



Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften
Department für Informatik

**Eine Methodik zur strategischen
Einflussbewertung von Methodenänderungen
in Entwicklungsprozessen**

Dissertation zur Erlangung des Grades eines
Doktors der Ingenieurwissenschaften

Vorgelegt von Roland Koppe
Gutachter Prof. Dr.-Ing. Axel Hahn
 Prof. Dr. Werner Damm
Tag der Disputation 27. April 2012

Für Annika und Charlotte

Abstract

The development of new and innovative products becomes a more and more complex task. At the same time, requirements for engineering processes regarding effort and cost reduction as well as improvements of product quality increase as enablers to place products successfully in the market.

New engineering methods and tools often promise large benefits for improving such engineering processes. However, it is not trivial to determine the effect which a new method has on an engineering process, and in addition if the new method pays-off for a company's specific application context. Evaluations of new methods without considering the specific application context are often not satisfying. Also expensive pilot studies rather lead to a specific picture about the influence of a new method than to a more generic statement of its impact.

In this work a methodology will be developed to analyse and estimate the possible impact of process changes quantitatively, e.g. new methods on engineering processes. Thus, strategic decision making for the selection of new methods for achieving process optimizations will be supported. A standard to describe engineering processes will be enhanced to represent cause-effect relationships between process elements and to describe process behaviour. Simulation of alternative engineering processes will be used to estimate the possible benefit of a process change.

Zusammenfassung

Die Entwicklung neuer und innovativer Produkte stellt eine immer komplexer werdende Aufgabe dar. Gleichzeitig steigen die Anforderungen an die Entwicklungsprozesse zur Reduktion des Aufwandes und der Kosten sowie der Verbesserung der Produktqualität, um Produkte gut im Markt positionieren zu können.

Neue Methoden und Werkzeuge versprechen große Potentiale zur Optimierung von Entwicklungsprozessen. Doch, ob sich eine neue Methode im konkreten Anwendungskontext eines Unternehmens wirklich lohnt beziehungsweise wie der Einfluss einer neuen Methode auf einen Entwicklungsprozess zu bewerten ist, ist nicht einfach zu beantworten. So ist eine vom Entwicklungskontext losgelöste Evaluierung von Methoden oft nicht ausreichend und auch kostspielige Pilotprojekte liefern in der Regel nur ein mögliches Bild des Einflusses von vielen.

In dieser Arbeit wird eine Methodik entwickelt, welche den Einfluss von Methodenänderungen in Entwicklungsprozessen quantitativ bewertet. Dadurch wird die strategische Entscheidungsfindung für die Auswahl neuer Methoden für angestrebte Prozessoptimierungen unterstützt. Dazu wird ein Standard zur Beschreibung von Entwicklungsprozessen um die Abbildung von Wirkzusammenhängen zwischen Prozesselementen und Prozessverhalten erweitert. Mit Hilfe von Simulation alternativer Entwicklungsprozesse wird dann der Mehrwert einer Prozessänderung bewertet.

Inhaltsverzeichnis

Abstract	v
Zusammenfassung	vii
Inhaltsverzeichnis	ix
Abbildungsverzeichnis.....	xi
Tabellenverzeichnis	xiii
Glossar	xv
1 Einführung	19
1.1 Problemstellung	21
1.2 Klärung wesentlicher Begriffe.....	22
1.3 Lösungsansatz und Ziele	23
1.4 Abgrenzung	25
1.5 Beispiel einer Methodenänderung	26
1.6 Aufbau der Arbeit	27
2 Stand der Technik	29
2.1 Anforderungen an eine Einflussanalyse und -bewertung	30
2.2 Methoden zur Leistungs- und Nutzenbewertung.....	31
2.3 Methoden zur Modellierung von Projekten.....	35
2.4 Methoden zur Modellierung von Prozessen	40
2.5 Methoden zur Beschreibung von Methoden.....	47
2.6 Ansätze zur Analyse von Prozessen	49
2.7 Stochastische Ansätze zur Analyse von Prozessen	53
2.8 Simulative Ansätze zur Analyse von Prozessen.....	58
2.9 Handlungsbedarf zur Einflussanalyse und Einflussbewertung	72
3 Methodik zur Einflussbewertung.....	75
3.1 Vorgehen für die Entwicklung von Simulationsmodellen.....	75
3.2 Beiträge dieser Arbeit	77
3.3 Überblick der Methodik.....	77
3.4 Definition der Ziele der Einflussanalyse	79
3.5 Beschreibung des Entwicklungsprozesses.....	81
3.6 Identifizierung und Beschreibung von Wirkzusammenhängen.....	87
3.7 Sammlung und Analyse empirischer Daten	92
3.8 Quantifizierung von Wirkzusammenhängen	99
3.9 Validierung des Modells.....	101

3.10	Beschreibung der neuen Methode	102
3.11	Virtuelle Integration der neuen Methode	103
3.12	Simulation der Entwicklungsprozesse	105
3.13	Bewertung des Einflusses der neuen Methode	106
3.14	Sensitivitätsanalyse	116
3.15	Einführungsstrategie	117
3.16	Zusammenfassung.....	119
4	Modell zur Einflussbewertung	121
4.1	Wichtige Konzepte und Erweiterungen	121
4.2	Definition des Modells.....	123
4.3	Zusammenfassung.....	131
5	Prototypische Umsetzung.....	133
5.1	Technologien.....	133
5.2	Architektur	134
5.3	Implementierung	135
5.4	Zusammenfassung.....	143
6	Evaluierung	145
6.1	Gammatone Filterbank.....	146
6.2	High Level Synthesis	149
6.3	Software Inspections	153
6.4	Software Reviews	156
6.5	Lebensmittel Produktentwicklung	158
6.6	Zusammenfassung und Bewertung	160
6.7	Anwendungsbedingungen der Methodik	165
7	Zusammenfassung und Ausblick	171
	Literaturverzeichnis.....	175
	Anhang A – Leitfaden zur Erhebung von Entwicklungsprozessen.....	191

Abbildungsverzeichnis

Abbildung 1.1	Beispiel eines einfachen Entwicklungsprozesses	26
Abbildung 1.2	Beispiel einer Prozessänderung durch eine neue Methode	26
Abbildung 1.3	Aufbau der vorliegenden Arbeit	27
Abbildung 2.1	Aufbau des Kapitels mit den untersuchten Themengebieten	29
Abbildung 2.2	Die Projektdimensionen Zeit, Kosten und Qualität.....	35
Abbildung 2.3	Notation der Critical Path Method nach (Burghardt 2006)	37
Abbildung 2.4	Notation der Graphical Evaluation and Review Technique nach (Pritsker 1966)	39
Abbildung 2.5	Konzepte der Structured Analysis and Design Technique nach (Ross 1977).....	41
Abbildung 2.6	Methoden Definition und Prozess in SPEM 2.0 (OMG 2008).....	46
Abbildung 2.7	Pyramide des Wissens über Wirkzusammenhänge und der dazu notwendige Aufwand zur Modellierung und Analyse dieser Wirkzusammenhänge	50
Abbildung 2.8	Einfache Markov-Kette mit absorbierenden Zustand.....	54
Abbildung 2.9	Wirkzusammenhänge bei System Dynamics nach (Park et al. 2008)	60
Abbildung 2.10	Aufgaben und Einflussfaktoren in der Test Phase nach (Rus und Collofello 2001).....	62
Abbildung 3.1	Überblick über die Beiträge dieser Arbeit.....	77
Abbildung 3.2	Der Goal Question Metrik Ansatz	80
Abbildung 3.3	Definition von Prozesselementen des Entwicklungsprozesses	85
Abbildung 3.4	Instanziierung der definierten Prozesselemente zu einem konkreten Prozessmodell.....	86
Abbildung 3.5	Grundsätzliche Wirkungen in einem Entwicklungsprozess	88
Abbildung 3.6	Beispiel einer Einflussmatrix zur Charakterisierung der Stärke eines Einflusses einer Eigenschaft auf andere Eigenschaften	90
Abbildung 3.7	Beschreibung von Wirkzusammenhänge im Entwicklungsprozess ...	91
Abbildung 3.8	Exemplarische Entwicklung funktionaler Korrektheit.....	100
Abbildung 3.9	Alternativer Entwicklungsprozess mit seinen Wirkzusammenhängen	104
Abbildung 3.10	Vergleich zweier Entwicklungsprozesse	107
Abbildung 3.11	Vergleich von 80%-Abdeckungsintervallen für sich überschneidende Stichproben.....	110

Abbildung 3.12	Bewertung von sich überschneidender Abdeckungsintervalle	110
Abbildung 3.13	Bewertung von Hypothesen für oder gegen eine Methodenänderung	112
Abbildung 3.14	Gewinn als Differenz zwischen zwei Entwicklungsprozessen gegenüber der Anzahl durchzuführender Entwicklungsprozesse	115
Abbildung 3.15	Aufwand von Methoden zur Gewinnung von Daten und Informationen.....	117
Abbildung 4.1	Wesentliche Konzepte des SPEM 2.0 Frameworks.....	122
Abbildung 4.2	Wesentliche Konzepte zur Beschreibung von Methoden und konkreten Prozessen.....	124
Abbildung 5.1	Architektur der prototypischen Umsetzung	134
Abbildung 5.2	Definition von Prozesselementen	136
Abbildung 5.3	Beschreibung des konkreten Entwicklungsprozesses	137
Abbildung 5.4	Beschreibung von Wirkzusammenhängen im Entwicklungsprozess	138
Abbildung 5.5	Modellierung von Verhaltensmodellen.....	139
Abbildung 5.6	Beschreibung eines Entscheidungsmodells	140
Abbildung 5.7	Schrittweise Simulation eines modellierten Entwicklungsprozesses	141
Abbildung 5.8	Analyse und Bewertungen für mehrfache Simulationen	142
Abbildung 5.9	Wizard zur Konfiguration von Eigenschaftswerten und Simulationsläufen	143
Abbildung 6.1	Verteilung des Aufwands über die verschiedenen Iterationen im Experiment.....	148
Abbildung 6.2	Modellierte Entwicklungsprozesse zur High Level Synthesis.....	150
Abbildung 6.3	Verteilung des Aufwands für zwei alternative Entwicklungsprozesse.....	152
Abbildung 6.4	25% und 75% Quantile der Differenz des Aufwands zweier alternativer Entwicklungsprozesse und unterschiedlich komplexe Designs.....	152
Abbildung 6.5	Software-Entwicklungsprozess mit Design und Code Inspections ..	154
Abbildung 6.6	Vergleich alternativer Software-Entwicklungsprozesse ohne sowie mit Design beziehungsweise Code Inspections	155
Abbildung 6.7	Entwicklungsprozess aus der Lebensmittelindustrie	158
Abbildung 6.8	Mittlere Aufwandsverteilung für unterschiedliche Aufgaben	159
Abbildung 6.9	Einfluss von modellierten Rezeptwissen und Übergangswahrscheinlichkeiten	159

Tabellenverzeichnis

Tabelle 2.1	Gegenüberstellung von System Dynamics und Discrete Event Simulation.....	58
Tabelle 3.1	Definition von Zielen, Fragen und Metriken.....	80
Tabelle 3.2	Eigenschaften von Artefakten	83
Tabelle 3.3	Eigenschaften von Rollen.....	83
Tabelle 3.4	Eigenschaften von Werkzeugen	84
Tabelle 3.5	Methoden zur Sammlung und Analyse von Daten.....	96
Tabelle 3.6	Kritische Punkte der t - bzw. der Normalverteilung ($n \rightarrow \infty$)	109
Tabelle 3.7	Bewertung von Hypothesen zur Empfehlung für eine Methodenänderung bzw. Beibehaltung des aktuellen Entwicklungsprozesses	111
Tabelle 6.1	Übersicht betrachteter Untersuchungen und Anwendungsbeispiele	145
Tabelle 6.2	Protokollierter Aufwand, Werkzeuglaufzeit und Eigenschaftswerte	147
Tabelle 6.3	Verteilung der Effektivität unterschiedlicher Inspektionsmethoden und den mit ihnen verbundenen Aufwand zur Fehlerbeseitigung nach (Briand et al. 1998)	154
Tabelle 6.4	Durchzuführende Entwicklungen bis zum Mehrwert eines neuen Werkzeugs	160
Tabelle 6.5	Kriterien zur Anwendung der Methodik	166

Glossar

Das Glossar gibt eine Erläuterung beziehungsweise Definition, der in dieser Arbeit verwendeten Begriffe. Neben dieser Erklärung werden auch Abkürzungen in diesem Verzeichnis aufgeführt.

Begriff	Erläuterung
Artefakt	Ein Artefakt, wie etwa ein Anforderungsdokument, ist ein Ergebnis von Entwicklungsaufgaben oder ist für die Durchführung von Entwicklungsaufgaben erforderlich.
Aufgabe	Eine Aufgabe ist eine atomare Tätigkeit, welche für einen bestimmten Untersuchungszweck nicht weiter in sinnvolle Einheiten herunter gebrochen werden kann, vergleiche (OMG 2008, S. 89f).
Aktivität	Eine Aktivität beschreibt im Gegensatz zu einer Aufgabe eine Sammlung von Prozesselementen und dient der hierarchischen Strukturierung von Prozessmodellen (OMG 2008, S. 97f).
Entwicklungsprozess	Ein Entwicklungsprozess ist das Vorgehen zur Entwicklung eines Produktes, indem Eingaben durch bestimmte Aufgaben in Ausgaben transformiert werden. Vergleiche (Lindemann 2007, S. 16).
Iteration	Eine Iteration beschreibt die Wiederholung einer Aufgabe zu einem späteren Zeitpunkt. Iterationen beziehungsweise Rückschritte zu vorherigen Aufgaben werden beispielsweise auf Grund von Fehlern in einem Artefakt notwendig.
Komplexität	Komplexität wird je nach Anwendungsgebiet und selbst in einer Domäne unterschiedlich verstanden. Allgemein kann Komplexität etwa in die Schwierigkeit der Beschreibung, die Schwierigkeit der Erstellung und den Grad der Organisation eines Systems differenziert werden (Lloyd 2001). Für diese Arbeit bietet sich ein Verständnis von Schwierigkeit der Erstellung (Aufwand) an.
Markov-Kette	Eine Markov-Kette ist ein stochastischer Prozess bei dem Wahrscheinlichkeiten für den Übergang von einem Zustand in einen anderen definiert werden (Meintrup und Schäffler 2005). Durch analytische oder simulative Methoden lassen sich etwa Aussagen über die Anzahl benötigter Iterationen eines Zustands ableiten. Für Entwicklungsprozesse können Aufgaben auf Zustände abgebildet werden.

Methode	Eine Methode ist eine formalisierte Beschreibung durchzuführender Aufgaben, um zielgerichtet ein bestimmtes Problem zu lösen. Mit einer Methode wird das „wie“ etwas zu lösen ist beschrieben (Lindemann 2007, S. 57).
Methodik	Eine Methodik beschreibt im Gegensatz zu einer Methode ein Vorgehen, bestehend aus mehreren Methoden. Eine Methodik beschreibt eher das „was“ zu tun ist, um ein Problem zu lösen (Lindemann 2007, S. 57).
Metrik	Eine Metrik oder auch Measurement ist eine Maßzahl, welche bestimmte Eigenschaften eines Produkts oder Eigenschaften eines Prozesses charakterisiert und für Bewertungen verwendet werden kann. Vergleiche (Ebert und Dumke 2007).
Modell	Ein Modell ist eine Abbildung oder Abstraktion eines relevanten Ausschnitts der realen Welt beziehungsweise eines Systems. Modelle werden damit zur Beschreibung und Untersuchung von Phänomenen der realen Welt oder von Systemen verwendet. Vergleiche (Kastens und Büning 2008).
Monte Carlo Simulation	Bei der Monte Carlo Simulation werden Wahrscheinlichkeitsverteilungen verwendet, um Einflussgrößen von Modellen zufällig zu setzen. Mehrfache Simulationsläufe erlauben die Variation von Einflussgrößen und damit die Nachahmung des Verhaltens eines Systems (Madachy 2008), (Lemieux 2009).
Produkt	Ein Produkt ist das Ergebnis eines Entwicklungsprojekts und verfügt über einen Produktlebenszyklus von der ersten Idee bis hin zur Entsorgung (Ehrlenspiel 2009).
Projekt	Ein Projekt ist ein einmaliges Vorhaben, welches durch eine Begrenzung der Rahmenbedingungen Zeit, Kosten und Qualität gekennzeichnet ist (Burghardt 2006).
Prozessbeschreibung	Eine Prozessbeschreibung verfügt über eine syntaktische und semantische Beschreibung der Prozesselemente eines Entwicklungsprozesses (OMG 2008) und ist damit ein Modell der realen Welt (Sommerville 2007, S. 8).
Prozesselement	In Prozessbeschreibungen werden verschiedene Elemente unterschieden. Hierzu zählen im Wesentlichen Aufgaben, Aktivitäten, Abläufe, Artefakte, Rollen und Werkzeuge. Vergleiche (OMG 2008), (Sommerville 2007, S. 8).

Prozessmodell	Siehe Prozessbeschreibung.
Qualität	Qualität beschreibt die Güte der Erfüllung gesetzter Anforderungen an die inhärenten Merkmale eines Produkts. Vergleiche dazu (EN ISO 9000:2005).
Reifegrad	Der Begriff Reifegrad bezieht sich auf die Fähigkeiten und Leistung eines Entwicklungsprozesses oder einer Organisation. Ein wichtiger Aspekt von Reifegradmodellen wie CMMI oder SPICE ist die Verbesserung von Prozessen.
Return-on-investment	Return-on-investment beziehungsweise Rentabilität ist eine Kennzahl, welche den zu erwartenden Nutzen einer Investition als Quotient aus Gewinn und Kosten beschreibt (Götze 2006).
Simulation	Simulation ist eine Methode das Verhalten eines Systems der realen Welt zu imitieren. Durch virtuelle Experimente können Erkenntnisse über die reale Welt gesammelt werden, ohne diese selbst ändern zu müssen (Law und Kelton 1991).
System	Ein System ist eine Zusammenfassung von Elementen zwischen denen bestimmte Beziehungen existieren und die sich zudem gegenseitig über Wirkzusammenhänge beeinflussen können. Demnach sind auch Entwicklungsprozesse mit ihren Prozesselementen, Beziehungen und Wirkzusammenhängen ein System.
Time-to-market	Time-to-market ist eine Kennzahl, welche die Dauer von der Entwicklung eines Produkts bis hin zur Einführung des Produkts in den Markt beschreibt. Je geringer diese Dauer, desto höhere Erlöse werden durch das Produkt erwartet.
Verhaltensmodell	Ein Verhaltensmodell oder Behavior Model beschreibt in dieser Arbeit das Verhalten einer Aufgabe beziehungsweise modellierter Wirkzusammenhänge zwischen Prozesselementen zum Zweck der Simulation eines Entwicklungsprozesses.
VHDL	Very High Speed Integrated Circuit Hardware Description Language ist eine Sprache zur Beschreibung von Hardware.
Wahrscheinlichkeit	In der Statistik wird unter Wahrscheinlichkeit die Häufigkeit des Auftretens eines bestimmten Ereignisses gegenüber anderen assoziierten Ereignissen verstanden. Vergleiche (Kohn 2005, S. 204ff).

1 Einführung

Die Entwicklung neuer Produkte stellt eine immer schwieriger werdende Aufgabe dar. Auf der einen Seite sollen Produkte die umfangreicher werdenden Bedürfnisse der Kunden decken und gleichzeitig einen solch innovativen Charakter haben, dass sie sich gut im Markt positionieren lassen, um Unternehmen ein entsprechendes Gewinnpotential zu ermöglichen, vergleiche (Lindemann 2007). Auf der anderen Seite steigen die Anforderungen an den jeweiligen Entwicklungsprozess für solche Produkte mit Blick auf die Projektdimensionen Qualität, Kosten und Zeit. Die Qualität eines Produktes soll verbessert werden, die Kosten und die Entwicklungsdauer reduziert werden. Gerade vor dem Hintergrund von Kennzahlen wie return-on-investment und time-to-market ist die Reduzierung der Entwicklungsdauer ein kritischer Erfolgsfaktor (Scheer et al. 2006), (Ophrey 2005). So kann ein verspäteter Markteintritt zu deutlichen Gewinneinbußen führen, da bereits ähnliche Produkte der Konkurrenz auf dem Markt zu finden sind und daher nur noch niedrigere Preise erzielbar sind (von Ahsen et al. 2009), (Kessler und Bierly 2002).

Über die Bedeutung eines rechtzeitigen Markteintritts hinaus, ist die strategische Perspektive erfolgreicher Produktentwicklung nicht zu vernachlässigen. So existieren allgemeine Ansätze wie etwa die Balanced Scorecard (Kaplan und Norton 1997), Six Sigma (Töpfer 2007) oder auch das Total Quality Management (Hummel und Malorny 2002), welche auf eine grundsätzliche und strategische Verbesserung von Organisationen und Prozessen zielen. Ergänzend zu solchen Ansätzen werden Reifegradmodelle wie das Capability Maturity Model Integration (CMMI) (CMU/SEI 2006) oder auch der internationale Standard Software Process Improvement and Capability Determination (SPICE, ISO / IEC 15504) verwendet, um die Reife von Organisationen und Prozessen und insbesondere von Entwicklungsprozessen zu verbessern. Dabei streben Reifegradmodelle eine kontinuierliche Verbesserung durch das Verstehen entsprechender Prozesse sowie den Aufbau von Wissen an. Grundsätzlich wird bei all diesen Ansätzen auf eine Steigerung der Qualität und eine Reduktion der Kosten gezielt, welches auch der operativen projektbezogenen Leistung zu Gute kommen soll.

Neue Methoden und Werkzeuge

Neue Methoden und Werkzeuge versprechen im Kontext der Verbesserung von Entwicklungsprozessen große Potentiale. Hinter solchen Optimierungsversprechen stehen die Reduzierung der notwendigen Entwicklungsdauer, die Reduzierung der entsprechenden Entwicklungskosten und die Steigerung der Produktqualität. Hierzu werden laufend neue Ansätze, Methoden und Werkzeuge in Forschung und Entwicklung entwickelt, die einen Mehrwert für die Produktentwicklung leisten sollen. Dieses zeigt sich auch in der aktuellen Forschungsförderung bei Programmen wie der Hightech-Strategie 2020 und KMU Innovativ des Bundesministeriums für Bildung und Forschung sowie der Forschung zu Information and Communication Technologies (ICT) in Förderprogrammen der europäischen Kommission.

Doch ob sich eine neue Methode oder ein neues Werkzeug im Prozesskontext für spezifische Produkte eines Unternehmens wirklich lohnt beziehungsweise wie der Einfluss einer neuen Methode oder eines neuen Werkzeugs auf einen Entwicklungsprozess zu bewerten ist, ist nicht einfach zu beantworten.

Eine möglichst sichere quantitative Bewertung des Einflusses einer neuen Methode oder eines neuen Werkzeugs auf einen unternehmensspezifischen Entwicklungsprozess würde die Entscheidungsbasis von Managern für oder gegen Investitionen in Prozessänderungen signifikant vergrößern. Dabei ist eine Entscheidung in mehreren Dimensionen zu treffen, etwa im Kontext des Dreiecks der Projektdimensionen Zeit, Kosten und Produktqualität (Burghardt 2006). Da die Anschaffung und Einführung neuer Methoden und Werkzeuge mit zum Teil immensen Kosten verbunden ist, fließt als strategische Komponente zudem die Amortisationszeit einer neuen Methode oder eines neuen Werkzeuges in die Entscheidungsfindung ein. Denn wie oft muss eine Methode erfolgreich angewendet werden, bis durch eine Investition Gewinne erzielt werden?

Im Bereich der Wirtschaftlichkeitsrechnung und der Investitionsrechnung existieren unterschiedliche Methoden, um den potentiellen Mehrwert einer Investition abzuschätzen (Götze 2006). Allerdings basieren diese Methoden in der Regel auf abstrakten Annahmen, fokussieren auf Produktionsprozesse und betrachten nicht die Besonderheiten von Entwicklungsprozessen und die einen Entwicklungsprozess beeinflussenden Wirkzusammenhänge. Entsprechend dieser Abstraktion können auch nur vage Aussagen über den Einfluss einer Methodenänderung für Entwicklungsprozesse vorgenommen werden.

Es stellt sich also die Frage, wie neue Methoden oder Werkzeuge für einen ausgewählten Prozess- und Produktkontext möglichst sicher, das heißt mit geringem Risiko und vertretbarem Aufwand bewertet werden können. Eine vom Entwicklungskontext losgelöste oder isolierte Evaluierung von Methoden oder Werkzeugen ist dabei oft nicht ausreichend. Durch solche Evaluierungen werden erforderliche Änderungen an vorge-

lagerten Entwicklungsschritten in der Regel vernachlässigt. Auch der Einfluss einer neuen Methode oder eines neuen Werkzeugs auf folgende Entwicklungsschritte oder folgende Prozesse wird nicht adäquat beachtet.

Als Alternative zu isolierten Evaluierungen bieten sich beispielsweise Pilotprojekte an. Zwar ermöglichen Pilotprojekte die Beobachtung und Analyse grundsätzlichen Prozessverhaltens, doch ist mit ihnen oft ein hoher zeitlicher und monetärer Aufwand verbunden. Da Pilotprojekte außerdem nur eine begrenzte Kontrolle über verschiedenste Einflussfaktoren erlauben, sind statistisch signifikante und valide Aussagen über den positiven oder negativen Einfluss einer Methode oder eines Werkzeugs nicht möglich (Münch et al. 2005). So können andere Projektkonstellationen, bedingt durch unterschiedliche Mitarbeiter-, Aufgaben- oder Produktausprägungen höchst verschiedene Ergebnisse liefern.

1.1 Problemstellung

Es besteht also ein grundlegendes Problem in der Analyse und Bewertung des Einflusses einer Methodenänderung auf einen Entwicklungsprozess und damit auch in der Beurteilung des Potenzials beziehungsweise der Rentabilität einer neuen Methode auf gleichartige Entwicklungsprozesse.

Eine besondere Herausforderung für die Bewertung von Änderungen in Entwicklungsprozessen wird durch die Einzigartigkeit von konkreten Entwicklungsprozessen bestimmt. In klassischen Produktionsprozessen, beispielsweise von Massenwaren, sind die notwendigen Aktivitäten und deren Ablauf klar definiert. Im Gegensatz dazu ist für Entwicklungsprozesse zwar das grundsätzliche Vorgehen definiert, die tatsächliche Ausprägung des Ablaufes der Aktivitäten und die Gestaltung des Produktes ergeben sich genauso, wie mögliche Probleme und Lösungsansätze, erst während der Entwicklung. Somit existiert eine mehr oder weniger große Unsicherheit in der Entwicklung. Vergleiche dazu (Lang 2004), (Lindemann 2007, S. 8).

Das grundlegende Problem zur Bewertung von Einflüssen in Entwicklungsprozessen ist kein neues. Vor allen Dingen aus dem Bereich des Software Engineering stammen viele Arbeiten zur Abschätzung von Prozessänderungen durch Simulation. Wie der Stand der Technik in Kapitel 2 zeigen wird, fehlt es existierenden Arbeiten unter anderem an einem oder mehreren der folgenden Punkte

- eine Methodik zur Analyse beziehungsweise Bewertung von Einflüssen durch Prozessänderungen bezogen auf Entwicklungsprozesse,
- die Anwendung von Einflussanalyse und Einflussbewertungen über spezielle Probleme, beispielsweise über die Software-Entwicklung, hinaus,

- ein Ansatz zur Auswahl relevanter Charakteristika entscheidender Design Artefakte und deren Wirkzusammenhänge, sowie die Beeinflussung dieser Eigenschaften durch beispielsweise Werkzeuge, Methoden und Mitarbeiter sowie
- ein Modell zur Abbildung von Entwicklungsprozessen und dem damit verbundenen Wissen sowie deren inhärenten Wirkzusammenhängen.

Ein Problem der Einflussanalyse und Einflussbewertung stellt auch der Kontext der Untersuchung von Prozessänderungen dar. So ist bei einer Evaluierung von neuen Methoden oder Werkzeugen beispielsweise zu beachten, dass

- die betrachtete Methode Bedingungen an vorherige Prozessschritte stellt, wie etwa bestimmte Anforderungen zur Anwendung der neuen Methode im gegebenen Prozesskontext,
- die betrachtete Methode Folgeschritte des Prozesses oder folgende Prozesse beeinflusst, indem die Methode bestimmte Zwischenergebnisse des Prozesses verändert und
- die neue Methode bestimmte Aufwendungen für die Anschaffung sowie die Einführung der Methode in die Organisation und den betrachteten Prozesskontext erfordert.

1.2 Klärung wesentlicher Begriffe

Im Rahmen dieser Arbeit werden verschiedene Begriffe verwendet, welche in diesem Abschnitt definiert beziehungsweise geklärt werden. Eine Übersicht weiterer Begriffe findet sich im Glossar auf Seite xv.

- **Modell.** Ein Modell ist eine Abbildung oder Abstraktion eines Ausschnitts der realen Welt. Modelle werden dabei zur Beschreibung und Untersuchung von Phänomenen der realen Welt verwendet. Vergleiche (Kastens und Büning 2008).
- **Metamodell.** Ein Metamodell bestimmt einen formalen Rahmen zur Beschreibung von Modellen. Ein Metamodell definiert also für Modelle verwendbare Konzepte und Beziehungen sowie deren Syntax und Semantik. Vergleiche dazu (Brugger 2005, S. 493ff). Ein bekanntes Beispiel für ein Metamodell ist die Unified Modelling Language (UML) (OMG 2009a).
- **Produkt.** Ein Produkt ist das Ergebnis eines Entwicklungsprojektes und verfügt über einen Produktlebenszyklus von der ersten Idee bis hin zur Entsorgung (Ehrlenspiel 2009). Der Fokus dieser Arbeit liegt dabei auf der Produktentwicklung.

- **Artefakt.** Ein Artefakt ist ein Ergebnis von Entwicklungsaufgaben oder ist für die Durchführung von Entwicklungsaufgaben erforderlich. Dabei ist ein Artefakt eine abstrakte Repräsentation eines Teils des zu entwickelnden Produktes, wie etwa ein Anforderungsdokument.
- **Entwicklungsprozess.** Ein Entwicklungsprozess ist das Vorgehen zur Entwicklung eines Produktes, indem Eingaben durch bestimmte Aufgaben in Ausgaben transformiert werden. Vergleiche (Lindemann 2007, S. 16).
- **Prozessmodell.** Ein Prozessmodell oder auch Prozessbeschreibung ist eine abstrakte Beschreibung eines Entwicklungsprozesses. Ein Prozessmodell verfügt mit einer Beschreibung von etwa Aktivitäten, Aufgaben, Abläufen, Artefakten, Rollen und Werkzeugen über verschiedene miteinander in Beziehung stehende Prozesselemente (Sommerville 2007, S. 8), (OMG 2008).
- **Wirkzusammenhang.** Ein Wirkzusammenhang beschreibt Ursache und Wirkung eines beobachtbaren Phänomens der realen Welt. Für diese Arbeit sind Wirkzusammenhänge zwischen Prozesselementen beziehungsweise Eigenschaften dieser Prozesselemente vom besonderen Interesse.
- **Aufgabe und Aktivität.** In der Literatur werden die Begriffe Aufgabe und Aktivität häufig synonym gebraucht oder es findet keine Differenzierung statt. So wird etwa im Projektmanagement und in der Netzplantechnik der Begriff Vorgang verwendet. Für diese Arbeit beschreiben Aufgaben atomare Tätigkeiten, welche für einen bestimmten Untersuchungszweck nicht weiter in sinnvolle Teile aufgebrochen werden können. Aktivitäten beschreiben hingegen eine Sammlung von Prozesselementen und dienen der hierarchischen Strukturierung von Prozessmodellen. Vergleiche dazu (OMG 2008).

1.3 Lösungsansatz und Ziele

Diese Arbeit stellt eine Methodik zur Einflussbewertung von Methodenänderungen in Entwicklungsprozessen vor, mit der durch ein feingranulares Verständnis von Wirkzusammenhängen eine verlässliche Bewertung von neuen Methoden und Werkzeugen ermöglicht werden soll. Entsprechend ergibt sich die Notwendigkeit Entwicklungsprozesse mit ihren Wirkzusammenhängen gezielter und detaillierter zu betrachten. Die im vorherigen Abschnitt bereits angesprochene Unsicherheit bezüglich des Verhaltens von Entwicklungsprozessen erfordert eine probabilistische Modellierung von Wirkzusammenhängen. Mit Hilfe von Simulation von Modellen alternativer Entwicklungsprozesse können verschiedene Szenarien alternativer Entwicklungsprozesse exploriert und der Mehrwert einer Prozessänderung bewertet werden.

Eine Einschränkung für diese Betrachtungsweise wird allerdings durch den bedingten Aufwand der Untersuchung gegeben. Die in dieser Arbeit entwickelte Methodik ermöglicht es, mit überschaubarem Aufwand erste Abschätzungen über den Mehrwert einer neuen Methode für einen ausgewählten Prozesskontext vorzunehmen. Durch einen iterativen Ansatz soll das Wissen über Entwicklungsprozesse kontinuierlich weiter aufgebaut, Wirkzusammenhänge formalisiert und fundierte Einflussanalysen ermöglicht werden.

Als Basis fundierter Aussagen wird dazu die Extraktion, Modellierung und Integration des implizit oder explizit vorhandenen Expertenwissens beziehungsweise empirischer Daten über die Wirkzusammenhänge zwischen relevanten Prozesselementen des betrachteten Entwicklungsprozesses gefordert.

Aus dem skizzierten Lösungsansatz ergibt sich damit die wissenschaftliche Fragestellung dieser Arbeit:

Wie kann eine Methodik gestaltet sein, um Entwicklungsprozesse und ihre Wirkzusammenhänge zu verstehen, zu beschreiben und auf dieser Basis Einflussanalysen und Einflussbewertungen für Änderungen an Entwicklungsprozessen, etwa durch neue Methoden, durchzuführen?

Die wissenschaftliche Fragestellung lässt sich in mehrere Ziele unterteilen, die mit der wissenschaftlichen Wertschöpfung einhergehen. Diese Ziele werden wie folgt definiert:

Ziel 1. Erhebung des Stands der Technik und Beschreibung der Grundlagen zu verwandten und relevanten Themengebieten der Fragestellung. Die Auseinandersetzung mit dem Stand der Technik ermöglicht zum einen die Auswahl interessanter Ansätze für eine Einflussbewertung. Auf der anderen Seite leitet sich der Handlungsbedarf dieser Arbeit aus dem Stand der Technik ab.

Ziel 2. Entwicklung einer Methodik zur fundierten Einflussanalyse von Methodenänderungen in Entwicklungsprozessen. Eine entsprechende Methodik leitet von der Beschreibung von Entwicklungsprozessen und neuen Methoden über die Identifizierung und Modellierung von Wirkzusammenhängen bis hin zur quantitativen Bewertung der Einflüsse einer neuen Methode auf einen Entwicklungsprozess an.

Ziel 3. Spezifikation eines Metamodells zur Beschreibung von Entwicklungsprozessen und Wirkzusammenhängen zwischen verschiedenen Prozesselementen. Um Entwicklungsprozesse zu verstehen und Wirkzusammenhänge für eine Einflussbewertung formalisieren zu können, ist ein entsprechendes Metamodell notwendig. Das Metamodell ist dabei aus dem Stand der Technik abzuleiten.

Ziel 4. Metamodell und Methodik werden durch ein prototypisches Werkzeug unterstützt. Der Prototyp soll die Modellierung von Entwicklungsprozessen und Wirkzusammenhängen erlauben. Dazu zählen auch die Wiederverwendung bereits beschriebener Prozesselemente und der stetige Aufbau von Wissen über Entwicklungsprozesse. Zudem ermöglicht der Prototyp die Einflussanalyse und bereitet entsprechende Daten zur Bewertung des Einflusses einer Methodenänderung für einen Entwicklungsprozess auf.

Ziel 5. Metamodell und Methodik werden anhand verschiedener Szenarien und Fallstudien erprobt. Die Evaluierung zeigt zum einen die Anwendung des entwickelten Ansatzes in verschiedenen Anwendungsdomänen. Weiterhin werden die Ergebnisse der Arbeit kritisch bewertet.

1.4 Abgrenzung

Diese Arbeit grenzt sich durch die im Folgenden genannten Punkte gegenüber anderen Arbeiten zur Einflussanalyse ab, welche in der Regel nur ein Teil der genannten Punkte tangieren. Die Bewertung des Stands der Technik in Kapitel 2 sowie die Evaluierung der Arbeit in Kapitel 6 wird diese illustrieren.

- Die Methodik ist durch ihre Generalität auf Entwicklungsprozesse verschiedener Domänen anwendbar. Die Methodik fokussiert, im Gegensatz zu anderen Arbeiten, nicht auf bestimmte Entwicklungsprozesse oder Anwendungsdomänen.
- Die Methodik fokussiert nicht auf ein bestimmtes Entwicklungsvorgehen. Es können verschiedene Vorgehensmodelle mit der Methodik untersucht werden.
- Einflüsse neuer Methoden oder neuer Werkzeuge werden explizit durch Wirkzusammenhänge beschrieben.
- Der Detaillierungsgrad von zu modellierenden Wirkzusammenhängen wird anhand des Prozesskontextes bestimmt.
- Die Methodik erlaubt die schrittweise Verfeinerung von Wirkzusammenhängen und damit die kontinuierliche Verbesserung der Bewertungssicherheit.
- Die Methodik basiert auf einem standardkonformen Modell zur Beschreibung von Entwicklungsprozessen und Wirkzusammenhängen.

Im Weiteren wird der Fokus dieser Arbeit auf die Bewertung des Einflusses einer Methodenänderung auf einen Entwicklungsprozess untersucht. Damit ist das Ziel verbunden, Aussagen über den Mehrwert einer neuen Methode vornehmen zu können, um Entscheidungen für oder gegen eine Investition in eine neue Methode treffen zu können. Dabei betrachtet die Arbeit das Thema nicht als Optimierungsproblem beispielsweise der linearen oder nicht-linearen Programmierung (Luenberger 2003), sondern interessiert sich für den grundsätzlichen Einfluss einer neuen Methode.

1.5 Beispiel einer Methodenänderung

Im Verlauf der Arbeit wird ein Beispiel verwendet, um die Anwendung existierender Ansätze und Ergebnisse der Arbeit zu veranschaulichen. Das Beispiel beschreibt einen stark vereinfachten Entwicklungsprozess aus der Hardware domäne, zur Entwicklung eines Filters zur Kantenerkennung in einem Nachtsichtgerät für Automobile.

Der existierende Entwicklungsprozess ist durch die manuelle Entwicklung von einer algorithmischen Verhaltensbeschreibung hin zu einer hardwarenahen Sprache geprägt. Durch eine neue Methode, der sogenannten High Level Synthesis (HLS), soll nun der manuelle Entwicklungsaufwand reduziert werden, indem HLS automatisch die algorithmische Verhaltensbeschreibung in eine hardwarenahe Sprache übersetzt (Coussy und Morawiec 2008).

Abbildung 1.1 zeigt ein Modell des Entwicklungsprozesses, welcher aus drei Aufgaben besteht. Mit der Aufgabe *Behavioral Design* wird eine algorithmische Verhaltensbeschreibung in einer Sprache wie C/C++ oder SystemC vorgenommen. Es folgt mit der Aufgabe des Register Transfer Level Designs (*RTL Design*) die Übersetzung der Verhaltensbeschreibung in das hardwarenahe Very High Speed Integrated Circuit Hardware Description Language (VHDL). Schließlich wird in der Aufgabe *Analysis* das resultierende Design überprüft. Sollten dabei Fehler gefunden werden, erfolgt ein Rücksprung zu einer der früheren Aufgaben.

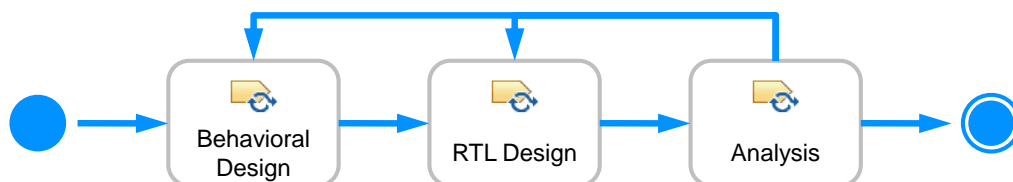


Abbildung 1.1 Beispiel eines einfachen Entwicklungsprozesses

Abbildung 1.2 veranschaulicht die angenommene Prozessänderung durch die neue Methode HLS. Dabei wird angenommen, dass die neue Methode die Aufgabe *RTL Design* komplett ersetzt. Änderungen am Design werden durch die Aufgabe *Behavioral Design* vorgenommen. Zudem existiert nur noch ein Rücksprung von der *Analysis* Aufgabe zur Aufgabe des *Behavioral Design*.

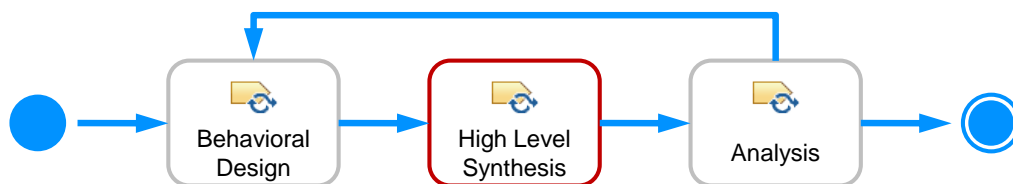


Abbildung 1.2 Beispiel einer Prozessänderung durch eine neue Methode

Für die Einflussanalyse und Einflussbewertung ist nun zu prüfen, welche weiteren Änderungen mit der Einführung von HLS einhergehen und wie der Einfluss von HLS auf den Entwicklungsprozess zu bewerten ist. Dazu stellen sich Fragen wie etwa:

- Reduziert die High Level Synthesis den Entwicklungsaufwand?
- Gibt es dank High Level Synthesis weniger Fehler im Design?
- Ab wann rechnet sich die Investition in High Level Synthesis?

Ausführlicher wird das Beispiel während der Evaluierung in Abschnitt 6.2 als durchgeführte Fallstudie aufgegriffen.

1.6 Aufbau der Arbeit

Das zurückliegende Kapitel gab eine Einführung in das Thema dieser Arbeit und stellte das Problem, den angestrebten Lösungsansatz und die damit verbundenen Ziele dar. Außerdem wurde ein Beispiel einer Methodenänderung eingeführt, welche zur Erläuterung der Arbeit in den folgenden Kapiteln wieder aufgegriffen wird.

Abbildung 1.3 veranschaulicht den Aufbau der Arbeit.

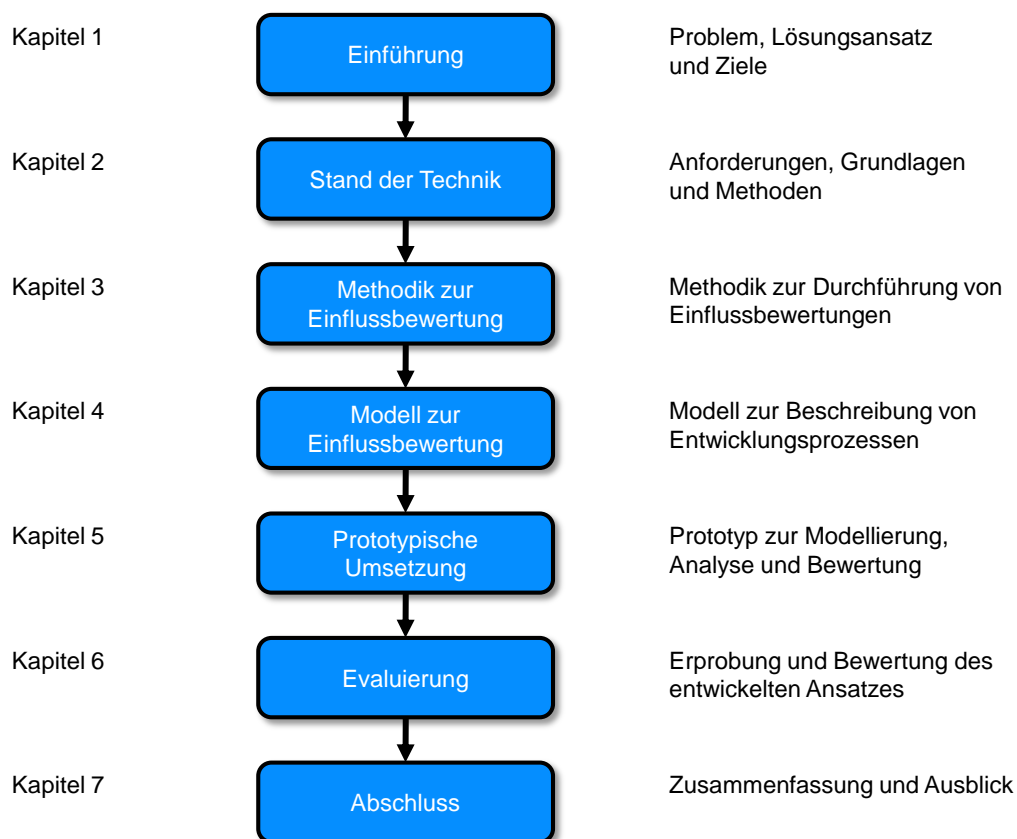


Abbildung 1.3 Aufbau der vorliegenden Arbeit

Im Kapitel 2 werden zunächst die Anforderungen an eine Lösung zur Umsetzung der genannten Ziele definiert. Es folgt dann ein Überblick über existierende Methoden und Ansätze zur Beschreibung, Analyse und Bewertung von Entwicklungsprozessen.

