrachteten Projektart bei PSIPENTA existiert keine zeitliche Trennung zwischen Entwurf, Entwicklung und Test und die Integration wird kontinuierlich durchgeführt. Daher wird eine Anforderung genau dann als realisiert verstanden, wenn sie implementiert, integriert und getestet wurde.

Die Entwicklung bei PSIPENTA läuft in zwei Iterationen ab. Zu Beginn jeder Iteration findet eine RE-Phase statt. RE und Entwicklung laufen sequenziell ab. Nur in der Realisierungsphase der zweiten Iteration können RE-Aktivitäten zur Realisierung parallel ausgeführt werden. Die Menge der realisierten Anforderungen ist daher in der RE-Phase der ersten Iteration Null und verläuft ab Realisierungsbeginn bis zum Ende der ersten Iteration logarithmisch. Der logarithmische Verlauf lässt sich dadurch erklären, dass in den späteren Phasen einer Iteration verstärkt Änderungen an bereits realisierten Anforderungen durchgeführt werden. Es wird angenommen, dass die Kosten für die Änderung von bereits realisierten Anforderungen deutlich höher sind als die für deren Erstrealisierung. Durch den Aufwand für Änderungen bleiben weniger Kapazitäten für die Realisierung weiterer Anforderungen. Während der zweiten RE-Phase werden keine weiteren Anforderungen realisiert. In der zweiten Realisierungsphase entwickelt sich die Zahl der realisierten Anforderungen analog zur ersten Iteration wieder logarithmisch. Der Verlauf der Referenzkurve ist in Abbildung 3-34 dargestellt. Es ist zu beachten, dass die vorgegebene Referenzkurve nur als grobe Näherung verstanden werden darf. Sie stellt eine Einschätzung der Experten von PSIPENTA dar, wie sich die Menge der realisierten Anforderungen in einem typischen Projekt verhält. Sie stützt sich auf deren Erfahrung in zahlreichen ähnlichen Projekten, basiert jedoch nicht auf dokumentierten Zahlen. Über ihren qualitativen Verlauf herrscht große Sicherheit bei den Experten von PSIPENTA. Für die Validierung des Simulationsmodells ist somit die korrekte Wiedergabe des qualitativen Verlaufs entscheidend.

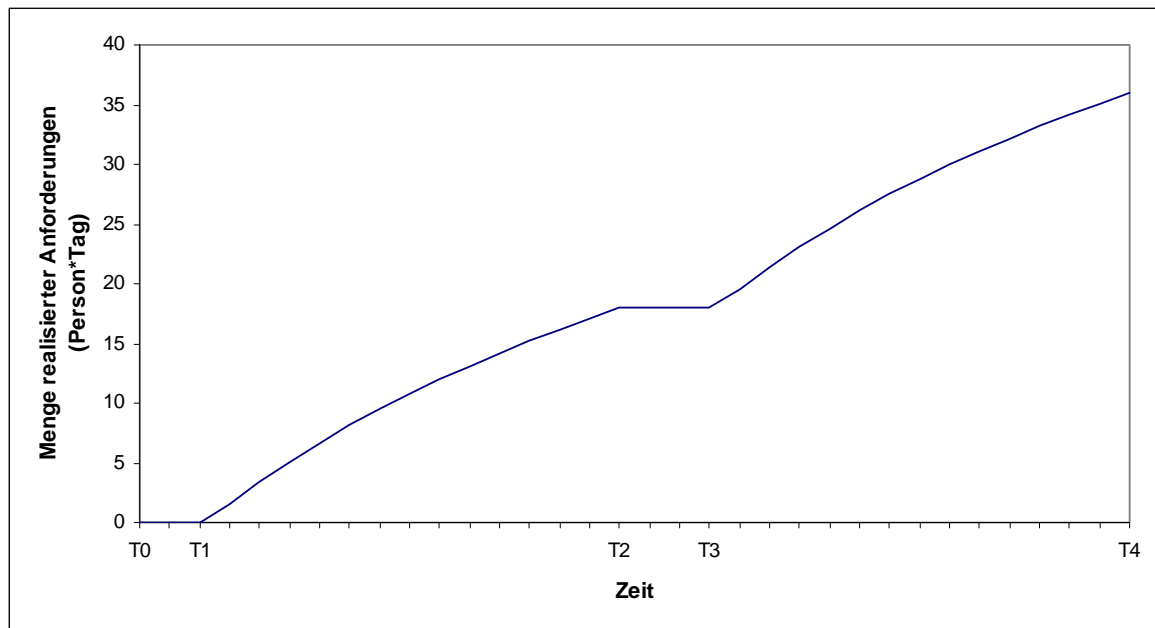


Abbildung 3-34: Referenzkurve für die Menge realisierter Anforderungen in einem typischen kleineren Projekt bei PSIPENTA (die Menge der Anforderungen wird in Form des für sie tatsächlich notwendigen Erstrealisierungsaufwands gemessen)

T0: Projekt-Kick-Off und Beginn der RE-Phase der ersten Iteration

T1: Ende der RE-Phase und Beginn der Realisierungsphase der ersten Iteration

T2: Ende erste Iteration und Beginn der RE-Phase der zweiten Iteration (ca. nach Ablauf der Hälfte der betrachteten Zeitspanne)

T3: Ende der RE-Phase und Beginn der Realisierungsphase der zweiten Iteration

T4: Entwicklungsende

Als zweites Referenzverhalten wurde von den Experten von PSIPENTA die Menge der hinzukommenden oder geänderten Anforderungen pro Zeitspanne im Projektverlauf vorgegeben. Wie bereits bei der ersten Referenzkurve erklärt, verwenden wir als Maßzahl für eine Menge von Anforderungen den mit ihrer Erstrealisierung verbundenen Aufwand in Personentagen. Daher ergibt sich für die Menge der hinzukommenden oder geänderten Anforderungen pro Zeitspanne im Modell die Einheit *Personentag/Tag*, also *Person*.

Der Verlauf der Referenzkurve für die Menge der hinzukommenden oder geänderten Anforderungen pro Zeitspanne ist in Abbildung 3-35 dargestellt. Während der initialen Anforderungsanalyse ist die Menge der hinzukommenden oder geänderten Anforderungen sehr groß, da zu Beginn des Projekts sich die Stakeholder erst über die gewünschten Anforderungen klar werden müssen und die Unsicherheit bezüglich des zu entwickelnden Systems noch recht hoch ist. Sie fällt dann während der Realisierungsphase der ersten Iteration auf einen sehr niedrigen Wert. Dies ist darauf zurückzuführen, dass Änderungswünsche zwar registriert werden, aber nur vereinzelt gleich beschlossen werden. Entscheidungen über die Annahme von Änderungen oder neuer Anforderungen werden hauptsächlich während der Analysephasen zu Beginn jeder Iteration getroffen. In der zweiten Analysephase ist die Zahl hinzukommender oder geänderter Anforderungen wieder sehr hoch. Dies spiegelt zum einen die Änderungswünsche wider, die sich im Laufe der ersten Realisierungsphase aufgestaut haben. Außerdem werden in der zweiten Analysephase Reviews der bisherigen Entwicklungsergebnisse durchgeführt. Dabei wird überprüft, ob diese den Wünschen der Stakeholder gerecht werden. Daraus ergeben sich in der Regel viele Änderungen an bestehenden Anforderungen und es werden zudem auch neue Anforderungen eingebracht. In der zweiten Realisierungsphase treten mehr

Änderungen als in der ersten Realisierungsphase auf, da diese nicht auf eine spätere Iteration verschoben sondern unmittelbar beschlossen und umgesetzt werden. Nach einer groben Schätzung ist die Summe der hinzukommenden und geänderten Anforderungen in beiden Iterationen ungefähr gleich. Wie schon die erste Referenzkurve stellt auch diese nur eine grobe Schätzung dar, die auf der Erfahrung der Experten von PSIPENTA mit zahlreichen ähnlichen Projekten basiert. Sie stellt eine intuitive Schätzung dar und stützt sich nicht auf dokumentierte Zahlen ab. Über den qualitativen Verlauf der Kurve besteht seitens der Experten von PSIPENTA große Sicherheit. Wie schon bei der FAUSER AG geht man bei PSIPENTA davon aus, dass bis Projektende noch Änderungen beschlossen werden, obwohl diese nicht mehr umgesetzt werden können. Dies kann so interpretiert werden, dass entsprechende Änderungen erst im nächsten Release umgesetzt werden. In der Validierung des Simulationsmodells ist die korrekte Wiedergabe des qualitativen Verlaufs der Referenzkurve zu überprüfen.

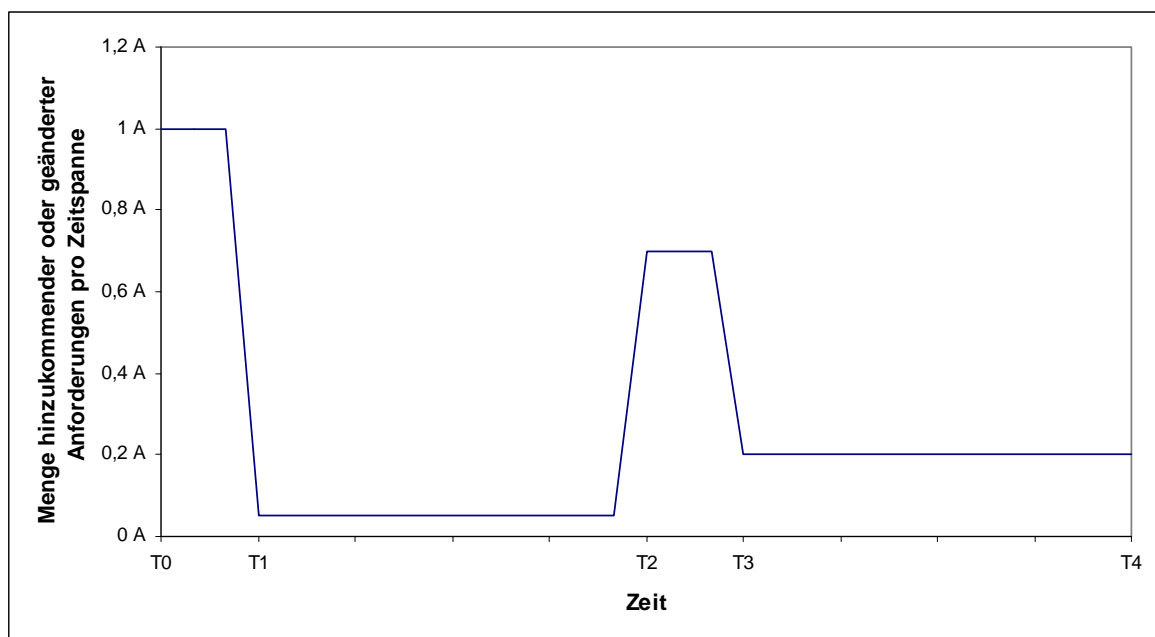


Abbildung 3-35: Referenzkurve für die Menge der hinzukommenden oder geänderten Anforderungen pro Zeitspanne in einem typischen kleineren Projekt bei PSIPENTA

A: Menge der in der RE-Phase der ersten Iteration hinzukommenden oder geänderten Anforderungen pro Zeitspanne (in beiden Iterationen treten ungefähr gleich viele hinzukommende oder geänderte Anforderungen auf)

T0: Projekt-Kick-Off und Beginn der RE-Phase der ersten Iteration

T1: Ende der RE-Phase und Beginn der Realisierungsphase der ersten Iteration

T2: Ende erste Iteration und Beginn der RE-Phase der zweiten Iteration (ca. nach Ablauf der Hälfte der betrachteten Zeitspanne)

T3: Ende der RE-Phase und Beginn der Realisierungsphase der zweiten Iteration

T4: Entwicklungsende

Identifikation der Grundmechanismen

Gemeinsam mit den Experten von PSIPENTA wurden die wichtigsten Grundmechanismen identifiziert, die das typische Verhalten des zu simulierenden Softwareentwicklungsprozesses erzeugen. Die identifizierten Grundmechanismen entsprechen größtenteils denen aus der Fallstudie bei der FAUSER AG. Die Experten von

PSIPENTA gaben jedoch an, dass das Verständnis für stabilisierbare Anforderungen nicht indirekt durch das Verständnis anderer Anforderungen gesteigert wird. Dementsprechend sind die zugehörigen Wirkbeziehungen in dieser Fallstudie ausgenommen. Es ergaben sich folgende als relevant eingestufte Grundmechanismen:

1. Aufwand für die Analyse der stabilisierbaren Anforderungen (+) → Verständnis der stabilisierbaren Anforderungen (+)
2. Aufwand für die Analyse der stabilisierbaren Anforderungen (+) → Aufwand für RE (+)
3. Aufwand für die Analyse der stabilen Anforderungen (+) → Aufwand für RE (+)
4. Aufwand für die Analyse der nicht stabilisierbaren Anforderungen (+) → Aufwand für RE (+)
5. Verständnis der stabilisierbaren Anforderungen (+) → Änderungsrate der stabilisierbaren Anforderungen (-)
6. Änderungsrate der stabilisierbaren Anforderungen (+) → Änderungen der stabilisierbaren Anforderungen (+)
7. Aufwand für RE (+) → Gesamtprojektkosten (+)
8. Aufwand für RE (+) → Projektdauer (+)
9. Aufwand für Entwicklung (+) → Gesamtprojektkosten (+)
10. Aufwand für Entwicklung (+) → Projektdauer (+)
11. Projektdauer (+) → Änderungen der stabilisierbaren Anforderungen (+)
12. Projektdauer (+) → Änderungen der nicht stabilisierbaren Anforderungen (+)
13. Änderungen der stabilisierbaren Anforderungen (+) → Aufwand für die Analyse der stabilisierbaren Anforderungen (+)
14. Änderungen der stabilisierbaren Anforderungen (+) → Aufwand für die Entwicklung (+)
15. Änderungen der nicht stabilisierbaren Anforderungen (+) → Aufwand für die Analyse der nicht stabilisierbaren Anforderungen (+)
16. Änderungen der nicht stabilisierbaren Anforderungen (+) → Aufwand für die Entwicklung (+)

Erstellung des Ursache-Wirkungs-Netzwerks

Aus der Zusammenführung der Grundmechanismen ergibt sich das in Abbildung 3-36 dargestellte Ursache-Wirkungs-Netzwerk.

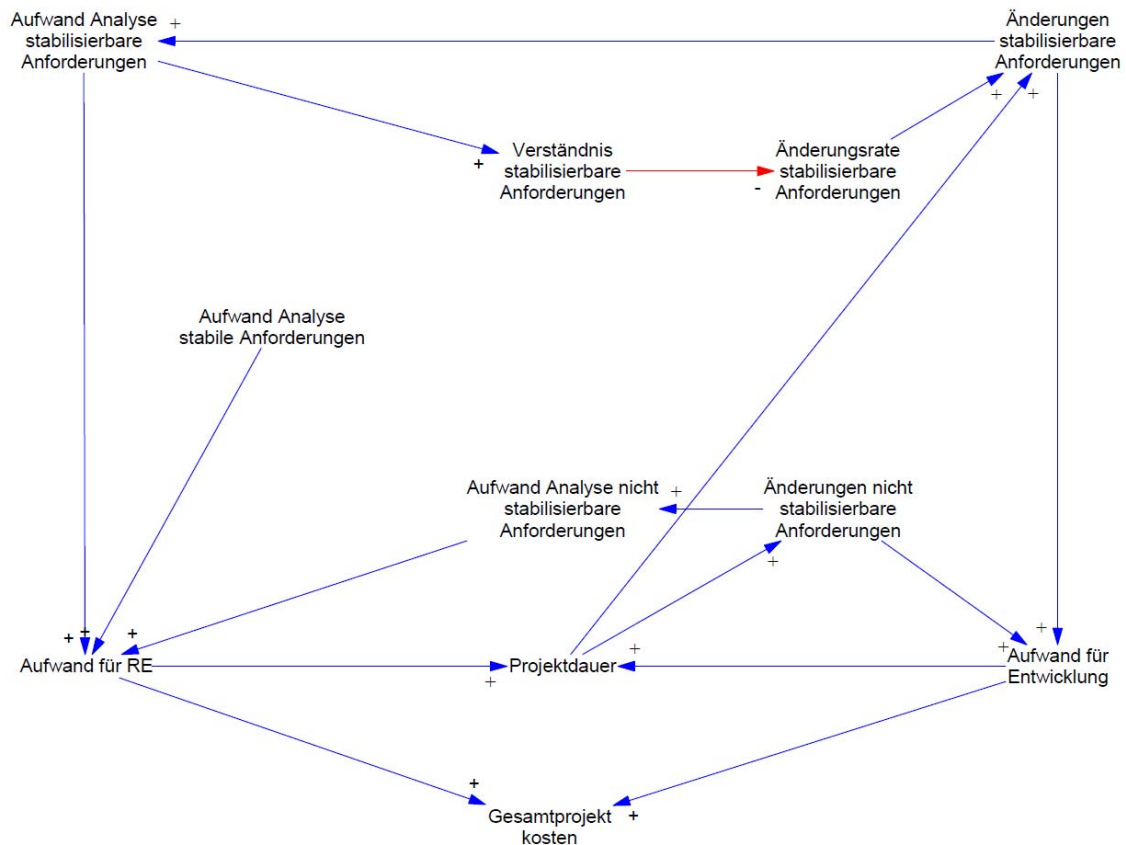


Abbildung 3-36: Ursache-Wirkungs-Netzwerk für die Fallstudie bei PSIPENTA

Über die Kombination der Grundmechanismen ergeben sich folgende Wirkungsketten positiver Polarität:

1. Aufwand für die Analyse der stabilisierbaren Anforderungen (+) → Aufwand für RE (+) → Projektdauer (+) → Änderungen der stabilisierbaren Anforderungen (+) → Aufwand für die Analyse der stabilisierbaren Anforderungen (+)
2. Änderungen der nicht stabilisierbaren Anforderungen (+) → Aufwand für die Entwicklung (+) → Projektdauer (+) → Änderungen der nicht stabilisierbaren Anforderungen (+)
3. Änderungen der stabilisierbaren Anforderungen (+) → Aufwand für die Entwicklung (+) → Projektdauer (+) → Änderungen der stabilisierbaren Anforderungen (+)
4. Änderungen der nicht stabilisierbaren Anforderungen (+) → Aufwand für die Analyse der nicht stabilisierbaren Anforderungen (+) → Aufwand für RE (+) → Projektdauer (+) → Änderungen der nicht stabilisierbaren Anforderungen (+)

Zudem ergibt sich folgende Wirkungskette negativer Polarität:

1. Aufwand für die Analyse der stabilisierbaren Anforderungen (+) → Verständnis der stabilisierbaren Anforderungen (+) → Änderungsrate der stabilisierbaren Anforderungen (-) → Änderungen der stabilisierbaren Anforderungen (-) → Aufwand für die Analyse der stabilisierbaren Anforderungen (-)

Aufstellen der mathematischen Gleichungen sowie Erstellung des Flussgraphen und Kalibrierung des Modells

Die Gleichungen, die das Modell vollständig beschreiben, finden sich in Anhang B.

Analog zum Modell aus der Fallstudie bei der FAUSER AG besteht das Modell aus sieben Modulen:

1. Entwicklung der stabilisierbaren Anforderungen (Abbildung 3-6)
2. Entwicklung der nicht stabilisierbaren Anforderungen (Abbildung 3-7)
3. Entwicklung der stabilen Anforderungen (Abbildung 3-8)
4. Beeinflussung der Änderungsrate (Abbildung 3-37)
5. Aufwandsverteilung (Abbildung 3-38 und Abbildung 3-39)
6. Kostenrechnung (Abbildung 3-12)
7. Referenzgrößen (Abbildung 3-13)

Für die Fallstudie bei PSIPENTA wurde das in der ersten Fallstudie entwickelte System Dynamics Modell als Grundlage verwendet und entsprechend kalibriert. PSIPENTA verwendet einen iterativen Entwicklungsprozess. Um diesen korrekt wiederzugeben waren strukturelle Änderungen im Simulationsmodell notwendig. Weiterhin wurde von den Experten von PSIPENTA angegeben, dass das Verständnis der stabilisierbaren Anforderungen nicht indirekt durch das Verständnis anderer Anforderungstypen gesteigert wird, wovon in der Fallstudie bei der FAUSER AG ausgegangen wurde. Die Änderungsrate der stabilisierbaren Anforderungen musste daher anders berechnet werden, was wiederum strukturelle Änderungen am Simulationsmodell nach sich zog. Somit mussten die Module „Beeinflussung der Änderungsrate“, „Aufwandsverteilung“ und „Referenzgrößen“ strukturell für die Fallstudie bei PSIPENTA angepasst werden.

Die Beschreibungen der Module 1-3 und 6 und 7 finden sich in Kapitel 3.4.1.

Das Modul „Beeinflussung der Änderungsrate“ dient der Ermittlung der Änderungsrate der stabilisierbaren Anforderungen. Die Berechnung der Änderungsrate erfolgt wie bereits in der ersten Fallstudie über Hilfsvariablen, die das Verständnis der stabilisierbaren Anforderungen darstellen. Über diese abstrakte Größe erfolgt in zwei Schritten die Ableitung der Änderungsrate für stabilisierbare Anforderungen aus dem Aufwand für die Analyse der stabilisierbaren Anforderungen und dem Entwicklungsstatus der Anforderungen. Anders als bei der Fallstudie bei der FAUSER AG wird der Anteil der bereits analysierten Anforderungen nicht in die Rechnung mit einbezogen. Die Experten von PSIPENTA nehmen an, dass eine indirekte Steigerung des Verständnisses der stabilisierbaren Anforderungen durch das Verständnis anderer Anforderungen vernachlässigbar ist. Bei der Berechnung des Verständnisses der stabilisierbaren Anforderungen werden der Verständnisgewinn durch die Anforderungsanalyse sowie Lerneffekte während der Realisierung berücksichtigt. Präzise Werte für das Verhältnis zwischen Analyseaufwand, Verständnis und Änderungsrate konnten nicht mit Sicherheit angegeben werden. Die Experten von PSIPENTA stellten lediglich grobe Schätzungen an. Bezüglich des qualitativen Verlaufs des Verhältnisses bestand hingegen große Sicherheit. Es wurde von den Experten angegeben, dass bei steigendem Aufwand für die Analyse von stabilisierbaren Anforderungen das Verständnis für diese logarithmisch zunimmt. Den Hauptbeitrag hinsichtlich des Verständnisgewinns sehen sie in der Anforderungsanalyse. Lerneffekte während der Realisierung tragen in geringerem Umfang bei. Anders als in der Fallstudie bei der FAUSER AG gingen die Experten von PSIPENTA nicht von einem

linearen Verhältnis zwischen Verständnis der stabilisierbaren Anforderungen und ihrer Änderungsrate aus. Bei steigendem Verständnis gehen sie zunächst von einer starken Absenkung der Änderungsrate aus. Diese Absenkung flacht jedoch zunehmend ab. Wie auch in der ersten Fallstudie wurde ein gewisses Mindestlevel an Verständnis als gegeben angenommen. Bei Unterschreitung dieses Levels kann die Änderungsrate enorm abweichen. Daher ist in der Simulation zu beachten, dass dieses Minimum nicht unterschritten wird. Insgesamt ergibt sich trotz der Abweichungen ein ähnliches Bild wie bei der FAUSER AG wenn man das Verhältnis von Analyseaufwand für stabilisierbare Anforderungen und ihrer Änderungsrate betrachtet. Steigt der Aufwand so fällt die Änderungsrate zunächst stark ab. Die Absenkung flacht aber zunehmend ab.

Wie bereits in der ersten Fallstudie führten wir aufgrund der Unsicherheit bzgl. des genauen Verhältnisses zwischen Analyseaufwand, Verständnis und Änderungsrate der stabilisierbaren Anforderungen mehrere Tests durch. In diesen wurde unter Beibehaltung des qualitativen Verlaufs des Verhältnisses dessen quantitativer Verlauf variiert. Aus den Ergebnissen der Tests konnte geschlussfolgert werden, dass trotz der Unsicherheit in diesen Eingabeparametern die Aussagen, die wir aus der Simulation ziehen, aufrechterhalten werden können.

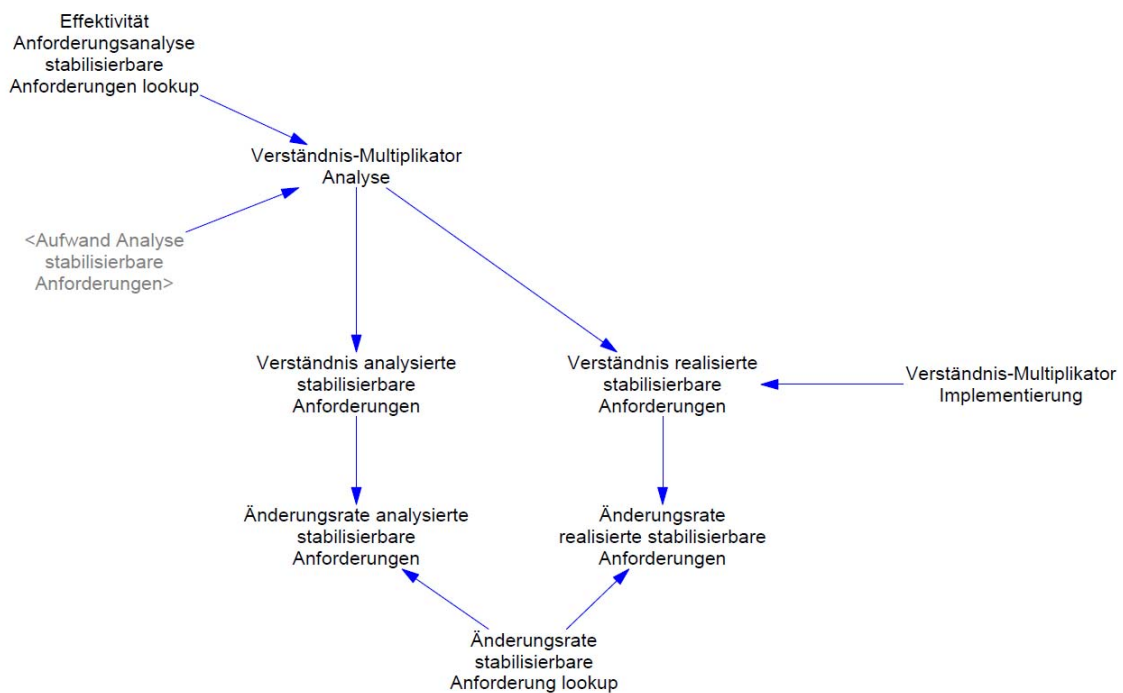


Abbildung 3-37: Flussgraph des Moduls „Beeinflussung der Änderungsrate“

Im angepassten Modul „Aufwandsverteilung“ werden weiterhin die Analyse- und Änderungsrate der einzelnen Anforderungstypen bestimmt. Im Gegensatz zur Fallstudie bei der FAUSER AG war bei PSIPENTA ein Softwareentwicklungsprozess mit zwei Iterationen abzubilden. Aus diesem Grund waren einige Anpassungen nötig, damit das Modul korrekt bestimmt zu welchem Zeitpunkt Anforderungsanalyse und Realisierung erfolgen. Der Flussgraph des Moduls wird in Abbildung 3-38 dargestellt. Der Übersichtlichkeit halber zeigt Abbildung 3-39 den Ausschnitt aus dem Modul, der die

Berechnung der Analyse- und Realisierungsrate der stabilisierbaren Anforderungen betrifft. Für die Ermittlung von Analyse- und Realisierungsraten wurde entsprechend der Vorgaben der Experten von PSIPENTA angenommen, dass die Reihenfolge, in der Anforderungen analysiert beziehungsweise realisiert werden zufällig ist, also nicht davon abhängt, ob eine Anforderung stabil, stabilisierbar oder nicht stabilisierbar ist. Analog zur Fallstudie bei der FAUSER AG wurde angenommen, dass Änderungen an bereits realisierten Anforderungen im Vergleich zu ihrer Erstrealisierung deutlich teurer sind. Demzufolge fallen die zugehörigen Realisierungsraten geringer aus.

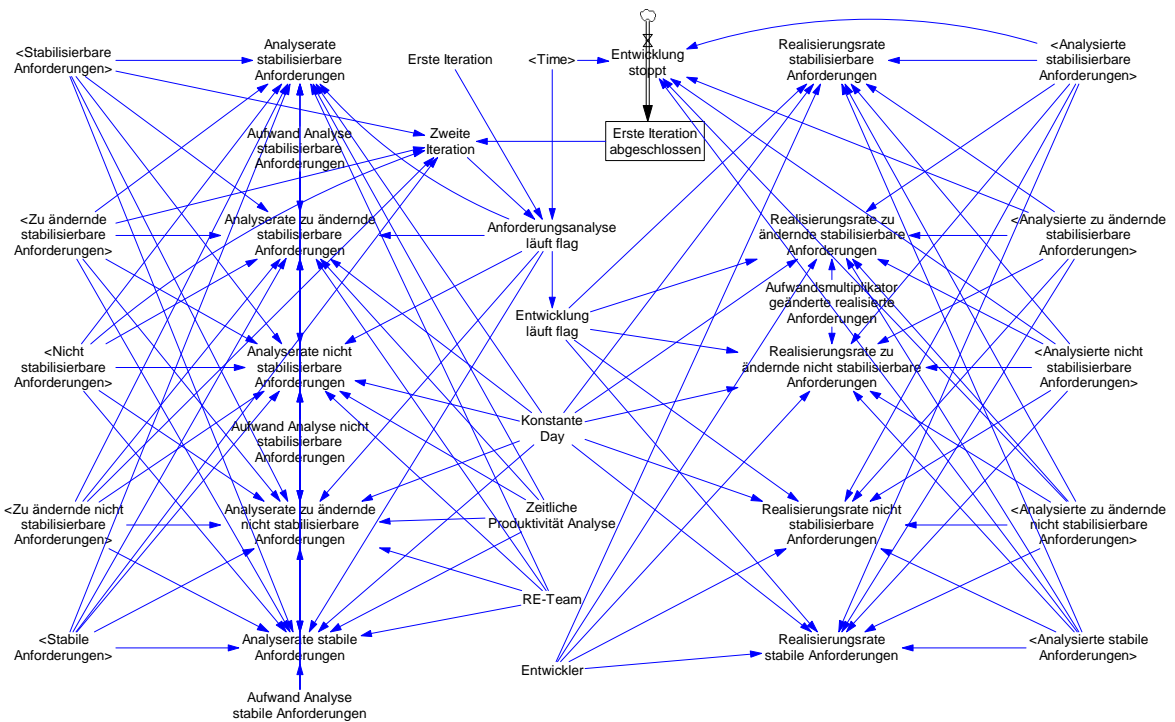


Abbildung 3-38: Flussgraph des Moduls „Aufwandsverteilung“

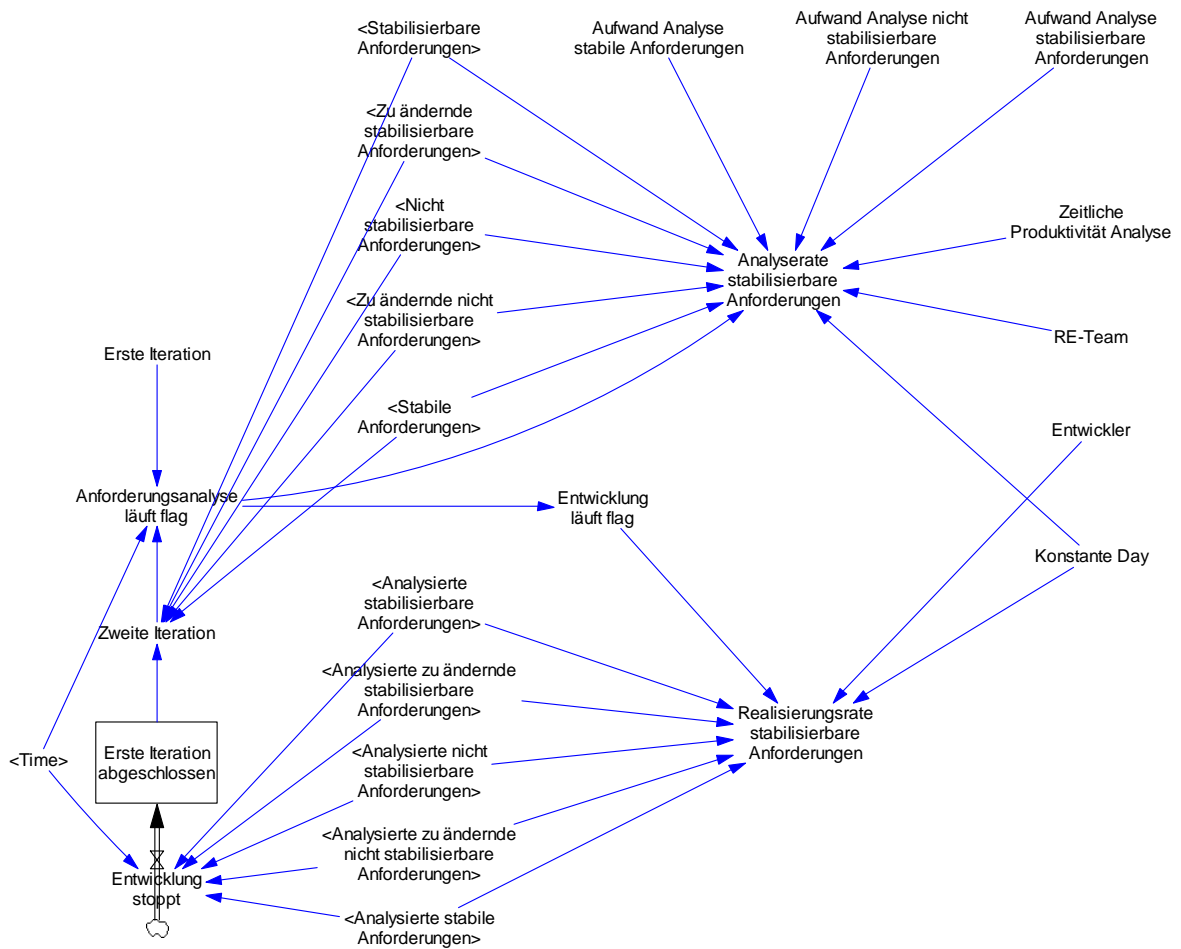


Abbildung 3-39: Ausschnitt aus dem Flussgraphen des Moduls „Aufwandsverteilung“

In Tabelle 3-5 sind einige zentrale Eingabeparameter für die Simulation aufgeführt. Eine Unterscheidung zwischen stabilisierbaren Anforderungen und nicht stabilisierbaren Anforderungen ist bei PSIPENTA bereits Praxis. Laut der Experten von PSIPENTA kann mit relativ großer Sicherheit korrekt angegeben werden, ob eine Anforderung stabil, stabilisierbar oder nicht stabilisierbar ist.

Parameter	Wert
Anteil stabiler Anforderungen	50%
Anteil stabilisierbarer Anforderungen	35%
Anteil nicht stabilisierbarer Anforderungen	15%
Anteil der stabilisierbaren Anforderungen, der im Projektverlauf von Änderungen betroffen ist	2/3
Anteil der nicht stabilisierbaren Anforderungen, der im Projektverlauf von Änderungen betroffen ist	10%
Anteil des RE-Aufwands am Gesamtentwicklungsaufwand für eine stabile Anforderung	1/11
Anteil des RE-Aufwands am Gesamtentwicklungsaufwand für eine stabilisierbare Anforderung	1/6
Anteil des RE-Aufwands am Gesamtentwicklungsaufwand für eine nicht stabilisierbare Anforderung	1/11
Durchschnittlicher Faktor, um den die Durchführung einer Änderung einer bereits realisierten Anforderungen teurer ist als deren Erstrealisierung	1,5

Tabelle 3-5: Zentrale Eingabeparameter für das Simulationsmodell bei PSIPENTA

Verifikation und Validierung des Modells

Zur Verifikation des Modells wurden in einem ersten Schritt erneut die direkten strukturellen Tests aus der ersten Fallstudie durchgeführt, die sicherstellen sollen, dass die Grundmechanismen vom Modell korrekt abgebildet werden. Diesbezüglich wurde für jeden Grundmechanismus der betreffende Ausschnitt des Modells einem Review unterzogen. Der Schwerpunkt wurde dabei auf geänderte Teile des Modells gelegt. Die Durchführung der Reviews folgte dem in Kapitel 3.4.1 beschriebenen Vorgehen.

Die Konsistenz der Einheiten der Modellgrößen wurde werkzeuggestützt überprüft.

Zusammen mit den Experten von PSIPENTA wurden Sensitivitätsanalysen für bestimmte Parameter durchgeführt. Das Verhalten des Modells wurde bei der Änderung der Parameter, auch für Extremwerte, beobachtet und hinsichtlich seiner Realitätsnähe bewertet. Wie bereits in der Fallstudie bei der FAUSER AG untersuchten wir Parameter, die

- die Menge der zu bearbeitenden Anforderungen festlegen,
- die Unterteilung der Anforderungen in stabilisierbare, nicht stabilisierbare und stabile Anforderungen festlegen,
- die Änderungsrate festlegen,
- den Aufwand für die Anforderungsanalyse festlegen und
- die Kosten von Änderungen an bereits realisierten Anforderungen festlegen

Wie auch schon in der Fallstudie bei der FAUSER AG wurde festgestellt, dass das Simulationsmodell bei extrem niedrigen RE-Aufwänden keine sinnvollen Ergebnisse ausgibt. Das Modell erlaubt eine Reduzierung des RE-Aufwands auf unrealistisch niedrige Werte wie 0,01% des Gesamtaufwands. Eine solch starke Reduzierung wird von den Experten von PSIPENTA als nicht möglich angesehen, da sonst keine sinnvolle Realisierung erfolgen kann. Zudem kann bei einem so niedrigen RE-Aufwand die idealisierte Betrachtung von nicht stabilisierbaren und stabilen Anforderungen auch in

Näherung nicht mehr aufrecht erhalten werden, da aufgrund stark mangelndem Verständnis nahezu alle Anforderungen als instabil und zu einem gewissen Grad als stabilisierbar angesehen werden müssen. Das für eine sinnvolle Realisierung und eine Trennung der verschiedenen Anforderungstypen nötige Mindestmaß an RE kann laut den Experten von PSIPENTA allgemein jedoch nicht angegeben werden, da es stark abhängig vom bereits vor dem Projektstart vorhandenen Grundverständnis für die Anforderungen ist. Es muss bei der Durchführung von Experimenten mit dem Simulationsmodell stets bewertet werden, ob der notwendige Mindestaufwand im RE eingehalten wurde. Das Verhalten des System Dynamics Modells bei der Variierung der anderen untersuchten Parameter wurde von den Experten von PSIPENTA als realistisch eingestuft.

Für die Validierung des Modells kamen erneut direkte strukturelle Tests und Verhaltenstests zum Einsatz.

Zunächst wurde in Zusammenarbeit mit den Experten von PSIPENTA untersucht, ob sich alle wesentlichen, in der Realität beobachteten und angenommenen kausalen Zusammenhänge im Modell wiederfinden. Andererseits wurde überprüft ob sich alle im Simulationsmodell inbegriffenen kausalen Zusammenhänge aus der Realität ableiten.

Weiterhin wurde gemeinsam mit den Experten von PSIPENTA überprüft, ob die eingestellten Modellparameter korrekt sind. Besonderes Augenmerk fand hier die Überprüfung, wie sich Unsicherheiten in den Eingangsgrößen im Modell auswirken. In Bezug auf das quantitative Verhältnis zwischen Aufwand für die Analyse stabilisierbarer Anforderungen, Lerneffekten während der Realisierung und Verständnis der stabilisierbaren Anforderungen beziehungsweise ihrer Änderungsrate herrschte Unsicherheit seitens der Experten von PSIPENTA. Qualitative Aussagen konnten in diesem Zusammenhang von den Experten mit großer Sicherheit getroffen werden. Beispielsweise herrschte Einigkeit darüber, dass bei steigendem Analyseaufwand für stabilisierbare Anforderungen deren Änderungsrate zunächst stark abnimmt, die Abnahme aber zunehmend abflacht. Wie sich dieser Verlauf genau quantitativ gestaltet ist jedoch unsicher. Um diese Ungenauigkeit abzufangen wurde eine Reihe von Tests durchgeführt, in denen für unsichere Eingabeparameter ein breites Spektrum denkbarer Belegungen getestet wurde. In den Tests wurde bewertet, ob die Ausgaben des Simulationsmodells jeweils sinnvoll sind und insbesondere, ob das Referenzverhalten erreicht werden kann. Die Tests ergaben, dass das Modell robust gegenüber den angemerkten Ungenauigkeiten in den Eingangsparametern ist, also auch bei deutlichen Abweichungen noch sinnvolle Ausgaben produziert. Zudem konnte geschlussfolgert werden, dass die von uns aus dem Simulationsmodell abgeleiteten Aussagen auch bei Abweichungen in den unsicheren Parametern haltbar sind. Dies trifft beispielsweise auf die Aussage zu, dass es bei einer Variierung des Analyseaufwands für stabilisierbare Anforderungen hinsichtlich der Projektlaufzeit ein Minimum gibt und sich dieses abhängig von der Änderungsrate verschiebt. Bestimmte quantitative Aussagen sind jedoch nicht möglich. So kann in unserem Beispiel der genaue Wert, für den das Minimum in der Realität zu erwarten ist, nicht anhand des Simulationsmodells bestimmt werden.

Mittels ausgiebiger Verhaltenstests wurde überprüft, ob das Simulationsmodell ein sinnvolles und realitätsnahes Verhalten erzeugt. Die Bewertung erfolgte hierbei unter Einbeziehung der Experten von PSIPENTA. Eine zentrale Rolle spielte dabei die korrekte Wiedergabe des Referenzverhaltens. Das Referenzverhalten „Menge realisierter Anforderungen“ (Abbildung 3-40) wird durch das Simulationsmodell sehr genau erreicht. Auch der Referenzwert „Menge der hinzukommenden oder geänderten Anforderungen pro Zeitspanne“ konnte zufriedenstellend wiedergegeben werden (Abbildung 3-41). Dabei nähert das Simulationsmodell Phasen niedriger RE-Aktivität mittels mehrerer kurzer

Phasen hoher RE-Aktivität an. Dementsprechend ergibt sich im Simulationsmodell in der Realisierungsphase der zweiten Iteration nicht ein konstant niedriges Niveau an durchgeführten Änderungen, wie der Referenzwert vorgibt, sondern eine Phase ohne Änderungen unterbrochen durch kurze Perioden mit vielen Änderungen. Auch mit dieser geringfügigen Einschränkung kann die Wiedergabe des Referenzverhaltens als ausreichend angenommen werden.

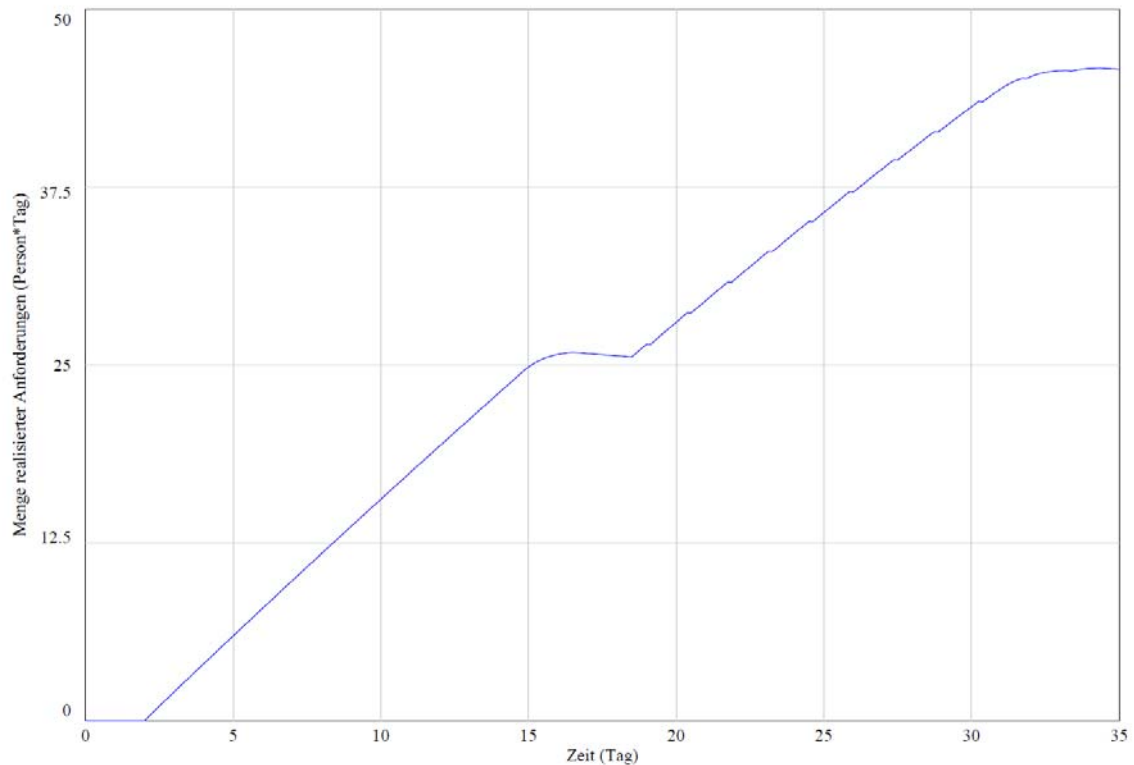


Abbildung 3-40: Reproduktion des Referenzverhaltens „Menge realisierter Anforderungen“ mittels des Simulationsmodells (die Menge der Anforderungen wird in Form des für sie tatsächlich notwendigen Erstrealisierungsaufwands gemessen)

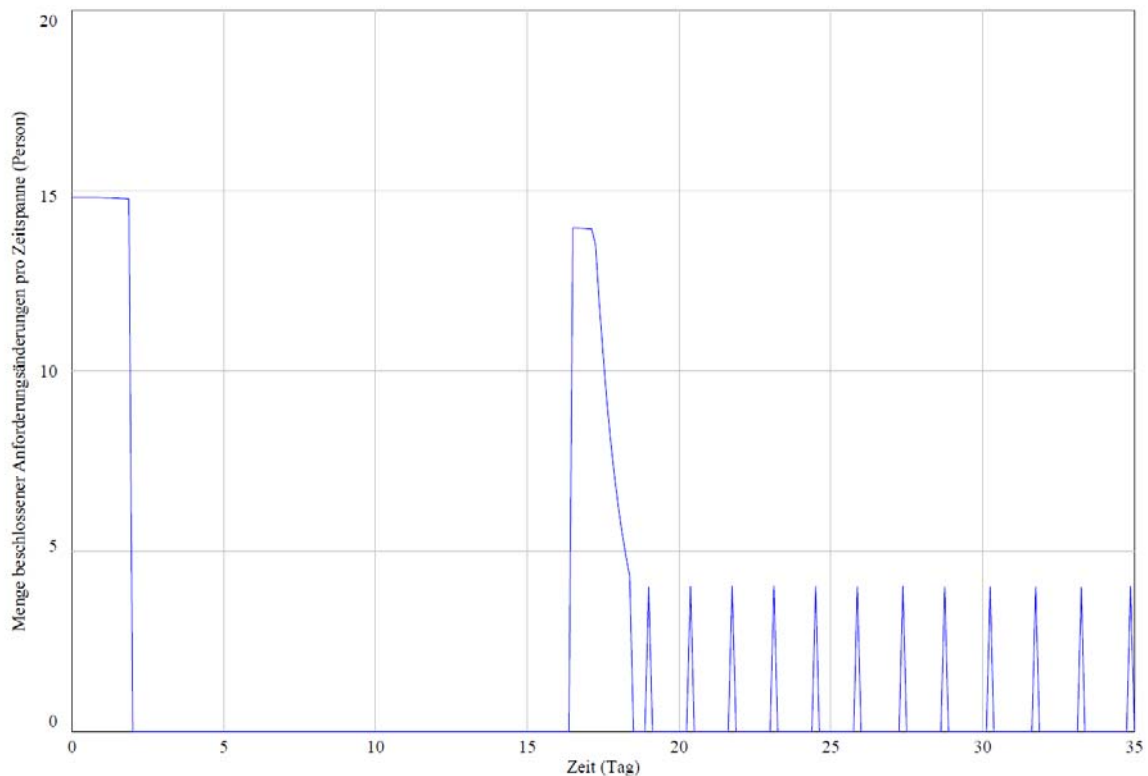


Abbildung 3-41: Reproduktion des Referenzverhaltens „Menge der hinzukommenden oder geänderten Anforderungen pro Zeitspanne“ mittels des Simulationsmodells (die Menge der Anforderungsänderungen wird in Form des für sie tatsächlich notwendigen Erstrealisierungsaufwands gemessen, die Menge pro Zeitspanne demzufolge in Personentag/Tag)

3.5.2 Durchführung von Experimenten

Anhand des angepassten Simulationsmodells wurden mehrere Experimente durchgeführt, durch die die in der Problembeschreibung aufgestellten Fragen beantwortet werden sollten. Mittels einer Sensitivitätsanalyse wurden verschiedene Modellparameter untersucht. Dabei wurden die Auswirkungen einer Variierung des RE-Aufwands für alle Anforderungstypen gleichzeitig, des RE-Aufwands für stabilisierbare Anforderungen allein, der Verteilung des RE-Aufwands auf die verschiedenen Anforderungstypen, der Änderungsrate der stabilisierbaren Anforderungen und der Stabilisierbarkeit der Anforderungen beobachtet.