Kapitel 3 stellt die in dieser Arbeit entwickelte Methodik zur Einflussbewertung vor. Die Methodik umfasst das Vorgehen zur Beschreibung von Entwicklungsprozessen bis hin zur Bewertung des Einflusses einer neuen Methode oder eines neuen Werkzeugs auf einen Entwicklungsprozess. Die Beschreibung der einzelnen Schritte erfolgt dabei anhand des in Abschnitt 1.5 eingeführten Beispiels einer Methodenänderung. Die Methodik wird durch die Beschreibung des zugrunde liegenden Modells für die Modellierung von Methoden und Entwicklungsprozessen in Kapitel 4 ergänzt. Der Fokus liegt dabei auf den für Einflussanalysen und Einflussbewertungen notwendigen Konzepten. Basierend auf den vorherigen Kapiteln 3 und 4 wird dann in Kapitel 5 die prototypische Umsetzung der entwickelten Methodik vorgestellt.

In Kapitel 6 werden verschiedene Untersuchungen und Anwendungsbeispiele vorgestellt, welche für die Grundlagen und Evaluierung dieser Arbeit durchgeführt wurden. Dazu zählen Untersuchungen aus unterschiedlichen Anwendungsdomänen.

Abschließend fasst Kapitel 7 die Ergebnisse dieser Arbeit zusammen und gibt einen Ausblick für die weitere Forschung im Bereich der Einflussanalyse und Einflussbewertung für Entwicklungsprozesse.

2 Stand der Technik

Methoden zur Beschreibung, Analyse und Bewertung von Methoden und Entwicklungsprozessen

Für diese Arbeit wurden die in Abbildung 2.1 gezeigten Themengebiete in Bezug auf den Stand der Technik als relevant mit Blick auf den Aufbau von Wissen über Entwicklungsprozesse, Wirkzusammenhänge und die Einflussanalyse und Einflussbewertung von Methodenänderungen in Entwicklungsprozessen identifiziert.

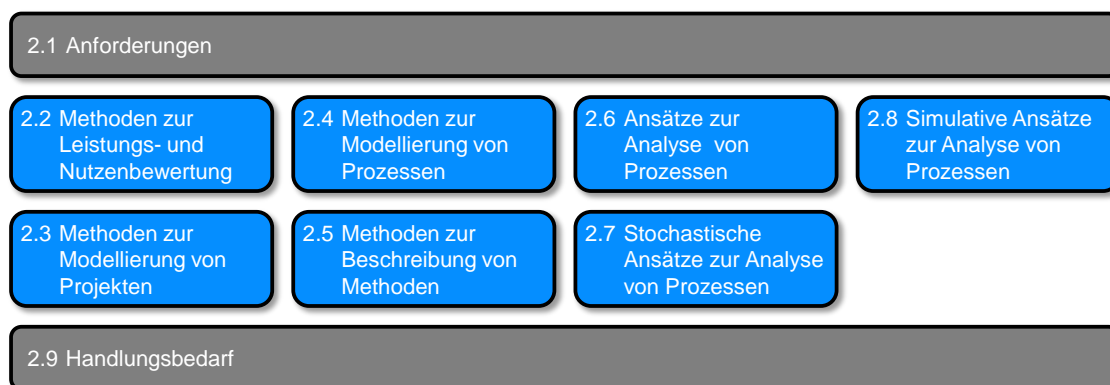


Abbildung 2.1 Aufbau des Kapitels mit den untersuchten Themengebieten

In Abschnitt 2.1 werden zunächst Anforderungen an Ansätze und Methoden aufgestellt, welche für die Beschreibung, Analyse und Bewertung von Methoden und Entwicklungsprozessen erforderlich scheinen und welche es für diese Arbeit zu erfüllen gilt.

Zu den betrachteten Themengebieten zählen Methoden der Leistungs- und Nutzenbewertung (Abschnitt 2.2), Methoden zur Modellierung von Projekten (Abschnitt 2.3) und Prozessen (Abschnitt 2.4) sowie zur Beschreibung von Methoden (Abschnitt 2.5). Für die Analyse von Entwicklungsprozessen werden in Abschnitt 2.6 zunächst grundsätzliche Ansätze betrachtet. In den Abschnitten 2.7 und 2.8 werden dann stochastische Ansätze und simulative Ansätze zur Analyse von Entwicklungsprozessen vorgestellt.

Schließlich wird der Handlungsbedarf aus den vorgestellten Ansätzen und Methoden gegenüber der aufgestellten Anforderungen in Abschnitt 2.9 festgestellt.

2.1 Anforderungen an eine Einflussanalyse und -bewertung

Das Ziel der quantitativen Einflussanalyse und Einflussbewertung von Methodenänderungen in Entwicklungsprozessen erfordert miteinander vergleichbare Kennzahlen, welche durch eine Analyse des Verhaltens alternativer Entwicklungsprozesse beschrieben werden. Um nun Wissen über Entwicklungsprozesse aufzubauen und auf dieser Basis das Verhalten von Entwicklungsprozessen zu beschreiben, ist eine Modellierung von Entwicklungsprozessen und Wirkzusammenhängen notwendig. Dabei ist der bereits angesprochenen Unsicherheit in Entwicklungsprozessen Sorge zu tragen. Schließlich erfordert die Durchführung von Einflussanalysen ein methodisches und zielgerichtetes Vorgehen.

Im Folgenden werden aus diesen Überlegungen heraus Anforderungen an Ansätze zur Einflussanalyse und Einflussbewertung von Methodenänderungen in Entwicklungsprozessen aufgestellt. Die in den folgenden Abschnitten betrachteten Ansätze und Methoden werden dann im Hinblick auf ihre Eignung für diese Arbeit untersucht.

Anforderung 1. Kennzahlen. Änderungen in Entwicklungsprozessen, wie etwa die Einführung einer neuen Methode, werden als Investition betrachtet. Mit Blick auf die strategische Entscheidungsfindung sind Methoden und Kennzahlen erforderlich, welche eine quantitative Darstellung und Bewertung einer Investition erlauben.

Anforderung 2. Unsicherheit. Auch wenn das Vorgehen zur Entwicklung eines Produktes durch einen Entwicklungsprozess definiert ist, ergibt sich der konkrete Ablauf eines Prozesses erst während der Entwicklung. Die Änderung von Produkteigenschaften, Aufwände und Entscheidungen für das weitere Vorgehen sind nicht deterministisch. Die damit einhergehende Unsicherheit ist geeignet abzubilden.

Anforderung 3. Analyse. Für die Analyse und Bewertung des Einflusses einer Änderung in Entwicklungsprozessen sind die kritischen Dimensionen Zeit, Kosten und Qualität heranzuziehen. Ergebnisse von Analysen sind gemäß gängigen Methoden der Statistik quantitativ sowie grafisch aufzubereiten.

Anforderung 4. Modellierung. Um Entwicklungsprozesse grundsätzlich zu verstehen und Einflussanalysen zu erlauben, ist ein Konzept erforderlich, welches die Beschreibung von Methoden, Entwicklungsprozessen und Wirkzusammenhängen gestattet.

Anforderung 5. Durchgängigkeit. Die Beschreibung von Methoden- und Prozesswissen ist ohne einen Bruch zur notwendigen Modellierung für Einflussanalysen zu ermöglichen. Konkrete Modelle zur Einflussanalyse sind automatisiert aus der Beschreibung von Prozessen und Wirkzusammenhängen abzuleiten.

Anforderung 6. Wiederverwendung. Einmal formuliertes Wissen über Methoden, Entwicklungsprozesse und Wirkzusammenhänge ist auch für verwandte Entwick-

lungsprozesse eines Anwendungskontextes zur Verfügung zu stellen. Eine immer wieder neue Erfindung des Rades ist zu vermeiden.

Anforderung 7. Wirkzusammenhängen. Modelle von Wirkzusammenhängen zur Beschreibung des Prozessverhaltens unterscheiden sich je nach Anwendungskontext und des betrachteten Abstraktionsgrades. Die Beschreibung verschieden abstrakter Modelle von Wirkzusammenhängen ist zu unterstützen.

Anforderung 8. Methodik. Die Einflussanalyse erfordert die Auseinandersetzung mit den anwendungsspezifischen Gegebenheiten eines Entwicklungsprozesses. Hierzu ist eine Methodik notwendig, welche die Modellierung von Entwicklungsprozessen leitet und daraus Bewertungen des Mehrwerts von Methodenänderungen erlaubt.

Anforderung 9. Verfeinerung. In der Praxis ist es unrealistisch anzunehmen, zu Beginn einer Untersuchung alle notwendigen Daten zur Abschätzung des Prozessverhaltens bereits zur Verfügung zu haben. Vielmehr muss die Methodik eine schrittweise Verfeinerung von Modellen mit abstrakten Annahmen hin zu detaillierten Modellen unterstützen.

Anforderung 10. Werkzeugunterstützung. Die Modellierung von Methodenwissen und Wirkzusammenhängen sowie auch die Analyse von Entwicklungsprozessen erfordern eine entsprechende werkzeuggestützte Unterstützung, um effektiv und effizient zu Ergebnissen und Aussagen über den Mehrwert einer Änderung zu gelangen.

In den folgenden Abschnitten werden die eingangs genannten Themengebiete vorgestellt und ihr Bezug sowie ihre zweckmäßige Anwendbarkeit für diese Arbeit besprochen. Zudem werden die definierten Anforderungen auf ihre Erfüllung durch besprochene Ansätze geprüft, sofern sie einen entsprechenden Bezug aufweisen. In Abschnitt 2.9 werden die Erkenntnisse des Stands der Technik dann den Anforderungen gegenüber gestellt und der weitere Handlungsbedarf formuliert.

2.2 Methoden zur Leistungs- und Nutzenbewertung

Die Änderung eines Entwicklungsprozesses, beispielsweise durch die Integration einer neuen Methode oder eines neuen Werkzeugs, kann als Investition betrachtet werden. Eine Investition ergibt sich dabei aus möglichen Kosten für die Anschaffung der neuen Methode oder des neuen Werkzeugs und der mit der Einführung einhergehenden Kosten für etwa Lizenzen und Schulungen und damit verbundene Nutzen. Methoden im Themengebiet der Leistungs- und Nutzenbewertung versuchen nun die Vorteilhaftigkeit einer Investition oder Handlungsalternative mit anderen Alternativen zu vergleichen und auf dieser Basis eine Bewertung beziehungsweise Entscheidung für eine bestimmte

Investition zu treffen. Im Folgenden werden Methoden vorgestellt, die ein solches Problem aus verschiedenen Blickwinkeln betrachten.

Balazova entwickelt für die Anwendungsdomäne der Mechatronik eine Methode zur Leistungsbewertung und Leistungssteigerung (Balazova 2004). Die Methode betrachtet die drei Handlungsfelder Mensch, Organisation und Technik mit einem wesentlichen Einfluss auf die Leistung der Produktentwicklung. Die genannten Handlungsfelder werden in detailliertere Handlungselemente herunter gebrochen, für die jeweils Verbesserungen angestrebt werden. Beispiele für solche Handlungsfelder sind etwa die „Motivierung der Entwickler“, „Wissensmanagement“ und der „Einsatz von Entwicklungsmethoden“. Mögliche Verbesserungsmaßnahmen sollen dabei durch einen Vergleich des aktuellen Zustands gegenüber einem idealen Zustand für den jeweiligen Anwendungskontext im Unternehmen identifiziert werden.

Für eine Leistungsbewertung wird zunächst das Soll-Profil der Produktentwicklung in den Handlungsfeldern erfasst. Dabei werden die Handlungselemente ausgewählt, die einen großen angenommenen Einfluss auf die Leistung der Produktentwicklung besitzen und gleichzeitig einen hohen Beitrag zur Erfüllung von Zielen in den Dimensionen Zeit, Kosten und Qualität liefern. Dazu wird der Einfluss eines jeden Handlungselements auf andere Handlungselemente in Form einer Matrix beschrieben, bei der die betrachteten Handlungselemente in die Zeilen- beziehungsweise Spaltenköpfe eingetragen werden. Die Stärke des Einflusses wird durch Expertenbefragung auf einer Skala von 1 für einen schwachen Einfluss bis 3 für einen starken Einfluss bewertet. Die dann folgende Berechnung der Zeilen- und Spaltensummen erlaubt einen Überblick über die Größe des Einflusses von Handlungselementen aufeinander. Analog zur Bewertung des Einflusses erfolgt auch die Bewertung des Zielbeitrags eines jeden Handlungselements in Form einer Matrix. Schließlich wird die Stärke des Einflusses eines Handlungselements dem Zielbeitrag gegenübergestellt und solche Handlungselemente gewählt, die einen großen Einfluss wie auch Zielbeitrag leisten. Für jedes ausgewählte Handlungselement wird die angestrebte Entwicklungsstufe (Leistungsniveau) qualitativ festgesetzt. Der damit bestimmte Soll-Zustand wird dann gegen den Ist-Zustand verglichen, welcher in einem Workshop mit Experten ermittelt wird. Schließlich werden auf Basis der Differenz der identifizierten Profile mögliche Zieloptionen für eine Leistungssteigerung untersucht.

Der Ansatz betrachtet also verschiedene Handlungselemente und deren Entwicklungsstufen für die Produktentwicklung in einem bestimmten Anwendungskontext und erlaubt die Ableitung von Handlungsalternativen für mögliche Verbesserungen. Interessant sind hier die Methoden zur Bewertung des Einflusses von Handlungselementen untereinander und die Wirkung auf anvisierte Ziele. Im Gegensatz zu dieser Arbeit wird allerdings vom eigentlichen Entwicklungsprozess mit seinen Wirkzusammenhängen

und der inhärenten Unsicherheit abstrahiert. Eine quantitative Einflussanalyse und Einflussbewertung für neue Methoden oder Werkzeuge in den Dimensionen Zeit, Kosten und Qualität wird damit also noch nicht zur Verfügung gestellt.

Das sogenannte Benefit Asset Pricing Model (BAPM) soll die Bewertung neuer Technologien oder Werkzeuge für die Produktentwicklung ermöglichen (Schabacker 2001). Dazu bedient es sich Methoden zur Berechnung der Rendite und des Risikos von Kapitalmarktanlagen. Bei diesem Ansatz werden zunächst die Unternehmensstrategie und die Prozessziele erfasst. Auf dieser Grundlage werden dann Werkzeuge oder Technologien gesucht und ausgewählt, die prinzipiell für die Erfüllung der gesetzten Ziele in Frage kommen. Im nächsten Schritt werden die potentiellen Nutzen einer Investitionsalternative in sogenannte Nutzenkategorien eingeordnet, welche verschiedene Aspekte der Produktentwicklung mit Blick auf die Dimensionen Zeit, Kosten und Qualität beschreiben. So würde der Nutzen „Steigerung der Produktivität“ etwa in die Nutzenkategorie „Prozessperformance“ eingeordnet werden können. Weiterhin werden die Nutzenkategorien Produktqualität, Werkzeugeinsatz, Mitarbeiterumfeld, Servicequalität und Projektperformance unterschieden. Dann werden die identifizierten Nutzen in gewichteten Nutzenklassen direkt, direkt schwer, indirekt, indirekt schwer zu quantifizierende Nutzen sowie Synergieeffekte bewertet. Ist ein Nutzen nicht monetär quantifizierbar, wird dieser durch Folgerungen und Vergleiche mittels Expertenwissen quantifiziert. Dabei werden für schwer quantifizierbare Nutzen unter anderem Erfahrungskurven auf Basis von durchgeführten Projekten und Ausfallwahrscheinlichkeiten herangezogen, um die mögliche Rendite eines Nutzens zu bestimmen. Als Methode zur Prognose entsprechender Renditen wird dabei die Szenario-Technik vorgeschlagen, bei der die Eintrittswahrscheinlichkeit eines bestimmten Umweltzustands mit einer bestimmten Rendite subjektiv durch einen Experten gewählt wird.

Der Ansatz erlaubt die Strukturierung möglicher Nutzen und deren Bewertung für eine Investitionsalternative. Dabei kann der Faktor der Unsicherheit von Entwicklungsprozessen durch Wahrscheinlichkeiten abgebildet werden und verschiedene Szenarien von Renditen auf Basis unterschiedlicher Nutzenbewertungen exploriert werden. Allerdings beschreibt der Ansatz weder Entwicklungsprozesse noch Wirkzusammenhänge und damit neue Methoden im Kontext eines Entwicklungsprozesses explizit. Bedingt durch manuelle Variationen von Eintrittswahrscheinlichkeiten und beispielsweise Erfahrungskurven ist eine weitere Dokumentation notwendig, da eine spätere Nachvollziehbarkeit von Annahmen und Ergebnissen sonst schwierig bis unklar bleibt.

Ein weiterer Ansatzpunkt der Nutzenbewertung findet sich im Bereich von Technologiebewertungen. So entwickelt etwa Kröll eine Methode zur Technologiebewertung für eine ergebnisorientierte Produktentwicklung (Kröll 2008). Der Begriff Technologie wird in diesem Zusammenhang als Wissen über Wege zur Lösung eines technischen

Problems verstanden. Zwar liegt der Fokus des Ansatzes statt auf Entwicklungsprozessen auf der Auswahl einer technischen Lösung für ein Produkt, doch ist die grundlegende Herangehensweise der Bewertung interessant. Für eine Bewertung werden zunächst abstrakte Bewertungskriterien wie etwa Kosten und Qualität auf kleinere Zielkriterien herunter gebrochen. Für das Kriterium Kosten sind dieses etwa Investitionskosten, Lizenzgebühren und Entwicklungskosten. Für jedes Zielkriterium wird auf Basis empirischer Daten beziehungsweise Expertenbefragung eine passende Wahrscheinlichkeitsverteilung gewählt und diese parametrisiert. Ergänzend zu Wahrscheinlichkeitsverteilungen werden Transformationsfunktionen eingesetzt, um einer zufälligen Ausprägung eine Maßzahl zuzuordnen, welche die Güte einer Ausprägung beschreibt, etwa als Zielerfüllung von 0% bis 100%. Schließlich werden die Bewertungskriterien aus den Zielkriterien aggregiert. Dazu werden durch Monte Carlo Simulation Zufallswerte gemäß den Wahrscheinlichkeitsverteilungen für jedes Zielkriterium bestimmt, diese transformiert und im Anschluss an eine Gewichtung aufaddiert. Die mehrfache Simulation erlaubt dann Aussagen über mehr oder weniger wahrscheinliche Ausprägungen eines Bewertungskriteriums. Über einen qualitativen Vergleich der Bewertungskriterien werden dann unterschiedliche Technologien bewertet.

Im Gegensatz zu den anderen vorgestellten Ansätzen werden hier Verteilungen für Zielkriterien wie etwa Qualität detailliert und der Einfluss der Unsicherheit solcher Eigenschaften für eine Bewertung betrachtet. Insgesamt beziehen sich die Bewertungen allerdings auf mögliche Lösungsalternativen und nicht auf den eigentlichen Entwicklungsprozess. Dementsprechend wird auch hier kein Ansatz zur Beschreibung von Entwicklungsprozessen und Wirkzusammenhängen geliefert, der für eine Einflussanalyse und Einflussbewertung von Methodenänderungen in Entwicklungsprozessen verwendet werden könnte.

Zusammenfassung

Die vorgestellten Arbeiten der Leistungs- und Nutzenbewertung befassen sich mit der Bewertung der Leistung und des Nutzens von Investitionsalternativen. Dazu werden entweder der Einfluss von Handlungselementen oder der Nutzen und das Risiko einer Alternative durch Expertenbefragung bewertet. Kernpunkte der Nutzen- und Risikobewertung sind hier die Verwendung von Wahrscheinlichkeitsverteilungen auf Basis von Expertenwissen oder empirischen Daten und darauf aufsetzende Simulationen. Als Ergebnis werden dadurch fundiertere Entscheidungen auf Basis von Kennzahlen (**Anforderung 1**) ermöglicht, die über klassische Methoden der Investitions- oder Wirtschaftlichkeitsrechnung hinausgehen, vergleiche (Götze 2006). In einer ersten Näherung sind diese Methoden also bereits für eine Betrachtung der in **Anforderung 2** (Unsicherheit) angesprochenen Unsicherheit von Entwicklungsprozessen verwendbar. Dennoch abstrahieren die genannten Arbeiten weit von der Beschreibung von Ent-

wicklungsprozessen und deren Wirkzusammenhängen, so dass eine Betrachtung des Verhaltens von Entwicklungsprozessen nicht ermöglicht wird und die Bewertung von Änderungen eher auf einer abstrakteren Ebene verbleibt.

2.3 Methoden zur Modellierung von Projekten

Methoden der Leistungs- und Nutzenbewertung fehlt es an einer Beschreibung der Arbeit, die während einer Entwicklung durchgeführt wird. Es existieren zahlreiche Methoden zur Modellierung von Projekten, welche eben dazu verwendet werden, den Ablauf einer Entwicklung zu beschreiben. In diesem Abschnitt wird ein Überblick über entsprechende Methoden der Modellierung gegeben und mögliche Analysen vorgestellt.

Unter dem Begriff Projektmanagement ist die Gesamtheit von Führungsaufgaben, Organisation, Techniken und Mitteln für die Initiierung, Definition, Planung, Steuerung und den Abschluss von Projekten zu verstehen (DIN 69901-5:2009-01). Ein Projekt ist dabei im Wesentlichen ein einmaliges Vorhaben, welches sich durch eine Begrenzung der Rahmenbedingungen Zeit, Kosten und Qualität auszeichnet (Burghardt 2006). Kernpunkte des Projektmanagements sind dementsprechend insbesondere die Planung und Steuerung von Projekten und damit der Faktoren mit Einfluss auf Zeit, Kosten und Qualität als wichtige Kennzahlen.

Wie in Abbildung 2.2 dargestellt ist, stehen sich die genannten Dimensionen gegenüber und beeinflussen sich gegenseitig. So bedingt beispielsweise eine Reduzierung der notwendigen Entwicklungsdauer bei gleichbleibender Qualität einen Anstieg der geplanten Entwicklungskosten, da zum Beispiel mehr Mitarbeiter eingesetzt werden müssen.

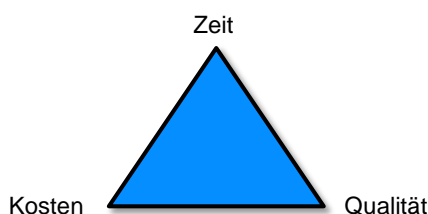


Abbildung 2.2 Die Projektdimensionen Zeit, Kosten und Qualität

Im Gegensatz zu einem Projekt, insbesondere einem Entwicklungsprojekt, beschreibt ein Modell eines Entwicklungsprozesses die notwendigen Aktivitäten beziehungsweise Aufgaben zur Durchführung eines Projektes mit dem Ziel ein Produkt zu realisieren. Eine Beschreibung eines Entwicklungsprozesses ist dabei allgemeiner als ein konkreter Projektablauf zu sehen und kann daher für mehrere vergleichbare Entwicklungsvorhaben wiederverwendet werden. Entsprechend dieser Beziehung sind die genannten Dimensionen und Einflüsse auf diese Dimensionen von besonderem Interesse, um den Einfluss einer Änderung auf einen Entwicklungsprozess zu bewerten.

Im Folgenden werden Methoden der Netzplantechnik zur Beschreibung von Projektabläufen und zur Planung und Kontrolle von Projekten beschrieben und ihre Eignung für die Modellierung (**Anforderung 4**) und die Analyse (**Anforderung 3**) von Entwicklungsprozessen untersucht.

2.3.1 Netzplantechnik

Der Begriff Netzplantechnik subsummiert eine ganze Reihe von Methoden, die im Projektmanagement für die Planung und Kontrolle von Projekten eingesetzt werden. Die Netzplantechnik erlaubt die grafische Beschreibung von Abläufen in einem Projekt. Dabei wird der Begriff Vorgang verwendet, um die notwendige Arbeit zu beschreiben – synonym werden auch die Begriffe Aufgabe und Aktivität verwendet. Im Allgemeinen besteht ein Netzplan dabei aus Knoten und Kanten, welche die Knoten miteinander verbinden. Unterschieden werden Netzpläne in

- Vorgangspfeil-Netzpläne, bei denen Aktivitäten durch Pfeile dargestellt werden und Knoten als Ergebnisse oder Meilensteine verstanden werden,
- Vorgangsknoten-Netzpläne, bei denen Aktivitäten durch Knoten repräsentiert werden, und der Ablauf durch Kanten beschrieben wird sowie
- Ereignisknoten-Netzpläne, bei denen Aktivitäten durch Kanten und auftretende Ereignisse durch Knoten beschrieben werden.

Weiterhin können zwei grundlegende Arten von Methoden der Netzplantechnik unterschieden werden:

- deterministische Methoden und
- stochastische Methoden.

Deterministische Methoden beschreiben einen festgelegten Projektablauf bei dem die Reihenfolge einzelner Aktivitäten vorgeben wird. Eine Änderung oder Abweichung der Folge bestimmter Aktivitäten ist bei solchen Methoden für die initiale Planung des Ablaufes nicht vorgesehen. Zu diesen Methoden zählen etwa die Critical Path Method (CPM) und die Program Evaluation and Review Technique (PERT).

Auf der anderen Seite werden stochastische Methoden zur Beschreibung von Abläufen verwendet, wenn die Folge möglicher Aktivitäten nicht eindeutig planbar ist. Wahrscheinlichkeiten bestimmen hierbei die Auswahl von folgenden Aktivitäten. Zu diesen Methoden, die auch als Entscheidungsnetzpläne bezeichnet werden, zählt beispielsweise die Graphical Evaluation and Review Technique (GERT).

Im Folgenden werden CPM, PERT und GERT als Vertreter der Netzplantechnik kurz vorgestellt. Für einen weiteren Einblick sei auf entsprechende Literatur wie (Burghardt 2006) und (Corsten und Corsten 2000) verwiesen.

Critical Path Method

Mitte der 1950er Jahre wurde die Critical Path Method in den USA zur Planung von Investitionsvorhaben und Instandhaltungsarbeiten entwickelt. Als Methode der deterministischen Netzplantechnik werden dabei Abläufe als Vorgangspfeil-Netzplan modelliert, also Aktivitäten als Pfeil beziehungsweise Kante dargestellt.

Dabei erhalten alle Kanten einen eindeutigen Bezeichner zur Beschreibung der durchzuführenden Aktivität sowie eine konstante Dauer der Aktivität zugeordnet. Die Angabe der Dauer einer Aktivität basiert in der Regel auf Expertenschätzungen. Seltener existieren statistisch verwertbare Daten, die hierfür herangezogen werden können. Jeder Knoten verfügt über vier Angaben:

- einen eindeutigen Bezeichner zur Identifikation des Knotens,
- den frühesten Beginn der folgenden Aktivität,
- den spätesten Beginn der folgenden Aktivität und
- die Pufferzeit zwischen dem frühesten und spätesten Beginn.

Der früheste Beginn einer Aktivität wird dabei durch die sogenannte Vorwärtsrechnung bestimmt. Dazu wird der früheste Beginn einer Aktivität aus jeweils dem frühesten Beginn vorhergegangener Aktivitäten und deren Dauer berechnet. Analog dazu erfolgt die Berechnung des spätesten Beginns einer Aktivität durch die Rückwärtsrechnung. Von der letzten Aktivität ausgehend, wird die Dauer einer Aktivität vom spätesten Beginn einer Aktivität subtrahiert. Die Pufferzeit ergibt sich dann aus der Differenz zwischen dem spätesten und dem frühesten Beginn.

Abbildung 2.3 zeigt die Anwendung der Critical Path Method am Beispiel eines Entwicklungsprozesses für ein System bestehend aus Hardware und Software. Der kritische Pfad ergibt sich aus der Folge der Aktivitäten bei denen der folgende Puffer Null ist. Der kritische Pfad läuft durch die Aktivitäten der Software-Entwicklung im unteren Bereich der Abbildung und ist fett markiert.

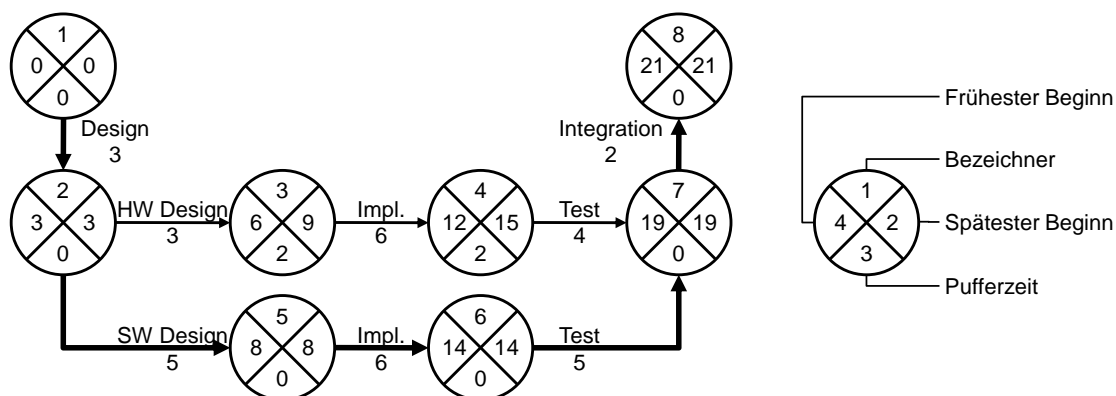


Abbildung 2.3 Notation der Critical Path Method nach (Burghardt 2006)

Die vorgestellte Notation kann um weitere Elemente zur Beschreibung von Aktivitäten erweitert werden. Solche Erweiterungen umfassen beispielsweise das früheste Ende einer Aktivität und das späteste Ende einer Aktivität.

Program Evaluation and Review Technique

Ende der 1950er Jahre wurde im Rahmen des Raketenprojekts Polaris die Program Evaluation and Review Technique in den USA entwickelt (Miller 1965). Als Ereignisknoten-Netzplan liegt der Fokus bei dieser Methode auf den Ereignissen, die eine Aktivität begrenzen.

Der wesentliche Unterschied zwischen dem Vorgangspfeil-Netzplan CPM und dem Ereignisknoten-Netzplan PERT liegt in der Angabe der notwendigen Dauer einer Aktivität. So wird allein bei der Program Evaluation and Review Method unter den Methoden der Netzplantechnik die Dauer einer Aktivität über drei Schätzungen vorgenommen:

- die **optimistische** (geringste) Dauer einer Aktivität,
- die **wahrscheinliche** Dauer einer Aktivität und
- die **pessimistische** (längste) Dauer einer Aktivität.

Die erwartete Dauer einer Aktivität wird dann gemäß der Beta-Verteilung berechnet, wobei die optimistische Dauer (t_o) und die pessimistische Dauer (t_p) jeweils einfach und die wahrscheinliche Dauer (t_w) vierfach gewichtet wird.

$$t_E = \frac{(t_o + 4t_w + t_p)}{6}$$

Dieses Vorgehen erlaubt eine genauere Schätzung als eine einfache Punktschätzung, die bei anderen Methoden verwendet wird, erfordert bei Anpassungen aber einen entsprechend größeren Aufwand zur Überarbeitung der Schätzungen (Burghardt 2006).

Die Notation ist ähnlich der Notation von CPM. Dabei wird das Ereignis im Knoten explizit beschrieben und Pfeile erhalten Angaben zu den Schätzwerten der optimistischen, wahrscheinlichen und pessimistischen Zeit, sowie die daraus berechnete erwartete Dauer der Aktivität.

Graphical Evaluation and Review Technique

Die Mitte der 1960er Jahre entwickelte Graphical Evaluation and Review Technique (GERT) zählt zu den stochastischen Entscheidungs-Netzplänen. Die Folge von Aktivitäten wird bei dieser Methode entweder probabilistisch oder deterministisch durch verschiedene grafische Elemente beschrieben. Im Gegensatz zu den anderen Methoden sind bei der Graphical Evaluation and Review Technique auch Rücksprünge zu vorherigen Aktivitäten möglich (Pritsker 1966).

Abbildung 2.4 zeigt die bei GERT möglichen Knotentypen. Die verschiedenen logischen Eingänge können mit entweder probabilistischen oder deterministischen Ausgängen kombiniert werden, wobei die erste Aktivität im Ablauf lediglich aus einem Ausgang und die letzte Aktivität aus einem Eingang besteht. Aktivitäten werden durch eine Kante zwischen den Knoten abgebildet und erhalten eine bestimmte Wahrscheinlichkeit (p_a) beziehungsweise eine Dauer (t_a) zugeordnet.

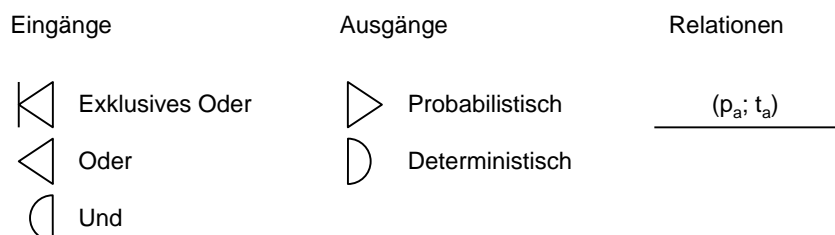


Abbildung 2.4 Notation der Graphical Evaluation and Review Technique nach (Pritsker 1966)

GERT ermöglicht durch die verschiedenen Knotentypen also die Definition von Ablaufregeln, welche die nächste Aktivität entweder deterministisch oder probabilistisch auswählen. Der deterministische Ausgang bestimmt dabei alle ausgehenden Kanten mit derselben Wahrscheinlichkeit – Aktivitäten werden parallel ausgeführt. Demgegenüber teilen sich die Wahrscheinlichkeiten für Kanten, die aus einem probabilistischen Knoten ausgehen auf. Hierbei existiert immer nur eine folgende Aktivität (Kante).

Zur Analyse von Netzplänen der Graphical Evaluation and Review Technique werden beispielsweise Simulation (Pritsker und Whitehouse 1966) oder auch analytische Methoden mit Hilfe einer Abbildung auf Markov-Ketten eingesetzt (Neumann 1979), (Nicolai 1980). Zu Markov-Ketten vergleiche auch Abschnitt 2.7.1.

2.3.2 Analysen der Netzplantechnik

In der Netzplantechnik wird der früheste Beginn einer jeden Aktivität durch die sogenannte Vorwärtsrechnung berechnet. Dabei wird der jeweilige früheste Beginn aus dem frühesten Beginn der letzten Aktivität und ihrer Dauer als Summe bestimmt.

Für die Berechnung des spätesten Beginns einer jeden Aktivität wird die Rückwärtsrechnung verwendet, bei der vom spätesten Beginn der letzten Aktivität die jeweilige Dauer einer Aktivität subtrahiert wird.

Die Differenz zwischen spätestem Beginn und frühestem Beginn ergibt dann die Pufferzeit einer Aktivität. Die Pufferzeit beschreibt damit die maximale Dauer einer Aktivität, um die Zeitplanung des Projektendes nicht zu gefährden.

Der kritische Pfad eines Netzplans beschreibt die Folge von Aktivitäten, welche das Projektende bestimmt. Eine Zeitüberschreitung von Aktivitäten auf diesem Pfad führt in der Konsequenz zu einer Verschiebung des Projektendes.

2.3.3 Zusammenfassung

Die vorgestellten Methoden der Netzplantechnik erlauben die grundlegende Beschreibung von Projektablaufen. Dabei werden im Allgemeinen die Dauer und die Folge von Aktivitäten betrachtet. Auch Ressourcen werden in der Praxis mit entsprechenden Werkzeugen modelliert, was die Berechnung der notwendigen Einsatzmittel und der damit verbundenen Kosten gestattet. Analysen ermöglichen Aussagen über die aktuell zu erwartete Projektdauer, die Projektkosten, sowie die Aufdeckung kritischer Pfade.

Die deterministischen Methoden beschreiben starre Abläufe, bei denen alle Aktivitäten in der vorgegebenen Reihenfolge durchgeführt werden. Iterationen zur Überarbeitung von Design Artefakten werden nicht beachtet. Probabilistische Methoden ermöglichen hingegen die Beschreibung von mehr oder weniger wahrscheinlichen Abläufen, bei dem die Reihenfolge der Aktivitäten nicht eindeutig vorhergesehen werden kann.

Mit keiner der genannten Methoden können Entwicklungsmethoden und Prozesselemente wie etwa Design Artefakte oder Rollen modelliert werden. Auch Wirkzusammenhänge werden bei den genannten Methoden nicht betrachtet. Somit sind diese Methoden für eine detaillierte Beschreibung von Entwicklungsprozessen (**Anforderung 4**, Modellierung) nicht geeignet. Die **Anforderung 2** (Unsicherheit) kann durch probabilistische Methoden unterstützt werden, wobei für eine Analyse beispielsweise Markov-Ketten oder Simulation eingesetzt werden. Die **Anforderung 3** (Analyse) wird in Bezug auf die Dimensionen Zeit beziehungsweise Kosten teilweise beachtet, wobei der Fokus auf einzelnen Projekten liegt. Die Dimension Qualität wird nicht explizit abgebildet und geht allenfalls als Fertigstellungsgrad in die Betrachtung der Dauer einer Aktivität ein.

Da die Methoden der Netzplantechnik offenbar nur sehr begrenzt in der Modellierung von Entwicklungsprozessen geeignet sind, werden im folgenden Abschnitt Methoden untersucht, welche für eine detailliertere Prozessmodellierung verwendbar sind.

2.4 Methoden zur Modellierung von Prozessen

Das Themengebiet der Prozessmodellierung befasst sich mit der Beschreibung von Prozessen. Zu den Anwendungsgebieten von Prozessmodellen zählen die Dokumentation, die Analyse oder auch die automatische Ausführung von Prozessen als sogenannte Workflows. Damit nehmen diese Methoden einen anderen Blickwinkel als die Methoden der Modellierung von Projekten ein, so dass eher die Dokumentation und

Kommunikation gegenüber einer Planung im Vordergrund steht. Entsprechend vielfältig sind die existierenden Modelle und Sprachen zur Abbildung von Prozessen.

Im Folgenden werden verschiedenen Methoden vorgestellt, die für eine Beschreibung von Entwicklungsprozessen eingesetzt werden können. Im Wesentlichen werden mit Hilfe dieser Methoden qualitative Analysen zur Struktur von Prozessen und im Speziellen von Entwicklungsprozessen ermöglicht.

2.4.1 Structured Analysis and Design Technique

Die Structured Analysis and Design Technique (SADT) wurde Ende der 1970 Jahre zur Beschreibung beliebiger Strukturelemente und deren Beziehungen zueinander entwickelt (Ross 1977). Die grafische Methode erlaubt die hierarchische Dekomposition von relevanten Ausschnitten eines betrachteten Systems. Mit einem Blick auf Entwicklungsprozesse können Aufgaben und Aktivitäten ineinander geschachtelt und ihr prinzipieller Ablauf modelliert werden. Der Fokus der Methode liegt dabei auf der Kommunizierbarkeit zwischen Modellierer und Anwender eines Modells etwa für die Analyse des abgebildeten Systems.

Die Methode umfasst fünf Strukturelemente, die im Folgenden auf die Elemente in Entwicklungsprozessen abgebildet werden. Abbildung 2.5 veranschaulicht diese Konzepte als Anwendung auf Entwicklungsprozesse.

- **Knoten** beschreiben **Aktivitäten**,
- Aktivitäten werden durch **Eingaben** und **Ausgaben** wie beispielsweise Design Artefakte miteinander verbunden und durch Kanten dargestellt,
- **Mechanismen** beschreiben unterstützende oder notwendige Elemente, die für die Durchführung einer Aktivität erforderlich sind, wie etwa Ressourcen, und
- **Kontrollen** erlauben die Beschreibung von Rahmenbedingungen für die Durchführung einer Aktivität.

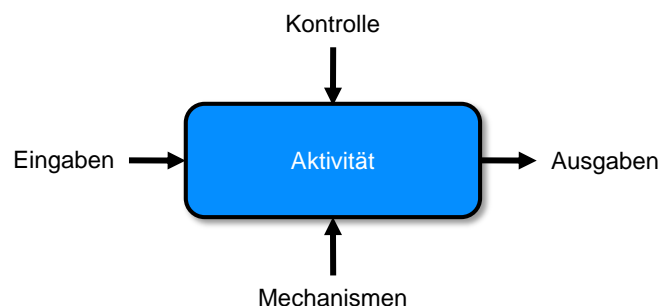


Abbildung 2.5 Konzepte der Structured Analysis and Design Technique nach (Ross 1977)

Durch die Generizität der Structured Analysis and Design Technique lassen sich zwar Entwicklungsprozesse beschreiben, doch bleibt ein Interpretationsspielraum bestehen.

Da der Ablauf von Aktivitäten und das Verhalten von Aktivitäten nicht explizit modelliert werden, sind in dieser Form auch keine Analysen zum möglichen Ablauf von Aktivitäten möglich. Auch wird nicht zwischen Aktivitäten und atomaren Aufgaben differenziert. Zudem formalisiert der Ansatz nicht in der Weise, als dass entstehende Modelle für quantitative Analysen genutzt werden könnten. Auswirkungen von Änderungen, beispielsweise auf den Aufwand einer Aktivität, können in dieser Form keine Aussagen getroffen werden. Analysen beschränken sich damit auf die Struktur und Beziehungen von Aktivitäten.

2.4.2 Integration Definition

Unter den Begriff Integration Definition fallen eine Menge von Methoden zur Modellierung und Analyse von Systemen, Anforderungen und Prozessen. Diese Methoden wurden vom US amerikanischen National Institute of Standards and Technology (NIST) Anfang bis Mitte der 1990er Jahre als Standard spezifiziert.

IDEF0 „Integrated Definition for Function Modeling“ spezifiziert ein Modell zur Beschreibung von Funktionen, Aktivitäten und Entscheidungen innerhalb von Organisationen und Systemen (IDEF 1993). Die Methode basiert im Wesentlichen auf der bereits vorgestellten SADT Methode und zielt auf die Kommunikation zwischen Modellierer und Anwender sowie auf die Analyse solcher Modelle. So ist auch ein IDEF0 Modell nicht zur Beschreibung von Abläufen von Entwicklungsprozessen intendiert.

IDEF1 „Integrated Definition for Information Modeling“ beschreibt ein Modell zur strukturellen und semantischen Abbildung von Informationen und Anforderungen für eine Organisation (Mayer 1992). Dabei wird die Methode für die Identifikation von aktuell gemanagten Informationen, die Analyse von Problemen im Informationsfluss und die Beschreibung von Soll-Informationen verwendet. Bei dieser Methode werden Entitäten mit ihren Attributen und Relationen unterschieden und grafisch modelliert. Resultierende Diagramme ähneln entsprechenden Entity Relationship (ER) Diagrammen. Als Erweiterung dieser Methode dient IDEF1X zur logischen Modellierung von relationalen Datenbanken.

IDEF3 „Process Description Capture Method“ spezifiziert eine Methode, um Prozesse inklusive dynamischer Aspekte wie Verhalten zu beschreiben (Mayer et al. 1995). Die Methode zielt dabei im Wesentlichen auf die initiale Dokumentation vorhandener Prozesse und beschreibt dazu Prioritäten und kausale Zusammenhänge zwischen Ereignissen. Insbesondere soll die Methode dazu dienen, Modelle für die Simulation zu erstellen. Bei der Modellierung wird in die Beschreibung von Abläufen von Aktivitäten und die Transformation von Objekten unterschieden.