Auswirkung des RE-Aufwands auf die Änderungsrate der stabilisierbaren Anforderungen

In einem ersten Experiment variierten wir den RE-Aufwand zum einen nur für stabilisierbare Anforderungen und zum anderen für alle Anforderungstypen. Wir untersuchten dabei die Auswirkungen auf die Änderungsrate der Anforderungen im Projektverlauf. In Abbildung 3-42 werden die Ergebnisse dargestellt.

Die Änderungsrate wird während der beiden Analysephasen jeweils zu Beginn einer Iteration stark gesenkt. Die Senkung im Modell spiegelt korrekt wider, dass durch die durchgeführten Analysen das Verständnis der Stakeholder für die Anforderungen enorm steigt und somit die Änderungsrate gedrückt wird. In der zweiten Realisierungsphase gibt es zahlreiche Einschübe, in denen eine Anforderungsanalyse stattfindet. Daher wird auch

in dieser die Änderungsrate abgesenkt, was durch das Simulationsmodell korrekt wiedergegeben wird. Wir stellen fest, dass ein erhöhter RE-Aufwand zu einer stärkeren Reduzierung der Änderungsrate führt. Dieser Effekt ist zu erwarten, da ein höherer Analyseaufwand zu einem höheren Verständnis der Anforderungen und damit wiederum zu einer niedrigeren Änderungsrate führt. Er deckt sich zudem mit den Erfahrungen der Experten von PSIPENTA. Die Absenkung der Änderungsrate ist jedoch zeitlich nach hinten versetzt, was auf die längere Dauer der Analyse zurückzuführen ist. Abhängig von der Variierung des RE-Aufwands ergibt sich im beobachteten Bereich eine signifikante Schwankung der Änderungsrate von bis zu 34,3%.

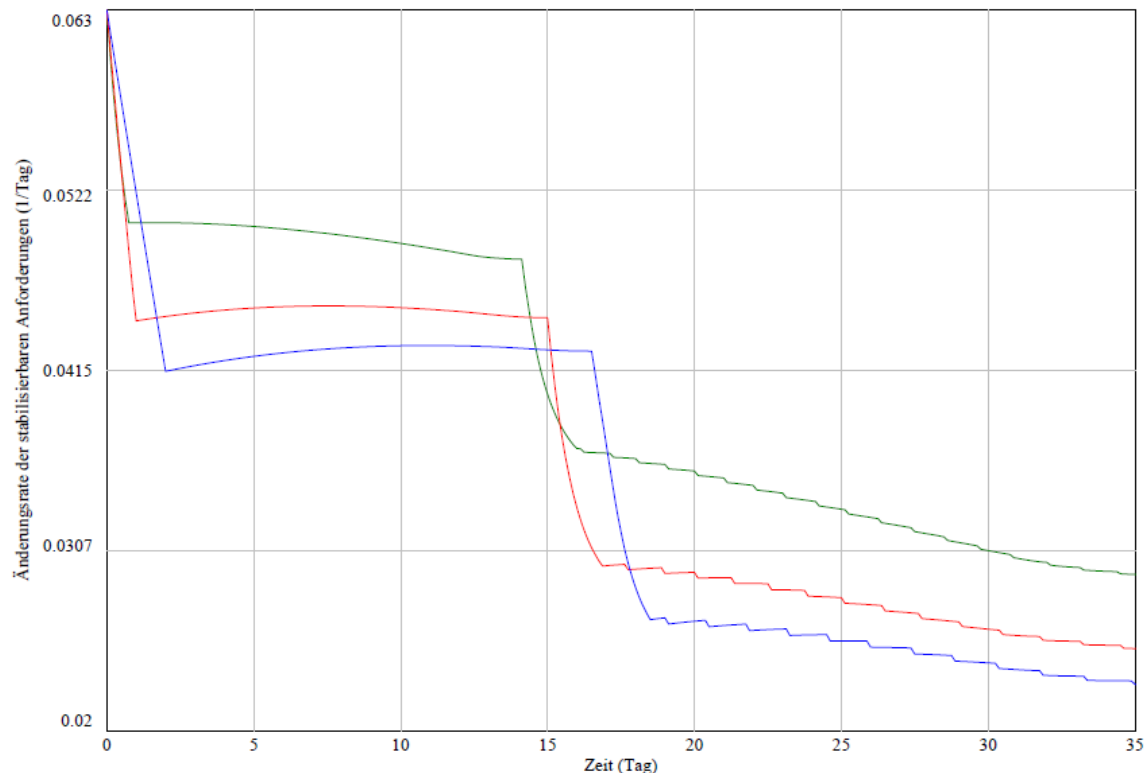


Abbildung 3-42: Simulation des zeitlichen Verlaufs der Änderungsrate der stabilisierbaren Anforderungen (die Änderungsrate wird im Prozentsatz der Anforderungen, die sich pro Tag ändert, gemessen)

Blau: Unveränderter RE-Aufwand

Rot: um 50% reduzierter RE-Aufwand für alle Anforderungstypen

Grün: um 75% reduzierter RE-Aufwand für alle Anforderungstypen

Auswirkung von Schwerpunktsetzungen im RE auf die Projektlaufzeit und die Projektkosten

In einem zweiten Experiment untersuchten wir, wie sich Projektlaufzeit und Projektkosten verhalten, wenn der für RE eingesetzte Aufwand unterschiedlich stark auf stabilisierbare Anforderungen konzentriert wird. Wir führten eine Reihe von Simulationsläufen durch, bei denen wir den Gesamtaufwand konstant hielten und den Anteil des Aufwands, der für stabilisierbare Anforderungen eingesetzt wird, variierten. Wir beobachteten zu welchem Zeitpunkt 90% der Anforderungen realisiert wurden und welche Projektkosten bis dahin entstanden.

Tabelle 3-6 zeigt die Ergebnisse dieses Experiments. Das Simulationsmodell zeigt für den Fall, dass der RE-Aufwand auf stabilisierbare Anforderungen konzentriert wird, eine Reduzierung der Projektlaufzeit und der Projektkosten an. Gerade in dieser Fallstudie ist dies unmittelbar nachvollziehbar, da die Experten von PSIPENTA davon ausgehen, dass eine Analyse von nicht stabilisierbaren und stabilen Anforderungen keine Auswirkungen auf die Änderungsrate der Anforderungen hat und dies im Simulationsmodell entsprechend abgebildet ist. Es ist jedoch zu beachten, dass für Anforderungen jeden Typs immer ein bestimmtes Mindestmaß an RE-Aufwand eingesetzt werden muss, damit einerseits eine sinnvolle Realisierung möglich ist und andererseits die idealisierte Aufteilung in die verschiedenen Anforderungstypen näherungsweise mit der Wirklichkeit übereinstimmt. Die Fokussierung des RE-Aufwands auf stabilisierbare Anforderungen wird von PSIPENTA als sinnvoll erachtet und bereits vielfach im Unternehmen eingesetzt. Im beobachteten Bereich schwanken die Gesamtprojektkosten um bis zu 4,6%. Da bei PSIPENTA die Änderungskosten normalerweise 8% der Gesamtkosten betragen entspricht dies rund einer Schwankung der Änderungskosten um 150%.

Zum Realisierungsaufwand relativer Aufwand für die Analyse der einzelnen Anforderungstypen [dimensionslos]			Tag an dem 90% der Anforderungen realisiert wurden	Kosten für die Realisierung von 90% der Anforderungen [Personentage]
Aufwand stabilisierbare Anforderungen	Aufwand nicht stabilisierbare Anforderungen	Aufwand stabile Anforderungen		
0,025	0,1875	0,1875	34,5	61,41
0,050	0,175	0,175	33,000	59,84
0,100	0,150	0,150	32,000	58,83
4/30	4/30	4/30	31,875	58,73
0,200	0,100	0,100	31,625	58,71
0,250	0,075	0,075	31,500	58,71

Tabelle 3-6: Vergleich der Projektlaufzeit bei unterschiedlicher Konzentration des RE-Aufwands auf verschiedene Anforderungstypen und gleichbleibenden Gesamtaufwand (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

Auswirkung der Variierung des RE-Aufwands auf die Projektlaufzeit und die Projektkosten

Weiterhin untersuchten wir, wie sich eine Variierung des RE-Aufwands für stabilisierbare Anforderungen bei gleichzeitig konstantem RE-Aufwand für die anderen Anforderungstypen auf Projektlaufzeit und Projektkosten auswirkt.

Abbildung 3-43 und Abbildung 3-44 zeigen den Verlauf von Projektlaufzeit und Projektkosten in der Simulation abhängig vom eingesetzten RE-Aufwand für stabilisierbare Anforderungen. Wie auch in der Fallstudie bei der FAUSER AG zeichnen sich abhängig vom RE-Aufwand für stabilisierbare Anforderungen in Projektlaufzeit und Projektkosten lokale Minima ab. Aufgrund der Granularität des Simulationsmodells lassen sich diese jedoch nur grob eingrenzen.

Das beobachtete Minimum stellt das optimale Verhältnis zwischen Analyseaufwand und Änderungskosten dar. Je mehr Aufwand im RE betrieben wird, desto höher ist das Verständnis für die Anforderungen und umso geringer die Änderungsrate. Man spart somit

also Änderungskosten ein. Selbstverständlich können durch einen beliebig hohen zusätzlichen RE-Aufwand nicht beliebig viele Änderungskosten eingespart werden. Zudem ist für gewöhnlich zu erwarten, dass bei einem zu geringen RE-Aufwand das System kaum verstanden und die Änderungsrate sowie die Änderungskosten extrem hoch ausfallen. Es lassen sich theoretisch Szenarien konstruieren, in denen dies nicht der Fall ist, wie zum Beispiel ein Projekt in dem schon vor Beginn die Anforderungen zu hundert Prozent klar sind und es zu keinen Änderungen kommen wird. In diesem Fall ist der Aufwand für die Anforderungsanalyse vergeudet und das Minimum in Projektlaufzeit und –kosten stellt sich für den Fall ein, dass keinerlei Analyse durchgeführt wird. Dies ist jedoch nur ein theoretisches Beispiel und für diese Arbeit nicht relevant, da nur Projekte mit besonders instabilen Anforderungen betrachtet werden.

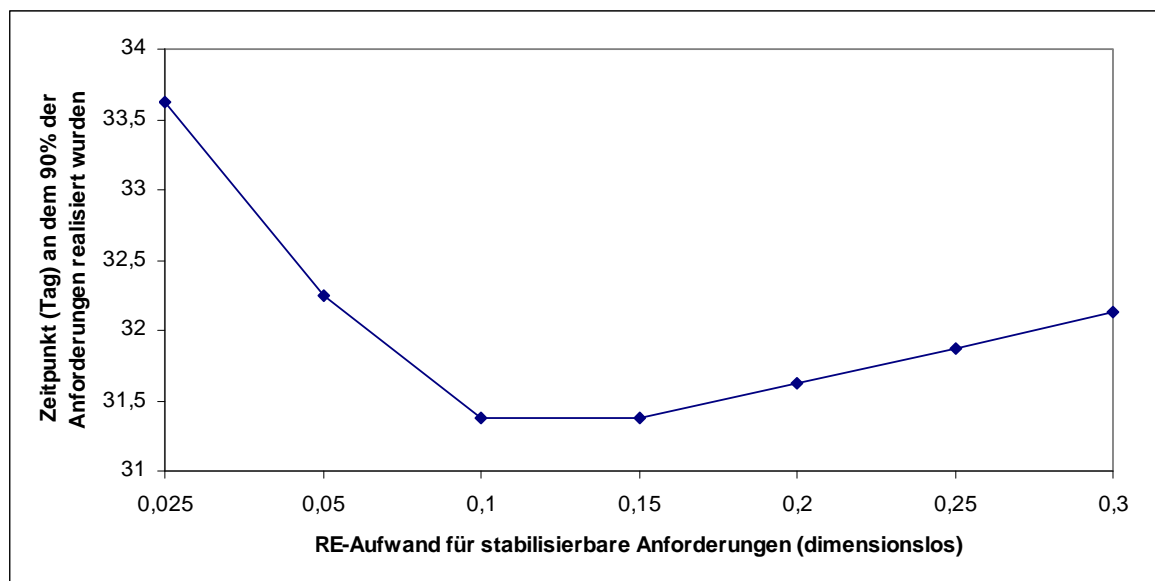


Abbildung 3-43: Simulation der Projektlaufzeit bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

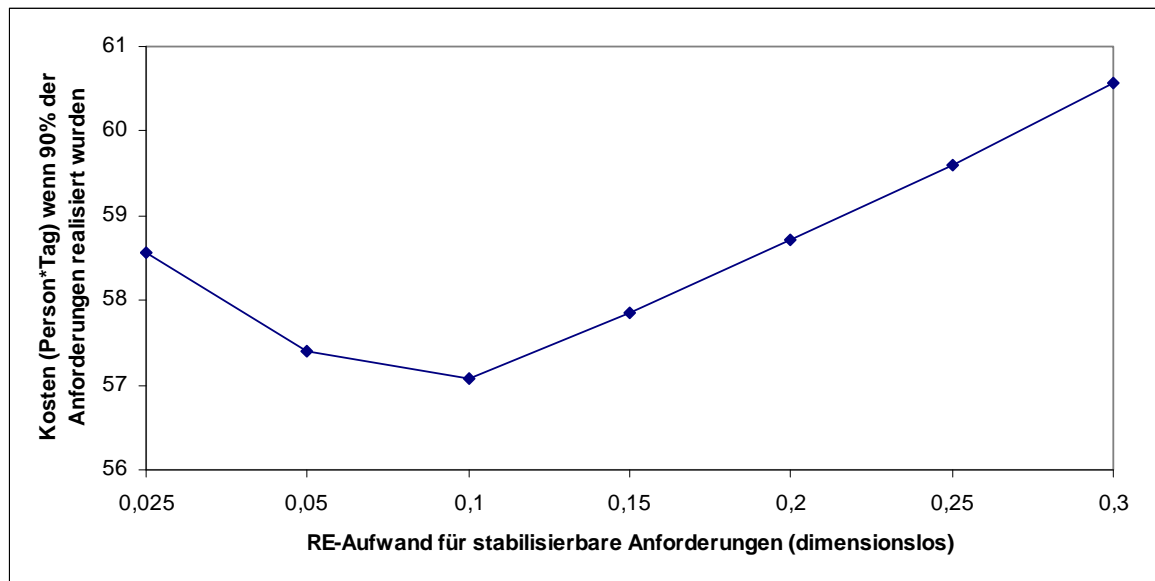


Abbildung 3-44: Simulation der Projektkosten bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

In Abbildung 3-45 und Tabelle 3-7 sind die durch Simulation bestimmten Kosten für Anforderungsänderungen abhängig vom jeweils eingesetzten RE-Aufwand für stabilisierbare Anforderungen dargestellt. Es ist klar erkennbar, dass die Änderungskosten bei höheren RE-Aufwänden geringer ausfallen. Im Referenzfall ergeben sich Änderungskosten von rund 4,7 Personentagen, was in etwa 8,0% des Gesamtentwicklungsaufwands entspricht. Bei einer Erhöhung des RE-Aufwands für stabilisierbare Anforderungen um 25% betragen die Änderungskosten nur noch rund 5,6% des Gesamtaufwands. Eine Reduzierung des RE-Aufwands für stabilisierbare Anforderungen um 50% führt zu einer Erhöhung der Änderungskosten auf ca. 13,3% des Gesamtaufwands.

Für die Gesamtprojektkosten ergibt sich im beobachteten Bereich abhängig von der Variierung des Aufwands für stabilisierbare Anforderungen eine Schwankung von bis zu 6,1%. Dabei schwanken die Änderungskosten um den Faktor 6. Wie auch schon in der Fallstudie bei der FAUSER AG sind bei PSIPENTA die Änderungskosten deutlich geringer als von Boehm und Basili (2001) angenommen. Würden wir wie Boehm und Basili annehmen, dass bei der Entwicklung von einfachen nicht kritischen Systemen Änderungen 40-50% der Gesamtkosten ausmachen, so würde sich durch die Variierung des Aufwands für stabilisierbare Anforderungen eine deutlich höhere Schwankung der Gesamtprojektkosten ergeben. Es muss jedoch berücksichtigt werden, dass Boehm und Basili sich nicht nur auf kleinere Projekte beziehen, sondern auch auf größere, bei denen Änderungen an bereits realisierten Anforderungen deutlich teurer ausfallen.

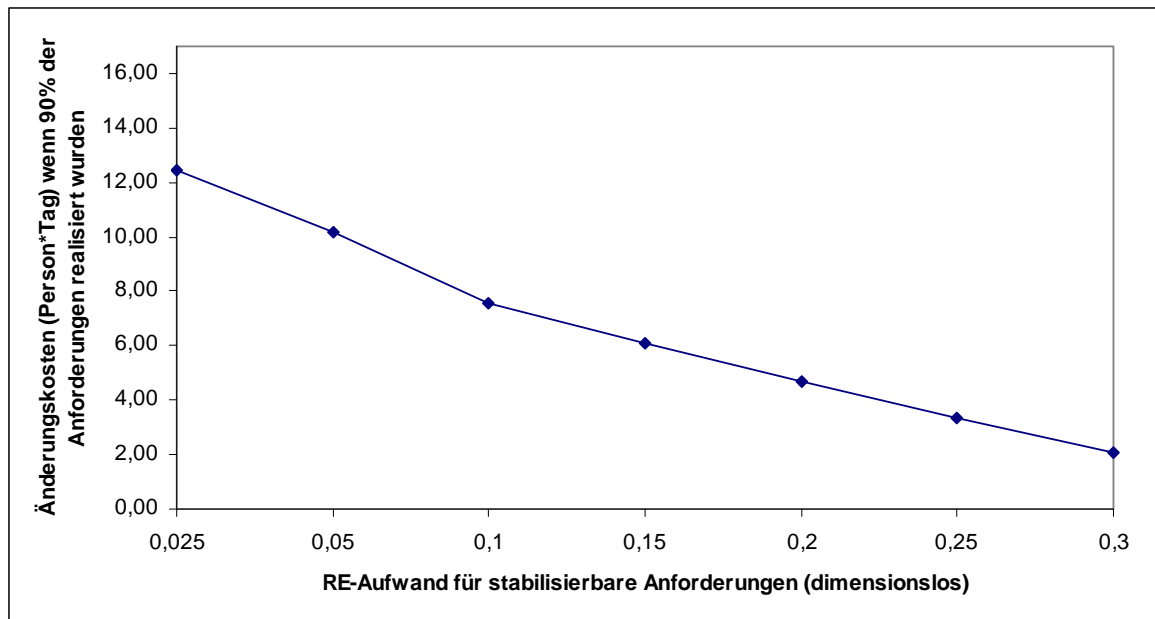


Abbildung 3-45: Simulation der Änderungskosten bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

Zum Realisierungsaufwand relativer Aufwand für die Analyse der stabilisierbaren Anforderungen	Anteil der Änderungskosten am Gesamtaufwand
0,025	21,2%
0,05	17,7%
0,1	13,3%
0,15	10,5%
0,2	8,0%
0,25	5,6%
0,3	3,4%

Tabelle 3-7: Simulation des Anteils der Änderungskosten am Gesamtaufwand bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

Wir führten ein weiteres Simulationsexperiment durch, in dem wir neben dem Aufwand für stabilisierbare Anforderungen auch den von stabilen und nicht stabilisierbaren Anforderungen variierten. Die bei PSIPENTA übliche Fokussierung auf stabilisierbare Anforderungen wurde dabei aufgehoben. Dies bedeutet, dass wir für jeden Anforderungstyp in der Analyse den gleichen Aufwand annahmen. Die Ergebnisse sind in Abbildung 3-46 und Abbildung 3-47 dargestellt. Es zeigt sich erneut, dass sich bei bestimmten RE-Aufwänden Minima in Projektlaufzeit und –kosten einstellen. Die Gesamtprojektkosten schwanken abhängig vom RE-Aufwand wesentlich stärker. Im beobachteten Bereich ergibt sich eine Schwankungsbreite von rund 20,7%. Die Änderungskosten schwanken dabei um 130%. In diesem Experiment zeigt sich, dass es bei der Variierung des RE-Aufwands ein signifikantes Optimierungspotenzial in den Projektkosten gibt.

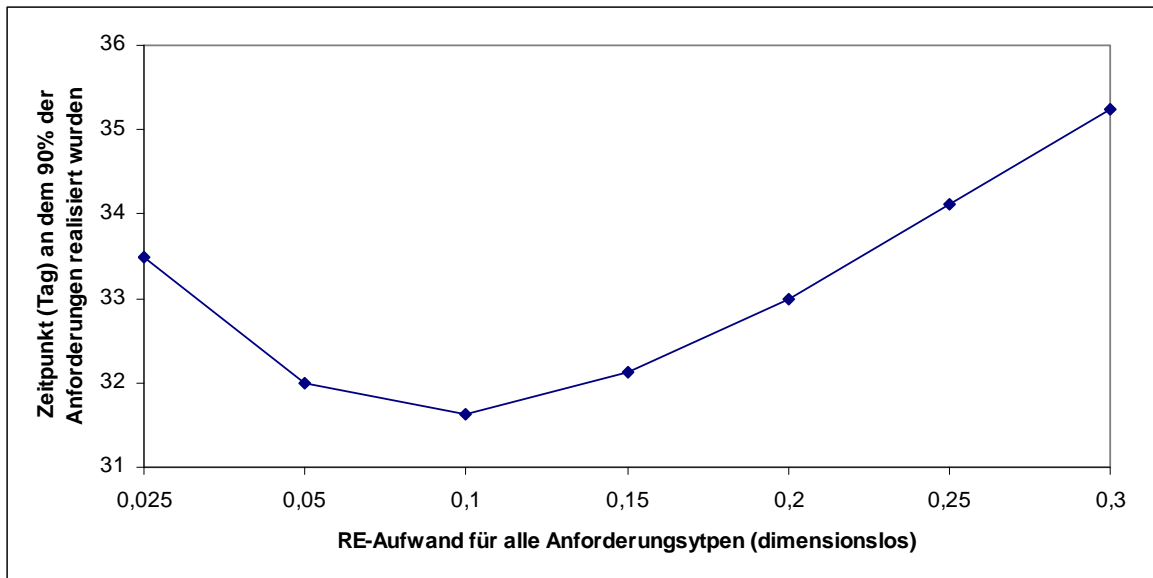


Abbildung 3-46: Simulation der Projektlaufzeit bei unterschiedlich hohem RE-Aufwand für alle Anforderungstypen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

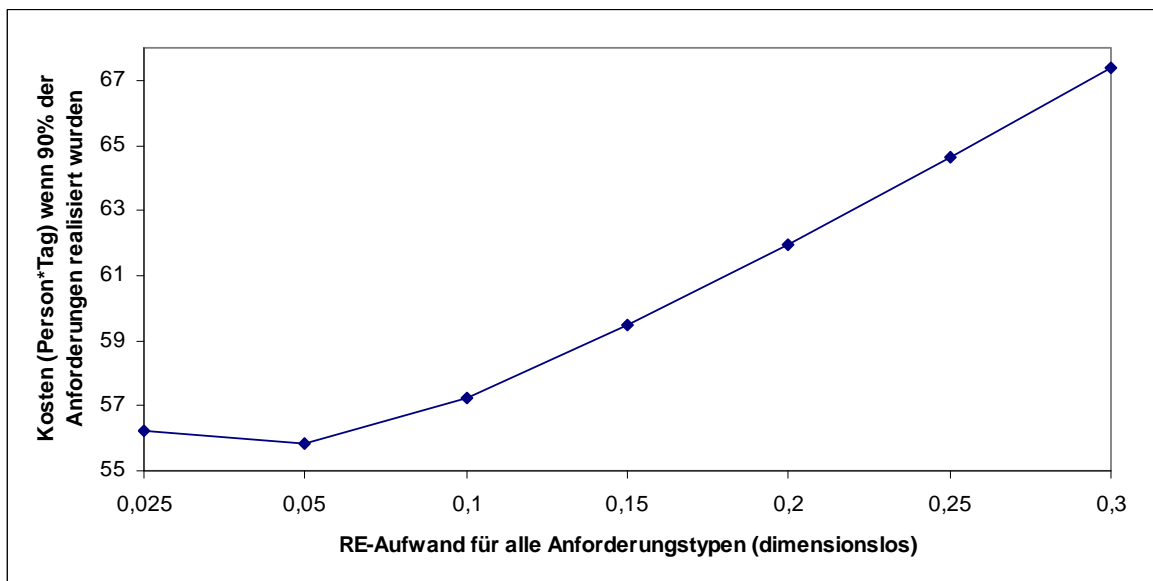


Abbildung 3-47: Simulation der Projektkosten bei unterschiedlich hohem RE-Aufwand für alle Anforderungstypen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

Auswirkung der Stabilisierbarkeit der Anforderungen auf den in Bezug auf Projektlaufzeit und Projektkosten optimalen RE-Aufwand für stabilisierbare Anforderungen

In einem weiteren Experiment untersuchten wir, wie sich eine geringere Stabilisierbarkeit der Anforderungen auf den hinsichtlich Projektlaufzeit und –kosten optimalen RE-Aufwand auswirkt. Hierzu schwächten wir die Absenkung der Änderungsrate der

stabilisierbaren Anforderungen durch ein steigendes Verständnis, wie in Abbildung 3-48 gezeigt, ab.

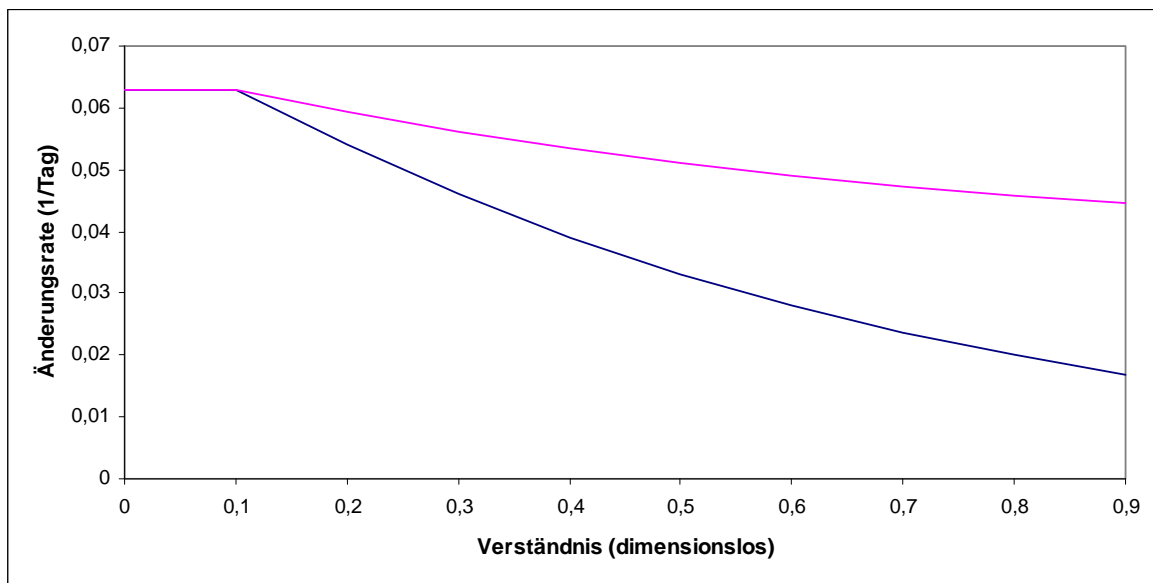


Abbildung 3-48: Änderungsrate der stabilisierbaren Anforderungen abhängig von deren Verständnis in der aktuellen Situation des betrachteten Projekts (schwarz) und bei geringerer Stabilisierbarkeit (rosa) (die Änderungsrate wird im Prozentsatz der Anforderungen, der sich pro Tag ändert, gemessen; das Verständnis ist eine Hilfsvariable, die bei der Umrechnung unter anderem von Analyseaufwand in resultierende Änderungsrate herangezogen wird, und ist dimensionslos)

Wir untersuchten zunächst, wie sich die geringere Stabilisierbarkeit auf die Projektlaufzeit auswirkt. Der Simulationslauf ergab, dass sich der Zeitpunkt, an dem 85% der Anforderungen realisiert wurden, um 5,375 Tage nach hinten verschiebt. Das Modell gibt also die erwartete Verzögerung aufgrund der höheren Änderungsrate korrekt wieder. Darüber hinaus variierten wir wiederum den RE-Aufwand für stabilisierbare Anforderungen, um festzustellen in welchem Bereich die lokalen Minima in Projektlaufzeit und –kosten bei der verringerten Stabilisierbarkeit der Anforderungen liegen.

Die Ergebnisse sind in Abbildung 3-49 und Abbildung 3-50 dargestellt. Wie auch bei der Fallstudie in der FAUSER AG erwarteten wir bei einer niedrigeren Stabilisierbarkeit der Anforderungen eine Verschiebung der Minima in Projektlaufzeit und –kosten hin zu niedrigeren RE-Aufwänden, da die Effizienz der Anforderungsanalyse hinsichtlich des Verständnissgewinns abnimmt (siehe Kapitel 3.4.2).

Die Simulation ergab, dass sich das Minimum in den Projektkosten bei geringerer Stabilisierbarkeit hin zu niedrigeren RE-Aufwänden verschiebt. Aus den Simulationsergebnissen geht nicht eindeutig hervor, dass sich auch das Minimum in der Projektlaufzeit entsprechend verhält. Das Intervall, in dem sich laut Simulationsmodell der hinsichtlich der Projektlaufzeit optimale RE-Aufwand für stabilisierbare Anforderungen befindet, verschiebt sich von]0,05; 0,2[im Ausgangsfall zu]0,05; 0,15[im Falle geringerer Stabilisierbarkeit. Das Ergebnis widerspricht jedoch nicht unseren Überlegungen und den Ergebnissen der Fallstudie bei der FAUSER AG. Die Verschiebung des Minimums in den Projektkosten mitsamt der starken Korrelation von Projektlaufzeit und –kosten legt nahe, dass auch in dieser Fallstudie sich das Minimum in der

Projektlaufzeit entsprechend verschiebt, selbst wenn das Modell dies aufgrund seiner Granularität nicht eindeutig zeigt. Auch die Einengung des Intervalls, in dem das Minimum liegt, auf den niedrigeren Wertebereich stützt diese These.

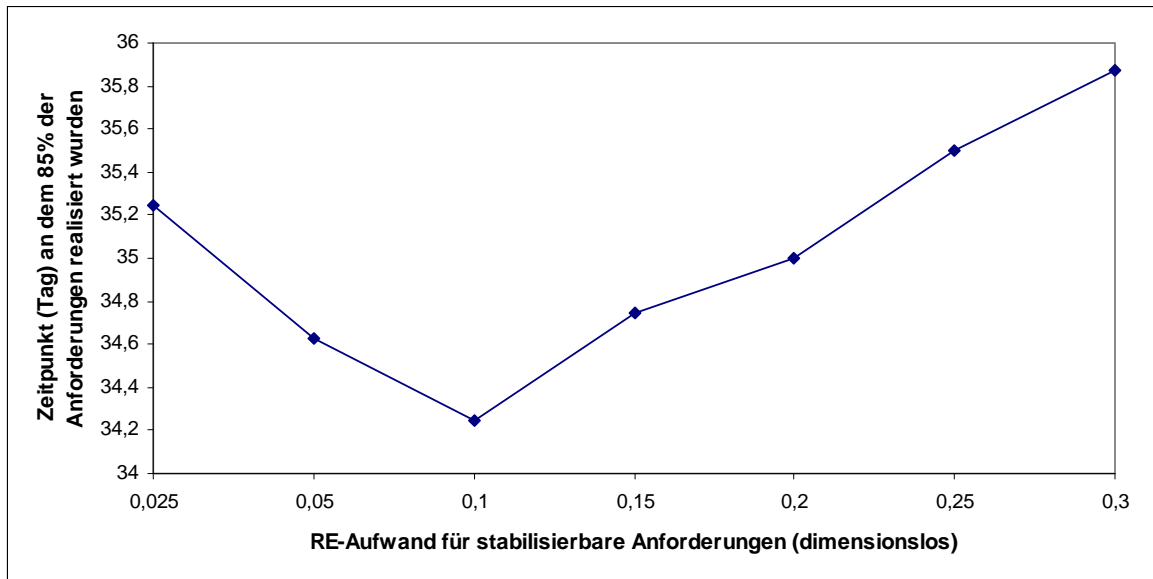


Abbildung 3-49: Simulation der Projektlaufzeit bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei geringerer Stabilisierbarkeit (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

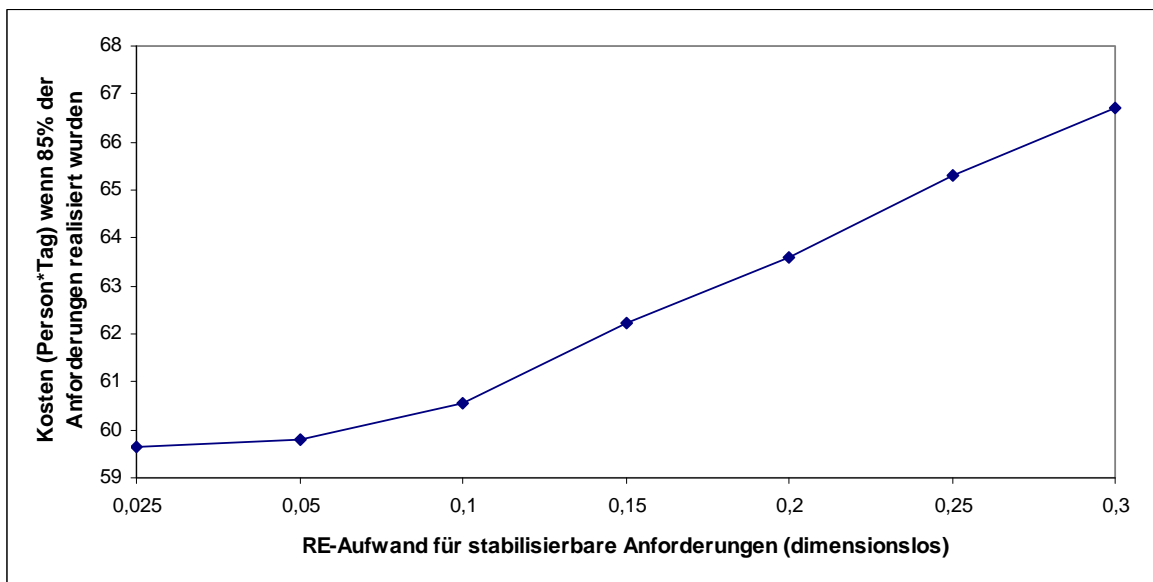


Abbildung 3-50: Simulation der Projektkosten bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei geringerer Stabilisierbarkeit (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

Auswirkung der Änderungsrate der stabilisierbaren Anforderungen auf den in Bezug auf Projektlaufzeit und Projektkosten optimalen RE-Aufwand für stabilisierbare Anforderungen

Wir führten weitere Experimente durch, in denen wir die Änderungsrate der stabilisierbaren Anforderungen um 100% (Abbildung 3-51 und Abbildung 3-52) und 200% (Abbildung 3-53 und Abbildung 3-54) erhöhten beziehungsweise um 50% senkten (Abbildung 3-55 und Abbildung 3-56). Für jede Änderungsrate variierten wir den RE-Aufwand für stabilisierbare Anforderungen, um die lokalen Minima in Projektlaufzeit und –kosten zu bestimmen.

Es ist zu beobachten, dass bei steigender Änderungsrate sich die beobachteten lokalen Minima von Projektlaufzeit und –kosten hin zu höheren RE-Aufwänden für stabilisierbare Anforderungen verschieben.

Es sei darauf hingewiesen, dass, wie bereits in Kapitel 3.4.2 erläutert, unsere Erhöhung der Änderungsrate gleichzeitig eine Erhöhung der Stabilisierbarkeit der Anforderungen bedeutet. Den Experten von PSIPENTA zufolge bringt eine Erhöhung der Änderungsrate normalerweise auch eine Erhöhung der Stabilisierbarkeit der Anforderungen mit sich. Es macht also in Bezug auf die Minimierung von Projektlaufzeit und –kosten in der Regel Sinn, mehr Aufwand in die Analyse von stabilisierbaren Anforderungen zu investieren, je instabiler die Anforderungen sind. Wie bereits in Kapitel 3.4.2 ausgeführt lässt sich dieser Effekt dadurch erklären, dass im Falle einer höheren Änderungsrate der gleiche Zusatzaufwand im RE eine höhere Einsparung in den Änderungskosten ergibt.