Neben der grafischen Modellierung werden für jede der genannten Methoden verschiedene Formulare zur Verfügung gestellt, um Entitäten der Modelle, ihr Verhalten und ihre Beziehungen zu anderen Entitäten textuell zu beschreiben. Insgesamt wird somit umfangreiches Material auch für die Beschreibung von Prozessen bereitgestellt. Allerdings bleibt eine syntaktische und semantische Strukturierung über die relativ einfachen Modelle hinaus dem Anwender der Methoden überlassen. Für diese Arbeit ist eine strukturierte Beschreibung von Methoden und Prozessen erforderlich, um relevante Prozesselemente wie etwa Aktivitäten, Aufgaben, Rollen und Produkte mit ihren möglichen Beziehungen eindeutig zu beschreiben, ohne einen breiten Interpretationsspielraum zuzulassen. Schließlich sollen unterschiedliche Szenarien von Entwicklungsprozessen über die Dimensionen Zeit, Kosten und Qualität verglichen werden, was zunächst keine der genannten Methoden unterstützt.

2.4.3 Ereignisgesteuerte Prozessketten

Die Methode der Ereignisgesteuerten Prozessketten (EPK) wurde Anfang der 1990er Jahre mit dem Ziel der semantischen Prozessmodellierung und zur Analyse integrierter Informationssysteme aus Sicht der betrieblichen Ebene an der Universität des Saarlandes entwickelt (Keller et al. 1992). Die Ereignisgesteuerten Prozessketten sind ein zentraler Bestandteil der Architektur Integrierter Informationssysteme (ARIS) (Scheer und Jost 2002) und des SAP R/3 Modellierungskonzeptes (Krallmann 2007).

Als grafische Modellierungssprache für Geschäftsprozesse unterscheidet EPK im Wesentlichen in Ereignisse und Funktionen, die alternierend den Ablauf eines Prozesses beschreiben und durch Pfeile miteinander verbunden werden. Durch unterschiedliche Verknüpfungsoperatoren werden mögliche Pfade eines Geschäftsprozesses modelliert. Die grundlegenden Konzepte sind:

- **Funktionen** beschreiben die Durchführung eines betrieblichen Vorgangs bei dem **Eingaben** in **Ausgaben** transformiert werden, um ein bestimmtes Ziel zu erreichen.
- **Ereignisse** beschreiben einen eingetretenen (betriebswirtschaftlichen) Zustand und können Funktionen auslösen. Ein modellierter Geschäftsprozess beginnt und endet mit einem Ereignis.

Es werden die drei Verknüpfungsoperatoren AND als konjunktive Verknüpfung, OR als adjunktive und XOR als disjunktive Verknüpfung unterschieden. Dabei dürfen allerdings Ergebnisse nicht durch OR und XOR mit mehreren Funktionen verbunden werden.

Mit den sogenannten erweiterten Ereignisgesteuerten Prozessketten (eEPK) werden weitere Konzepte zur Beschreibung von Elementen der Organisation, Anwendungen, verwendete Daten, Informationsträger und Sachmittel als grafische Notation eingeführt

(Krallmann 2007). Durch diese Erweiterungen soll eine detailliertere Modellierung von entsprechenden Prozessen ermöglicht werden. Dabei werden die genannten Konzepte jeweils an entsprechende Funktionen annotiert und können als „Funktion verwendet / verändert [Element]“ gelesen werden.

Im Vergleich zu den Methoden SADT und IDEF wird bei EPK eine weitere Strukturierung von Elementen mit Blick auf Syntax und Semantik vorgenommen. Allerdings handelt es sich auch bei den EPK im Wesentlichen um eine Sprache zur grafischen Beschreibung von Prozessabläufen, die keine Beschreibung des Verhaltens von Funktionen beziehungsweise Aktivitäten erlaubt. Zudem sind die Ereignisgesteuerten Prozessketten vornehmlich im deutschsprachigen Raum verbreitet und unterliegen keinem Standard.

2.4.4 Business Process Model and Notation

Seit Anfang der 2000er Jahren wird mit der Business Process Modeling Notation (BPMN) ein Metamodell für die (grafische) Beschreibung von Geschäftsprozessen entwickelt. Heute wird die BPMN unter dem neuen Namen Business Process Model and Notation von der Object Management Group (OMG) als Standard weiterentwickelt (OMG 2009b). Dabei liegt der Schwerpunkt von BPMN auf der Unterstützung der Kommunikation zwischen Prozessanalysten, Prozessentwicklern und Prozessverantwortlichen, indem eine Syntax und eine Semantik für die Beschreibung von Geschäftsprozessen spezifiziert werden. In der aktuellen Version 2.0 (OMG 2011) wird die Notation um die textuelle Beschreibung einer grundlegenden Ausführungssemantik für Prozesse erweitert. Damit wird auf eine Übersetzung von BPMN in Workflowsprachen wie die Business Process Execution Language (BPEL) oder XML Process Definition Language (XPDL) zur automatischen Ausführung von Prozessen gezielt (Allweyer 2009).

In BPMN werden verschiedene Elemente für eine detaillierte Beschreibung von Geschäftsprozessen definiert, welche in die fünf folgenden Kategorien eingeordnet werden können:

- **Flow Objects** dienen zur Beschreibung eines Prozessablaufs. Zu den Elementen zählen dabei Ereignisse, Aktivitäten, die in Aufgaben und Unterprozesse unterschieden werden, sowie Gateways. Alle diese Elemente und Spezialisierungen davon werden als Knoten dargestellt.
- **Data Objects** werden in drei weitere Datenobjekte spezialisiert zu denen Data Inputs, Data Outputs und Data Stores zählen. Somit können Eingaben und Ausgaben von Aufgaben unterschieden werden.
- **Connecting Objects** beschreiben Verbindungen und Beziehungen zwischen Flow Objects oder anderen Elementen. Unterschieden werden hier der Ablauf zwischen

Aktivitäten, der Nachrichtenfluss, Beziehungen zwischen Daten und weitere Beziehungen. Diese Verbindungen werden als Kanten dargestellt.

- **Swimlanes** gruppieren die zuvor genannten Flow Objects in Pools oder untergeordnete Lanes und beschreiben als Container die Zugehörigkeit eines Flow Objects zu einem Prozessbeteiligten wie etwa einem Benutzer oder einer Rolle.
- **Artifacts** ermöglichen die Beschreibung weitergehender Informationen. Zu den Artefakten zählen Gruppen zur grafischen Zusammenfassung von Elementen und Annotationen zur Kommentierung von Elementen oder Beziehungen. Artifacts sind nicht mit Design Artefakten in der Produktentwicklung zu verwechseln.

Mit BPMN steht im Vergleich zu EPKs ein Standard zur Verfügung, der eine detaillierte Beschreibung von Geschäftsprozessen ermöglicht. Mit einem Blick auf die zuvor vorgestellten Methoden SADT und IDEF geht BPMN erheblich weiter ins Detail und spezifiziert unterschiedliche Modellierungselemente mit Syntax und Semantik. Für eine genauere und eindeutige Prozessbeschreibung ist BPMN daher deutlich besser geeignet. Für eine Beschreibung von Entwicklungsprozessen und deren Verhalten bleiben dennoch Schwächen bestehen, da im Wesentlichen auf den Ablauf eines Prozesses fokussiert wird. Wissen und Informationen etwa zu der Frage, wie und mit welchen Ressourcen eine Aufgabe durchgeführt wird, können nicht oder nur eingeschränkt abgebildet werden.

2.4.5 Software & Systems Process Engineering Meta-Model

Das Software & Systems Process Engineering Meta-Model (SPEM) wird seit Anfang der 2000er Jahre von der Object Management Group (OMG) entwickelt. Heute liegt SPEM in der Version 2.0 als Standard der OMG vor. Das Metamodell erlaubt unter anderem die Modellierung, Dokumentation und Beschreibung von Methoden und Entwicklungsprozessen (OMG 2008). Damit verfügt SPEM über fundierte Konzepte zur Beschreibung von Wissen über Entwicklungsprozesse, welche über die bereits vorgestellten Methoden deutlich hinausgehen.

Zur Modellierung von Methoden und Prozessen nimmt SPEM eine strikte Trennung vor. Durch diese Trennung von Methoden- und Prozessbeschreibung wird eine Wiederverwendbarkeit definierter Methoden in konkreten Prozessen erreicht, also eine Bibliothek über Methodenwissen angelegt, die stetig erweitert werden kann.

Abbildung 2.6 zeigt die wesentlichen Kernkonzepte von SPEM zur Beschreibung von Methoden und Prozessen. Dabei werden Methodenbeschreibungen unter dem Begriff *Method Content* und Prozessbeschreibungen unter dem Begriff *Process* zusammengefasst. Diese Konzepte sind:

- **Task Definition** beschreibt eine atomare Aufgabe, die durch eine oder mehrere Rollen durchgeführt wird und Design Artefakte entwickelt oder bearbeitet. Eine

Aufgabe kann zudem mit notwendigen Werkzeugen assoziiert werden. In konkreten Prozessen (**Process**) wird mit Hilfe des Konzepts **Task Use** auf eine Task Definition verwiesen. Weiterhin können durch das Konzept **Activity** hierarchische Work Breakdown Strukturen (WBS) aufgebaut werden.

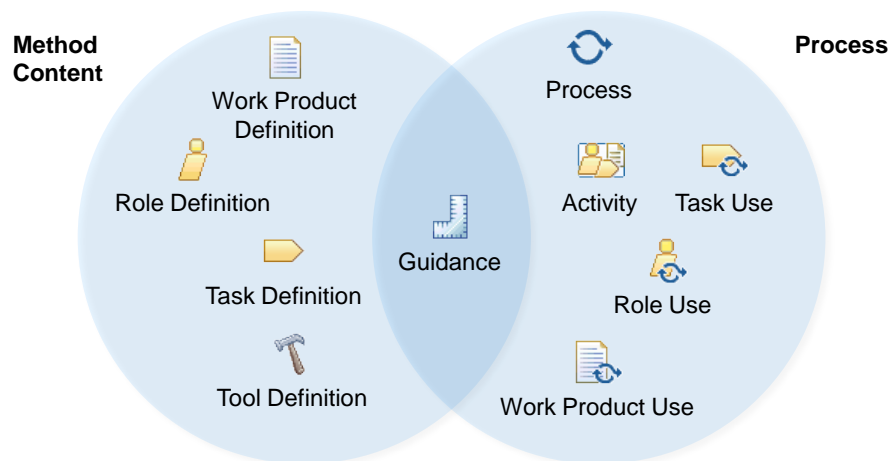


Abbildung 2.6 Methoden Definition und Prozess in SPEM 2.0 (OMG 2008)

- **Role Definition** beschreibt eine Rolle, welche eine bestimmte Aufgabe durchführt oder für ein Design Artefakt verantwortlich ist. Durch **Role Use** wird die Teilnahme einer Rolle in einem konkreten Prozess beschrieben.
- **Tool Definition** dient dazu Werkzeuge, die für die Durchführung einer Aufgabe erforderlich sind, zu beschreiben.
- **Work Product Definition** erlaubt die Beschreibung von Design Artefakten und Beziehungen zwischen solchen Artefakten. Ein Design Artefakt kann sowohl eine erforderliche Eingabe für Aufgaben oder auch ein Ergebnis einer Aufgabe sein. In Prozessen verweist das Konzept **Work Product Use** auf die entsprechende Definition.
- **Guidance** ermöglicht die Beschreibung verschiedener Hilfestellungen für alle zuvor genannten Konzepte. Dieses können beispielsweise Checklisten und Best Practices für Aufgaben oder Anleitungen für Werkzeuge sein.

Neben den genannten Konzepten verfügt SPEM 2.0 über weitere Konzepte zur detaillierten Beschreibung von Entwicklungsprozessen für die ein syntaktisches und semantisches Gerüst definiert wird.

Die Beschreibung des Wissens über Entwicklungsprozesse mit dem Software & Systems Process Engineering Meta-Model ermöglicht also im Gegensatz zu anderen Modellierungsansätzen das tiefere Verstehen von Prozessen, den Aufbau von Methoden- und Prozesswissen, sowie Best Practices. Somit kann eine Prozessmodellierung für detaillierte Analysen als bei den anderen Methoden verwendet werden.

2.4.6 Zusammenfassung

Die zunächst vorgestellten Methoden SADT und IDEF, die auch zur Beschreibung von Prozessen eingesetzt werden können, definieren allgemeine Konzepte zur Beschreibung von Abhängigkeiten. Auf dieser abstrakten Ebene haben konkrete Modelle das Problem, dass sie unterschiedlich interpretierbar sind. Weder dem Modellierer noch den Anwendern der entstehenden Modelle wird eine gemeinsame Basis geliefert, Entwicklungsprozesse eindeutig zu beschreiben und zu verstehen. Die **Anforderung 4** (Modellierung) wird daher nur sehr begrenzt erfüllt.

Die Methoden EPK und die BPMN werden im Wesentlichen für die Beschreibung von Geschäftsprozessen eingesetzt. Dabei fokussieren beide Ansätze auf die grafische Modellierung der Abläufe eines Prozesses. Mit der BPMN wird dieser Fokus auf die automatische Ausführung von Geschäftsprozessen mittels Workflows erweitert. Im Gegensatz zu den ersten Methoden werden hier schon weitere Konzepte der syntaktischen und semantischen Modellierung von Prozessen angeboten, so dass die **Anforderung 4** (Modellierung) schon besser getroffen wird.

Mit SPEM wird schließlich ein Standard bereitgestellt, der die Beschreibung von wiederverwendbaren Prozesselementen und feingranularen Methodenbeschreibungen ermöglicht. Eine Instanziierung von definierten Prozesselementen erlaubt die relativ schnelle Beschreibung von Entwicklungsprozessen und ein detailliertes Verstehen entsprechender Prozesse. Gerade im Kontext von Entwicklungsprozessen ist dieses Wissen für die Planung von Prozessen und die Anwendung von Methoden, Techniken und Werkzeugen von wichtiger Relevanz und für diese Arbeit nutzbar. Zusammenfassend erfüllt SPEM die **Anforderung 4** (Modellierung) und **Anforderung 6** (Wiederverwendbarkeit) am besten unter den untersuchten Methoden. Offen bleiben allerdings weiterhin die Modellierung von Wirkzusammenhängen und insgesamt das Problem der Einflussanalyse und Einflussbewertung.

Da unter den in diesem Abschnitt untersuchten Sprachen nur SPEM eine ausreichende Beschreibung von Methoden ermöglicht, werden im folgenden Abschnitt nochmal grundsätzliche Methoden zur Beschreibung von Methoden untersucht.

2.5 Methoden zur Beschreibung von Methoden

Die Disziplin des Method Engineering beschäftigt sich mit der Dokumentation und Beschreibung von Entwicklungsmethoden. Zudem zielt das Method Engineering auf die Entwicklung neuer Methoden und die Anpassung von Methoden, Techniken und unterstützenden Werkzeugen für die Entwicklung von Informationssystemen (Brinkkemper 1996), (Brinkkemper et al. 1996).

Ein wichtiger Aspekt des Method Engineerings ist dabei die Beschreibung des existierenden Methodenwissens, auf deren Basis neue Methoden und Werkzeuge entwickelt werden oder Methoden neu kombiniert werden. Das Problem der Beschreibung existierender Methoden und Werkzeuge ist damit auch für diese Arbeit von Bedeutung. Nur wenn existierende beziehungsweise angewendete Methoden beschrieben sind und verstanden werden, sind fundierte Verbesserungen an Entwicklungsprozessen möglich. Dieses zeigt sich etwa bei der Frage, welche Auswirkungen der Einsatz einer neuen Methode auf andere Methoden und damit auf den restlichen Entwicklungsprozess hat.

Der St. Gallerer Ansatz zur Beschreibung von Methoden versteht eine Methode als eine Anleitung für den gesamten Entwicklungsprozess eines Produktes (Heym 1993). Heym beschreibt das Method Engineering als Disziplin mit einem strukturierten Vorgehen welche die Teilbereiche des Methodenvergleichs und der Methodenharmonisierung, der Methodenentwicklung und der Methodenanpassung, sowie das Management des Entwicklung Knowhows behandelt (Heym 1995). Die durch das Method Engineering entstehenden Methoden bieten demnach effektive Lösungen für unternehmensspezifische Probleme und Entwicklungssituationen an. Heym entwickelt dazu ein Modell zur Spezifikation von Entwicklungsmethoden auf Grundlage einer sogenannten integrierten Methodenbank. Das Modell umfasst die Beschreibung von Aktivitäten, Ergebnissen, Rollen, Techniken und Metamodellen. Im Fokus steht dabei die Dokumentation von Entwicklungserfahrungen (Heym und Osterle 1992).

Karlsson entwickelt ein konzeptuelles Framework zur Betrachtung von Konfigurationen im Method Engineering (Karlsson 2003). Eine Konfiguration fasst eine Menge von Methoden für einen bestimmten Zweck, beispielsweise für die Anwendung auf ein bestimmtes Entwicklungsproblem, zusammen. Es werden zwei Typen von Konfigurationen unterschieden. Ein sogenanntes Configuration Package stellt dabei eine optimale Konfiguration für ein Produkt oder ein Projekt dar. Zur Beschreibung von situationsspezifischem Vorgehen werden Configuration Templates auf Basis der Configuration Packages beschrieben. Ziel des Ansatzes ist Definition von wiederverwendbaren Konfigurationen für bestimmte Arten von Entwicklungsprojekten.

Die Unified Method Architecture (UMA), ein Metamodell welches von IBM entwickelt wird, dient ebenso wie die zuvor genannten Ansätze zur Beschreibung von Methoden und Vorgehen (Gau 2006). Neben Aktivitäten, Meilensteinen, Rollen und Produkten können explizit Hilfestellungen und Kategorien zur Beschreibung von Methodenelementen und Prozesselementen modelliert werden (Haumer 2007). Wesentliche Konzepte von UMA sind in die Spezifikation des Software & Systems Process Engineering Metamodel (SPEM 2.0) (OMG 2008) eingegangen. SPEM 2.0 dient als Standard der Object Management Group (OMG) zur strukturierten Beschreibung von Methoden und

konkreten Prozessen für beliebige Vorgehensmodelle. Das Metamodell erlaubt außerdem die Beachtung von Konfigurationen für bestimmte Anwendungen.

Zusammenfassung

Die vorgestellten Ansätze des Method Engineering erlauben die Beschreibung und Modellierung von Methoden und deren Anwendung in konkreten Entwicklungsprozessen. Explizite Konzepte ermöglichen die semantische und strukturelle Beschreibung von Prozesselementen und erlauben damit das explizite Festhalten von Wissen in Form von Methodenbeschreibungen.

Für diese Arbeit wird die Beschreibung solchen Wissens als essentiell für Einflussanalysen von Methodenänderungen betrachtet. Ohne detailliertes Wissen über den existierenden Entwicklungsprozess und neue Methoden ist eine gezielte Untersuchung nicht verlässlich möglich. Alle vorgestellten Ansätze erfüllen grundsätzlich zu einem ansehnlichen Teil die **Anforderung 4** (Modellierung) in Bezug auf die Beschreibung von Methoden sowie **Anforderung 6** (Wiederverwendung). Allerdings bietet keiner der Ansätze eine Modellierung von Wirkzusammenhängen an und auch Einflussanalysen und Einflussbewertungen zu Methodenänderungen werden nicht ermöglicht.

2.6 Ansätze zur Analyse von Prozessen

Für die Analyse von Prozessen können grundlegend zwei Ansätze unterschieden werden. Auf der einen Seite stehen analytische Methoden, welche die mathematische Untersuchung von Zusammenhängen in Entwicklungsprozessen erlauben. Demgegenüber stehen Methoden welche die Beschreibung von detaillierten Wirkzusammenhängen erlauben, welche dann durch Simulation exploriert werden können.

Abbildung 2.7 zeigt den Zusammenhang zwischen dem notwendigen Aufwand zur Modellierung von Entwicklungsprozessen und deren Verhalten, welches durch Wirkzusammenhänge charakterisiert wird. Mit der Modellierung detaillierter Wirkzusammenhänge steigen das Wissen über den betrachteten Entwicklungsprozess beziehungsweise der eingesetzten Methoden und deren Zusammenwirken. Es ist offensichtlich, dass für eine detailliertere Modellierung mehr Aufwand erbracht werden muss, um entsprechende Modelle eines Entwicklungsprozesses aufzubauen. Damit ist auch klar, dass die Modellierung von Entwicklungsprozessen und Wirkzusammenhänge immer im Anwendungskontext betrachtet werden muss und die zu untersuchenden Fragestellungen die Granularität von Modellen bestimmen.

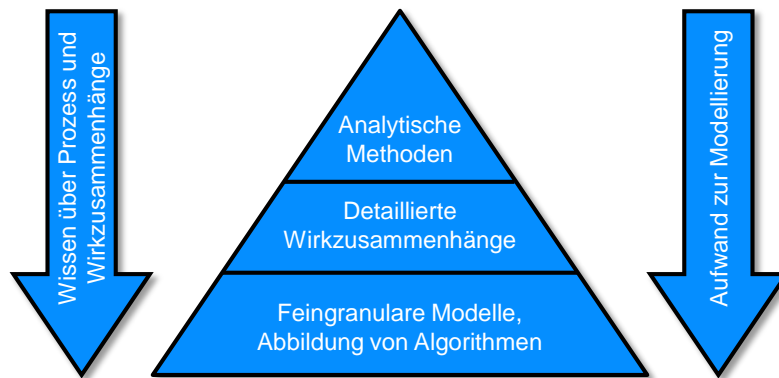


Abbildung 2.7 Pyramide des Wissens über Wirkzusammenhänge und der dazu notwendige Aufwand zur Modellierung und Analyse dieser Wirkzusammenhänge

Analytische Methoden

In der obersten Ebene der Pyramide finden sich analytische Methoden. Analytische Methoden liefern exakte Ergebnisse und eignen sich zur Untersuchung von Problemen, wenn dazugehörige Modelle adäquat genug bestimmt werden können. Dieses ist im Allgemeinen der Fall, wenn Zusammenhänge zwischen unterschiedlichen Einflussfaktoren auf relativ einfache Weise durch mathematische Funktionen mit Hilfe statistischer Verfahren abgebildet werden können.

Die Grundlage für die Aufstellung solcher mathematischer Funktionen sind empirisch erhobene Daten über unterschiedlichste Projektsituationen und über verschiedene Anwendungskontexte hinweg. Eine breite Datenbasis ermöglicht dabei die Abschätzung einzelner Faktoren auf beobachtbares Verhalten im Entwicklungsprozess. Dabei werden Methoden wie etwa Korrelationen und Regressionen verwendet, um entsprechende Modelle zu bilden. Für die Anwendung statistischer Analysemethoden sei auf die einschlägige Literatur verwiesen, etwa (Kohn 2005) und (Pruscha 2006).

Methoden aus dem Bereich der Zeit- und Kostenplanung für Entwicklungsprojekte verwenden solch analytische Methoden zur Beschreibung von Modellen. Eines der bekanntesten Modelle ist etwa das Constructive Cost Model (COCOMO) von Boehm et al. (Boehm et al. 1995), (Boehm et al. 2000a). Auf Basis empirischer Daten wurde ein Modell entwickelt, welches den zu erwartenden Aufwand einer Entwicklung durch Angabe verschiedener Parameter abschätzen kann. Der Aufwand (*Effort*) wird dabei im Wesentlichen durch die Größe des Produkts (*Size*) etwa in Source Lines of Code bestimmt. Dabei beschreibt der Faktor *Productivity* die angenommene Produktivität in Aufwand pro Größe und der Exponent *E* etwa nachteilige Effekte für größere oder schwierigere Produkte. Zudem gehen Kostentreiber wie etwa Zuverlässigkeit, Komplexität und Wiederverwendbarkeit als Faktoren ein (*EM*).

$$Effort = Productivity \cdot Size^E \cdot \prod_{i=1}^n EM_i$$

Ein weiteres Modell wurde mit dem Software Life-cycle Model (SLIM) von Putnam entwickelt (Putnam 1978), (Putnam und Myers 1991). Ähnlich dem Modell von Boehm ist der wesentliche Einflussfaktor die angenommene Größe des Produkts (*Size*). Bestimmt wird der Aufwand weiterhin durch die Produktivität (*Productivity*) und gegebenenfalls durch Kostentreiber (*B*). Im Gegensatz zum Modell von Boehm geht allerdings auch die Entwicklungszeit (*Time*) mit ein. Dabei sinkt der Aufwand mit einer längeren Entwicklungsdauer.

$$Effort = \left[\frac{Size}{Productivity \cdot Time^{\frac{4}{3}}} \right]^3 \cdot B$$

Neben diesen Modellen von Boehm und Putnam existieren zahlreiche Abwandlungen, Erweiterungen und Alternativen, vergleiche (Pillai und Sukumaran Nair 1997), (Boehm et al. 2000b). Allen Modellen gemein ist, dass die verschiedenen Einflussfaktoren durch Experten geschätzt beziehungsweise auf Basis gesammelter Erfahrung auf den betrachteten Projektkontext kalibriert werden müssen (Fenton und Neil 2000).

Der Vorteil dieser Methoden liegt in ihrer relativ schnellen Anwendung. Nach der Kalibrierung der notwendigen Parameter liefern diese Methoden ein Ergebnis. Allerdings stellt sich in der Anwendung heraus, dass die Kalibrierung sowie auch die Interpretation der Ergebnisse solcher Modelle mit Problemen verbunden sind, vergleiche (Dejaeger et al. 2011), (Malhotra und Jain 2011), (Boehm et al. 1995). So basiert eine Kalibrierung beispielsweise fundamental auf der Erfahrung des Anwenders in der Einschätzung von Parametern. Ebenso dient im Wesentlichen die Größe des zu entwickelnde Produkts als Eingangsparameter. Über die Qualität, welche den Prozessfortschritt wesentlich bestimmt, werden im Allgemeinen keine Aussagen – allenfalls statische Annahmen – getroffen. Auf der anderen Seite sind die Ergebnisse teilweise nicht eindeutig und vergleichbar interpretierbar, da eben kein explizites Wissen über Wirkzusammenhänge existiert. Vergleiche etwa (Mohagheghi et al. 2005) und (Ordonez und Haddad 2008).

Wirkzusammenhänge

Viele Probleme der Wirklichkeit sind wegen ihrer Komplexität nicht durch exakte mathematische Modelle beschreibbar. Daher werden analytische Modelle für umfangreiche Systeme auch als Block Box bezeichnet (Raffo 1993). So kann bei komplexen Systemen, wie etwa bei Entwicklungsprozessen, nicht davon ausgegangen werden, dass dieselben erfassten Kontextbedingungen deterministisch zu demselben Prozessverhalten führen. Beispielsweise können nicht alle Kontextbedingungen, wie etwa die Erfahrung

aller beteiligten Entwickler exakt quantifiziert werden. Zudem beeinflussen unbekannte Störgrößen das Verhalten eines Entwicklungsprozesses zum Teil erheblich. So muss der Einfluss des Zufalls durch Wahrscheinlichkeitsverteilungen für bestimmte Kontextbedingungen approximiert werden (Law und Kelton 1991), (Banks 1998), (Bossel 2004) und (Müller und Pfahl 2008).

Auf der zweiten Ebene der Pyramide (Abbildung 2.7) werden Zusammenhänge und Einflüsse zwischen Prozesselementen explizit modelliert. Durch diesen Ansatz werden kritische Wirkzusammenhänge detaillierter, als eine Kombination mathematischer Funktionen, betrachtet. Dabei können Wirkzusammenhänge probabilistisch durch Wahrscheinlichkeitsverteilungen beschrieben werden, um dem Einfluss von Störgrößen und Varianz beziehungsweise Risiko, im Gegensatz zu abstrakten analytischen Modellen, Rechnung zu tragen.

Auf der untersten Ebene der Pyramide (Abbildung 2.7) werden ausgewählte Teile von Prozessen oder Algorithmen von bestimmten Methoden durch feingranulare Modelle beschrieben (Buschermöhle und Oelerink 2008). Solche Modelle können beispielsweise Algorithmen von Werkzeugen nachbilden und somit noch genauere Aussagen über Wirkzusammenhänge beschreiben. Dabei kann das Verhalten von Algorithmen im Detail etwa durch Aktivitätsdiagramme oder Zustandsdiagramme beschrieben werden und die Auswirkungen auf bestimmte Einflussfaktoren beobachtet und analysiert werden. Auch ist in diesem Zusammenhang eine Optimierung von Modellen möglich, um Aussagen über die Verbesserung von Prozessen, Werkzeugen und Zusammenhängen treffen zu können.

Wie bereits oben angesprochen, nimmt mit jeder Ebene der Pyramide das modellierte Wissen über einen Entwicklungsprozess zu. Damit einher geht allerdings der notwendige Aufwand für die Modellierung entsprechender Modelle. Für die in dieser Arbeit entwickelte Methodik wird in Kauf genommen, Aufwand zu investieren, um Modelle auf der zweiten Ebene zu modellieren. Auch wenn sich diese Arbeit in die zweite Ebene der Abstraktion einordnet, ist dennoch eine Einbeziehung von Modellen anderer Ebenen von Interesse. So sollen durchaus detaillierte Modelle mit in die Betrachtung einbezogen werden können, wenn dieses erforderlich scheint. Ist der kritische Einfluss bestimmter Zusammenhänge adäquat mit abstrakteren Modellen abbildbar, so sollen auch solche Modelle anwendbar sein.

In den folgenden Abschnitten werden stochastische und simulative Ansätze zur Analyse von Prozessen vorgestellt.

2.7 Stochastische Ansätze zur Analyse von Prozessen

Zahlreiche Arbeiten versuchen den Einfluss von Prozessänderungen durch stochastische Prozesse zu bewerten beziehungsweise nutzen die Idee stochastischer Prozesse. Eine besondere Beachtung kommt dabei den bereits angesprochenen Markov-Ketten zu, die oft zur stochastischen Beschreibung von Prozessen verwendet werden. Im Folgenden wird daher zunächst kurz der Ansatz der Markov-Ketten eingeführt. Danach werden Ansätze diskutiert, welche die Idee stochastischer Prozesse nutzen, um Einflussanalysen für Prozesse durchzuführen.

2.7.1 Markov-Ketten

Eine Markov-Kette verfügt über eine endliche Menge von Zuständen $S = \{s_1, s_2, \dots, s_r\}$ und eine endliche Menge von Transitionen $T = \{t_1, t_2, \dots, t_r\}$ zwischen diesen Zuständen. Eine Anfangsverteilung beschreibt den Zustand in dem der Prozess startet. Der Zustand in dem der Prozess endet wird als absorbierender Zustand bezeichnet.

Für die Untersuchung von Entwicklungsprozessen werden nun Aufgaben beziehungsweise als atomar betrachtet Tätigkeiten auf Zustände abgebildet und der Ablauf zwischen diesen durch Transitionen modelliert. Jede Transition verfügt dabei über einen bestimmten Wert p_{ij} , welcher die Übergangswahrscheinlichkeit von einem Zustand i zu einem anderen Zustand j beschreibt. Für Markov-Ketten erster Ordnung hängen diese Übergangswahrscheinlichkeiten nicht von der Vergangenheit ab und bleiben konstant über die Zeit. Für diese Übergangswahrscheinlichkeiten gelten:

$$p_{ij} \geq 0$$

$$\sum_{j \in S} p_{ij} = 1 \text{ mit } i \in S$$

Neben diesen sogenannten stationären Übergangswahrscheinlichkeiten lassen sich auch ändernde Wahrscheinlichkeiten modellieren. Eine Anwendung nicht-stationärer Übergangswahrscheinlichkeiten für Entwicklungsprozesse in der Literatur ist allerdings nicht bekannt. Für die weitere Theorie hinter stochastischen Prozessen und insbesondere Markov-Ketten sei an dieser Stelle auf die Literatur verwiesen (Grinstead und Snell 1997), (Meintrup und Schäffler 2005).

Abbildung 2.8 zeigt das Modell des Entwicklungsprozesses aus Abschnitt 1.5. Vom Startzustand, der keine eingehende Transition besitzt, geht es über die Aufgaben *Behavioral Design* und *RTL Design* zu Analysis, welche die Ergebnisse der vorherigen Schritte überprüft. Werden Fehler in den entsprechenden Design Artefakten gefunden, so ist ein Rücksprung zu vorherigen Aufgaben notwendig. Diese Rücksprünge besitzen

als Transition eine bestimmte Wahrscheinlichkeit. Für einen Rücksprung von *Analysis* zu *Behavioral Design* ist dieses beispielsweise 20%.

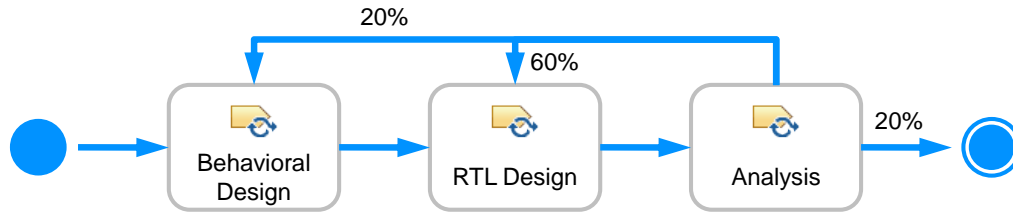


Abbildung 2.8 Einfache Markov-Kette mit absorbierenden Zustand

Der gezeigte Entwicklungsprozess lässt sich nun als Markov-Kette analytisch oder durch Simulation untersuchen. Im Folgenden wird eine analytische Lösung für die Ermittlung der zu erwartenden Iterationshäufigkeit jeder Aufgabe vorgestellt. Dazu werden die gezeigten Aufgaben beziehungsweise Zustände und die Übergangswahrscheinlichkeiten der Transitionen zunächst in einer Matrix P abgebildet. Jeder Eintrag p_{ij} beschreibt die Wahrscheinlichkeit, von einer Aufgabe i (Zeile) zu einer Aufgabe j (Spalte) zu gehen.

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0,2 & 0,6 & 0 & 0,2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Aus der kanonischen Form der Matrix P lassen sich die für die analytische Untersuchung relevanten Übergangswahrscheinlichkeiten auswählen. Dazu ist P so zu ordnen, dass sich alle nicht absorbierenden Zustände oben in der Matrix befinden – das ist im Beispiel bereits der Fall. Im Folgenden interessiert dann die Matrix T , welche keine absorbierenden Zustände mehr enthält.

$$P = \begin{pmatrix} T & S \\ 0 & I \end{pmatrix}$$

Durch die Fundamentalmatrix $N = (E - T)^{-1}$ wird die zu erwartende Häufigkeit von Iterationen einer jeden betrachteten Aufgabe bestimmt, wobei E den T entsprechenden Einheitsvektor beschreibt. Die Fundamentalmatrix lässt sich mit Hilfe des Gauss-Jordan-Algorithmus in der Form $(E - T|E)$ berechnen, indem diese auf die Form $(E|N)$ gebracht wird.

$$T = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0,2 & 0,6 & 0 \end{pmatrix} \quad E - T = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & -0,2 & -0,6 & 1 \end{pmatrix}$$

Daraus ergibt sich die Fundamentalmatrix N für das Beispiel wie folgt.

$$N = \begin{pmatrix} 1 & 2 & 5 & 5 \\ 0 & 2 & 5 & 5 \\ 0 & 1 & 5 & 5 \\ 0 & 1 & 4 & 5 \end{pmatrix}$$

Aus der ersten Zeile von N lässt sich nun die mittlere Häufigkeit an Iterationen jeder Aufgabe ablesen, wenn der Prozess im Startzustand beginnt. Demnach würde Behavioral Design zwei Mal, RTL Design fünf Mal und auch Analysis fünf Mal durchgeführt werden. Um nun mit Hilfe von Markov-Ketten Einflussanalysen zu erlauben, können die Übergangswahrscheinlichkeiten zwischen Aufgaben variiert werden und der Einfluss einer solchen Änderung auf die Iterationshäufigkeit einer Aufgaben untersucht werden.

2.7.2 Ansätze auf Basis stochastischer Prozesse

Smith und Eppinger entwickeln ein Vorhersagemodell für Iterationen in sequentiellen Entwicklungsprojekten auf Basis der sogenannten Design Structure Matrix und Markov-Ketten (Smith und Eppinger 1997). Die Design Structure Matrix (Steward 1981) ermöglicht dabei die binäre Beschreibung von Beziehungen zwischen unterschiedlichen Elementen in Form einer Matrix. Somit wird die Design Structure Matrix zur Abbildung des Informationsflusses zwischen Aufgaben beziehungsweise deren Ablauf verwendet (Eppinger et al. 1994). Die in der Matrix abgebildeten Aufgaben und Beziehungen werden dann auf eine Markov-Kette abgebildet. Damit entspricht die Verwendung der Design Structure Matrix im Wesentlichen der vorgestellten Matrix der Transitionen. Auf der Diagonale wird dabei die angenommene mittlere Dauer einer jeden Aufgabe pro Iteration notiert.

Schließlich wird die benötigte Entwicklungsdauer analytisch ermittelt, wobei angenommen wird, dass sich die Dauer einer Aufgabe nicht durch Einflussfaktoren verändert und über die gesamte Entwicklungsdauer konstant bleibt. Als Aufweichung dieser Annahme erlauben die Autoren die einmalige Änderung der Dauer einer Aufgabe ab der zweiten Iteration. Somit soll die analytische Lösung einer Markov-Kette weiterhin möglich bleiben, da ansonsten Simulation verwendet werden müsste (Smith und Eppinger 1997). In weiteren Arbeiten wird Simulation eingesetzt, um den Einfluss variierender Zeiten von Aufgaben, etwa durch Lerneffekte, zu untersuchen (Cho und Eppinger 2001).

Johnson beschreibt in seiner Arbeit die Notwendigkeit die Performanz von Entwicklungsprozessen zu beobachten und zu verbessern, um als Unternehmen erfolgreich im Markt zu bleiben (Johnson 1996). Zur Abbildung von Design Prozessen (etwa in der Hardware-Entwicklung) wird ein sequentielles und iteratives Prozessmodell entwickelt, welches sich der Design Structure Matrix und einer grafischen Notation von Markov-

Ketten bedient. So baut die Arbeit auf den Arbeiten von Smith und Eppinger auf. Mit Hilfe simulativer Methoden (Johnson et al. 1996) und analytischer Methoden (Johnson und Brockman 1998) werden die Auswirkungen der Wahrscheinlichkeit einer Transition von einer Aufgabe zu einer anderen Aufgabe auf die Dauer eines Prozesses untersucht. Außerdem führen die Autoren auf Basis von Markov-Ketten analytisch Sensitivitätsanalysen für die Übergangswahrscheinlichkeiten durch (Johnson und Brockman 1996).

Browning arbeitet an der Modellierung und Analyse von Zeit, Kosten und Leistung von Entwicklungsprozessen und verwendet die Idee von Übergangswahrscheinlichkeiten (Browning 1998). Wie bei den bereits vorgestellten Arbeiten wird die Design Structure Matrix verwendet, um Abhängigkeiten zwischen Aufgaben zu modellieren. Als Erweiterung bezieht Browning trianguläre Wahrscheinlichkeitsverteilungen (Minimum, wahrscheinlichster Wert und Maximum) für die initiale Belegung von Dauer und Kosten von Aufgaben mit ein. Auch Lerneffekte und Überarbeitungsaufwand mit einer Auswirkung auf Dauer und Kosten werden betrachtet, indem zwischen initialem Aufwand und einem weiteren Überarbeitungsaufwand unterscheiden wird. In diesem Zusammenhang wird dann das Risiko von Produktentwicklungen in die Untersuchungen mit einbezogen (Browning et al. 2002), (Browning und Eppinger 2002). Durch Simulation wird ein Bild über mögliche Ergebnisse von Prozessaufwand und Prozesskosten gegeben.

Auch Deissenboeck und Pizka untersuchen den ökonomischen Einfluss von Software Prozessvariationen, also den Einfluss einer Prozessänderung, mit Hilfe von Markov-Ketten (Deissenboeck und Pizka 2007). Dabei verwenden die Autoren die klassische Modellierung von Markov-Ketten. Die Autoren betrachten dabei keine variierenden Aufwände wie es durch Wahrscheinlichkeitsverteilungen möglich wäre. Allerdings wird auf Basis der Erfahrung eine konstante Reduktion des Aufwandes einer Aufgabe bei jeder Iteration modelliert. Durch analytisches Vorgehen beziehungsweise Simulation wird der Einfluss einer Prozessänderung auf den notwendigen Aufwand und den damit verbundenen Kosten untersucht (Deissenboeck und Pizka 2008). Dabei wird der Aufwand eines Entwicklungsprozesses durch die Summe der durchschnittlichen Aufwände aller Aufgaben (a) und deren erwarteten Iterationshäufigkeit beschrieben.

$$\text{Aufwand} = \sum_{a \in A} \text{Aufwand}(a) \cdot \text{Iterationen}(a)$$

Zur Untersuchung des ökonomischen Einflusses einer Prozessänderung wird die mittlere Differenz der erwarteten Aufwände der unterschiedlichen Entwicklungsprozesse betrachtet. Der Aufwand des neuen Entwicklungsprozesses (*Soll*) wird dabei vom Aufwand des aktuellen Entwicklungsprozesses (*Ist*) subtrahiert. Eine weitere Betrachtung

ung von Wahrscheinlichkeit und Risiko zum Mehrwert einer Prozessänderung erfolgt nur über eine Variation der Übergangswahrscheinlichkeiten.

$$\Delta \text{Aufwand} = \text{Aufwand}_{\text{Ist}} - \text{Aufwand}_{\text{Soll}}$$

2.7.3 Zusammenfassung

Stochastische Prozesse wie Markov-Ketten ermöglichen die Modellierung und Untersuchung von Entwicklungsprozessen auf einer abstrakten Ebene. Dabei wird den Übergängen von einer Aufgabe zu einer anderen Aufgabe eine bestimmte Wahrscheinlichkeit zugeordnet. Durch analytische Methoden oder Simulation lassen sich dann die Iterationshäufigkeiten einzelner Aufgaben und damit der Entwicklungsaufwand ermitteln.

Allerdings werden einige abstrahierende Annahmen bei den vorgestellten Arbeiten für die Untersuchung von Entwicklungsprozessen getroffen, um klassische Markov-Ketten verwenden zu können. So wird angenommen, dass

- sich die Übergangswahrscheinlichkeiten zwischen Aufgaben nicht während der Entwicklung verändern,
- die Dauer von Aufgaben über Iterationen konstant bleibt oder sich lediglich von der ersten zur zweiten Iteration verändern,
- das Prozessverhalten beeinflussende Eigenschaften von Prozesselementen allgemein abgeschätzt werden können und diese sich nicht über die Zeit verändern.

Zudem liegen die Schwerpunkte vorgestellter Arbeiten auf einzelnen Entwicklungsprojekten und eine Optimierung des Entwicklungsaufwandes durch eine Reorganisation von durchzuführenden Aufgaben. Durch keinen der genannten Ansätze werden Methoden und weitergehendes Wissen von Entwicklungsprozessen beschrieben, wie es für diese Arbeit gefordert wird. Auch von existierenden Wirkzusammenhängen wird bei klassischen Markov-Ketten abstrahiert, so dass der Idee dieser Arbeit nicht vollständig nachgekommen werden kann. Um dennoch weitere Details zur Unsicherheit von Entwicklungsprozessen mit einzubeziehen, etwa eine Wahrscheinlichkeitsverteilung für den Aufwand einer Aufgabe, verwenden vorgestellte Arbeiten Übergangswahrscheinlichkeiten und Simulation.

Als Fazit kann festgestellt werden, dass Markov-Ketten beziehungsweise Übergangswahrscheinlichkeiten für erste Einflussanalysen von Methodenänderungen sinnvoll einsetzbar sind, da sie schnell Ergebnisse liefern und relativ wenig Datenmaterial benötigen. Somit wird eine Basis für die **Anforderung 2** (Unsicherheit) und **Anforderung 3** (Analyse) bereitgestellt und auch zu **Anforderung 1** (Kennzahlen) werden Beiträge mit Blick auf den Entwicklungsaufwand geliefert.

2.8 Simulative Ansätze zur Analyse von Prozessen

Ein anderer Ansatz zur Untersuchung von Entwicklungsprozessen wird mit dem Bereich der Prozesssimulation aufgespannt. Simulation zielt auf die Imitation beobachtbaren Verhaltens. Auf Grundlage von Simulationsmodellen können dann kontrollierte virtuelle Experimente durchgeführt werden, welche in der Wirklichkeit nicht oder nur in Verbindung mit großen Aufwänden möglich wären (Law und Kelton 1991). Dabei werden, im Gegensatz zum klassischen stochastischen Ansatz, Wirkzusammenhänge explizit beschrieben und das Verhalten von Prozessen damit detaillierter modelliert. Auch die Änderung von Eigenschaften von Prozesselementen und Wirkzusammenhängen über die Zeit kann mit Hilfe von Simulation untersucht werden.

Simulation wird für verschiedene Fragestellungen eingesetzt. Die wichtigsten Fragestellungen, die mit Hilfe von Simulation untersucht werden sind nach (Kellner et al. 1999), (Raffo et al. 1999a) und (Müller und Pfahl 2008):

- Strategisches Management und Unterstützung der Entscheidungsfindung,
- Planung, Kontrolle und operatives Management von Projekten,
- Prozessoptimierung und Einführung neuer Technologien sowie
- Verstehen, Training und Lernen.

Für die Simulation von Systemen und im Speziellen von Entwicklungsprozessen lassen sich im Wesentlichen zwei Ansätze unterscheiden. Unter dem Begriff System Dynamics fallen Simulationsmodelle, welche dynamische Wirkzusammenhänge zwischen Elementen eines Systems beschreiben (Bossel 2004). Untersuchungen mit System Dynamics zielen auf grundsätzliche Erkenntnisse zu Wirkzusammenhängen. Modelle der Discrete Event Simulation gehen weiter ins Detail und erlauben auch die Abbildung von Abläufen sowie Verhalten für einzelne Prozesselemente (Law und Kelton 1991). Tabelle 2.1 stellt Merkmale der Methoden System Dynamics und Discrete Event Simulation gegenüber.

Tabelle 2.1 Gegenüberstellung von System Dynamics und Discrete Event Simulation

Merkmal	System Dynamics	Discrete Event Simulation
Untersuchung	System als Ganzes	Einzelne Aufgaben bis hin zum ganzen System
Wirkzusammenhänge	Grundlegende Wirkungen	Abbildung auf einzelne Elemente möglich
Verhalten	Kontinuierliche Änderung von Eigenschaften über die Zeit	Änderung von Eigenschaften zu bestimmten Zeitpunkten
Prozessmodell	Nein	Möglich
Modellierung	Expertenwissen	Expertenwissen und Daten
Aussagen	Eher qualitativ	Qualitativ und quantitativ

Ansätze, die versuchen System Dynamics und Discrete Event Simulation zu kombinieren, werden als Hybrid Simulation bezeichnet. Ein Vergleich existierender Ansätze ist etwa in (Zhang et al. 2008c) zu finden.

Im Folgenden werden diese drei Arten der Simulation zunächst kurz charakterisiert. Nach dieser Charakterisierung werden dann Ansätze zur Analyse von Entwicklungsprozessen auf Basis von Simulation vorgestellt. Die wesentlichen Arbeiten stammen dabei aus dem Bereich der Software-Entwicklung, in dem sich seit den 1990er Jahren mit den Themen Software Process Improvement und Software Process Simulation beschäftigt wird.

System Dynamics

Die ersten Arbeiten zu dem Ansatz System Dynamics wurden von Forrester in den 1950er Jahren durchgeführt, um den Einfluss einzelner Elemente auf das Verhalten industrieller Unternehmen zu untersuchen (Forrester 1958). Dabei zielt System Dynamics auf die Entwicklung einer Grundlage zur Unterstützung der Entscheidungsfindung im Management. Mit System Dynamics werden also unterschiedliche Elemente beziehungsweise Eigenschaften beschrieben, die sich durch Wirkzusammenhänge kontinuierlich über die Zeit verändern (Bossel 2004), (Madachy 2008). Dabei beeinflussen sich entsprechende Eigenschaften gegenseitig. Im Ergebnis ist der Zustand des betrachteten Systems zu einem Zeitpunkt t vom vorherigen Zustand des Systems zum Zeitpunkt $t - \Delta t$ abhängig. Für die Eigenschaften eines Systems bedeutet dies insbesondere, dass diese auch durch sich selbst beeinflusst werden können. Bei der Modellierung werden also Differentialgleichungen betrachtet. Für den neuen Wert p' einer Eigenschaft p , der Zeitdifferenz Δt und der aktuellen Veränderung $d(p)/dt$ bedeutet dies

$$p'(t) = p(t - \Delta t) + \frac{d(p)}{dt} \cdot \Delta t$$

Um Zusammenhänge und Wirkungen zwischen unterschiedlichen Eigenschaften beziehungsweise Faktoren abzubilden, werden sogenannte Einfluss- oder Kausal-diagramme verwendet. Abbildung 2.9 zeigt ein Beispiel für ein solches Diagramm.

Das Diagramm illustriert den Einfluss neuer Anforderungen während der Entwicklung auf andere Faktoren eines Entwicklungsprozesses nach (Park et al. 2008). Die verschiedenen Faktoren werden durch Pfeile miteinander verbunden. Durch ein Plus (+) beziehungsweise Minus (−) wird die Art des qualitativen Einflusses beschrieben. So bedingen mehr neue Anforderungen die Anzahl der notwendigen Überarbeitungen. Diese reduzieren wiederum die Motivation. Eine gute Motivation wirkt im Weiteren positiv auf die Fluktuation der Mitarbeiter. Diese Kontinuität der Mitarbeiter fördert dann die Erfahrung in der Entwicklung und reduziert damit den notwendigen Aufwand.

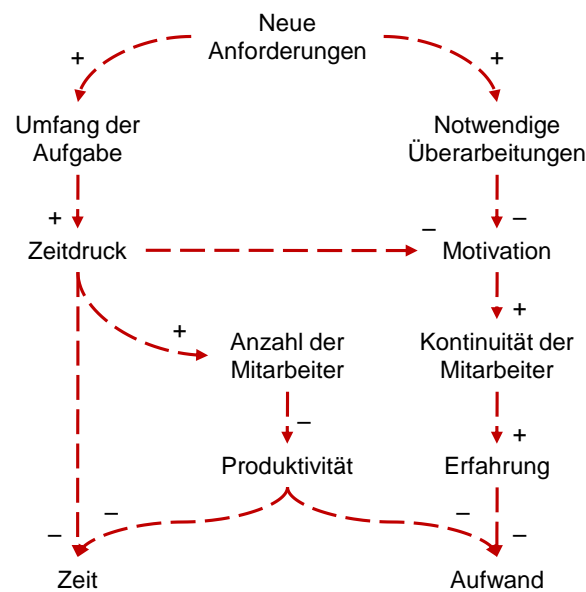


Abbildung 2.9 Wirkzusammenhänge bei System Dynamics nach (Park et al. 2008)

Discrete Event Simulation

Mit Discrete Event Simulation werden auftretende Ereignisse und deren Auswirkungen in einem System explizit modelliert. Ein Ereignis beeinflusst dabei den Zustand eines Systems, indem beim Auftreten eines Ereignisses Eigenschaften von Elementen des Systems verändert werden. Im Gegensatz zu System Dynamics wird das Verhalten also nicht in gleichmäßig definierten Zeitabständen betrachtet sondern eben zu bestimmten Ereignissen. Zudem können Ereignisse auch weitere Ereignisse auslösen. Die Dynamik eines Systems wird bei Discrete Event Simulation also zu auftretenden Ereignissen bestimmt (Law und Kelton 1991).

Für Entwicklungsprozesse können auftretende Ereignisse somit beispielsweise als Ablauf der einzelnen Aufgaben abgebildet werden. Mit der Auswahl einer Aufgabe wird der neue Zustand des Systems durch Berechnung der neuen Eigenschaften bestimmt und schließlich folgende Aufgaben als neues Ereignis gewählt.

Hybrid Simulation

Hybride Ansätze von Simulation versuchen die Vorteile der beiden vorgestellten Ansätze von System Dynamics und Discrete Event Simulation miteinander zu verbinden. Dabei werden Entwicklungsprozesse etwa diskret durch eine Folge von Aufgaben modelliert. Das dynamische Verhalten eines Entwicklungsprozesses wird dann auf einzelne Aufgaben oder das Gesamtsystem abgebildet, die andere Eigenschaften eines Entwicklungsprozesses beeinflussen.

2.8.1 Ansätze auf Basis von System Dynamics

Einer der ersten und bekanntesten Arbeiten zur Untersuchung von Software-Entwicklungsprozessen mit Hilfe von System Dynamics ist das Modell von Abdel-Hamid und Madnick (Abdel-Hamid und Madnick 1991). Der Fokus der Autoren liegt dabei auf dem Thema Software Project Management und verschiedenen Einflussfaktoren in der Software-Entwicklung. Die Autoren beschreiben eine ganze Reihe von Faktoren aus den Bereichen Human Resources, Software Production, Planning und Control (Abdel-Hamid und Madnick 1989), die sich gegenseitig beeinflussen. Ein Beispiel für einen Wirkzusammenhang ist der Einfluss des Erfahrungsmix der Mitarbeiter auf die potentielle Produktivität der Entwicklung.

Der Ansatz geht allerdings nicht über die Identifikation der angesprochenen Bereiche, entsprechender Einflussfaktoren und deren Simulation hinaus. So werden etwa Methoden und Entwicklungsprozesse mit ihren Aufgaben nicht explizit beschrieben, sondern eine abstraktere Managementsicht auf das spezielle Problem der Software-Entwicklung eingenommen.

Madachy untersucht die Auswirkungen einer abnahmeorientierten Software-Entwicklung auf den Aufwand eines Entwicklungsprozesses (Madachy 1996). Im Gegensatz zu dem Ansatz von Abdel-Hamid und Madnick wird der Entwicklungsprozess hier in einzelne Aufgaben detailliert. Dabei wird zwischen dem Verhalten von Aufgaben, der Generierung und Beseitigung von Fehlern, sowie der Entwicklung des Aufwandes differenziert. Madachy beschreibt somit eine ganze Reihe von Einflüssen auf das Fehlerverhalten und die Entwicklung des Aufwandes und bildet diese in einem festen System Dynamics Modell ab. Auch in diesem Ansatz wird mit den beschriebenen Wirkzusammenhängen eine erste Basis für das Verstehen von Entwicklungsprozessen gelegt, die allerdings nicht auf eine Modellierung und Beschreibung von Methoden- und Prozesswissen eingeht.

Für die Einteilung des Personals in Software-Entwicklungsprozessen entwickeln Collofello et al. ein dynamisches Modell (Collofello et al. 1998). Das Modell beschreibt die Einflüsse von Zeitdruck, Produktivität, Kommunikation, Erfahrung, erzeugte Fehler, Überarbeitungsdauer und Personalprofil auf den Aufwand und die Zeitplanung der Entwicklung. Damit versuchen die Autoren das Verhalten des Entwicklungssystems ähnlich wie auch Park et al., wie in Abbildung 2.9 gezeigt, zu beschreiben. Durch Simulation und Sensitivitätsanalysen wird dann der Einfluss der Erfahrung von Mitarbeitern auf den Aufwand der Entwicklung untersucht. Auch hier wird weder eine Beschreibung von Methoden noch eine Prozessmodellierung vorgenommen oder ein Ansatz dazu entwickelt.

Ruiz et al. verwenden den Ansatz der System Dynamics mit dem Ziel die Prozessoptimierung und Entscheidungsfindung zu unterstützen (Ruiz et al. 2002). Dabei legen die Autoren einen besonderen Blick auf den Reifegrad betrachteter Prozesse nach dem Capability Maturity Model (CMM). Die Autoren definieren ein Basismodul von abstrakten Wirkzusammenhängen, welches für verschiedene Phasen eines Entwicklungsprozesse mehrfach instanziiert wird. Betrachtet werden dabei die Anzahl ausstehender Aufgaben, durchgeführter Aufgaben, ausstehende Aufgaben für die Überarbeitung von Artefakten und die Qualität der Artefakte für die Phasen Anforderungen, Design, Code und Test. Trotz dieser Einteilung von Aufgaben wird keine Modellierung verwendet, die für den Aufbau von Methoden- und Prozesswissen verwendet werden könnte. Zudem weisen die modellierten Wirkzusammenhänge eine deutlich höhere Abstraktion als in den bereits angesprochenen Arbeiten auf.

Rus beschäftigt sich mit der Modellierung des Einflusses von Software Qualität auf die Dimensionen Zeit und Kosten von Software-Entwicklungsprozessen (Rus 1998). Rus und Collofello betrachten mit Hilfe von System Dynamics den Einfluss unterschiedlicher Fehlerpräventions- und Analysemethoden auf die Dauer und Kosten von Entwicklungsprojekten (Rus und Collofello 2001). Dabei unterteilen die Autoren ihr Modell auf verschiedene Aufgaben des Entwicklungsprozesses, im Speziellen der Test Phase. Abbildung 2.10 illustriert die betrachteten Aufgaben und deren Einflussfaktoren. Das vorgestellte Prozessmodell (ähnlich der SADT Modellierung) fungiert allerdings nur als grafische Notation zur Kommunikation und wird nicht zur Simulation oder weiteren Beschreibung von Wissen über einen Entwicklungsprozess verwendet.

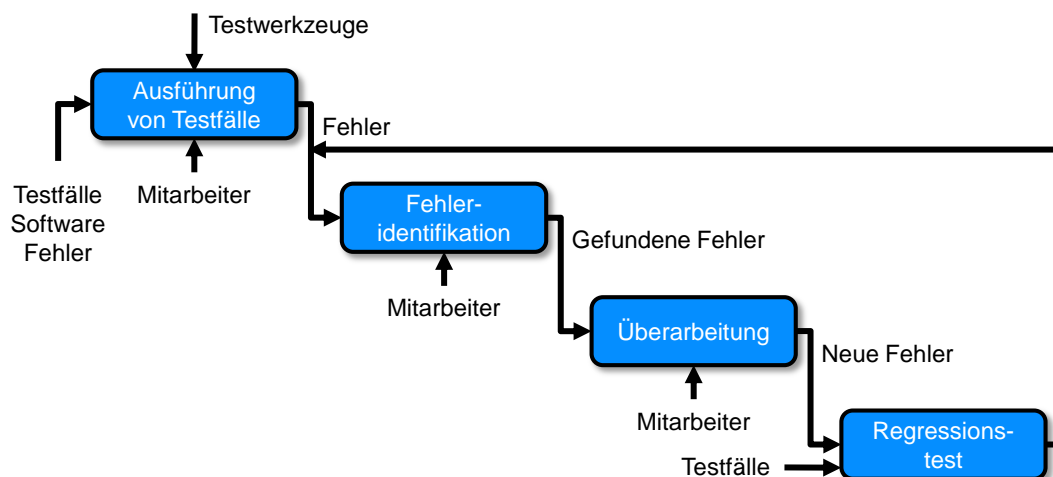


Abbildung 2.10 Aufgaben und Einflussfaktoren in der Test Phase nach (Rus und Collofello 2001)

Khosrovian et al. stellen einen Ansatz mit dem Namen GENSIM 2.0 vor, in dem wiederverwendbare Muster von System Dynamics Modellen für Software-Entwicklungsprozesse definiert werden (Khosrovian et al. 2008a), (Khosrovian et al. 2008b),

(Garousi et al. 2009). Dabei werden zwei Muster unterschieden. Zum einen wird ein Muster für die Kombination zweier Aufgaben für die Entwicklung und die Verifikation beschrieben. Das zweite Muster beschreibt den Zusammenhang zwischen der Entwicklung von Testfällen und deren Validierung. Diese generischen Muster können dann mehrfach miteinander kombiniert werden, um das Verhalten eines Entwicklungsprozesses abzubilden. Um die bereitgestellten Muster auf einen Anwendungskontext zu kalibrieren werden eine Reihe von Parametern definiert und ein Vorgehen dazu beschrieben (Khosrovian et al. 2008c). Allerdings mangelt es auch hier an einer Modellierung von Entwicklungsprozessen. Zudem sind Wirkzusammenhänge durch die beschriebenen Muster festgelegt und nicht dazu intendiert erweitert oder auf den Anwendungskontext angepasst zu werden.

Weitere Arbeiten beschäftigen sich beispielsweise mit der Untersuchung von sich fortlaufend verbessernden Software-Entwicklungsprozessen (Wernick und Hall 2004), der semi-quantitativen Beschreibung von Wirkzusammenhängen (Zhang und Kitchenham 2006), (Zhang et al. 2008b) und der Simulation von Geschäftsprozessen (An und Jeng 2005).

Zusammenfassung

Die im vorherigen Abschnitt beschriebenen Ansätze zeigen verschiedene Anwendungen der Methode der System Dynamics auf. Dabei verwenden alle Ansätze eine recht hohe Abstraktion tatsächlicher Entwicklungsprozesse. So werden zwar zum Teil auch Aufgaben eines Entwicklungsprozesses betrachtet, doch entspricht dieses eher einer Detaillierung von Wirkzusammenhängen als einer Prozessmodellierung, die für den weiteren Aufbau von Methoden- und Prozesswissen verwendbar wäre. Insgesamt werden damit Beiträge zu **Anforderung 3** (Analyse), **Anforderung 4** (Modellierung) in Bezug auf Wirkzusammenhänge, **Anforderung 6** (Wiederverwendung), **Anforderung 7** (Wirkzusammenhänge) und **Anforderung 8** (Methodik) geliefert, welche die definierten Anforderungen allerdings noch nicht erfüllen können.

Im Folgenden Abschnitt werden nun verschiedene Ansätze auf Basis von Discrete Event Simulation vorgestellt.

2.8.2 Ansätze auf Basis von Discrete Event Simulation

Einer der bekanntesten Vertreter zum Thema Software Process Simulation ist Raffo. Raffo untersucht den Einfluss von Prozessänderungen auf die Dimensionen Zeit, Kosten und Qualität in der Software-Entwicklung (Raffo 1993). Mit dem vorgestellten Ansatz wird (erstmalig) auf die Analyse des Einflusses einer Prozessänderung gezielt. Dazu werden drei Ebenen der Modellierung beschrieben. Die abstrakteste Ebene beschreibt den Prozess mit seinen Aufgaben und möglichen Abläufen zwischen diesen Aufgaben

als einfaches Diagramm. Für die weiteren Ebenen werden Aktivitätsdiagramme für die Modellierung von Aufgaben und Zustandsdiagramme zur Beschreibung des Prozessverhaltens verwendet. Über eine gewichtete Funktion über die Anzahl der Fehler im Quellcode und Aufwand wird dann die Performance zweier Prozessalternativen exemplarisch verglichen. In weiteren Arbeiten untersuchen Raffo et al. die Auswirkungen von Code Inspections sowie den Einfluss von Design und Code Reviews auf den Aufwand und die Kosten eines Software-Entwicklungsprozesses (Raffo et al. 1999b) sowie den Einfluss eines neuen Werkzeugs zur Qualitätssicherung in der Software-Entwicklung (Raffo et al. 2007) und (Raffo et al. 2008). Die Ergebnisse der Untersuchungen werden mit den Kennzahlen return-on-investment und Kapitalwert bewertet.

Der beschriebene Ansatz geht damit schon deutlicher in Richtung von Einflussanalysen für Entwicklungsprozesse, als dieses bei den anderen Ansätzen der Fall ist. Auch wird ein grundsätzliches Modell für die Modellierung von Entwicklungsprozessen geliefert. Allerdings ist die gewählte Modellierung gegenüber den in Abschnitt 2.4 beziehungsweise 2.5 vorgestellten Ansätzen der Methoden- und Prozessbeschreibung weit unterlegen.

Rus et al. beschreiben eine Methodik zur Entwicklung diskreter Simulationsmodelle für Software-Entwicklungsprozesse (Rus et al. 2002), (Rus et al. 2003). Das Vorgehen umfasst die Definition von Zielen, die Sammlung von Daten sowie die Entwicklung und Kalibrierung von diskreten Simulationsmodellen. Der Ausgangspunkt des Vorgehens bildet das Verstehen und die Dokumentation des betrachteten Entwicklungsprozesses. Einflussdiagramme werden dann für die Darstellung und Beschreibung von Wirkzusammenhängen verwendet, die durch empirische Daten kalibriert werden. Danach werden die gesammelten Erkenntnisse manuell in ein Simulationsmodell überführt.

Trotz des methodischen Vorgehens existiert ein deutlicher Bruch zwischen der Dokumentation eines Entwicklungsprozesses und der Entwicklung eines entsprechenden Simulationsmodells. Die Beschreibung von Methoden- und Prozesswissen liegt hinter den Methoden der Prozessmodellierung (Abschnitt 2.4) und des Method Engineerings (Abschnitt 2.5) zurück. Zudem wird nur die Modellierung eines Entwicklungsprozesses betrachtet. In der Konsequenz fehlt es an einem Ansatz zur Analyse beziehungsweise Bewertung von Einflüssen, etwa durch Methodenänderungen.

Müller entwickelt in seiner Arbeit einen Ansatz zur Simulation von Software-Entwicklungsprozessen zur Untersuchung der Auswirkungen von Qualitätssicherungsmaßnahmen auf Basis von Discrete Event Simulation (Müller 2007). Es werden unterschiedliche Modelle zu Wirkzusammenhängen definiert, welche die Entwicklung von Fehlern über die Zeit sowie den notwendigen Aufwand der Entwicklung und Beseitigung von Fehlern beschreiben. Dabei werden die Anzahl eingebauter Fehler, die Anzahl gefundener Fehler sowie die Anzahl korrigierter Fehler unterschieden. Die Anzahl

entsprechender Fehler wird durch grundsätzliche Annahmen zur Injection Rate, Detection Rate und Correction Rate bestimmt. Im Weiteren wird der notwendige Aufwand einer jeden Aufgabe durch Wahrscheinlichkeitsverteilungen bestimmt, in welche auch die Komplexität des betrachteten Produkts eingeht.

Der vorgestellte Ansatz fokussiert auf die Betrachtung der Fehlerentwicklung während eines Software-Entwicklungsprozesses. Dazu werden Muster für die Aufgaben der Entwicklung, Analyse und Fehlerbeseitigung unter Verwendung der genannten Modelle beschrieben, die eine Wiederverwendung der Modelle für verschiedene Phasen der Entwicklung erlauben soll. Eine grundlegende Basis für die Beschreibung des Wissens über Methoden und Prozesse wird damit allerdings nicht geliefert. Über diese Modelle hinaus wird ein Vorgehen für die Anpassung beziehungsweise Anwendung des Modells vorgeschlagen, dieses ist allerdings sehr abstrakt und gibt nur eine erste Orientierung für die Entwicklung von Simulationsmodellen auf Basis der genannten Muster.

Hassine et al. entwickeln ein Simulationsmodell für Hardware-Entwicklungsprozesse (Hassine und Barke 2008a), (Hassine und Barke 2008b). Dazu entwickeln die Autoren ein Request-Service Modell auf Basis von Petri-Netzen welches auf die Domäne der Chipentwicklung fokussiert. Ressourcen wie etwa Rollen oder Werkzeuge werden dabei als Anbieter von Diensten verstanden. So verfügt eine Ressource über eine Anzahl von Eigenschaften, die jeweils einen bestimmten Wert besitzen. Artefakte verfügen über die drei Eigenschaften Qualität, Komplexität und Format des Artefakts (z.B. Text). Artefakte als Eingabe werden durch Aufgaben in Ausgaben unter Verwendung von Ressourcen, denen bestimmte Kosten zugerechnet werden, transformiert. Im Gegensatz zu den anderen Ansätzen steht hier die Veränderung von Ressourcen und die Durchführbarkeit von Aufgaben für Projekte im Mittelpunkt (Hinrichs et al. 2009). Eine Modellierung von Methoden, Prozessen und Wirkzusammenhängen im angestrebten Sinne, erfolgt hier nicht. Auch wird keine Methodik vorgeschlagen, die Einflussanalysen erlauben würde.

Zusammenfassung

Die beschriebenen Ansätze der Discrete Event Simulation gehen klar in Richtung einer Prozessmodellierung im Gegensatz zu den Ansätzen der System Dynamics. Dennoch bietet kein Ansatz ein entsprechend weitgehendes Modell für den Aufbau von Methoden- und Prozesswissen sowie Wirkzusammenhängen an. Zum Teil werden durch die betrachteten Ansätze auch Vorgehen für die Untersuchung von Entwicklungsprozessen mit Hilfe von Simulation vorgestellt, die allerdings Brüche zwischen der Modellierung von Prozessen auf der einen Seite und der Simulation auf der anderen Seite aufweisen.

Die vorgestellten Arbeiten liefern damit im Wesentlichen, trotz der angesprochenen Schwächen, Beiträge zu **Anforderung 1** (Kennzahlen), **Anforderung 2** (Unsicherheit), **Anforderung 3** (Analyse), **Anforderung 4** (Modellierung), **Anforderung 6** (Wiederverwendung) und **Anforderung 8** (Methodik).

Der folgende Abschnitt gibt schließlich einen Überblick über Ansätze auf Basis von Hybrid Simulation.

2.8.3 Hybrid Simulation Ansätze

Martin und Raffo kombinieren ein System Dynamics Modell über das Verhalten der Umwelt eines Software-Entwicklungsprozesses mit einer diskreten Beschreibung des Ablaufes von Aufgaben, Aktivitäten oder Phasen in der Entwicklung (Martin und Raffo 2000), (Martin und Raffo 2001). Mit Hilfe der hybriden Modellierung werden dynamischen Aspekte wie Produktivität, Fehlerraten und Erfahrung von Mitarbeitern betrachtet, die sich über die Zeit verändern und einen Einfluss etwa auf den Aufwand der Entwicklung besitzen. Der Mehrwert der diskreten Modellierung zeigt sich hier zum einen bei der Abbildung einzelner Aufgaben und zum anderen in der Abbildung der Einflüsse auf die Anzahl neuer Fehler einzelner Entwicklungsartefakte. Zusammenfassend bestimmen die dynamischen Einflussfaktoren der Umwelt die eigentlichen Aufgaben der Entwicklung, der Schwerpunkt der Modellierung liegt also auf System Dynamics.

Zwar soll durch die Kombination der beiden Ansätze eine bessere Abbildung der Wirklichkeit erlaubt werden, dieses geht allerdings, wie bei allen hybriden Modellierungen, zu Lasten einer umfangreicheren und schwierigeren Modellierung. Es fehlt zudem wiederum an einer Möglichkeit zur grundlegenden Beschreibung von Methoden- und Prozesswissen sowie an einer Methodik, welche bei der Entwicklung hybrider Simulationsmodelle unterstützt.

Donzelli und Iazeolla modellieren den Ablauf von Aufgaben mit Hilfe von Discrete Event Simulation am Beispiel eines Software-Entwicklungsprozesses (Donzelli und Iazeolla 2001), (Donzelli 2006). Eine Aufgabe verfügt dabei über ein dynamisches Verhalten, welches sich jeweils für einen Durchlauf einer Aufgabe zeigt. So werden Eigenschaften von eingehenden Artefakten in Eigenschaften von ausgehenden Artefakten unter Verwendung von dynamischen Modellen transformiert. Ein solches dynamisches Modell basiert dabei etwa auf der aktuellen Entwicklungszeit und bestimmt die Anzahl, in das ausgehende Artefakt, eingebauter Fehler.

Auch bei diesem Ansatz fehlt es an einer durchgehenden Methodik für die Entwicklung entsprechender Modelle. Zudem wird keine weitergehende Beschreibung von Methoden- und Prozesswissen unterstützt.

Lakey beschreibt eine Kombination von System Dynamics mit Discrete Event Simulation, bei der das Verhalten von Aufgaben über Funktionen von Eingaben für eine Aufgabe und Umgebungsvariablen wie Zeitdruck, Erfahrung und Produktivität modelliert wird (Lakey 2003). Dynamische Aspekte werden dabei innerhalb von Aufgaben durch mehrfache Iterationen einer Aufgabe betrachtet. Im Anschluss an diese Iterationen einer Aufgabe werden die betrachteten Eigenschaften, wie etwa der notwendige Aufwand, aggregiert. Dieser Ansatz legt den Schwerpunkt demnach auf Discrete Event Simulation. Allerdings werden Aufgaben nur abstrakt betrachtet und weitere Prozesselemente werden lediglich indirekt durch Funktionen zur Beschreibung von Wirkzusammenhängen modelliert. Entsprechend fehlt es an einer Beschreibung von Methoden- und Prozesswissen, sowie an einer durchgehenden Methodik.

Über die bereits vorgestellten Ansätze hinaus entwickeln Choi et al. einen hybriden Ansatz, bei dem jede Aufgabe über ein System Dynamics Modell verfügt und der Ablauf von Aufgaben durch eine diskrete Modellierung bestimmt wird (Choi et al. 2006). Dabei werden die dynamischen Eigenschaftswerte bei einem Übergang von einer Aufgabe zu einer anderen Aufgabe an diese weitergegeben, um dann dort variiert zu werden. Durch diese strikte Aufteilung soll eine bessere Aggregation von betrachteten Eigenschaften wie etwa die Anzahl der Fehler eines Artefakts oder auch der notwendige Aufwand je Aufgabe ermöglicht werden, als dieses bei den anderen genannten Ansätzen möglich ist. Zudem erlaubt der Ansatz die Modularisierung von dynamischen Modellen einer Aufgabe. Trotz der Trennung zwischen Aufgaben und deren Verhalten, sowie der grundsätzlichen Wiederverwendbarkeit von dynamischen Modellen fehlt es an einer Methodik, wie entsprechende Modelle entwickelt werden können. Im Weiteren betrachtet der Ansatz nicht das Thema der Methoden- und Prozessbeschreibung, wie es für diese Arbeit angestrebt wird.

Park et al. stellen einen Ansatz vor, bei dem ausgehend von einer Prozessbeschreibung, auf Grundlage des Software & Systems Process Engineering Meta-Models in der Version 1.0, ein hybrides Simulationsmodell abgeleitet wird (Park et al. 2007), (Park et al. 2008). Der Ansatz zielt damit auf eine Reduktion des Aufwandes zur Entwicklung von Simulationsmodellen, eine Reduzierung der Komplexität von Simulationsmodellen sowie auf eine Verbesserung der Wiederverwendung von Modellen. Ein Entwicklungsprozess wird dazu in Form von Anwendungsfalldiagrammen und Aktivitätsdiagrammen beschrieben. Nach der Übersetzung in ein hybrides Modell, wie es von Choi et al. vorgestellt wird, werden Funktionen zur Abbildung von Wirkzusammenhängen in das Simulationsmodell integriert und das damit beschriebene Modell simuliert.

Zwar wird bei diesem Ansatz bereits vorhandenes Methoden- und Prozesswissen verwendet, doch besteht auch hierbei ein deutlicher Bruch zwischen der Prozessbeschreibung und der Modellierung von Wirkzusammenhängen. Zudem ist das in dem Ansatz

beschriebene Vorgehen sehr allgemein und beschränkt sich im Wesentlichen auf die Transformation notwendiger Modelle, im Gegensatz zu einer durchgehenden Methodik.

Schließlich stellen auch Zhang et al. ein hybrides Simulationsmodell aus der Bereich der Software-Entwicklung zur Untersuchung „Test-und-Fix“ Entwicklungen vor (Zhang et al. 2008a). Bei diesem Ansatz werden die Aufgaben der Implementierung und des Tests beziehungsweise der Korrektur von Fehlern unterschieden. Für die erste Aufgabe wird ein dynamisches Modell verwendet, welches die Generierung von Fehlern während der Entwicklung beschreibt. Für die zweite Aufgabe wird hingegen ein diskretes Modell verwendet, welches das Auffinden und die Beseitigung von Fehlern beschreibt. Insgesamt werden damit inkrementelle Entwicklungen betrachtet, bei denen Implementierung, Test und Korrektur parallel beziehungsweise gemeinsam durchgeführt werden. Auch bei diesem Ansatz wird keine Methodik vorgestellt, welche die Beschreibung von Methoden- und Prozesswissen sowie Einflussanalysen auf dieser Basis erlaubt.

Birkhölzer et al. beschreiben einen Ansatz für eine Bibliothek von wiederverwendbaren Komponenten für diskrete, dynamische und hybride Simulationsmodelle (Birkhölzer et al. 2010). Der Fokus der Bibliothek liegt dabei auf Modellen zur Beschreibung von Software-Entwicklungsprozessen. Diese wiederverwendbaren Komponenten werden in einer generischen Form beschrieben, so dass diese für unterschiedliche Implementierungen eingesetzt werden können und basieren auf Erkenntnissen aus anderen Ansätzen, wie etwa nach Boehm (COCOMO II), Khosrovian et al. (GENSIM 2.0) und (Madachy 2008). Definiert werden parametrisierbare Komponenten beispielsweise für die Entwicklung von Anforderungen, Design Aufgaben mit und ohne Beachtung von Erfahrung von Entwicklern und Lerneffekten. Die Beschreibung von Methoden- und Prozesswissen sowie die Definition einer Methodik für Einflussanalysen ist jedoch nicht Gegenstand des Ansatzes.

Zusammenfassung

In den Bereich der Hybrid Simulation fallen unterschiedliche Ansätze zur Kombination von System Dynamics und Discrete Event Simulation. Mit Blick auf eine Modellierung von Methoden- und Prozesswissen sind hier insbesondere Ansätze geeignet, welche den Ablauf von Prozessen durch eine diskrete Modellierung beschreiben und dynamisches Verhalten innerhalb einzelner Aufgaben betrachten. Auch wenn der Ansatz nach Park et al. bereits existierende Beschreibungen von Entwicklungsprozessen einbezieht, fehlt es doch insgesamt an einer durchgehenden Methodik und dem Aufbau von Methoden- und Prozesswissen mit ihren Wirkzusammenhängen.

Insgesamt werden mit den vorgestellten hybriden Ansätzen also Beiträge zu **Anforderung 1** (Kennzahlen), **Anforderung 2** (Unsicherheit), **Anforderung 3** (Analyse), **Anfor-**

forderung 4 (Modellierung), **Anforderung 6** (Wiederverwendung) und **Anforderung 7** (Wirkzusammenhänge) geliefert.

Zusammenfassend kann festgehalten werden, dass Prozesssimulation offensichtlich ein geeignetes Mittel zur Untersuchung von Entwicklungsprozessen darstellt. So erlaubt Prozesssimulation grundsätzlich die Abbildung der Unsicherheit von Entwicklungsprozessen auf verschiedenen Abstraktionsebenen und eine Variation des Prozessverhaltens auf Basis identifizierter Einflussgrößen. Für die Entwicklung von Simulationsmodellen ist es jedoch erforderlich, sich mit den Stärken und Gefahren von Simulation auseinander zu setzen, was im Folgenden getan wird.

2.8.4 Stärken und Gefahren von Simulation

Simulation bietet eine Reihe von Stärken zur Untersuchung von Systemen. Allerdings sind auch einige Gefahren bei der Anwendung von Simulation zu beachten. Im Folgenden werden diese Stärken und Gefahren von Simulation genannt, denen man sich bei der Verwendung dieser Methode bewusst sein sollte.

Vorteile von Simulation nach (Law und Kelton 1991) sind:

- Die meisten komplexen Systeme der Wirklichkeit mit stochastischen Elementen können nicht akkurat mit mathematischen Modellen beschrieben werden, um sie effizient analytisch zu lösen. Daher ist die Simulation oft die einzige Möglichkeit um Untersuchungen komplexer Systeme durchzuführen.
- Simulation erlaubt die Analyse und Bewertung existierender System unter bestimmten festgelegten Bedingungen.
- Alternative Systeme, so auch alternative Entwicklungsprozesse, können gegen festgelegte Anforderungen oder Ziele verglichen werden. Damit ist ebenso ein Vergleich von Alternativen untereinander möglich.
- Simulation ermöglicht Experimente mit einem System ohne die Wirklichkeit zu verändern und dennoch Erkenntnisse über mögliches Verhalten zu gewinnen. Änderungen in der Wirklichkeit sind zudem mit hohem Aufwand und Kosten verbunden und stellen in der Regel hohe Risiken dar.
- Simulation erlaubt die Untersuchung eines Systems in verschiedenen Zeithorizonten. So können kurze Zeitsequenzen der Wirklichkeit detailliert untersucht werden oder im Gegensatz zur Wirklichkeit, die benötigte Zeit zur Untersuchung (z.B. bei Beobachtungen) drastisch gestaucht und reduziert werden.

Für die erfolgreiche Verwendung von Simulation sollten einige Aspekte beachtet werden. Nach (Law und Kelton 1991) sind dieses:

- Jede Simulation eines stochastischen Modells liefert lediglich eine Ausprägung des Verhaltens eines Modells für die spezifizierten Eingangsbelegungen von Parametern des Modells. Entsprechend sind für jede interessante Belegung von

Parametern mehrere Simulationsläufe durchzuführen, um eine repräsentative Sicht über mögliches Verhalten zu erhalten.

- Die Entwicklung von Simulationsmodellen ist oft aufwendig und damit teuer. Daher ist ein grundlegendes Vorgehen zur Entwicklung von entsprechenden Modellen erforderlich um eine effiziente Modellierung zu unterstützen.
- Die Anzahl der simulierten Ergebnisse und die Verwendung von Animation werden oft höher bewertet, als ihr eigentlicher Informationsgehalt. Dementsprechend ist ein besonderes Augenmerk auf die Validität und Aussagekraft von Modellen zu legen.

Bei der Untersuchung von Systemen oder Entwicklungsprozessen mit Hilfe von Simulation können verschiedene Probleme auftreten, denen man sich bewusst und bei der Durchführung einer entsprechenden Studie beachtet werden sollten. Zu diesen Problemen zählen nach (Solomon 1983) und (Law und McComas 1989) und (Ulgen et al. 1996) unter anderem:

- Fehlende oder nicht genau spezifizierte Ziele der Untersuchung,
- Ungeeignete Abstraktionsebene des Modells,
- Fehlende Kommunikation mit dem Management während der Untersuchung,
- Fehlende Kompetenzen in Operations Research und Statistik,
- Verwendung komplexer (kommerzieller) nicht dokumentierter Modelle zur Beschreibung von Verhalten,
- Ungeeignete Verwendung von Animation,
- Fehlende Betrachtung von Quellen zufälligen Verhaltens,
- Verwendung beliebiger Wahrscheinlichkeitsverteilungen ohne Beschreibung einer entsprechenden Eignung,
- Analyse der Ergebnisse der Simulation durch statistische Methoden, welche die Unabhängigkeit von Parametern voraussetzen,
- Behandlung der Ergebnisse einer einfachen oder weniger Simulationen als die Wahrheit,
- Vergleich von alternativen Modellen auf Basis einfacher oder weniger Simulationen jedes Modells und
- Verwendung falscher oder fehlerhafter Zahlen zur Bewertung einzelner Eigenschaften.

Auch wenn die Liste der möglichen Probleme von Simulation umfangreich ist, so überwiegen doch die Vorteile für diese Arbeit. Mit Blick auf die Komplexität von Entwicklungsprozessen, Wirkzusammenhängen und möglichen unbekanntem Einflüssen ist hier die Verwendung von Wahrscheinlichkeitsverteilungen für die Abbildung beobachtbarer Phänomene interessant.

2.8.5 Zusammenfassung

Simulation ist eine Methode, welche die Abbildung von beobachtbarem Verhalten der Wirklichkeit in ein Modell erlaubt. Modelle eines Systems oder Entwicklungsprozesses können dann ohne Auswirkung auf die Wirklichkeit verändert und als virtuelles Experiment untersucht werden.

Mit dem Ansatz der System Dynamics werden Wirkzusammenhänge zwischen Elementen eines Systems und deren Änderung über die Zeit beschrieben. System Dynamics zielt bei der üblichen Abstraktion auf die Unterstützung der Entscheidungsfindung des Managements für entsprechende Änderungen eines Systems.

Der Ansatz der Discrete Event Simulation geht hingegen weiter ins Detail als der Ansatz der System Dynamics. Hierbei werden auftretende Ereignisse zu bestimmten Zeitpunkten betrachtet. So können im Gegensatz zum Ansatz der System Dynamics auch Aufgaben, deren Ablauf und deren Verhalten abgebildet werden.

Hybrid Simulation Ansätze kombinieren System Dynamics mit Discrete Event Simulation, um die Stärken beider Ansätze zu nutzen. Damit lassen sich die Aufgaben eines Entwicklungsprozesses und dynamische Aspekte darstellen. Die Entwicklung entsprechender Modelle ist allerdings aufwändiger als bei den anderen beiden Ansätzen.

Zur Entwicklung von Simulationsmodellen existiert eine ganze Reihe von Werkzeugen, die auch durch die genannten Ansätze verwendet werden. Allerdings ist keine Ansatz bekannt, welche auf den Aufbau von Methoden- und Prozesswissen mit ihren Wirkzusammenhängen zielt und eine solche Beschreibung mit der Simulation von Entwicklungsprozessen mit dem Ziel von Einflussanalysen verbindet beziehungsweise eine definierte Methodik dafür bereitstellt. So fehlt es beispielsweise an einer Modellierung von Artefakten, Wirkzusammenhänge werden lediglich als Funktion beschrieben oder es existiert ein deutlicher Bruch zwischen der Prozessmodellierung und der Simulation von Prozessen. Auch eine durchgehende Werkzeugunterstützung ist nicht bekannt (**Anforderung 10**). Für die praktische Anwendung ist außerdem eine Methodik erforderlich, welche die grundlegende Beschreibung von Entwicklungsprozessen auf relativ einfache Weise ermöglicht und diese für eine Simulation und die Auswertung von möglichen Prozessänderungen erlaubt. Dabei sind die genannten möglichen Probleme von Simulation zu beachten.

2.9 Handlungsbedarf zur Einflussanalyse und Einflussbewertung

Im Folgenden werden die Erkenntnisse aus dem Stand der Technik in Bezug auf die definierten Anforderungen (Abschnitt 2.1) an eine Methodik zur Einflussbewertung von Methodenänderungen in Entwicklungsprozessen zusammengefasst und der Handlungsbedarf dargestellt.

Anforderung 1. Kennzahlen. Kennzahlen zur Bewertung von Änderungen in Entwicklungsprozessen werden grundsätzlich bereits durch die Methoden der Leistungs- und Nutzenbewertung eingeführt. Für eine Bewertung von Entwicklungsprozessen sind die Dimensionen Zeit, Kosten und Qualität wichtige Größen, welche etwa um Rentabilitätsbetrachtungen erweitert werden. Die betrachteten Ansätze zur Analyse beziehungsweise Simulation von Entwicklungsprozessen fokussieren allerdings in der Regel auf eine dieser Größen. Für fundierte Einflussanalysen und Einflussbewertungen sind die genannten Kennzahlen zu unterstützen.

Anforderung 2. Unsicherheit. Die Unsicherheit beziehungsweise Varianz in Entwicklungsprozessen ist ein wichtiges Problem. Ansätze zur stochastischen Analyse von Prozessen sowie Ansätze der Simulation beachten diesen Einflussfaktor. Dabei sind stochastische Modellierungen mit Übergangswahrscheinlichkeiten insbesondere dann geeignet, wenn kein ausreichendes Wissen über Wirkzusammenhänge existiert. Ansonsten eignet sich Simulation besser zur Abbildung von Wirkzusammenhängen und der Unsicherheit in Entwicklungsprozessen, da hierfür detailliertere Modelle beschrieben und untersucht werden können.

Anforderung 3. Analyse. Alle betrachteten Ansätze liefern eine Basis zur Analyse beziehungsweise Bewertung von Entwicklungsprozessen. Die stochastischen Methoden fokussieren allerdings auf den Aufwand von Prozessen. Bei simulativen Ansätzen werden die Dimensionen Zeit, Kosten und Qualität je nach Anwendungskontext mehr oder weniger ausgeprägt betrachtet. Bereits zu Beginn einer Einflussanalyse müssen die Ziele formuliert und Dimensionen bestimmt werden, in denen eine Bewertung durchgeführt werden soll.

Anforderung 4. Modellierung. Einzig durch die Methoden zur Modellierung von Prozessen beziehungsweise des Method Engineering wird eine ausreichende Beschreibung von Methoden- und Prozesswissen ermöglicht. Diese Beschreibung bezieht allerdings noch keine Wirkzusammenhänge zwischen Prozesselementen mit ein. Hierzu eignen sich Einflussdiagramme und Modellierungen, wie sie in den Ansätzen der System Dynamics Simulation verwendet werden. Es ist demnach ein Metamodell und eine Methodik notwendig, welche eine entsprechende Modellierung unterstützt.

Anforderung 5. Durchgängigkeit. Allein im Bereich der hybriden Simulation von Entwicklungsprozessen beschreiben Park et al. einen Ansatz bei dem eine Prozessbe-

beschreibung durch Transformation für eine Simulation verwendet wird (Park et al. 2008). Dabei besteht jedoch weiterhin ein Bruch zwischen der Beschreibung von Wirkzusammenhängen und deren Modellierung für Simulationsmodelle. Hier ist eine durchgehende Modellierung auf Basis von modellierten Entwicklungsprozessen anzustreben, welche direkt für die Einflussanalyse und Einflussbewertung verwendet werden kann.