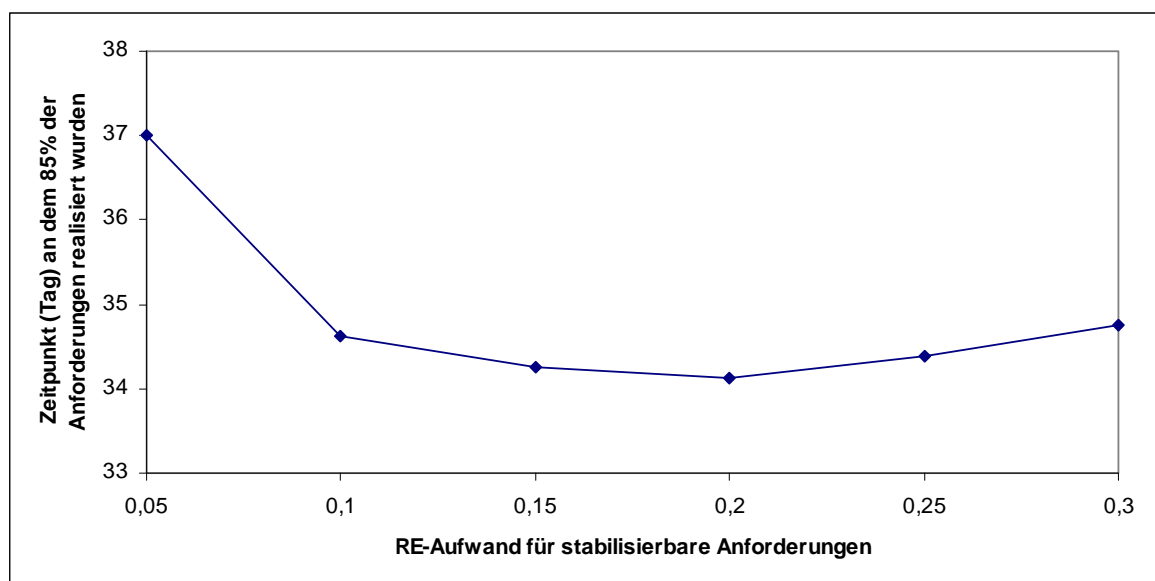


Abbildung 3-51: Simulation der Projektlaufzeit bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei 100% höherer Änderungsrate der stabilisierbaren Anforderungen (der RE-Aufwand wird in Prozenttagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

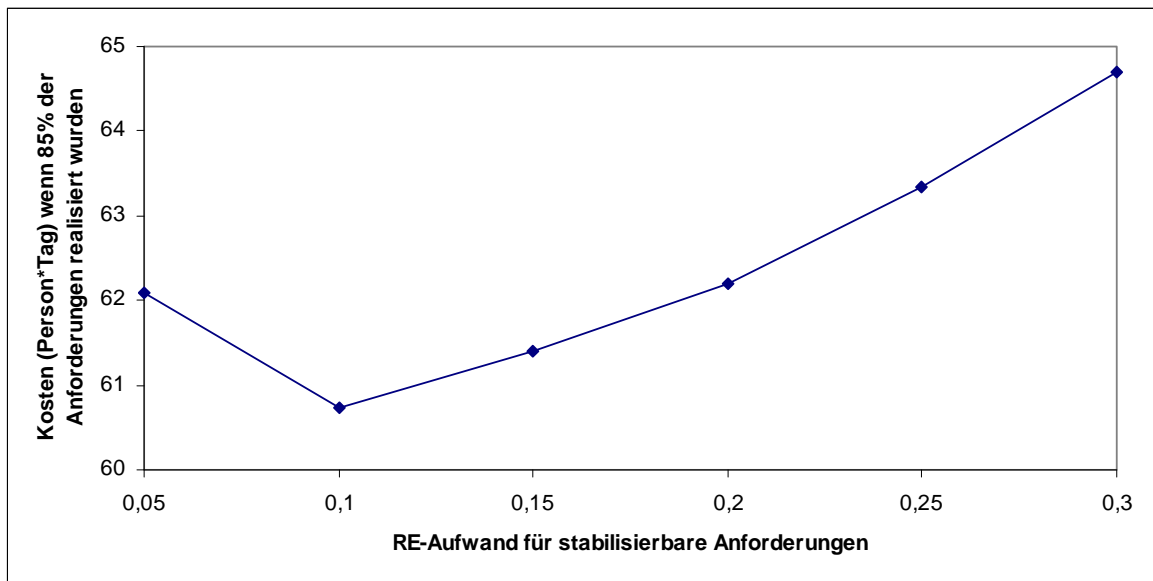


Abbildung 3-52: Simulation der Projektkosten bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei 100% höherer Änderungsrate der stabilisierbaren Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

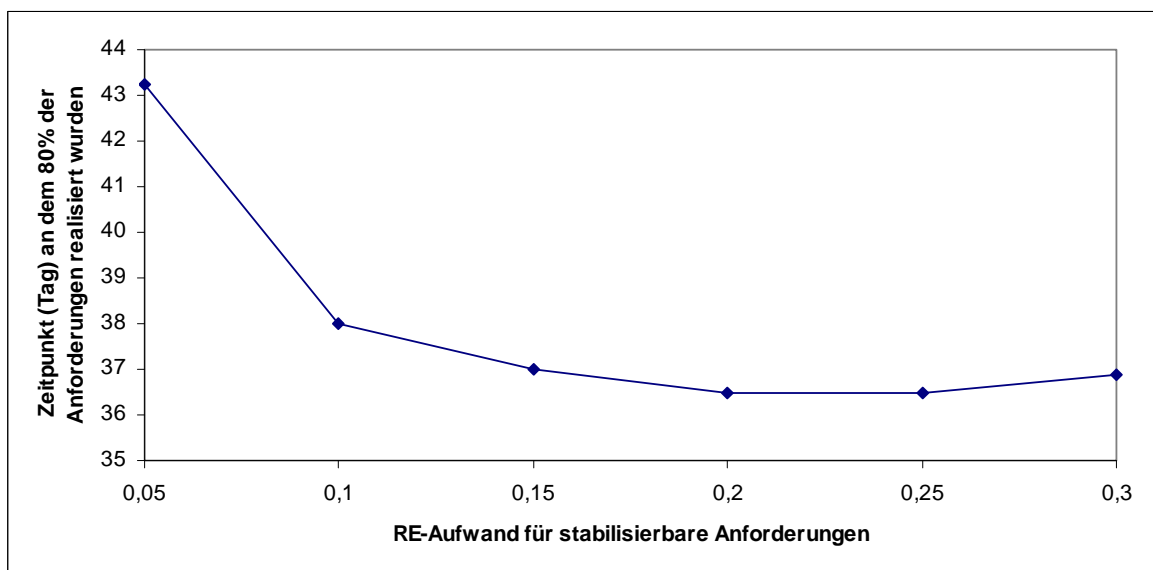


Abbildung 3-53: Simulation der Projektlaufzeit bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei 200% höherer Änderungsrate der stabilisierbaren Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

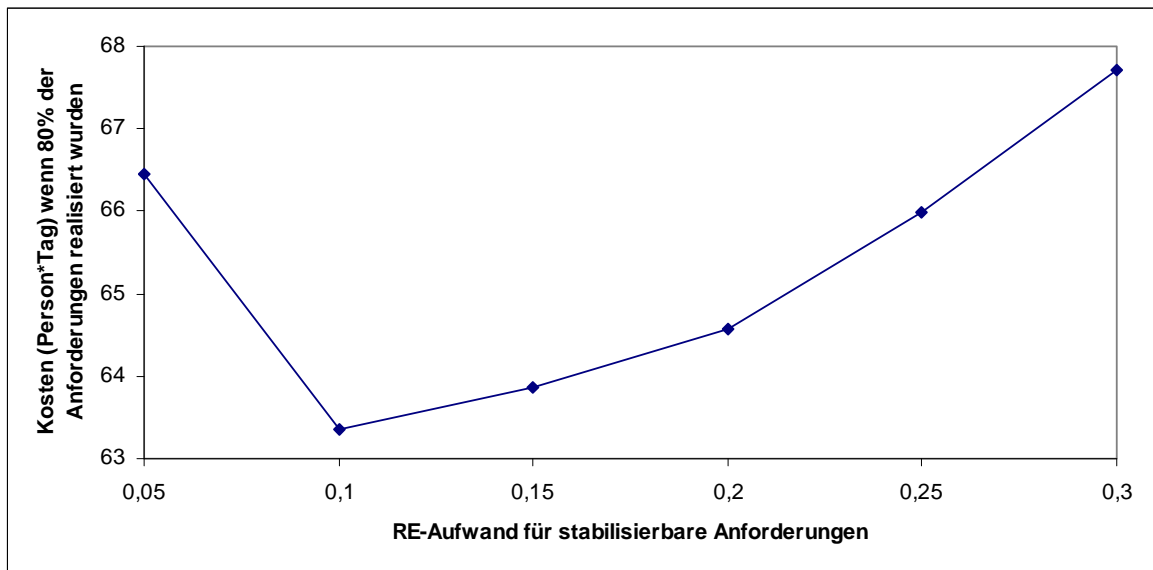


Abbildung 3-54: Simulation der Projektkosten bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei 200% höherer Änderungsrate der stabilisierbaren Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

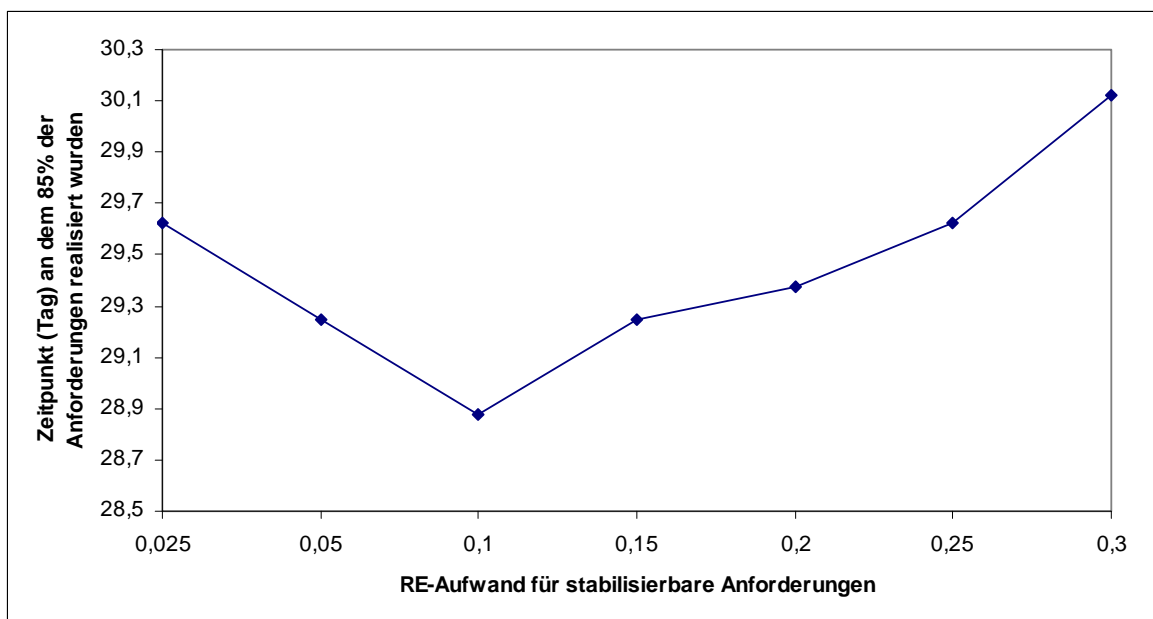


Abbildung 3-55: Simulation der Projektlaufzeit bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei 50% niedrigerer Änderungsrate der stabilisierbaren Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

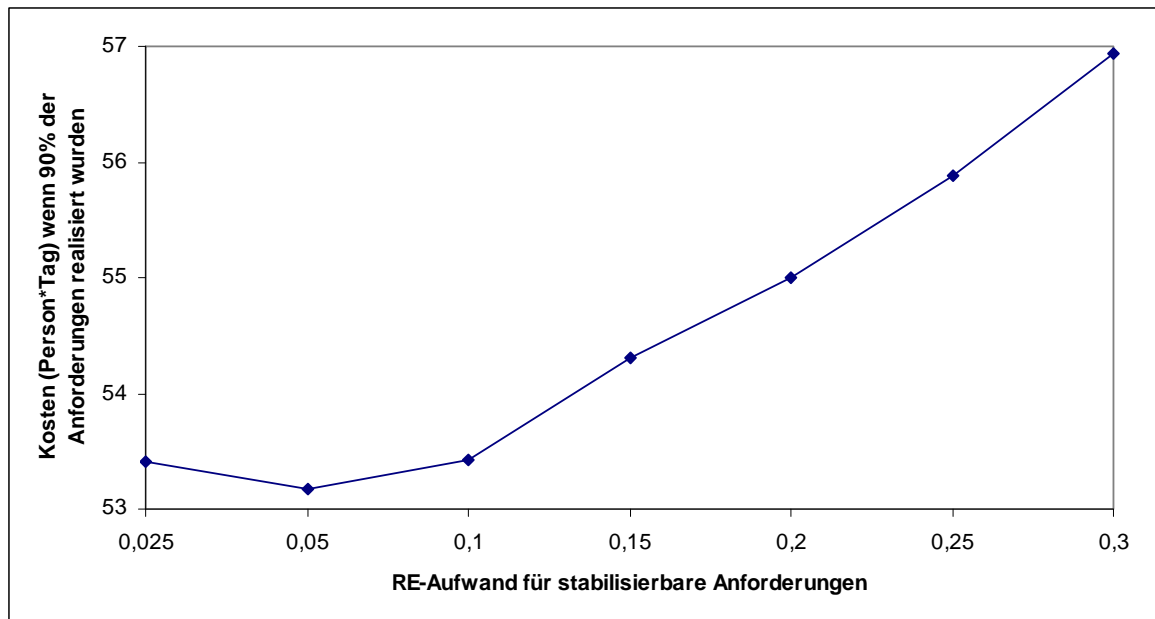


Abbildung 3-56: Simulation der Projektkosten bei unterschiedlich hohem RE-Aufwand für stabilisierbare Anforderungen bei 50% niedrigerer Änderungsrate der stabilisierbaren Anforderungen (der RE-Aufwand wird in Personentagen Analyseaufwand pro Personentag Realisierungsaufwand einer Anforderung gemessen, ist also dimensionslos)

3.5.3 Zusammenfassung der Ergebnisse

Die Ergebnisse dieser Fallstudie decken sich sehr gut mit denen aus der Fallstudie bei der FAUSER AG.

Wir konnten feststellen, dass sich eine Variierung des RE-Aufwands auf die Änderungsrate der Anforderungen auswirkt. Je mehr Aufwand für die Anforderungsanalyse aufgebracht wird, umso stärker wird die Änderungsrate reduziert. Die entscheidende Rolle spielt dabei der Aufwand für die Analyse der stabilisierbaren Anforderungen. Die Änderungsraten der anderen Anforderungstypen werden ohnehin als konstant angenommen. Eine Analyse dieser Anforderungstypen ist zwar zu einem bestimmten Mindestmaß nötig, doch führt sie laut den Experten von PSIPENTA in ihren typischen Projekten zu keinem indirekten Verständniskennwert für die stabilisierbaren Anforderungen. Sie ist daher für die Absenkung der Änderungsrate nicht mitverantwortlich. Eine Konzentration des RE-Aufwands auf stabilisierbare Anforderungen macht daher hinsichtlich einer Einsparung von Änderungskosten Sinn. Dies lässt sich verallgemeinern, da die Analyse von stabilisierbaren Anforderungen stets den größten Effekt hinsichtlich der Senkung der Änderungsrate hat.

Es konnte auch in dieser Fallstudie beobachtet werden, dass sich hinsichtlich Projektlaufzeit und –kosten für bestimmte RE-Aufwände Minima ausbilden. Dies bedeutet, dass für die Reduktion von Projektlaufzeit und –kosten eine optimale Balance zwischen eingesetztem RE-Aufwand und durch Anforderungsänderungen bewirktem Änderungsaufwand existiert. Hierbei ist jedoch zu beachten, dass das Simulationsmodell zwar geeignet ist, qualitativ zu bestimmen, dass es ein solches Optimum gibt, es jedoch nicht gesichert ist, dass auch eine quantitative Aussage gemacht werden kann. Bei der Datenerhebung konnten die Experten von PSIPENTA bezüglich bestimmter Größen quantitativ nur grobe Abschätzungen vornehmen. Dies betrifft vor allem die Beziehung zwischen RE-Aufwand und Verständniskennwert sowie zwischen gewonnenem Verständnis

für die Anforderungen und Änderungsrate. Die Aussagen zu den Grundmechanismen sowie der qualitativen Ausgestaltung der Beziehung zwischen RE-Aufwand, Verständnis und Änderungsrate können jedoch als gesichert angenommen werden. Dementsprechend kann das System Dynamics Modell herangezogen werden, um qualitative Aussagen zu machen. Das Minimum für die Projektlaufzeit stellte sich bei etlichen Experimenten für andere RE-Aufwände ein, als das Minimum in den Projektkosten. Es muss also bei der Entscheidung über den Umfang der Anforderungsanalyse stets zwischen Projektlaufzeit und -kosten abgewogen werden. Wie bereits in Kapitel 3.4.3 diskutiert, gibt es noch weitere Parameter, die vom RE-Aufwand abhängen und in der Planung des Umfangs der RE-Aktivitäten berücksichtigt werden sollten, wie zum Beispiel die Qualität des entwickelten Systems. Zudem ist zu beachten, dass, wie bereits in Kapitel 3.5.1 erläutert, ein bestimmtes Mindestmaß an RE-Aufwand nicht unterschritten werden darf, da ansonsten keine sinnvolle Realisierung möglich ist und eine realistische Aufteilung der Anforderungen in die verschiedenen Anforderungstypen nicht aufrecht erhalten werden kann.

Den Simulationsergebnissen zufolge verschiebt sich bei einer geringeren Stabilisierbarkeit der Anforderungen das Minimum in den Projektkosten, wie schon in der ersten Fallstudie beobachtet, hin zu niedrigeren RE-Aufwänden. Bezüglich der Verschiebung des Minimums in der Projektlaufzeit lassen die Simulationsergebnisse keinen klaren Schluss zu. Die Ergebnisse widersprechen aber auch nicht der in der Fallstudie bei der FAUSER AG beobachteten Verschiebung des Minimums hin zu niedrigeren RE-Aufwänden bei geringerer Stabilisierbarkeit. Es ergibt sich somit insgesamt das Bild, dass eine geringere Stabilisierbarkeit der Anforderungen für die Minimierung sowohl von Projektlaufzeit als auch –kosten Einsparungen im RE-Aufwand erforderlich macht.

Weiterhin konnten wir eine Verschiebung der Minima von Projektkosten und –laufzeit abhängig von der Änderungsrate der Anforderungen beobachten. Je instabiler die Anforderungen waren, desto weiter verschoben sich die Minima hin zu höheren RE-Aufwänden. Zu beachten ist hierbei, dass wir auch in dieser Fallstudie von einer Erhöhung der Änderungsrate analog zu Abbildung 3-33 ausgehen. Wie bereits ausgeführt geht bei einer solchen Betrachtungsweise mit einer Erhöhung der Änderungsrate stets eine Erhöhung der Stabilisierbarkeit der Anforderungen einher. Der hier beobachtete Effekt kann damit auf die Verschiebung der Minima aufgrund einer geänderten Stabilisierbarkeit der Anforderungen zurückgeführt werden.

Grundvoraussetzung für die Anwendung der in diesem Kapitel abgeleiteten RE-Vorgehensweisen zur Minimierung von Projektlaufzeit und –kosten ist, dass die Anforderungen zum Großteil korrekt den verschiedenen Anforderungstypen zugeordnet werden können. Bei PSIPENTA wird bereits erfolgreich zwischen stabilisierbaren und nicht stabilisierbaren Anforderungen unterschieden. Darüber hinaus ist es nötig Änderungsrate und Stabilisierbarkeit der stabilisierbaren Anforderungen abschätzen zu können. Eine exakte Bestimmung ist zwar laut den Experten von PSIPENTA schwierig, aber in der Regel kann gut abgeschätzt werden, ob und in welche Richtung Änderungsrate und Stabilisierbarkeit vom Normalfall abweichen.

Die Experten der PSIPENTA bewerteten die qualitativen Ergebnisse der Simulation als richtig. Wie auch schon bei der Fallstudie bei der FAUSER AG gaben die Beteiligten an, dass durch die Durchführung der Simulation das Verständnis für den eigenen Softwareentwicklungsprozess und die grundlegenden Zusammenhänge zwischen Analyseaufwand, Verständnis der Anforderungen, Änderungskosten und Gesamtprojektkosten gesteigert wurde. Weiterhin kamen auch sie zu der Erkenntnis, dass

abhängig von der jeweils angetroffenen Projektsituation es sinnvoll ist unterschiedlich viel Aufwand für RE zu betreiben.

1. Für Anforderungen, die als stabilisierbar eingestuft werden, sollten umfangreichere Analysen in der RE-Phase angestellt werden als für andere Anforderungen.
2. Wird die *Stabilisierbarkeit* der Anforderungen höher als in einem „durchschnittlichen“ Projekt eingeschätzt, so sind umfangreichere Analysen als üblich in der RE-Phase durchzuführen. Ist die Stabilisierbarkeit niedriger, sollte entsprechend weniger Analyseaufwand betrieben werden.
3. Ist der *Anteil der stabilisierbaren Anforderungen* in einem Projekt höher als in einem „durchschnittlichen“ Projekt, so stellt dies eine insgesamt höhere Stabilisierbarkeit der Anforderungen dar und es sollte entsprechend mehr Analyseaufwand in der RE-Phase aufgewendet werden. Analog wird weniger Analyseaufwand betrieben, wenn der Anteil der stabilisierbaren Anforderungen vergleichsweise niedrig ist.
4. Wird die *Änderungsrate der Anforderungen* in einem Projekt höher als in einem „durchschnittlichen“ Projekt eingeschätzt, so weist dies auf eine insgesamt höhere Stabilisierbarkeit der Anforderungen hin. In diesem Fall sollten umfangreichere Analysen als üblich in der RE-Phase durchgeführt werden. Wird die Änderungsrate hingegen niedriger als gewöhnlich eingeschätzt so sollte entsprechend weniger Analyseaufwand betrieben werden.

4 REPMUS – Requirements Engineering Process for Market-Driven Projects Facing Unstable Requirements in Small and Medium Enterprises

In diesem Kapitel definieren wir einen beispielhaften RE-Prozess, der die aus der Simulation gewonnenen Erkenntnisse integriert und gleichzeitig die Anforderungen an das RE in änderungsintensiven Projekten aus dem Kapitel 2 berücksichtigt. Unser Prozess baut dabei auf dem Grundgerüst von Wiegers' Ansatz (Wiegers 2005) auf, beschreibt auf detaillierterer Ebene als Wiegers, wie die einzelnen Aktivitäten ausgeführt werden sollten und weicht in einigen Punkten ab, um den speziellen Anforderungen der von uns betrachteten Projektklasse nachzukommen. Mittels einer Literaturstudie und Interviews mit Industriepartnern sammelten wir die in der betrachteten Projektklasse erfolgreich eingesetzten RE-Techniken. Diese wurden bei der Ausgestaltung berücksichtigt beziehungsweise integriert.

4.1 Motivation und Beitrag

Uns ist kein RE-Prozess aus der Literatur bekannt, der sich den speziellen Herausforderungen von änderungsintensiven und marktgetriebenen Softwareentwicklungsprojekten von KMU widmet. Gerade in KMU wird mit Entwicklungsprozessen oft nachlässig umgegangen, obwohl ein klarer, definierter Entwicklungsprozess wesentlich zum Projekterfolg beitragen und das Risiko des Scheiterns eines Projektes mindern kann. Dies betrifft in besonderem Maße den RE-Prozess. Es finden sich in der Literatur Hinweise, wie mit besonders instabilen Anforderungen umgegangen werden kann. Auch findet sich eine Reihe von Empfehlungen, welche RE-Techniken für den Einsatz in Projekten geeignet sind, in denen eine kurze Projektlaufzeit hohe Priorität hat. Auf dieser Basis bauen wir auf und integrieren in einem Prozess geeignete Techniken, die sich in änderungsintensiven Projekten bewährt haben und zudem berücksichtigen, dass eine geringe Projektlaufzeit und geringe Kosten angestrebt werden. Als Grundlage, beziehungsweise Grundgerüst für unseren Prozess verwenden wir den bewährten Ansatz von Wiegers (2005). Von diesem übernehmen wir den grundlegenden Workflow und die Aktivitäten, die dem Prozess zugrunde liegen. Wir nehmen in unseren Ansatz teilweise detailliertere Beschreibungen zur Durchführung der einzelnen Aktivitäten auf, definieren Produkte, die dabei erstellt werden und nehmen Anpassungen des Ansatzes vor, um den speziellen Anforderungen der von uns betrachteten Projektklasse nachzukommen. Eine zentrale Rolle spielen in unserem RE-Prozess zudem die Erkenntnisse aus Kapitel 3. Wie im vorangegangenen Kapitel gezeigt, ist es für den hinsichtlich Projektlaufzeit und –kosten geeigneten Umgang mit instabilen Anforderungen sinnvoll den RE-Aufwand an die Stabilität und Stabilisierbarkeit der Anforderungen anzupassen. Wichtiger Bestandteil des Prozesses ist demnach die Bewertung des Änderungsrisikos der Anforderungen und eine entsprechende Anpassung der Vorgehensweise an dieses Risiko. Essenziell ist zudem das Zurechtschneiden des Prozesses auf die speziellen Bedürfnisse von KMU und auch die Einbeziehung der besonderen Schwierigkeiten, die sich in marktgetriebenen Softwareentwicklungsprojekten ergeben.

Unser Beitrag in diesem Kapitel liegt zum einen darin, dass aus der Literatur die für die von uns betrachtete Projektklasse geeigneten RE-Techniken zusammengetragen und

diejenigen ausgewählt werden, die in Kombination eingesetzt werden können. Dabei sind vor allem die speziellen Anforderungen an den RE-Prozess zu berücksichtigen, die sich von der Instabilität der Anforderungen, der Markgetriebenheit des Softwareentwicklungsprojektes und aus den organisatorischen Gegebenheiten bei KMU ableiten. Von Bedeutung ist der Mehrwert der sich aus der Kombination und den Synergieeffekten der einzelnen Techniken ergibt. Ein weiteres Novum und somit einer der wichtigen Beiträge in diesem Kapitel ist, dass unser RE-Prozess den Aufwand, der für die Analyse von Anforderungen betrieben wird, an das ihnen eigene Änderungsrisiko anpasst. Es wird dabei berücksichtigt, wie instabil eine Anforderung eingeschätzt wird und inwieweit sie als stabilisierbar gilt. Das Ergebnis dieses Kapitels ist ein RE-Prozess, der sich den speziellen Herausforderungen von änderungsintensiven und marktgetriebenen Softwareentwicklungsprojekten von KMU widmet, sich dabei auf bewährte Techniken aus der Praxis abstützt und neue Erkenntnisse dieser Arbeit, vor allem in Bezug auf die Bewertung und Berücksichtigung des Änderungsrisikos von Anforderungen, integriert.

4.2 Kurzbeschreibung des RE-Ansatzes von Wiegers

REPMUS baut auf dem RE-Ansatz von Wiegers auf. In diesem Kapitel geben wir einen kurzen Überblick über diesen Ansatz. Eine detaillierte Beschreibung kann Wiegers (2005) entnommen werden. Die Unterschiede zwischen REPMUS und Wiegers' Ansatz werden in den Kapiteln 4.7 und 4.9 dargestellt und bewertet.

Bei Wiegers' RE-Ansatz finden sich zwei unterschiedliche Phasen. Eine erste RE-Phase wird ganz zu Beginn des Projekts durchgeführt. In dieser wird die Systemvision entwickelt und die Grenzen des Systems werden abgesteckt. Benutzerklassen und Benutzervertreter werden identifiziert und es wird bestimmt wer Entscheidungen über Anforderungen trifft. Die Techniken, mit denen Anforderungen ermittelt werden sollen, werden ausgewählt. Schließlich wird versucht sämtliche Use Cases zu identifizieren und zu priorisieren. Die zweite RE-Phase wird direkt zu Beginn jeder Iteration durchgeführt. Sie erfolgt ebenfalls sequenziell vor Beginn der Realisierung. In dieser Phase werden die Use Cases präzisiert, Qualitätsattribute spezifiziert, funktionale Anforderungen abgeleitet, dokumentiert und modelliert. Die Spezifikation wird einem Review unterzogen und es werden Prototypen entwickelt. Daraufhin wird die Architektur entwickelt und Anforderungen den einzelnen Komponenten zugewiesen. Schließlich werden Testfälle entwickelt und Use Cases, funktionale Anforderungen, Analysemodelle und Prototypen validiert. Abbildung 4-1 stellt die einzelnen Schritte in Wiegers' RE-Prozess dar. Die einzelnen Schritte werden normalerweise in ihrer numerischen Reihenfolge durchgeführt. Davon kann jedoch auch abgewichen werden.

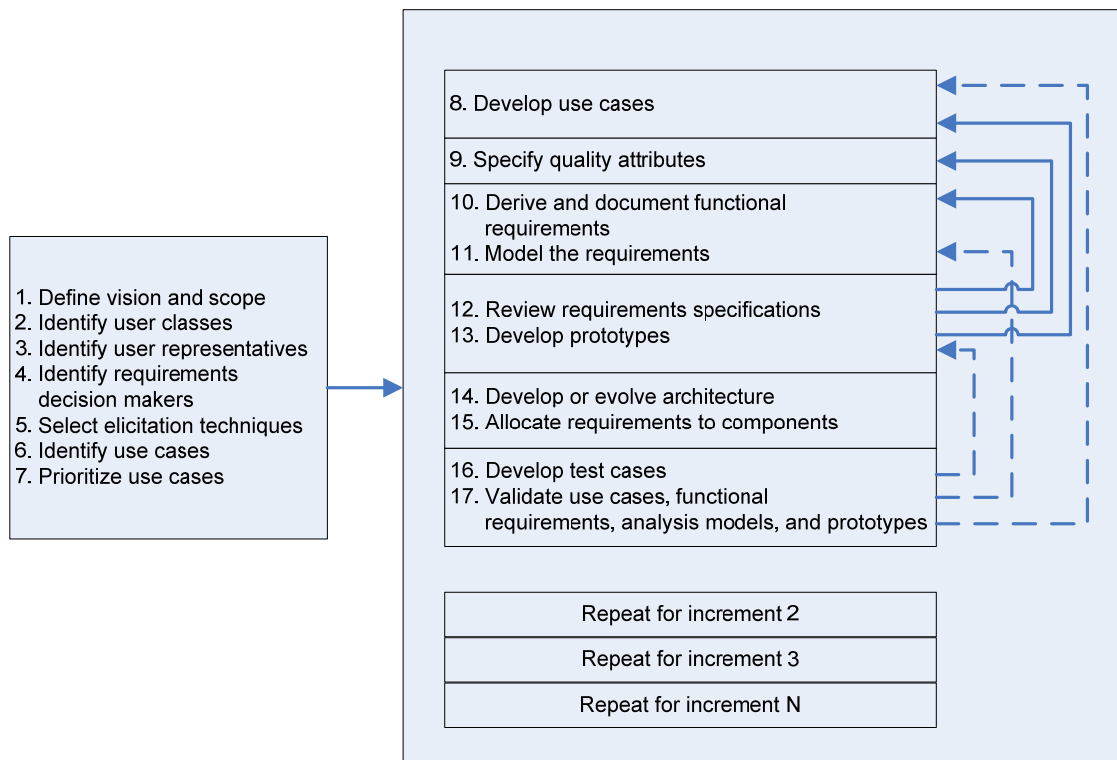


Abbildung 4-1: RE-Prozess in Wiegiers' RE-Ansatz (nach Wiegiers 2005)

Wiegiers unterteilt den RE-Prozess in die Aktivitäten „elicitation“, „analysis“, „specification“, „validation“ und „requirements management“. Die Aktivitäten werden nicht zwangswise linear durchgeführt, sondern verschränkt, inkrementell und iterativ (siehe hierzu Abbildung 4-2).

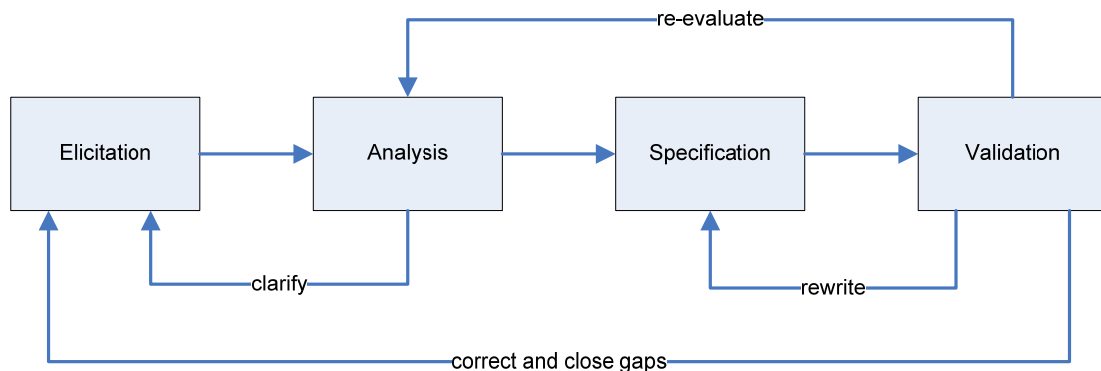


Abbildung 4-2: Grundlegender Workflow in Wiegiers' RE-Ansatz (nach Wiegiers 2005)

Wiegiers schlägt für die einzelnen RE-Aktivitäten eine Reihe von Praktiken vor. Diese sind in Tabelle 4-1 zusammengefasst.

Elicitation
Prozess für die Anforderungsentwicklung definieren
Vision aufstellen und Systemgrenzen abstecken
Benutzerklassen und ihre Charakteristika identifizieren
Für jede Benutzerklasse einen Produktchampion bestimmen, der die Benutzerklasse repräsentiert und Entscheidungen über Anforderungen trifft
Fokusgruppen einrichten, in denen der Input typischer Benutzer hinsichtlich gewünschter Funktionalitäts- und Qualitätscharakteristika gesammelt wird
Use Cases identifizieren
Systemereignisse und –antworten identifizieren
Ermittlungsworkshops abhalten
Feldbeobachtung durchführen
Problembereiche bestehender Systeme analysieren
Anforderungen aus anderen Projekten wiederverwenden
Analysis
Kontextdiagramm zeichnen
Prototypen erstellen und bewerten
Machbarkeit der Anforderungen analysieren
Anforderungen priorisieren
Anforderungen modellieren
Glossar erstellen
Anforderungen Subsystemen zuordnen
Quality Function Deployment zur Bewertung der Wichtigkeit von Anforderungen einsetzen
Specification
Anforderungen gemäß einem Template für die Spezifikation dokumentieren
Quellen für Anforderungen identifizieren
Anforderungen eindeutig bezeichnen
Qualitätsanforderungen spezifizieren
Geschäftsregeln dokumentieren
Validation
Anforderungsdokumente inspizieren
Anforderungen testen
Akzeptanzkriterien definieren
Requirements Management
Prozess für das Änderungsmanagement definieren
Change Control Board einrichten
Impactanalyse für Anforderungsänderungen durchführen
Baseline der Anforderungsdokumente festlegen und diese unter Versionskontrolle stellen
Änderungshistorie erstellen
Zustand von Anforderungen verfolgen
Anforderungsinstabilität messen
Ein Werkzeug für das Requirements Management benutzen
Anforderungen mit einer Traceability Matrix verfolgen

Tabelle 4-1: Empfohlene Praktiken zur Durchführung der Aktivitäten im RE-Ansatz von Wiegers

In den folgenden Kapiteln wird REPMUS beschrieben. Es wird dabei unter anderem darauf eingegangen welche Unterschiede zu Wiegers' Ansatz bestehen. Eine Zusammenfassung der Unterschiede folgt in Kapitel 4.9.

4.3 Einordnung in den Entwicklungsprozess

REPMUS beginnt, nachdem der Kick-Off des Projektes erfolgte und setzt eine bereits erstellte Systemvision voraus. Als Output liefert der Prozess eine Anforderungsspezifikation. Er läuft teilweise parallel zur Entwicklung ab und endet erst wenn keine Änderungen am System mehr durchgeführt werden sollen, was in der Regel erst bei Fertigstellung des letzten Releases des Systems der Fall ist.

Projekte mit besonders instabilen Anforderungen beinhalten in der Regel ein hohes Risiko des Scheiterns. Daher sollte vor Projektbeginn unter anderem eine Einschätzung des Änderungsrisikos vorgenommen werden und inwieweit dieses Risiko abgefangen werden kann. Dabei sind Faktoren wie die Zuverlässigkeit und Verfügbarkeit der Stakeholder, die Instabilität der Anforderungen, die Beherrschbarkeit dieser Instabilität und viele weitere Parameter zu berücksichtigen. Bei einem für die Organisation zu hohem Risiko kann es sinnvoll sein, sich gegen eine Durchführung des Projektes zu entscheiden. Solche Betrachtungen sind jedoch nicht Teil dieser Arbeit.

4.4 Generelle Strategie

In diesem Kapitel stellen wir die generelle Strategie von REPMUS vor und begründen sie. Dazu führen wir wichtige Grundannahmen, die die Wahl des Vorgehens beeinflussen, auf.

Änderungswünsche an Anforderungen treten während des gesamten Projekts auf: In der Vergangenheit wurde oft gefordert, dass nach Abschluss einer initialen RE-Phase die Anforderungen stabil bleiben müssen. Diese Sicht wird inzwischen als unrealistisch betrachtet (Reifer 2000). Kotonya und Sommerville (1998) argumentieren, dass Anforderungsänderungen unvermeidbar sind. Sie führen an, dass Anforderungsänderungen nicht zwingend auf Mängel im RE zurückzuführen sind. Zum selben Schluss sind wir in Kapitel 2.2.1 gekommen, in dem wir eine Reihe von Ursachen für Anforderungsänderungen aufführen, die nicht von der Vorgehensweise im Projekt beeinflusst werden können. Wir gehen daher davon aus, dass sich in der von uns betrachteten Projektklasse eine beträchtliche Zahl von Anforderungen während des gesamten Projekts ändert, und sich dies für viele Anforderungen auch nicht verhindern lässt. Wir gehen von dem üblichen Fall aus, dass man für einen Teil der Anforderungen mit relativer Sicherheit sagen kann, dass er stabil bleiben wird, und ein anderer Teil als instabil gilt. Eine weitere Annahme ist, dass sich unter den auftretenden Änderungen eine beträchtliche Anzahl befindet, die nicht ignoriert werden kann ohne den Projekterfolg zu gefährden. Zudem gehen wir davon aus, dass eine große Zahl an neuen Anforderungen hinzukommt, von denen ebenfalls viele nicht außen vor gelassen werden können.

RE wird während des gesamten Projekts betrieben: Laut der Studie von Hofmann und Lehner (2001) ist es in einem Umfeld mit instabilen Anforderungen wichtig, dass der RE-Prozess Lernprozesse während des Projektverlaufs berücksichtigt und Abstand davon nimmt die Anforderungsspezifikation „einzufrieren“. Aus diesen Gründen erheben, analysieren, spezifizieren, validieren und verhandeln wir Anforderungen, seien dies neue oder geänderte, während des gesamten Projektes.

RE wird iterativ betrieben: Wie schon von Wiegers (2005) empfohlen, setzt REPMUS auf eine iterative Durchführung des RE. Gerade unter der Annahme, dass sich Anforderungen während des gesamten Projekts in großem Umfang ändern werden, ist es sinnvoll den RE-Aufwand über die Projektlaufzeit zu verteilen. Die iterative Vorgehensweise im RE nutzen wir gezielt, um die Identifikation von Änderungswünschen zu katalysieren. Zentral sind in dieser Hinsicht zum einen Iterationsreviews, in denen die Stakeholder anhand der Entwicklungsergebnisse überprüfen, ob ihre Vorstellungen sich korrekt in diesen wiederfinden, oder ob bestimmte Anforderungen geändert werden müssten. Zum anderen ist das Identifizieren von Änderungswünschen ein Bestandteil der Anforderungsermittlung zu Beginn jeder Iteration. Über während der Entwicklung auftauchende Änderungswünsche kann falls notwendig auch unmittelbar entschieden werden. Der Normalfall bei REPMUS ist jedoch, dass Änderungswünsche durch Iterationsreviews und Analysephasen zu Beginn einer Iteration identifiziert werden und die Entscheidung über ihre Annahme in der RE-Phase zu Beginn einer Iteration erfolgt. Bei der Entwicklung der ersten Version eines Produkts für den Markt stehen anfangs noch keine tatsächlichen Benutzer und Kunden als Stakeholder zur Verfügung. Diese können erst in das Projekt involviert werden, wenn die erste Version auf den Markt kommt. Dies ist ein weiterer Faktor, der impliziert, dass die RE-Aktivitäten iterativ durchgeführt werden sollten (Sawyer et al. 1999). Regnell et al. (1998) betonen ebenfalls, dass bei marktgetriebener Softwareentwicklung Anforderungen während des gesamten Projekts ermittelt werden sollten.

Exponentielle Zunahme der Kosten für Änderungen an Anforderungen im Projektverlauf: Je später im Projekt sich eine Anforderung ändert, desto teurer ist es die damit verbundenen Anpassungen an Entwicklungsartefakten vorzunehmen (Boehm 1981; Boehm und Basili, 2001; Nurmaliani et al. 2006; Pohl 2007).

Gründlichere Analyse von Anforderungen abhängig von ihrem Änderungsrisiko und ihrer Stabilisierbarkeit: Einer der zentralen Aspekte von REPMUS ist zu bestimmen bis zu welchem Grad welche Anforderung analysiert werden soll. Hofmann und Lehner (2001) betonen, dass dieser Aspekt wesentlich den Projekterfolg beeinflusst. Wie schon in Kapitel 2.2.1 diskutiert gibt es eine Reihe von Anforderungsänderungen, die durch ein entsprechendes Vorgehen vermieden werden können. Drei der Hauptgründe für die Instabilität von Anforderungen sind mangelndes Verständnis der Anforderungen, falsche Abschätzung der Kosten von Anforderungen und falsche Einschätzung der Machbarkeit von Anforderungen. In Kapitel 3 haben wir gezeigt, dass sich durch eine Steigerung des Analyseaufwands die Änderungsrate bestimmter Anforderungen reduzieren lässt und somit Änderungsaufwand eingespart werden kann. Wir konnten dabei feststellen, dass es insbesondere Sinn macht, den Analyseaufwand auf Anforderungen zu konzentrieren, die einem hohen Änderungsrisiko ausgesetzt sind. Unser Ansatz sieht daher vor, dass Anforderungen, die als stabilisierbar gelten, die also instabil eingeschätzt werden und bei denen man sich von einer Analyse eine Reduzierung dieser Instabilität verspricht, besonders gründlich analysiert werden. Auch Tomayko (2002) hält fest, dass schlecht verstandene Anforderungen ein Risiko für teure Änderungen am System beinhalten, und ihnen daher besondere Beachtung zuteil werden sollte. Einer besonders gründlichen Analyse sollten Anforderungen mit einem hohen Impact, also solche die besonders hohe Änderungskosten verursachen können, unterzogen werden. Dies gilt insbesondere für viele nichtfunktionale Anforderungen (Goetz 2002). Anforderungen die als besonders instabil gelten, die aber nicht stabilisiert werden können, werden keiner besonders gründlichen Analyse unterzogen.

Anpassung des Gesamtaufwands für RE an Änderungsrate und Stabilisierbarkeit der Anforderungen im Projekt: REPMUS strebt an, den Aufwand für RE hinsichtlich der Projektlaufzeit und –kosten günstig zu gestalten. Eine Reduzierung des RE-Aufwands führt nicht unbedingt zu einer Verringerung von Projektlaufzeit und –kosten. Wie die Studie von Nikula et al. (2000) zeigt, kann es sogar sinnvoll sein im Angesicht von TTM den RE-Aufwand zu erhöhen. Unsere Simulationsergebnisse haben ergeben, dass bei einer Variierung des RE-Aufwands Projektlaufzeit und –kosten ein Minimum durchlaufen. Für bestimmte Projektparameter gibt es also jeweils für die Zeit- beziehungsweise Kostenoptimierung ein optimales Maß an RE-Aufwand. Die Minima in Projektlaufzeit und –kosten können jedoch nicht genau bestimmt werden. In der Literatur werden unterschiedliche Grundwerte für RE angegeben. Die Arbeit von Boehm (1981) zeigt, dass Anfang der 80er ca. 6 Prozent der Kosten und 9-12 Prozent des Zeitaufwands für RE investiert wurden. Hofmann und Lehner (2001) betonen, dass sich seit der Studie von Boehm der Umgang mit RE stark verändert hat. Ihre Studie ergibt für das RE einen Anteil an den Gesamtkosten von 15,7 Prozent und einen Anteil an der Projektlaufzeit von 38,6 Prozent. COCOMO II (Boehm et al. 1997) gibt für einfache Projekte an, dass das RE 7 Prozent der Kosten und 22 Prozent der Projektlaufzeit ausmachen sollte. Letztendlich bleibt der optimale Anteil des RE am Aufwand stark abhängig von Organisation, Domäne und Art des Projektes. Eine erfahrene Organisation kennt in der Regel das Maß an RE-Aufwand, das sich in ihren typischen Projekten bewährt hat. Dieses ist stark von organisations- und domänenspezifischen Einflussfaktoren beeinflusst, wie dem Innovationsgrad der entwickelten Produkte, der Erfahrung der Organisation, etc. Für das laufende Projekt wird abgeschätzt, wie die Änderungsrate von Projekten, die für die Organisation typisch sind, abweicht und ob die Anforderungen besser oder schlechter stabilisierbar sind. Falls die Anforderungen als instabiler oder besser stabilisierbar eingeschätzt werden, so sollte mehr RE-Aufwand als im Regelfall aufgewendet werden. Bei stabileren oder schlechter stabilisierbaren Anforderungen hingegen sollte entsprechend weniger Aufwand für RE betrieben werden.

Wie bereits anfangs erwähnt ist TTM für marktgetriebene Unternehmen überlebenskritisch (Novorita und Grube 1996). Daher wird eine geringe Projektlaufzeit angestrebt. Card (1995) führt drei grundlegende Strategien auf, wie dies geschehen kann:

- Aufgaben gleichzeitig durchführen
- Aufgaben schneller durchführen
- Weniger Aufgaben durchführen

REPMUS versucht alle drei Punkte einzubeziehen.

Zur Entwicklung teilweise paralleles RE: Wir ermitteln, analysieren, spezifizieren, validieren und verhandeln Anforderungen zumindest teilweise iterativ und parallel zur Entwicklung. Ein solches Vorgehen entspricht den von Lee (2002) und Shore (2005) vorgebrachten Ideen. Auch Powell et al. (1999) weisen darauf hin, dass durch parallele Ausführung von Entwicklungsaktivitäten die Entwicklungszeit verkürzt werden kann. Zur Entwicklung paralleles und iteratives RE hat zusätzlich den Vorteil, dass man Feedback durch Prototypen und frühe Releases sammeln kann. Alexander und Stevens (2002) sowie Karlsson et al. (2002) betonen die Wichtigkeit eines solchen Feedbacks bei änderungsintensiven Projekten. Insbesondere wenn es sich bei dem Projekt um die Entwicklung der ersten Version eines Produkts handelt und es keine tatsächlichen Kunden und Benutzer gibt, sind Prototypen und frühe Releases wichtig, um auf diese Weise zumindest potenzielle Kunden und Benutzer einbeziehen zu können. Pinheiro (2002) unterstreicht die Wichtigkeit von Feedback stimuliert durch die Prototypentwicklung bei

zeitkritischen Projekten. Dieses Feedback ist ein wichtiges Instrument um rechtzeitig Änderungswünsche zu entdecken. Mittels Time Boxing in der Entwicklung wird die Regelmäßigkeit und Planbarkeit des Feedbacks gewährleistet. Oft treten Änderungen trotz einer initialen Analyse der Anforderungen erst dann auf, wenn Ergebnisse aus der Entwicklung vorliegen. Beginnt man mit der Implementierung bereits früh im Projekt und präsentiert den Stakeholdern Prototypen, so können einige Änderungen in frühere Phasen des Projektes verschoben werden (Jones 1996b). Laut der Studie von McPhee und Eberlein (2002) ist Prototyping generell als nützliche Technik für TTM-Projekte anzusehen. Frühe Releases können zudem den Druck durch TTM reduzieren. Zwischen der Spezifikation und der Implementierung einer Anforderung kann sich diese ändern. Wird die Analyse und Spezifikation einer Anforderung aufgrund des iterativen Vorgehens nach hinten verschoben, so umgeht man diese Änderung unter Umständen. Genauer gesagt, analysiert und spezifiziert man nur die neue, geänderte und nicht die ursprüngliche Anforderung, die später nicht mehr von Belang ist (Ambler 2002). Auf diese Weise wird ebenfalls der Aufwand reduziert.

Effiziente Durchführung der RE-Aktivitäten: Wir versuchen die RE-Aktivitäten möglichst effizient durchzuführen. Eins der Hauptmittel dazu ist die Verkürzung der Kommunikationswege, die dadurch erreicht wird, dass sämtliche Stakeholder, inklusive der Entwickler, in RE-Workshops zusammengeführt werden. Auf diesem Weg kann eine schnelle Abstimmung erfolgen. McPhee und Eberlein (2000) weisen darauf hin, dass die Entwicklungszeit so positiv beeinflusst werden kann. Zudem zeigt sich in der Praxis, dass mangelnde Kommunikation oft zu vermehrten Änderungen führt (Strens und Sugden 1995; Hall et al. 2002) oder die Qualität der Anforderungen und damit des Produkts verringert (Hofmann und Lehner 1995; Karlsson et al. 2002). Laut Jones (1996b) kann die Integration aller Stakeholder, inklusive der Entwickler, in Workshops die Änderungsrate sogar halbieren. Dies ist darauf zurückzuführen, dass unter diesen Bedingungen Anforderungen seltener missverstanden werden. In der Praxis zeigt sich, dass Gruppendiskussionen über Anforderungen das Verständnis meist beträchtlich erhöhen (Dahlstedt et al. 2003). Die Studien von Chatzoglou (1997) sowie Emam und Madhavji (1995) zeigen, dass insbesondere die Beteiligung von Benutzern und die Kommunikation mit diesen gemeinhin als essenziell für den Projekterfolg angesehen werden. Sieht man von Kunden und Benutzern ab, spielen in Projekten von KMU meist nur wenige Stakeholder eine entscheidende Rolle, und es ist in der Regel leicht diese in Workshops zusammenzuführen. Wie bereits zuvor erläutert können bei der Entwicklung der ersten Version eines Produkts Kunden und Benutzer ohnehin nicht berücksichtigt werden. Die Hauptschwierigkeit ist hierbei also die Integration von potenziellen Kunden und Benutzern bei der Entwicklung der ersten Version und tatsächlichen Kunden und Benutzern bei der Entwicklung von späteren Versionen. REPMUS baut vor allem auf die Verwendung von informellen und semi-formellen Notationen. Diese erlauben im Gegensatz zu formalen Notationen eine aufwandsärmere Erstellung und Änderung von Dokumenten und sind zudem leichter verständlich. REPMUS ist sehr stark von der Partizipation aller relevanten Stakeholder abhängig. Die bessere Lesbarkeit dieser Art von Notationen erleichtert die Einbindung der Stakeholder. Weiterhin muss bedacht werden, dass KMU meist nur leichtgewichtige Methoden verwenden, seien dies Prozesse oder einzelne Techniken. Sie besitzen oft nur unzureichende Kenntnisse über formale Notationen.

Regelmäßige Einbeziehung aller relevanten Stakeholder während des gesamten Projekts: Die Stakeholder müssen auch in späteren Phasen des Projektes eingebunden werden. Zum einen, weil wir RE während des gesamten Projekts durchführen und zum anderen wegen der Instabilität der Anforderungen. Wenn sich Anforderungen beständig ändern, dann ist es wichtig, regelmäßig oder kontinuierlich Informationen über etwaige Änderungen

einzuholen. Die primäre Quelle für Informationen über sich ändernde Anforderungen sind die Stakeholder. Laut Pinheiro (2002) hilft die Einbeziehung von Benutzern dabei schwierig zu verstehende Anforderungen zu klären. So werden Änderungen aufgrund von mangelndem Verständnis reduziert. Dies gilt nicht nur für Benutzer sondern für alle Stakeholder. Boehm et al. (1994) betonen, dass es generell für den Projekterfolg wichtig ist, alle relevanten Stakeholder zu beteiligen, und nicht nur eine Teilmenge. REPMUS sieht eine Beteiligung aller relevanten Stakeholder in regelmäßigen Intervallen vor. Von einer kontinuierlichen Beteiligung wie bei eXtreme Programming sehen wir ab. In Projekten, bei denen es viele Stakeholder mit unterschiedlichen Interessen und Sichten gibt, ist eine solche nur sehr schwer zu erreichen. Es ist möglich, wenn auch schwierig, wie bei XP einen Kunden zu finden, der permanent vor Ort verfügbar ist. Wir halten es jedoch für so gut wie unmöglich eine ganze Reihe von Stakeholdern zu einer kontinuierlichen Beteiligung zu bewegen oder jemanden zu finden, der die Interessen und Sichten aller relevanten Stakeholder ausreichend repräsentieren kann.