Anforderung 6. Wiederverwendung. Das Thema Wiederverwendung wird durch die Methoden der Prozessmodellierung beziehungsweise des Method Engineerings behandelt. Verschiedene Ansätze aus dem Bereich der Simulation erlauben die Wiederverwendung von generischen Mustern in Simulationsmodellen. Eine Kombination dieser Ansätze existiert in den betrachteten Arbeiten nicht.

Anforderung 7. Wirkzusammenhänge. Die vorgestellten stochastischen Ansätze sowie die Ansätze von Simulation erlauben die Modellierung von Wirkzusammenhängen. Dabei bilden die stochastischen Ansätze Wirkzusammenhänge in Form von beobachteten Übergangswahrscheinlichkeiten recht abstrakt ab. Dies ist insbesondere dann interessant, wenn wenig Wissen über einen Entwicklungsprozess existiert. Die Ansätze der Simulation beschreiben Wirkzusammenhänge detaillierter und bezogen auf einen bestimmten Anwendungskontext. Für diese Arbeit sind beide Ansätze sowie auch deren Kombination zu unterstützen, um eine problembezogene Modellierung zu erlauben.

Anforderung 8. Methodik. In allen betrachteten Bereichen werden Vorgehen beschrieben, die sich allerdings ausschließlich auf beschränkte Aspekte konzentrieren. Bei einem Blick auf die Einflussanalyse von Methodenänderungen werden im Bereich Simulation Vorgehen vorgeschlagen, welche die Methoden- und Prozessbeschreibung vernachlässigen und auch nicht bis zu einem Vergleich alternativer Entwicklungsprozesse reichen. Es gilt für diese Arbeit also existierende Ansätze und Vorgehen zu kombinieren und ein Instrument für entsprechende Einflussanalysen bereitzustellen.

Anforderung 9. Verfeinerung. Die untersuchten Ansätze zur Simulation von Entwicklungsprozessen erlauben grundsätzlich eine Beschreibung von Wirkzusammenhängen auf unterschiedlichen Abstraktionsebenen. Allerdings wird über die Ansätze der Wiederverwendung von Modellen in der Simulation kein Ansatz beschrieben, welcher einen einfachen Austausch unterschiedlicher Modelle etwa verschiedener Abstraktionsebenen erlaubt.

Anforderung 10. Werkzeugunterstützung. Sowohl für die Beschreibung von Methoden und Prozessen als auch für die Simulation von Prozessen existieren entsprechend spezialisierte Werkzeuge. Insbesondere erlauben Simulationswerkzeuge eine hohe Flexibilität für die Modellierung von Simulationsmodellen, eine Methoden- und Prozessbeschreibung wird bei den eingesetzten Werkzeugen der betrachteten Ansätze jedoch nicht unterstützt. Auch die Einflussanalyse und Einflussbewertung steht bei den verwendeten Simulationswerkzeugen nicht im Fokus.

Die Bewertung der Anforderungen gegenüber betrachteten Arbeiten zeigt, dass verschiedene Lücken mit Blick auf den Lösungsansatz dieser Arbeit existieren. In den folgenden Kapiteln sollen diese Lücken durch die vorliegende Arbeit geschlossen werden. Im Kapitel 3 wird dazu die hier entwickelte Methodik am Beispiel des Entwicklungsprozesses aus Abschnitt 1.5 vorgestellt. Methoden, Entwicklungsprozesse und Wirkzusammenhängen werden dabei auf Basis eines Metamodells beschrieben, welches ergänzend dazu in Kapitel 4 definiert wird.

3 Methodik zur Einflussbewertung

Das Ziel der Methodik ist die Bereitstellung eines Instruments, welches die Untersuchung von Änderungen in Entwicklungsprozessen und den damit verbundenen Einflüssen auf die Dimensionen von Zeit, Kosten und Produktqualität vor der eigentlichen Implementierung einer solchen Änderung erlaubt.

Im Abschnitt 3.1 werden zunächst Vorgehen vorgestellt, welche in der Literatur für die Entwicklung von Simulationsmodellen beziehungsweise die Evaluation neuer Methoden und Werkzeuge vorgeschlagen werden. Darauf aufbauend werden die Beiträge dieser Arbeit illustriert (Abschnitt 3.2) und die entwickelte Methodik in den dann folgenden Abschnitten vorgestellt.

3.1 Vorgehen für die Entwicklung von Simulationsmodellen

Rus et al. beschreiben eine systematische Methodik zur Entwicklung von diskreten Simulationsmodellen von Software-Entwicklungsprozessen (Rus et al. 2003). Ihre Motivation liegt dabei in der Verbesserung des Vorgehens von ad-hoc Methoden zu einem definierten Vorgehen, wie es für den Bereich der System Dynamics beschrieben ist (Pfahl 2001), (Rus 1998). Zwar fokussieren die Autoren dabei auf Entwicklungsprozesse von Software und legen ebenso den Schwerpunkt der Anwendung auf die diskrete Simulation von Prozessmodellen, doch sind die grundlegenden Schritte auch für andere Domänen anwendbar. Rus et al. verwenden für die Entwicklung des Simulationsmodells einen paarweisen Ansatz. Bei diesem Ansatz entwickeln und diskutieren die Entwickler des Modells und Auftraggeber ein gemeinsames Simulationsmodell. Das vorgestellte Vorgehen gliedert sich im Wesentlichen in:

1. Identifikation und Spezifikation der Anforderung beziehungsweise der Ziele und Fragen, die mit Hilfe des Simulationsmodells beantwortet werden sollen,
2. Analyse und Erstellung eines statischen Prozessmodells,
3. Beschreibung der Beziehungen zwischen bestimmten Prozessparametern,
4. Sammlung von empirischen Daten und Quantifizierung von Beziehung und
5. Implementierung, Simulation, Validierung und Verifikation.

Zur Ermittlung des ökonomischen Einflusses von Prozessvariationen im Bereich der Softwareentwicklung verwenden Deissenboeck und Pizka (Deissenboeck und Pizka 2007) ein Prozessmodell auf Basis von Markov-Ketten. Die Autoren benutzen das folgende Vorgehen zur Einflussbewertung einer Prozessänderung auf die Dimension Zeit:

1. Definition eines Referenzmodells für den Anwendungsbereich,
2. Kalibrierung des Referenzmodells durch empirische Studien,
3. Parametrisierung von Übergangswahrscheinlichkeiten zwischen Aufgaben,
4. Beschreibung der Prozessalternativen auf Basis des Referenzmodells und
5. Simulation der Prozessmodelle und Vergleich der Ergebnisse.

Basili beschäftigt sich mit der quantitativen Evaluation von Methoden und Werkzeugen zur Softwareentwicklung (Basili 1985). Zur Ausrichtung einer Untersuchung wird der Goal-Question-Metric (GQM) Ansatz verwendet (Basili et al. 1994), bei dem zunächst eine Menge von Zielen definiert wird von denen dann quantifizierbare Fragen abgeleitet und diese in messbare Metriken überführt werden. Ziele sind dabei recht vage definiert und geben eine Idee über das zu Erreichende, die dann bis zu Metriken detailliert werden. Das folgende Vorgehen wird dazu vorgeschlagen:

1. Generierung einer Menge von Zielen auf Basis der organisationspezifischen Anforderungen,
2. Ableitung einer Menge von zielgerichteten Fragen oder Hypothesen, welche die definierten Ziele quantifizieren,
3. Entwicklung einer Menge von Daten-Metriken und Daten-Verteilungen, welche die notwendigen Informationen zur Beantwortung der Fragen beinhalten,
4. Definition eines Vorgehens zur möglichst genauen Sammlung von Daten,
5. Validierung der gesammelten Daten und
6. Analyse der gesammelten Daten zur Beantwortung der Fragen.

Damit ist diese Methode auch im Rahmen dieser Arbeit interessant, wenn es darum geht, welche Fragestellungen mit der Einflussanalyse untersucht werden sollen und welche Daten dafür zu erheben sind.

Auch wenn die genannten Vorgehen zur Analyse und Bewertung von Entwicklungsprozessen aus der Domäne der Softwareentwicklung stammen, so ist ihr Vorgehen doch relativ generisch und kann damit auch auf andere Entwicklungsprozesse angewendet werden. Dementsprechend werden die genannten Vorgehen für die Definition der in dieser Arbeit entwickelten Methodik herangezogen.

3.2 Beiträge dieser Arbeit

Die Beiträge dieser Arbeit werden, wie Abbildung 3.1 zeigt, auf mehrere Kapitel aufgeteilt. Kapitel 3 beschreibt die Methodik zur Einflussanalyse und Einflussbewertung von Methodenänderungen. In Kapitel 4 wird ein Metamodell zur Beschreibung von Entwicklungsprozessen und Wirkzusammenhängen definiert, welches auch für die Simulation alternativer Entwicklungsprozesse verwendet wird. Zudem ist für die effiziente Modellierung, Simulation und Einflussanalyse eine entsprechende Werkzeugunterstützung notwendig, welche in Kapitel 5 vorgestellt wird.

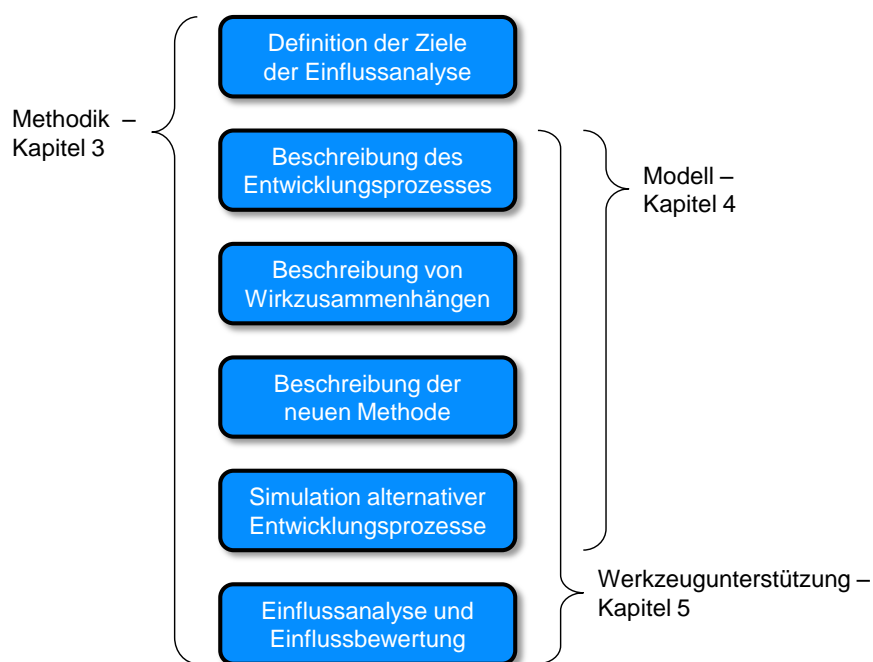


Abbildung 3.1 Überblick über die Beiträge dieser Arbeit

Bevor die einzelnen Schritte der Methodik erläutert werden, wird im Folgenden Abschnitt zunächst ein Überblick über das Vorgehen gegeben.

3.3 Überblick der Methodik

Die Einflussanalyse wird mit der Festlegung der mit der Untersuchung verbundenen Ziele und Fragestellungen begonnen (Abschnitt 3.4). Die Ziele der Einflussanalyse sind für alle folgenden Schritte von Interesse und bestimmen wesentlich deren Ausrichtung.

Für die Beschreibung von Entwicklungsprozessen wird zunächst ein Modell aus dem existierenden Entwicklungsprozess abgeleitet (Abschnitt 3.5), welches die wesentlichen Prozesselemente wie Aufgaben, Design Artefakte, Rollen und Werkzeuge beinhaltet (Ist-Prozess). Im nächsten Schritt folgt die Identifizierung und Beschreibung von

Wirkzusammenhängen, welche kritische Eigenschaften von Prozesselementen und deren Wirkzusammenhänge qualitativ beschreibt und eine Grundlage für die Diskussion von Wirkzusammenhängen bildet (Abschnitt 3.6). Im Anschluss an die qualitative Beschreibung von Wirkzusammenhängen ist es notwendig, empirische Daten für eine Quantifizierung der angenommenen Wirkzusammenhänge zu sammeln und diese zu analysieren (Abschnitt 3.7). Damit stehen charakteristische Daten über den betrachteten Entwicklungsprozess und dessen Ausprägungen von Wirkzusammenhängen zur Verfügung, welche zur anwendungsspezifischen Quantifizierung und Kalibrierung der modellierten Wirkzusammenhänge dienen (Abschnitt 3.8).

Bei jedem Schritt erfolgt eine Validierung der Modellierung gegen die Erfahrung von Experten des Entwicklungsprozesses und gegebenenfalls werden Rückschritte und Überarbeitungen der Modelle vorgenommen, bis diese der Erfahrung entsprechen (Abschnitt 3.9). Die zuvor skizzierten Schritte liefern damit ein valides Modell des existierenden Entwicklungsprozesses.

Parallel beziehungsweise im Anschluss an die Modellierung des existierenden Entwicklungsprozesses wird die Methodenänderung beschrieben. Dafür sind die gleichen Schritte wie zuvor für den existierenden Entwicklungsprozess durchzuführen, wobei der Fokus insbesondere auf Änderungen am Entwicklungsprozess gelegt wird (Abschnitt 3.10). Mit der Beschreibung der neuen Methode beziehungsweise der Prozessänderung erfolgt dann die virtuelle Integration der neuen Methode in den existierenden Entwicklungsprozess (Abschnitt 3.11). Damit stehen zwei Modelle alternativer Entwicklungsprozesse zur Verfügung, die simuliert und gegeneinander verglichen werden können.

Im letzten Schritt werden die Prozessmodelle gegeneinander simuliert und bewertet. Die Simulation der Modelle der Entwicklungsprozesse liefert dabei ein charakteristisches Bild über das Verhalten der Entwicklungsprozesse (Abschnitt 3.12). Dieses Bild beinhaltet eine Übersicht von mehr oder weniger wahrscheinlichen Prozessverhalten und damit Ergebnisse für mehr oder weniger wahrscheinliche Ausprägungen der Dimensionen von Zeit, Kosten und Qualität. Wichtig hierbei ist wiederum die Validierung der Ergebnisse. Sollten die Ergebnisse nicht den Erwartungen entsprechen beziehungsweise nicht sinnvoll interpretierbar sein, so ist das Modell des existierenden Entwicklungsprozesses oder entsprechend das Modell von alternativen Entwicklungsprozessen zu überarbeiten.

Im Anschluss an die Simulation erfolgt die Bewertung des charakteristischen Bildes des jeweiligen Prozessverhaltens (Abschnitt 3.13). Inhalt dieser Bewertung ist der Vergleich zwischen den Prozessalternativen und die strategische Interpretation der Ergebnisse für die definierten Ziele der Einflussanalyse. Unterschiedliche Szenarien der Prozessalternativen können dabei durch Sensitivitätsanalysen exploriert werden (Abschnitt 3.14).

Die Methodik zielt auf den kontinuierlichen Aufbau von Wissen über Methoden und Entwicklungsprozesse und unterstützt daher auch Einflussanalysen, wenn zunächst eher wenige Daten oder Annahmen bekannt sind. Mit jeder Iteration des Vorgehens wird daher angestrebt weiteres Wissen über den Entwicklungsprozess aufzubauen, was wiederum bessere Aussagen über den Einfluss einer möglichen Methodenänderung ermöglicht. In der Konsequenz führen damit erste Untersuchungen beispielsweise durch Expertenbefragungen hin zu Einflussbewertungen basierend auf einer automatisierten Sammlung und Analyse von operativen Prozess- und Produktdaten.

Die einzelnen Schritte der Methodik werden nun in den Abschnitten 3.4 bis 3.14 erläutert. Dabei wird das in Abschnitt 1.5 vorgestellte Beispiel eines Entwicklungsprozesses zur Illustration der einzelnen Schritte herangezogen. Darüber hinaus wird in Abschnitt 3.15 eine Einführungsstrategie für Einflussanalysen und Einflussbewertungen für konkrete Anwendungen vorgeschlagen.

3.4 Definition der Ziele der Einflussanalyse

Die Untersuchung des Einflusses einer neuen Methode oder allgemein einer Prozessänderung auf einen Entwicklungsprozess beginnt mit der Definition und Dokumentation der mit der Untersuchung einhergehenden Ziele und Fragestellungen. Auch wenn der in dieser Arbeit verfolgte Ansatz die detaillierte Beschreibung von Entwicklungsprozessen und Wirkzusammenhängen fordert, wäre es ein hoffnungsloses und nicht zielführendes Unterfangen, „alles“ zu modellieren. Vielmehr sind der Fokus und das Abstraktionsniveau der Untersuchung zuvor festzulegen.

Erst wenn die Ziele der Einflussanalyse definiert sind, ist eine zielgerichtete und effiziente Modellierung des existierenden Entwicklungsprozesses sowie des alternativen Entwicklungsprozesses möglich (Law und Kelton 1991), (Johnson et al. 1996), (Bossel 2004). Im Kontrast dazu führt ein zu breit gewähltes Zielgebiet beziehungsweise ein Verzicht auf die Definition von Zielen zu einer Aufweichung der Einflussanalyse von zugrundeliegenden Modellen. In der Konsequenz werden auch die möglichen Aussagen der Einflussanalyse aufgeweicht und beantworten nicht adäquat die eigentlichen Fragestellungen der Untersuchung.

Im Allgemeinen werden mit der Einführung einer neuen Methode Verbesserungen bezüglich der benötigten Zeit, der Kosten und der Produktqualität angestrebt. Diese Dimensionen müssen weiter detailliert werden, um die zielgerichtete Modellierung und spätere Validierung der Ziele vornehmen zu können.

Mit Hilfe des Goal-Question-Metric (GQM) Ansatzes werden einzelne Ziele in spezifische Fragestellungen herunter gebrochen, welche wiederum in messbare Metriken detailliert werden (Basili et al. 1994). Dieser Zusammenhang ist in Abbildung 3.2

veranschaulicht. Metriken geben dabei Hinweise auf die ein Ziel beeinflussenden Elemente oder Eigenschaften von Prozesselementen, die es bei der Modellierung zu beachten gilt. Entsprechend wird durch dieses Vorgehen der mögliche Raum zur Modellierung der relevanten Charakteristika des Entwicklungsprozesses und der neuen Methode zielgerichtet eingeschränkt.

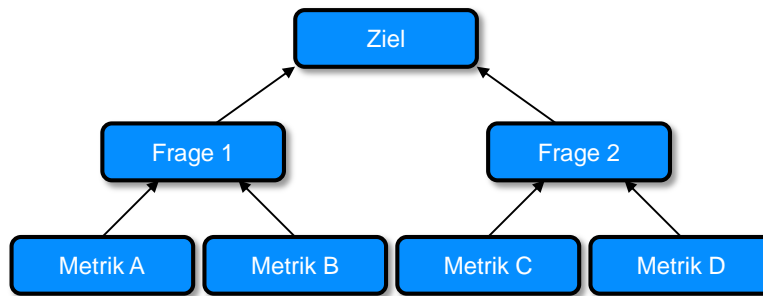


Abbildung 3.2 Der Goal Question Metrik Ansatz

3.4.1 Anwendung

Für das Szenario der High Level Synthesis interessiert im Wesentlichen die drastische Reduzierung des benötigten Entwicklungsaufwands und die Reduzierung der damit verbundenen Kosten. Für die Einflussbewertung sind dementsprechend die Aufwände und die Kosten des bisherigen Entwicklungsprozesses dem alternativen Entwicklungsprozess gegenüberzustellen. Somit lassen sich etwa die in Tabelle 3.1 dargestellten drei Ebenen von Zielen, Fragen und Metriken ableiten. Ähnliche Strukturen können auch für die Dimensionen von Kosten und Produktqualität aufgebaut werden.

Tabelle 3.1 Definition von Zielen, Fragen und Metriken

Ziel	<ul style="list-style-type: none"> • Reduziere den notwendigen Aufwand des betrachteten Entwicklungsprozesses aus Sicht des Prozessverantwortlichen
Frage	<ul style="list-style-type: none"> • Was ist der aktuelle minimale, maximale und wahrscheinlichste Aufwand des Entwicklungsprozesses?
Metrik	<ul style="list-style-type: none"> • Summe der Aufwände jeder ausgeführten Aufgabe • Subjektive Einschätzung des Prozessverantwortlichen

3.4.2 Methoden

Um die Ziele, Fragen und Metriken zu definieren, sind verschiedene Methoden einsetzbar, bei denen im Wesentlichen Prozessverantwortliche mitwirken. Da zu Beginn der Untersuchung zunächst die Richtung festgelegt werden muss, sind Methoden zweckmäßig, die den Ergebnisraum nicht von Anfang an zu sehr einschränken. Zu diesen Methoden zählen

- **Interviews** mit Prozessverantwortlichen und gegebenenfalls Entwicklern zur Festlegung der Ziele und Fragen sowie zur Identifikation nutzbarer Metriken oder
- **Fokusgruppen** in denen mehrere Prozessverantwortliche und Entwickler zusammen die Ziele, Fragen und Metriken entwickeln.

Tabelle 3.5 auf Seite 96 gibt eine Übersicht der genannten Methoden.

Für die nächsten Schritte der Beschreibung des existierenden Entwicklungsprozesses und der neuen Methode sind die Ziele, Fragen und Metriken eine wichtige Grundlage. Im Folgenden wird die in diesem Abschnitt festgelegte Untersuchungsrichtung durch die Modellierung der Entwicklungsprozesse und der inhärenten Wirkzusammenhänge detailliert.

3.5 Beschreibung des Entwicklungsprozesses

Die Beschreibung des Entwicklungsprozesses zielt auf das Verstehen und die Dokumentation des existierenden Entwicklungsprozesses. So wird in diesem Schritt eine erste Verfeinerung des Problems vorgenommen, welches durch die Definition der Ziele, Fragen und Metriken aus Abschnitt 3.4 beschrieben wurde. Dementsprechend wird ein Modell des Entwicklungsprozesses aufgebaut, welches Prozesselemente mit einem signifikanten Einfluss auf die beschriebenen Ziele beschreibt. – Auf die Definition des Metamodells zur Modellierung von Entwicklungsprozessen und Wirkzusammenhängen wird in Kapitel 4 detaillierter eingegangen.

Zunächst werden die wesentlichen Prozesselemente identifiziert und modelliert. Dabei werden wiederverwendbare Definitionen von Prozesselementen beschrieben, die dann für den konkreten Entwicklungsprozess instanziiert werden.

Aktivitäten

Aktivitäten sind solche Tätigkeiten, die sinnvoll in kleinere Aufgaben herunter gebrochen werden können. Für die zielorientierte Modellierung ist hier die Frage zu stellen, ob die Beschreibung einzelner Aufgaben innerhalb einer Aktivität einen Mehrwert für die Untersuchung darstellt. Dieses ist im Allgemeinen der Fall, wenn

- der Aufwand beziehungsweise die Kosten einer einzelnen Aufgabe interessieren,
- eine Aufgabe einen signifikanten Einfluss auf Design Artefakte besitzt oder
- eine Aufgabe durch die neue Methode signifikant beeinflusst, ausgetauscht oder hinzugefügt wird.

Werden einzelne Aufgaben im Rahmen des Anwendungskontextes vorwiegend gemeinsam betrachtet, so ist eine besondere Modellierung einzelner Aufgaben innerhalb einer Aktivität zumindest für die Einflussanalyse nicht erforderlich. So lassen sich mehrere

Aufgaben zu einer aggregierten Aufgabe zusammenfassen. Soll die Modellierung gleichzeitig als Wissensbasis im Sinne des Method Engineering verwendet werden oder ist eine genaue Prozessdokumentation von Interesse, so ist entsprechend eine anwendungsspezifische Entscheidung über den Detaillierungsgrad zu treffen.

Aufgaben

Aufgaben sind solche Tätigkeiten, die Design Artefakte erstellen, bearbeiteten oder verwenden. Die Arbeit, welche durch eine Aufgabe beschrieben wird, wird durch andere Design Artefakte, Rollen oder Werkzeuge beeinflusst. Dementsprechend ist für die Modellierung einer Aufgabe zu untersuchen, welche Prozesselemente eine Bedeutung für eine Aufgabe haben. Assoziierte Prozesselemente sind zu modellieren, wenn

- diese erforderlich sind um die Aufgabe durchzuführen,
- das Ergebnis der Aufgabe darstellen oder
- die durch die Aufgabe ausgedrückte Arbeit signifikant beeinflussen.

Analyse Aufgaben sind solche Aufgaben, welche die Ergebnisse vorheriger Aufgaben, also ausgehende Design Artefakte, überprüfen und validieren. Als solche Aufgaben verfügen sie über bestimmte Ziele, die an Eigenschaften von Design Artefakten gestellt werden. Die Erfüllung oder Nichterfüllung von Zielen beeinflusst dabei die Wahl der folgenden Aufgabe, was durch ein Entscheidungsmodell abgebildet wird. Analyse Aufgaben werden wie einfache Aufgaben modelliert. Ziele und Entscheidungsmodelle werden dann im konkreten Prozesskontext definiert, siehe dazu Abschnitt 3.5.1.

Design Artefakte

Design Artefakte stellen notwendige Eingaben für Aufgaben dar oder werden durch Aufgaben erstellt und bearbeitet. Design Artefakte sind also zu modellieren, wenn diese

- eine wesentliche Eingabe für eine Aufgabe darstellen,
- durch eine Aufgabe erstellt und bearbeitet werden und
- die Eigenschaften eines Design Artefakts Aufgaben wesentlich beeinflussen.

Die Qualität und Komplexität eines Design Artefakts beeinflussen die Anzahl notwendiger Iterationen zur Überarbeitung eines Design Artefakts, den notwendigen Aufwand sowie die Kosten einer entsprechenden Aufgabe.

Charakteristische Eigenschaften von Design Artefakten sind für den jeweiligen Anwendungskontext auf Basis der Metriken zu konkretisieren. Tabelle 3.2 zeigt einige Beispiele, die als charakteristisch für ein Design Artefakt gelten. Es ist unbedingt darauf zu achten, dass die Modellierung aller möglichen Eigenschaften nicht sinnvoll ist. Ist beispielsweise nicht zu erwarten, dass eine bestimmte Eigenschaft gut abzuschätzen oder während der Datenerhebung und Datenanalyse extrahiert werden kann, so sollte

eine aggregierte Eigenschaft wie „Qualität“ verwendet werden, die im Anwendungskontext zu definieren ist.

Tabelle 3.2 Eigenschaften von Artefakten

Eigenschaft	Beispiele für Quellen
<ul style="list-style-type: none"> • Qualität <ul style="list-style-type: none"> ○ Anforderungserfüllung ○ Funktionale Korrektheit ○ Anzahl der Fehler ○ Wartbarkeit • Umfang <ul style="list-style-type: none"> ○ Anzahl der Module ○ Function Points ○ Lines of Code • Komplexität 	(Boehm et al. 1976) (Madachy 1996) (Rus et al. 1999) (Park et al. 2007) (Müller und Pfahl 2008) (Choi und Bae 2009)

Rollen

Rollen beeinflussen ähnlich wie Artefakte die Arbeit, welche durch eine Aufgabe repräsentiert wird, als auch Eigenschaften von assoziierten Prozesselementen. Assoziierte Prozesselemente sind dabei etwa an einer Aufgabe beteiligte Rollen oder Design Artefakte als Ergebnis einer Aufgabe. Für die Modellierung des Entwicklungsprozesses sind Rollen zu betrachten, die einen wesentlichen und nach Möglichkeit messbaren Einfluss auf andere Prozesselemente haben. Dieses sind im Wesentlichen Rollen, die primär mit der Durchführung einer Aufgabe betraut sind. Ist der zu erwartende Einfluss einer Rolle auf eine Aufgabe marginal, beispielsweise bei automatisierten Aufgaben, so kann auf eine Modellierung für die Einflussanalyse verzichtet werden.

Tabelle 3.3 zeigt Beispiele für Eigenschaften von Rollen, die einen Einfluss auf andere Prozesselemente besitzen. Auch hier ist es nicht zweckmäßig beliebige Eigenschaften zu modellieren. Vielmehr ist abzuwägen, welche Eigenschaften als charakteristisch betrachtet werden können.

Tabelle 3.3 Eigenschaften von Rollen

Eigenschaft	Beispiele für Quellen
<ul style="list-style-type: none"> • Kosten pro Stunde • Anzahl der Entwickler • Erfahrung • Fähigkeiten • Motivation, Teamkompetenz 	(Madachy 1996) (Rus et al. 1999) (Park et al. 2007) (Müller und Pfahl 2008) (Choi und Bae 2009)

Werkzeuge

Werkzeuge werden verwendet, um eine bestimmte Arbeit innerhalb einer Aufgabe durchzuführen. Dabei werden Werkzeuge im Allgemeinen von Rollen bedient, was bei der Modellierung von Eigenschaften dieser Prozesselemente beachtet werden sollte. Werkzeuge dienen der Bearbeitung oder Überprüfung von Design Artefakten und besitzen daher bestimmte Eigenschaften, die einen Einfluss auf die Dauer und Kosten einer Aufgabe oder auf assoziierte Artefakte besitzen. Auch hier gilt, dass zu erwartende Einfluss eines Werkzeugs auf die betrachtete Aufgabe bestimmt, ob es modelliert wird oder ob auf eine Beschreibung verzichtet werden kann.

Der Einfluss eines Werkzeugs auf eine Aufgabe kann schwierig zu bewerten sein und lediglich indirekt durch den mit einer Aufgabe verbundenen Aufwand charakterisiert sein. Sollten die Eigenschaften eines Werkzeugs also schwierig zu erfassen sein und sich die Untersuchung nicht auf einen Austausch eines bestimmten Werkzeugs beziehen, kann auf eine entsprechende Modellierung verzichtet werden.

Für Werkzeuge können etwa die in Tabelle 3.4 abgebildeten Eigenschaften von Bedeutung sein.

Tabelle 3.4 Eigenschaften von Werkzeugen

Eigenschaft	Beispiele für Quellen
<ul style="list-style-type: none"> • Lizenzkosten • Usability • Fehlererkennungsrate • Automatisierungsgrad 	(Müller und Pfahl 2008)

Eigenschaften

Design Artefakte, Rollen und Werkzeuge verfügen also über charakteristische und anwendungsspezifische Eigenschaften. Die in Abschnitt 3.4 definierten Ziele, Fragen und Metriken dienen als Grundlage für die Identifizierung und Auswahl von notwendigen Eigenschaften. Für identifizierte Eigenschaften und die Beschreibung von Wirkungszusammenhängen sollte bereits frühzeitig festgestellt werden,

- ob Daten bereits existieren,
- welche Daten zu sammeln sind,
- woher und wie entsprechende Daten bezogen werden können und
- wer für die Bereitstellung der Daten verantwortlich ist.

3.5.1 Anwendung

Abbildung 3.3 zeigt das Ergebnis der Identifikation und Definition der Prozesselemente am Beispiel aus Abschnitt 1.5. Die Definition besteht aus den Aufgaben *Behavioral Design*, *RTL Design* und *Analysis*. Die Rolle *Entwickler* ist an allen drei Aufgaben beteiligt und besitzt mit der charakteristischen Eigenschaft *Erfahrung* einen wesentlichen Einfluss auf den Aufwand der Aufgaben, welcher untersucht werden soll. Die Aufgabe *Behavioral Design* erstellt und bearbeitet das *C/C++* Artefakt, welches über die Eigenschaft *funktionale Korrektheit* verfügt. Das *C/C++* Artefakt stellt eine notwendige Eingabe für die Aufgabe *RTL Design* dar, bei der das *C/C++* Artefakt manuell durch den *Entwickler* in ein *VHDL* Artefakt transformiert beziehungsweise dieses bearbeitet wird. Auch das *VHDL* Artefakt verfügt über eine charakteristische Eigenschaft zur Beschreibung der funktionalen Korrektheit, welche von der *Analysis* Aufgabe überprüft wird.

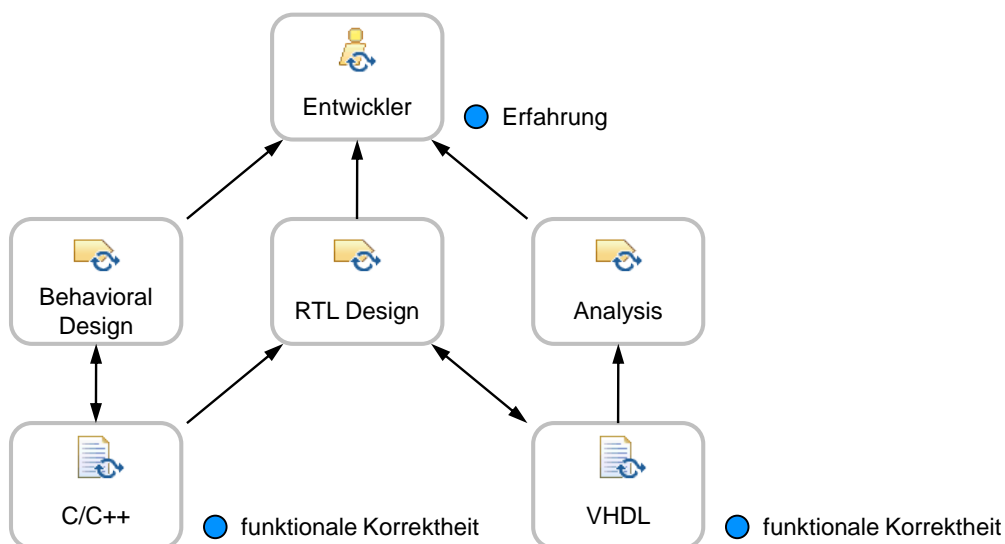


Abbildung 3.3 Definition von Prozesselementen des Entwicklungsprozesses

Für die obige Definition von Prozesselementen lässt sich der Ablauf des Entwicklungsprozesses beinahe direkt aus den Eingaben und Ausgaben der Aufgaben ablesen. Für die konkrete Prozessbeschreibung wird nun allerdings der Ablauf des Prozesses formal gemacht. Abbildung 3.4 zeigt die Prozessbeschreibung auf Basis der definierten Prozesselemente. Im Vergleich der beiden Abbildungen zeigt sich, dass ein Rücksprung von der *Analysis* Aufgabe zu den Aufgaben *Behavioral Design* und *RTL Design* möglich ist, der aus der alleinigen Definition der Prozesselemente nicht hervorgeht. Für das Beispiel wird angenommen, dass *Analysis* über ein Ziel verfügt, welches auf der Eigenschaft *funktionale Korrektheit* des *VHDL* Artefakts definiert ist und einen Zielwert von 100% besitzt.

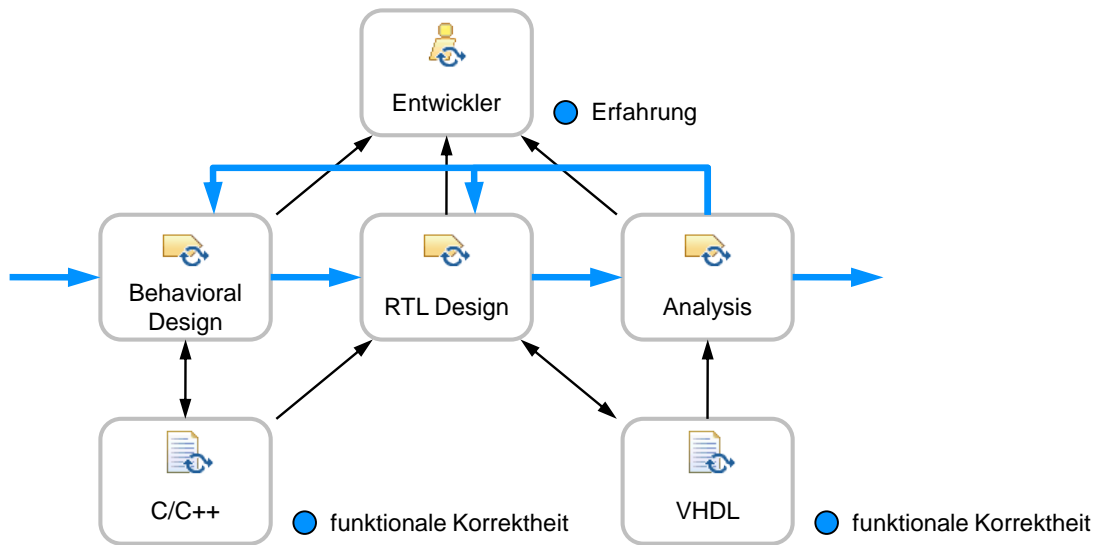


Abbildung 3.4 Instanziierung der definierten Prozesselemente zu einem konkreten Prozessmodell

Die *Analysis* Aufgabe verfügt also über ein Entscheidungsmodell, welches die Eigenschaft der funktionalen Korrektheit gegen den Zielwert prüft und je nach Erfüllung die folgende Aufgabe bestimmt. Für das Beispiel könnte ein Entscheidungsmodell in Pseudocode wie folgt aussehen:

```

Wenn funktionale Korrektheit = 100% dann
    Beende den Prozess
Sonst
    P = Zufallszahl zwischen 0 und 1
    Wenn P <= 0.25 dann
        Wähle Behavioral Design als folgende Aufgabe
    Sonst
        Wähle RTL Design als folgende Aufgabe
  
```

3.5.2 Methoden

Zur Identifizierung von Prozesselementen des Entwicklungsprozesses können vorhandene **Dokumentation** oder auch modellierte Prozesse, etwa als Ereignisgesteuerte Prozessketten oder in der Business Process Model and Notation aber auch Projektpläne beziehungsweise Netzpläne herangezogen werden.

Bei der Verwendung existierender Dokumentation ist sicherzustellen, dass diese Dokumentation dem tatsächlichen Leben des Entwicklungsprozesses entspricht, damit ein valides Modell über den Entwicklungsprozess für Einflussanalysen entwickelt werden kann. **Unstrukturierte** oder **teil-strukturierte Interviews** mit Prozessverantwortlichen helfen bei der Validierung der vorhandenen Dokumentation. Zudem sind Interviews und **Conceptual Modeling** dazu geeignet, Modelle des Entwicklungsprozesses aufzubauen, wenn keine Dokumentation vorhanden ist oder diese nicht dem Leben des Entwick-

lungsprozesses entspricht. Dazu können auch kleine **Fokusgruppen** mit 2 bis 4 Experten durchgeführt werden. Hierbei kann von der Interaktion der Beteiligten und unterschiedlichen Einsichten in den Entwicklungsprozess profitiert werden.

Neben der Ermittlung von Prozesselementen und deren Beziehungen zueinander, spielt die Bestimmung der den Entwicklungsprozess beeinflussenden Eigenschaften von Prozesselementen eine kritische Rolle. Um diese Eigenschaften zu ermitteln eignen sich teil-strukturierte Interviews oder **Brainstorming**. Wobei bei beiden Methoden auf eine zielgerichtete Umsetzung zu achten ist. So ermittelte Eigenschaften werden in einer Liste von Eigenschaften gesammelt, (Lindemann 2007), (Frankenberger 1997), um ihren Einfluss bei der Identifizierung von Wirkzusammenhängen näher zu beschreiben.

Neben der Identifizierung von Eigenschaften sollte auch bereits eine initiale Bewertung des angenommenen Einflusses in einer qualitativen Form notiert werden. Das Ziel dabei ist die Reduzierung des notwendigen Aufwands für die Modellierung von Eigenschaften, die lediglich einen marginalen Einfluss besitzen.

Tabelle 3.5 auf Seite 96 gibt eine Übersicht der genannten Methoden.

3.6 Identifizierung und Beschreibung von Wirkzusammenhängen

Der vorhergehende Schritt liefert neben der Definition von wiederverwendbaren Prozesselementen ein Prozessmodell für einen konkreten Entwicklungsprozess. Jedes definierte Prozesselement verfügt über eine Reihe von charakteristischen Eigenschaften. In diesem Schritt werden Wirkzusammenhänge zwischen Prozesselementen beziehungsweise den Eigenschaften von Prozesselementen explizit modelliert. Dabei können im Laufe der Modellierung von Wirkzusammenhängen Eigenschaften von Prozesselementen erweitert oder revidiert werden, wenn ein entsprechender Bedarf durch die Identifizierung von Wirkzusammenhängen entsteht.

Grundsätzlich ergeben sich existierende Wirkzusammenhänge aus der Definition von Prozesselementen des Entwicklungsprozesses. So haben die mit einer Aufgabe verbundenen Prozesselemente in der Regel auch eine Auswirkung auf die Aufgabe. Abbildung 3.5 strukturiert diesen grundsätzlichen Zusammenhang in einem Wirkungsdiagramm, welches auch für die Beschreibung von Wirkzusammenhängen eingesetzt wird.

Design Aufgaben bearbeiten oder erstellen Artefakte mit bestimmten Eigenschaften (*Output Produkt Eigenschaft*). Diese Eigenschaften werden durch Eigenschaften eingehender Design Artefakte (*Input Produkt Eigenschaften*), durchführender Rollen (*Rollen Eigenschaften*) oder verwendeten Werkzeugen (*Werkzeug Eigenschaften*) beeinflusst. Auf der anderen Seite validieren Analyse Aufgaben Eigenschaften eines Design Artefakts (*Output Produkt Eigenschaft*) gegen ein festgelegtes Ziel. Die Erfüllung eines

Ziels wird durch die *Qualität der Validierung* beeinflusst, welche wiederum von Eigenschaften beteiligter Rollen oder verwendeter Werkzeuge abhängt. Die Erfüllung oder Nichterfüllung der mit einer Analyse Aufgabe verbundenen Ziele bestimmt damit die Auswahl der folgenden Aufgabe. Entsprechend wird die *Erfolgswahrscheinlichkeit* maßgeblich von den zuvor genannten Eigenschaften beeinflusst. Für die Änderung von Eigenschaftswerten sowie die Entscheidung für folgende Aufgaben werden entsprechende Modelle beschrieben. Siehe dazu Abschnitt 4.2.8.

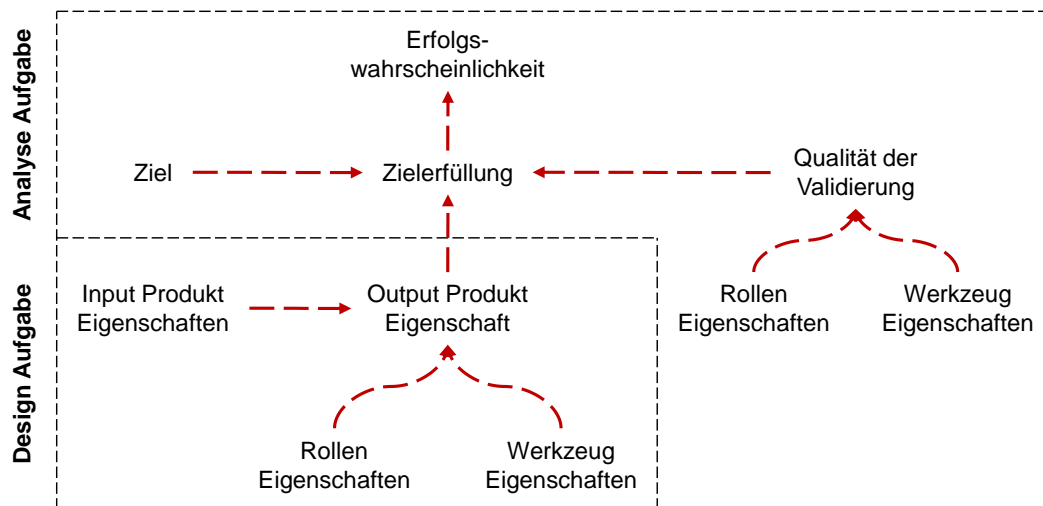


Abbildung 3.5 Grundsätzliche Wirkungen in einem Entwicklungsprozess

Wirkzusammenhänge ergeben sich also aus der Betrachtung der an einer Aufgabe beteiligten Prozesselemente und ihren charakteristischen Eigenschaften. Wesentlich bei der Identifizierung und Beschreibung von Wirkzusammenhängen ist auch in diesem Schritt die Beachtung der definierten Ziele der Untersuchung.