In den folgenden Kapiteln wird REPMUS im Detail vorgestellt.

4.5 Anforderungscharakterisierung

Wir definieren für Anforderungen eine Reihe von Attributen. REPMUS ist in diesem Punkt konform zu Wiegers' Ansatz. Die explizitere Miteinbeziehung des Änderungsrisikos stellt jedoch eine Anpassung dar.

Jeder Anforderung wird eine **Quelle** zugeordnet. Dies ist in der Regel eine Stakeholderklasse. Dokumente können auch Quellen für Anforderungen sein. Dies können beispielsweise Standards oder Gesetzestexte sein.

Für jede Anforderung wird ihre **Wichtigkeit** ermittelt. Diese leitet sich vom Geschäftswert der jeweiligen Anforderung ab. Wir definieren Wichtigkeit jedoch allgemein als Grad des Unwillens die Anforderung fallen zu lassen. Dies bringt zum Ausdruck, dass die Einhaltung von beispielsweise Standards oder gesetzlichen Regelungen eine hohe Wichtigkeit zur Folge hat, ohne dass damit ein Geschäftswert verbunden ist. Die Wichtigkeit sollte in grobe Kategorien eingeteilt werden, zum Beispiel in „sehr gering“, „gering“, „mittel“, „hoch“ und „sehr hoch“.

Für jede Anforderung werden die mit ihr verbundenen **Risiken** betrachtet. Besondere Beachtung kommt dem Änderungsrisiko einer Anforderung bei. Dazu gehört zum einen die Stabilität einer Anforderung, welche anzeigt für wie wahrscheinlich man es hält, dass die Anforderung sich ändern wird. Es handelt sich dabei allerdings nicht um einen präzisen Wert sondern nur um eine grobe Kategorisierung, zum Beispiel in „sehr gering“, „gering“, „mittel“, „hoch“ und „sehr hoch“. Für Anforderungen, die besonders instabil eingeschätzt werden, wird angegeben was die Gründe für die vermutete Instabilität sind. Zum anderen ist der Impact der Anforderungen zu beachten. Dieser bezieht sich auf die Zahl der Komponenten, die durch eine Änderung der Anforderung schätzungsweise betroffen werden würden. Auch dieser Wert ist nicht präzise sondern nur eine grobe Kategorisierung. Schließlich wird für instabile Anforderungen noch aufgeführt, inwieweit sie als stabilisierbar gelten, d.h. inwiefern durch entsprechende Analysen ihre Änderungswahrscheinlichkeit reduziert werden kann. Auch die Einschätzung der Stabilisierbarkeit von Anforderungen erfolgt in groben Kategorien wie zum Beispiel „sehr gering“, „gering“, „mittel“, „hoch“ und „sehr hoch“.

Zu jeder Anforderung werden die **Kosten** ihrer Realisierung abgeschätzt.

Zu jeder Anforderung wird ihre **Priorität** bestimmt. Diese hängt von Wichtigkeit, Kosten und Risiko der Anforderung ab. Sie stellt die Grundlage für die Entscheidung dar, wann eine Anforderung realisiert werden soll.

Für jede Anforderung wird ihr **Zustand** mitgeführt. Mögliche Zustände einer Anforderung sind entsprechend Wiegers (2005) „vorgeschlagen“, „angenommen“, „abgelehnt“, „realisiert“, „verifiziert“ und „gelöscht“.

Anforderungen, die in der Anforderungsermittlung eingebracht werden, erhalten den Zustand „vorgeschlagen“. Auch Anforderungsänderungswünsche erhalten diesen Zustand und werden analog zu Anforderungen behandelt. In der Anforderungsanalyse wird darüber entschieden, ob die Anforderung fallengelassen, geändert oder umgesetzt werden soll. In ersterem Fall geht sie in den Zustand „abgelehnt“ über. Soll sie geändert werden, so erhält sie ebenfalls den Zustand „abgelehnt“ und es wird eine neue geänderte Anforderung mit dem Zustand „vorgeschlagen“ eingebracht. Soll sie realisiert werden, ist ihr neuer Zustand „angenommen“. Wurde eine angenommene Anforderung umgesetzt so geht sie in den Zustand „realisiert“ über. Sobald die Erfüllung einer Anforderung im Produkt bestätigt wurde, erhält diese den Zustand „verifiziert“. Ergibt die Verifikation, dass die Anforderung im Produkt nicht erfüllt wird, so geht sie zurück in den Zustand „angenommen“. Es ist möglich eine bereits angenommene, realisierte oder verifizierte Anforderung wieder herauszunehmen. Sie erhält in diesem Fall den Zustand „gelöscht“.

Die möglichen Zustandsübergänge sind in Abbildung 4-3 dargestellt.

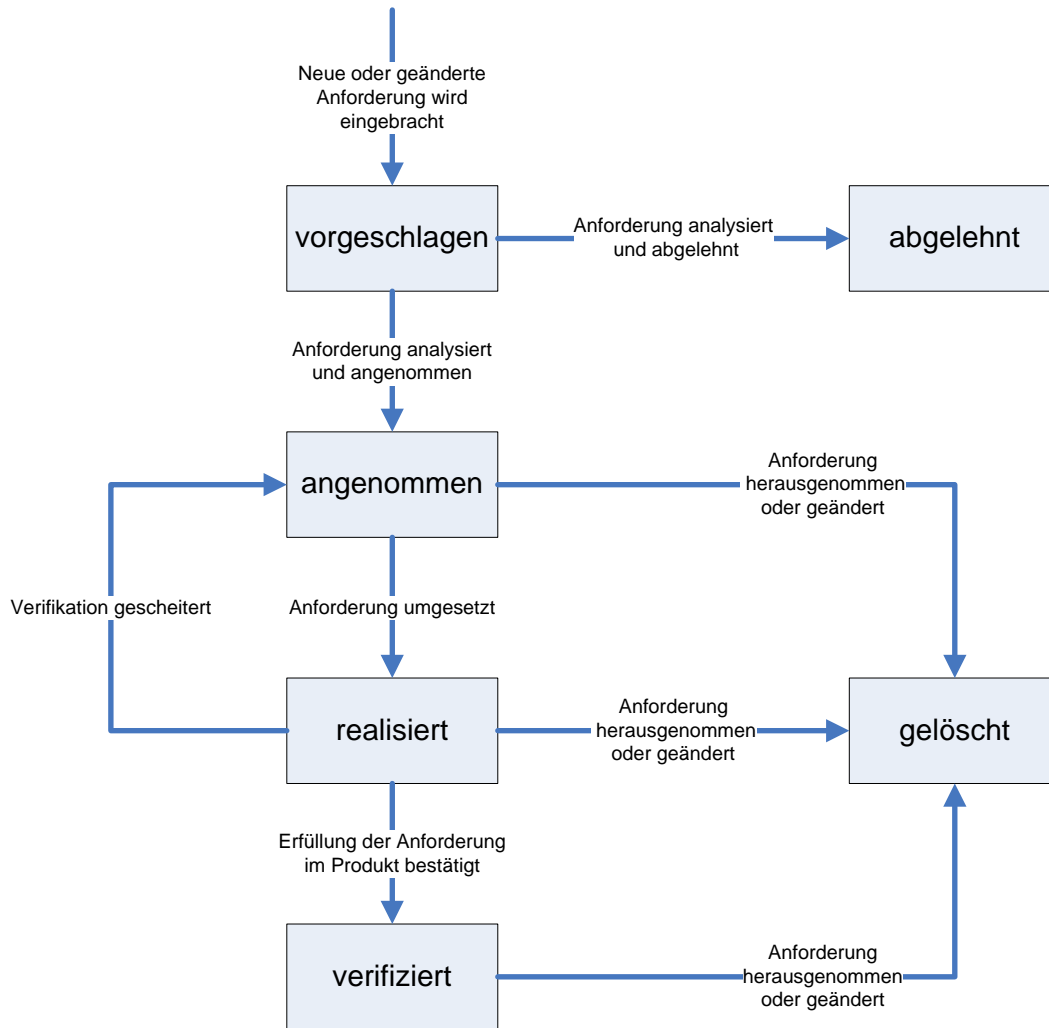


Abbildung 4-3: Zustandsübergänge von Anforderungen

4.6 Workflows

In diesem Kapitel beschreiben wir die grundlegenden Workflows von REPMUS, d.h. die Art und Weise wie die einzelnen Aktivitäten ineinandergreifen. Wir gehen dabei auch auf die beiden Phasen, in die der RE-Prozess untergliedert ist, ein.

Der grundlegende Workflow von REPMUS entspricht dem in Wiegers' Ansatz. REPMUS ist iterativ. Der Zyklus, der durchlaufen wird, besteht aus den Aktivitäten Anforderungen ermitteln, analysieren, spezifizieren und validieren. Dies stellt jedoch keine strikte sequenzielle Reihenfolge der Aktivitäten dar. Es kann in bestimmten Fällen zu vorangehenden Aktivitäten zurückgesprungen werden (siehe Abbildung 4-4). Zu betonen ist die starke Verzahnung der einzelnen Aktivitäten. Wenn man beispielsweise während der Anforderungvalidierung Unstimmigkeiten in der Spezifikation entdeckt, korrigiert man diese unmittelbar, springt also kurz zur Aktivität „Anforderungen spezifizieren“ zurück. Die starke Verzahnung der Aktivitäten entspricht der, von uns in kleineren und mittleren Projekten beobachteten, Praxis (Broy et al. 2010). Dies steigert die Akzeptanz des Ansatzes.

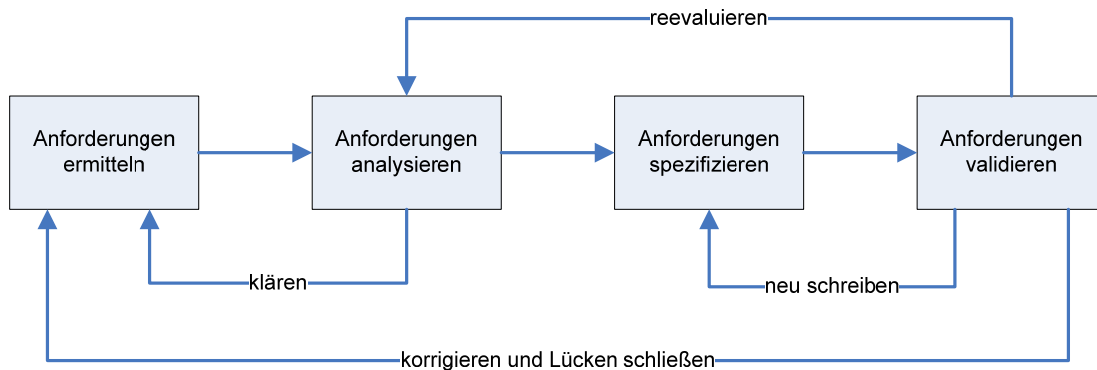


Abbildung 4-4: Grundlegender Zyklus des RE-Prozesses (nach Wiegers 2005)

Mit dem Requirements Management gibt es noch eine weitere Grundaktivität. Entgegen dem Ansatz von Wiegers umfasst das Requirements Management jedoch nicht den gesamten Anforderungsänderungsprozess. Da wir zum einen von einer starken Instabilität der Anforderungen ausgehen, ist das Behandeln von Änderungswünschen nicht so sehr ein Sonderfall sondern macht einen Großteil des RE aus. Zum anderen gehen wir davon aus, dass das RE während des gesamten Projekts in kleinen Iterationen erfolgt, und der Schwerpunkt des RE zu Beginn des Projektes eher gering ausfällt. Anforderungsänderungen müssen ebenfalls ermittelt, analysiert, spezifiziert und validiert werden. Daher beinhaltet unser grundlegender Zyklus des RE-Prozesses gleichzeitig den Änderungsprozess. Eine solche Vereinfachung des Prozesses kommt der Arbeitsweise in kleinen Projekten von KMU entgegen. Hierbei muss beachtet werden, dass Änderungswünsche selbstverständlich auch außerhalb der eigentlichen RE-Phasen eingebracht und falls nötig unmittelbar auch bewertet und beschlossen werden können. Der Normalfall sieht jedoch vor, dass über Änderungswünsche, die außerhalb einer RE-Phase eingebracht werden, erst in der nächsten RE-Phase entschieden wird.

Bestimmte Aktivitäten weisen bei REPMUS Subaktivitäten auf. Diese finden sich bei Wiegers' Ansatz nicht explizit als solche und stellen eine Detaillierung dar. Der Grundzyklus mit allen Subaktivitäten ist in Abbildung 4-5 dargestellt.

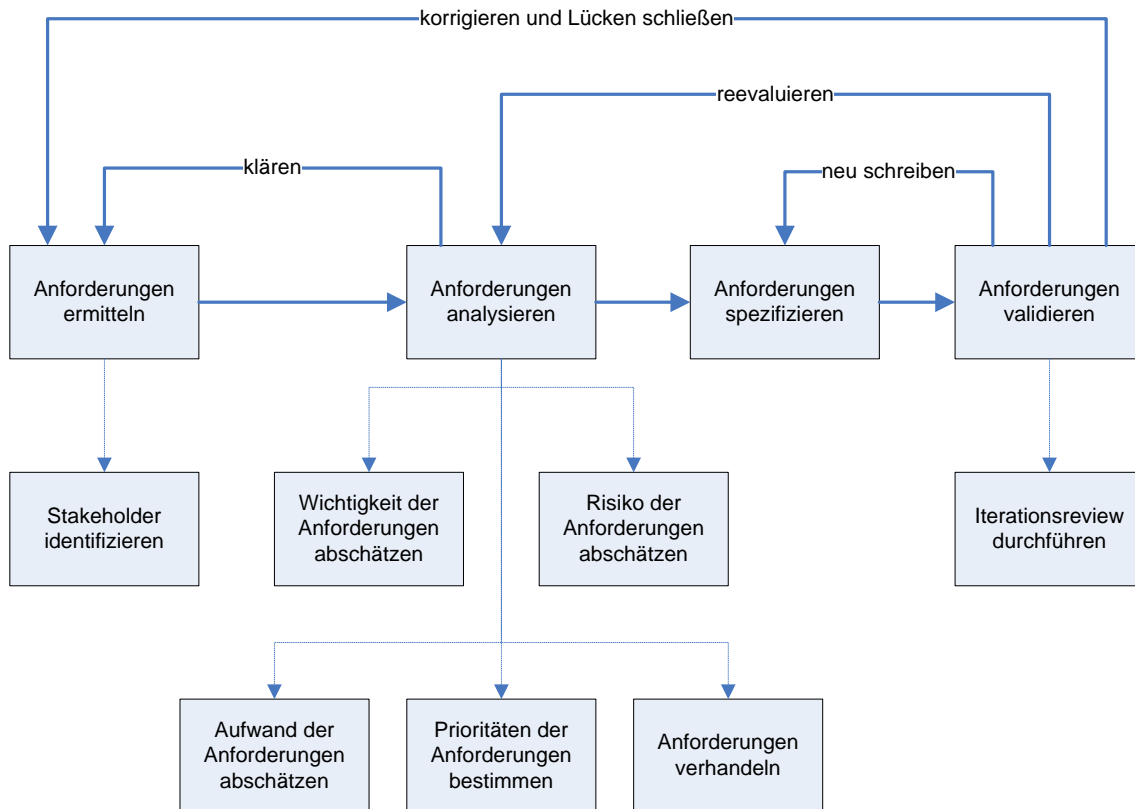


Abbildung 4-5: Grundlegender RE-Zyklus mit allen Subaktivitäten

REPMUS sieht entsprechend der Empfehlung von Wiegers für iteratives RE zwei Phasen vor:

1. Die Phase „**Initiales RE**“ ist eine intensive RE-Phase zu Beginn des Projektes und dauert bis zum Beginn von Design und Implementierung.
2. Die Phase „**Nebenläufiges RE**“ startet wenn mit dem Design begonnen wird und dauert bis Projektende. Dabei werden am Ende von Entwicklungsiterationen immer wieder kleine RE-Iterationen eingeschoben, die parallel zur Entwicklung ablaufen.

Bevor Design und Implementierung beginnen wird die Phase „Initiales RE“ durchgeführt. Die Identifikation der Stakeholder wird hauptsächlich während dieser Phase durchgeführt. Der grundlegende RE-Zyklus bestehend aus Anforderungsermittlung, -analyse, -spezifikation und -validierung wird mehrfach durchlaufen. In dieser Phase kann das Verständnis für Anforderungen erhöht und die Anforderungen stabilisiert werden. Dies ist insbesondere bei änderungskritischen Anforderungen wichtig. Wir konzentrieren uns in den ersten Iterationen zunächst auf high-level Anforderungen. In den folgenden Iterationen setzen wir uns schwerpunktmäßig mit Anforderungen auseinander, die ein hohes Änderungsrisiko aufweisen sowie denen, die in der ersten Entwicklungsiteration umgesetzt werden sollen. REPMUS macht in Ergänzung zu Wiegers Vorgaben die Länge der Phase abhängig von der Möglichkeit die Anforderungen zu stabilisieren. Je länger die Phase ist, desto stärker können Anforderungen stabilisiert und damit Aufwand für spätere Änderungen reduziert werden (Berry 2002; McPhee und Eberlein 2002). Es muss jedoch beachtet werden, dass das Verständnis für bestimmte Anforderungen wesentlich einfacher und schneller gewonnen werden kann, wenn bereits Feedback vorhanden ist, sei dies aus der Entwicklung in Form von Prototypen oder auch aus dem Einsatz des Systems in der

Zielumgebung. Man muss abwägen, wann der geeignete Zeitpunkt ist, mit Design und Implementierung zu beginnen und das RE parallel dazu weiterzuführen.

Die Phase „Nebenläufiges RE“ startet wenn mit dem Design begonnen wird. Ihr Ablauf entspricht im Wesentlichen der ersten Phase. Es kann jedoch zusätzlich Feedback aus der Entwicklung und dem Einsatz des Systems gesammelt und für die RE-Aktivitäten genutzt werden.

4.7 Aktivitäten

In diesem Kapitel wollen wir die einzelnen Aktivitäten von REPMUS vorstellen und erläutern. Wir beschreiben ihre Bedeutung für den Prozess und klären die grundlegende Art, wie sie durchgeführt werden sollen. Wir definieren welche Inputs sie verwenden und welche Outputs sie produzieren. Nicht alle Inputs müssen für die Ausführung der Aktivität vorliegen. Manche werden erst später im Prozess erstellt und können erst in Folgeiterationen berücksichtigt werden. Aufgrund der iterativen Vorgehensweise stellen fast alle Outputs gleichzeitig Inputs bei der nächsten Durchführung der Aktivität dar. Zu jeder Aktivität werden alle ihre Subaktivitäten, falls vorhanden, angegeben. Alle Aktivitäten werden normalerweise in Workshops abgehalten, in denen alle relevanten Stakeholder, die Entwickler inbegriffen, versammelt werden.

Die Aktivitäten sind dem Ansatz von Wiegers entnommen. Wir beschreiben detailliert, wie sie durchzuführen sind. Auf diesem Detaillierungsgrad sind die speziellen Anforderungen der von uns betrachteten Projektklasse zu berücksichtigen, so dass sich in der genauen Ausgestaltung der Aktivitäten Abweichungen von Wiegers' Ansatz ergeben.

4.7.1 Anforderungen ermitteln

4.7.1.1 Unterschiede zu Wiegers' Ansatz

Im Gegensatz zu Wiegers' Ansatz erfolgt in dieser Aktivität nicht die Erstellung der Systemvision. Diese setzt REPMUS als Input voraus.

Wiegers' Ansatz bezieht in die RE-Workshops nur Benutzer ein, während REPMUS alle relevanten Stakeholder daran beteiligt. In marktgetriebenen Softwareentwicklungsprojekten stehen bei der Entwicklung des ersten Release jedoch keine tatsächlichen Benutzer zur Verfügung und es muss auf andere Stakeholderklassen zurückgegriffen werden (Carmel und Becker 1995; Sawyer et al. 1999; Sawyer 2000). Handelt es sich bei dem Projekt um die Weiterentwicklung eines Produkts, so stellt sich dieses Problem nicht. Wie in Kapitel 2.2.1 aufgeführt sind die Hauptgründe für Anforderungsänderungen mangelndes Verständnis und falsche Abschätzung von Kosten, Priorität und Machbarkeit der Anforderungen. Diesen Faktoren kann durch eine Einbeziehung aller relevanten Stakeholder entgegengewirkt werden. Boehm et al. (1994) betonen zudem, dass eine Beteiligung aller relevanten Stakeholder für den Projekterfolg generell wichtig ist.

Weiterhin unterscheidet sich REPMUS von Wiegers' Ansatz darin, dass im Projektverlauf weitere Überprüfungen durchgeführt werden, ob zusätzliche Stakeholder einbezogen werden sollen. In marktgetriebenen Softwareentwicklungsprojekten, bei denen eine erste Version eines Produkts entwickelt wird, sind erst nach dem ersten Release tatsächliche Kunden und Benutzer verfügbar (Sawyer et al. 1999). Erneute Prüfungen im

Projektverlauf, ob weitere Stakeholder beteiligt werden sollten, sind daher nötig. Zudem steigt durch erneute Prüfung die Chance, alle relevanten Stakeholder zu berücksichtigen.

Unser Ansatz setzt auf informelle und semi-formelle Modellierungstechniken, um schwer zu verstehende Bereiche der Domäne genauer zu untersuchen. Wiegers' Ansatz hingegen untersucht sämtliche Bereiche der Domäne durch das Erstellen von Use Cases, der Identifikation von externen Ereignissen und den entsprechenden Antworten des Systems, Kontextdiagrammen, Feldbeobachtung und der Analyse von Problembereichen bestehender Systeme. Dabei wird kein Schwerpunkt auf schwer zu verstehende Bereiche gelegt. Das Vorgehen von REPMUS ist in diesem Punkt dem Streben nach einer möglichst effizienten Durchführung des RE geschuldet, um dem Ziel einer möglichst geringen Projektlaufzeit nachzukommen.

Wiegers setzt keinen so starken Fokus wie REPMUS auf das Nutzen von Feedback und Lernprozessen aus der Entwicklung, um die Anforderungsermittlung zu stimulieren. Gerade in Projekten, in denen viele Anforderungen kaum verstanden sind, ist dies wichtig, um das tatsächlich gewünschte System zu entwickeln (Regnell et al. 1998; Alexander und Stevens 2002; Karlsson et al. 2002).

Sein Ansatz setzt auch keinen Ideengeber wie unser Template zur Anforderungsermittlung ein. Die Anforderungsermittlung orientiert sich bei ihm lediglich an Use Cases und ist damit stark auf funktionale Anforderungen fokussiert. Wiegers fordert zwar, dass Anforderungen aus anderen Projekten wiederverwendet werden, stellt dafür aber keine Methode zur Verfügung, während wir mittels des Templates zur Anforderungsermittlung ein Werkzeug zumindest für die Wiederverwendung von high-level Anforderungen bereitstellen. Die Verwendung eines solchen Templates wird von Kotonya und Sommerville (1998) allgemein als Mittel zur schnelleren und vollständigeren Anforderungsermittlung angesehen und ist daher gerade in Projekten, in denen eine hohe Unsicherheit bezüglich der Anforderungen herrscht und zudem eine geringe Projektlaufzeit angestrebt wird, nützlich.

Während bei REPMUS die Reihenfolge, in der Anforderungen ermittelt beziehungsweise präzisiert werden, abhängig von ihrer Priorität und ihrem Änderungsrisiko ist, macht Wiegers' Ansatz diesbezüglich keine Vorgaben. Wie bereits mehrfach erläutert und in der Simulation in Kapitel 3 verdeutlicht kann durch eine frühe Analyse instabiler aber stabilisierbarer Anforderungen deren Änderungswahrscheinlichkeit gesenkt werden. Während bei Anforderungen, deren Änderungsrate nicht beeinflusst werden kann, es durchaus Sinn machen kann ihre Analyse auf einen späteren Zeitpunkt zu verschieben, um diese nicht mehrfach bei jeder Änderung erneut durchführen zu müssen. Laut Shore (2005) können durch eine frühe Analyse von stabilisierbaren Anforderungen Änderungskosten eingespart werden. Zu diesem Zweck müssen diese Anforderungen früh in detailliertere Anforderungen aufgebrochen werden. REPMUS setzt zudem darauf besonders kostenintensive Anforderungen früh aufzubrechen und zu analysieren, um Kosten- und Zeitfestlegungen möglichst präzise zu halten. Falsche Einschätzung der Kosten von Anforderungen wurden in Kapitel 2.2.1 als eine der Hauptursachen für die Änderung von Anforderungen identifiziert. Dieser wirkt REPMUS somit zumindest bei besonders kostenintensiven Anforderungen entgegen.

4.7.1.2 Beschreibung der Aktivität

Inputs	Outputs
Template für die Liste der Stakeholderklassen	
Liste der Stakeholder	Aktualisierte Liste der Stakeholder
Liste der Stakeholderklassen	Aktualisierte Liste der Stakeholderklassen
Template für die Anforderungsermittlung	Aktualisiertes Template für die Anforderungsermittlung
Bericht der Anforderungsvalidierung	
Liste der Anforderungen	Aktualisierte Liste der Anforderungen
Domänenmodelle	Aktualisierte Domänenmodelle
Systemvision	

Aufgaben
Stakeholder identifizieren
Neue Anforderungen aufstellen
Änderungswünsche an Anforderungen identifizieren
Quelle der Anforderungen dokumentieren
Im Bericht der Anforderungsvalidierung aufgeführte Fehler in der Liste der Anforderungen korrigieren

Das Ziel und die Notwendigkeit dieser Aktivität sind unmittelbar klar. Die Anforderungen sind der Entwicklung zu Beginn des Projektes nicht bekannt. Auch gibt es in der Regel keine Person, die alleine alle Anforderungen kennt. Sie müssen von unterschiedlichen Quellen in Erfahrung gebracht werden. Ziel dieser Aktivität ist es also die Quellen für Anforderungen zu identifizieren und die Anforderungen in einer Liste zusammenzutragen.

Die Anforderungsermittlung umfasst sowohl das Ermitteln neuer Anforderungen als auch die Identifikation von Änderungswünschen. In der Folge sprechen wir der Einfachheit halber nur noch von Anforderungen und adressieren damit gleichzeitig Änderungswünsche.

Wie die für das Projekt relevanten Stakeholder identifiziert werden wird in der entsprechenden Unteraktivität beschrieben.

Zu Beginn des Projekts werden ausgehend von der Systemvision Anforderungen ermittelt. Wie von Kotonya und Sommerville (1998) empfohlen nutzen wir ein Template für die Anforderungsermittlung als Leitfaden und Ideengeber bei der Ermittlung der Anforderungen. Durch dieses verläuft die Anforderungsermittlung strukturiert. Es ist zudem ein Werkzeug, das dabei hilft möglichst viele Anforderungen zu identifizieren. Auf diese Weise werden weniger Anforderungen übersehen, was wiederum spätere Änderungen reduziert. Da das Template eine umfangreiche Liste von spezifischen Qualitätsanforderungen beinhaltet, wird gewährleistet, dass schon zu Beginn des Projektes zu jeder dieser Qualitätsanforderungen eine Aussage gemacht wird. Gerade diese Art von Anforderungen wird oft in den frühen Phasen des Projektes übersehen, was später zu vielfältigen Änderungen führen kann. Der Umstand, dass durch den Einsatz des Templates zu allen Anforderungsklassen Anforderungen aufgestellt werden, erleichtert in der Anforderungsanalyse die Einschätzung von Änderungsrisiken. Es kann sich als nützlich erweisen das Template für die Anforderungsermittlung weiterzuentwickeln. Entwickelt ein

Unternehmen oft ähnliche Systeme, kann das Template an diese angepasst werden. So kann auch eine Klassifizierung der funktionalen Anforderungen im Template erarbeitet werden, die normalerweise nicht vorhanden ist.

Anforderungen werden sukzessive weiter aufgebrochen, um zu detaillierteren Anforderungen zu gelangen. Auf diese Weise entsteht eine Hierarchie von Anforderungen. Die neu ermittelten Anforderungen erhalten den Zustand „vorgeschlagen“. Im Folgenden sei ein Beispiel für eine Anforderungshierarchie aus einem industriellen Projekt gegeben (Paragon Software 2002). Es handelt sich dabei lediglich um einen Ausschnitt aus der Liste der Anforderungen.

- 1 *Creating and storing paintings – A user acting as an administrator can create and enter information about paintings, artists, galleries and locations and store them on the server on “Apogee Arts, Inc.”*
 - 1.1 *Every painting always belongs to only one "Theme" for example such as modern art or nature, etc.*
 - 1.2 *The system has a drop-down list of themes already created for an administrator to choose from while adding a new painting information.*
 - 1.3 *The graphical user interface (GUI) should allow the information of only one painting to be entered at a time.*
 - 1.4 *The “Painting Information” for a painting includes:*
 - 1.4.1 *The name or title of the painting. Every painting must have a unique name/title by which it can be identified.*
 - 1.4.2 *The image/picture of the painting. The image for a painting can be selected only from folders in the local file system through the GUI. Only jpg and gif files are considered as image files. Image size should be 300 X 300 pixels on the screen.*
 - 1.4.3 *The name of the artist and this should be linked to the “Artist Information” for that particular artist.*
 - 1.4.4 *The year in which the painting was known to have been created.*
 - 1.4.5 *The theme under which the painting can be classified. This should provide a link to “Theme Information” for that particular theme.*
 - 1.4.6 *The name and location of the gallery where the piece of art is currently located. The name of the gallery should provide a link to “Gallery Information” for that particular art gallery. The location (city and country where the gallery is located) should provide a link to “Country Information” for that particular location.*
 - 1.4.7 *A short description of the painting. This description can include references to other paintings, artists, art galleries and locations in which case, they should all point to the corresponding information section.*
 - 1.4.8 *The GUI should provide suitable fields and options to allow all the above information/description to be added to the particular painting information. All links should be accordingly displayed in the GUI.*
- 2 *User-friendliness – Since users of the VirtualArtViewer will be of all sorts and types and not all of them will be very experienced with computers, it is essential that the application be as user-friendly as possible.*

- 3 *Reliability – Reliability of the VirtualArtViewer is not expected to be essential, but nonetheless important. Accepted rate of failure is 1 crash per fortnight. Increased rate of failure will decimate the impression of “Apogee Arts, Inc.” products in the market.*
- 4 *Performance – It is highly essential that the VirtualArtViewer application runs efficiently. Any “search” should not take more than 5 seconds.*

Um in der Anforderungsanalyse sinnvolle Abschätzungen von Wichtigkeit, Aufwand und Risiko der Anforderungen treffen zu können müssen die Anforderungen auf ein entsprechendes Detaillevel heruntergebrochen werden. In obigem Beispiel sind die Anforderungen 1.1-1.3, 1.4.1-1.4.8 sowie 3 und 4 auf einem dafür geeigneten Detaillevel. Die Anforderung 1 ist bereits in detailliertere Anforderungen zerlegt, so dass für sie keine direkten Abschätzungen getroffen werden müssen. Für die Anforderung 2 ist es jedoch kaum möglich sinnvolle Abschätzungen zu treffen sofern sie nicht weiter detailliert wird.

Workshops zur Anforderungsermittlung, bei denen alle relevanten Stakeholder beteiligt werden, sehen wir als für die Methode unverzichtbar an. Diese können bei Bedarf, zum Beispiel wenn keine Vertreter bestimmter Stakeholderklassen für die Workshops zur Verfügung stehen, durch Fragebögen, strukturierte oder unstrukturierte Interviews ergänzt werden. Vor allem Workshops weisen den Vorteil auf, dass sie aufgrund intensiver Kommunikation schnell einen Austausch zwischen verschiedenen Parteien und Meinungen ermöglichen. Sie sind sehr gut geeignet die Bedürfnisse der Stakeholder zu erforschen und von diesen Anforderungen abzuleiten (Gottesdiener 2002). Gemeinsame Diskussionen über unterschiedliche Auffassungen fördern und beschleunigen den Kreativitätsprozess. Wie bereits in Kapitel 4.4 auseinandergesetzt kann durch den Austausch in Workshops schnell ein besseres Verständnis der Anforderungen erlangt werden kann. Dadurch wiederum werden spätere Änderungen verhindert. Näheres zum Thema RE-Workshops kann der entsprechenden Literatur entnommen werden (Martin 1991; Potts et al. 1994; Wood und Silver 1995; Graham 1998; Alexander und Stevens 2002; Gottesdiener 2002).

Es gibt eine Reihe weiterer Techniken, die bei der Ermittlung von Anforderungen eingesetzt werden können. Insbesondere in dem Fall, dass die erste Version eines Systems erstellt werden soll und noch keine tatsächlichen Kunden und Benutzer in das Projekt eingebunden werden können sollte man auf Marktstudien zurückgreifen (Keil und Carmel 1995; Deifel 1999). Falls bereits Systeme existieren, die eine hohe Ähnlichkeit mit dem zu entwickelnden System aufweisen, dann kann eine Analyse ihrer Dokumentation eine nützliche Ergänzung sein. Obwohl diese Vorgehensweise sehr aufwändig ist, kann sie sich bei wirklich ähnlichen Systemen als effizient und nützlich erweisen. Bei der Ermittlung der Anforderungen ist es außerdem wichtig, die Geschäftsziele sowie die Domäne zu verstehen. Entsprechende Experten bringen ihr Wissen in das Projekt ein. Es ist jedoch wichtig, dass auch die anderen Stakeholder und insbesondere die Entwicklung die Geschäftsziele und die Domäne verstehen. Nützliches Hilfsmittel um das Verständnis zu fördern ist die Geschäftsprozess- beziehungsweise Domänenmodellierung. Modelle stellen einen guten Katalysator für die Ideenfindung bei der Anforderungsermittlung dar. Hierbei sollten in unserer Situation vorrangig informelle und semi-formelle Modellierungstechniken eingesetzt werden. Diese werden laut der Studie von McPhee und Eberlein (2002) für TTM-Projekte als sehr nützlich angesehen. Zudem erleichtern sie, im Gegensatz zu formalen Modellen, die Integration von Stakeholdern, die mit speziellen Modellierungstechniken nicht vertraut sind, da sie intuitiver und schneller erlernbar sind. Szenarios (Rolland et al. 1998) werden vielfach als nützliches und einfaches Instrument zur Unterstützung der Anforderungsermittlung genannt (McPhee und Eberlein 2002;

Sommerville und Sawyer 1997) und sind zudem weit verbreitet (Neill et al. 2003). Erhöhtes Verständnis bedeutet gleichzeitig höhere Stabilität der Anforderungen und damit weniger Aufwand aufgrund von Änderungen. Die Erstellung der Modelle stellt jedoch einen gewissen Aufwand für sich selbst dar. Daher muss abgewogen werden wie viel Aufwand darin investiert werden soll. Je schlechter bestimmte Bereiche der Domäne verstanden sind, desto mehr Aufwand sollte man tendenziell für deren Verständnis betreiben. Zeitintensive und wenig effiziente Vorgehensweisen wie Feldbeobachtung und Apprenticing (Rupp 2007) sollten in der von uns betrachteten Projektklasse möglichst vermieden werden. Nur wenn die Anforderungen eines bestimmten Benutzertyps schlecht verstanden sind und kein Vertreter dieser Benutzerklasse am Projekt beteiligt werden kann, sollte man diese Techniken in Betracht ziehen.

Die gefundenen Anforderungen werden in der Liste der Anforderungen zusammengefasst. Es wird für jeden Eintrag die Quelle, also die Stakeholder oder Dokumente von denen die jeweilige Anforderung stammt, notiert.

Sind im Bericht der Anforderungvalidierung aus einer früheren Iteration Fehler in der Liste der Anforderungen aufgeführt, so müssen diese korrigiert werden.

Die Anforderungen werden nicht bei der ersten Ausführung dieser Aktivität vollständig ermittelt. Stattdessen erfolgt die Anforderungsermittlung iterativ. Ein entscheidender Punkt ist daher zu steuern welche Anforderungen zu welchem Zeitpunkt ermittelt werden. Während der initialen RE-Phase werden alle high-level Anforderungen ermittelt. Diese müssen bekannt sein, damit der Umfang und die Grenzen des Systems korrekt abgeschätzt werden können. Zudem sollten Anforderungen, die als kosten- oder zeitintensiv angesehen werden, genauer aufgespaltet werden. Man wird also entsprechende high-level Anforderungen weiter aufbrechen und somit die zugehörigen detaillierteren Anforderungen ebenfalls ermitteln. Dies ist nötig um Kosten- und Zeitfestlegungen möglichst präzise treffen zu können. Während des gesamten Projektes konzentriert man sich bei der Anforderungsermittlung vorrangig auf Anforderungen mit einem hohen Änderungsrisiko sowie diejenigen die als nächstes umgesetzt werden sollen. Das bedeutet man versucht Anforderungen zu ermitteln, die mit bereits identifizierten Anforderungen mit den eben beschriebenen Eigenschaften, verwandt sind, d.h. diesen untergeordnete Anforderungen und themenverwandte Anforderungen. Erstere erhält man durch Aufspaltung von entsprechenden Anforderungen. Letztere können entweder aufgrund der individuellen Erfahrungen der Stakeholder identifiziert oder mithilfe der Anforderungshierarchie bestimmt werden. Anforderungen, die als instabil gelten, bei denen aber klar ist, dass sie auch durch Analysen nicht stabilisiert werden können, zum Beispiel weil die Instabilität darauf zurückzuführen ist, dass zu verwendende Technologien, der Markt oder einzuhaltende Standards sich ändern, sollten erst zu einem späteren Zeitpunkt unmittelbar vor ihrer Umsetzung detaillierter aufgebrochen werden. Dies gilt allerdings nur wenn sie keinen Einfluss auf die aktuell betrachteten Anforderungen haben. Wie Shore (2005) anmerkt liegt der Nutzen der verzögerten Betrachtung dieser Anforderungen darin, dass ihre Analyse mit voranschreitender Entwicklung leichter vonstatten geht und darin, dass die Anforderungen aufgrund von Änderungen nicht immer wieder neu analysiert werden, ohne dass dadurch ein Nutzen für ihre Stabilisierung entsteht.

Wie auch von Regnell et al. (1998) empfohlen, wird die Anforderungsermittlung während des gesamten Projektes durchgeführt. Dies liegt wie in Kapitel 4.4 dargelegt zum einen daran, dass wir davon ausgehen, dass sich während des gesamten Projekts Anforderungen ändern und neue Anforderungen hinzukommen. Zum anderen liegt es daran, dass wir zur Entwicklung paralleles RE einsetzen, um Zeit einzusparen, und um durch Feedbackzyklen Synergieeffekte mit anderen Entwicklungsaktivitäten zu nutzen.

Stakeholder identifizieren

Inputs	Outputs
Template für die Liste der Stakeholderklassen	
Liste der Stakeholder	Aktualisierte Liste der Stakeholder
Liste der Stakeholderklassen	Aktualisierte Liste der Stakeholderklassen

Aufgaben
Stakeholderklassen identifizieren
Einen oder mehrere Vertreter für jede Stakeholderklasse benennen

Ziel dieser Unteraktivität ist es diejenigen Stakeholder zu identifizieren, die in das Projekt integriert werden sollen. Wie bereits in Kapitel 4.4 auseinandergesetzt ist es wichtig alle relevanten Stakeholder in das Projekt zu involvieren. Man ist auf diese angewiesen um Informationen über Anforderungen und Anforderungsänderungen zu bekommen. Sie werden zudem benötigt um Verständnisprobleme bezüglich der Anforderungen zu klären und Entscheidungen über Anforderungen zu treffen.

Zur Identifizierung von Stakeholdern nutzen wir eine Adaption der Methode von Alexander und Stevens (2002). Diese ergänzen wir durch die Verwendung einer Stakeholderklassifizierung. Wir versuchen alle Klassen von Stakeholdern, die für das Projekt relevant sein könnten, aufzustellen und für jede der Klassen einen oder mehrere Vertreter zu finden. Als Ausgangspunkt dient das Template für die Liste der Stakeholderklassen. Die darin enthaltenen Klassen werden auf Brauchbarkeit für das Projekt untersucht. Bei Bedarf werden weitere Klassen hinzugefügt. Dies kann auch dadurch geschehen, dass man die hierarchisch organisierten Klassen aus dem Template weiter aufbricht. Wurde eine Reihe von Stakeholdern identifiziert, so veranstaltet man mit diesen einen Workshop. Darin wird beraten welche weiteren Stakeholderklassen berücksichtigt werden sollten und welche Personen als Vertreter für Stakeholderklassen am Projekt beteiligt werden können. Somit werden eventuell weitere Stakeholder hinzugefügt. Dies wiederholt man so lange bis die Gruppe stabil bleibt. Sharp und Finkelstein (1999) geben eine Übersicht über alternative Methoden zur Stakeholderidentifikation. Bei der Auswahl der Stakeholder ist darauf zu achten, dass diese ausreichend Entscheidungskompetenz mit sich bringen. Die Stakeholder sollten im Projektverlauf unmittelbar Entscheidungen treffen können. Es würde den zügigen Ablauf hemmen wenn sie stets die Zustimmung anderer, nicht ins Projekt eingebundener Personen einholen müssten. Dies darf nur in Ausnahmesituationen der Fall sein. Den Stakeholderklassen „Auftraggeber“, beziehungsweise „Kunde“, oder „Marketing und Vertrieb“ sowie „Management des Auftragnehmers“ kommt demzufolge eine wichtige Rolle zu, da sie in der Regel die entsprechende Entscheidungskompetenz innehaben. Bei der Auswahl der Stakeholder sollte zudem deren Motivation und die Möglichkeit sie regelmäßig in das Projekt zu involvieren beachtet werden. Ist ein Kandidat nur selten verfügbar oder räumt dem Projekt kaum Priorität ein sollte er falls möglich durch einen anderen Vertreter für die Stakeholderklasse ersetzt werden.

Die identifizierten Stakeholder und Stakeholderklassen werden in die Liste der Stakeholder beziehungsweise die Liste der Stakeholderklassen eingetragen.

Diese Unteraktivität wird in der Regel nur einmal oder einige wenige Male in den frühen Phasen des Projekts durchgeführt. Später wird nur noch von Zeit zu Zeit überprüft, ob

weitere Stakeholder benötigt werden. Insbesondere wenn sich die technischen oder organisatorischen Rahmenbedingungen des Projektes ändern, sollte man beraten, ob neue Stakeholder beziehungsweise Stakeholderklassen einzubeziehen sind. Wie Sawyer et al. (1999) anmerken ist eine Schwierigkeit bei der Entwicklung der ersten Version eines Produkts, dass es noch keine tatsächlichen Endnutzer und Kunden gibt. In diesem Fall empfehlen sie, wie auch Hofmann und Lehner (2001) und Karlsson et al. (2002), verstärkt auf andere Stakeholder zurückzugreifen wie Entwickler, Marketing, potenzielle Benutzer und Kunden, etc. Sobald eine erste Version des Produkts am Markt ist sollten weitere Stakeholder wie tatsächliche Benutzer und Kunden sowie der Support involviert werden.

4.7.2 Anforderungen analysieren

4.7.2.1 Unterschiede zu Wiegers' Ansatz

Wiegers' Ansatz legt nicht wie REPMUS einen Schwerpunkt auf die Bewertung des Änderungsrisikos. Insbesondere erfolgt keine Abschätzung von Stabilität und Stabilisierbarkeit der Anforderungen. Im Gegensatz zu REPMUS definiert Wiegers ausführliche Maßnahmen zur Risikokontrolle, deren Erfolg zudem kontrolliert wird. REPMUS verwendet als einziges Mittel zur Risikobegrenzung das Abändern oder Fallenlassen von Anforderungen. REPMUS weicht in diesem Punkt von Wiegers' Ansatz ab, da in kleinen Projekten von KMU einfache Risikobegrenzungsmaßnahmen ausreichend sind. Eine ausführliche Bewertung des Änderungsrisikos ist hingegen erforderlich, da das Vorgehen in REPMUS an das Änderungsrisiko der Anforderungen angepasst wird.

Wiegers betreibt ausführlichere Konfliktanalysen. Weiterhin führt er für alle Anforderungen gründliche Analysen durch, um das Verständnis für sie zu erhöhen. REPMUS hingegen macht den Analyseaufwand abhängig von der Stabilisierbarkeit und dem Impact der Anforderungen. Dies spiegelt die in Kapitel 3 gewonnene Erkenntnis wider, dass es sich hinsichtlich der Minimierung von Projektlaufzeit und –kosten lohnt den investierten Analyseaufwand bis zu einem bestimmten Grad auf stabilisierbare Anforderungen zu konzentrieren. Mit der zusätzlichen Fokussierung auf Anforderungen mit hohem Impact folgen wir der Empfehlung von Goetz (2002), Anforderungen, die hohe Änderungskosten verursachen können, besonders gründlich zu analysieren.

Feedback und Lernprozesse aus der Entwicklung werden bei REPMUS mehr in den Vordergrund gerückt, während Wiegers verstärkt auf der Entwicklung vorangehende Analysen setzt. Bei änderungsintensiven Projekten, bei denen die Anforderungen nur schlecht verstanden sind ist es jedoch oft schwierig vor Beginn der Realisierung ein ausreichendes Verständnis für alle Anforderungen zu gewinnen. Feedback aus der Entwicklung wird in solchen Projekten als essenziell angesehen (Alexander und Stevens 2002; Karlsson et al. 2002). Dies gilt insbesondere im Fall von TTM-Projekten (Pinheiro 2002; McPhee und Eberlein 2002).

Schließlich ist noch hervorzuheben, dass REPMUS, anders als Wiegers, abhängig von Änderungsrate und Stabilisierbarkeit der Anforderungen den Gesamtaufwand für RE, insbesondere in der Anforderungsanalyse variiert. Damit trägt REPMUS dem in Kapitel 3 beobachteten Umstand Rechnung, dass abhängig von Änderungsrate und Stabilisierbarkeit der Anforderungen unterschiedlich hohe RE-Aufwände hinsichtlich Projektlaufzeit und –kosten günstig sind.