Wirkzusammenhänge werden zunächst qualitativ mit Hilfe von Wirkungsdiagrammen beschrieben. Dabei wird die Wirkrichtung einer Eigenschaft auf andere Eigenschaften durch Pfeile repräsentiert. Die Qualität der Wirkung wird dann üblicher Weise durch die Symbole „+“ für positive Effekte und „-“ für negative Effekte beschrieben. Die Anwendung dieser Notation wird im folgenden Abschnitt 3.6.1 demonstriert. Zudem erweist es sich als zweckmäßig, den Zusammenhang zwischen verschiedenen Eigenschaften zusätzlich textuell zu beschreiben, um eine korrekte Interpretation des angenommenen Wirkzusammenhangs zu erlauben. Beispiele zur Abbildung von Wirkzusammenhängen sind etwa bei (Rus et al. 2003), (Bossel 2004) und (Müller und Pfahl 2008) zu finden.

Einflussmatrix

Neben der einfachen qualitativen Beschreibung eines Wirkzusammenhangs mit Hilfe eines Wirkungsdiagramms kann der Einfluss von Eigenschaften auf andere Eigenschaften von Prozesselementen in Form einer Matrix qualitativ beschrieben werden. Dieses Vorgehen ist insbesondere geeignet, wenn viele Eigenschaften und Wirkzusammenhänge zu untersuchen sind.

Mit Hilfe einer Einflussmatrix lässt sich eine Gewichtung vornehmen, um die Stärke des Einflusses einer Eigenschaft abzuschätzen und eine Entscheidung für die Modellierung einer Eigenschaft innerhalb eines Wirkzusammenhangs zu treffen. Dabei werden die betrachteten Eigenschaften der Prozesselemente als Zeilen- und Spaltenbeschriftung verwendet. Der Einfluss einer Eigenschaft auf andere Eigenschaften wird durch Experten bewertet, wobei hier eine Skala von -3 bis 3 vorgeschlagen wird. Eine negative Bewertung stellt dabei einen entsprechend negativen beziehungsweise reduzierenden Einfluss dar. Entsprechendes gilt für eine positive Bewertung. Die Summe der Beträge einer Zeile beschreibt dann die Stärke des Einflusses einer Eigenschaft auf alle anderen betrachteten Eigenschaften. Für die Modellierung von Wirkzusammenhängen ist insbesondere die Summe der Beträge einer Spalte interessant. Hieraus ist abzulesen, wie groß der vermutete Einfluss von Eigenschaften (Zeilen) auf eine Eigenschaft (Spalte) ist. Das Produkt aus Zeilen- und Spaltensumme einer Eigenschaft wird als Dynamik-Index bezeichnet. Ein hoher Dynamik-Index charakterisiert mit hoher Wahrscheinlichkeit einen starken Einfluss einer Eigenschaft in Wirkzusammenhängen. Zum Thema Einflussmatrix siehe auch (Balazova 2004) und (Gausemeier et al. 2009).

Abbildung 3.6 veranschaulicht den Aufbau einer Einflussmatrix für fünf Eigenschaften exemplarisch. Aus der Matrix lassen sich mehrere Dinge ablesen. Beispielsweise wird die *Eigenschaft 2* lediglich durch *Eigenschaft 3*, dafür aber stark negativ beeinflusst. *Eigenschaft 3* wird durch alle betrachteten Eigenschaften beeinflusst. *Eigenschaft 3* hat mit dem Wert 49 den größten Dynamik-Index und kann daher als kritische Eigenschaft in Wirkzusammenhängen bewertet werden und sollte daher bei der weiteren Untersuchung besonders beachtet werden.

Im Gegensatz dazu weißt *Eigenschaft 2* einen geringen Dynamik-Index auf und beeinflusst andere Eigenschaften vermutlich eher gering. Hier kann im Anwendungskontext überlegt werden, die Eigenschaft aus Wirkzusammenhängen zu abstrahieren anstatt sie zu modellieren. Für spätere Iterationen der Methodik werden gegebenenfalls weitere Zusammenhänge zwischen Eigenschaften beachtet, als dieses für erste Iterationen der Methodik zielführend ist.

Einflussbewertung von Eigenschaften	Eigenschaft 1	Eigenschaft 2	Eigenschaft 3	Eigenschaft 4	Eigenschaft 5	Zeilensumme
Eigenschaft 1		0	3	2	-2	7
Eigenschaft 2	1		1	-1	0	3
Eigenschaft 3	-2	-2		3	0	7
Eigenschaft 4	0	0	2		3	5
Eigenschaft 5	1	0	-1	0		2
Spaltensumme	4	2	7	6	5	

Abbildung 3.6 Beispiel einer Einflussmatrix zur Charakterisierung der Stärke eines Einflusses einer Eigenschaft auf andere Eigenschaften

In der Anwendung der Einflussmatrix wird die Stärke des Einflusses einer Eigenschaft auf andere Eigenschaften durch Expertenbefragung etwa in Interviews oder Fokusgruppen erhoben. Zur Handhabung einer entsprechenden Einflussmatrix empfiehlt sich der Einsatz einer Tabellenkalkulation und die automatische Berechnung von Zeilen- und Spaltensummen, sowie des Dynamik-Indexes.

3.6.1 Anwendung

Das Modell des Entwicklungsprozesses aus Abschnitt 1.5 wird nun um die Beschreibung qualitativer Wirkzusammenhänge erweitert. So verfügt das Modell über drei Wirkzusammenhänge, welche die Annahmen über den Entwicklungsprozess repräsentieren. Abbildung 3.7 zeigt das entsprechende Modell des Entwicklungsprozesses.

Die gezeigten Beziehungen sind dabei folgendermaßen zu interpretieren:

- Die *Erfahrung* eines *Entwicklers* hat eine Auswirkung auf die Aufgabe *RTL Design*. Dabei führt eine höhere Erfahrung zu einem geringeren Aufwand und damit zu geringeren Kosten (-). Hinweis: Diese Annahme berücksichtigt hier nicht, dass verschiedene Entwickler einen unterschiedlichen Stundensatz erhalten könnten.
- Die *funktionale Korrektheit* des *C/C++* Artefakts beeinflusst die *funktionale Korrektheit* des *VHDL*. Besitzt das *C/C++* Artefakt eine hohe funktionale Korrektheit, so ist anzunehmen, dass die funktionale Korrektheit des *VHDL* auch hoch sein wird (+).
- Die *funktionale Korrektheit* des *VHDL* besitzt eine Auswirkung auf die Wahrscheinlichkeit notwendiger Iterationen. Eine hohe funktionale Korrektheit reduziert die Wahrscheinlichkeit für Iterationen (-). Entsprechend dieser An-

nahme werden dadurch auch der Aufwand und die Kosten für *RTL Design* reduziert.

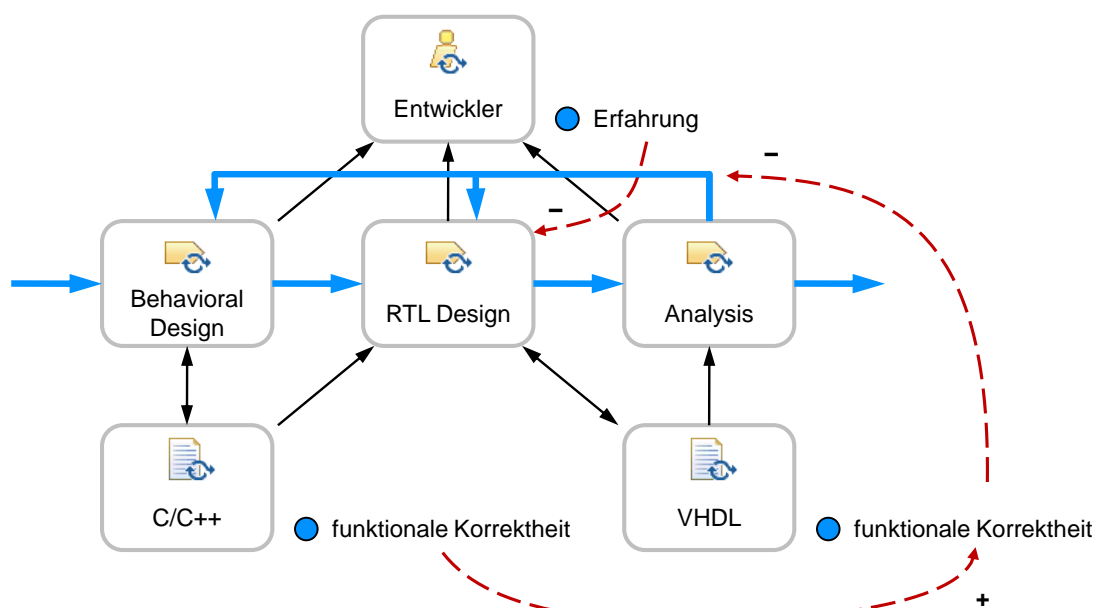


Abbildung 3.7 Beschreibung von Wirkzusammenhängen im Entwicklungsprozess

Der letzte Wirkzusammenhang ergibt sich durch die Beschreibung der Analyse Aufgabe. Denn diese überprüft die aktuelle Qualität *funktionale Korrektheit* des eingehenden *VHDL* Artefakts gegen ein definiertes Ziel. Wird dieses Ziel nicht erreicht, so ist eine Iteration zur Überarbeitung des *VHDL* Artefakts notwendig. Dementsprechend führt eine hohe *funktionale Korrektheit* des *VHDL* zur schnelleren Zielerfüllung und Fortsetzung des Entwicklungsprozesses.

3.6.2 Methoden

Zur Identifizierung von Wirkzusammenhängen eignen sich insbesondere **unstrukturierte** oder **teil-strukturierte Interviews** mit erfahrenen Entwicklern beziehungsweise Prozessverantwortlichen. Ergänzend dazu können **Brainstorming** und **Fokusgruppen** verwendet werden, um relativ schnell einen breiten aber zielgerichteten Überblick über existierende beziehungsweise angenommene Wirkzusammenhänge zu erhalten. Dabei sollte darauf geachtet werden, dass die Richtung sowie die angenommene Stärke eines Wirkzusammenhangs beschrieben wird, wenn dieses möglich ist. Brainstorming und Fokusgruppen erlauben in diesem Zusammenhang Synergien zwischen den beteiligten Personen und können bisher unbekannte oder implizite Wirkzusammenhänge ergeben. Die Einbeziehung von Prozessverantwortlichen kann hier sinnvoll sein, um einen Überblick über größere Zusammenhänge, wie etwa die

Auswirkungen auf unterschiedliche Aufgaben oder Aktivitäten, folgende Phasen oder andere Prozesse zu erhalten.

Tabelle 3.5 auf Seite 96 gibt eine Übersicht der genannten Methoden.

In den folgenden beiden Schritten werden die identifizierten Wirkzusammenhänge näher untersucht. Zu dieser Untersuchung zählen die Sammlung und Analyse empirischer Daten sowie die Quantifizierung der identifizierten Wirkzusammenhänge.

3.7 Sammlung und Analyse empirischer Daten

Die Sammlung und Analyse empirischer Daten ist die Voraussetzung für eine adäquate Modellierung des Entwicklungsprozesses und seiner Charakteristika. So führen nur sinnvolle und korrekte Daten zu validen quantitativen Modellen des Prozessverhaltens und damit zu belastbaren Aussagen über den Mehrwert einer neuen Methode. Auch bei der Sammlung empirischer Daten ist die Ausrichtung auf die im ersten Schritt festgelegten Ziele der Untersuchung wesentlich, um den notwendigen Aufwand für die Sammlung und Analyse der Daten in einem bestimmten Rahmen zu halten (Müller und Pfahl 2008).

Mit diesem Schritt sind folgenden Ziele verbunden:

- Bestimmung des Wertebereichs für Iterationen, Aufwand und Kosten einer jeden Aufgabe mit minimalen, wahrscheinlichen und maximalen Werten,
- Bestimmung des Wertebereichs von Eigenschaften mit minimalen, wahrscheinlichen und maximalen Ausprägungen,
- Entwicklung der Werte von Eigenschaften über die Zeit,
- Finden von Zusammenhängen gemäß der zuvor beschriebenen Wirkzusammenhänge und
- Validierung angenommener Wirkzusammenhänge.

Grundsätzlich können zwei Arten der Sammlung empirischer Daten unterschieden werden: die Extraktion von Expertenwissen sowie die automatische Sammlung von Produkt- und Prozessdaten. Die Extraktion von Expertenwissen eignet sich für die Einordnung von Ausprägungen sowie für die Validierung gesammelter Daten über Eigenschaften und Wirkzusammenhänge. Durch Expertenwissen können zum einen grundsätzlich bekannte oder angenommene Zusammenhänge festgestellt werden. Bei einzelnen Befragungen muss allerdings beachtet werden, dass diese möglicherweise eine hohe Unsicherheit beziehungsweise Risiko beinhalten. Entsprechend ist neben einem Mittelwert nach den besten, schlechtesten und wahrscheinlichsten Ausprägungen zu fragen und damit die mögliche Varianz abzubilden. Auf der anderen Seite können Produkt- und Prozessdaten wie etwa Qualitätseigenschaften (etwa Anzahl der Fehler), Eigenschaften des Produktumfangs (etwa Anzahl der Module), die aktuelle Iteration

und der Aufwand einer Aufgabe durch entsprechende Protokolle und Werkzeuge automatisch gesammelt werden.

Für erste Einflussanalysen sind durchaus Aussagen von Experten für die Quantifizierung von angenommenen Wirkzusammenhängen zweckmäßig. Insbesondere wenn bisher keine Daten für einen Wirkzusammenhang existieren oder der Aufwand für die Sammlung und Analyse entsprechender Daten unverhältnismäßig über den kritischen Einfluss eines Wirkzusammenhangs hinausgeht.

Sowohl Daten, die auf Basis von Expertenwissen gesammelt werden, als auch Daten, die durch automatische Datensammlung ermittelt werden, erfordern eine entsprechende Auswertung. Dazu zählen Methoden der deskriptiven Statistik sowie der schließenden Statistik (Kohn 2005), (Sachs und Hedderich 2006), (Fahrmeir et al. 2009). Für einen Einblick in die Auswertung qualitativer und quantitativer Daten sei außerdem auf die Literatur (Seaman 2008) und (Bortz und Döring 2006) verwiesen. Ergänzend zu den statistischen Methoden existieren verschiedene Werkzeuge im Bereich des Knowledge Discovery in Databases beziehungsweise dem Data Mining, welche für die (semi-) automatische Analyse von Daten und der Modellbildung aus Daten verwendet werden können (Ester und Sander 2000), (Witten et al. 2011).

Die Validität der gesammelten Daten spielt eine kritische Rolle bei der Kalibrierung der Wirkzusammenhänge als Verhaltensmodelle (Abschnitt 4.2.8) eines Entwicklungsprozesses. So liefern beispielsweise eine zu geringe Datenbasis oder falsche Daten ein Bild, welches nicht der Wirklichkeit entsprechen muss oder ein Bild, welches lediglich extremes Verhalten zeigt. Um die empirische Validität einer Untersuchung sicherzustellen, müssen bei der Sammlung und Analyse von Daten und Informationen über einen Entwicklungsprozess die folgenden Aspekte beachtet werden (Easterbrook et al. 2008):

- **Konstruktvalidität.** Ist das Konstrukt vollständig, präzise und richtig interpretierbar? Wird das richtige gemessen?
- **Interne Validität.** Sind das Untersuchungsdesign und die verwendeten Methoden angemessen und folgen die Ergebnisse tatsächlich aus den erhobenen Daten?
- **Externe Validität.** Sind die Ergebnisse auf andere Anwendungen des untersuchten Problems generalisierbar?
- **Reliabilität.** Sind die Ergebnisse wiederholbar beziehungsweise reproduzierbar und sind sie unabhängig vom Untersuchenden (Bias)?

Strategien, um die Validität der gesammelten und analysierten Daten sicherzustellen und zu erhöhen, sind nach (Creswell 2002):

- **Triangulation.** Für die Datenbasis sollen verschiedene Datenquellen mit verschiedenen Erhebungsmethoden verwendet werden. Beispiele sind Daten aus einer Befragung und aus einer Beobachtung.

- **Member-checking.** Den Befragten beziehungsweise den Betroffenen werden die Ergebnisse der gesammelten und aufbereiteten Daten vorgestellt. Die Korrektheit dieser Ergebnisse wird erfragt und wenn notwendig werden im Anschluss Korrekturen vorgenommen.
- **Rich, thick description.** Die ausführliche Beschreibung des Vorgehens und die logische Ableitung von Ergebnissen bildet die Grundlage für weitere Analysen.
- **Clarify the bias.** Das Problem der Verzerrung von Daten, insbesondere die Selbstreflektion, muss beachtet werden.
- **Present negative or discrepant information.** Es sind auch Ergebnisse darzustellen, die nicht den Erwartungen entsprechen.
- **Spend prolonged time in field.** Die Auseinandersetzung mit dem Feld und das Verbringen längerer Zeit im Feld der Untersuchung unterstützen die Schaffung eines umfassenden Verständnisses.
- **Use peer debriefing.** Die Ergebnisse sollen kritisch begutachtet und mit einem Experten diskutiert werden.
- **Use an external auditor.** Ein externer Experte soll das Vorgehen und die Ergebnisse der Untersuchung begutachten.

3.7.1 Methoden

Bei der Erfassung von Daten bewährt es sich, zunächst mit offenen Methoden zu beginnen um das Problem der Einflussanalyse und den Kontext der Untersuchung zu erfassen, ohne den möglichen Raum der Untersuchung zu sehr einzuschränken. Ist das Problem der Untersuchung erfasst und klar, welche Art von Daten gesammelt werden sollen, eignen sich (teil-) strukturierte Methoden, die ein gezieltes Gerüst für die Fragestellung vorgeben, den Ergebnisraum aber noch nicht voll einschränken. Schließlich eignen sich strukturierte Methoden etwa der automatischen Datenerfassung, wenn empirische Ergebnisse die gezielten Fragestellungen belegen oder widerlegen sollen.

In Anhang A ist der im Rahmen dieser Arbeit entwickelte Leitfaden für die Erhebung von Entwicklungsprozessen zu finden. Dieser dient dabei als Grundlage etwa für die Durchführung von Interviews und Fokusgruppen oder auch zur Entwicklung anwendungsspezifischer Fragebögen.

Die folgende Tabelle 3.5 gibt einen Überblick über verbreitete Methoden zur Datensammlung. Der Überblick dient der grundsätzlichen Einordnung verschiedener Methoden. Die bei einer Untersuchung von Entwicklungsprozessen zur Verfügung stehenden Mittel und Fragestellungen sind anwendungsspezifisch, so dass entsprechend des Problems der Einflussanalyse geeignete Methoden im Kosten-Nutzen-Verhältnis ausgewählt werden müssen. Die aufgeführten Methoden sind nach ihrer Offenheit bezüglich

der zu untersuchenden Fragestellungen sortiert. Zunächst werden eher offene Methoden, dann eher strukturierte Methoden genannt.

Für einen tieferen Einblick zur Anwendung bestimmter Methoden sei auf die einschlägige Literatur verwiesen:

- Methoden mit Anwendung in Fallstudien (Yin 1994),
- Überblick über Forschungsmethoden (Creswell 2002) und (Bryman 2008),
- Methoden mit Anwendung im Software Engineering (Singer et al. 2008), (Seaman 2008) und (Rosenberg 2008) sowie
- Methoden bei der Evaluierung von Software Engineering Methoden und Werkzeugen (Kitchenham 1996a), und als zehnteilige Serie (Kitchenham 1996b)

Für die Analyse der erhobenen Daten werden im Wesentlichen zwei Arten unterschieden. Zum einen eignen sich qualitative Analysen dazu allgemeine Aussagen über bestimmte Fragestellungen in einem Anwendungskontext treffen zu können. Auf der anderen Seite wird die statistische Analyse verwendet, um eine quantitative Bewertung der gesammelten Daten vorzunehmen. So schließen sich die beiden Arten der Analyse nicht gegenseitig aus, sondern sind vielmehr in Kombination zu verwenden.

Tabelle 3.5 Methoden zur Sammlung und Analyse von Daten

Methoden	Zweck / Vorteile / Nachteile	zusätzliche Quellen
unstrukturiertes Interview → qualitativ	Grobe Einordnung von Inhalten + guter Einstieg, um Themen / Probleme nicht einzuschränken + relativ schnell einsetzbar für Einstieg – Ergebnis kann am Ziel vorbeigehen – keine verallgemeinernde Aussagen möglich (subjektiv)	(Batinic und Appel 2008) (Salant und Dillman 1994)
teil-strukturiertes Interview → qualitativ / statistisch	Strukturierte Einordnung von Inhalten + erlaubt die Interaktion mit dem Befragten im Gegensatz zu strukturierten Interviews + liefert relativ gut vergleichbare Aussagen durch Interviewleitfaden – nicht sinnvoll, wenn eine Auswahl an Antworten gegeben werden kann	(Faulkner 2000) (Salant und Dillman 1994)
strukturiertes Interview vergleiche Fragebogen → statistisch	Gezielte Abfrage von Informationen + gute, ökonomische Durchführung und Auswertbarkeit + gut vergleichbare Aussagen – Eingrenzung der gewonnenen Informationen durch zu enge Antwortmöglichkeiten	(Salant und Dillman 1994)
Fragebogen → statistisch	Gezielte schriftliche Abfrage von Informationen + kein Interviewer notwendig, sehr ökonomisch + große Stichproben möglich + Anonymität, führt zu mehr Beteiligungsbereitschaft – geringe Rücklaufquote möglich – unvollständige Antworten – keine Reaktion auf Verständnisprobleme möglich	(Pfleeger und Kitchenham 2001)
Brainstorming → qualitativ	Liefert eine Menge von Ideen zu einem Themenbereich + guter Einstieg zur Generierung von Ideen + gibt einen breiten Überblick über mögliche und wichtige Themen und Probleme – professionelle Moderation notwendig, sonst eventuell nicht zielführend	(Singer et al. 2008)

Tabelle 3.5 Methoden zur Sammlung und Analyse von Daten (Fortsetzung 1)

Methode	Zweck / Vorteile / Nachteile	zusätzliche Quellen
Fokusgruppe 6 - 10 Teilnehmer → qualitativ	Gezielte, dynamische Diskussion liefert Erfahrungen, Meinungen und Ideen + liefert qualitativ hochwertige (unbekannte und implizite) Informationen + sehr effizient, bei geeigneter Vorbereitung – sehr genaue Planung und Mischung der Teilnehmer notwendig – professionelle Moderation notwendig	(Morgan 1998) (Batinic und Appel 2008) (Kontio et al. 2008)
nicht-teilnehmende Beobachtung → qualitativ / statistisch	Liefert Aussagen über Nutzungsverhalten und Reaktionen während des Arbeitsprozesses (z.B. Werkzeuge) + keine Verzerrung durch Eingreifen des Beobachters + Verhalten mit Blick auf Usability erhebbar – nur sinnvoll bei tatsächlich beobachtbarem Verhalten	(Seaman 2008)
teilnehmende Beobachtung → qualitativ / statistisch	Liefert einen detaillierten Einblick in die tatsächliche Arbeit + Hilft beim Verstehen von Teilproblemen und Zusammenhängen – erfordert sehr hohen Aufwand	(Seaman 2008)
Conceptual Modeling → qualitativ	Liefert einen subjektive Überblick über die physikalische Umwelt des Befragten (z.B. verwendete Werkzeuge, Kommunikation, Vorgehen im Detail) + das mentale Modell eines Befragten wird akkurat expliziert – die Interpretation des Modells kann problematisch sein – mehrere Befragte liefern unterschiedliche Abstraktionen und Modelle	(Singer et al. 2008)
Arbeitstagebuch → qualitativ / statistisch	Liefert eine Übersicht von Ereignissen deren Folge und Dauer (z.B. von durchgeführten Aufgaben) + aktuelle und relativ genaue Dokumentation der Arbeit im Vergleich zu retrospektiven Interviews + relativ geringer Erhebungsaufwand für den Untersuchenden – nicht alle Ereignisse werden überhaupt oder fehlerhaft notiert	(Singer et al. 2008)
Think-Aloud Protokolle → qualitativ	Liefert detaillierte Einblicke in die tatsächliche Arbeit + liefert zeitlich direkte Einblicke in das Vorgehen bei der Arbeit – Auswertung der gesammelten Daten kann sehr aufwändig sein	(Singer et al. 2008)

Tabelle 3.5 Methoden zur Sammlung und Analyse von Daten (Fortsetzung 2)

Methoden	Zweck / Vorteile / Nachteile	zusätzliche Quellen
Instrumenting Systems System Monitoring → statistisch	<p>Automatische Erhebung von Prozess- und Produktdaten</p> <ul style="list-style-type: none"> + sehr ökonomisch, da kein Eingreifen erforderlich ist + liefert genaue Aussagen für tatsächlich durchgeführte Aufgaben, deren Folge und Dauer, sowie verwendete Werkzeuge - Analyse protokollierter Ereignisse kann problematisch sein, da der Grund für Ereignisse unbekannt ist - rechtliche und ethische Probleme müssen beachtet werden - Aufwand für die Implementierung des Tracking Systems 	(Singer et al. 2008)
Dokumentationsanalyse → qualitativ	<p>Analyse der Prozess- und Produktstruktur</p> <ul style="list-style-type: none"> + stabile Basis für die Modellierung + wird nicht durch die Untersuchung beeinflusst - Dokumentation vom Autor beeinflusst - Auswahl begrenzt durch Selektivität, rechtliche Aspekte 	(Yin 1994)
Arbeitsdatenbank → qualitativ / statistisch	<p>Analyse von Arbeitstagebüchern, Issue und Bug Tracking Protokollen</p> <ul style="list-style-type: none"> + viele Daten aus unterschiedlichen System können schnell zur Verfügung stehen + Daten werden nicht durch die Untersuchung beeinflusst - Qualität und Quantität der Daten wird stark durch protokollierenden Entwickler subjektiv beeinflusst 	(Singer et al. 2008)
Werkzeug Protokoll Analyse → statistisch	<p>Sammlung und Analyse von existierenden Protokollen / Logs</p> <ul style="list-style-type: none"> + sehr ökonomisch, da kein Eingreifen erforderlich ist + liefert genaue Aussagen über durchgeführte Aufgaben - Protokolle decken möglicherweise interessante Informationen nicht ab 	(Singer et al. 2008)

3.8 Quantifizierung von Wirkzusammenhängen

In den vorigen beiden Schritten wurden beeinflussenden Eigenschaften von Prozesselementen und grundsätzliche Wirkzusammenhänge identifiziert und dokumentiert. Zudem wurden Daten über den Entwicklungsprozess und damit über Wirkzusammenhänge gesammelt und analysiert. Auf dieser Basis werden nun die identifizierten Wirkzusammenhänge konkretisiert und anhand der gesammelten Daten überprüft. Somit werden quantitative Modelle abgeleitet, die das Verhalten des untersuchten Entwicklungsprozesses beschreiben. Für erste Iterationen der Methodik sind hier durchaus abstraktere Modelle auf Basis von Annahmen akzeptabel. Mit weiteren Iterationen und dem damit verbundenen Aufbau von Wissen wird dann eine spezialisiertere und fundiertere Modellierung von Wirkzusammenhängen angestrebt.

Die deskriptive Analyse der gesammelten Daten erlaubt die Eingrenzung der Wertebereiche untersuchter Eigenschaften von Prozesselementen. Ein solcher Wertebereich ergibt sich beispielsweise aus der Dauer einer Aufgabe. So ist für die Durchführung einer Aufgabe mit einer minimalen Dauer und einer maximalen Dauer zu rechnen. Zwischen diesen Extremen liegen allerdings beliebig viele Ausprägungen, die durch diese Extremwerte und beispielsweise den Mittelwert nicht adäquat abgedeckt werden. So stellt eine empirische Verteilung die wahren Ausprägungen der Dauer einer Aufgabe genauer dar. Dieses gilt in ähnlicher Weise auch für Eigenschaften von anderen Prozesselementen wie beispielsweise die Erfahrung eines Entwicklers oder die Qualität eines Design Artefakts.

Also werden hier Wahrscheinlichkeitsverteilungen verwendet, welche die Abbildung von entsprechendem empirischem Prozessverhalten erlauben. Dabei wird ein Wirkzusammenhang als parametrisierte Wahrscheinlichkeitsverteilung betrachtet. Ein Beispiel dafür wird im folgenden Abschnitt 3.8.1 gegeben.

3.8.1 Anwendung

Abbildung 3.8 veranschaulicht exemplarisch die Quantifizierung eines Wirkzusammenhangs mit einem Einfluss auf die funktionale Korrektheit des VHDL Artefakts. Auf Basis der identifizierten Wirkzusammenhänge und gesammelten Daten sowie der Analyse gesammelter Daten soll hier angenommen werden, dass die Wahrscheinlichkeit ein funktional korrektes VHDL Artefakt zu erhalten normalverteilt ist. Mit dem Fortschritt im Entwicklungsprozess wird die funktionale Korrektheit mit jeder Iteration der Aufgabe RTL Design im Wesentlichen erhöht. Es sind allerdings auch Rückschläge möglich, die eben durch eine solche Wahrscheinlichkeitsverteilung berücksichtigt werden. Mit steigender funktionaler Korrektheit ist außerdem anzunehmen, dass die

Abweichung und Varianz der tatsächlichen funktionalen Korrektheit vom Mittelwert sinkt und die Verteilung dadurch schmäler wird. Dementsprechend nähern sich auch die Verteilungen der vollen funktionalen Korrektheit an.

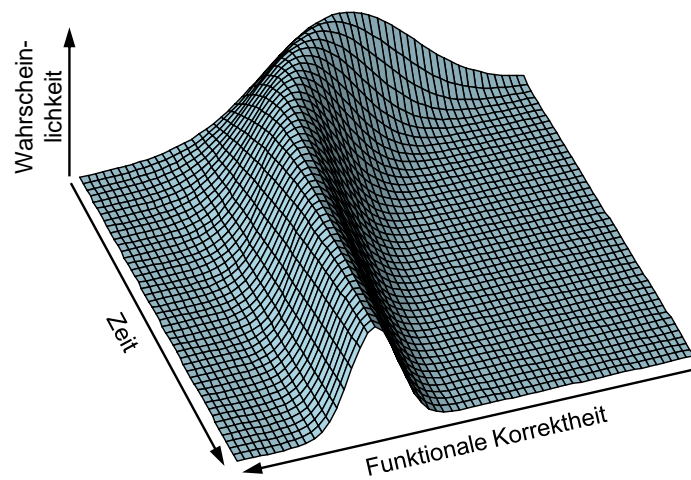


Abbildung 3.8 Exemplarische Entwicklung funktionaler Korrektheit

Die Form der Wahrscheinlichkeitsverteilungen basiert fundamental auf beeinflussenden Eigenschaften. So ist die funktionale Korrektheit des VHDL bei der manuellen Transformation vom C/C++ Artefakt in ein VHDL Artefakt abhängig von der Erfahrung des entsprechenden Entwicklers. Solche Einflüsse zeigen sich in Form einer Wahrscheinlichkeitsverteilung sowie in der Varianz der jeweiligen Verteilung. Beispielsweise kann angenommen werden, dass erfahrene Entwickler bereits initial eine bessere Verteilung bestimmen.

3.8.2 Methoden

Wirkzusammenhängen werden auf Basis der im vorherigen Schritt gesammelten Daten quantifiziert. Grundsätzlich sind dazu zunächst die Methoden der deskriptiven Statistik zu verwenden, um einen Überblick über die gesammelten Daten zu erhalten. Für die Ableitung von Zusammenhängen zwischen verschiedenen Einflussfaktoren eignen sich Methoden der schließenden Statistik wie Korrelationen und Regressionen. Bei der Modellbildung ist unbedingt auf die Validität der Daten und der abgeleiteten Zusammenhänge zu achten, vergleiche Abschnitt 3.7. Auch wenn statistische Methoden wie Korrelationen auf einem Zusammenhang hinweisen ist dieser doch kritisch zu validieren, da ein gemeinsames Auftreten zweier Phänomene keine Beziehung unterstellen kann (Raffo 1993). Ein beliebtes Beispiel für einen solchen Scheinzusammenhang ist die Beziehung zwischen dem Geburtenrückgang und dem Rückgang der heimischen Störche. So ist wohl nicht anzunehmen, dass Störche Neugeborene bringen, noch, dass Neugeborene Storchenküken bringen (Höfer et al. 2004).

3.9 Validierung des Modells

Die Aussagen, die sich aus einem Modell über die reale Welt ableiten lassen, sind nur so gut, wie das Modell selbst und wie die darin enthaltenen Annahmen das System beziehungsweise den Entwicklungsprozess der realen Welt wiedergeben. Es ist daher unverzichtbar das durch die vorherigen Schritte entwickelte Modell des Entwicklungsprozesses zu validieren. Dabei kann und sollte die Validierung auch im Anschluss an jeden vorgestellten Schritt erfolgen, um die entsprechenden Ergebnisse eines Schrittes zu überprüfen und damit das Modell inkrementell zu verbessern. Dazu lässt sich eine Reihe von Fragen stellen, die im Rahmen der Validierung untersucht beziehungsweise beantwortet werden müssen. Diese sind nicht abschließend:

- Entspricht das Modell des Entwicklungsprozesses der Wirklichkeit in einer angemessenen Abstraktion?
- Sind alle wesentlichen Prozesselemente modelliert?
- Sind alle kritischen Eigenschaften von Prozesselementen, die andere Eigenschaften und den Fortschritt des Entwicklungsprozesses bestimmen abgebildet?
- Unterstützt die Datensammlung und Datenanalyse die Annahmen über identifizierte Wirkzusammenhänge beziehungsweise liefert die Datenanalyse Hinweise auf weitere oder andere Wirkzusammenhänge?
- Bilden die quantifizierten Wirkzusammenhänge die gesammelten Daten und die angenommenen Zusammenhänge ab?

Nach Law und Kelton eignen sich insbesondere die im Folgenden erläuterten Techniken, um die Validität eines Simulationsmodells sicherzustellen (Law und Kelton 1991):

- Umfangreiche Modelle sollten durch mehrere Personen entwickelt werden, welche die Ergebnisse der Modellierung gegeneinander prüfen. Dazu eignen sich beispielsweise auch strukturierte Walk-Throughs (Abnahmen).
- Wenn möglich, sollte das Modell zunächst unter vereinfachten Annahmen simuliert werden, für welche die Ergebnisse der Simulation leicht nachvollziehbar sind.
- Die Schritt-für-Schritt-Simulation ermöglicht die Beobachtung von aktuellen Werten und die Entwicklung dieser Werte über die Zeit. Dieses Verhalten kann dann auf Plausibilität geprüft werden.
- Das Modell sollte mehrfach mit unterschiedlichen Belegungen von Eigenschaftswerten durchlaufen werden und die Ergebnisse der Simulation mit realen Daten beziehungsweise Erfahrungen von Experten abgeglichen werden.
- Bei einem Vergleich von simulierten Ergebnissen gegen existierende Daten beziehungsweise Erfahrungen ist die Differenz der Ergebnisse ein Indikator für die Validität des Modells.

Ein wesentlicher Punkt bei der Validierung des Modells ist die Gegenüberstellung der durch Simulation erstellten Ergebnisse und Beobachtungen in der realen Welt. So sind die Ergebnisse der Simulation gegen existierende Daten des Entwicklungsprozesses, wie etwa Aufwände oder die Erfahrung von Experten, zu prüfen. Liegen ausreichend quantitative Daten vor, so lässt sich die Validität des Modells durch einen statistischen Vergleich dieser Daten bestimmen. Insbesondere Methoden der Statistik wie die Varianz und das Konfidenzintervall erlauben eine Einschätzung der Validität auf Basis vorhandener Daten. Vergleiche dazu auch Abschnitt 3.13.

Trotz der genannten Methoden ist die Güte eines Modells von der subjektiven und anwendungsspezifischen Einschätzung von entsprechenden Experten des Entwicklungsprozesses und dem mit dem Modell verbundenen Zweck abhängig (Law und Kelton 1991), (Banks 1998).

3.10 Beschreibung der neuen Methode

Als Ergebnis der bisherigen Schritte liegt ein validiertes Modell des existierenden Entwicklungsprozesses vor, welches auf einer wiederverwendbaren Beschreibung von Methoden basiert. Dieses Modell bildet die Grundlage für die Beschreibung der neuen Methode beziehungsweise der Prozessänderung. So sind in diesem Schritt die Änderungen beziehungsweise Auswirkungen einer neuen Methode auf den existierenden Entwicklungsprozess zu untersuchen.

Dazu lassen sich verschiedene Aspekte identifizieren, die bei der Modellierung beachtet werden müssen. Die folgenden Fragen sollen zur Orientierung dienen:

- Welche Aufgaben werden neu in den Entwicklungsprozess eingeführt?
- Welche Aufgaben werden überflüssig?
- Welche Aufgaben werden durch die Einführung der neuen Methode verändert?
- Sind neue oder andere Artefakte als Eingabe für neue Aufgaben notwendig?
- Wie verändern sich die Eigenschaften von Prozesselementen durch die Einführung der neuen Methode?
- Welche neuen oder anderen Wirkzusammenhänge gibt es durch die Einführung der neuen Methode?
- Wie verändert sich der Aufwand bisheriger Aufgaben durch die neue Methode?

Mit den genannten Fragen als Basis, wird die neue Methode analog zum existierenden Entwicklungsprozess beschrieben. Zunächst werden also die neuen Prozesselemente definiert (Abschnitt 3.5), dann neue beziehungsweise sich ändernde Wirkzusammenhänge identifiziert (Abschnitt 3.6) sowie empirische Daten gesammelt, analysiert (Abschnitt 3.7) und die entsprechenden Wirkzusammenhänge quantifiziert (Abschnitt

3.8). Abschließend gelten auch für die Validierung des Modells der neuen Methode die in Abschnitt 3.9 aufgeführten Punkte.

Um den Einfluss einer neuen Methode auf einen existierenden Entwicklungsprozesses zu bestimmen, lässt sich der Einfluss der Prozessänderung in Form von Annahmen modellieren, die als geänderte Wirkzusammenhänge modelliert werden. Als Grundlage dazu können die Methoden zur Sammlung und Analyse von Daten (siehe Tabelle 3.5) verwendet werden, um die Art und Größe des Einflusses gezielt abzuschätzen.

Alternativ und fundierter zur Modellierung von Annahmen können Experimente und (kleine) Pilotprojekte durchgeführt werden. Entsprechende Experimente oder Pilotprojekte sollen sich dabei auf den Einfluss der neuen Methode konzentrieren. Bei Experimenten oder entsprechenden Pilotprojekten zur Einordnung des Verhaltens der neuen Methode ist ein zielorientierter Ansatz zu verfolgen. Wie die obigen Fragen schon zeigen, sind grundlegende Annahmen zur neuen Methode zu untersuchen. Im Wesentlichen interessieren also Einflüsse auf Eigenschaften von Prozesselementen und Einflüsse auf den Aufwand assoziierter Aufgaben. Da Pilotprojekte oder Experimente nur einen bestimmten Ausschnitt also ein oder wenige Szenarien möglicher Einflüsse abdecken, ist zu prüfen, in wie weit qualitative Wirkzusammenhänge „exakt“ quantifiziert werden können. Bleibt eine Unsicherheit über den Einfluss bestehen, ist dieser mit entsprechenden Wahrscheinlichkeitsverteilungen und einer entsprechenden Varianz des möglichen Einflusses zu modellieren. Vergleiche dazu auch (Münch et al. 2005) und (Basili 2007).

Sowohl für Ergebnisse auf Basis von Experimenten oder Pilotprojekten, als auch für quantifizierte Wirkzusammenhänge auf Basis anderer Methoden, wie beispielsweise Interviews mit Experten, eignen sich insbesondere Sensitivitätsanalysen, um während der Einflussanalyse „was wäre wenn“ Fragen zu beantworten. Mit Hilfe von Sensitivitätsanalysen kann somit auch der Einfluss unsicherer beziehungsweise variierender Parameter oder Eigenschaften abgeschätzt werden.

Im folgenden Schritt wird die Beschreibung der neuen Methode virtuell in die Beschreibung des existierenden Entwicklungsprozesses eingebracht.

3.11 Virtuelle Integration der neuen Methode

Die virtuelle Integration der neuen Methode in den existierenden Entwicklungsprozess umfasst die Beschreibung eines alternativen Entwicklungsprozesses auf Basis der für den existierenden Entwicklungsprozess definierten Prozesselemente und Wirkzusammenhänge. Dabei wird also der alternative Entwicklungsprozess gegenüber dem existierenden Entwicklungsprozess modelliert.

3.11.1 Anwendung

Die neue Methode beschreibt die automatische Transformation eines SystemC Design Artefakts in ein entsprechendes VHDL Artefakt mit Hilfe der High Level Synthesis, vergleiche Abschnitt 1.5. Abbildung 3.9 zeigt das Modell des alternativen Entwicklungsprozesses mit seinen Wirkzusammenhängen. Vergleiche dazu den in Abbildung 3.7 auf Seite 91 illustrieren existierenden Entwicklungsprozess.

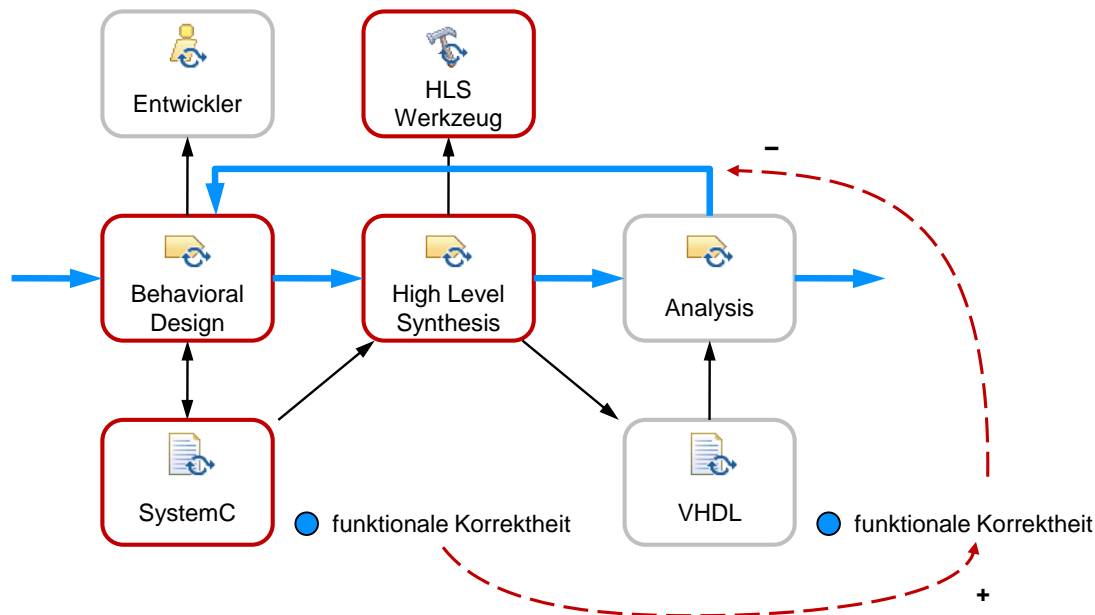


Abbildung 3.9 Alternativer Entwicklungsprozess mit seinen Wirkzusammenhängen

Der alternative Entwicklungsprozess verfügt in diesem Beispiel weiterhin über drei Aufgaben und zwei Design Artefakte, die für die Durchführung einer Aufgabe notwendig sind oder durch eine Aufgabe erstellt beziehungsweise bearbeitet werden. Die Änderungen im alternativen Entwicklungsprozess gegenüber dem existierenden Entwicklungsprozess sind in der Abbildung 3.9 rot markiert. So erstellt oder bearbeitet die Aufgabe *Behavioral Design* nun ein *SystemC* Artefakt im Gegensatz zu einem C/C++ Artefakt. Die Verhaltensbeschreibung in *SystemC* erlaubt dabei durch bestimmte Sprachkonstrukte eine automatische Transformation in ein *VHDL* Artefakt. Diese Transformation übernimmt die weitestgehend automatisierte Aufgabe der *High Level Synthesis* unter Verwendung eines entsprechenden Werkzeugs. Das Ergebnis dieser Aufgabe ist ein *VHDL* Artefakt, dessen Eigenschaft *funktionale Korrektheit* wie im existierenden Entwicklungsprozess auch gegen ein bestimmtes Ziel geprüft wird. Wird dieses Ziel nicht erfüllt, so sind nun ein Rückschritt zur Verhaltensbeschreibung im *Behavioral Design* und damit eine erneute Synthese und Analyse notwendig.