4.7.2.2 Beschreibung der Aktivität

Inputs	Outputs
Liste der Anforderungen	Aktualisierte Liste der Anforderungen
Wichtigkeit der Anforderungen	Wichtigkeit der Anforderungen
Aufwandsabschätzung der Anforderungen	Aufwandsabschätzung der Anforderungen
Priorität der Anforderungen	Priorität der Anforderungen
Risikoabschätzung der Anforderungen	Risikoabschätzung der Anforderungen
Traceability Links	
Änderungshistorie	
Domänenmodelle	
Anforderungsspezifikation	
Bericht der Anforderungvalidierung	
Iterationsreviewbericht	
Analysemodelle	Aktualisierte Analysemodelle
Konflikte zwischen Anforderungen	Aktualisierte Konflikte zwischen Anforderungen

Aufgaben
Wichtigkeit der Anforderungen abschätzen
Risiko der Anforderungen abschätzen
Aufwand der Anforderungen abschätzen
Priorität der Anforderungen bestimmen
Analysen zur Erhöhung des Verständnisses der Anforderungen durchführen
Konflikte zwischen Anforderungen identifizieren
Anforderungen verhandeln

Mit dieser Aktivität werden mehrere Ziele verfolgt. Zum einen sollen die für die Planung essenziellen Attribute Priorität und Risiko bestimmt werden. Um diese zu ermitteln müssen wir wiederum Wichtigkeit und Aufwand der Anforderungen abschätzen. Zum anderen sollen das Verständnis für die Anforderungen erhöht und Konflikte zwischen Anforderungen identifiziert und aufgelöst werden. Weiterhin wird über Änderungswünsche entschieden und es gilt Anforderungen, die ein zu hohes Risiko aufweisen, zu ändern oder fallen zu lassen.

Wie Wichtigkeit, Risiko, Aufwand und Priorität der Anforderungen bestimmt beziehungsweise abgeschätzt werden wird in den entsprechenden Unteraktivitäten beschrieben. Die Auflösung von Konflikten, die Entscheidung über Änderungswünschen sowie über risikobehaftete Anforderungen erfolgt in der Unteraktivität „Anforderungen verhandeln“.

Eine der wichtigsten Aufgaben bei der Anforderungsanalyse ist es das Verständnis für die Anforderungen zu erhöhen. Falls Domänenmodelle vorliegen, sind diese wie auch die bisher erstellten Teile der Anforderungsspezifikation ein nützlicher Input, der zum besseren Verstehen der Anforderungen und ihrer Implikationen beiträgt. Diskussionen in Workshops und das Erstellen von Analysemodellen sind weitere effektive Techniken um Anforderungen zu verstehen (Sugden und Strens 1996; Hofmann und Lehner 2001; Dahlstedt et al. 2003). Es ist jedoch bei der Verwendung von Modellierungstechniken

darauf zu achten, dass die verwendeten Notationen für die Stakeholder verständlich sind, und dass der Erstellungsaufwand nicht zu hoch ist (Emam und Madhavji 1995). In Hinblick auf das Anstreben einer geringen Projektlaufzeit haben sich in der Praxis informelle und semi-formelle Modellierungstechniken als nützlich erwiesen (McPhee und Eberlein 2002). Zum Beispiel aus Kapitel 4.7.1 könnte man eine graphische Skizze des Teils der GUI angefertigt werden, der die Informationen zu Bildern anzeigt und somit die Anforderungen 1.4.1-1.4.8 umsetzen soll. So könnten mit diesem informellen Modell die entsprechenden Anforderungen visualisiert werden und die Stakeholder können überprüfen, ob diese ihren Bedürfnissen entsprechen.

Man muss stets den Nutzen gegen den Zeitaufwand abwägen. Wie in Kapitel 4.4 erläutert, analysieren wir Anforderungen, die als stabilisierbar gelten oder einen hohen Impact aufweisen, besonders gründlich. In unserem Beispiel würde dies bedeuten, dass man Skizzen nur zu den Teilen der GUI erstellt, bei denen noch Unsicherheit in den zugehörigen Anforderungen herrscht. Zudem existiert bei REPMUS die Alternative mit der Realisierung zu beginnen und die Ergebnisse aus der Entwicklung quasi als Analysemodelle zu verwenden.

Aufgabe der Anforderungsanalyse ist zudem das Identifizieren von Konflikten zwischen den Anforderungen. Mit der Anforderungshierarchie steht bereits ein Beziehungsgeflecht der Anforderungen zur Verfügung. In der Anforderungshierarchie nah verwandte Anforderungen sind gute Kandidaten für sich widersprechende Anforderungen. Viele Konflikte können somit gefunden werden, wenn man nah verwandte Anforderungen gemeinsam analysiert. Das Gruppieren von Anforderungen ist für TTM-Projekte eine gängige und in der Praxis bewährte Methode, um Konflikte aufzudecken (Karlsson et al. 2002). Die gefundenen Konflikte werden im Produkt „Konflikte zwischen Anforderungen“ erfasst.

Die Anforderungsanalyse wird iterativ durchgeführt. Der iterative Ablauf ergibt sich daraus, dass in späteren Iterationen weitere Anforderungen in die Liste der Anforderungen aufgenommen werden und analysiert werden müssen.

Wichtigkeit der Anforderungen abschätzen

Inputs	Outputs
Liste der Anforderungen	Aktualisierte Liste der Anforderungen
Wichtigkeit der Anforderungen	Wichtigkeit der Anforderungen

Das Attribut Wichtigkeit wird benötigt um die Priorität von Anforderungen zu bestimmen. Zudem wird es bei der Konfliktauflösung benötigt um zu entscheiden welche Anforderungen Vorrang erhalten.

Eine Beteiligung aller relevanten Stakeholder ist, auch wenn in der Praxis oft zu schwer erreichen, gerade bei der Bestimmung der Wichtigkeit der Anforderungen von entscheidender Bedeutung (Hofmann und Lehner 2001). In der von uns betrachteten Projektklasse gibt es eine Vielzahl von nur schlecht verstandenen und sich ändernden Anforderungen. Das schlechte Verständnis und die Änderungswahrscheinlichkeit betreffen dabei auch die Wichtigkeit der Anforderungen. Aufgrund von Anforderungsänderungen kann sich die Wichtigkeit stark verschieben. Ebenso kann es geschehen, dass die Wichtigkeit von Anforderungen nachdem sich das Verständnis für sie erhöht hat völlig anders gesetzt wird. Um ein bestmögliches Verständnis der Anforderungen zu erreichen, ist es sinnvoll möglichst viele Stakeholder einzubeziehen, da diese zusätzliches Wissen

einbringen. Außerdem muss bedacht werden, dass verschiedene Stakeholdergruppen den Anforderungen ganz unterschiedliche Wichtigkeiten einräumen. Es würde das Ergebnis verzerren, wenn bestimmte Stakeholder nicht beteiligt würden.

Für die präzise Ermittlung der Wichtigkeit von Anforderungen hat sich der paarweise Vergleich der Anforderungen als besonders praktikabel erwiesen. Einige Stakeholderklassen, wie zum Beispiel Kunden und Marketing, sind versucht allen Anforderungen eine hohe Wichtigkeit zu geben. Bewertet man die Wichtigkeit von Anforderungen immer relativ zu anderen, so kann diesem Problem begegnet werden. Der paarweise Vergleich ergibt daher eine differenzierte Bewertung der Wichtigkeit der Anforderungen. Sowohl die Tendenz von Kunden alle Anforderungen als sehr wichtig einzuordnen wenn absolute Werte vergeben werden als auch die Umgehung dieses Problems durch den Einsatz von relativen Werten hat Davis (2003) in zwei Fallstudien gezeigt. Ab einer bestimmten Menge an Anforderungen kann durch Binäre Suchbäume (Karlsson et al. 1998) und Prioritätsgruppen (Karlsson et al. 1998) der Zeitaufwand gering gehalten werden. Bei einer geringen Anzahl an Anforderungen genügt eine geordnete Liste. Ein solches Vorgehen ist für alle Stakeholder leicht verständlich. Zudem ist der benötigte Aufwand bei kleinen und mittleren Projekten mit wenigen Anforderungen relativ gering. Ein besonders wichtiger Aspekt ist, dass das Vorgehen es erlaubt auf einfache Weise neuen Anforderungen beziehungsweise geänderten Anforderungen eine Wichtigkeit zuzuordnen. Alternative Möglichkeiten den Anforderungen Wichtigkeiten zuzuordnen werden in Karlsson et al. (1998), Karlsson et al. (2004) und Laurent et al. (2007) zusammengefasst. Hinsichtlich der Zeiteffizienz sind insbesondere die von Karlsson et al. (1998) beschriebenen minimalen Spannbäume sowie die Erweiterung des Analytic Hierarchy Process (Saaty 1980) um Hierarchien interessant.

Anforderungen, die für das Produkt unentbehrlich sind, sollten bei der Einteilung der Wichtigkeit außen vor gelassen werden und einen Sonderstatus erhalten (Wieggers 1999). Dies sind beispielsweise Anforderungen die sich aus gesetzlichen Regelungen ableiten, denen nicht ausgewichen werden kann, oder Anforderungen, ohne die das Produkt keinen Sinn macht (zum Beispiel bei der Entwicklung eines Personalmanagementsystems die Möglichkeit neue Personen einzufügen). Diese Anforderungen werden automatisch an den Anfang der nach Wichtigkeit geordneten Liste der Anforderungen gestellt. Sie stehen untereinander in keiner besonderen Reihenfolge.

Die Ergebnisse dieser Aktivität werden im Teilprodukt „Wichtigkeit der Anforderungen“ dokumentiert.

Die Wichtigkeitsabschätzung erfolgt iterativ, da in jeder Iteration neue Anforderungen hinzukommen können. Zudem kann sich die Wichtigkeit ändern. Alle neuen Anforderungen werden in die nach Wichtigkeit geordnete Liste der Anforderungen eingefügt.

Aufwand der Anforderungen abschätzen

Inputs	Outputs
Liste der Anforderungen	Aktualisierte Liste der Anforderungen
Aufwandsabschätzung der Anforderungen	Aufwandsabschätzung der Anforderungen
Bericht der Anforderungsvalidierung	
Iterationsreviewbericht	

Die Abschätzung des Aufwands zur Realisierung der einzelnen Anforderungen ist einer der zentralen Planungsparameter. Aufwände werden für die genaue Zeitplanung benötigt. Zudem braucht man sie um den Umfang von Iterationen und Releases geeignet festlegen zu können.

Wiegiers (2005) sieht vor, dass der Aufwand jeder Anforderung grob nach Kategorien (1-9) abgeschätzt wird. Ähnlich verfahren zum Beispiel auch Regnell et al. (1998). Eine weitere Möglichkeit ist das Bestimmen der relativen Kosten von Anforderungen über paarweisen Vergleich (Karlsson und Ryan 1997). In beiden Vorgehensweisen verlässt man sich bei der Kostenschätzung auf die intuitive Einschätzung von erfahrenen Entwicklern und den an der Realisierung beteiligten Stakeholdern. Durch die Beteiligung mehrerer Personen und insbesondere Personen mit Erfahrung in der Entwicklung ähnlicher Systeme kann die Präzision der Schätzung erhöht werden. Die intuitive Einschätzung der Kosten durch entsprechende Experten entspricht dem von uns in der Praxis beobachteten Vorgehen.

Bei der Abschätzung der Kosten pro Anforderung ist jedoch zu beachten, dass Anforderungen miteinander in Beziehung stehen können. Dies kann zur Folge haben, dass die Kosten für eine bestimmte Anforderung abhängig von der Ausgestaltung einer anderen Anforderung sind. Dies muss bei der Kostenschätzung berücksichtigt werden. Auch bei Änderungen von Anforderungen muss für sämtliche abhängigen Anforderungen überprüft werden, ob deren Kostenschätzung angepasst werden muss. Ein solches Vorgehen ist in der Praxis üblich und wird beispielsweise bei Nokia Siemens Networks seit Jahren erfolgreich eingesetzt.

Es ist nicht für alle Anforderungen möglich ihren Realisierungsaufwand isoliert anzugeben. Dies betrifft vor allem spezifische Qualitätsanforderungen wie die untenstehende aus dem Beispiel in Kapitel 4.7.1.

4 Performance – It is highly essential that the VirtualArtViewer application runs efficiently. Any “search” should not take more than 5 seconds.

Da eine Antwortzeit von höchstens fünf Sekunden bei einer Suche gefordert wird, müssen alle an einer Suche beteiligten Funktionen entsprechenden Performanzanforderungen genügen. Eine mögliche Lösung für dieses Problem ist das Einführen eines Multiplikators für die betreffende Anforderung, der auf den geschätzten Gesamtaufwand oder nur auf den von bestimmten Anforderungen angerechnet wird. Ein ähnliches Vorgehen verwendet COCOMO II (Boehm et al. 1997). Eine weitere Möglichkeit wäre es, Anforderungen, deren Kosten nicht isoliert angegeben werden können, weiter aufzuspalten. In dem gewählten Beispiel ist dies jedoch nicht praktikabel, so dass die erste Variante vorzuziehen ist.

Die Ergebnisse dieser Aktivität werden im Teilprodukt „Aufwandsabschätzung der Anforderungen“ dokumentiert.

Die Aufwandsschätzungen werden im Verlauf des Projekts immer wieder aktualisiert. Die Anforderungen werden zunehmend besser verstanden und somit sind Aufwandsschätzungen leichter vorzunehmen und zudem präziser. Das Feedback aus dem Iterationsreview wird für die Korrektur der Aufwandsschätzungen herangezogen.

Priorität der Anforderungen bestimmen

Inputs	Outputs
Liste der Anforderungen	Aktualisierte Liste der Anforderungen
Wichtigkeit der Anforderungen	
Aufwandsabschätzung der Anforderungen	
Priorität der Anforderungen	Priorität der Anforderungen

Die Priorität stellt einen der wichtigsten Planungsparameter dar. Sie wird herangezogen, um zu entscheiden, wann und ob eine Anforderung realisiert werden soll. Sie stellt dabei jedoch lediglich einen Indikator dar. In marktgetriebenen Unternehmen stellt die Priorisierung der Anforderungen den wichtigsten Teil der Anforderungsanalyse dar (Carlshamre 2001; Potts 2005; Regnell et al. 2003; Sawyer et al. 1999).

Bei der Bestimmung der Priorität der Anforderungen sollte deren Return on Investment die zentrale Rolle spielen (Karlsson und Ryan 1997, Regnell et al. 1998; Sawyer et al. 1999; Wiegers 1999). Die Priorität einer Anforderung richtet bei REPMUS entsprechend der Vorgaben von Wiegers nach ihrer Wichtigkeit, nach dem für ihre Realisierung geschätzten Aufwand sowie dem mit ihr verbundenen Risiko. Anforderungen, mit denen ein hoher Geschäftswert verbunden ist und die daher eine hohe Wichtigkeit aufweisen, sollten in einem frühen Release realisiert werden. Dadurch wird bewirkt, dass bei einer frühzeitigen Beendigung des Projekts die Anforderungen mit denen der höchste Geschäftswert verbunden ist, beziehungsweise die allgemein als wichtig angesehen werden, bereits realisiert wurden und nur weniger wichtige Anforderungen außen vor bleiben. Hohe Kosten wiederum verringern die Priorität einer Anforderung, da der Return on Investment bei diesen geringer ausfällt. Je höher das technische Risiko, das mit der Umsetzung einer Anforderung verbunden ist, desto niedriger ist seine Priorität.

In kleinen Projekten genügt eine informelle Einschätzung der Priorität der Stakeholder unter Berücksichtigung der beschriebenen Einflussfaktoren (Wiegers 1999).

Die Priorität wird herangezogen um zu bestimmen, in welchem Release eine Anforderung realisiert werden soll. Dörner (1996) warnt davor, Entscheidungen nur auf Mechanismen zu stützen, da dies das Verantwortungsgefühl der Beteiligten verringert. Entsprechend stellen die Prioritäten nur die Basis für die Releaseplanung dar. Die letztendliche Entscheidung welche Anforderungen im nächsten Release realisiert werden sollen verbleibt bei den Stakeholdern. Dies ist insbesondere auch deshalb erforderlich, da zusätzlich darauf geachtet werden muss, dass ein Release immer eine sinnvolle Menge von Anforderungen enthalten muss, so dass das daraus resultierende System einsetzbar ist.

Die Ergebnisse dieser Aktivität werden im Teilprodukt „Priorität der Anforderungen“ festgehalten.

Die Priorität einer Anforderung kann sich im Verlaufe des Projekts ändern, insbesondere wenn das Verständnis für die Anforderungen steigt. Die Prioritäten sollten daher im Laufe des Projektes aktualisiert werden.

Risiko der Anforderungen abschätzen

Inputs	Outputs
Liste der Anforderungen	Aktualisierte Liste der Anforderungen
Aufwandsabschätzung der Anforderungen	
Risikoabschätzung der Anforderungen	Risikoabschätzung der Anforderungen
Traceability Links	
Änderungshistorie	
Bericht der Anforderungvalidierung	
Iterationsreviewbericht	

Aufgaben
Impact der Anforderungen abschätzen
Stabilität der Anforderungen abschätzen
Stabilisierbarkeit der Anforderungen abschätzen
Weitere Risiken der Anforderungen abschätzen

Das Risiko der Anforderungen ist ein wichtiger Input für die Planung. Insbesondere die Abschätzung von Stabilität, Stabilisierbarkeit und Impact der Anforderungen sind wichtige Parameter, um zu bestimmen wie detailliert die Anforderungen analysiert werden sollen.

Eine der Aufgaben in der Risikoanalyse ist es den Impact von Anforderungen abzuschätzen. Arnold (1993) gibt einen Überblick über die gängigen Ansätze zur Impactanalyse. Unser Ziel ist jedoch ein anderes als das der üblichen Impactanalysemethoden. Wir brauchen eine Abschätzung des Impacts um zu entscheiden wie genau Anforderungen analysiert werden sollen. Daher müssen wir den Impact noch vor der Implementierung der betroffenen Teile des Systems abschätzen. Üblicherweise verwendet man Impactanalyse, um auftretende Change Requests zu bewerten; d.h. die Änderungskosten am bestehenden System müssen abgeschätzt werden. Wir hingegen benötigen nicht die Abschätzung der aktuellen Kosten, sondern der potenziellen zukünftigen Kosten wie dies im Risikomanagement der Fall ist (Higuera und Haimes 1996; Jones 1994). Unser Vorgehen bezüglich der Impactanalyse entspricht in seinen Grundzügen dem im REPEAT Ansatz (Regnell et al. 1998) von Telelogic.

Der Impact repräsentiert in unserem Ansatz wie bei REPEAT die Anzahl der von einer Anforderung betroffenen Komponenten. Der Impactwert stellt dabei nur eine grobe Kategorisierung in fünf Stufen dar. Die verschiedenen Stufen werden in Tabelle 4-2 beschrieben.

1	Nur eine Komponente ist betroffen
2	Einige wenige Komponenten sind betroffen
3	Weniger als die Hälfte der Komponenten sind betroffen
4	Mehr als die Hälfte der Komponenten sind betroffen
5	Fast alle Komponenten sind betroffen

Tabelle 4-2: Kategorisierung des Impacts

Bei der Ermittlung des Impacts einer Anforderung sollten ihre Geschwister in der Anforderungshierarchie ebenfalls betrachtet werden. Diese haben oft einen ähnlichen Impact. Der Impactwert einer Anforderung fasst immer den Impact ihrer Kinder in der Anforderungshierarchie zusammen. Daher sind auch die Impactwerte von Eltern und Kindern der Anforderungen in die Impactabschätzung einzubeziehen. Zusätzlicher Input kann aus der Aufwandsabschätzung und durch Brainstorming, Checklisten und Worst-Case-Szenarien gewonnen werden (Strens und Sugden 1996). Unter Zuhilfenahme der gesammelten Informationen schätzen die Stakeholder intuitiv den Impact jeder Anforderung ab. Sind Experten für bestimmte Anforderungsklassen verfügbar, so nehmen diese die Abschätzungen für die jeweiligen Anforderungen vor. Die Impactabschätzung kann später noch angepasst werden. Zum einen wird eine Anpassung vorgenommen wenn sich das Verständnis der Anforderung wesentlich verbessert hat. Außerdem werden Änderungshistorie sowie der Bericht der Anforderungvalidierung herangezogen um gesammeltes Feedback aus der Entwicklung in die Impactabschätzung einzubeziehen. Eine weitere einfache Möglichkeit Rückkopplung aus der Entwicklung zu bekommen ist die Einbeziehung der Entwickler in die fortlaufende Anpassung der Impactabschätzung. Falls sich eine Anforderung geändert hat, so kann dabei die tatsächliche Zahl der betroffenen Komponenten gemessen werden. Mittels dieser Informationen kann die Impactabschätzung präzisiert werden. Wenn sich der Impact einer oder mehrerer Geschwister, Eltern oder Kinder der Anforderung in der Anforderungshierarchie ändert, so muss unter Umständen auch die Impactabschätzung der Anforderung selbst angepasst werden. Für bestimmte Anforderungen werden im Laufe des Projekts Traceability-Informationen gesammelt. Diese können ebenfalls genutzt werden um die Impactabschätzung zu präzisieren. Abhängig vom gewählten Entwurf kann sich der Impactwert von Anforderungen ebenfalls ändern. Sobald der Entwurf steht, bzw. wenn er geändert wird, müssen unter Umständen die Impactwerte einiger Anforderungen angepasst werden. Verlässlichere Werte für den Impact von Anforderungen würde man bekommen wenn man das komplette Beziehungsgeflecht zwischen Anforderungen und anderen Entwicklungsartefakten bestimmen würde. Dies stellt jedoch einen bedeutenden Aufwand dar, den wir aufgrund des Anstrebens einer geringen Projektlaufzeit meiden. In TTM-Projekten hat es sich in der Praxis bewährt nur ein grobes Beziehungsgeflecht über die Gruppierung von Anforderungen zu erstellen (Karlsson et al. 2002).

Wir schätzen für jede Anforderung ihre Stabilität ab. Es genügt dabei die Stabilität durch grobe Kategorien wie zum Beispiel „sehr gering“, „gering“, „mittel“, „hoch“ und „sehr hoch“ auszudrücken. Für instabile Anforderungen dokumentieren wir zusätzlich welches die Hauptfaktoren für ihre Instabilität sind. Ein Beispiel hierfür wäre „mangelndes Verständnis der Anforderung“. Bei der Stabilitätsabschätzung einer Anforderung betrachten wir wiederum die Stabilität ihrer Geschwister in der Anforderungshierarchie. Diese haben oft eine ähnliche Stabilität. Weitere Hinweise werden entsprechend der Empfehlung von Strens und Sugden (1996) durch Worst-Case-Szenarien, Brainstorming und Checklisten gesammelt. Unter Einbeziehungen der gesammelten Informationen schätzen die Stakeholder zusammen mit etwaigen Experten für die Abschätzung der

Stabilität von Anforderungen, wie beispielsweise erfahrene Entwickler, die Stabilität von Anforderungen ab. Falls Experten für bestimmte Anforderungsklassen verfügbar sind, so nehmen diese für diese die entscheidende Rolle bei der Stabilitätsabschätzung ein. Es muss beachtet werden, dass die Quelle einer Anforderung nicht immer am besten geeignet ist, ihre Stabilität abzuschätzen. Ein wankelmütiger Kunde beispielsweise wird sich oft selbst seine Unentschlossenheit nicht eingestehen. Die anfängliche Abschätzung der Stabilität kann später angepasst werden. Feedback aus der Entwicklung kann ein nützlicher Input für die Stabilitätsanalyse sein (Wieggers 2005). Diesbezüglich nützliche Informationen können zum einen der Änderungshistorie entnommen werden (Berkovich et al. 2009) und zum anderem dem Bericht der Anforderungvalidierung. Ein probates Mittel um Erkenntnisse aus der Entwicklung bzgl. der Stabilität von Anforderungen berücksichtigen zu können ist zudem die Einbeziehung der Entwickler in die Stabilitätsanalyse. Eine Korrektur der Stabilitätsabschätzung wird zum einen vorgenommen, wenn sich das Verständnis für eine Anforderung wesentlich verbessert. Dies kann zum Beispiel der Fall sein wenn sie einer ausführlichen Analysen unterzogen wurde. Wenn sich eine Anforderung ändert überprüfen wir ob der Grund für die Änderung der von uns erwartete war. Ist dem der Fall, so wird die Stabilitätsabschätzung normalerweise gleich bleiben. Stößt man so jedoch auf einen bisher nicht beachteten Grund für Änderungen, kann eine Anpassung der Abschätzung notwendig sein. Trifft ein anfangs angenommener Auslöser für Änderungen im Laufe des Projektes lange Zeit nicht ein, so muss die Abschätzung eventuell entsprechend angepasst werden. Die Stabilitätsabschätzung einer Anforderung sollte auch überprüft werden wenn sich die Stabilitätswerte ihrer Geschwister in der Anforderungshierarchie ändern.

Für instabile Anforderungen führen wir eine Abschätzung ihrer Stabilisierbarkeit durch. Dabei wird abgeschätzt, ob durch eine entsprechende Analyse die Änderungswahrscheinlichkeit der Anforderungen gesenkt werden kann. Es erfolgt lediglich eine Einschätzung entlang grober Kategorien der Stabilisierbarkeit wie zum Beispiel „sehr gering“, „gering“, „mittel“, „hoch“ und „sehr hoch“. Für die Abschätzung der Stabilisierbarkeit von instabilen Anforderungen sind die Ursachen für deren Instabilität zu betrachten. Bestimmten Änderungsgründen kann gut entgegengewirkt werden während andere nur schlecht oder gar nicht beeinflussbar sind. In der Regel lassen sich insbesondere schlecht verstandene Anforderungen stabilisieren. In bestimmten Fällen ist eine Anforderung zwar instabil und nicht stabilisierbar, es kann aber abgeschätzt werden, zu welchem Zeitpunkt die Anforderung stabilisierbar wird. Beispiele hierfür wären, dass im Laufe des Projekts sich ein vom System einzuhaltendes Protokoll ändern wird und die Details dieser Änderung noch nicht im Vornherein bekannt sind, oder dass ein bestimmter Stakeholder erst zu einem späteren Zeitpunkt in das Projekt einbezogen werden kann. Für diese Fälle ist es sinnvoll die betreffenden Anforderungen entsprechend zu markieren, so dass zu dem Zeitpunkt, an dem sie stabilisierbar werden, direkt eine entsprechende Analyse und Neubewertung der Anforderungen durchgeführt werden kann. Weiterhin wird für die Stabilisierbarkeitsabschätzung die Stabilisierbarkeit von Geschwisteranforderungen in der Anforderungshierarchie mitbetrachtet, da diese oft ähnlich ist. Allgemeine Aussagen, welchen Änderungsursachen entgegengewirkt werden kann, bzw. welche Arten von Anforderungen gut stabilisierbar sind, können nicht getroffen werden, da sich je nach Projektsituation und entwickelnder Organisation große Unterschiede ergeben. Die Einschätzung der Stabilisierbarkeit wird von den Stakeholdern vorgenommen, falls möglich unterstützt durch Personen, die mit der Einschätzung der Stabilisierbarkeit von Anforderungen Erfahrung haben, wie erfahrene Projektleiter und Entwickler. Im Projektverlauf kann anhand Erfahrungen aus der Entwicklung oder aus einem besseren Verständnis der jeweiligen Anforderung heraus die Abschätzung der Stabilisierbarkeit aktualisiert werden.

Entwickelt ein Unternehmen oft ähnliche Projekte, so kann es sich zunutze machen, dass in diesen bestimmte Anforderungsklassen oft ähnliche Änderungsrisiken aufweisen. Es kann beispielsweise sein, dass in einer Reihe von Projekten bestimmte Stakeholder eine tragende Rolle spielen, die sich nur sehr schwer über die Gestaltung der GUI einig werden. Daher können Anforderungen, die die GUI betreffen, tendenziell als instabil angesehen werden. Ein anderes Beispiel wäre, dass sich in dem Unternehmen ein bestimmtes Standardvorgehen hinsichtlich der Security-Anforderungen durchgesetzt hat. Die entsprechenden Anforderungen gelten daher meist als stabil. Weiterhin können beispielsweise Performanzanforderungen oder allgemein spezifische Qualitätsanforderungen stets einen hohen Impact aufweisen. Denkbar wäre auch, dass sich bestimmte von außerhalb des Unternehmens festgelegte Kommunikationsprotokolle beständig und unvorhersehbar ändern, so dass diesbezügliche Anforderungen als instabil und nicht stabilisierbar gelten. Entwickelt ein Unternehmen oft ähnliche Projekte, ist es ratsam das Template für die Anforderungsermittlung in Bezug auf Änderungsrisiken zu erweitern, und für einzelne Anforderungsklassen festzuhalten wie ihr Impact, ihre Stabilität und ihre Stabilisierbarkeit in der Regel gestaltet sind. Wir nehmen jedoch davon Abstand allgemeingültige Aussagen zum Änderungsrisiko bestimmter Anforderungsklassen zu treffen, da abhängig von Unternehmen, Domäne und Projekt sich diesbezüglich starke Unterschiede ergeben können.

Laut den Aussagen der Experten der FAUSER AG und von PSIPENTA in den beiden Fallstudien aus Kapitel 3 können sowohl Stabilität als auch Stabilisierbarkeit von Anforderungen mit großer Sicherheit abgeschätzt werden. Dabei ist jedoch zu beachten, dass beide Organisationen Projekte durchführen bei denen sie viel Erfahrung mit der Entwicklung ähnlicher Systeme sowie mit der Domäne allgemein haben. Sind die Kenntnisse über die Domäne und die Erfahrung mit der Entwicklung ähnlicher Systeme gering oder stehen kaum erfahrene Entwickler und Projektleiter zur Verfügung, so ist anzunehmen, dass man zumindest nicht mit der von der FAUSER AG und PSIPENTA angegebenen großen Sicherheit Stabilität und Stabilisierbarkeit von Anforderungen abschätzen kann. Entsprechend der Fähigkeit eines Unternehmens Stabilität und Stabilisierbarkeit von Anforderungen abzuschätzen sollten die Kategorien, in denen diese Abschätzungen vorgenommen werden angepasst werden. So wäre es beispielsweise für den Fall einer schlechteren Abschätzbarkeit vorteilhaft Stabilität und Stabilisierbarkeit nur entlang dreier Kategorien, zum Beispiel „gering“, „mittel“ und „hoch“, zu bewerten.

Wir wollen kurz anhand folgender Beispielanforderungen aus Kapitel 4.7.1 klar machen wie die Analyse des Änderungsrisikos aussehen kann:

- 1.1 *Every painting always belongs to only one "Theme" for example such as modern art or nature, etc.*
- 1.2 *The system has a drop-down list of themes already created for an administrator to choose from while adding a new painting information.*
- 1.3 *The graphical user interface (GUI) should allow the information of only one painting to be entered at a time.*
- 1.4 *The "Painting Information" for a painting includes:*
 - 1.4.1 *The name or title of the painting. Every painting must have a unique name/title by which it can be identified.*

- 1.4.2 *The image/picture of the painting. The image for a painting can be selected only from folders in the local file system through the GUI. Only jpg and gif files are considered as image files. Image size should be 300 X 300 pixels on the screen.*
 - 1.4.3 *The name of the artist and this should be linked to the “Artist Information” for that particular artist.*
 - 1.4.4 *The year in which the painting was known to have been created.*
 - 1.4.5 *The theme under which the painting can be classified. This should provide a link to “Theme Information” for that particular theme.*
 - 1.4.6 *The name and location of the gallery where the piece of art is currently located. The name of the gallery should provide a link to “Gallery Information” for that particular art gallery. The location (city and country where the gallery is located) should provide a link to “Country Information” for that particular location.*
 - 1.4.7 *A short description of the painting. This description can include references to other paintings, artists, art galleries and locations in which case, they should all point to the corresponding information section.*
 - 1.4.8 *The GUI should provide suitable fields and options to allow all the above information/description to be added to the particular painting information. All links should be accordingly displayed in the GUI.*
- 4 *Performance – It is highly essential that the VirtualArtViewer application runs efficiently. Any “search” should not take more than 5 seconds.*

Wir vermuten, dass Änderungen an den Anforderungen 1.1, 1.2 und 1.3 jeweils nur eine Komponente betreffen. Der Impactwert für jede dieser Anforderungen ist also 1. Abhängig vom Entwurf könnten auch mehrere Komponenten betroffen sein, jedoch höchstens einige wenige, was einem Impactwert von 2 entsprechen würde. Sobald der Entwurf steht muss daher der Impactwert eventuell angepasst werden. Gleiches gilt für die Anforderungen 1.4.1-1.4.7. Letztere sind sich alle sehr ähnlich, was die Abschätzung ihres Impacts besonders leicht macht sobald man diese für eine getroffen hat. Auch die Anforderung 1.4.8 betrifft nur eine Komponente. Für die Abschätzung des Impactwerts von Anforderung 1.4 können ihre Kinder herangezogen werden. Wir gehen davon aus, dass Änderungen an den Anforderungen 1.4.1-1.4.7 alle dieselbe Komponente betreffen, Änderungen an 1.4.8 jedoch eine andere. Daher betrifft die Anforderung 1.4 einige wenige Komponenten, erhält also den Impactwert 2. Wir nehmen in unserem Beispiel an, dass die Stakeholder relativ sicher sind, welche Informationen zu einem Bild angezeigt werden sollen. Dementsprechend wird die Stabilität der Anforderungen 1.4.1-1.4.8 als „hoch“ festgehalten. Gleiches gilt für die Anforderungen 1.1-1.3. Unsicherheit besteht jedoch hinsichtlich der genauen Performanzanforderungen, da die Stakeholder sich uneinig darüber sind, ob fünf Sekunden noch eine zumutbare Wartezeit für den Benutzer darstellen. Anforderung 4 ordnen wir daher den Stabilitätsgrad „gering“ zu. Als Grund für die Instabilität dieser Anforderung führen wir „mangelndes Verständnis“ auf. Wir nehmen an, dass die Unsicherheit hinsichtlich der Performanzanforderung durch Befragung potenzieller Benutzer beseitigt werden kann. Diese Anforderung gilt also als stabilisierbar.

Neben dem Änderungsrisiko müssen auch andere Risiken abgeschätzt werden. Dies wird in der Regel rein intuitiv durch Stakeholder mit entsprechenden Kenntnissen und Erfahrung vorgenommen.

Die Ergebnisse dieser Aktivität werden im Teilprodukt „Risikoabschätzung der Anforderungen“ festgehalten.

Anforderungen verhandeln

Inputs	Outputs
Liste der Anforderungen	Aktualisierte Liste der Anforderungen
Wichtigkeit der Anforderungen	
Aufwandsabschätzung der Anforderungen	
Risikoabschätzung der Anforderungen	
Konflikte zwischen Anforderungen	Aktualisierte Konflikte zwischen Anforderungen

Aufgaben
Über Änderungen an Anforderungen entscheiden
Konflikte zwischen Anforderungen auflösen
Konflikte mit den Budget- und Zeitvorgaben auflösen
Anforderungen mit zu hohem Risiko ändern

In der Anforderungsverhandlung wird darüber entschieden, ob Änderungswünsche angenommen werden sollen. Konflikte zwischen Anforderungen sowie mit den Budget- und Zeitvorgaben sollen aufgelöst werden. Zudem gilt es Anforderungen, die ein zu hohes Risiko aufweisen, zu ändern oder fallen zu lassen.

Zunächst sollen in dieser Aktivität eingebrachte Änderungswünsche bewertet werden. Ein Change Control Board entscheidet über die Annahme der Änderungen. Es ist wichtig, dass alle von den Änderungen betroffenen Stakeholder an der Bewertung der Änderungen beteiligt werden, damit die Konsequenzen, d.h. sowohl Vor- als auch Nachteile, der Änderungen bekannt werden. Es hat sich in der Praxis bewährt zentrale Stakeholder in das Change Control Board zu integrieren. In KMU werden vielfach Varianten eingesetzt, in denen einzelne Personen, wie beispielsweise der Projektleiter oder ein wichtiger Kunde allein über Änderungen entscheiden. In kleinen Projekten, insbesondere wenn eine geringe Projektlaufzeit angestrebt wird, ist es generell empfehlenswert einfache Abstimmungssysteme (Jarke et al. 1992) oder alternativ auch Ansätze, die auf der Theory-W von Boehm et al. (1994) basieren, zu verwenden. Pohl (1996) führt an, dass auch Methoden wie Quality Function Deployment (Hauser und Clausing 1998) angepasst werden können, um die Verhandlung von Anforderungen zu unterstützen.

Eine weitere Aufgabe in der Anforderungsverhandlung ist es, die in der Anforderungsanalyse oder während der Entwicklung identifizierten Konflikte zwischen Anforderungen aufzulösen. Sind zwei Anforderungen im Konflikt miteinander so können sie nicht beide gleichzeitig realisiert werden. Zudem kann es sein, dass nicht alle Anforderungen innerhalb der Budget- und Zeitvorgaben realisiert werden können oder dass das Risiko bestimmter Anforderungen als zu hoch angesehen wird. Um diesen Problemen zu begegnen werden einzelne Anforderungen abgeändert oder ganz fallen gelassen. Es muss darüber entschieden werden, bei welchen Anforderungen und auf welche Weise dies geschieht. Dabei bietet die Wichtigkeit der betroffenen Anforderungen eine Basis für die Diskussion. Aber auch andere Attribute wie Aufwand und Risiko der Anforderungen sollten beachtet werden.

Die Liste der Anforderungen und der Konflikte werden entsprechend angepasst.

Wurde in der Anforderungsverhandlung beschlossen eine Anforderung fallen zu lassen so geht sie in den Zustand „abgelehnt“ über. Wird sie geändert erhält sie ebenfalls den Zustand „abgelehnt“, es wird aber eine entsprechende neue Anforderung im Zustand „vorgeschlagen“ eingebracht. Ist eine Anforderung in unveränderter Form für die Realisierung vorgesehen erhält sie den Zustand „angenommen“.

4.7.3 Anforderungen spezifizieren

4.7.3.1 Unterschiede zu Wiegers' Ansatz

Während REPMUS wenn möglich nur informelle und semi-formelle Notationen verwendet und bestrebt ist die Spezifikation leicht änderbar, leicht verständlich, kurz und zielgerichtet zu halten fordert Wiegers dies nicht. REPMUS nutzt diese Art von Notationen, da sie in der Regel eine aufwandsärmere Erstellung und Änderung der Spezifikation erlauben und somit in TTM-Projekten vorteilhaft sind (McPhee und Eberlein 2002). Informelle und semi-formelle Notationen sind zudem für die meisten Stakeholder leichter zu verstehen, während formale Notationen für manche oft kaum verständlich sind. Wie in Kapitel 2.2.1 angeführt, ist die falsche Interpretation von Anforderungsdokumenten eine Quelle von Anforderungsänderungen. Dem kann durch eine leichter verständliche Spezifikation entgegengewirkt werden.

Die Spezifikation ist bei Wiegers wesentlich umfassender und detailreicher. REPMUS variiert hingegen den Detaillierungsgrad je nach Änderungsrisiko der jeweiligen Anforderung. Dies spiegelt das Bestreben von REPMUS wider, den RE-Aufwand möglichst effizient einzusetzen.

4.7.3.2 Beschreibung der Aktivität

Inputs	Outputs
Anforderungsspezifikation	Aktualisierte Anforderungsspezifikation
Liste der Anforderungen	
Bericht der Anforderungvalidierung	
Iterationsreviewbericht	
Domänenmodelle	
Analysemodelle	

Aufgaben
Spezifikation erstellen
Im Bericht der Anforderungvalidierung aufgeführte Fehler in der Anforderungsspezifikation korrigieren

Die Anforderungsspezifikation ist der wichtigste Output der RE-Aktivitäten. Sie wird in Design und Implementierung benötigt und stellt die detaillierte Beschreibung der Anforderungen dar. Das Aufschreiben von Anforderungen trägt gleichzeitig dazu bei, diese besser zu verstehen (Sugden und Strens 1996).

REPMUS schreibt nicht vor, auf welche Art und Weise Anforderungen spezifiziert werden. Es gibt jedoch einige Aspekte, die bei der Wahl der Spezifikationsmethode

beachtet werden sollten. Da eine geringe Projektlaufzeit angestrebt wird, sollte der Aufwand für die Erstellung der Spezifikation gering gehalten werden. Da sich die Anforderungen oft ändern und dies auch Änderungen an der Spezifikation zur Folge hat sollte diese leicht änderbar gehalten werden (Strens und Sugden 1995; Sawyer et al. 1999). Aufgrund der intensiven Beteiligung der Stakeholder ist es für den reibungslosen Ablauf förderlich wenn diese mit Leichtigkeit die Spezifikation lesen und verstehen können. Insbesondere in kleineren und mittleren Unternehmen wird die leichte Verständlichkeit der Anforderungen als wichtig erachtet (Karlsson et al. 2002). Um den genannten Aspekten gerecht zu werden, sollten sogenannte leichtgewichtige Notationen verwendet werden. Darunter versteht man gemeinhin Notationen, die eine aufwandsarme Erstellung und Änderung von Dokumenten erlauben. Zudem sind sie auch für Laien leicht verständlich. Viele informelle und semi-formelle Notationen werden diesen Anforderungen gerecht. Ein Beispiel für eine informelle Notation ist die natürliche Sprache. Semi-formelle Notationen sind beispielsweise Use Cases, Szenarios und UML-Sequenzdiagramme. Laut einer Studie von McPhee und Eberlein (2002) werden informelle und semi-formelle Modellierungstechniken als sehr nützliche RE-Techniken für TTM-Projekte angesehen. Gleiches gilt laut der Studie auch für Use Cases (Cockburn 2001) und Szenarien (Rolland et al. 1998). Diese beiden Techniken sind besonders gut geeignet, da sie einfach zu verstehen und einzusetzen sind. Zudem sind sie bereits weit verbreitet (Neill et al. 2003). Natürliche Sprache stellt in kleineren und mittleren Betrieben eine bewährte Methode zur Anforderungsspezifikation dar (Karlsson et al. 2002). Die Spezifikation sollte möglichst kurz gehalten werden um Zeit einzusparen. Außerdem sollte sie stets zielgerichtet angelegt werden (Ambler 2002; Davies 2005). Bei den meisten Anforderungen hat die Spezifikation lediglich den Zweck diese der Entwicklung zu kommunizieren. Es sollte nur das dafür notwendige Minimum an Dokumentation angelegt werden. Dieses kann bestimmt werden, indem man die Stakeholder, darunter im Speziellen die Entwickler, befragt, welchen Grad an Dokumentation sie unbedingt benötigen. Falls Domänen- oder Analysemodelle erstellt wurden, können diese eventuell für die Anforderungsspezifikation weiterverwendet werden. Die Stakeholder entscheiden in welchem Detaillierungsgrad eine Anforderung spezifiziert wird. Anforderungen, für die in der Analyse viel Aufwand betrieben wurde, weil sie als stabilisierbar gelten oder einen hohen Impact aufweisen, werden in der Regel auch detaillierter spezifiziert. Die Aktivität umfasst nicht nur das Erstellen der Spezifikation sondern auch das Aktualisieren der bestehenden Spezifikation, um aufgetretene Änderungen einzubeziehen. Dies beinhaltet auch das Korrigieren von Fehlern, die im Bericht der Anforderungvalidierung aufgeführt sind.

Die Anforderungsspezifikation erfolgt iterativ, da in späteren Iterationen neue Anforderungen hinzukommen oder erst später fertig analysiert werden.

4.7.4 Anforderungen validieren

4.7.4.1 Unterschiede zu Wiegiers' Ansatz

REPMUS weist keine grundsätzlichen Unterschiede zu Wiegiers' Ansatz in der Anforderungvalidierung auf.

4.7.4.2 Beschreibung der Aktivität

Inputs	Outputs
Liste der Anforderungen	
Änderungshistorie	
Anforderungsspezifikation	
Bericht der Anforderungvalidierung	Aktualisierter Bericht der Anforderungvalidierung
Iterationsreviewbericht	Iterationsreviewbericht
Konflikte zwischen Anforderungen	Aktualisierte Konflikte zwischen Anforderungen

Aufgaben
Iterationsreview durchführen
Liste der Anforderungen auf Korrektheit prüfen
Anforderungsspezifikation auf Korrektheit prüfen
Qualität der Anforderungsspezifikation prüfen

Durch die Anforderungvalidierung soll überprüft werden, ob die Spezifikation die von den Stakeholdern aufgestellten Anforderungen richtig wiedergibt. Zudem soll gewährleistet werden, dass die spezifizierten Anforderungen den Qualitätsmaßstäben genügen.

Es wird in der Praxis als nützlich betrachtet bei der Anforderungvalidierung multiple Stakeholder zu beteiligen (Hofmann und Lehner 2001; Sawyer et al. 1999). REPMUS sieht daher vor, dass die Stakeholder gemeinsam für alle identifizierten Anforderungen überprüfen, ob sie korrekt in der Spezifikation wiedergegeben werden. Darüber hinaus kontrollieren sie, ob diese den vereinbarten Qualitätsmaßstäben entspricht. Dabei sollte eine Überprüfung auf Korrektheit, Verständlichkeit, Modifizierbarkeit und Eindeutigkeit erfolgen. Darüber hinaus wird auch überprüft, ob die in der Liste der Anforderungen aufgeführten Anforderungen den Wünschen der Stakeholder entsprechen.

Sawyer et al. (1999) empfehlen, diese Tätigkeit durch den Einsatz von Checklisten zu unterstützen. Hofmann und Lehner (2001) führen als weitere geeignete Techniken Walkthroughs und Inspektionen an (siehe hierzu Fagan 1986 sowie Freeman und Weinberg 1990). Die Studie von McPhee und Eberlein (2002) deutet darauf hin, dass die Anforderungvalidierung, insbesondere auch der Einsatz von Checklisten, eine nützliche Technik für TTM-Projekte darstellt. Jones (1996b) behauptet, dass Inspektionen die Änderungsrate der Anforderungen um bis zu 30 Prozent verringern können. Auch Sawyer et al. (1999) empfehlen den Einsatz von Inspektionen für marktgetriebene und einer hohen Änderungsrate ausgesetzte Softwareentwicklungsprojekte.

Die Ergebnisse der Anforderungvalidierung werden im Bericht der Anforderungvalidierung zusammengefasst.

In dieser Aktivität können sowohl Fehler in der Anforderungsspezifikation als auch in der Liste der Anforderungen gefunden werden. Je nachdem springt man entweder zur Anforderungsermittlung oder Anforderungsspezifikation zurück, um die gefundenen Fehler zu beheben. Müssen Anforderungen neu bewertet werden geht man erneut zur Anforderungsanalyse über. Da sowohl die Liste der Anforderungen als auch die Anforderungsspezifikation beständig erweitert werden, muss die Anforderungvalidierung

iterativ durchgeführt werden. Bei jeder Ausführung versucht man alle bisherigen Ergebnisse zu überprüfen.

Anforderungen werden zusätzlich während der Reviews am Ende von Entwicklungsiterationen validiert. Wie dabei vorgegangen wird beschreiben wir in der Unteraktivität „Iterationsreview durchführen“.

Iterationsreview durchführen

Inputs	Outputs
Liste der Anforderungen	
Änderungshistorie	
	Bericht der Anforderungvalidierung
	Iterationsreviewbericht
Konflikte zwischen Anforderungen	Aktualisierte Konflikte zwischen Anforderungen

Aufgaben
Prüfen, ob die Ergebnisse der Iteration den Vorstellungen der Stakeholder entsprechen
Abweichungen bei den Kosten kommunizieren
Aufgetretene Konflikte zwischen Anforderungen kommunizieren
Missverständnisse zwischen Stakeholdern und Entwicklern klären

Iterationsreviews werden durchgeführt um Feedback aus der Entwicklung zu sammeln. Dieses kann zum einen genutzt werden, um Anforderungen besser zu verstehen. Zum anderen unterstützt es die Anforderungvalidierung. Kosten- und Risikoabschätzungen können anhand des Feedbacks präzisiert werden. Konflikte und Fehler in den Anforderungen, die in der Entwicklung entdeckt wurden, können den RE-Verantwortlichen bekannt gemacht werden. Zudem ist es zu diesem Zeitpunkt möglich Unklarheiten bezüglich der Anforderungen zu klären, wenn Anforderungen von der Entwicklung nicht oder falsch verstanden wurden.

Die Entwickler präsentieren am Ende jeder Iteration deren Ergebnisse. Die Stakeholder überprüfen, ob die Ergebnisse ihren Vorstellungen entsprechen. Oft können Anforderungen besser verstanden werden, wenn sie zumindest teilweise realisiert sind (Sugden und Strens 1996). Besonders bei komplizierten Anforderungen kann ihre Realisierung oder Teilrealisierung wesentlich zu ihrem Verständnis beitragen. Des Weiteren können die Stakeholder anhand der Ergebnisse der Iteration feststellen, ob das Spezifizierte, auf dem die Realisierung basiert, wirklich ihren Wünschen entspricht und somit die Korrektheit der Spezifikation überprüfen. Werden während der Entwicklung Konflikte oder Fehler in den Anforderungen entdeckt, so können die Stakeholder spätestens zu diesem Zeitpunkt darüber informiert werden. Alternativ kann dies schon während der Iteration geschehen, und somit der Konflikt beziehungsweise der Fehler sofort aufgelöst werden. Stellt sich heraus, dass Anforderungen von der Entwicklung falsch verstanden wurden, diese eventuell auch falsch spezifiziert wurden, so können spätestens an dieser Stelle Unklarheiten beseitigt werden. Besser ist jedoch eine Beseitigung der Unklarheiten noch innerhalb der Iteration. Darüber hinaus können einige der Planungsparameter korrigiert werden. Im Laufe der Entwicklung kann sich unter Umständen abzeichnen, dass die Kostenschätzungen falsch sind. Dies wird spätestens an dieser Stelle den Stakeholdern und insbesondere dem Management kommuniziert, so dass

entsprechende Korrekturen gemacht werden können. Die Änderungshistorie der Iteration wird ebenfalls präsentiert. Die tatsächlichen Kosten der durchgeführten Änderungen können für eine Korrektur der Impactabschätzungen genutzt werden. Anhand der Art und Häufigkeit der Änderungen wird die Stabilitätsabschätzung angepasst.

Es gibt eine Reihe von Techniken, mittels derer das Iterationsreview unterstützt werden kann. Dazu gehören u. a. Walkthroughs, Simulationen und Akzeptanztests.

Die Ergebnisse dieser Aktivität werden im Iterationsreviewbericht sowie dem Produkt „Konflikte zwischen Anforderungen“ festgehalten.

4.7.5 Anforderungen managen

4.7.5.1 Unterschiede zu Wiegers' Ansatz

In Wiegers' Ansatz ist der Änderungsprozess Teil des Anforderungsmanagement. Bei REPMUS ist dieser jedoch, wie in Kapitel 4.6 beschrieben, in den grundlegenden RE-Zyklus, bestehend aus Anforderungsermittlung, -analyse, -spezifikation und -validierung integriert.

Während bei Wiegers Traceability-Informationen für alle Anforderungen gesammelt werden, geschieht dies nach REPMUS lediglich bei besonders instabilen Anforderungen. Die Verfolgung sämtlicher Anforderungen wird in der Praxis in TTM-Projekten aufgrund des hohen damit verbundenen Aufwands nicht angestrebt (Ambler 2002; McPhee und Eberlein 2002). REPMUS sammelt daher nur für besonders instabile Anforderungen Traceability-Informationen, da bei diesen der Kosten-Nutzen-Effekt zwischen eingesparten Änderungskosten und für die Anforderungsverfolgung eingesetzten Aufwand am günstigsten ist.

4.7.5.2 Beschreibung der Aktivität

Inputs	Outputs
Liste der Anforderungen	Aktualisierte Liste der Anforderungen
Traceability Links	Traceability Links
Risikoabschätzung der Anforderungen	
Änderungshistorie	Änderungshistorie

Aufgaben
Zu verfolgende Anforderungen auswählen
Traceability-Informationen sammeln
Änderungen protokollieren

Können Anforderungen zu den Entwicklungsartefakten, in denen sie sich niederschlagen, verfolgt werden, so reduziert sich der Aufwand bei der Durchführung von Änderungen an ihnen. Daher sollen in dieser Aktivität Traceability-Informationen für besonders instabile Anforderungen gesammelt werden. Weiterhin sollen Informationen über Änderungen in der laufenden Entwicklung gesammelt werden. Diese werden für die Verfeinerung der Risikoabschätzung benötigt.

Traceability-Informationen für alle Anforderungen zu erfassen stellt einen großen Aufwand dar (Ambler 2002). In der Studie von McPhee und Eberlein (2002) wurde die

Anforderungsverfolgung als weniger wichtig für TTM-Projekte eingestuft. Wir sammeln Traceability-Informationen daher lediglich für besonders instabile Anforderungen. Von diesen erwarten wir die meisten Änderungen. Die damit verbundenen Änderungskosten können reduziert werden, wenn Traceability-Informationen zur Verfügung stehen. Bei instabilen Anforderungen zahlt sich der Aufwand für die Verfolgung der Anforderungen am ehesten aus, da diese hohe Änderungskosten erzeugen. Auch die Kosten für die Änderung von Anforderungen mit hohem Impact sind hoch. Diese zu verfolgen stellt aber gerade wegen ihrem Einfluss auf weite Teile des Systems einen hohen Aufwand dar. Für eine Anforderung, die einen hohen Impact hat, die jedoch äußerst stabil ist, würde sich die Verfolgung nicht auszahlen. Den optimalen Effekt würde man erzielen, wenn Stabilität und Impact gegen den Aufwand für die Verfolgung aufgewogen würden. Der Einfachheit halber, und da dies wiederum zusätzlichen Aufwand bedeuten würde, beschränken wir uns auf besonders instabile Anforderungen. Die Auswahl der zu verfolgenden Anforderungen wird unter Zuhilfenahme der Stabilitätsabschätzung der Anforderungen vorgenommen. Für die ausgewählten Anforderungen wird dokumentiert auf welche Artefakte beziehungsweise Teile von Artefakten sie sich auswirken.

Im Verlauf des Projektes kommt es zu einer Vielzahl an Änderungen der Anforderungen. Wir protokollieren für jede Anforderung die Anzahl und die Art der Änderungen, die sie betreffen. Zudem wird der Zeitpunkt, an dem die Änderungen auftauchen, festgehalten. Darüber hinaus halten wir bei der Durchführung der Änderung fest, welche tatsächlichen Kosten dabei entstehen.

Die Ergebnisse dieser Aktivität werden in den Teilprodukten „Traceability Links“ und „Änderungshistorie“ festgehalten.

Die Aktivität wird kontinuierlich parallel zur Entwicklung durchgeführt.

4.8 Produkte

In diesem Kapitel beschreiben wir alle Produkte, die in unserem Ansatz erzeugt und verwendet werden. Produkte können Teilprodukte aufweisen. Wir treffen keine Aussage darüber, ob jedes Produkt, beziehungsweise Teilprodukt einem eigenen Dokument entspricht. In manchen Fällen, wie zum Beispiel bei der Liste der Anforderungen und einigen ihrer Teilprodukte ist es unserer Meinung nach sinnvoll diese in einem einzelnen Dokument zusammenzufassen und in anderen wie zum Beispiel den Bericht der Anforderungvalidierung und dem Iterationsreviewbericht macht es Sinn, getrennte Dokumente zu erstellen.

4.8.1 Liste der Anforderungen

Dieses Produkt listet die bereits identifizierten Anforderungen auf. Dazu zählen auch Änderungswünsche. Neben dem Namen der Anforderung werden noch eine Reihe weiterer Informationen festgehalten. Wichtigkeit, Kosten, Priorität, Risiko, Traceability Links und Änderungshistorie der einzelnen Anforderungen werden in den zugehörigen Teilprodukten aufgeführt. Darüber hinaus wird die Quelle der Anforderungen dokumentiert. Dies ist in der Regel ein Stakeholder, kann aber auch ein Dokument, wie zum Beispiel die Beschreibung eines Standards, sein. Schließlich wird noch der Zustand der Anforderungen festgehalten. Der Zustand einer Anforderung kann wie in Kapitel 4.4 dargestellt „vorgeschlagen“, „angenommen“, „abgelehnt“, „realisiert“, „gelöscht“ oder „verifiziert“ sein.

Es gibt Anforderungen verschiedenen Abstraktionsgrads, wobei abstraktere Anforderungen durch konkretere Anforderungen detailliert werden. Dabei können beliebig viele Abstraktionsebenen auftauchen. Es ergibt sich daher eine Hierarchie von Anforderungen. In der Liste der Anforderungen werden die Anforderungen gemäß dieser Hierarchie geordnet.

Wichtigkeit der Anforderungen

In diesem Teilprodukt wird die Wichtigkeit der Anforderungen dokumentiert.

Aufwandsabschätzung der Anforderungen

Hier werden die Kostenschätzungen zu den einzelnen Anforderungen aufgeführt. Für jede Anforderung wird in einem absoluten Wert, zum Beispiel in Personentagen oder Function Points, angegeben welcher Aufwand für die Realisierung der Anforderung erwartet wird.

Prioritäten der Anforderungen

In diesem Teilprodukt sind die Prioritäten der einzelnen Anforderungen aufgelistet.

Risikoabschätzung der Anforderungen

Dieses Teilprodukt führt Risiken, die mit den einzelnen Anforderungen verbunden sind, auf. Neben der Auflistung der Risiken wird ein Risikowert angegeben, der wiederum eine grobe Kategorisierung darstellt, zum Beispiel in „sehr gering“, „gering“, „mittel“, „hoch“ und „sehr hoch“. Besondere Beachtung findet das Änderungsrisiko der Anforderungen. Zu jeder Anforderung wird ihr Impact aufgeführt. Dieser drückt aus, wie viele Komponenten von einer Änderung der Anforderung betroffen sein können. Es wird jedoch nicht die genaue Anzahl festgehalten sondern nur die Größenordnung der Anzahl betroffener Komponenten (Tabelle 4-2). Es werden Stabilitätsabschätzungen zu den einzelnen Anforderungen aufgeführt. Die Stabilität wird anhand grober Kategorien dargestellt, zum Beispiel „sehr gering“, „gering“, „mittel“, „hoch“ und „sehr hoch“. Der Wert drückt aus, für wie wahrscheinlich eine Änderung der Anforderung gehalten wird. Falls eine Anforderung als instabil angesehen wird, so wird zusätzlich aufgeführt aus welchem Grund die Anforderung sich vermutlich ändern wird, zum Beispiel „mangelndes Verständnis der Anforderung“. Schließlich wird noch für instabile Anforderungen ihre Stabilisierbarkeit festgehalten, also inwieweit sie durch eine entsprechende Analyse stabil gemacht werden können. Die Stabilisierbarkeit wird ebenfalls in groben Kategorien dargestellt, zum Beispiel „sehr gering“, „gering“, „mittel“, „hoch“ und „sehr hoch“. Für Anforderungen, bei denen abgeschätzt werden kann, dass sie zu einem bestimmten Zeitpunkt ihre Stabilisierbarkeit ändern werden, wird dieser Zeitpunkt entsprechend vermerkt.

Traceability Links

Für instabile Anforderungen werden in diesem Teilprodukt Traceability-Informationen festgehalten. Es wird dabei dokumentiert welche Entwicklungsprodukte oder Teile von Produkten von den Anforderungen abhängen.

Änderungshistorie

In diesem Teilprodukt werden Informationen über Änderungen der Anforderungen gesammelt. Für jede Anforderung wird dokumentiert welche Änderungen daran durchgeführt wurden. Dabei werden die Art der Änderung sowie der Zeitpunkt, zu dem die

Änderung erfolgte, festgehalten. Die Kosten, die bei der Durchführung der Änderung angefallen sind, werden ebenfalls protokolliert.

4.8.2 Liste der Stakeholder

In diesem Produkt werden alle Stakeholder, die an dem Projekt beteiligt werden, mit ihrem Namen und der Stakeholderklasse, die sie repräsentieren, aufgelistet. Eine Stakeholderklasse kann dabei durch mehrere Stakeholder vertreten werden.

4.8.3 Liste der Stakeholderklassen

Dieses Produkt listet alle Stakeholderklassen, die in dem Projekt berücksichtigt werden, auf. Als Stakeholderklasse sehen wir alle Parteien an, die von der Entwicklung, vom Einsatz und vom Betrieb des zu entwickelnden Systems betroffen sind. Dazu zählen sowohl Benutzer, als auch andere Systeme, Entwickler, Qualitätsmanager, Prozessverantwortliche, Marketing, Experten für Standards und Gesetzestexte, u. v. m. Es wird zu jeder Stakeholderklasse ihr Name sowie bei Bedarf eine kurze Beschreibungen aufgeführt.

4.8.4 Template für die Liste der Stakeholderklassen

In diesem Produkt findet sich eine generische Liste von Stakeholderklassen samt Kurzbeschreibungen. Diese stellt die Ausgangsbasis für die Identifikation von Stakeholderklassen in einem konkreten Projekt dar. Wir haben ein generisches Template basierend auf den Viewpoint-Klassen von Kotonya und Sommerville (1996) entwickelt. Dieses ist in Tabelle 4-3 dargestellt. Es ist empfehlenswert, das Template unternehmensintern weiterzuentwickeln. Idealerweise ist pro Domäne beziehungsweise Projekttyp ein eigenes Template vorhanden.

Stakeholderklasse	Kurzbeschreibung
Direkt	Stakeholder, die unmittelbar mit dem System interagieren
Benutzer	Personen, die mit dem zu entwickelnden System interagieren
Endbenutzer	Die primären Benutzer des Systems
Administratoren	Personen, die das System verwalten
System	Systeme, die mit dem zu entwickelnden System interagieren
Indirekt	Stakeholder, die ein indirektes Interesse an dem System haben
Engineering	Personen, die wesentliche Aufgaben am Entwicklungsprozess wahrnehmen
Entwickler	Personen, die das System entwickeln
Tester	Personen, die das System testen
Technologieexperten	Personen, die Erfahrung mit der Entwicklung von ähnlichen Systemen haben und Risiken bei den verwendeten Technologien gut abschätzen können
Wartung	Personen, die das System warten
Qualitätsmanagement	Personen, die die Einhaltung der Qualitätsmaßstäbe kontrollieren
Prozessverantwortliche	Personen, die den Entwicklungsprozess vorgeben und dessen Einhaltung kontrollieren
Dokumentation	Personen, die die Dokumentation des Systems erstellen
Regelnd	Rechtliche und auf Regelungen bezogene Aspekte
Standards	Richtlinien, die vom System einzuhalten sind
Rechtsabteilung	Personen, die mit Gesetzen, die bei der Entwicklung zu beachten sind vertraut sind
Organisation	Unternehmerische Aspekte
Auftraggeber, Kunden	Personen, die das System erwerben werden
Management des Auftragnehmers	Vertreter der Auftragnehmerorganisation
Marketing/Vertrieb	Personen, die ein Interesse an der Vermarktbarkeit des Systems haben
Akquisiteur	Personen, die für die Pflege und den Aufbau von Kundenbeziehungen verantwortlich sind
Support	Personen, die für den Kundensupport verantwortlich sind
Umgebung	Aspekte, die die Zielumgebung des Systems betreffen
Physikalische Einflüsse	Personen, die die physikalischen Einschränkungen des Systems in der Zielumgebung verstehen
Indirekt vom System Betroffene	Personen, die vom Einsatz des Systems betroffen sind, ohne dass sie mit ihm interagieren

Tabelle 4-3: Template für die Liste der Stakeholderklassen

4.8.5 Template für die Anforderungsermittlung

In diesem Template findet sich eine generische Klassifizierung von Anforderungen. Es kann als Startpunkt bei der Ermittlung von Anforderungen verwendet werden.

Es gibt eine Reihe von Arbeiten, die Anforderungen in unterschiedliche Typen einordnen und somit eine Klassifikation von Anforderungen erzeugen, zum Beispiel Davis (1993), IEEE (1998a), Kotonya und Sommerville (1998) sowie van Lamsweerde (2001). Unser Template für die Anforderungsermittlung basiert auf dem Ansatz von Glinz (2007). In unserer Klassifizierung brechen wir die Anforderungsklassen von Glinz weiter auf. Wir integrieren im Bereich der spezifischen Qualitätsanforderungen, wie von Glinz vorgeschlagen, die Qualitätsaspekte aus der ISO/IEC 9126 (ISO/IEC 2001). Die Klassifikation von funktionalen Anforderungen ist sehr stark von der jeweiligen Domäne abhängig. Daher ist eine generische Klassifizierung der funktionalen Anforderungen nicht möglich. Wenn Unternehmen oft ähnliche Projekte durchführen, sind sie jedoch in der Lage für ihre Art von Projekten beziehungsweise die von ihnen betrachtete Domäne eine allgemeine Klassifizierung der funktionalen Anforderungen vorzunehmen. In diesem Fall können sie das Template erweitern und somit die Wiederverwendung von Anforderungsklassen ermöglichen.

Das Template für die Anforderungsermittlung ist in Tabelle 4-4 dargestellt.

Projektanforderungen	Anforderungen, die die Planung und Verwaltung des Projekts betreffen
Ökonomische Anforderungen	Anforderungen, die die Ressourcenverfügbarkeit für das Projekt betreffen
Kostenbeschränkungen	Anforderungen, die das Projekt finanziell einschränken
Zeitbeschränkungen	Anforderungen, die das Projekt zeitlich einschränken
Prozessanforderungen	Anforderungen an den Softwareentwicklungsprozess, nach dem das System entwickelt wird
Prozessstandards	Standards, denen der Softwareentwicklungsprozess genügen muss
Dokumentation	Anforderungen an die Art und den Umfang der zu erstellenden Dokumentation
Anforderungen an durchzuführende Tätigkeiten	Anforderungen, die vorgeben, wie bestimmte Entwicklungstätigkeiten durchgeführt werden sollen
Systemanforderungen	Direkte Anforderungen an die Eigenschaften des zu entwickelnden Systems
Funktionale Anforderungen	Anforderungen, die das erwartete Verhalten des Systems oder einer Systemkomponente in Form von Reaktionen auf gegebene Stimuli beschreiben. Dazu gehören auch Anforderungen, die die Funktionen und Daten beschreiben, die für das Verarbeiten der Stimuli und das Erzeugen der Reaktionen benötigt werden
Performanzanforderungen	Anforderungen, bei denen sich das Interesse auf Zeitverhalten, Geschwindigkeit, Volumen oder Durchsatz richtet
Timing	Anforderungen, die die zeitliche Abstimmung des Systemverhaltens betreffen
Geschwindigkeit	Anforderungen an die Ausführungszeiten der einzelnen Funktionen des Systems
Volumen	Anforderungen, die das vom System verarbeitbare Datenvolumen betreffen
Durchsatz	Anforderungen an die Menge an Daten, die innerhalb eines bestimmten Zeitintervalls verarbeitet werden können
Verbrauchsverhalten	Anforderungen bezüglich des Ressourcenverbrauchs des Systems
Spezifische Qualitätsanforderungen	Anforderungen, bei denen sich das Interesse auf einen der in der ISO/IEC 9126 definierten Qualitätsaspekte richtet.
Zuverlässigkeit	Anforderungen an die Fähigkeit, ein bestimmtes Leistungsniveau unter bestimmten Bedingungen aufrechtzuerhalten
Reife	Anforderungen an die Versagenshäufigkeit durch Fehlerzustände
Fehlertoleranz	Anforderungen an die Fähigkeit, ein spezifiziertes Leistungsniveau bei Software-Fehlern oder Nicht-Einhaltung einer spezifizierten Schnittstelle aufrecht zu erhalten
Robustheit	Anforderungen an die Stabilität bei nicht vorhergesehenen Eingaben

Wiederherstellbarkeit	Anforderungen an die Fähigkeit, bei einem Versagen ein spezifiziertes Leistungsniveau wiederherzustellen und die direkt betroffenen Daten wiederzugewinnen
Benutzbarkeit	Anforderungen an die Fähigkeit des Systems, verstanden, erlernt und bedient zu werden sowie seine Attraktivität gegenüber dem Benutzer
Erlernbarkeit	Anforderungen bezüglich des Aufwands für den Benutzer, die Bedienung des Systems zu erlernen
Verständlichkeit	Anforderungen bezüglich des Aufwands für den Benutzer, das System zu verstehen
Attraktivität	Anforderungen bezüglich der Anziehungskraft des Systems auf den Benutzer
Bedienbarkeit	Anforderungen bezüglich des Aufwands für den Benutzer, das System zu bedienen
Übertragbarkeit	Anforderungen an Fähigkeit des Systems, von einer Umgebung in eine andere übertragen zu werden
Anpassbarkeit	Anforderungen an die Fähigkeit, an verschiedene spezifizierte Umgebungen angepasst zu werden
Installierbarkeit	Anforderungen an die Fähigkeit, in einer spezifizierten Umgebung installiert zu werden
Interoperabilität	Anforderungen an die Fähigkeit, mit vorgegebenen Systemen zusammenzuwirken
Ersetzbarkeit	Anforderungen an die Fähigkeit, anstelle eines anderen Systems für den selben Zweck in der selben Umgebung eingesetzt werden zu können
Änderbarkeit	Anforderungen bezüglich der Möglichkeit sowie des Aufwands zur Durchführung von Änderungen am System
Security	Anforderungen an die Fähigkeit, unberechtigten Zugriff auf Programme und Daten zu verhindern
Safety	Anforderungen, die Aussagen zur Begrenzung des Risikos von Schaden treffen, der Personen, Unternehmen, Umwelt oder Besitz durch das System zugefügt werden kann
Constraints	Alle sonstigen Anforderungen
Physikalische Anforderungen	Einschränkungen des Systems aufgrund physikalischer Gegebenheiten in der Umgebung
Juristische Anforderungen	Einschränkungen des Systems aufgrund gesetzlicher Bestimmungen
Kulturelle Anforderungen	Einschränkungen des Systems aufgrund kultureller Gegebenheiten
...	

Tabelle 4-4: Template für die Anforderungsermittlung

4.8.6 Konflikte zwischen Anforderungen

In diesem Produkt werden die identifizierten Konflikte zwischen Anforderungen gesammelt. Es werden jeweils die Namen der Anforderungen sowie eine Kurzbeschreibung des Konfliktes aufgeführt (siehe Kapitel 4.7.2).

4.8.7 Anforderungsspezifikation

Dieses Produkt stellt die detaillierte Spezifikation der Anforderungen dar (siehe Kapitel 4.7.3).

4.8.8 Bericht der Anforderungsvalidierung

In diesem Produkt werden die Ergebnisse der Anforderungsvalidierung festgehalten (siehe Kapitel 4.7.4). Dem Bericht der Anforderungsvalidierung ist das Teilprodukt Iterationsreviewbericht zugeordnet.

Iterationsreviewbericht

Am Ende jeder Entwicklungsiteration wird ein Review durchgeführt. Dessen Ergebnisse werden im Iterationsreviewbericht dokumentiert (siehe Kapitel 4.7.4).

4.8.9 Domänenmodelle

Dieses Produkt umfasst eine Menge von Modellen der Domäne (siehe Kapitel 4.7.1).

4.8.10 Analysemodelle

Zu den Analysemodellen gehören alle Modelle, die während der Anforderungsanalyse zum besseren Verständnis der Anforderungen erstellt wurden (siehe Kapitel 4.7.2).

4.8.11 Systemvision

In der Systemvision wird das zu entwickelnde System auf sehr hohem Abstraktionsniveau beschrieben. Darin enthalten ist die Absteckung der Systemgrenzen. Dieses Dokument wird nicht im Rahmen von REPMUS erstellt und stellt lediglich einen Input dar.

4.9 Zusammenfassung der Unterschiede zwischen REPMUS und Wiegerts' Ansatz

In diesem Kapitel fassen wir die grundlegendsten Unterschiede zwischen REPMUS und Wiegerts' Ansatz, die bereits in den vorangegangenen Kapiteln aufgezeigt wurden, zusammen.

REPMUS geht in Bezug auf die Stakeholderidentifikation geeigneter auf die Besonderheiten der betrachteten Projektklasse ein. Während Wiegerts die Benutzer in den Vordergrund rückt trägt REPMUS der Tatsache Rechnung, dass speziell bei der Entwicklung der ersten Version eines Produktes für den Markt verstärkt auf andere

Stakeholderklassen zurückgegriffen werden muss. Ebenso berücksichtigt REPMUS, dass in änderungsintensiven Softwareentwicklungsprojekten häufig nicht alle Stakeholder zu Beginn bekannt sind und erst im späteren Projektverlauf identifiziert und einbezogen werden können. Die größere Abdeckung verschiedener Stakeholderklassen ermöglicht eine präzisere Aufnahme der Anforderungen und reduziert Änderungsrisiken.

In der Domänenanalyse legt REPMUS einen Schwerpunkt auf schlecht verstandene bzw. komplexe Bereiche. Für diese wird bei REPMUS ein deutlich höherer Aufwand aufgebracht als für gut verstandene Bereiche. Wiegers fordert keine solche Schwerpunktsetzung. Dieses Vorgehen ist in Projekten, in denen die Anforderungen und die Domäne nur schlecht verstanden sind vorteilhaft, um der Instabilität der Anforderungen auf effiziente Art und Weise entgegenzuwirken.

REPMUS legt einen starken Schwerpunkt auf das Nutzen von Feedback aus der Entwicklung, um die Anforderungsermittlung und die Anforderungsanalyse zu stimulieren. Ein solches Vorgehen trägt gerade bei Projekten, in denen die Anforderungen nur schlecht verstanden sind, dazu bei, dass die Anforderungen zu dem Punkt konvergieren an dem sie das tatsächlich gewünschte bzw. benötigte System beschreiben. Bei Wiegers spielt Entwicklungsfeedback eine wesentlich geringere Rolle.

REPMUS nutzt im Gegensatz zu Wiegers' Ansatz ein Template, um die Anforderungsermittlung zu katalysieren. Durch das Template wird eine schnellere und vollständigere Anforderungsermittlung begünstigt. Gerade im Kontext instabiler und schlecht verstandener Anforderungen ist es von Vorteil mittels eines Templates zu erzwingen, dass die Stakeholder alle darin enthaltenen Anforderungsklassen in ihren Überlegungen berücksichtigen und somit eine möglichst vollständige Ermittlung der Anforderungen zu begünstigen.

Unser Ansatz nimmt eine differenzierte Behandlung der Anforderungen abhängig von ihrer Stabilisierbarkeit und ihrem Impact vor. So werden für instabile aber gut stabilisierbare Anforderungen sowie Anforderungen, die einen hohen Impact aufweisen, gründlichere Analysen durchgeführt als für andere. Auch die Spezifikation wird für solche Anforderungen detailreicher gestaltet. Mit diesem Vorgehen geht einher, dass der Gesamtaufwand für RE im Projekt bei höherer Instabilität oder Stabilisierbarkeit der Anforderungen höher ausfällt. Das Änderungsrisiko ist bei REPMUS zudem ein Faktor, der bei der Entscheidung berücksichtigt wird, in welcher Reihenfolge Anforderungen präzisiert, also in detailliertere Anforderungen aufgebrochen, und analysiert werden sollen. Nach Wiegers' Ansatz werden keine solchen Unterscheidungen gemacht. Wie in Kapitel 3 gezeigt erlaubt die Berücksichtigung von Instabilität und Stabilisierbarkeit der Anforderungen und eine Fokussierung des Aufwands auf stabilisierbare Anforderungen, wie dies bei REPMUS der Fall ist, eine effizientere Stabilisierung der Anforderungen und eine Reduzierung der Projektkosten. Um ein solches Vorgehen zu ermöglichen führt REPMUS eine differenzierte Bewertung des Änderungsrisikos vor. Wiegers' Ansatz legt keinen solchen Schwerpunkt in der Risikoanalyse.

Nach REPMUS werden lediglich informelle und semi-formelle Notationen, die möglichst leicht verständlich und änderbar sein sollen, verwendet. Dies bezieht sich auf Domänenmodelle, Analysemodelle und die Anforderungsspezifikation. Wiegers trifft keinerlei solche Einschränkung. Informelle und semi-formelle Notationen kommen der Arbeitsweise von KMU entgegen. Zudem ist in änderungsintensiven Projekten eine leichte Änderbarkeit der Entwicklungsartefakte besonders wichtig, um hohen Änderungskosten entgegenzuwirken.

REPMUS sammelt aus Effizienzgründen nur für besonders instabile Anforderungen Traceability-Informationen während Wiegers dies für alle Anforderungen fordert. REPMUS zielt hierbei auf eine möglichst optimale Balance ab zwischen den Kosten für die Anforderungsverfolgung und der Einsparung in den Änderungskosten, die sich durch die leichtere Auffindbarkeit von zu ändernden Artefakten ergibt.

In Tabelle 4-5 sind die zuvor aufgeführten grundlegendsten Unterschiede zwischen REPMUS und Wiegers' Ansatz überblicksartig zusammengestellt.

REPMUS	Wiegers' Ansatz
Berücksichtigung vielfältiger Stakeholderklassen	Fokussierung auf Benutzer
Überprüfung im Projektverlauf, ob weitere Stakeholder einbezogen werden sollen	Keine Einbeziehung zusätzlicher Stakeholder im Projektverlauf
Schwerpunktsetzung in der Domänenanalyse auf schlecht verstandene und komplexe Bereiche	Keine Schwerpunktsetzung in der Domänenanalyse
Intensive Nutzung von Entwicklungsfeedback zur Stimulation der Anforderungsermittlung und zur Erlangung eines besseren Verständnisses der Anforderungen	Lernprozesse in der Entwicklung spielen eine untergeordnete Rolle in Bezug auf Anforderungsermittlung und Verständnisgewinn
Nutzen eines Templates zur Katalysierung der Anforderungsermittlung und zur Wiederverwendung von Anforderungen	Kein Template als Katalysator für die Ermittlung und Wiederverwendung von Anforderungen
Frühere Detaillierung und Analyse von stabilisierbaren Anforderungen und solchen mit hohem Impact	Reihenfolge, in der die Anforderungen detailliert und analysiert werden, ist nicht abhängig von ihrem Änderungsrisiko
Intensivere Analyse von stabilisierbaren Anforderungen und solchen mit hohem Impact	Keine Variierung des Analysegrads einer Anforderung abhängig von ihrer Stabilisierbarkeit und ihrem Impact
Detailreichere Spezifikation von stabilisierbaren Anforderungen und solchen mit hohem Impact	Keine Variierung des Detaillierungsgrads der Spezifikation einer Anforderung abhängig von ihrer Stabilisierbarkeit und ihrem Impact
Differenzierte Bewertung des Änderungsrisikos	Keine Schwerpunktsetzung in der Risikoabschätzung
Variierung des Gesamtaufwands für RE abhängig von Instabilität und Stabilisierbarkeit der Anforderungen	Keine Variierung des Gesamtaufwands für RE abhängig von Instabilität und Stabilisierbarkeit der Anforderungen
Beschränkung auf informelle und semi-formelle Notationen	Keine Beschränkung auf informelle und semi-formelle Notationen
Sammeln von Traceability-Informationen für besonders instabile Anforderungen	Sammeln von Traceability-Informationen für alle Anforderungen

Tabelle 4-5: Zusammenfassung der grundlegendsten Unterschiede zwischen REPMUS und Wiegers' Ansatz

5 Bewertung von REPMUS anhand der Anforderungen an das Requirements Engineering in änderungsintensiven Projekten

In diesem Kapitel soll REPMUS einer kritischen Bewertung unterzogen werden. Wir bewerten REPMUS unter Zuhilfenahme des Anforderungskatalogs an das RE in zeitbeschränkten und änderungsintensiven Projekten, der in Kapitel 2 erarbeitet wurde.

5.1 Erreichung der Zielsetzungen des Requirements Engineering

In diesem Abschnitt diskutieren wir inwiefern REPMUS die in Kapitel 2.1.3 identifizierten Zielsetzungen des RE erfüllt.

Zielsetzung „Alle Bedürfnisse aller relevanten Stakeholder herausfinden“

Hier sind zusätzlich die folgenden untergeordneten Zielsetzungen zu berücksichtigen: „Anforderungen aufstellen, die notwendig sind, um die Bedürfnisse der Stakeholder abzudecken“, „Gründe für Anforderungen ermitteln“ und „Änderungen der Bedürfnisse der Stakeholder beachten“.

Zunächst ist zu beachten, dass die Gesamtheit der Stakeholder berücksichtigt werden soll. Die Identifikation der Stakeholder und ihre Einbindung in das Projekt sind wichtig, um die wahren Bedürfnisse der Stakeholder herauszufinden (Hofmann und Lehner 2001). REPMUS sucht methodisch nach für das Projekt relevanten Klassen von Stakeholdern. Dies wird durch ein umfassendes Template für die Identifikation von Stakeholderklassen unterstützt. Vertreter für die einzelnen Stakeholderklassen werden benannt und in der Liste der Stakeholderklassen festgehalten. Im Projektverlauf wird überprüft, ob weitere Stakeholder benötigt werden, so dass sich insgesamt eine sehr breite Berücksichtigung der verschiedenen Stakeholder ergibt.

Dadurch, dass bei REPMUS Anforderungen während des gesamten Projekts ermittelt werden und Feedback aus der Entwicklung genutzt werden kann, wird begünstigt, dass sukzessive alle Anforderungen gefunden werden, die nötig sind, um die Bedürfnisse der Stakeholder abzudecken. Es kann jedoch passieren, dass bestimmte Anforderungen fallen gelassen werden müssen, beispielsweise weil sie im Konflikt mit anderen stehen, oder nicht genug Budget zur Verfügung steht, um sie zu realisieren, so dass nicht alle Bedürfnisse der Stakeholder abgedeckt werden können. Dies ist jedoch prinzipiell unvermeidbar. Die Stakeholder stellen ihre Anforderungen selbst auf. Daher leiten sich die Anforderungen, so sie denn richtig verstanden wurden von den Bedürfnissen der Stakeholder ab. In der Anforderungvalidierung überprüfen die Stakeholder noch vor der Realisierung explizit, ob die aufgestellten Anforderungen ihren Bedürfnissen entsprechen. Mittels des Feedbacks aus der Entwicklung und dabei entstandenen Prototypen können komplizierte Anforderungen besser verstanden werden. Mit dem gewonnenen Verständnis können die Anforderungen überarbeitet werden, so dass sie besser die Bedürfnisse der Stakeholder widerspiegeln.

Zu den Anforderungen werden ihre jeweiligen Quellen festgehalten. Eine Anforderung kann mittels der Anforderungshierarchie zu ihren übergeordneten Anforderungen zurückverfolgt werden. Somit werden die Gründe für die Anforderungen dokumentiert.

Dadurch dass Anforderungen während des gesamten Projekts ermittelt werden und jederzeit geändert werden können ist sichergestellt, dass geänderte Bedürfnisse der Stakeholder Beachtung finden.

Zielsetzung „Ein breites Verständnis der Domäne, der Organisation und der Geschäftsprozesse gewinnen“

Um ein breites Verständnis der Domäne, der Organisation und der Geschäftsprozesse zu gewinnen, bindet REPMUS Domänenexperten in den Prozess ein. Zudem werden zur Förderung des Verständnisses von besonders komplizierten Bereichen der Domäne informelle und semi-formelle Geschäftsprozess- und Domänenmodellierungstechniken eingesetzt. Weiterhin können mittels früher Releases Erfahrungen aus dem Einsatz des Systems gesammelt werden. So werden zusätzliche Informationen über die Domäne für das Projekt gewonnen.

Zielsetzung „Auswirkungen des Systems auf die Geschäftsabläufe verstehen“

Mittels früher Releases können Erfahrungen aus dem Einsatz des Systems gesammelt werden. Durch diese können die Auswirkungen des Systems auf die Geschäftsabläufe besser abgeschätzt werden.

Zielsetzung „Return on Investment des Systems und der einzelnen Anforderungen ermitteln“

Der Return on Investment des Systems und der einzelnen Anforderungen wird von REPMUS ermittelt, indem Wichtigkeit und Kosten der Anforderungen abgeschätzt werden. In der Wichtigkeit ist der Geschäftswert der Anforderungen enthalten. Aber auch Anforderungen, die keinen direkten Geschäftswert besitzen, für das System, und damit dessen Profitabilität, jedoch unverzichtbar sind, weisen eine hohe Wichtigkeit auf. Während der Iterationsreviews wird regelmäßig überprüft, ob die Kostenschätzungen korrigiert werden müssen.

Zielsetzung „Systemgrenzen abstecken“

Ein Abstecken der Systemgrenzen wird von REPMUS als Input vorausgesetzt.

Zielsetzung „Die notwendigen Anforderungen zur Erreichung der wichtigsten Ziele auswählen“

Zunächst werden high-level Anforderungen aufgestellt, welche den Zielen des Projekts entsprechen. Von diesen werden Anforderungen abgeleitet. Mittels der Anforderungshierarchie sind alle Anforderungen somit auf die Ziele zurückführbar. Der Geschäftswert, beziehungsweise allgemein die Wichtigkeit, der Anforderungen hat einen beträchtlichen Einfluss auf ihre Priorität. Daher kann davon ausgegangen werden, dass die zur Realisierung ausgewählten Anforderungen die wichtigsten Ziele des Projekts abdecken.

Zielsetzung „Nur machbare Anforderungen berücksichtigen“

Hierbei sind die untergeordneten Zielsetzungen „Sich klar machen, welche Anforderungen nicht realisiert werden können“, „Konflikte zwischen Anforderungen auflösen und dabei den wichtigeren den Vorrang geben“, „Ermitteln, welche Anforderungen in der

Entwicklung Vorrang haben“, „Budget beachten“, „Zeitplan beachten“ sowie „Den Stakeholdern die Machbarkeit der Anforderungen kommunizieren“ zu beachten.

REPMUS führt in der Anforderungsanalyse eine auf Gruppierung von Anforderungen basierende Konfliktanalyse durch, so dass Konflikte schon vor der Realisierung entdeckt und aufgelöst werden können. Bei der Auflösung der Konflikte wird die Priorität der Anforderungen berücksichtigt. Zu Konflikten gehören auch Konflikte mit dem Zeit- und Budgetplan. Konflikte, die erst während der Entwicklung entdeckt werden, werden spätestens im nächsten Iterationsreview den Stakeholdern kommuniziert, die entsprechende Korrekturen vornehmen. Falls sich während der Entwicklung abzeichnet, dass Anforderungen nicht innerhalb des Budgets oder innerhalb des Zeitplans realisiert werden können, so wird dies ebenfalls spätestens im nächsten Iterationsreview an die Stakeholder kommuniziert.

Bei der Ermittlung der Priorität der Anforderungen werden neben ihrer Wichtigkeit auch ihre Kosten berücksichtigt, so dass Anforderungen, deren Nutzen den mit ihnen verbundenen zeitlichen und finanziellen Aufwand nicht rechtfertigt, im Vergleich zu anderen vernachlässigt werden.

Die Stakeholder sind über die Machbarkeit der Anforderungen informiert, da sie an den damit verbundenen Betrachtungen direkt beteiligt sind.

Zielsetzung „Anforderungen den Entwicklern kommunizieren“

Zu diesem Punkt sind auch die untergeordneten Zielsetzungen „Gewährleisten, dass die Anforderungen für die Entwickler verständlich sind“ und „Anforderungen korrekt erfassen“ zu berücksichtigen.

Bei REPMUS werden die Anforderungen den Entwicklern auf zwei Weisen kommuniziert. Zum einen sind Vertreter der Entwicklung in die RE-Aktivitäten involviert, zum anderen wird eine Anforderungsspezifikation erstellt, in der die relevanten Informationen über die Anforderungen festgehalten werden. Dabei werden leicht verständliche Notationen verwendet. Die verwendeten informellen und semi-formellen Notationen sind jedoch nicht immer eindeutig und erlauben nicht in jedem Fall eine präzise und korrekte Abbildung der Anforderungen. Daher ist die Korrektheit der Anforderungsspezifikation nicht immer gewährleistet. Dieser Mangel wird einerseits dadurch ausgeglichen, dass Entwickler an RE-Workshops beteiligt werden und somit auch ohne Anforderungsspezifikation ein gewisses Verständnis für die Anforderungen besitzen. Andererseits ist es mittels kurzer Feedbackzyklen kurzfristig möglich, in Iterationsreviews Unklarheiten bezüglich der Anforderungen zu klären.

Die korrekte Erfassung der Anforderungen wird in der Anforderungvalidierung explizit durch die Stakeholder überprüft. Werden in der Entwicklung Fehler in den Anforderungen entdeckt, so werden diese spätestens in den Iterationsreviews den Stakeholdern kommuniziert, damit entsprechende Korrekturen vorgenommen werden können. Auch aufgetretene Missverständnisse bezüglich der Anforderungen können in Iterationsreviews geklärt werden.

Zielsetzung „Einhaltung der Anforderungen gewährleisten“

An dieser Stelle sind auch die untergeordneten Zielsetzungen „Risiken in der Entwicklung identifizieren und Vorkehrungsmaßnahmen sowie Ausweichpläne bereitstellen“ und „Risiken den Stakeholdern kommunizieren“ zu beachten.

Risiken werden identifiziert. Besondere Beachtung findet dabei die Abschätzung des Änderungsrisikos der Anforderungen. Vorkehrungsmaßnahmen sind zum einen das Fallenlassen oder Abändern von Anforderungen sowie das Durchführen von zusätzlichen Analysen, um das Verständnis für die Anforderungen zu erhöhen. Ausweichpläne werden nicht definiert. Eine Stärke des Ansatzes liegt darin, dass aufgrund des zur Entwicklung parallelen und relativ aufwandsarmen RE flexibel auf auftretende Änderungen und Probleme reagiert werden kann.

Die Stakeholder sind über die Risiken informiert, da sie an der Risikoanalyse direkt beteiligt werden.

Um die Einhaltung der Anforderungen zu gewährleisten wird zudem in Iterationsreviews explizit überprüft, ob die Ergebnisse der Entwicklung den Vorstellungen der Stakeholder beziehungsweise den spezifizierten Anforderungen entsprechen. Dies wird durch den Einsatz von Akzeptanztests unterstützt.

5.2 Berücksichtigung der Gründe für Änderungen der Anforderungen

In diesem Abschnitt diskutieren wir inwiefern REPMUS die in Kapitel 2.2.1 identifizierten Gründe für Anforderungsänderungen berücksichtigt. Wir beschränken uns in unserer Betrachtung auf die Faktoren, die durch die Vorgehensweise im Projekt beeinflusst werden können.

Änderungsgrund „Faktoren bezüglich des Verständnisses der Anforderungen“

Hierbei ist eine Reihe von Änderungsursachen zu berücksichtigen. Anforderungen können während der frühen Phasen des Projekts nicht richtig verstanden worden sein, Stakeholder können im Laufe des Projektes ein besseres Verständnis für das gewünschte System entwickeln und Anforderungen können aufgrund von Kommunikationsproblemen falsch interpretiert werden.

Der optimale Umgang mit dieser Ursache für Anforderungsänderungen ist ein zentrales Element von REPMUS. Es wird abgeschätzt wie instabil die Anforderungen sind und inwieweit sie durch Analysen zum besseren Verständnis stabilisiert werden können. Davon ausgehend wird bestimmt, wie viel Aufwand in die Anforderungsanalyse investiert werden soll um die Summe aus Änderungskosten und Analyseaufwand möglichst gering zu halten. Gerade Anforderungen, die als instabil aber stabilisierbar gelten oder besonders hohe Änderungskosten verursachen können, werden besonders gründlich in den frühen Phasen des Projekts untersucht. REPMUS versucht daher Anforderungsänderungen aufgrund von mangelndem Verständnis zu vermeiden. Dies jedoch nur in dem Grad, bei dem der Nutzen der Anforderungsanalyse noch ihren Aufwand rechtfertigt. Um ein ausreichendes Verständnis für die Anforderungen zu erhalten, wird zum einen versucht mittels des Einsatzes von Templates zur Anforderungsermittlung eine möglichst vollständige Abdeckung aller high-level Anforderungen zu erhalten. Zum anderen werden mittels informeller und semi-formeller Modellierungstechniken Domäne und Anforderungen genauer untersucht. REPMUS baut jedoch auch auf Lernprozesse während der Entwicklung, besonders dann wenn eine weitergehende Anforderungsanalyse vor Beginn der Entwicklung als nicht effizient angesehen wird oder erwartet wird, dass mittels Feedback aus der Entwicklung bestimmte Anforderungen wesentlich leichter und besser verstanden werden können. Durch dieses Vorgehen erfolgen einige notwendige Änderungen der Anforderungen erst im Laufe des Projektes. Aufgrund der Abwägung

zwischen Analyseaufwand und Änderungsaufwand sehen wir dies jedoch nicht als nachteilig an.

REPMUS versucht alle relevanten Stakeholder bereits zu einem möglichst frühen Zeitpunkt in das Projekt zu integrieren und unterstützt dies durch eine methodische Suche nach den für das Projekt relevanten Stakeholdern. Bei den verschiedenen RE-Aktivitäten werden soweit möglich alle relevanten Stakeholder beteiligt. Diese intensive Integration schafft die Möglichkeit, dass die Stakeholder schon zu Beginn des Projekts ein gutes Verständnis für das gewünschte System und die Anforderungen entwickeln. Auf diese Weise können spätere Änderungen, die daher rühren, dass die Stakeholder erst im Laufe des Projekts ein besseres Verständnis für das gewünschte System entwickeln, verhindert werden.