Mit der Beschreibung des alternativen Entwicklungsprozesses gehen für das Beispiel die folgenden Annahmen einher:

- Die neue Methode der High Level Synthesis erfordert in *Behavioral Design* einen größeren Aufwand, da nun weitere Informationen für die automatische Synthese modelliert werden müssen.
- Die Wahrscheinlichkeit funktional korrekten VHDL Code zu erhalten, steht in direkter Beziehung zu der funktionalen Korrektheit des SystemC Artefakts, so sichert der Werkzeughersteller beispielsweise zu, aus funktional korrektem SystemC funktional korrektes VHDL zu erzeugen.
- Das HLS Werkzeug hat bestimmte Anschaffungs- beziehungsweise Lizenzkosten.

Aus den beschriebenen Annahmen lassen sich verschiedene Szenarien für den möglichen Einfluss der neuen Methode auf den Entwicklungsprozess ableiten. Für das Beispiel seien zwei Szenarien vorgestellt:

- Der zusätzliche Aufwand in Behavioral Design und die für das neue Werkzeug anfallenden Kosten sind höher als der Aufwand, der durch die automatischen Transformation des SystemC in VHDL eingespart wird.
- Der automatisch generierte VHDL Code ist funktional korrekt und bedarf später keiner manuellen Überarbeitung mehr. Zudem ist der Aufwand des alternativen Entwicklungsprozesses signifikant kleiner als der aktuelle Aufwand.

Die beiden genannten Szenarien sind denkbar und sollen durch die Simulation der Entwicklungsprozesse bestätigt oder widerlegt werden.

3.12 Simulation der Entwicklungsprozesse

In diesem Schritt wird der existierende Entwicklungsprozess gegen den geänderten alternativen Entwicklungsprozess simuliert. Bei der Simulation wird das mögliche Verhalten eines Entwicklungsprozesses durch mehrere Simulationsläufe exploriert und gegen die in Abschnitt 3.4 definierten Ziele der Einflussanalyse untersucht.

Grundsätzlich können hierbei zwei Arten von Simulation unterschieden werden: Schritt-für-Schritt Simulation und mehrfache Simulationsläufe.

Die Schritt-für-Schritt Simulation eignet sich insbesondere für die Untersuchung und Validierung des Prozessverhaltens im Detail, da während einer solchen Simulation die Änderung von Eigenschaftswerten von Prozesselementen wie auch die Veränderung des Aufwands und der Kosten von Aufgaben über die Zeit beobachtet werden können. Allerdings ist diese Art der Simulation nicht für statistische Schlüsse über das generelle Prozessverhalten geeignet, da lediglich ein Simulationslauf und damit ein mögliches Szenario von vielen, sich aus den probabilistischen Wirkzusammenhängen ergebenden Prozessverhalten, untersucht werden kann.

Auf der anderen Seite ermöglichen mehrfache Simulationsläufe der Entwicklungsprozesse die Anwendung von statistischen Methoden zur Auswertung des möglichen Prozessverhaltens in verschiedenen Szenarien. Mit Bezug auf die Ziele der Einflussanalyse, die in Abschnitt 3.4 definiert wurden, erlauben die Methoden der deskriptiven Statistik etwa die Bewertung von Minimum, Maximum, Mittel, Median, Quantile und Varianz. Histogramme, Diagrammen der Verteilung und Box Plots helfen bei der Interpretation der statistischen Werte.

Wie bereits in Abschnitt 3.9 angesprochen, sind die Ergebnisse der Simulation und die zugrundeliegenden Annahmen über die Entwicklungsprozesse zu validieren. So weisen Ergebnisse, welche auf Basis von vielen Annahmen ermittelt wurden, eine höhere Unsicherheit auf, als Ergebnisse, die auf Basis von quantitativen anwendungsspezifischen Daten ermittelt wurden. Diese Unsicherheit spiegelt sich in der Varianz der möglichen Ergebnisse wieder.

Im Folgenden Abschnitt werden Methoden für die Bewertung des Einflusses einer neuen Methode auf einen Entwicklungsprozess vorgestellt.

3.13 Bewertung des Einflusses der neuen Methode

Die Einflussbewertung einer Methodenänderung auf einen Entwicklungsprozess wird durch verschiedene Methoden unterstützt, die in diesem Abschnitt betrachtet werden. Im Allgemeinen liefern die vorgestellten Methoden ein Bild über mögliches und extremes Verhalten von Prozessszenarien auf Basis statistischer Methoden. Die Bewertung des Einflusses einer neuen Methode beziehungsweise einer Prozessänderung wird durch den Vergleich der charakteristischen Bilder der Entwicklungsprozesse vorgenommen. Zu diesen Einflussanalysen zählen

- der Vergleich statistischer Maße,
- der Vergleich von Konfidenzintervallen,
- der Vergleich von Abdeckungsintervallen (Quantil-Intervallen),
- Hypothesentest zur Bewertung der Sicherheit von Differenzen sowie
- die Betrachtung der strategischen Bedeutung des Einflusses.

Die Vorstellung der genannten Methoden erfolgt im Folgenden am vereinfachten Beispiel normalverteilter Ergebnisse zweier zu vergleichender Entwicklungsprozesse *IST* und *SOLL*. Dabei beschreibt *IST* den existierenden Entwicklungsprozess und *SOLL* den Entwicklungsprozess mit der Methodenänderung. Die dazu analogen Verteilungen beziehungsweise Stichproben *I* und *S* beziehen sich hier auf die benötigten Aufwände der jeweiligen Entwicklungsprozesse. Eine Bewertung der Kosten oder der Produktqualität kann sinngemäß dazu erfolgen. Entsprechend der intendierten Zielrichtung sind Aufwand und Kosten zu minimieren und Produktqualität zu maximieren.

3.13.1 Vergleich statistischer Maße

Abbildung 3.10 zeigt zwei Verteilungen für die beiden Entwicklungsprozesse *IST* und *SOLL*. Auf der Abszisse werden die zu untersuchenden Werte der simulierten Entwicklungsprozesse, wie beispielsweise der Aufwand der Entwicklungsprozesse, aufgetragen. Die Ordinate gibt dazu die Wahrscheinlichkeit an, einen bestimmten Wert zu erreichen.

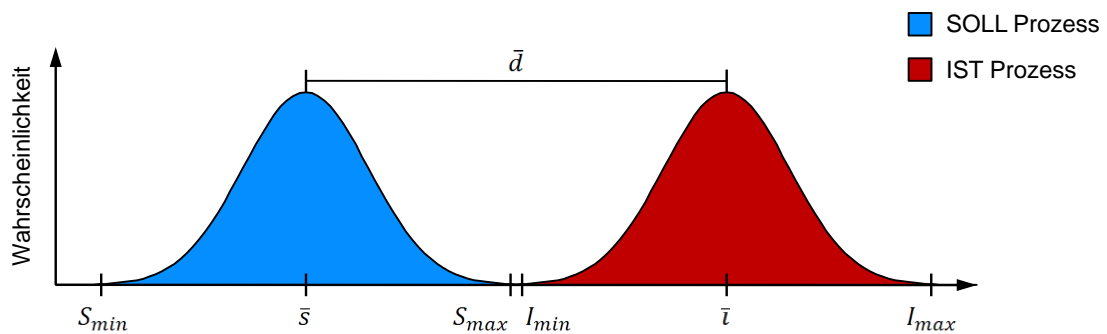


Abbildung 3.10 Vergleich zweier Entwicklungsprozesse

Durch direkten Vergleich der Statistiken der simulierten Stichproben *I* und *S* der beiden Entwicklungsprozesse *IST* und *SOLL* lässt sich die folgende Aussage ableiten:

- *SOLL* ist absolut vorteilhaft gegenüber *IST*, wenn $S_{max} < I_{min}$ ist.

Bei diesem einfachen Vergleich ist allerdings zu beachten, dass die Anzahl der durchgeführten Simulationsläufe einen signifikanten Einfluss auf die Verteilungen hat. So werden durch Simulation nicht alle möglichen Szenarien abgedeckt. Insbesondere werden extreme Fälle wie Minimum und Maximum selten erreicht. Was eine sinnvolle Anzahl an Simulationen ist, richtet sich nach dem Anwendungskontext und den untersuchten Fragestellungen und ist daher bei der praktischen Anwendung festzulegen, vergleiche (Law und Kelton 1991). Für diese Arbeit wird eine Anzahl von 1000 Simulationsläufen als Richtwert definiert.

Für die weitere Betrachtung der Entwicklungsprozesse interessiert die Differenz D der beiden Stichproben (Simulationsergebnisse) *I* und *S*, welche den Unterschied zwischen den beiden Entwicklungsprozessen *IST* und *SOLL* beschreibt (Law und Kelton 1991). Dabei wird im Folgenden davon ausgegangen, dass Entwicklungsprozess *SOLL* vor dem Entwicklungsprozess *IST* liegt.

$$D = I - S \quad \text{beziehungsweise} \quad \bar{d} = \bar{i} - \bar{s}$$

Dabei stehen \bar{i} und \bar{s} für die Mittelwerte der entsprechenden Stichproben *I* und *S*. Die mittlere Differenz der beiden Stichproben wird mit \bar{d} bezeichnet. Der Mittelwert für eine Stichprobe X der Länge n und den Beobachtungen x_i wird wie folgt berechnet:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Aus der mittleren Differenz können die folgenden Aussagen abgeleitet werden:

- *SOLL* ist vorteilhaft gegenüber *IST*, wenn $\bar{s} < \bar{i}$ ist und damit $\bar{d} > 0$ gilt.
- Je größer die mittlere Differenz \bar{d} ist, desto größer der Vorteil von *SOLL* gegenüber *IST*.
- Im besten Fall liegt der Vorteil von *SOLL* gegenüber *IST* bei D_{max} .
- Im schlechtesten Fall liegt der Vorteil von *SOLL* gegenüber *IST* bei D_{min} .

Für Vergleiche über deskriptive Statistiken ist zu bemerken, dass die Varianz der simulierten Stichproben ein Indikator für die Güte der Modellierung der Entwicklungsprozesse beziehungsweise für das Risiko der Aussagen ist – vorausgesetzt, diese Unsicherheit wurde durch entsprechende Wahrscheinlichkeitsverteilungen modelliert. Je geringer die Varianz der betrachteten Stichproben ist, desto sicherer sind die Aussagen, die sich aus einem Vergleich ergeben. Dabei sind der Anwendungskontext und der Reifegrad der Entwicklungsprozesse mit einzubeziehen. So weisen unreife Entwicklungsprozesse schon in der Wirklichkeit eine höhere Varianz auf, als reifere Entwicklungsprozesse. Die Varianz ohne systematischen Fehler für eine Stichprobe X der Länge n , den Beobachtungen x_i und dem Mittelwert \bar{x} berechnet sich wie folgt:

$$Var(X) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

3.13.2 Vergleich von Konfidenzintervallen

Da sich mit unterschiedlichen Simulationsläufen auch die statistischen Maße wie der Mittelwert der Stichprobe verändern, wird das Konfidenzintervall eines entsprechenden Maßes betrachtet. Das Konfidenzintervall beschreibt eine obere und untere Grenze, zwischen denen der wahre Wert des Maßes mit einer bestimmten Fehlerwahrscheinlichkeit α zu finden ist. Damit ist ein Konfidenzintervall einer Punktschätzung vorzuziehen. Das Konfidenzintervall der mittleren Differenz einer Stichprobe D der Länge n ergibt sich dabei aus (Law und Kelton 1991):

$$[d_u; d_o] = \bar{d} \pm t_{n-1, 1-\alpha/2} \sqrt{\frac{Var(D)}{n}}$$

Dabei beschreibt t kritische Punkte der studentischen t -Verteilung für einen bestimmten Freiheitsgrad $n - 1$ und einer bestimmten Fehlerwahrscheinlichkeit α . Tabelle 3.6 zeigt häufig verwendete kritische Punkte der studentischen t -Verteilung für bis zu 10.000 Freiheitsgraden beziehungsweise der Normalverteilung für $n \rightarrow \infty$ nach dem Gesetz der

Großen Zahl. Je nach interessierender Fehlerwahrscheinlichkeit und der Anzahl der Simulationsläufe kann der entsprechende kritische Wert aus der Tabelle abgelesen und für die Berechnung des Konfidenzintervalls herangezogen werden.

Tabelle 3.6 Kritische Punkte der *t*- bzw. der Normalverteilung ($n \rightarrow \infty$)

n – 1	einseitiger Vertrauensbereich $1 - \alpha / 2$							
	0,750	0,875	0,900	0,950	0,975	0,990	0,995	0,999
	Wahrscheinlichkeit $1 - \alpha$							
	50%	75%	80%	90%	95%	98%	99%	99,8%
1	1,000	2,414	3,078	6,314	12,706	31,821	63,657	318,309
10	0,700	1,221	1,372	1,812	2,228	2,764	3,169	4,144
100	0,677	1,157	1,290	1,660	1,984	2,364	2,626	3,174
1000	0,675	1,151	1,282	1,646	1,962	2,330	2,581	3,098
10000	0,675	1,150	1,282	1,645	1,960	2,327	2,576	3,091
$n \rightarrow \infty$	0,674	1,150	1,282	1,645	1,960	2,326	2,576	3,090

Schließlich können die folgenden Aussagen festgehalten werden:

- *SOLL* ist vorteilhaft gegenüber *IST*, wenn für die simulierten Stichproben *I* und *S* $0 < d_u$ gilt.
- Je größer die untere Grenze des Intervalls d_u ist, desto größer der Vorteil von *SOLL* gegenüber *IST*.
- Je schmaler das Differenzintervall $[d_u; d_o]$ ($d_o - d_u$ strebt gegen „Null“) bei einer kleinen Fehlerwahrscheinlichkeit α ist, desto sicherer ist die Bewertung des Vorteils.
- Enthält das Konfidenzintervall den Wert 0, so kann keine Aussage über die Vorteilhaftigkeit getätigt werden.

3.13.3 Vergleich von Abdeckungsintervallen

Der Vergleich von Abdeckungsintervallen ermöglicht die weitere Untersuchung der Differenz zweier Entwicklungsprozesse. Dabei werden die Grenzen der Intervalle durch ausgewählte Quantile der Stichproben *I* und *S* festgelegt. Abbildung 3.11 zeigt zwei sich überschneidende Verteilungen und die Abdeckung zweier Intervalle, die jeweils $p_S = 80\%$ der simulierten Szenarien enthalten. Entsprechend beschreiben i_{10} und s_{10} die 10%-Quantile der Verteilungen und i_{90} beziehungsweise s_{90} die 90%-Quantile der Verteilungen. Dabei sind 10% der jeweiligen Beobachtungen links der 10%-Quantile und 90% der jeweiligen Beobachtung links von den 90%-Quantilen zu finden.

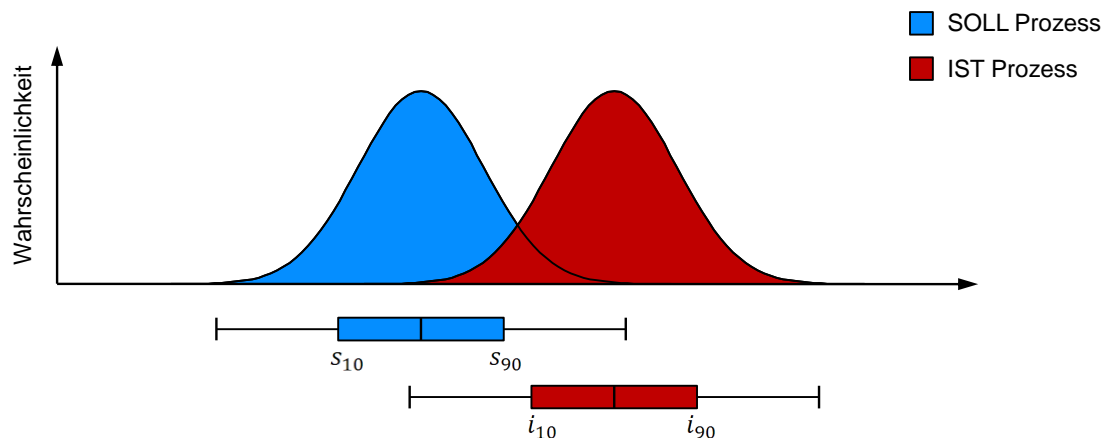


Abbildung 3.11 Vergleich von 80%-Abdeckungsintervallen für sich überschneidende Stichproben

Die folgenden Aussagen ergeben sich aus den Abdeckungsintervallen:

- *SOLL* ist in p_S der Szenarien absolut vorteilhaft gegenüber *IST*, wenn $s_{90} \leq i_{10}$ gilt.
- *SOLL* liegt in p_S der Szenarien mindestens $d = i_{10} - s_{90}$ Einheiten vor *IST*.

Überschneiden sich die Abdeckungsintervalle zweier Verteilungen, weil diese entweder eine geringe Differenz aufweisen oder die Abdeckungsintervalle sehr groß gewählt wurden, so ist eine Bewertung der Sicherheit beziehungsweise des Risikos der Aussagen von Interesse. Abbildung 3.12 zeigt eine entsprechende Überschneidung. Für 80%-Abdeckungsintervalle zweier Stichproben I und S wird definiert:

$$S_{80\%} = \{s \in S | s \geq s_{10} \wedge s \leq s_{90}\}$$

$$S_Z = \{s \in S | s \geq i_{10} \wedge s \leq s_{90}\}$$

$$p_Q = 1 - \frac{|S_Z|}{|S_{80\%}|}$$

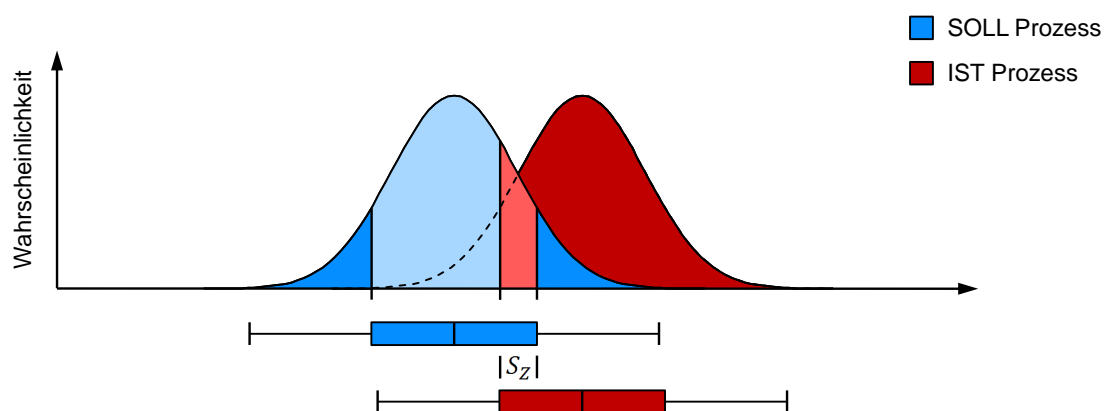


Abbildung 3.12 Bewertung von sich überschneidender Abdeckungsintervalle

Die Bewertung des Risikos von Aussagen wird wie folgt vorgenommen:

- *SOLL* ist in p_S der Szenarien mit einer Wahrscheinlichkeit von p_Q vorteilhaft gegenüber *IST*.

3.13.4 Bewertung der Sicherheit von Differenzen

Überschneiden sich zwei Verteilungen, so ist ein Maß hilfreich, welches den Grad der Überschneidung der Verteilungen als Indikator für die Sicherheit beziehungsweise des Risikos der getroffenen Aussagen beschreibt. Dazu eignet sich hier insbesondere die Unterscheidung von Fehlern 1. Art (α -Fehler) und 2. Art (β -Fehler) aus dem Bereich der Hypothesentests aus der Statistik. Tabelle 3.7 zeigt die grundsätzliche Bewertung für Hypothesen zu zwei alternativen Entwicklungsprozessen nach der Signalentdeckungstheorie. Vergleiche dazu (Velden 1982) und (Wickens und Hollands 1999).

Tabelle 3.7 Bewertung von Hypothesen zur Empfehlung für eine Methodenänderung bzw. Beibehaltung des aktuellen Entwicklungsprozesses

Empfehlung für	wahrer Sachverhalt	
	Keine Verbesserung (H_0)	Verbesserung (H_1)
Beibehaltung des aktuellen Entwicklungsprozesses	Korrekte Zurückweisung Wahrscheinlichkeit: $1 - \alpha$	Falsche Zurückweisung Wahrscheinlichkeit: β
Methodenänderung	Falsche Entscheidung Wahrscheinlichkeit: α	Korrekte Entscheidung Wahrscheinlichkeit: $1 - \beta$

Bei einem Hypothesentest zur Unterscheidung von Fehlern 1. Art und Fehlern 2. Art wird zunächst eine Nullhypothese (H_0) aufgestellt, die im Folgenden durch die Stichprobe I untersucht wird: „die Methodenänderung bringt keine Verbesserung“. Die Alternativhypothese (H_1) bildet das Gegenstück dazu: „die Methodenänderung bringt eine Verbesserung“.

Um zu einer Entscheidung für eine Methodenänderung beziehungsweise für die Beibehaltung des aktuellen Entwicklungsprozesses zu gelangen, ist es erforderlich entweder einen anwendungsspezifischen kritischen Wert k oder eine Fehlerwahrscheinlichkeit α zu definieren. Dabei gibt der kritische Wert an, ab wann sich für eine Methodenänderung oder gegen eine Methodenänderung entschieden wird, etwa in Aufwand oder Kosten. Der kritische Wert k lässt sich dabei in das α -Quantil der Verteilung I und umgekehrt überführen.

Abbildung 3.13 zeigt die beiden sich überschneidenden Stichproben I und S der Entwicklungsprozesse *IST* und *SOLL*.

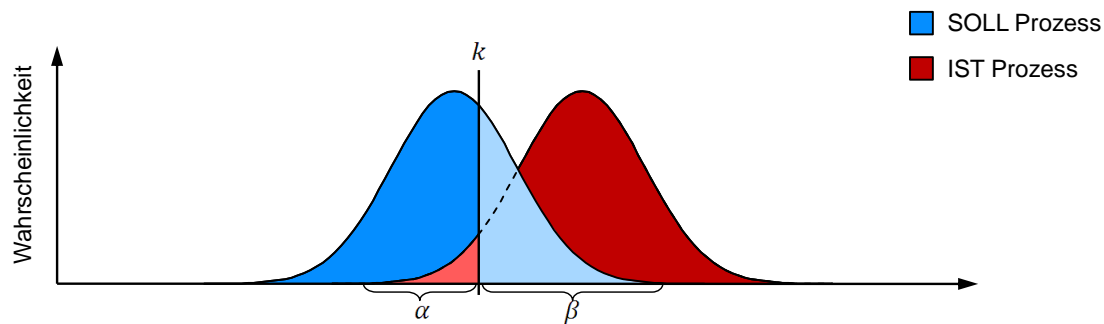


Abbildung 3.13 Bewertung von Hypothesen für oder gegen eine Methodenänderung

Alle Fälle links von $k = i_\alpha$ der Verteilung I (rechts) beschreiben die Fehlerwahrscheinlichkeit α (kleine Fläche über α), dass eine Methodenänderung empfohlen wird, obwohl diese keine Verbesserung erbringt (H_0). Alle Fälle rechts von $k = i_\alpha$ der Verteilung S (links) beschreiben die Fehlerwahrscheinlichkeit β (Fläche über β), dass der aktuelle Entwicklungsprozess beibehalten wird, obwohl die Methodenänderung eine Verbesserung hätte erbringen können.

Daraus lässt sich nun ableiten, mit welcher Wahrscheinlichkeit der Entwicklungsprozess *SOLL* besser als der Entwicklungsprozess *IST* ist.

$$k = i_\alpha$$

$$S_k = \{s \in S | s \leq k\}$$

$$p_S = 1 - \beta = \frac{|S_k|}{|S|}$$

Entsprechend können die folgenden Aussagen daraus abgeleitet werden:

- *SOLL* ist mit einer Wahrscheinlichkeit von p_S besser als *IST* bei einer Fehlerwahrscheinlichkeit von α .
- Wird die Fehlerwahrscheinlichkeit α reduziert, so sinkt auch die Wahrscheinlichkeit p_S für den Vorteil von *SOLL*.
- Je kleiner α und je größer p_S , desto sicherer ist der Vorteil von *SOLL* gegenüber *IST*.

Bei der Bewertung der Sicherheit über die Betrachtung der Fehler 1. Art und 2. Art ist anwendungsspezifisch zu untersuchen, ob sich die Verteilung wie dargestellt überschneiden. Besitzt etwa die Verteilung S eine hohe Varianz, so dass $S_{min} < I_{max}$ und $S_{max} > I_{max}$, könnte beispielsweise β überbewertet werden.

3.13.5 Strategische Bewertung

Aus den zuvor ermittelten Maßen zum Vergleich zweier Entwicklungsprozesse ist für eine strategische Bewertung nun die Differenz der betrachteten Entwicklungsprozesse

zu untersuchen. Bei der Wirtschaftlichkeitsrechnung werden Methoden eingesetzt, welche die Kosten einer Investition mit den einhergehenden Gewinnen vergleicht (Götze 2006). Dabei bezeichnet eine Investition die Kosten einer Methodenänderung und der Gewinn wird als Differenz zwischen den betrachteten Entwicklungsprozessen verstanden.

Da die Methodik offen bei der Modellierung des Aufwandes beziehungsweise der Kosten von Entwicklungsprozessen ist, können zwei Arten der Bewertung unterschieden werden. So wird entweder über den berechneten Aufwand der simulierten Entwicklungsprozesse oder direkt über die durch die Simulation berechneten Kosten bewertet. Dabei ist bei der Modellierung der alternativen Entwicklungsprozesse die explizite Beschreibung von anfallenden Kosten vorzuziehen, da nur so adäquat die Kosten einzelner Aufgaben und Aktivitäten berechnet werden können. Beispiele für Einflussfaktoren auf diese Kosten sind etwa unterschiedliche Stundenlöhne von Rollen oder die Verwendung von Werkzeugen und damit anfallende Lizenzkosten. Somit stellt die Berechnung der Kosten über den Aufwand der Entwicklungsprozesse eine weitere Approximation der Wirklichkeit dar. Ob während der Modellierung nur Aufwand oder auch Kosten abgebildet werden, hängt dabei klar von den Zielen der Einflussanalyse ab und ist im Anwendungsfall zu definieren.

Bewertung über den Aufwand

Um den Vorteil beziehungsweise Gewinn einer Investition über den Aufwand zu ermitteln, wird ein durchschnittlicher Faktor zur Beschreibung der Kosten pro Stunde eingeführt. Der Gewinn G eines Entwicklungsprozesses S gegenüber einem Entwicklungsprozess I ergibt sich demnach aus

$$G = (A_I - A_S) \cdot k_h$$

wobei A_I der Aufwand des existierenden Entwicklungsprozesses, A_S der Aufwand des Entwicklungsprozesses mit der Methodenänderung und k_h die durchschnittlichen Kosten pro Stunde beschreiben.

Bewertung über die Kosten

Für die Bewertung einer Investition über die Kosten wird als Gewinn direkt die entsprechende Differenz der Kosten der Entwicklungsprozesse verwendet.

$$G = K_I - K_S$$

Investitionskosten

Für eine Investition werden hier grundsätzlich zwei Arten von Kosten unterschieden. Zum einen Kosten, die durch den direkten Einsatz der neuen Methode beziehungsweise

der Methodenänderungen im Entwicklungsprozess zu beobachten sind und Kosten, die mit der Einführung einer neuen Methode einhergehen. Dabei sollen Kosten, die mit dem Einsatz einer Methode einhergehen wie etwa Lizenzkosten oder Reduktion von Kosten durch Lerneffekte schon bei der Modellierung des Entwicklungsprozesses betrachtet werden. Auf der anderen Seite stehen Kosten für die Anschaffung, Kosten für Schulungen und Trainings sowie andere Einführungskosten beispielsweise auch für die Durchführung einer Einflussanalyse wie in dieser Arbeit. Die letzteren Kosten werden bei der Bewertung des alternativen Entwicklungsprozesses als Investitionskosten oder kurz Investition bezeichnet.

Kennzahlen der Wirtschaftlichkeitsrechnung

Die Rentabilität oder auch return-on-investment einer Investition errechnet sich aus dem Gewinn, der durch die Methodenänderung ermöglicht wird, und den Kosten der Investition (Götze 2006).

$$\text{Rentabilität} = \frac{\text{Gewinn}}{\text{Investition}}$$

Die Fragestellung, ab wann sich eine Investition lohnt, wird durch die statische Amortisationsrechnung beantwortet, welche das Verhältnis zwischen den Kosten der Investition und dem möglichen Gewinn beschreibt (Götze 2006). Die Amortisationszeit gibt hier die Anzahl durchzuführender Entwicklungen bis zu einem Vorteil gegenüber dem existierenden Entwicklungsprozess an.

$$\text{Amortisation} = \frac{\text{Investition}}{\text{Gewinn}}$$

Der Kapitalwert einer Investition gibt Auskunft darüber, ob eine Investition heute vorteilhaft gegenüber einer vergleichbaren Anlage am Kapitalmarkt wäre. Dabei beschreibt T die voraussichtliche Anwendung der neuen Methode beziehungsweise des neuen Entwicklungsprozesses in Jahren, Gewinn_t den durch eine Anwendung der neuen Methode resultierenden Gewinn über alle Projekte jährlich und q einen risiko-adequaten Kalkulationszinssatz. Vergleiche dazu (Götze 2006).

$$\text{Kapitalwert} = -\text{Investition} + \sum_{t=0}^T \text{Gewinn}_t \cdot (1 + q)^{-t}$$

Bei einem Kapitalwert größer Null, ist mit der Investition ein Vorteil gegenüber dem bisherigen Entwicklungsprozess zu erwarten. Dabei ist zu beachten, dass der Mehrwert der Prozessänderung über eine entsprechende Anzahl an durchgeführten Projekten und einer bestimmten Anwendungsdauer bewertet werden muss. Der Faktor Gewinn_t beschreibt also den zu erwartenden Gewinn über mehrere Projekte in einem Jahr t .

Anwendung der Methoden

Für die Auswertung der genannten Kennzahlen werden die in den vorherigen Abschnitten definierten Differenz- und Risikomaße verwendet. So werden für die Schätzungen des Gewinns die folgenden Maße in die definierten Kennzahlen eingesetzt, die immer in Zusammenhang mit den Risikomaßen zu sehen sind:

- der schlechteste zu erwartende Gewinn D_{min} ,
- der mittlere zu erwartende Gewinn \bar{d} ,
- die untere und obere Grenze von Konfidenzintervallen um \bar{d} ,
- der beste zu erwartende Gewinn D_{max} sowie
- ausgewählte Quantile des zu erwartenden Gewinns.

Abbildung 3.14 illustriert die Auswirkung möglicher Gewinne auf die Anzahl durchzuführender Entwicklungsprozesse N mit der Methodenänderung, bis diese sich gegen die Investitionskosten I lohnt. Dies entspricht der Betrachtung der Amortisationszeit. Auf der Abszisse ist die Anzahl der durchzuführenden Entwicklungsprozesse und auf der Ordinate die möglichen Gewinne beziehungsweise die Investitionskosten aufgetragen. Diese Gegenüberstellung entspricht der sogenannten Break-Event-Point Analyse und betrachtet die Gewinnschwelle.

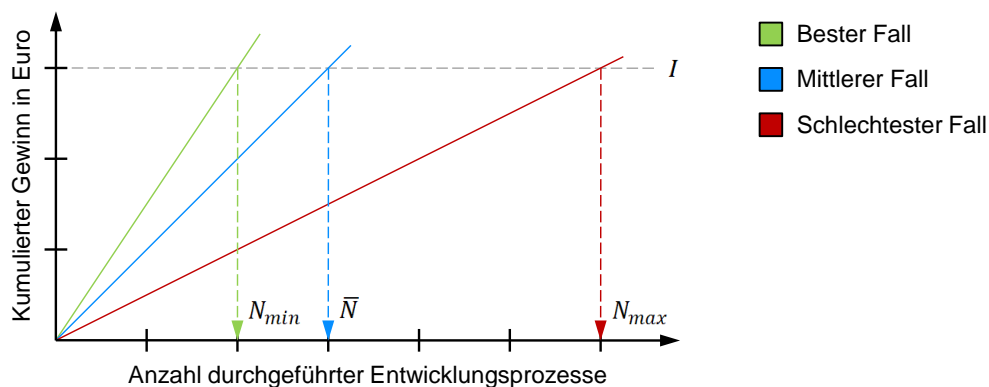


Abbildung 3.14 Gewinn als Differenz zwischen zwei Entwicklungsprozessen gegenüber der Anzahl durchzuführender Entwicklungsprozesse

Durch die unterschiedlichen Bewertungen der Differenz der beiden betrachteten alternativen Entwicklungsprozesse wird damit eine breitere Betrachtung notwendiger durchzuführender Entwicklungsprozesse ermöglicht. Damit geht die Untersuchung über die übliche Betrachtung der im Mittel \bar{N} durchzuführender Entwicklungsprozesse hinaus und gibt Informationen zu bestmöglichen N_{min} und maximal N_{max} durchzuführenden Entwicklungsprozessen.

3.14 Sensitivitätsanalyse

Die vorgestellte Bewertung fokussiert bisher auf einen festen Satz an Eingangsparametern, also einer bestimmten initialen Belegung von Eigenschaftswerten von Prozess-elementen. Für eine weitergehende Untersuchung ist anzunehmen, dass nicht alle diese Eigenschaften fest und bekannt sind, sondern es durchaus Variationen in der Belegung geben kann. Um die Auswirkungen anderer Eigenschaftswerte auf die Bewertung einer Methodenänderung zu untersuchen eignen sich daher Sensitivitätsanalysen.

Die in den vorherigen Abschnitt modellierten und bewerteten Entwicklungsprozesse dienen bei der Sensitivitätsanalyse als Vergleichsbasis zu anderen Szenarien. Ein neues entsprechendes Szenario wird durch die Variation eines oder mehrerer initialer Belegungen von Eigenschaftswerten definiert. Dabei ist festzuhalten, dass bei einer Variation von mehreren Eigenschaftswerten gegebenenfalls die Auswirkungen auf die Bewertung nicht vollständig nachvollzogen werden können. Es ist daher zu empfehlen je Simulation und Bewertung lediglich eine oder wenige Änderungen der initialen Belegung vorzunehmen.

Durch eine Sensitivitätsanalyse sollen etwa die folgenden Fragestellungen mit einem Einfluss auf das Prozessverhalten und die Bewertung der Entwicklungsprozesse untersucht werden:

- Welchen Einfluss hat ein minimal möglicher initialer Eigenschaftswert?
- Welchen Einfluss hat ein mittlerer initialer Eigenschaftswert?
- Welchen Einfluss hat ein maximal möglicher initialer Eigenschaftswert?

Weiterhin können selbstverständlich auch weitere Werte des Wertebereichs einer Eigenschaft untersucht werden. Dabei ist grundsätzlich vor der Sensitivitätsanalyse zu entscheiden, welche Eigenschaften in welchen Wertebereichen untersucht werden sollen und welche Variationen auch in Kombination untersucht werden sollen, da der Raum für mögliche Belegungen exponentiell wächst. Allein für die drei vorgestellten Fragestellungen (Minimum, Mittel, Maximum) und n Eigenschaften ergeben sich 3^n untersuchbare Szenarien. Daher sollte die Auswahl der zu variierenden Eigenschaften durch die folgenden Fragen geleitet werden:

- Welche Werte von Eigenschaften können variieren?
- Welche Werte von Eigenschaften sind unsicher?
- Welche Werte von Eigenschaften können von außen beeinflusst werden?

Die Sensitivitätsanalyse komplettiert die entwickelte Methodik und erlaubt die Untersuchung des Einfluss variierter Eigenschaftswerte auf das Prozessverhalten und die Bewertung des Mehrwerts einer neuen Methode. Der nächste Abschnitt stellt eine Einführungsstrategie der Methodik in Organisationen dar.

3.15 Einführungsstrategie

In Abschnitt 3.7 wurden Methoden vorgestellt, die für die Sammlung empirischer Daten verwendet werden können und die Basis für die Modellierung von Entwicklungsprozessen und deren Wirkzusammenhängen und Verhalten bilden. Die Reife einer Organisation ist in diesem Zusammenhang ein wichtiger Faktor, der die Auswahl möglicher Methoden für die Sammlung und Analyse von Daten einschränkt beziehungsweise leitet. So beschreiben reife Organisation, etwa im Sinne von CMMI, bereits ihre Entwicklungsprozesse und verfügen über Methoden zur quantitativen Messung von Prozess- und Produktdaten. Weniger reife Organisationen verfügen allerdings nicht explizit über solche Informationen, so dass weitere Methoden herangezogen werden müssen.

Abbildung 3.15 zeigt eine Auswahl möglicher Methoden für die Sammlung und Analyse von Daten und Informationen über Entwicklungsprozesse. Die Abbildung illustriert dabei eine Einordnung ausgewählter Methoden zur Erhebung von Informationen und konkreter Daten gegenüber dem mit ihnen verbundenem initialen Aufwand. Selbstverständlich sind die Menge und Qualität der Daten sowie der benötigte Aufwand von der konkreten Umsetzung der jeweiligen Methode abhängig.

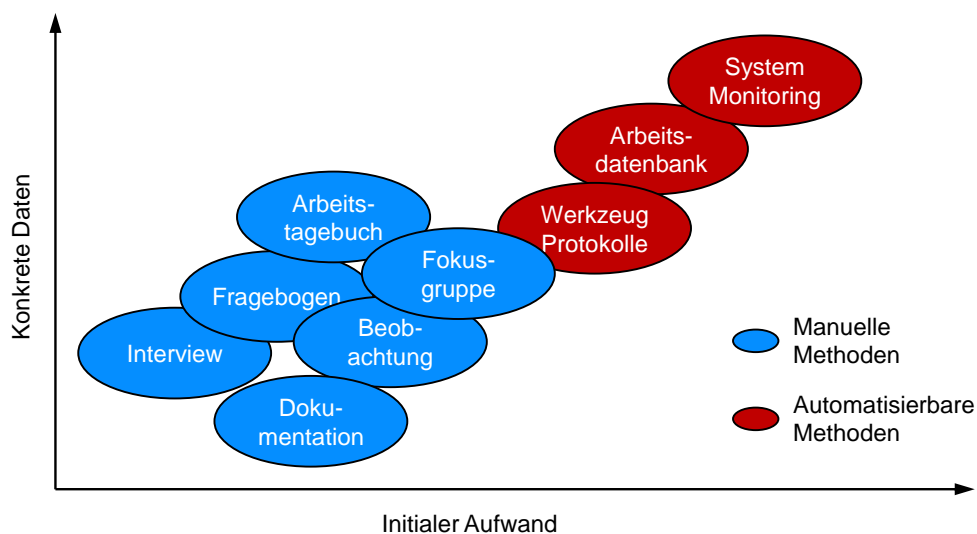


Abbildung 3.15 Aufwand von Methoden zur Gewinnung von Daten und Informationen

Es können grundsätzlich eher manuelle und automatisierbare Methoden unterschieden werden. Dabei sind die eher manuellen Methoden gut geeignet, um einen Einstieg in die Untersuchung zu erlauben. Ergänzend dazu erfordern automatisierbare Methoden einen hohen initialen Aufwand zur Erfassung von Prozess- und Produktdaten in einer Entwicklungsumgebung. Entsprechend hängt der Aufwand vom Anwendungskontext und der Fragestellung sowie der Anwendung der eingesetzten Methoden ab. Auf der

anderen Seite sinkt der relative Aufwand mit der Reife einer Organisation gerade bei automatisierbaren Methoden, da hier entsprechende Vorarbeiten in der Organisation zu erwarten sind. Allgemein kann festgehalten werden, dass jeweils im Anwendungskontext ein geeigneter Methodenmix ausgewählt werden muss.

Mit jeder Anwendung der Methodik wird Wissen über die betrachteten Entwicklungsprozesse kontinuierlich aufgebaut und vertieft. Diese Reifung der Entwicklungsprozesse und der Organisation führt in der Konsequenz auch zu einer Implementierung von Werkzeugen für die Sammlung und Analyse von Prozess- und Produktdaten während der Entwicklung, welche wiederum für Einflussanalysen verwendet werden können.

Um die grundsätzliche Einführung der Methodik in Organisationen möglichst zeitnah zu ermöglichen, werden die Aufgaben der Einflussanalyse im Folgenden auf relativ schnell anzuwendende Methoden abgebildet. Natürlich hängt die konkrete Ausprägung der Einführung von den organisationspezifischen Gegebenheiten ab und davon, welche Informationen beziehungsweise Daten bereits zur Verfügung stehen.

Kick-off Workshop

In einem Kick-off Workshop wird zunächst das Vorgehen der Methodik vom Untersuchungsleiter der Einflussanalyse vorgestellt. Der Untersuchungsleiter übernimmt die Moderation des Workshops. Dann werden die Ziele an die Einflussanalyse durch Prozessverantwortliche beziehungsweise Prozessexperten definiert (Abschnitt 3.4). Zudem wird definiert, welche Informationen und Daten zur Verfügung stehen und genutzt werden können, um die Einflussanalyse durchzuführen.

Informationen zum Entwicklungsprozess

Um den existierenden Entwicklungsprozess zu beschreiben (Abschnitt 3.5) und Wirkzusammenhänge zwischen Prozesselementen und Eigenschaften von Prozesselementen zu ermitteln (Abschnitt 3.6), wird entweder die existierende Prozessdokumentation herangezogen oder Interviews beziehungsweise eine kleine Fokusgruppe durchgeführt.

- Steht eine Prozessdokumentation zur Verfügung, so werden aus entsprechenden Dokumenten und Projektplänen die Aufgaben, Artefakte, Rollen und Werkzeuge sowie der Ablauf des Entwicklungsprozesses soweit möglich abgeleitet. Bleiben aus der Dokumentation Fragen offen, so werden Interviews oder eine Fokusgruppe durchgeführt.
- Steht keine (ausreichende) Prozessdokumentation zur Verfügung so werden Interviews oder eine Fokusgruppe zum Entwicklungsprozess durchgeführt. Dabei wird der Entwicklungsprozess mit seinen Wirkzusammenhängen interaktiv modelliert und diskutiert.

Der erhobene Entwicklungsprozess wird aufbereitet modelliert und den Prozessverantwortlichen beziehungsweise Prozessexperten vorgestellt, um die Gültigkeit des Modells zu validieren (Abschnitt 3.9).

Zur Beschreibung des Einflusses einer Methodenänderung werden ebenfalls die genannten Interviews oder die Fokusgruppe verwendet. Dabei ist zu untersuchen, welche Aufgaben und Wirkzusammenhänge sich wie verändern würden (Abschnitt 3.10).

Datensammlung, -analyse und Quantifizierung von Wirkzusammenhängen

Existieren ausreichende Daten zum Entwicklungsprozess, so werden diese durch Methoden der deskriptiven und schließenden Statistik analysiert und Modelle über Wirkzusammenhänge abgeleitet. Stehen solche Daten nicht zur Verfügung oder sind diese Daten nur eingeschränkt repräsentativ, so werden erforderliche Daten durch die Methoden von Interviews, Fokusgruppen oder Fragebögen durchgeführt (Abschnitt 3.7). Die resultierenden Daten werden aufbereitet und für die Quantifizierung der Wirkzusammenhänge herangezogen (Abschnitt 3.8). Dabei sind die quantifizierten Wirkzusammenhänge gegen die Erfahrung von Prozessverantwortlichen und Prozessexperten zu validieren (Abschnitt 3.9).

Für die quantitative Beschreibung der Methodenänderung ergeben sich die zuvor bereits genannten Methoden zur Datensammlung und Datenanalyse. Für erste „was wäre wenn“ Einflussanalysen werden Interviews mit Methodenexperten oder Prozessverantwortlichen durchgeführt.

Simulation, Bewertung und Vorstellung der Ergebnisse

Im letzten Schritt erfolgt die Vorstellung und Bewertung des Einflusses der Methodenänderung auf den Entwicklungsprozess, indem die Beschreibung der neuen Methode (Abschnitt 3.10) in das Modell des existierenden Entwicklungsprozesses integriert wird (Abschnitt 3.11) und die Prozessalternativen simuliert werden (Abschnitt 3.12). Die Ergebnisse werden aufbereitet, gegen die Erfahrung beteiligter Experten validiert und den Prozessverantwortlichen beziehungsweise Auftraggebern in einem abschließenden Termin vorgestellt.