Die zeitlich verschränkte Durchführung von RE und den übrigen Entwicklungsaktivitäten bedeutet, dass auch während der Entwicklung das Verständnis für die Anforderungen aufgrund durchgeführter Analysen steigt. Dies bedeutet unmittelbar eine höhere Änderungsrate im Laufe des Projekts im Vergleich zu Vorgehensweisen in denen das RE sequenziell vor den übrigen Entwicklungsaktivitäten erfolgt. Die verschränkte Durchführung ist ein Zugeständnis von REPMUS an TTM. Andererseits ist hervorzuheben, dass durch das Nutzen von Lernprozessen während der Entwicklung und Feedback aus Entwicklung und Einsatz des Systems oft im Projektverlauf ein höheres Verständnis der Stakeholder für die Anforderungen erreicht werden kann als dies bei einer sequenziellen Durchführung von RE und den übrigen Entwicklungsaktivitäten der Fall wäre (Alexander und Stevens 2002; Karlsson et al. 2002), was wiederum die Änderungsrate verringert.

Falsche Interpretationen von Anforderungen aufgrund von Kommunikationsproblemen können zu Änderungen führen. REPMUS fordert den Einsatz von informelle und semi-formelle Notationen für die Anforderungsspezifikation, wie zum Beispiel natürliche Sprache, Use Cases, Szenarios oder UML-Sequenzdiagramme. Diese sind einerseits leicht verständlich, so dass die Stakeholder sie in der Regel gut verstehen können. Andererseits fehlt ihnen jedoch die formale Präzision, so dass es zu Fehlinterpretationen aufgrund von Ungenauigkeiten, beziehungsweise Mehrdeutigkeiten der Spezifikation kommen kann. Das wichtigste Instrument von REPMUS, um dem entgegenzuwirken ist alle relevanten Stakeholder in RE-Workshops zusammenzuführen und mittels direkter Kommunikation ein gemeinsames Verständnis der Anforderungen zu erreichen. Von essenzieller Wichtigkeit ist dabei, dass auch die Entwickler mit einbezogen werden, um die Kommunikationsschnittstelle zwischen RE und Realisierung auf diese Weise zu überbrücken.

Ungenauigkeiten in der Anforderungsspezifikation können zu deren Fehlinterpretation seitens der Entwickler führen. REPMUS sieht eine explizite Überprüfung der Anforderungsliste und der Anforderungsspezifikation vor, um Ungenauigkeiten und Fehler in den Anforderungen vor deren Realisierung zu beheben. In Iterationsreviews besteht zudem die Möglichkeit, Unklarheiten in den Anforderungen zu klären und Fehlinterpretationen seitens der Entwicklung aufzudecken.

Änderungsgrund „Falsche Schätzung der Kosten von Anforderungen“

Sowohl das Budget als auch der Zeitplan können eventuell für die Realisierung der ausgewählten Anforderungen nicht ausreichend sein.

Aufwandsschätzungen zu den Anforderungen werden von den an der Realisierung beteiligten Stakeholdern, also vorrangig der Entwickler, selbst vorgenommen, wodurch

sich ihre Präzision erhöht. REPMUS verwendet kein umfangreiches Kostenschätzungsmodell wie zum Beispiel COCOMO. Aufwandsschätzungen werden stattdessen rein intuitiv vorgenommen. Der Umstand, dass einige Anforderungen nicht zu Beginn oder gar nicht gründlich analysiert und somit schlecht verstanden sind, trägt zu zeitweisen Ungenauigkeiten in den Aufwandsschätzungen bei, woraus zusätzliche Anforderungsänderungen resultieren können.

Änderungsgrund „Prioritäten der Anforderungen ändern sich“

REPMUS nutzt für die Ermittlung der Wichtigkeit von Anforderungen relative Werte, was die Präzision der Prioritätseinschätzung erhöht (Davis 2003). Zudem werden sämtliche Stakeholder an der Prioritätsabschätzung beteiligt. Auch dies führt zu einer höheren Genauigkeit der Priorität der Anforderungen (Hofmann und Lehner 2001). Ebenfalls zu beachten ist, dass sich Prioritäten mit wachsendem Verständnis der Anforderungen ändern können. Da die RE-Aktivitäten zeitlich verschränkt mit der Entwicklung durchgeführt werden und das Verständnis für die Anforderungen auch während dieser noch stark wächst, wird sich auch die Einschätzung der Priorität der einzelnen Anforderungen im Projektverlauf ändern. Dies kann zu Anforderungsänderungen führen.

Änderungsgrund „Machbarkeit der Anforderungen wurde falsch eingeschätzt“

Anforderungen können aufgrund technischer Probleme nicht realisiert werden oder sind eventuell nicht innerhalb des Zeitplans und des Budgets realisierbar.

Wie auch bei der Einschätzung des Priorität ist bei der korrekten Einschätzung der Machbarkeit bei REPMUS positiv hervorzuheben, dass alle relevanten Stakeholder und insbesondere die Entwickler an der Abschätzung beteiligt werden. Dadurch und durch die explizit vorgenommenen Risikoabschätzungen kann die Machbarkeit relativ gut eingeschätzt werden. Zu betonen ist dabei die Schwerpunktlegung von REPMUS auf die Analyse des Änderungsrisikos von Anforderungen. Aufgrund der zeitlichen Verschränkung von RE und Realisierung und des damit einhergehenden starken Verständnisergebnisses auch im Projektverlauf kann jedoch im Gegensatz zu sequenziellen Modellen in der initialen RE-Phase die Machbarkeit weniger präzise abgeschätzt werden. Durch den Nutzen von Feedback aus der Entwicklung kann ab einem bestimmten Zeitpunkt im Projekt oft jedoch sogar ein höheres Verständnis der Anforderungen erreicht werden. Ab diesem Zeitpunkt kann man davon ausgehen, dass die Machbarkeit der Anforderungen mindestens so gut wie in sequenziellen Vorgehensweisen eingeschätzt werden kann.

Änderungsgrund „Konflikte zwischen Anforderungen werden gefunden“

REPMUS führt in der Anforderungsanalyse eine Konfliktanalyse zwischen den Anforderungen durch. Diese wird durch Gruppierung von Anforderungen erleichtert. Somit wird diesem Änderungsgrund entgegengewirkt. Analog zu den Überlegungen bezüglich Priorität und Machbarkeit der Anforderungen werden aufgrund der zeitlich verschränkten Durchführung von RE und Realisierung manche Konflikte etwas später als bei sequenziellen Vorgehensweisen gefunden, wodurch einige Anforderungsänderungen zeitlich verzögert werden.

Änderungsgrund „Neue Stakeholder werden am Projekt beteiligt“

REPMUS führt zu Beginn des Projekts eine methodische Suche nach relevanten Stakeholdern durch, die zudem durch den Einsatz eines Templates unterstützt wird. Dieses

Vorgehen begünstigt eine möglichst vollständige Abdeckung aller Stakeholderklassen bereits zu Beginn des Projekts und wirkt somit der hier betrachteten Änderungsursache entgegen. Es wird jedoch während des Projektverlaufs erneut geprüft, ob weitere Stakeholder hinzuzufügen sind. Insbesondere in marktgetriebenen Softwareentwicklungsprojekten, in denen die erste Version eines Produkts entwickelt wird, ist es unvermeidbar, dass bestimmte Stakeholder erst im Projektverlauf identifiziert und beteiligt werden können. Durch die Möglichkeit Stakeholder auch später im Projekt noch zu integrieren begünstigt REPMUS Anforderungsänderungen. Die Alternative, die Stakeholder, die erst spät im Projektverlauf hinzukommen können, nicht am Projekt zu beteiligen würde jedoch den Projekterfolg gefährden, da somit womöglich wichtige Anforderungen ignoriert werden würden.

5.3 Berücksichtigung der Schwierigkeiten aufgrund von Änderungen der Anforderungen

In diesem Abschnitt führen wir auf, mit welchen Mitteln REPMUS den in Kapitel 2.2.2 identifizierten Schwierigkeiten begegnet, die bei auftretenden Anforderungsänderungen zu bewältigen sind.

Notwendige Änderungen müssen identifiziert werden

Es kann prinzipiell nicht verhindert werden, dass Änderungsvorschläge identifiziert und über sie entschieden werden muss. Das Einbringen von Änderungsvorschlägen zu verhindern würde zur Folge haben, dass man nicht das Produkt entwickelt, das wirklich gewünscht ist. REPMUS unterscheidet sich in diesem Punkt nicht von anderen Ansätzen.

Zusätzlicher Aufwand entsteht

Stakeholder müssen erneut versammelt werden, um über die Annahme von Änderungsvorschlägen zu entscheiden: Dies kann prinzipiell nicht verhindert werden. Bei REPMUS werden aufgrund der teilweise zur Entwicklung parallelen Durchführung des RE die Stakeholder ohnehin zu späteren Zeitpunkten erneut versammelt. Zu diesen Gelegenheiten kann ebenfalls über die Annahme von Änderungsvorschlägen entschieden werden, ohne dass dies einen bedeutenden Zusatzaufwand darstellt. REPMUS verringert somit die üblicherweise durch diesen Faktor erwarteten Kosten etwas.

Artefakte müssen angepasst werden: Da REPMUS nicht für alle Anforderungen Traceability-Informationen sammelt ist die Identifikation der anzupassenden Artefakte nicht immer so leicht zu bewerkstelligen wie dies beispielsweise bei Wiegers' Ansatz der Fall ist. Unsere Methode führt jedoch für Anforderungen, die als besonders instabil gelten, eine Anforderungsverfolgung durch. REPMUS versucht gerade in diesem Punkt den oft durchaus bedeutenden Aufwand für die Anforderungsverfolgung zu vermeiden und sammelt nur für Anforderungen, bei denen dieser Aufwand als lohnenswert empfunden wird, die entsprechenden Traceability-Informationen. Die Modifikation der Anforderungsspezifikation ist aufgrund der Verwendung von leichtgewichtigen Notationen, also solchen, die eine aufwandsarme Änderung erlauben, relativ einfach.

Anfänglicher Aufwand kann vergeudet sein, wenn Anforderungen sich ändern: REPMUS wägt ab, ob Anforderungen durch eine gründliche Anforderungsanalyse soweit stabilisiert werden können, dass sich der eingesetzte Aufwand dahingehend bezahlt macht, als dass potenzieller späterer Änderungsaufwand vermieden wird. Für Anforderungen, die als nicht

stabilisierbar gelten, weil sie aufgrund von Faktoren instabil sind, die nicht vom Vorgehen im Projekt beeinflusst werden können, wird die Anforderungsanalyse zeitlich nach hinten verschoben. Dies geschieht um zu vermeiden, dass anfänglicher Aufwand vergeudet wird, wenn sich diese Anforderungen ändern.

Anforderungen werden möglicherweise abgelehnt, da man den Zusatzaufwand scheut: Außer dass REPMUS bei den Projektbeteiligten ein Bewusstsein über die Notwendigkeit von Anforderungsänderungen schafft, unternimmt der Ansatz keine expliziten Maßnahmen, um dieser Schwierigkeit zu begegnen.

Nicht alle Konflikte können zu Projektbeginn erkannt werden

Wie auch XP und Wiegers' Ansatz hält REPMUS keine Mittel bereit, um den mit diesem Punkt assoziierten Schwierigkeiten – „Konfliktanalyse muss mehrfach im Laufe des Projekts durchgeführt werden“, „Entwicklungsergebnisse müssen teilweise verworfen werden, weil sie im Konflikt mit neuen Anforderungen stehen“ und „Wichtige Anforderungen werden möglicherweise abgelehnt, da sie im Konflikt mit alten Anforderungen stehen“ – zu begegnen. Die Studien von Nurmaliani und Zowghi (2006) deuten darauf hin, dass die Kosten für die Behebung von Konflikten eher gering sind. Demzufolge ist dieser Faktor weniger wichtig.

Nicht dokumentierte Überlegungen zu den Anforderungen sind später nicht mehr präsent, werden jedoch bei Anforderungsänderungen wieder benötigt

REPMUS begegnet diesem Problem, indem das Wissen und die angestellten Überlegungen zu den Anforderungen durch RE-Workshops, bei denen alle relevanten Stakeholderbeteiligt werden, einer Vielzahl von Personen vermittelt werden. Auf diese Weise steigt die Wahrscheinlichkeit, dass nicht dokumentierte Überlegungen auch später noch präsent sind.

Projektplan muss angepasst werden

Auch zu diesem Punkt weist REPMUS ähnlich wie XP und Wiegers' Ansatz keine Mechanismen auf, um den damit assoziierten Schwierigkeiten – „Schätzungen müssen aktualisiert werden“, „Inhalt von Releases und Iterationen kann sich ändern“ und „Zeitplan und Budget müssen angepasst werden“ – zu begegnen.

Kostenprobleme entstehen

Kosten des Systems sind zu Beginn des Projektes nicht bekannt: Zum einen führt REPMUS für alle Anforderungen Kostenschätzungen durch. Zum anderen legt unsere Methode einen Schwerpunkt darauf, das Änderungsrisiko der Anforderungen zu bestimmen. Darunter fällt unter anderem die Abschätzung der Stabilität der Anforderungen sowie ihres Impacts, also der bei einer Änderung auftretenden Kosten. Daher können auch die erwarteten Änderungskosten grob abgeschätzt werden.

Änderungen lassen sich möglicherweise nicht innerhalb des Budgets realisieren: Da REPMUS bereits zu Beginn des Projekts potenzielle Änderungskosten abschätzt, können frühzeitig das Budget oder der Umfang des Systems angepasst werden, um Anforderungsänderungen gerecht zu werden.

Späte Änderungen können besonders teuer sein: REPMUS versucht gerade besonders teure Anforderungsänderungen zu vermeiden. Zu diesem Zweck werden Anforderungen, deren Änderung als sehr aufwändig eingeschätzt wird, besonders gründlich analysiert werden,

um ein hohes Verständnis für diese zu erreichen und ihre Änderungswahrscheinlichkeit gering zu halten.

5.4 Zusammenfassung

Wir diskutierten in diesem Kapitel, inwiefern REPMUS die in Kapitel 2.1.3 identifizierten Zielsetzungen, die man mit dem RE verfolgt, erreicht. Unsere Analyse ergab, dass die Methode fast alle Zielsetzungen ausreichend abdeckt. Problematisch sehen wir lediglich die Zielsetzung „Anforderungen korrekt erfassen“ an. Unser Ansatz nutzt informelle und semi-formelle Spezifikationstechniken, um Zeit einzusparen und zudem, die für ihn essenzielle Partizipation der Stakeholder einfacher zu gestalten. Dies zieht jedoch Nachteile hinsichtlich der Korrektheit, insbesondere der Eindeutigkeit der Spezifikation nach sich. Durch die direkte Einbeziehung von Entwicklern in RE-Workshops sowie kurze Feedbackzyklen gleicht REPMUS die durch diesen Mangel entstehenden Kommunikationsprobleme aus. Insgesamt ergibt sich eine sehr gute Abdeckung der Zielsetzungen des RE.

Die Analyse hinsichtlich der Beachtung der verschiedenen Ursachen für Anforderungsänderungen seitens REPMUS ergab, dass der Prozess alle wesentlichen Gründe für Änderungen adressiert und ihnen sinnvoll entgegenwirkt. Besonders hervorzuheben ist dabei, dass der Hauptgrund für Anforderungsänderungen, nämlich das mangelnde Verständnis von Anforderungen, für REPMUS eine zentrale Rolle spielt. Der Ansatz versucht explizit das RE derart zu gestalten, dass das Verhältnis zwischen Aufwand im RE und Änderungsaufwand aufgrund von Änderungen wegen mangelndem Verständnis möglichst günstig ausfällt.

Auch die Schwierigkeiten aufgrund von Anforderungsänderungen werden von REPMUS größtenteils adressiert. Es gibt jedoch einige Problemfelder, für deren Bewältigung REPMUS keine expliziten Mittel bereithält: „Notwendige Änderungen müssen identifiziert werden“, „Nicht alle Konflikte können zu Projektbeginn erkannt werden“ und „Projektplan muss angepasst werden“. Dies sind jedoch Probleme, für deren Bewältigung auch XP und Wiegers' Ansatz keine Maßnahmen bereitstellen.

6 Zusammenfassung

In diesem abschließenden Kapitel fassen wir die Ergebnisse der Arbeit kurz zusammen.

Wir haben in dieser Arbeit die Anforderungen, die in änderungsintensiven Projekten an den RE-Prozess gestellt werden, identifiziert (Forschungsfrage 1). Wir haben angeführt, dass die Instabilität von Anforderungen eine Fülle von Ursachen haben kann. Die Gründe für Anforderungsänderungen haben wir in zwei Kategorien eingeteilt. Einerseits gibt es Gründe für Anforderungsänderungen, die projektextern, also vom Vorgehen im Projekt nicht beeinflussbar, sind. Andererseits gibt es Instabilitätsfaktoren, die vom gewählten Vorgehen im RE herrühren oder durch dieses beeinflusst werden können. Ein RE-Prozess sollte die aufgeführten Änderungsgründe berücksichtigen und falls praktikabel geeignete Mittel einsetzen, um ihnen entgegenzuwirken. Wir haben die Schwierigkeiten identifiziert, die sich einem Projekt bei auftretenden Anforderungsänderungen stellen, allen voran den Aufwand zur Anpassung der Entwicklungsartefakte. Ein RE-Prozess muss mit diesen Schwierigkeiten geeignet umgehen, beziehungsweise zu ihrer Linderung beitragen. Neben Spezifika der betrachteten Projektklasse, wie dem Umgang mit Ursachen für Anforderungsinstabilität beziehungsweise mit auftretenden Änderungen, haben wir eine Reihe an allgemeinen Zielsetzungen identifiziert, für deren Erreichung ein RE-Prozess sowohl in änderungsintensiven als auch in anderen Projekten effektive Mittel bereitstellen muss.

Eines der Hauptprobleme, dem sich ein Projekt mit besonders instabilen Anforderungen stellen muss, ist es, den Änderungsaufwand einzugrenzen. Unsere Arbeit betont, dass es nötig ist zwischen Änderungsaufwand und Aufwand für die Stabilisierung von Anforderungen abzuwägen. Die beispielsweise von einigen agilen Ansätzen gewählte Vorgehensweise, in TTM-Projekten mit starker Anforderungsinstabilität massiv Aufwand im RE einzusparen, muss im Lichte der eingangs erwähnten Tatsache, dass viele Anforderungsänderungen vom Vorgehen im RE selbst herrühren, kritisch betrachtet werden. Vertreter von agilen Methoden konstatieren oft, dass sich Anforderungen ohnehin ändern werden und darin investierter Aufwand somit oft vergeudet sein kann. Hierbei wird die Problematik jedoch nicht differenziert genug betrachtet. Geht man davon aus, dass ein Projekt vor allem Anforderungsänderungen aus externen Quellen, die vom Vorgehen im Projekt nicht beeinflusst werden können, ausgesetzt ist, so mag die Vorgehensweise sinnvoll sein, den Aufwand im RE auf ein Minimum zu reduzieren. Sind die Anforderungsänderungen jedoch durch das Vorgehen beeinflussbar, zum Beispiel rühren sie allein von einem Mangel an Verständnis der Anforderungen her, der mittels geeigneter Analysen beseitigt werden könnte, so ergibt sich ein anderes Bild: Wird der RE-Aufwand auf ein Minimum reduziert, so ist das Projekt einer wesentlich größeren Zahl an Änderungen ausgesetzt und der im RE eingesparte Aufwand kann von steigenden Änderungskosten überschattet werden.

Es ist also je nach den gegebenen Projektparametern sinnvoll das Vorgehen im RE, insbesondere die Gestaltung des RE-Aufwands, anzupassen. Wir haben in dieser Arbeit untersucht welche Vorgehensweisen hinsichtlich der Verteilung des RE-Aufwands sich im Rahmen von änderungsintensiven Projekten besonders eignen (Forschungsfrage 2). Zu diesem Zweck haben wir ein entsprechendes System Dynamics Modell entwickelt. Im Rahmen dieser Arbeit wurde das System Dynamics Modell in zwei industriellen Fallstudien entwickelt und angewandt. Dabei haben wir gezeigt, dass abhängig von bestimmten Projektparametern unterschiedliche Vorgehensweisen im RE vorteilhaft hinsichtlich Projektlaufzeit und -kosten sind. Zudem wurde anhand bestehender Studien

aufgezeigt welche Vorgehensweisen beziehungsweise Techniken für marktgetriebene Softwareentwicklungsprojekte von KMU, die einer hohen Anforderungsinstabilität ausgesetzt sind, vorteilhaft sind.

Weiterhin haben wir mittels des entwickelten System Dynamics Modells in beiden industriellen Fallstudien gezeigt, dass ein gesteigerter Aufwand im RE zu Beginn des Projektes die Änderungskosten aufgrund von Anforderungsinstabilität reduziert. Ähnliches ergaben die Literaturstudien, die im Zuge der Erstellung des Anforderungskatalogs für RE-Prozesse im von uns betrachteten Projektkontext sowie der Erstellung unseres RE-Prozesses angefertigt wurden. Es zeigte sich, dass die Intensivierung der RE-Phase zu Beginn des Projekts ein praktikabler Ansatz zur Vermeidung von Anforderungsänderungen ist. Mittels des System Dynamics Modells konnten wir zudem in beiden Fallstudien zeigen, dass der hinsichtlich der Erreichung geringer Projektlaufzeit und Projektkosten geeignete RE-Aufwand von der Stabilisierbarkeit der Anforderungen abhängig ist. Die Simulation zeigte weiterhin, dass es lohnenswert ist den RE-Aufwand auf diejenigen Anforderungen zu konzentrieren, die sich durch eine Anforderungsanalyse in hohem Maße stabilisieren lassen.

Eine vollständige Validierung des System Dynamics Modells konnte im Rahmen dieser Arbeit nicht durchgeführt werden, da das Modell über eine Vielzahl von Industrieprojekten kalibriert werden müsste. Daher können nur die zuvor beschriebenen qualitativen Aussagen mit dem Modell generiert werden. Bei einer genaueren Kalibrierung des Modells über etliche Industrieprojekte wäre es prinzipiell denkbar weitreichendere quantifizierte Aussagen aus dem Modell abzuleiten. Für die Zwecke dieser Arbeit ist dies jedoch nicht erforderlich.

Auf Basis der gewonnenen Erkenntnisse über RE-Vorgehensweisen, die einen hinsichtlich Projektlaufzeit und –kosten geeigneten Umgang mit Anforderungsinstabilität erlauben, und unter Einbeziehung der anfangs identifizierten Anforderungen an das RE in änderungsintensiven Projekten erstellten wir einen auf die spezifischen Bedürfnisse der von uns betrachteten Projektklasse zugeschnittenen RE-Prozess (Forschungsfrage 3). Dieser basiert auf dem Ansatz von Wiegers (2005). Eine zentrale Vorgehensweise des Prozesses ist es, Anforderungen, die gut stabilisierbar sind und solche, die einen hohen Impact aufweisen, einer detaillierteren Analyse zu unterziehen. Eine vollständige Validierung des RE-Prozesses konnte im Rahmen dieser Arbeit nicht durchgeführt werden, da eine solche den Einsatz des Prozesses in einer Vielzahl von industriellen Projekten erfordert. Aus diesem Grund wurde der Prozess weitgehend an den Ansatz von Wiegers angelehnt. Er stützt sich zudem stark auf bestehende Erkenntnisse aus der Literatur ab. Weiterhin ergab eine Analyse und Bewertung anhand der eingangs aufgestellten Anforderungen an das RE in änderungsintensiven Projekten, dass der erstellte RE-Prozess die charakteristischen Problemstellungen der von uns betrachteten Projektklasse geeignet adressiert.

Literaturverzeichnis

Abdel-Hamid T. K. und Madnick, S.E., *Software Projects Dynamics: an Integrated Approach*, Prentice-Hall, Englewood Cliffs, 1991.

Alexander, I. F. und Stevens, R., *Writing Better Requirements*, Addison-Wesley, Harlow, 2002.

Ambler, S. W., *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*, Wiley, New York, 2002.

Arnold, R. S. und Bohner, S. A., „Impact Analysis: Towards a Framework for Comparison”, Proceedings of the International Conference on Software Maintenance, 1993, 292–301.

Beck, K., *Extreme Programming Explained: Embrace Change*, Addison-Wesley, Boston 2000.

Beck, K., *Extreme Programming: Das Manifest*. Addison-Wesley, München, 2003.

Beck, K. und Fowler, M., *Extreme Programming Planen*, Addison-Wesley, München, 2001.

Berkovich, M., Esch, S., Leimeister, J. M. und Krcmar, H., “Requirements Engineering for Hybrid Products as Bundles of Hardware, Software and Service Elements – a Literature Review”, 9. Internationale Tagung Wirtschaftsinformatik (WI 2009), Wien, 2009.

Berry, D. M., „The Inevitable Pain of Software Development, Including of Extreme Programming, Caused by Requirements Volatility”, International Workshop on Time-Constrained Requirements Engineering, Essen, 2002.

Boehm, B. W., „Software Engineering“, *IEEE Transactions on Computers*, **25**(12), 1976, 1226-1241.

Boehm, B.W., *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, 1981.

Boehm, B. W., „Software Risk Management: Principles and Practices“, *IEEE Software*, **8**(1), 1991, 32-41.

- Boehm, B. W., „Requirements that Handle IKIWISI, COTS, and Rapid Change”, *IEEE Computer*, **33**(7), 2000, 99-102.
- Boehm, B. W., „Get Ready for the Agile Methods, With Care“, *IEEE Computer*, **35**(1), 2002, 64-69.
- Boehm, B. W., Abts, C., Clark, B. und Devnani-Chulani, S., „COCOMO II Model Definition Manual: Version 1.4“, University of Southern California, Los Angeles, 1997.
- Boehm, B. W. und Basili, V. R., „Software Defect Reduction Top 10 List“, *IEEE Computer*, **34**(1), 2001, 135-137.
- Boehm, B. W., Bose, P., Horowitz, E. und Lee, M., „Software Requirements as Negotiated Win Conditions“, Proceedings of the First International Conference on Requirements Engineering, Colorado Springs, 1994, 74-83.
- Boehm, B. W., McClean, R. K. und Urfrig, D. B., „Some Experience with Automated Aids to the Design of Large-Scale Reliable Software“, *IEEE Transactions on Software Engineering*, **1**(1), 1975, 125-33.
- Boehm, B. W. und Papaccio, P., „Understanding and Controlling Software Costs“, *IEEE Transactions on Software Engineering*, **14**(10), 1988, 1462-1477.
- Bossel H., *Modellbildung und Simulation*, Vieweg, Braunschweig, 1992.
- Brooks, F. P., *The Mythical Man Month*, Addison-Wesley, Boston, 1995.
- Broy, M., Gronau, N. und Wildemann, H. (Hrsg.), *Gestaltung interorganisationaler Software-Entwicklung: Herausforderungen durch Wandlungsfähigkeit und Wiederverwendung*, GITO Verlag, Berlin, 2010. (im Erscheinen)
- Card, D. N., „The RAD Fad: Is Timing Really Everything?“, *IEEE Software*, **12**(5), 1995, 20-22.
- Carlshamre, P., „A Usability Perspective on Requirements Engineering: From Methodology to Product Development“, Dissertation, Linköpings University, 2001.
- Carmel, E. und Becker, S., „A Process Model for Packaged Software Development“, *IEEE Transactions on Engineering Management*, **42**(1), 1995, 50-60.

Chatzoglou, P. D., „Factors Affecting Completion of the Requirements Capture Stage of Projects with Different Characteristics,” *Information and Software Technology*, **39(9)**, 1997, 627–640.

Christie, A. M. und Staley, M. J., „Organizational and Social Simulation of a Requirements Development Process” *Software Process: Improvement and Practice*, **5(2-3)**, 2000, 103–110.

Cockburn, A., *Writing Effective Use Cases*, Addison-Wesley, Boston, 2001.

Cockburn, A., *Agile Software Development*, Addison-Wesley, Boston, 2002.

Cockburn, A. und Highsmith, L., „Agile Software Development: The People Factor“, *IEEE Computer*, **34(11)**, 2001, 131-133.

Cohn, Mike, *User Stories Applied*, Addison-Wesley, Boston, 2004.

Coyle RG, *System Dynamics Modelling: A Practical Approach*, Chapman &Hall, London, 1996.

Curtis, B., Krasner, H. und Iscoe, N., „A Field Study of the Software Design Process for Large Systems”, *Communications of the ACM*, **31(11)**, 1988, 1268-1287.

Dahlstedt, Å. G., Karlsson, L., Persson, A., Natt och Dag, J. und Regnell, B. „Market-Driven Requirements Engineering Processes for Software Products: a Report on Current Practices.” Proceedings of the International Workshop on COTS and Product Software, Monterey Bay, 2003.

Daly, E., „Management of Software Development“, *IEEE Transaction on Software Engineering*, **3(3)**, 1977, 229-242.

Davies, R., *Agile Requirements*, 2005,
<http://www.methodsandtools.com/archive/archive.php?id=27> (16.12.2009).

Davis, A. M., *Software Requirements: Objects, Functions and States*, Prentice Hall, Englewood Cliffs, 1993.

Davis, A. M., „The Art of Requirements Triage”, *IEEE Computer*, **36(3)**, 2003, 42-49.

Deifel, B., „A Process Model for Requirements Engineering of CCOTS“, Proceedings of the Tenth International Workshop on Database and Expert Systems Applications, Florence, 1999, 316-320.

Dörner, D., *The Logic of Failure*, Addison-Wesley, Boston, 1996.

Emam, K. E. und Madhavji, N. H., „A Field Study of Requirements Engineering Practices in Information Systems Development“, Second IEEE International Symposium on Requirements Engineering, IEEE CS Press, Los Alamitos, 1995, 68–80.

Fagan, M., „Design and Code Inspections and Process Control in the Development of Programs“, IBM Report IBM-SDD-TR-21-572, 1974.

Fagan, M. E., „Advances in Software Inspections“, *IEEE Transactions on Software Engineering*, **12**(7), 1986, 744–751.

Ferreira, S., Collofello, J., Shunk, D., Mackulak, G. und Wolfe, P. „Utilization of Process Modeling and Simulation in Understanding the Effects of Requirements Volatility in Software Development“, *Proceedings of the Forth Software Process Simulation Modeling Workshop*, Portland, 2003,

Forrester, J. W., *Industrial Dynamics*, Productivity Press, Cambridge, 1961.

Forrester, J. W., *Principles of Systems*, Productivity Press, Cambridge, 1971.

Freeman, D. P. und Weinberg, G. M., *Handbook of Walkthroughs, Inspections and Technical Reviews*, Dorset House Publishing, New York, 1990.

Fritzsche, M., „Agile Methods and Requirements Engineering in Change Intensive Projects“, Proceedings of the 3rd International Conference on Evaluation of Novel Approaches to Software Engineering, Funchal, 2008, 81-88.

Fritzsche, M. und Broy, M.: „Criteria for the Evaluation of Requirements Engineering Methods in a Change Intensive Environment“, Technical Report TUM-I0829, Technische Universität München, 2008.

Fritzsche, M. und Keil, P., „Agile Methods and CMMI: Compatibility or Conflict?“, *e-Informatica Software Engineering Journal*, **1**(1), 2007a, 9-26.

Fritzsche, M. und Keil, P., „Agilität und Prozessreife: Erfüllbarkeit der CMMI-Prozessgebiete durch agile Methoden am Beispiel von XP“, in: Bleek, W.-G., Raasch, J. und Züllighoven, H., *Software Engineering 2007: Fachtagung des GI-Fachbereichs Softwaretechnik*, Hamburg, 2007b, 26-30.

Gause, D. C. und Weinberg, G.M., *Exploring Requirements – Quality before Design*, Dorset House, New York, 1989.

Geisberger, E., Broy, M., Berenbach, B., Kazmeier, J., Paulish, D. und Rudorfer, A., „Requirements Engineering Reference Model (REM)“, Technical Report TUM-I0618, Technische Universität München, 2006.

Glinz, M., „On Non-Functional Requirements“, Proceedings of the 15th International Requirements Engineering Conference, 2007, 21-26.

Goetz, R., „How Agile Processes Can Help in Time-Constrained Requirements Engineering“, International Workshop on Time-Constrained Requirements Engineering, Essen, 2002.

Gottesdiener, E., *Requirements by Collaboration*, Addison-Wesley, Boston, 2002.

Graham, I., *Requirements Engineering and Rapid Development*, Addison-Wesley, Harlow, 1998.

Hall, T., Beecham, S. und Rainer, A., „Requirements Problems in Twelve Software Companies: An Empirical Analysis“, Proceedings of the Conference on Empirical Assessment in Software Engineering, 2002, 153-160.

Hauser, J. R. und Clausing, D., „The House of Quality“, *Harvard Business Review*, May-June, 1988, pp. 63–73.

Highsmith, J. A., *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*, Dorset House Publishing, New York, 2000.

Higuera, R. P. und Haimes, Y. Y., „Software Risk Management“, Technical Report CMU/SEI-96-TR-012 ESC-TR-96-012, 1996.

Hofmann, H. F. und Lehner, F., „Requirements Engineering as a Success Factor in Software Projects“, *IEEE Software*, **18**(4), 2001, 58-66.

Höst, M., Regnell B., Natt och Dag, J., Nedstam, J. und Nyberg, C., „Exploring Bottlenecks in Market-Driven Requirements Management Processes with Discrete Event Simulation“, *Journal of Systems and Software*, **59**(3), 2001, 323–332.

Houston, D. X., „A Software Project Simulation Model for Risk Management“, Dissertation, Arizona State University, 2000.

Houston D. X., Mackulak, G. T. und Collofello, J. S., „Stochastic Simulation of Risk Factor Potential Effects for Software Development Risk Management“, *Journal of Systems and Software*, **59**(3), 2001, 247–257.

IEEE Standard 830-1998, „IEEE Recommended Practice for Software Requirements Specifications“, 1998a.

IEEE Standard 1233-1998, „IEEE Guide for Developing of System Requirements Specifications“, 1998b.

ISO/IEC Standard No. 9126, „Software Engineering – Product Quality; Parts 1–4“, International Organization for Standardization, 2001.

Jackson, M., *Software Requirements & Specifications*, ACM Press, Wokingham, 1995.

Jarke, M., Jacobs, S. und Pohl, K., *Group-Decision Support und Qualitätsmanagement*, Erich Schmidt Verlag, Berlin, 1992.

Jones, C., *Assessment and Control of Software Risks*, Yourdon Press Computing Series, Upper Saddle River, 1994.

Jones, C., *Estimating Software Costs*, McGraw-Hill, New York, 1996a.

Jones, C., „Strategies for Managing Requirements Creep“, *IEEE Computer*, **29**(6), 1996b, 92-94.

Känsälä, K., „Integrating Risk Assessment with Cost Estimation“, *IEEE Software*, **14**(3), 1997, 61-67.

Karlsson, J. und Ryan, K., „A Cost-Value Approach for Prioritizing Requirements“, *IEEE Software*, **14**(5), 1997, 67-75.

Karlsson, J., Wohlin, C. und Regnell, B., „An Evaluation of Methods for Prioritizing Software Requirements“, *Information and Software Technology*, **39**(14–15), 1998, 939–947.

Karlsson, L., Berander, P., Regnell, B. und Wohlin, C., „Requirements Prioritisation: An Experiment on Exhaustive Pair-Wise Comparisons versus Planning Game Partitioning“, *Proceedings of Empirical Assessment in Software Engineering*, Edinburgh, 2004, 145–154.

Karlsson, L., Dahlstedt, Å.G., Natt och Dag, J., Regnell, B. und Persson, A., „Challenges in Market-Driven Requirements Engineering: an Industrial Interview Study“, *International Workshop on Requirements Engineering: Foundations of Software Quality*, Essen, 2002, 37–49.

Keil, M. und Carmel, E., „Customer-Developer Links in Software Development“, *Communication of the ACM*, **38**(5), 1995, 33-44.

Kellner, M. I., Madachy, R. J. und Raffo, D. M., „Software Process Simulation Modeling: Why? What? How?“, *Journal of Systems and Software*, **46**(2-3), 1999, 91-105.

Kotonya, G. und Sommerville, I., „Requirements Engineering with Viewpoints“, *BCS/IEE Software Engineering Journal*, **11**(1), 1996, 5-18.

Kotonya, G. und Sommerville, I., *Requirements Engineering – Processes and Techniques*, Wiley, Chichester, 1998.

Kuppuswami, S., Vivekanandan, K. und Rodrigues, P., „A System Dynamics Simulation Model to Find the Effects of XP on Cost of Change Curve“, *Proceedings of the Fourth International Conference on Extreme Programming and Agile Processes in Software Engineering*, Springer, Berlin, 2003, 54–62.

Van Lamsweerde, A., „Goal-Oriented Requirements Engineering: A Guided Tour“, *Proceedings of the Fifth International Symposium on Requirements Engineering*, Toronto, 2001, 249-261.

Van Lamsweerde, A., *Requirements Engineering*, Wiley, Chichester, 2009.

Larman, C., *Agile & Iterative Development*, Addison Wesley, Boston, 2004.

Lee, M., „Just-in-Time Requirements Analysis: The Engine that Drives the Planning Game“, 2002, <http://www.agilealliance.org>.

Lee, B. und Miller, J., „Multi-Project Management in Software Engineering Using Simulation Modeling“, *Software Quality Journal*, **12**(1), 2004, 59–82.

Lehman, M. M., „Programs, Life Cycles, and Laws of Software Evolution“, Proceedings of the IEEE, **68**(9), 1980, 1060–1076.

Lerch, F. J., Ballou, D. J. und Harter, D. E., „Using Simulation-Based Experiments for Software Requirements Engineering“, *Annals of Software Engineering*, **3**, 1997, 345–366.

Lin, C. Y., Abdel-Hamid, T. K. und Sherif, J., „Software-Engineering Process Simulation Model (SEPS)“, *Journal of Systems and Software*, **38**(3), 1997, 263–277.

Lubars, M., Potts, C. und Richter, C., „A Review of the State of the Practice in Requirements Modelling“, Proceedings of the First IEEE International Symposium on Requirements Engineering, 1993, 2-14.

Macaulay, L. A., *Requirements Engineering*, Springer, London, 1996.

Martin, J., *Rapid Application Development*, Macmillan Publishing, Indianapolis, 1991.

Martin, R. und Raffo, D., „Application of a Hybrid Process Simulation Model to a Software Development Project“, *The Journal of Systems and Software*, **59**, 2001, 237–246.

McPhee, C. und Eberlein, A., „An Approach to Evaluating Requirements Engineering Methods for Applicability to Time-to-market Projects“, Proceedings of the 13th International Conference on Software and Systems Engineering and their Applications, Paris, 2000.

McPhee, C. und Eberlein, A., „Requirements Engineering for Time-to-Market Projects“, Proceedings of the 9th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, Lund, 2002, 17-24.

Mišić, V. B., Gevaert, H. und Rennie, M., „Extreme Dynamics: Towards a System Dynamics Model of the Extreme Programming Software Development Process“, Proceedings of the Fifth International Workshop on Software Process Simulation Modeling, Edinburgh, 2004, 237–242.

Moynihan, T., „How Experience Project Managers Assess Risk“, *IEEE Software*, **14**(3), 1997, 35-41.

Müller, M. und Pfahl, D., „Simulation Methods“, in: Singer, J., Shull, F. und Sjøberg, D., *Advanced Topics in Empirical Software Engineering: A Handbook*, Springer, Berlin, 2008, 117-152.

Neill, C. J. und Laplante, P. A., „Requirements Engineering: The State of the Practice“, *IEEE Software*, **20**(6), 2003.

Neu, H., Hanne, T., Münch, J., Nickel, S. und Wirsen, A., „Simulation-Based Risk Reduction for Planning Inspections“, *Proceedings 4th International Conference on Product Focused Software Process Improvement*, Springer, Berlin, 2002, 78–93.

Nikula, U., Sajaniemi, J. und Kaelviaeinen, H., „A State-of-the-Practice Survey on Requirements Engineering in Small- and Medium-Sized Enterprises“, Research Report, Lappeenranta University of Technology, 2000.

Novorita, R. und Grube, G., „Benefits of Structured Requirements Methods for Market-Based Enterprises“, *Proceeding of the Sixth Annual International Symposium on Systems Engineering: Practices and Tools*, Boston, 1996.

Nurmuliani, N., Zowghi, D., und Powell, S., „Analysis of Requirements Volatility During Software Development Life Cycle“, *Proceedings of the 2004 Australian Software Engineering Conference*, 2004, 28–37.

Nurmuliani, N., Zowghi, D. und Williams, S. P., „Requirements Volatility and its Impact on Change Effort: Evidence-based Research in Software Development Projects“, *Proceedings of the Eleventh Australian Workshop on Requirements Engineering*, Adelaide, 2006.

Nuseibeh, B. und Easterbrook, S., „Requirements Engineering: A Roadmap“, *Proceedings of the 22nd International Conference on Software Engineering*, Limerick, 2000, 35-46.

Padberg, F., „A Study on Optimal Scheduling for Software Projects“, *Software Process Improvement and Practice*, **11**(1), 2006, 77–91.

Paetsch, F., Eberlein, A. und Maurer, F., „Requirements Engineering and Agile Software Development“, *12th IEEE International Workshops on Enabling Technologies*, Linz, 2003, 308-313.

Paragon Software, „Requirements Document for the VirtualArtViewer“, 2002.

Pfahl, D., „An Integrated Approach to Simulation-Based Learning in Support of Strategic and Project Management in Software Organisations“, *PhD Theses in Experimental Software Engineering*, **8**, 2001.

Pfahl, D., „ProSim/RA: Software Process Simulation in Support of Risk Assessment“, in: Biffel, S., Aurum, A., Boehm, B., Erdogmus, H. und Grünbacher, P., *Value-based Software Engineering*, Springer Press, Berlin, 2005, 263–286.

Pfahl, D. und Lebsanft, K., „Using Simulation to Analyse the Impact of Software Requirement Volatility on Project Performance“, *Information and Software Technology* **42**(14), 2000, 1001-1008.

Pfahl, D. und Ruhe, G., „IMMoS: A Methodology for Integrated Measurement, Modelling, and Simulation“, *International Journal of Software Process: Improvement and Practice*, **7**, 2002, 189-210.

Pinheiro, F. A. C., „Requirements Honesty“, International Workshop on Time-Constrained Requirements Engineering, Essen, 2002.

Pohl, K., „Requirements Engineering: An Overview“, Technical Report TR 96/2, CREWS, 1996.

Pohl, K., *Requirements Engineering*, Dpunkt Verlag, Heidelberg, 2007.

Potts, C., „Invented Requirements and Imagined Customers: Requirements Engineering for Off-the- Shelf Software“, Proceedings of the Second IEEE International Symposium on Requirements Engineering, IEEE Computer Society Press, 1995, 128-130.

Potts, C., Takahashi, K. und Anton, A. I., „Inquiry-Based Requirements Analysis“, *IEEE Software*, **11**(2), 1994, 21-32.

Powell, A., Mander, K. und Brown, D., „Strategies for Lifecycle Concurrency and Iteration: A System Dynamics Approach“, *Journal of Systems and Software*, **46**(2-3), 1999, 151–162.

Randers, J., „Conceptualizing Dynamic Models of Social Systems: Lessons from a Study of Social Change“, Dissertation, Massachusetts Institute of Technology, 1973.

Randers, J., *Elements of the System Dynamics Method*, Productivity Press, Cambridge, 1980.

Regnell, B., Beremark, P. und Eklundh, O., „A Market-Driven Requirements Engineering Process: Results from an Industrial Process Improvement Programme“, *Requirements Engineering*, **3**, 1998, 121–129.

Reifer, D. J. „Requirements Management: The Search for Nirvana“, *IEEE Software*, **17**(3), 2000, 45-47.

Richardson, G. P. und Pugh, G. L., *Introduction to System Dynamics Modeling and Dynamo*, MIT Press, Cambridge, 1981.

Roberts, E. B., „Research and Development Policy Making“, *Technology Review*, **66**(8), 1964, 3-7.

Robertson, S., „Simulation Model Verification and Validation: Increase the Users' Confidence“, Proceedings of the 1997 Winter Simulation Conference, 1997, 53–59.

Robertson, S. und Robertson, J., *Mastering the Requirements Process*, ACM Press, Harlow, 1999.

Rolland, C., Ben Achour, C., Cauvet, C., Ralyte, J., Sutcliffe, A., Maiden, N., Jarke, M., Haumer, P., Pohl, K., Dubois, E., und Heymas, P., „A Proposal for Scenario Classification Framework“, *Requirements Engineering Journal*, **3**(1), 1998, 23-47.

Ropponen, J., *Beyond the IT Productivity Paradox*, Wiley, New York, 1999, 247-266.

Ropponen, J. und Lyytinen, K., „Components of Software Development Risk: How to Address Them? A Project Manager Survey“, *IEEE Transactions on Software Engineering*, **26**(2), 2000, 98-112.

Rupp, C., *Requirements-Engineering und –Management*, Hanser, München, 2007.

Saaty, T., *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*, McGraw-Hill, New York, 1980.

Sawyer, P., „Packaged Software: Challenges for RE“, Proceedings of the Sixth International Workshop on Requirements Engineering: Foundation for Software Quality, Stockholm, 2000, 137-142.

Sawyer, P., Sommerville, I. und Kotonya, G., „Improving Market-driven RE Processes“, Proceedings of the International Conference on Product Focused Software Process Improvement, Oulu, 1999, 222-236.

Schmidt, R., Lyytinen, K., Keil, M. und Culle, P., „Identifying Software Project Risks: An International Delphi Study“, *Journal of Management Information Systems*, **17**(4), 2001, 5-36.

Schwaber, K., „Scrum Development Process“, *OOPSLA '95 Workshop on Business Object Design and Implementation*, Austin, 1995.

Schwaber, K., „Agile Processes: Emergence of Essential Systems“, The Agile Alliance, <http://www.agilealliance.org/articles>, 2002.

Schwaber, K., *Agile Project Management with Scrum*, Microsoft Press, Redmond, 2004.

Schwaber, K. und Beedle, M., *Agile Software Development with Scrum*, Prentice-Hall, Upper Saddle River, 2002.

Sharp, H., Finkelstein, A. und Galal, G., „Stakeholder Identification in the Requirements Engineering Process“, Workshop on Requirements Engineering Processes, Florence, 1999, 387-391.

Shore, J., „Beyond Story Cards: Agile Requirements Collaboration“, 2005, <http://www.jamesshore.com>.

Sommerville, I. und Sawyer, P., *Requirements Engineering: A Good Practice Guide*, Wiley, Chichester, 1997.

Stallinger, F. und Grünbacher, P., „System Dynamics Modelling and Simulation of Collaborative Requirements Engineering“, *The Journal of Systems and Software*, **59**(3), 2001, 311–321.

Standish Group, The Chaos Report, 1995, <http://www.standishgroup.com>.

Stapleton, J., *Dynamic Systems Development Method: The Method in Practice*, Addison-Wesley, Boston, 1997.

Stark, G., Skillicorn, A. und Ameele, R., „An Examination of the Effects of Requirements Changes on Software Releases“, *Journal of Software Maintenance*, **11**(5), 1999, 293-309.

Strens, M. R. und Sugden, R. C., „Criteria for the Assessment of Representation Methods as Vehicles for Handling Change“, Proceedings of the International Symposium and Workshop on Systems Engineering of Computer Based Systems, Tucson, 1995, 233-239.

Strens, M. R. und Sugden, R. C., „Change Analysis: A Step Towards Meeting the Challenge of Changing Requirements“, Proceedings of the IEEE Symposium and Workshop on the Engineering of Computer-Based Systems, Friedrichshafen, 1996, 278-283.

Sugden, R. C. und Strens, M. R., „Strategies, Tactics and Methods For Handling Change“, Proceedings of the IEEE Symposium and Workshop on the Engineering of Computer-Based Systems, Friedrichshafen, 1996, 457-462.

Tank-Nielsen, C., „Sensitivity Analysis in System Dynamics“, in Randers, J., *Elements of the system dynamics method*, Productivity Press, Cambridge, 1980, 187-204.

Thayer, R. und Dorfman, M., *Software Requirements Engineering*, IEEE Computer Society Press, Los Alamitos, 1997.

Tomayko, J. E., „Engineering of Unstable Requirements Using Agile Methods“, International Workshop on Time-Constrained Requirements Engineering, Essen, 2002.

Ventana Simulation Environment (Vensim®): Reference Manual, Ventana Systems, Inc., 1997.

Walia, G. S. und Carver, J. C., „A Systematic Literature Review to Identify and Classify Software Requirements Errors“, *Information and Software Technology*, **51**(7), 2009, 1087-1109.

Walia, G. S., Carver, J. und Philip, T., „Requirement Error Abstraction and Classification: An Empirical Study“, Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering, Rio de Janeiro, 2006, 336-345.

Wan-Kadir, W. M. N. und Loucopoulos, P., „Relating evolving business rules to software design“, *Journal of Systems Architecture*, **50**, 2004, 367-382.

Wieggers, K. E., „First Things First: Prioritizing Requirements“. *Software Development*, **7**(9), 1999, 48-53.

Wieggers, K. E., *Software Requirements*, Microsoft Press, Redmond, 2005.

Wood, J. und Silver, D., *Joint Application Development*, Wiley, New York, 1995.

Zowghi, D. und Nurmuliani, N., „A Study of the Impact of Requirements Volatility on Software Project Performance“, Proceedings of the Ninth Asia-Pacific Software Engineering Conference, Gold Coast, 2002, 3- 11.

Anhang A. System Dynamics Modell FAUSER AG

Im Folgenden führen wir die Gleichungen auf, die das in der FAUSER-Fallstudie erstellte Modell vollständig beschreiben. Die verwendete Notation entstammt dem System Dynamics Modellierungstool Vensim 3.0 (Ventana Systems 1997).

Level

Analysierte nicht stabilisierbare Anforderungen =

INTEG (Nicht stabilisierbare Anforderungen analysieren - Analysierte nicht stabilisierbare Anforderungen ändern sich - Nicht stabilisierbare Anforderungen realisieren, 0)

~ Person * Day

Analysierte stabile Anforderungen =

INTEG (Stabile Anforderungen analysieren - Stabile Anforderungen realisieren, 0)

~ Person * Day

Analysierte stabilisierbare Anforderungen =

INTEG (Stabilisierbare Anforderungen analysieren - Stabilisierbare Anforderungen realisieren - Analysierte stabilisierbare Anforderungen ändern sich, 0)

~ Person * Day

Analysierte zu ändernde nicht stabilisierbare Anforderungen =

INTEG (Zu ändernde nicht stabilisierbare Anforderungen analysieren - Analysierte zu ändernde nicht stabilisierbare Anforderungen ändern sich - Zu ändernde nicht stabilisierbare Anforderungen realisieren, 0)

~ Person * Day

Analysierte zu ändernde stabilisierbare Anforderungen =

INTEG (Zu ändernde stabilisierbare Anforderungen analysieren - Analysierte zu ändernde stabilisierbare Anforderungen ändern sich - Zu ändernde stabilisierbare Anforderungen realisieren, 0)

~ Person * Day

Aufwand Anforderungsanalyse = INTEG (Anforderungen analysieren, 0)

~ Person * Day

Aufwand Entwicklung = INTEG (Anforderungen realisieren, 0)

~ Person * Day

Nicht stabilisierbare Anforderungen =

INTEG (Analysierte nicht stabilisierbare Anforderungen ändern sich - Nicht stabilisierbare Anforderungen analysieren, Zu bearbeitende Anforderungen * Anteil nicht stabilisierbare Anforderungen)

~ Person * Day

Realisierte nicht stabilisierbare Anforderungen =
INTEG (Nicht stabilisierbare Anforderungen realisieren + Zu ändernde nicht stabilisierbare Anforderungen realisieren - Realisierte nicht stabilisierbare Anforderungen ändern sich, 0)

~ Person * Day

Realisierte stabile Anforderungen = INTEG (Stabile Anforderungen realisieren, 0)

~ Person * Day

Realisierte stabilisierbare Anforderungen =
INTEG (Stabilisierbare Anforderungen realisieren + Zu ändernde stabilisierbare Anforderungen realisieren - Realisierte stabilisierbare Anforderungen ändern sich, 0)

~ Person * Day

Stabile Anforderungen =
INTEG (-Stabile Anforderungen analysieren, Zu bearbeitende Anforderungen * Anteil stabile Anforderungen)

~ Person * Day

Stabilisierbare Anforderungen =
INTEG (Analysierte stabilisierbare Anforderungen ändern sich - Stabilisierbare Anforderungen analysieren, Zu bearbeitende Anforderungen * Anteil stabilisierbare Anforderungen)

~ Person * Day

Zu ändernde nicht stabilisierbare Anforderungen =
INTEG (Analysierte zu ändernde nicht stabilisierbare Anforderungen ändern sich + Realisierte nicht stabilisierbare Anforderungen ändern sich - Zu ändernde nicht stabilisierbare Anforderungen analysieren, 0)

~ Person * Day

Zu ändernde stabilisierbare Anforderungen =
INTEG (Analysierte zu ändernde stabilisierbare Anforderungen ändern sich + Realisierte stabilisierbare Anforderungen ändern sich - Zu ändernde stabilisierbare Anforderungen analysieren, 0)

~ Person * Day

Raten

Analysierte nicht stabilisierbare Anforderungen ändern sich =
Analysierte nicht stabilisierbare Anforderungen * Änderungsrate nicht stabilisierbare Anforderungen

~ Person

Analysierte stabilisierbare Anforderungen ändern sich =
Analysierte stabilisierbare Anforderungen * Änderungsrate analysierte stabilisierbare Anforderungen

~ Person

Analysierte zu ändernde nicht stabilisierbare Anforderungen ändern sich =
Änderungsrate nicht stabilisierbare Anforderungen * Analysierte zu
ändernde nicht stabilisierbare Anforderungen

~ Person

Analysierte zu ändernde stabilisierbare Anforderungen ändern sich =
Änderungsrate realisierte stabilisierbare Anforderungen * Analysierte zu
ändernde stabilisierbare Anforderungen

~ Person

Anforderungen analysieren =
(Nicht stabilisierbare Anforderungen analysieren * Aufwand Analyse nicht
stabilisierbare Anforderungen) + (Stabile Anforderungen analysieren *
Aufwand Analyse stabile Anforderungen) + (Stabilisierbare Anforderungen
analysieren * Aufwand Analyse stabilisierbare Anforderungen) + (Zu
ändernde nicht stabilisierbare Anforderungen analysieren * Aufwand
Analyse nicht stabilisierbare Anforderungen) + (Zu ändernde stabilisierbare
Anforderungen analysieren * Aufwand Analyse stabilisierbare
Anforderungen)

~ Person

Anforderungen realisieren =
(Nicht stabilisierbare Anforderungen realisieren + Stabile Anforderungen
realisieren + Stabilisierbare Anforderungen realisieren + (Zu ändernde nicht
stabilisierbare Anforderungen realisieren * Aufwandsmultiplikator geänderte
realisierte Anforderungen) + (Zu ändernde stabilisierbare Anforderungen
realisieren * Aufwandsmultiplikator geänderte realisierte Anforderungen))

~ Person

Nicht stabilisierbare Anforderungen analysieren =
Analyserate nicht stabilisierbare Anforderungen

~ Person

Nicht stabilisierbare Anforderungen realisieren =
Realisierungsrate nicht stabilisierbare Anforderungen

~ Person

Realisierte nicht stabilisierbare Anforderungen ändern sich =
Realisierte nicht stabilisierbare Anforderungen * Änderungsrate nicht
stabilisierbare Anforderungen

~ Person

Realisierte stabilisierbare Anforderungen ändern sich =
Realisierte stabilisierbare Anforderungen * Änderungsrate realisierte
stabilisierbare Anforderungen

~ Person

Stabile Anforderungen analysieren = Analyserate stabile Anforderungen

~ Person

Stabile Anforderungen realisieren = Realisierungsrate stabile Anforderungen

~ Person

Stabilisierbare Anforderungen analysieren =

Analyserate stabilisierbare Anforderungen

~ Person

Stabilisierbare Anforderungen realisieren =

Realisierungsrate stabilisierbare Anforderungen

~ Person

Zu ändernde nicht stabilisierbare Anforderungen analysieren =

Analyserate zu ändernde nicht stabilisierbare Anforderungen

~ Person

Zu ändernde nicht stabilisierbare Anforderungen realisieren =

Realisierungsrate zu ändernde nicht stabilisierbare Anforderungen

~ Person

Zu ändernde stabilisierbare Anforderungen analysieren =

Analyserate zu ändernde stabilisierbare Anforderungen

~ Person

Zu ändernde stabilisierbare Anforderungen realisieren =

Realisierungsrate zu ändernde stabilisierbare Anforderungen

~ Person

Hilfsvariablen

Analyserate nicht stabilisierbare Anforderungen =

IF THEN ELSE (Anforderungsanalyse läuft flag, MIN (Nicht stabilisierbare Anforderungen / Konstante Day, ZIDZ (Nicht stabilisierbare Anforderungen, Nicht stabilisierbare Anforderungen + Stabile Anforderungen + Stabilisierbare Anforderungen + Zu ändernde nicht stabilisierbare Anforderungen + Zu ändernde stabilisierbare Anforderungen) * („RE-Team“ + IF THEN ELSE (Entwicklung läuft flag, „RE-Entwicklereinbindung während Entwicklung“ * Entwickler, Entwickler)) * Zeitliche Produktivität Analyse / Aufwand Analyse nicht stabilisierbare Anforderungen), 0)

~ Person

Analyserate stabile Anforderungen =

IF THEN ELSE (Anforderungsanalyse läuft flag, MIN (Stabile Anforderungen / Konstante Day, ZIDZ (Stabile Anforderungen, Nicht stabilisierbare Anforderungen + Stabile Anforderungen + Stabilisierbare Anforderungen + Zu ändernde nicht stabilisierbare Anforderungen + Zu ändernde stabilisierbare Anforderungen) * („RE-Team“ + IF THEN ELSE (Entwicklung läuft flag, „RE-Entwicklereinbindung während Entwicklung“ * Entwickler, Entwickler)) * Zeitliche Produktivität Analyse / Aufwand Analyse stabile Anforderungen), 0)

~ Person

Analyserate stabilisierbare Anforderungen =

IF THEN ELSE (Anforderungsanalyse läuft flag, MIN (Stabilisierbare Anforderungen / Konstante Day, ZIDZ (Stabilisierbare Anforderungen, Nicht stabilisierbare Anforderungen + Stabile Anforderungen + Stabilisierbare Anforderungen + Zu ändernde nicht stabilisierbare Anforderungen + Zu ändernde stabilisierbare Anforderungen) * („RE-Team“ + IF THEN ELSE (Entwicklung läuft flag, „RE-Entwicklereinbindung während Entwicklung“ * Entwickler, Entwickler)) * Zeitliche Produktivität Analyse / Aufwand Analyse stabilisierbare Anforderungen), 0)

~ Person

Analyserate zu ändernde nicht stabilisierbare Anforderungen =

IF THEN ELSE (Anforderungsanalyse läuft flag, MIN (Zu ändernde nicht stabilisierbare Anforderungen / Konstante Day, ZIDZ (Zu ändernde nicht stabilisierbare Anforderungen, Nicht stabilisierbare Anforderungen + Stabile Anforderungen + Stabilisierbare Anforderungen + Zu ändernde nicht stabilisierbare Anforderungen + Zu ändernde stabilisierbare Anforderungen) * („RE-Team“ + IF THEN ELSE (Entwicklung läuft flag, „RE-Entwicklereinbindung während Entwicklung“ * Entwickler, Entwickler)) * Zeitliche Produktivität Analyse / Aufwand Analyse nicht stabilisierbare Anforderungen), 0)

~ Person

Analyserate zu ändernde stabilisierbare Anforderungen =

IF THEN ELSE (Anforderungsanalyse läuft flag, MIN (Zu ändernde stabilisierbare Anforderungen / Konstante Day, ZIDZ (Zu ändernde stabilisierbare Anforderungen, Nicht stabilisierbare Anforderungen + Stabile Anforderungen + Stabilisierbare Anforderungen + Zu ändernde nicht stabilisierbare Anforderungen + Zu ändernde stabilisierbare Anforderungen) * („RE-Team“ + IF THEN ELSE (Entwicklung läuft flag, „RE-Entwicklereinbindung während Entwicklung“ * Entwickler, Entwickler)) * Zeitliche Produktivität Analyse / Aufwand Analyse stabilisierbare Anforderungen), 0)

~ Person

Änderungsrate analysierte stabilisierbare Anforderungen =

Änderungsrate stabilisierbare Anforderung lookup (Verständnis analysierte stabilisierbare Anforderungen)

~ 1 / Day

Änderungsrate realisierte stabilisierbare Anforderungen =

Änderungsrate stabilisierbare Anforderung lookup (Verständnis realisierte stabilisierbare Anforderungen)

~ 1 / Day

Anforderungsanalyse läuft flag =

IF THEN ELSE ((Analyisierte stabilisierbare Anforderungen + Analyisierte zu ändernde stabilisierbare Anforderungen + Analyisierte nicht stabilisierbare Anforderungen + Analyisierte zu ändernde nicht stabilisierbare Anforderungen + Analyisierte stabile Anforderungen) < 1, 1, IF THEN ELSE ((Stabilisierbare Anforderungen + Zu ändernde stabilisierbare Anforderungen + Nicht stabilisierbare Anforderungen + Zu ändernde nicht stabilisierbare Anforderungen + Stabile Anforderungen) > 1, Analysephasen, 0))

~ Dmnl

Entwicklung läuft flag =

IF THEN ELSE (Anforderungsanalyse läuft flag < 1, 1, Entwicklungsphasen)

~ Dmnl

Gesamtprojektkosten = Aufwand Anforderungsanalyse + Aufwand Entwicklung

~ Person * Day

Output Änderungsrate stabilisierbare Anforderungen =

ZIDZ ((Änderungsrate stabilisierbare Anforderung lookup (0) * Stabilisierbare Anforderungen + Änderungsrate analysierte stabilisierbare Anforderungen * Analyisierte stabilisierbare Anforderungen + Änderungsrate realisierte stabilisierbare Anforderungen * (Zu ändernde stabilisierbare Anforderungen + Realisierte stabilisierbare Anforderungen + Analyisierte zu ändernde stabilisierbare Anforderungen)), (Anteil stabilisierbare Anforderungen * Zu bearbeitende Anforderungen))

~ 1 / Day

Output Anforderungen werden analysiert =

Nicht stabilisierbare Anforderungen analysieren + Stabile Anforderungen analysieren + Stabilisierbare Anforderungen analysieren + Zu ändernde nicht stabilisierbare Anforderungen analysieren + Zu ändernde stabilisierbare Anforderungen analysieren

~ Person

Output Realisierte Anforderungen =

Realisierte nicht stabilisierbare Anforderungen + Realisierte stabile Anforderungen + Realisierte stabilisierbare Anforderungen

~ Person * Day

Output Verständnis stabilisierbare Anforderungen =

ZIDZ (Verständnis analysierte stabilisierbare Anforderungen * Analyisierte stabilisierbare Anforderungen + (Verständnis realisierte stabilisierbare Anforderungen * (Realisierte stabilisierbare Anforderungen + Zu ändernde stabilisierbare Anforderungen + Analyisierte zu ändernde stabilisierbare Anforderungen)), (Analyisierte stabilisierbare Anforderungen + Analyisierte zu ändernde stabilisierbare Anforderungen + Realisierte stabilisierbare Anforderungen + Zu ändernde stabilisierbare Anforderungen))

~ Dmnl

Realisierungsrate nicht stabilisierbare Anforderungen =

IF THEN ELSE (Entwicklung läuft flag, MIN (Analysierte nicht stabilisierbare Anforderungen / Konstante Day, ZIDZ (Analysierte nicht stabilisierbare Anforderungen, Analysierte nicht stabilisierbare Anforderungen + Analysierte stabile Anforderungen + Analysierte stabilisierbare Anforderungen + Analysierte zu ändernde nicht stabilisierbare Anforderungen + Analysierte zu ändernde stabilisierbare Anforderungen) * IF THEN ELSE (Anforderungsanalyse läuft flag, (1 – „RE-Entwicklereinbindung während Entwicklung“) * Entwickler, Entwickler)), 0)

~ Person

Realisierungsrate stabile Anforderungen =

IF THEN ELSE (Entwicklung läuft flag, MIN (Analysierte stabile Anforderungen / Konstante Day, ZIDZ (Analysierte stabile Anforderungen, Analysierte nicht stabilisierbare Anforderungen + Analysierte stabile Anforderungen + Analysierte stabilisierbare Anforderungen + Analysierte zu ändernde nicht stabilisierbare Anforderungen + Analysierte zu ändernde stabilisierbare Anforderungen) * IF THEN ELSE (Anforderungsanalyse läuft flag, (1 – „RE-Entwicklereinbindung während Entwicklung“) * Entwickler, Entwickler)), 0)

~ Person

Realisierungsrate stabilisierbare Anforderungen =

IF THEN ELSE (Entwicklung läuft flag, MIN (Analysierte stabilisierbare Anforderungen / Konstante Day, ZIDZ (Analysierte stabilisierbare Anforderungen, Analysierte nicht stabilisierbare Anforderungen + Analysierte stabile Anforderungen + Analysierte stabilisierbare Anforderungen + Analysierte zu ändernde nicht stabilisierbare Anforderungen + Analysierte zu ändernde stabilisierbare Anforderungen) * IF THEN ELSE (Anforderungsanalyse läuft flag, (1 – „RE-Entwicklereinbindung während Entwicklung“) * Entwickler, Entwickler)), 0)

~ Person

Realisierungsrate zu ändernde nicht stabilisierbare Anforderungen =

IF THEN ELSE (Entwicklung läuft flag, MIN (Analysierte zu ändernde nicht stabilisierbare Anforderungen/Konstante Day, ZIDZ (Analysierte zu ändernde nicht stabilisierbare Anforderungen, Analysierte nicht stabilisierbare Anforderungen + Analysierte stabile Anforderungen + Analysierte stabilisierbare Anforderungen + Analysierte zu ändernde nicht stabilisierbare Anforderungen + Analysierte zu ändernde stabilisierbare Anforderungen) * IF THEN ELSE (Anforderungsanalyse läuft flag, ((1 – „RE-Entwicklereinbindung während Entwicklung“) * Entwickler, Entwickler) / Aufwandsmultiplikator geänderte realisierte Anforderungen), 0)

~ Person

Realisierungsrate zu ändernde stabilisierbare Anforderungen =
 IF THEN ELSE (Entwicklung läuft flag, MIN (Analysierte zu ändernde stabilisierbare Anforderungen / Konstante Day, ZIDZ (Analysierte zu ändernde stabilisierbare Anforderungen, Analysierte nicht stabilisierbare Anforderungen + Analysierte stabile Anforderungen + Analysierte stabilisierbare Anforderungen + Analysierte zu ändernde nicht stabilisierbare Anforderungen + Analysierte zu ändernde stabilisierbare Anforderungen) * IF THEN ELSE (Anforderungsanalyse läuft flag, (1 – „RE-Entwicklereinbindung während Entwicklung“) * Entwickler, Entwickler) / Aufwandsmultiplikator geänderte realisierte Anforderungen), 0)

~ Person

Verständnis analysierte stabilisierbare Anforderungen =
 „Verständnis-Multiplikator Analyse“ + ((1 – „Verständnis-Multiplikator Analyse“) * "Verständnis-Multiplikator Anteil analysierte Anforderungen")

~ Dmnl

„Verständnis-Multiplikator Analyse“ =

Effektivität Anforderungsanalyse stabilisierbare Anforderungen lookup
 (Aufwand Analyse stabilisierbare Anforderungen)

~ Dmnl

„Verständnis-Multiplikator Anteil analysierte Anforderungen“ =

Indirekter Verständnigewinn lookup ((Analysierte nicht stabilisierbare Anforderungen + Analysierte stabile Anforderungen + Analysierte stabilisierbare Anforderungen + Analysierte zu ändernde nicht stabilisierbare Anforderungen + Analysierte zu ändernde stabilisierbare Anforderungen + Realisierte nicht stabilisierbare Anforderungen + Realisierte stabile Anforderungen + Realisierte stabilisierbare Anforderungen) / Zu bearbeitende Anforderungen)

~ Dmnl

Verständnis realisierte stabilisierbare Anforderungen =

„Verständnis-Multiplikator Analyse“ + ((1 – „Verständnis-Multiplikator Analyse“) * „Verständnis-Multiplikator Anteil analysierte Anforderungen“) + ((1 – „Verständnis-Multiplikator Analyse“) * (1 – „Verständnis-Multiplikator Anteil analysierte Anforderungen“) * „Verständnis-Multiplikator Implementierung“)

~ Dmnl

Konstanten

Analysephasen = 1 + STEP (-1, 11)

~ Dmnl

Änderungsrate nicht stabilisierbare Anforderungen = 0.002

~ 1 / Day

Änderungsrate stabilisierbare Anforderung lookup

[(0, 0) - (1, 0.015)], (0, 0.011), (0.1, 0.011), (0.7, 0.006), (0.85, 0.00475))

~ 1 / Day

Anteil nicht stabilisierbare Anforderungen = 0.2

~ Dmnl

Anteil stabile Anforderungen = 0.4

~ Dmnl

Anteil stabilisierbare Anforderungen = 0.4

~ Dmnl

Aufwand Analyse nicht stabilisierbare Anforderungen = 0.2

~ Dmnl

Aufwand Analyse stabile Anforderungen = 0.2

~ Dmnl

Aufwand Analyse stabilisierbare Anforderungen = 0.2

~ Dmnl

Aufwandsmultiplikator geänderte realisierte Anforderungen = 5

~ Dmnl

Effektivität Anforderungsanalyse stabilisierbare Anforderungen lookup

([(0, 0) - (0.6, 1)], (0, 0), (0.015, 0.4), (0.04, 0.56), (0.1, 0.63), (0.2, 0.7), (0.3, 0.75), (0.4, 0.78))

~ Dmnl

Entwickler = 1

~ Person

Entwicklungsphasen = 0 + STEP (1, 11)

~ Dmnl

Indirekter Verständniskennwert lookup ([(0, 0) - (1, 0.8)], (0, 0), (0.5, 0.06), (1, 0.1))

~ Dmnl

Konstante Day = 1

~Day

„RE-Entwicklereinbindung während Entwicklung“ = 1 / 2

~ Dmnl

„RE-Team“ = 3

~ Person

„Verständnis-Multiplikator Implementierung“ = 0.05

~ Dmnl

Zeitliche Produktivität Analyse = 1 / 3

~ Dmnl

Zu bearbeitende Anforderungen = 50

~ Person * Day

Anhang B. System Dynamics Modell PSIPENTA Software Systems GmbH

Im Folgenden führen wir die Gleichungen auf, die das in der PSIPENTA-Fallstudie erstellte Modell vollständig beschreiben. Die verwendete Notation entstammt dem System Dynamics Modellierungstool Vensim 3.0 (Ventana Systems 1997).

Level

Analysierte nicht stabilisierbare Anforderungen =

INTEG (Nicht stabilisierbare Anforderungen analysieren - Analysierte nicht stabilisierbare Anforderungen ändern sich - Nicht stabilisierbare Anforderungen realisieren, 0)

~ Person * Day

Analysierte stabile Anforderungen =

INTEG (Stabile Anforderungen analysieren - Stabile Anforderungen realisieren, 0)

~ Person * Day

Analysierte stabilisierbare Anforderungen =

INTEG (Stabilisierbare Anforderungen analysieren - Stabilisierbare Anforderungen realisieren - Analysierte stabilisierbare Anforderungen ändern sich, 0)

~ Person * Day

Analysierte zu ändernde nicht stabilisierbare Anforderungen =

INTEG (Zu ändernde nicht stabilisierbare Anforderungen analysieren - Analysierte zu ändernde nicht stabilisierbare Anforderungen ändern sich - Zu ändernde nicht stabilisierbare Anforderungen realisieren, 0)

~ Person * Day

Analysierte zu ändernde stabilisierbare Anforderungen =

INTEG (Zu ändernde stabilisierbare Anforderungen analysieren - Analysierte zu ändernde stabilisierbare Anforderungen ändern sich - Zu ändernde stabilisierbare Anforderungen realisieren, 0)

~ Person * Day

Aufwand Anforderungsanalyse = INTEG (Anforderungen analysieren, 0)

~ Person * Day

Aufwand Entwicklung = INTEG (Anforderungen realisieren, 0)

~ Person * Day

Erste Iteration abgeschlossen = INTEG (Entwicklung stoppt, 0)

~ Day

Nicht stabilisierbare Anforderungen =

INTEG (Analysierte nicht stabilisierbare Anforderungen ändern sich - Nicht stabilisierbare Anforderungen analysieren, Zu bearbeitende Anforderungen * Anteil nicht stabilisierbare Anforderungen)

~ Person * Day

Realisierte nicht stabilisierbare Anforderungen =

INTEG (Nicht stabilisierbare Anforderungen realisieren + Zu ändernde nicht stabilisierbare Anforderungen realisieren - Realisierte nicht stabilisierbare Anforderungen ändern sich, 0)

~ Person * Day

Realisierte stabile Anforderungen = INTEG (Stabile Anforderungen realisieren, 0)

~ Person * Day

Realisierte stabilisierbare Anforderungen =

INTEG (Stabilisierbare Anforderungen realisieren + Zu ändernde stabilisierbare Anforderungen realisieren - Realisierte stabilisierbare Anforderungen ändern sich, 0)

~ Person * Day

Stabile Anforderungen =

INTEG (-Stabile Anforderungen analysieren, Zu bearbeitende Anforderungen * Anteil stabile Anforderungen)

~ Person * Day

Stabilisierbare Anforderungen =

INTEG (Analysierte stabilisierbare Anforderungen ändern sich - Stabilisierbare Anforderungen analysieren, Zu bearbeitende Anforderungen * Anteil stabilisierbare Anforderungen)

~ Person * Day

Zu ändernde nicht stabilisierbare Anforderungen =

INTEG (Analysierte zu ändernde nicht stabilisierbare Anforderungen ändern sich + Realisierte nicht stabilisierbare Anforderungen ändern sich - Zu ändernde nicht stabilisierbare Anforderungen analysieren, 0)

~ Person * Day

Zu ändernde stabilisierbare Anforderungen =

INTEG (Analysierte zu ändernde stabilisierbare Anforderungen ändern sich + Realisierte stabilisierbare Anforderungen ändern sich - Zu ändernde stabilisierbare Anforderungen analysieren, 0)

~ Person * Day

Raten

Analysierte nicht stabilisierbare Anforderungen ändern sich =

Analysierte nicht stabilisierbare Anforderungen * Änderungsrate nicht stabilisierbare Anforderungen

~ Person

Analysierte stabilisierbare Anforderungen ändern sich =

Analysierte stabilisierbare Anforderungen * Änderungsrate analysierte stabilisierbare Anforderungen

~ Person

Analysierte zu ändernde nicht stabilisierbare Anforderungen ändern sich =

Änderungsrate nicht stabilisierbare Anforderungen * Analysierte zu ändernde nicht stabilisierbare Anforderungen

~ Person

Analysierte zu ändernde stabilisierbare Anforderungen ändern sich =

Änderungsrate realisierte stabilisierbare Anforderungen * Analysierte zu ändernde stabilisierbare Anforderungen

~ Person

Anforderungen analysieren =

(Nicht stabilisierbare Anforderungen analysieren * Aufwand Analyse nicht stabilisierbare Anforderungen) + (Stabile Anforderungen analysieren * Aufwand Analyse stabile Anforderungen) + (Stabilisierbare Anforderungen analysieren * Aufwand Analyse stabilisierbare Anforderungen) + (Zu ändernde nicht stabilisierbare Anforderungen analysieren * Aufwand Analyse nicht stabilisierbare Anforderungen) + (Zu ändernde stabilisierbare Anforderungen analysieren * Aufwand Analyse stabilisierbare Anforderungen)

~ Person

Anforderungen realisieren =

(Nicht stabilisierbare Anforderungen realisieren + Stabile Anforderungen realisieren + Stabilisierbare Anforderungen realisieren + (Zu ändernde nicht stabilisierbare Anforderungen realisieren * Aufwandsmultiplikator geänderte realisierte Anforderungen) + (Zu ändernde stabilisierbare Anforderungen realisieren * Aufwandsmultiplikator geänderte realisierte Anforderungen))

~ Person

Entwicklung stoppt =

IF THEN ELSE ((Time > 8) :AND: ((Analyzierte stabilisierbare Anforderungen + Analyzierte zu ändernde stabilisierbare Anforderungen + Analyzierte nicht stabilisierbare Anforderungen + Analyzierte zu ändernde nicht stabilisierbare Anforderungen + Analyzierte stabile Anforderungen) < 1), 1, 0)

~ Dmnl

Nicht stabilisierbare Anforderungen analysieren =

Analyserate nicht stabilisierbare Anforderungen

~ Person

Nicht stabilisierbare Anforderungen realisieren =

Realisierungsrate nicht stabilisierbare Anforderungen

~ Person

Realisierte nicht stabilisierbare Anforderungen ändern sich =

Realisierte nicht stabilisierbare Anforderungen * Änderungsrate nicht stabilisierbare Anforderungen

~ Person

Realisierte stabilisierbare Anforderungen ändern sich =

Realisierte stabilisierbare Anforderungen * Änderungsrate realisierte stabilisierbare Anforderungen

~ Person

Stabile Anforderungen analysieren = Analyserate stabile Anforderungen

~ Person

Stabile Anforderungen realisieren = Realisierungsrate stabile Anforderungen

~ Person

Stabilisierbare Anforderungen analysieren =

Analyserate stabilisierbare Anforderungen

~ Person

Stabilisierbare Anforderungen realisieren =

Realisierungsrate stabilisierbare Anforderungen

~ Person

Zu ändernde nicht stabilisierbare Anforderungen analysieren =

Analyserate zu ändernde nicht stabilisierbare Anforderungen

~ Person

Zu ändernde nicht stabilisierbare Anforderungen realisieren =

Realisierungsrate zu ändernde nicht stabilisierbare Anforderungen

~ Person

Zu ändernde stabilisierbare Anforderungen analysieren =

Analyserate zu ändernde stabilisierbare Anforderungen

~ Person

Zu ändernde stabilisierbare Anforderungen realisieren =

Realisierungsrate zu ändernde stabilisierbare Anforderungen

~ Person

Hilfsvariablen

Analyserate nicht stabilisierbare Anforderungen =

IF THEN ELSE (Anforderungsanalyse läuft flag, MIN (Nicht stabilisierbare Anforderungen / Konstante Day, ZIDZ ((Nicht stabilisierbare Anforderungen * Aufwand Analyse nicht stabilisierbare Anforderungen), (Nicht stabilisierbare Anforderungen * Aufwand Analyse nicht stabilisierbare Anforderungen) + (Stabile Anforderungen * Aufwand Analyse stabile Anforderungen) + (Stabilisierbare Anforderungen * Aufwand Analyse stabilisierbare Anforderungen) + (Zu ändernde nicht stabilisierbare Anforderungen * Aufwand Analyse nicht stabilisierbare Anforderungen) + (Zu ändernde stabilisierbare Anforderungen * Aufwand Analyse stabilisierbare Anforderungen)) * "RE-Team" * Zeitliche Produktivität Analyse / Aufwand Analyse nicht stabilisierbare Anforderungen), 0)

~ Person

Analyserate stabile Anforderungen =

IF THEN ELSE (Anforderungsanalyse läuft flag, MIN (Stabile Anforderungen / Konstante Day, ZIDZ ((Stabile Anforderungen * Aufwand Analyse stabile Anforderungen), (Nicht stabilisierbare Anforderungen * Aufwand Analyse nicht stabilisierbare Anforderungen) + (Stabile Anforderungen * Aufwand Analyse stabile Anforderungen) + (Stabilisierbare Anforderungen * Aufwand Analyse stabilisierbare Anforderungen) + (Zu ändernde nicht stabilisierbare Anforderungen * Aufwand Analyse nicht stabilisierbare Anforderungen) + (Zu ändernde stabilisierbare Anforderungen * Aufwand Analyse stabilisierbare Anforderungen)) * "RE-Team" * Zeitliche Produktivität Analyse / Aufwand Analyse stabile Anforderungen), 0)

~ Person

Analyserate stabilisierbare Anforderungen =

IF THEN ELSE (Anforderungsanalyse läuft flag, MIN (Stabilisierbare Anforderungen / Konstante Day, ZIDZ ((Stabilisierbare Anforderungen * Aufwand Analyse stabilisierbare Anforderungen), (Nicht stabilisierbare Anforderungen * Aufwand Analyse nicht stabilisierbare Anforderungen) + (Stabile Anforderungen * Aufwand Analyse stabile Anforderungen) + (Stabilisierbare Anforderungen * Aufwand Analyse stabilisierbare Anforderungen) + (Zu ändernde nicht stabilisierbare Anforderungen * Aufwand Analyse nicht stabilisierbare Anforderungen) + (Zu ändernde stabilisierbare Anforderungen * Aufwand Analyse stabilisierbare Anforderungen)) * "RE-Team" * Zeitliche Produktivität Analyse / Aufwand Analyse stabilisierbare Anforderungen), 0)

~ Person

Analyserate zu ändernde nicht stabilisierbare Anforderungen =

IF THEN ELSE (Anforderungsanalyse läuft flag, MIN (Zu ändernde nicht stabilisierbare Anforderungen / Konstante Day, ZIDZ ((Zu ändernde nicht stabilisierbare Anforderungen * Aufwand Analyse nicht stabilisierbare Anforderungen), (Nicht stabilisierbare Anforderungen * Aufwand Analyse nicht stabilisierbare Anforderungen) + (Stabile Anforderungen * Aufwand Analyse stabile Anforderungen) + (Stabilisierbare Anforderungen * Aufwand Analyse stabilisierbare Anforderungen) + (Zu ändernde nicht stabilisierbare Anforderungen * Aufwand Analyse nicht stabilisierbare Anforderungen) + (Zu ändernde stabilisierbare Anforderungen * Aufwand Analyse stabilisierbare Anforderungen)) * "RE-Team" * Zeitliche Produktivität Analyse / Aufwand Analyse nicht stabilisierbare Anforderungen), 0)

~ Person

Analyserate zu ändernde stabilisierbare Anforderungen =

IF THEN ELSE (Anforderungsanalyse läuft flag, MIN (Zu ändernde stabilisierbare Anforderungen / Konstante Day, ZIDZ ((Zu ändernde stabilisierbare Anforderungen * Aufwand Analyse stabilisierbare Anforderungen), (Nicht stabilisierbare Anforderungen * Aufwand Analyse nicht stabilisierbare Anforderungen) + (Stabile Anforderungen * Aufwand Analyse stabile Anforderungen) + (Stabilisierbare Anforderungen * Aufwand Analyse stabilisierbare Anforderungen) + (Zu ändernde nicht stabilisierbare Anforderungen * Aufwand Analyse nicht stabilisierbare Anforderungen) + (Zu ändernde stabilisierbare Anforderungen * Aufwand Analyse stabilisierbare Anforderungen)) * "RE-Team" * Zeitliche Produktivität Analyse / Aufwand Analyse stabilisierbare Anforderungen), 0)

~ Person

Änderungsrate analysierte stabilisierbare Anforderungen =

Änderungsrate stabilisierbare Anforderung lookup (Verständnis analysierte stabilisierbare Anforderungen)

~ 1 / Day

Änderungsrate realisierte stabilisierbare Anforderungen =

Änderungsrate stabilisierbare Anforderung lookup (Verständnis realisierte stabilisierbare Anforderungen)

~ 1 / Day

Anforderungsanalyse läuft flag =

IF THEN ELSE (Time < 8, Erste Iteration, Zweite Iteration)

~ Dmnl

Entwicklung läuft flag = IF THEN ELSE (Anforderungsanalyse läuft flag, 0, 1)

~ Dmnl

Gesamtprojektkosten = Aufwand Anforderungsanalyse + Aufwand Entwicklung

~ Person * Day

Output Änderungsrate stabilisierbare Anforderungen =

ZIDZ ((Änderungsrate stabilisierbare Anforderung lookup (0) * Stabilisierbare Anforderungen + Änderungsrate analysierte stabilisierbare Anforderungen * Analysierte stabilisierbare Anforderungen + Änderungsrate realisierte stabilisierbare Anforderungen * (Zu ändernde stabilisierbare Anforderungen + Realisierte stabilisierbare Anforderungen + Analysierte zu ändernde stabilisierbare Anforderungen)), (Anteil stabilisierbare Anforderungen * Zu bearbeitende Anforderungen))

~ 1 / Day

Output Anforderungen werden analysiert =

Nicht stabilisierbare Anforderungen analysieren + Stabile Anforderungen analysieren + Stabilisierbare Anforderungen analysieren + Zu ändernde nicht stabilisierbare Anforderungen analysieren + Zu ändernde stabilisierbare Anforderungen analysieren

~ Person

Output Realisierte Anforderungen =

Realisierte nicht stabilisierbare Anforderungen + Realisierte stabile Anforderungen + Realisierte stabilisierbare Anforderungen

~ Person * Day

Output Verständnis stabilisierbare Anforderungen =

ZIDZ (Verständnis analysierte stabilisierbare Anforderungen * Analysierte stabilisierbare Anforderungen + (Verständnis realisierte stabilisierbare Anforderungen * (Realisierte stabilisierbare Anforderungen + Zu ändernde stabilisierbare Anforderungen + Analysierte zu ändernde stabilisierbare Anforderungen)) + (Effektivität Anforderungsanalyse stabilisierbare Anforderungen lookup (0) * Stabilisierbare Anforderungen), (Anteil stabilisierbare Anforderungen * Zu bearbeitende Anforderungen))

~ Dmnl

Realisierungsrate nicht stabilisierbare Anforderungen =

IF THEN ELSE (Entwicklung läuft flag, MIN (Analysierte nicht stabilisierbare Anforderungen / Konstante Day, ZIDZ (Analysierte nicht stabilisierbare Anforderungen, Analysierte nicht stabilisierbare Anforderungen + Analysierte stabile Anforderungen + Analysierte stabilisierbare Anforderungen + Analysierte zu ändernde nicht stabilisierbare Anforderungen + Analysierte zu ändernde stabilisierbare Anforderungen) * Entwickler), 0)

~ Person

Realisierungsrate stabile Anforderungen =

IF THEN ELSE (Entwicklung läuft flag, MIN (Analysierte stabile Anforderungen / Konstante Day, ZIDZ (Analysierte stabile Anforderungen, Analysierte nicht stabilisierbare Anforderungen + Analysierte stabile Anforderungen + Analysierte stabilisierbare Anforderungen + Analysierte zu ändernde nicht stabilisierbare Anforderungen + Analysierte zu ändernde stabilisierbare Anforderungen) * Entwickler), 0)

~ Person

Realisierungsrate stabilisierbare Anforderungen =

IF THEN ELSE (Entwicklung läuft flag, MIN (Analysierte stabilisierbare Anforderungen / Konstante Day, ZIDZ (Analysierte stabilisierbare Anforderungen, Analysierte nicht stabilisierbare Anforderungen + Analysierte stabile Anforderungen + Analysierte stabilisierbare Anforderungen + Analysierte zu ändernde nicht stabilisierbare Anforderungen + Analysierte zu ändernde stabilisierbare Anforderungen) * Entwickler), 0)

~ Person

Realisierungsrate zu ändernde nicht stabilisierbare Anforderungen =

IF THEN ELSE (Entwicklung läuft flag, MIN (Analysierte zu ändernde nicht stabilisierbare Anforderungen / Konstante Day, ZIDZ (Analysierte zu ändernde nicht stabilisierbare Anforderungen, Analysierte nicht stabilisierbare Anforderungen + Analysierte stabile Anforderungen + Analysierte stabilisierbare Anforderungen + Analysierte zu ändernde nicht stabilisierbare Anforderungen + Analysierte zu ändernde stabilisierbare Anforderungen) * Entwickler / Aufwandsmultiplikator geänderte realisierte Anforderungen), 0)

~ Person

Realisierungsrate zu ändernde stabilisierbare Anforderungen =

IF THEN ELSE (Entwicklung läuft flag, MIN (Analyisierte zu ändernde stabilisierbare Anforderungen / Konstante Day, ZIDZ (Analyisierte zu ändernde stabilisierbare Anforderungen, Analyisierte nicht stabilisierbare Anforderungen + Analyisierte stabile Anforderungen + Analyisierte stabilisierbare Anforderungen + Analyisierte zu ändernde nicht stabilisierbare Anforderungen + Analyisierte zu ändernde stabilisierbare Anforderungen) * Entwickler / Aufwandsmultiplikator geänderte realisierte Anforderungen), 0)

~ Person

Verständnis analysierte stabilisierbare Anforderungen =

"Verständnis-Multiplikator Analyse" + ((1 - "Verständnis-Multiplikator Analyse") * "Verständnis-Multiplikator Anteil analysierte Anforderungen")

~ Dmnl

"Verständnis-Multiplikator Analyse" =

Effektivität Anforderungsanalyse stabilisierbare Anforderungen lookup (Aufwand Analyse stabilisierbare Anforderungen)

~ Dmnl

"Verständnis-Multiplikator Anteil analysierte Anforderungen" =

Indirekter Verständniskennwert lookup ((Analyisierte nicht stabilisierbare Anforderungen + Analyisierte stabile Anforderungen + Analyisierte stabilisierbare Anforderungen + Analyisierte zu ändernde nicht stabilisierbare Anforderungen + Analyisierte zu ändernde stabilisierbare Anforderungen + Realisierte nicht stabilisierbare Anforderungen + Realisierte stabile Anforderungen + Realisierte stabilisierbare Anforderungen) / Zu bearbeitende Anforderungen)

~ Dmnl

Verständnis realisierte stabilisierbare Anforderungen =

"Verständnis-Multiplikator Analyse" + ((1 - "Verständnis-Multiplikator Analyse") * "Verständnis-Multiplikator Anteil analysierte Anforderungen") + ((1 - "Verständnis-Multiplikator Analyse") * (1 - "Verständnis-Multiplikator Anteil analysierte Anforderungen") * "Verständnis-Multiplikator Implementierung")

~ Dmnl

Zweite Iteration =

IF THEN ELSE (((Stabilisierbare Anforderungen + Zu ändernde stabilisierbare Anforderungen + Nicht stabilisierbare Anforderungen + Zu ändernde nicht stabilisierbare Anforderungen + Stabile Anforderungen) > 4) :AND: (Erste Iteration abgeschlossen >= 1), 1, 0)

~ Dmnl

Konstanten

Änderungsrate nicht stabilisierbare Anforderungen = 0.0035

~ 1 / Day

Änderungsrate stabilisierbare Anforderung lookup

(([(0, 0) - (0.9, 0.07)], (0, 0.063), (0.1, 0.063), (0.2, 0.054), (0.3, 0.046), (0.4, 0.039), (0.5, 0.033), (0.6, 0.028), (0.7, 0.0237), (0.8, 0.02), (0.9, 0.0169))

~ 1 / Day

Anteil nicht stabilisierbare Anforderungen = 0.15

~ Dmnl

Anteil stabile Anforderungen = 0.5

~ Dmnl

Anteil stabilisierbare Anforderungen = 0.35

~ Dmnl

Aufwand Analyse nicht stabilisierbare Anforderungen = 0.1

~ Dmnl

Aufwand Analyse stabile Anforderungen = 0.1

~ Dmnl

Aufwand Analyse stabilisierbare Anforderungen = 0.2

~ Dmnl

Aufwandsmultiplikator geänderte realisierte Anforderungen = 1.5

~ Dmnl

Effektivität Anforderungsanalyse stabilisierbare Anforderungen lookup

(([(0, 0) - (0.33, 0.8)], (0, 0.2145), (0.025, 0.2145), (0.05, 0.39), (0.07, 0.4875), (0.1, 0.5525), (0.145, 0.6045), (0.2, 0.65), (0.27, 0.6825), (0.326972, 0.691228))

~ Dmnl

Entwickler = 2

~ Person

Erste Iteration = 1 + STEP (-1, 2)

~ Dmnl

Indirekter Verständnisgewinn lookup (([(0, 0) - (10, 10)], (0, 0), (1, 0))

~ Dmnl

Konstante Day = 1

~ Day

"RE-Team" = 2

~ Person

"Verständnis-Multiplikator Implementierung" = 0.35

~ Dmnl

Zeitliche Produktivität Analyse = 1

~ Dmnl

Zu bearbeitende Anforderungen = 50

~ Person * Day