3.16 Zusammenfassung

Im zurückliegenden Kapitel wurde eine Methodik entwickelt, um Entwicklungsprozesse mit ihrem Methoden- und Prozesswissen sowie deren Wirkzusammenhänge zu beschreiben. Ausgehend von definierten Zielen werden die Modellierung von Entwicklungsprozessen mit ihren spezifischen Prozesselementen und die Erfassung von Daten, sowie die Kalibrierung von Wirkzusammenhängen geleitet. Durch Simulation alter-

nativer Entwicklungsprozesse wird das mögliche Prozessverhalten exploriert. Verschiedene Analysen und Bewertungen erlauben eine Einschätzung der Differenz zweier Prozessalternativen und damit eine strategische Analyse sowie Bewertung von Methodenänderungen unter anderem durch bekannte Methoden der Investitionsrechnung wie Amortisation und Rentabilität.

Im Folgenden Kapitel 4 wird nun das Modell zur Modellierung von Methoden, Entwicklungsprozessen und Wirkzusammenhängen vorgestellt. Das Modell bildet dabei auch die konzeptionelle Grundlage für die Simulation von Entwicklungsprozessen und die prototypische Umsetzung des Ansatzes in Kapitel 5.

4 Modell zur Einflussbewertung

Dieses Kapitel definiert ein Metamodell, kurz auch nur Modell, zur Beschreibung von Entwicklungsprozessen und zur Einflussanalyse von Änderungen in einem Entwicklungsprozess durch neue Methoden oder neue Werkzeuge. Dabei basiert das Metamodell auf dem Software & Systems Process Engineering Meta-Model (SPEM) der Version 2.0 (OMG 2008). SPEM 2.0 ist ein Standard der Object Management Group, welcher als gemeinsamer Konsens aus verschiedenen, in der Praxis verwendeten, Modellen zur Methodenbeschreibung und Prozessmodellierung hervorgegangen ist. SPEM 2.0 wird in dieser Arbeit verwendet da es einen Teil der gesetzten Anforderungen am besten erfüllt:

- Entwicklungsprozesse können unter anderem mit ihren Aufgaben, Aktivitäten, Artefakten, Rollen und Werkzeugen beschrieben werden (**Anforderung 4**, Modellierung).
- Die Wiederverwendbarkeit von einmal definiertem Methoden- und Prozesswissen wird unterstützt (**Anforderung 6**, Wiederverwendung).
- Es existieren Werkzeuge im Bereich des Method Engineerings und der Prozessmodellierung auf Basis von SPEM beispielsweise im IBM Rational Method Composer (IBM 2006) und als quelloffene Lösung im Eclipse Process Framework Composer (Haumer 2007) (**Anforderung 10**, Werkzeugunterstützung).

Es folgt ein Überblick über die wichtigsten Konzepte des hier entwickelten Modells und schließlich eine Detaillierung der verwendeten beziehungsweise erweiterten Konzepte.

4.1 Wichtige Konzepte und Erweiterungen

SPEM 2.0 nimmt eine strikte Trennung zwischen der wiederverwendbaren methodischen Beschreibung von Prozesselementen (Definition) und der Anwendung solcher Prozesselemente in der Beschreibung von Entwicklungsprozessen vor. Die methodische Beschreibung dient dabei dem Aufbau einer Wissensdatenbank zum Austausch von aktuellen und konsistenten Informationen im organisationalen Anwendungskontext. Auf der anderen Seite „erben“ Prozessbeschreibungen von definierten Prozesselementen. Der Vorteil dieser Wiederverwendung zeigt sich deutlich bei der Reduzierung des

Aufwands zur Beschreibung mehrerer Prozessalternativen, wie es für die Einflussanalyse und den Vergleich von alternativen Entwicklungsprozessen erforderlich ist.

Abbildung 4.1 zeigt im linken Teil eine Auswahl der wesentlichen Konzepte zur Beschreibung von Methodenwissen und Prozessen mit SPEM 2.0. Im oberen linken Teil der Abbildung sind die vier wesentlichen Konzepte zur Definition von Prozesselementen zu erkennen. Zu diesen zählen *Task*, *Work Product*, *Role* und *Tool*. Prozessbeschreibungen basieren nun auf diesen Definitionen und sind im unteren Teil der Abbildung dargestellt. Die Beschreibung von Prozessen umfasst neben der Instanziierung der genannten Definitionen als *Task Use*, *Work Product Use* und *Role Use* die Konzepte *Activity* und *Process*, welche zum Aufbau von hierarchischen Strukturen verwendet werden können.

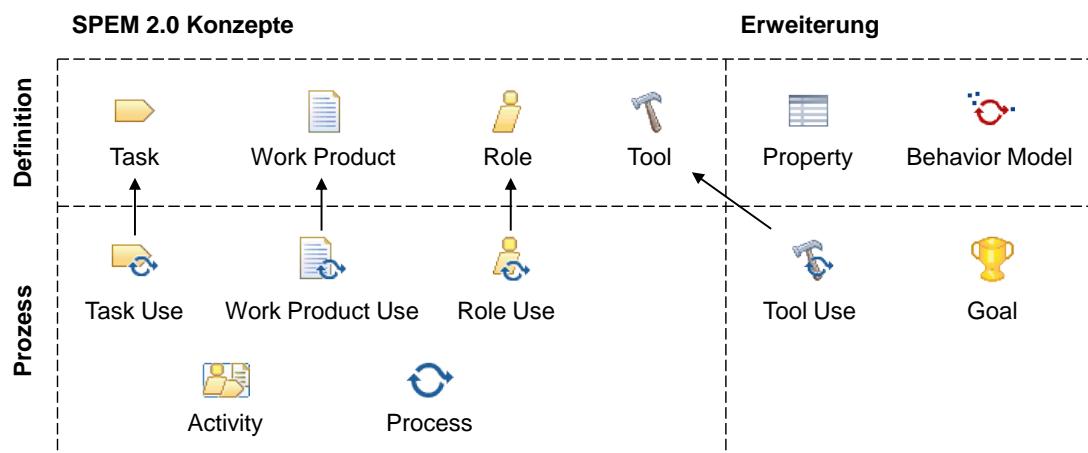


Abbildung 4.1 Wesentliche Konzepte des SPEM 2.0 Frameworks

Die durch SPEM 2.0 spezifizierten Konzepte erlauben die detaillierte Beschreibung von Prozesselementen mit ihren Relationen untereinander. Für Prozessbeschreibungen mit dem Ziel der Einflussanalyse werden auf der Seite der Definition im Wesentlichen zwei Konzepte erweitert. Das Konzept *Property* beschreibt charakteristische Eigenschaften von Prozesselementen und das Konzept *Behavior Model* wird zur Modellierung von Wirkzusammenhängen verwendet. Auf der Seite der Prozessbeschreibung wird das Konzept *Tool Use* erweitert und das Konzept *Goal* zur Beschreibung von Qualitätszielen für Eigenschaften von Design Artefakten eingeführt.

Der folgende Abschnitt beschreibt und definiert die genannten Konzepte detailliert und stellt den Zusammenhang zwischen den verschiedenen Konzepten dar.

4.2 Definition des Modells

Im Folgenden werden die wichtigsten Konzepte zur Beschreibung und Modellierung von Entwicklungsprozessen eingeführt. Dabei beschränkt sich die Beschreibung und Definition auf essentielle Ausschnitte von SPEM 2.0 beziehungsweise die genannten Erweiterungen. Für eine detailliertere Beschreibung sei daher auf die entsprechende SPEM 2.0 Spezifikation verwiesen (OMG 2008).

Jedes Prozesselement verfügt über eine Reihe von Attributen zur Beschreibung des Prozesselements. Diese Beschreibung erlaubt das Festhalten von Wissen über ein entsprechendes Prozesselement für einen bestimmten Anwendungskontext. SPEM 2.0 sieht für diese grundsätzliche Beschreibung unter anderem die folgenden Attribute vor (OMG 2008, S. 77):

- **name** ist ein eindeutiger Bezeichner des Prozesselements, welches zur Referenzierung des Prozesselements verwendet wird,
- **presentationName** ist ein Bezeichner, der die Bedeutung des Prozesselements als Name beschreibt (Beispiel: Entwickler),
- **briefDescription** ist eine kurze Beschreibung des Prozesselements mit ein bis zwei Sätzen, welche auf die Bedeutung schließen lässt,
- **mainDescription** ist eine umfangreiche Beschreibung des Prozesselements und seiner Bedeutung,
- **purpose** beschreibt die Bedeutung sowie den Zweck des Prozesselements und erklärt die Notwendigkeit des Prozesselements für den modellierten Prozess und
- **guidance** erlaubt die Assoziation von unterschiedlichen Hilfestellungen, beispielsweise in Form von Beispielen und Best Practices zum Einsatz eines Prozesselements in einem Entwicklungsprozess.

Abbildung 4.2 zeigt die Beziehungen zwischen den wesentlichen Konzepten des Metamodells. Aus Gründen der Übersichtlichkeit wird in der Abbildung auf die Darstellung der Konzepte *Work Product Use*, *Role Use*, *Tool Use* und *Process* verzichtet. So sind die ersten drei Konzepte von *Breakdown Element* abgeleitet und können somit durch das Konzept *Activity* zusammengefasst werden. Eine *Activity* ist selbst wieder ein *Breakdown Element*, wodurch hierarchische Work Breakdown Structures modelliert werden können.

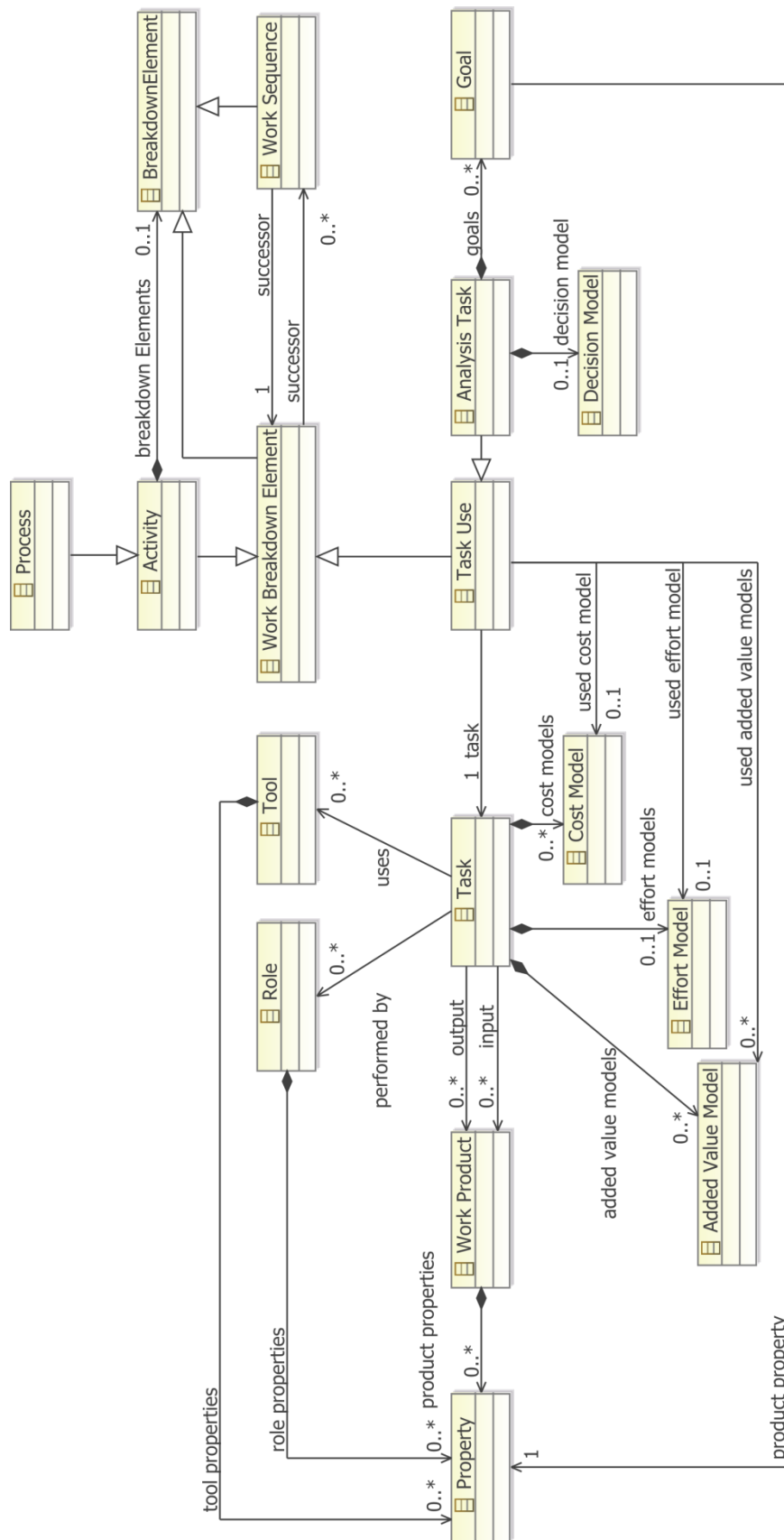


Abbildung 4.2 Wesentliche Konzepte zur Beschreibung von Methoden und konkreten Prozessen

4.2.1 Task

Aufgaben werden durch das Konzept *Task* definiert, vergleiche (OMG 2008, S. 89f). Eine Aufgabe ist eine atomare Tätigkeit, welche nicht in weitere eigenständige und sinnvolle Einheiten zerlegt werden kann. Die Granularität der Modellierung hängt von den mit der Einflussanalyse verbundenen Zielen ab. Weiterhin ist der Detaillierungsgrad vom kritischen Einfluss der betrachteten Aufgabe abhängig. So ist es unzweckmäßig kleinste Tätigkeiten zu modellieren, wenn der zu erwartende Einfluss unwesentlich ist und zu einer übergeordneten Aufgabe zusammengefasst werden kann.

Aufgaben werden von einer oder mehreren Rollen (*Role*) durchgeführt, um ein bestimmtes Ergebnis zu erreichen. Ein solches Ergebnis einer Aufgabe sind ein oder mehrere Design Artefakte (*Work Product*), welche durch die Aufgabe erstellt oder bearbeitet und damit verändert werden. Zur Durchführung einer Aufgabe können dabei eine Reihe von Werkzeugen (*Tools*) verwendet werden.

Zudem verfügt eine Aufgabe über eine Reihe von Verhaltensmodellen (*Behavior Model*), welche das Verhalten einer Aufgabe im Entwicklungsprozess beschreiben. Zu diesen Verhaltensmodellen zählen das *Effort Model* zur Modellierung des benötigten Aufwands einer Aufgabe, das *Cost Model* zur Beschreibung entstehender Kosten sowie *Added Value Models*, welche den Wertbeitrag einer Aufgabe zu einer Eigenschaft eines Design Artefakts beschreiben. Verhaltensmodelle formalisieren damit angenommene oder erhobene Wirkzusammenhänge. Näheres zu den Verhaltensmodellen ist in Abschnitt 4.2.8 zu finden.

Mit den genannten Konzepten wird eine syntaktische Beschreibung von Prozesselementen sowie eine Beschreibung des Verhaltens von Aufgaben ermöglicht. Für die Anwendung der Definition einer Aufgabe in einem Prozessmodell wird das Konzept *Task Use* verwendet (OMG 2008, S. 111f). Dabei wird der Ablauf zwischen *Task Use* durch das Konzept *Work Sequence* beschrieben (OMG 2008, S. 66f). Es werden zwei Arten von *Task Use* unterschieden:

- Design Aufgaben (*Task Use*) erstellen oder bearbeiten Design Artefakte und bestimmen damit die charakteristischen Eigenschaften (*Property*) der ausgehenden Design Artefakte und
- Analyse Aufgaben (*Analysis Task Use*) fokussieren im Gegensatz zu Design Aufgaben auf die Validierung erstellter oder bearbeiteter Design Artefakte gegen definierte Ziele (*Goal*).

Die Validierung der aktuellen charakteristischen Eigenschaften von Design Artefakten führt zu einer Entscheidung über das weitere Vorgehen im Entwicklungsprozess. Diese Entscheidung wird durch ein Entscheidungsmodell (*Decision Model*) beschrieben, siehe Abschnitt 4.2.8.

Für die Einflussanalyse verfügt eine Aufgabe über verschiedene Attribute, welche den aktuellen Zustand dieser Aufgabe im simulierten Entwicklungsprozess beschreiben. Zu diesen Attributen zählen:

- **iteration** ist ein Zähler, der die aktuelle Iteration der betrachteten Aufgabe angibt,
- **currentEffort** beschreibt den aktuellen Aufwand einer Aufgabe in Stunden für die menschliche Arbeit,
- **effort** beschreibt den kumulierten menschlichen Aufwand einer Aufgabe,
- **currentCost** beschreibt die aktuellen Kosten der Aufgabe in Euro für menschliche Arbeit und
- **cost** beschreibt die kumulierten Kosten für menschliche Arbeit.

Als Erweiterung zu den genannten Eigenschaften verfügt jede Aufgabe über entsprechende Attribute zur Beschreibung der Werkzeuglaufzeit (`currentToolEffort` und `toolEffort`) sowie zur Beschreibung der damit verbundenen Werkzeugkosten (`currentToolCost`, `toolCost`).

4.2.2 Activity und Process

Im Zusammenhang mit einer Aufgabe stehen die Konzepte *Activity* und *Process* zur Beschreibung von Aktivitäten beziehungsweise Prozessen (OMG 2008, S. 97f). Diese beiden Konzepte dienen damit der Hierarchisierung von Aufgaben in sogenannte Work Breakdown Structures. Eine Aktivität beschreibt eine Sammlung von mehreren Prozesselementen wie etwa Aufgaben, Design Artefakten, Rollen und Werkzeugen. Zum Zweck der Kommunikation der Modellierung werden verschiedene Arten von Aktivitäten unterschieden:

- eine **Aktivität** dient im Allgemeinen zur Strukturierung mehrerer Aufgaben oder Aktivitäten und beteiligter Rollen, Design Artefakte und Werkzeuge,
- eine **Iteration** kennzeichnet Teile eines Prozesses die, isoliert von anderen Teilen, üblicherweise wiederholt werden und
- eine **Phase** beschreibt signifikant unterschiedliche Abschnitte eines Prozesses.

Ein Prozess aggregiert ebenso wie eine Aktivität eine oder mehrere Aufgaben und assoziierte Prozesselemente oder Aktivitäten zu einer vollständigen Beschreibung eines Entwicklungsprozesses beziehungsweise zu einem relevanten Ausschnitt dessen.

4.2.3 Work Product

Das Konzept *Work Product* beschreibt Design Artefakte, welche eine Eingabe oder eine Ausgabe einer Aufgabe darstellen und durch diese erstellt oder bearbeitet werden (OMG 2008, S. 92f). Wesentlich für die Modellierung mit dem Ziel der Einflussanalyse ist die Auswahl von Design Artefakten mit einem signifikanten Einfluss auf den

Entwicklungsprozess beziehungsweise auf die modellierten Aufgaben. Ein Design Artefakt ist also zu modellieren, wenn dieses charakteristische Eigenschaften (*Property*) mit einem wesentlichen Einfluss besitzt. Gegebenenfalls können in der Realität vorhandene Design Artefakte zusammengefasst werden, wenn diese eine sinnvolle Aggregation erlauben.

Eigenschaften von Design Artefakten beziehen sich in der Regel auf die Qualität eines Design Artefakts oder dessen Umfang beziehungsweise Komplexität. Entsprechend besitzen diese Eigenschaften Auswirkungen auf andere Eigenschaften von Prozesselementen und werden durch andere Prozesselemente durch Wirkzusammenhänge beeinflusst. Beispiele für Eigenschaften sind etwa die Qualität, welche den Entwicklungsfortschritt bestimmt, und die Komplexität, welche den notwendigen Aufwand von Aufgaben bedingt. Solche Wirkzusammenhänge werden über *Behavior Models* von Aufgaben beschrieben (Abschnitt 4.2.8).

4.2.4 Role

Das Konzept *Role* beschreibt die Beteiligung einer oder mehrerer Rollen an einer bestimmten Aufgabe (OMG 2008, S. 87f). Die Beteiligung einer Rolle an einer Aufgabe kann in zwei Dimensionen unterschieden werden. Auf der einen Seite werden Aufgaben primär (*primarily*) von einer Rolle durchgeführt, die damit einen entsprechend großen Einfluss auf die Aufgabe und damit verbundene Design Artefakte besitzt. Auf der anderen Seite nimmt eine Rolle kleinere Tätigkeiten innerhalb einer Aufgabe wahr, so dass ihr Einfluss als geringer eingestuft werden kann (*additionally*). Aus dieser Annahme heraus werden für die Modellierung mit dem Ziel der Einflussanalyse vornehmlich primäre Rollen betrachtet. Die Entscheidung, ob eine Rolle einen signifikanten Einfluss besitzt oder nicht, muss jeweils im konkreten Anwendungskontext untersucht werden.

Rollen verfügen ebenso wie Design Artefakte über eine Menge von charakteristischen Eigenschaften. Diese Eigenschaften beeinflussen etwa den Aufwand und die Kosten einer Aufgabe oder auch die Eigenschaften von ausgehenden Design Artefakten einer Aufgabe, an der die Rolle beteiligt ist. Neben der Beeinflussung von solchen Eigenschaften von Design Artefakten existieren für Rollen Lerneffekte, die sich etwa über mehrere Iterationen einer Aufgabe zeigen. Entsprechende Wirkzusammenhänge werden über *Behavior Models* von Aufgaben beschrieben (Abschnitt 4.2.8).

4.2.5 Tool

Das Konzept *Tool* beschreibt ein bestimmtes Werkzeug oder eine bestimmte Methode, welche die Arbeit, die durch eine Aufgabe repräsentiert wird, beeinflusst (OMG 2008, S. 91f). Auch bei der Modellierung von Werkzeugen oder Methoden ist auf Zweck-

mäßigkeit zu achten, die sich in der Beteiligung eines *Tools* in Wirkzusammenhängen äußert. Werkzeuge und Methoden besitzen wie auch Design Artefakte und Rollen eine Reihe von charakteristischen Eigenschaften. Der Einfluss entsprechender Eigenschaften auf andere Eigenschaften von Prozesselementen beziehungsweise Aufwand und Kosten von Aufgaben wird ebenfalls durch *Behavior Models* beschrieben (Abschnitt 4.2.8).

4.2.6 Property

Das Konzept *Property* ist dient zur Beschreibung von charakteristischen Eigenschaften von Prozesselementen. Eigenschaften werden bei der Definition von Prozesselementen festgelegt. Dazu gehört neben dem Namen der Eigenschaft eine semantische Beschreibung, welche die korrekte Interpretation und Anwendung einer Eigenschaft in Prozessbeschreibungen ermöglicht.

Für eine Einflussanalyse werden Eigenschaften von Prozesselementen instanziiert und erhalten einen konkreten Wert zugewiesen. Das Attribut

- **value** beschreibt den aktuellen Zustand einer Eigenschaft als dezimalen Wert.

Der aktuelle Wert einer Eigenschaft beeinflusst damit das Prozessverhalten durch entsprechende Verhaltensmodelle (Abschnitt 4.2.8). Außerdem kann dieser Wert selbst wiederum durch Verhaltensmodelle beeinflusst werden.

4.2.7 Goal

Das Konzept *Goal* beschreibt ein für eine bestimmte Eigenschaft eines Design Artefakts definiertes Ziel. Dabei werden die aktuellen Eigenschaften eines Design Artefakts während einer Analyse Aufgabe gegen die definierten Ziele validiert. Ein Ziel verfügt dabei über die folgenden Attribute zur Beschreibung des Zielwertes:

- **minTarget** definiert den unteren Grenzwert zur Erreichung des Ziels,
- **minTolerance** beschreibt die Toleranz des unteren Grenzwertes in Prozent,
- **maxTarget** beschreibt den oberen Grenzwert zur Erreichung des Ziels und
- **maxTolerance** definiert die Toleranz des oberen Grenzwertes in Prozent.

Ein *Decision Model* übernimmt die Überprüfung der Eigenschaften eines Design Artefakts gegen die definierten Ziele. Damit beschreibt ein *Decision Model* außerdem das Entscheidungsverhalten zur Auswahl einer folgenden Aufgabe. Das *Decision Model* wird im folgenden Abschnitt beschrieben.

4.2.8 Behavior Model

Das Konzept *Behavior Model* wird verwendet, um das Verhalten der Arbeit einer Aufgabe im Entwicklungsprozess zu beschreiben. Grundsätzlich sind verschiedene Arten von Verhaltensmodellen wie etwa auf Basis von Zustandsdiagrammen oder mathe-

matische Funktionen denkbar. Dabei liefert ein Verhaltensmodell zu einer Reihe von Eingaben eine Ausgabe. Sowohl Eingaben als auch Ausgaben beziehen sich dabei auf definierte Eigenschaften von Prozesselementen. Im Speziellen werden zur Beachtung der Unsicherheit der Wertausprägungen von Eigenschaften und zur Abbildung von mehr oder weniger wahrscheinlichen Prozessverhalten probabilistische Funktionen verwendet.

Added Value Model

Ein *Added Value Model* beschreibt die Änderung von Eigenschaftswerten während der Durchführung einer Aufgabe. Damit wird ein *Added Value Model* verwendet, um entweder den Wertbeitrag einer Aufgabe zu einer Produkteigenschaft, wie etwa die Qualität eines Design Artefakts, oder die Veränderung von Eigenschaften einer Rolle, wie etwa die Steigerung der Erfahrung durch Lerneffekte, zu beschreiben. Formal ist ein *Added Value Model* a definiert als

$$a(i, R, T, P_{in}) = p_{out} \in \mathbb{R}$$

wobei die Parameter wie folgt zu verstehen sind

- i ist die aktuelle Iteration der Aufgabe, welche das *Added Value Model* enthält,
- R ist eine Menge von Eigenschaften von Rollen, welche die Aufgabe durchführen wie etwa die Erfahrung einer Rolle,
- T ist eine Menge von Eigenschaften von Werkzeugen oder Methoden, welche zur Durchführung der Aufgabe verwendet werden wie etwa die Genauigkeit,
- P_{in} ist eine Menge von Produkteigenschaften von Design Artefakten, die in die Aufgabe eingehen und dazu verwendet werden die Aufgabe durchzuführen wie etwa Qualität und
- p_{out} ist der neue Wert einer Eigenschaft einer Rolle oder eines Produktes.

Effort Model

Ein *Effort Model* beschreibt den für eine Aufgabe notwendigen Aufwand. Dabei werden der Aufwand für menschliche Arbeit und die notwendige Werkzeuglaufzeit unterschieden. Die Trennung des Aufwands in diese beiden Arten ermöglicht separat die Berechnung der Kosten für menschliche Arbeit etwa auf Basis des Stundenlohns einer Rolle und die Laufzeit von Werkzeugen. Ein *Effort Model* e ist eine Funktion

$$e(i, R, T, P_{in}) = effort \in \mathbb{R}^0$$

wobei die Parameter denen des *Added Value Models* entsprechen. Das Ergebnis eines *Effort Models* ist eine positive reale Zahl inklusive Null, welche den entsprechenden Aufwand in Stunden beschreibt.

Cost Model

Auf einem *Effort Model* aufbauend werden *Cost Models* definiert. So wie *Effort Models* den Aufwand für menschliche Arbeit und Werkzeuglaufzeit beschreiben, beschreiben *Cost Models* die aus dem Aufwand resultierenden Kosten. Im Speziellen kann ein *Cost Model* für Werkzeuge die Kosten verwendeter Lizenzen mit in die Berechnung einbeziehen. Ein *Cost Model* erlaubt damit die weitere Differenzierung der Kosten, beispielsweise durch unterschiedliche Stundensätze verschiedener Rollen. Ein *Cost Model* c ist eine Funktion

$$c(\textit{effort}, X) = \textit{cost} \in \mathbb{R}^0 \textit{ mit } X = R \vee T$$

wobei die oben genannten Parameter gelten und sich *effort* und X entweder auf die menschliche Arbeit beziehen, dann ist X eine Menge von Eigenschaften durchführender Rollen wie etwa „Kosten pro Stunde“ oder sich auf die Werkzeugverwendung beziehen, dann ist X eine Menge von Eigenschaften verwendeter Werkzeuge wie etwa „Lizenzkosten pro Stunde“. Das Ergebnis ist eine vom Aufwand abhängige positive reale Zahl inklusive Null, welche die Kosten in Euro beschreibt.

Decision Model

Ein *Decision Model* beschreibt das Entscheidungsverhalten für den Ablauf eines Entwicklungsprozesses im Abschluss einer Analyse Aufgabe. Die Auswahl der auf eine Analyse Aufgabe folgenden Aufgaben basiert auf der Erreichung bestimmter Ziele, die an Eigenschaften von Design Artefakten gestellt werden. Neben den zu erreichenden Zielen wird die Entscheidung für die folgende Aufgabe von Eigenschaften, von Rollen und Werkzeugen beeinflusst. So kann etwa die Erfahrung eines Testers einen signifikanten Einfluss auf die Korrektheit der Validierung der Produkteigenschaften haben. Ebenso kann die Korrektheit eines Werkzeugs einen Einfluss auf das Ergebnis der Validierung besitzen. Ein *Decision Model* d ist eine Funktion

$$d(i, R, T, P_{in}, G) = ws \in \textit{Successors}$$

wobei die Parameter wie oben gelten und G eine Menge von mit der Analyse Aufgabe verbundenen Zielen ist. Das Ergebnis eines *Decision Models* ist der Verweis auf eine folgende Aufgabe im Entwicklungsprozess aus der Menge der möglichen Nachfolger.

Können für eine Aufgabe keine sinnvollen Ziele definiert werden oder soll wegen des Modellierungsaufwandes darauf verzichtet werden, so wird auch eine probabilistische Auswahl der folgenden Aufgabe, vergleichbar zu Markov-Ketten (Abschnitt 2.7.1), zugelassen.

Für jedes der genannten Verhaltensmodelle kann die aktuelle Iteration einer Aufgabe verwendet werden, um das Prozessverhalten wie zum Beispiel Lerneffekte abzubilden.

Trotzdem wird empfohlen, wenn möglich und sinnvoll, Eigenschaften von in die Aufgabe eingehenden Prozesselementen zur Beschreibung des Verhaltens zu verwenden um damit das Wissen über Wirkzusammenhänge noch genauer zu beschreiben.

4.3 Zusammenfassung

Das zurückliegende Kapitel entwickelte auf Basis des Software & Systems Process Engineering Meta-Model (SPEM 2.0) ein Metamodell zur Beschreibung von Methodenwissen und Prozessmodellierung mit dem Ziel der Einflussanalyse von Methodenänderungen in Entwicklungsprozessen.

Zu den wesentlichen Konzepten zur Beschreibung und Modellierung von Methoden und Entwicklungsprozessen zählen Aufgaben (Task), Design Artefakte (Work Product), Rollen (Role) und Werkzeuge (Tool). Diese Konzepte wurden um Konzepte zur Beschreibung von Eigenschaften (Property) und Zielen (Goal) an Eigenschaften von Design Artefakten erweitert. Um Wirkzusammenhänge zwischen Prozesselementen beziehungsweise deren Eigenschaften zu modellieren, wurden sogenannte Verhaltensmodelle (Behavior Model) eingeführt. Verhaltensmodelle beschreiben die Änderung von Eigenschaftswerten, Aufwänden und Kosten sowie Entscheidungen für folgende Aufgaben und sind damit eine Formalisierung von Wirkzusammenhängen.

5 Prototypische Umsetzung

Die Beschreibung von Methoden- und Prozesswissen mit ihren Wirkzusammenhängen auf der einen Seite und die Einflussanalyse und Einflussbewertung auf der anderen Seite erfordern ein Werkzeug, welches bei der Anwendung der Methodik unterstützt. Ohne eine geeignete Werkzeugunterstützung (**Anforderung 10**) würden zumindest die **Anforderung 4** (Modellierung), **Anforderung 5** (Durchgängigkeit) sowie die Simulation und eine Einflussbewertung nicht ohne weiteres erfüllt beziehungsweise durchgeführt werden können.

Der in dieser Arbeit entwickelte Ansatz und der entwickelte Prototyp lassen sich in ein Gesamtkonzept zum Entwicklungsmanagement in der Produktentwicklung einordnen. Dazu gehören Werkzeuge zur Produktivitätsbewertung der Produktentwicklung, zur Unterstützung des Projektmanagements und des Projektcontrollings sowie Werkzeuge für das Produkt- und Prozesscontrolling, die in verschiedenen Forschungsarbeiten in der Abteilung Business Engineering an der Universität Oldenburg und am OFFIS Institut für Informatik Oldenburg unter Leitung von Prof. Dr.-Ing. Hahn entwickelt wurden und werden (Hausmann 2008), (Strickmann 2008), (Häusler et al. 2009), (große Austing und Hahn 2010).

In den folgenden Abschnitten werden die für die Umsetzung des in dieser Arbeit entwickelten Ansatzes verwendeten Technologien (Abschnitt 5.1), die grobe Architektur (Abschnitt 5.2) und die Implementierung beziehungsweise die Anwendung des Prototyps (Abschnitt 5.3) vorgestellt.

5.1 Technologien

Die Basis des Prototyps wird durch die plattformunabhängige Sprache Java mit dem Java Development Kit (JDK) in der Version 1.6 gebildet. Auf Java aufbauend wird die Entwicklungsplattform Eclipse¹ in der Version 3.6 verwendet. Eclipse bringt eine Reihe grundsätzlicher Funktionalitäten unter anderem für Rich Clients mit und erlaubt die

¹ <http://www.eclipse.org>

modularisierte Entwicklung und Kombination von Softwarekomponenten als sogenannte Plugins beziehungsweise Bundles.

Das Eclipse Process Framework² (EPF) in der Version 1.5 wird als Referenzimplementierung des Software & Systems Process Engineering Meta-Model (SPEM) zur Beschreibung von Methoden und Entwicklungsprozessen verwendet. Zudem werden das Eclipse Modeling Framework³ (EMF) in der Version 2.6 und das Graphical Modeling Framework⁴ (GMF) in der Version 1.4 zur Implementierung des Metamodells, wie es in Kapitel 4 konzeptionell vorgestellt wurde, benutzt. EMF und GMF erlauben die modellgetriebene Entwicklung von Basiseditoren zur Bearbeitung von Instanzen eines entsprechenden Modells – hier von Methodenbeschreibungen und Entwicklungsprozessen.

5.2 Architektur

Die Architektur der prototypischen Umsetzung teilt sich grob in drei Teile. Abbildung 5.1 zeigt diese Aufteilung in Benutzungsschnittstellen, Kernkomponenten und Erweiterungen. Die Pfeile zwischen den skizzierten Komponenten beschreiben den wesentlichen Informationsfluss.

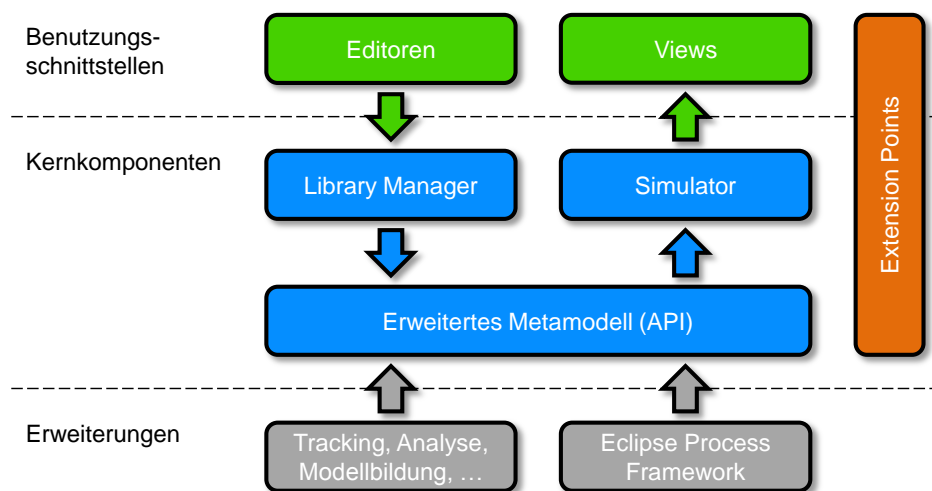


Abbildung 5.1 Architektur der prototypischen Umsetzung

Auf der obersten Ebene stehen Komponenten zur Interaktion mit dem Benutzer. Hier erlauben *Editoren* die Bearbeitung und Modellierung von Methoden, Entwicklungspro-

² <http://www.eclipse.org/epf>

³ <http://www.eclipse.org/modeling/emf>

⁴ <http://www.eclipse.org/modeling/gmp>

zessen sowie Wirkzusammenhängen. Daneben werden verschiedene *Views* bereitgestellt, welche den Zustand beziehungsweise die Ergebnisse von Simulationen betrachteter Entwicklungsprozesse darstellen. Zu den Kernkomponenten zählen der *Library Manager*, welcher die Verwaltung aktuell zu bearbeitender Entwicklungsprozesse übernimmt, das *erweiterte Metamodell* zur Beschreibung von Entwicklungsprozessen mit einer entsprechenden Application Programming Interface (API) sowie der *Simulator* zur Simulation von Entwicklungsprozessen. Auf der unteren Ebene sind Erweiterungen denkbar, welche etwa bei der Modellbildung und Analyse unterstützen – diese Komponenten sind allerdings nicht Teil der vorliegenden Arbeit. Eine Schnittstelle auf der unteren Ebene ermöglicht Modellierungen mit dem *Eclipse Process Framework*, welche importiert werden können. Orthogonal zu den ersten beiden Ebenen werden sogenannte *Extension Points* für die skizzierten Komponenten definiert. Diese *Extension Points* erlauben die Erweiterung der entwickelten Benutzungsschnittstellen sowie der Kernkomponenten, wenn dieses erforderlich scheint.

5.3 Implementierung

Die Grundlage der Methoden- und Prozessbeschreibung wird durch das in Kapitel 4 vorgestellte konzeptionelle Modell auf Basis des Software & Systems Process Engineering Meta-Model (SPEM) gebildet. Mit dem Eclipse Process Framework (EPF) Composer wird ein Werkzeug angeboten, welches SPEM als Referenzimplementierung quelloffen umsetzt. Entsprechend der vorgestellten konzeptionellen Erweiterungen (Kapitel 4) wird das verfügbare EPF Modell erweitert, um auch Wirkzusammenhänge für Einflussanalysen sowie Einflussbewertungen modellieren zu können. Auf eine detaillierte Diskussion der konkreten Implementierung wird an dieser Stelle verzichtet, da sie im Wesentlichen dem konzeptionellen Modell entspricht.

Für die Simulation modellierter Entwicklungsprozesse wurde ein diskreter Simulator entwickelt. Dabei werden die Beschreibung und das modellierte Verhalten eines Entwicklungsprozesses in ein ausführbares Modell überführt. Der Simulator führt dann, beginnend bei der ersten Aufgabe, eine Aufgabe mit dem spezifizierten Verhalten aus und wählt die nächste Aufgabe gemäß definierter Entscheidungsmodelle (Abschnitt 4.2.8). Jeweils nach der Ausführung einer Aufgabe werden die aktuellen Eigenschaftswerte von Prozesselementen sowie Aufwände und Kosten gespeichert, um diese Daten für Analysen bereitzustellen. *Extension Points* erlauben hierbei eine problembezogene Speicherung und Aufbereitung zur Verfügung stehender Daten aus der Simulation.

Im Folgenden wird die Umsetzung und die Anwendung der im vorherigen Abschnitt skizzierten Komponenten zur Unterstützung des in dieser Arbeit vorgeschlagenen Ansatzes anhand der Schritte der Methodik und des Beispiels aus Abschnitt 1.5 vorgestellt. Vergleiche entsprechend dazu auch die Beschreibung der Methodik in Kapitel 3.

5.3.1 Definition der Ziele der Einflussanalyse

Der entwickelte Prototyp verzichtet in der vorgestellten Form derzeit auf eine explizite Unterstützung für die Definition von Zielen für eine Einflussanalyse beziehungsweise Einflussbewertung. Die Definition entsprechender Ziele ist ein wichtiger Schritt, doch kann dieser auf relativ einfache Weise auch manuell erfolgen und bedarf daher nicht zwingend einer Werkzeugunterstützung.

5.3.2 Beschreibung des Entwicklungsprozesses

Die Beschreibung des Entwicklungsprozesses gliedert sich in zwei Abschnitte: die Definition von Prozesselementen und die Modellierung des Entwicklungsprozesses. Dabei werden zunächst die als wesentlich identifizierten Prozesselemente definiert, welche dann für die Beschreibung von konkreten Entwicklungsprozessen wiederverwendet werden.

Abbildung 5.2 zeigt die Definition von Prozesselementen am Beispiel des Entwicklungsprozesses aus Abschnitt 1.5 mit dem entwickelten Werkzeug, vergleiche auch Abschnitt 3.5. Die sogenannte *Library View* erlaubt die hierarchische und strukturierte Definition von Prozesselementen und spiegelt immer das aktuelle Modell der Definition von Prozesselementen und der Beschreibung von Entwicklungsprozessen wider.

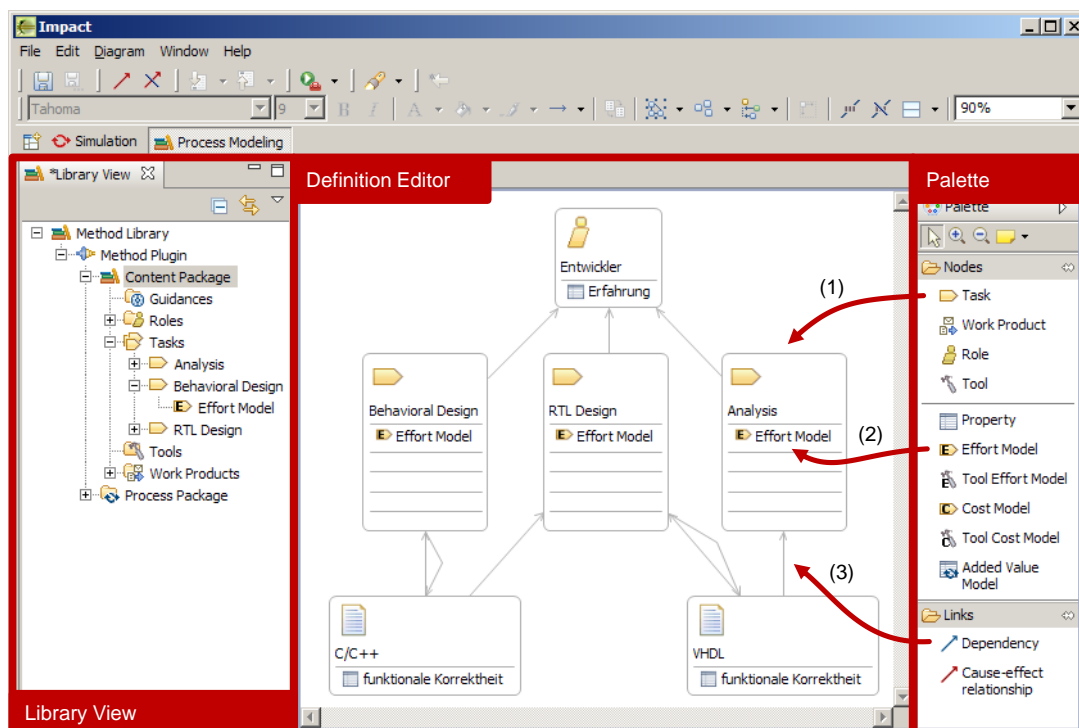


Abbildung 5.2 Definition von Prozesselementen

Den größten Teil nimmt der entwickelte Editor zur grafischen Definition von Prozesselementen ein. Zur Definition von Prozesselementen werden entsprechende Elemente (*Nodes*) aus der *Palette* im Editorbereich abgelegt (1). Eigenschaften und Verhaltensmodelle werden ebenso in der Palette ausgewählt und den entsprechenden Prozesselementen zugeordnet (2). Beziehungen zwischen Prozesselementen werden mit Hilfe des Werkzeugs *Dependency* aus der Rubrik *Links* modelliert (3).

Steht die Definition der wesentlichen Prozesselemente, lässt sich daraus per Drag & Drop aus der *Library View* der konkrete Entwicklungsprozess zusammensetzen (1), wie es in Abbildung 5.3 für das *VHDL* Artefakt veranschaulicht wird. Dabei werden die in der Definition beschriebenen Beziehungen zwischen Prozesselementen automatisch übernommen. Der Ablauf des Entwicklungsprozesses wird mit Hilfe des Werkzeugs *Work Order* aus der Palette definiert (2). Die Elemente *Activity*, *Iteration* und *Phase* erlauben eine hierarchische Strukturierung der Prozessbeschreibung. Mit einem Doppelklick auf ein solches Element öffnet sich ein neuer Editor, in welchem dann untergeordnete Prozesselemente auf die gleiche Art und Weise beschrieben werden können, um entsprechende Work Breakdown Structures aufzubauen.

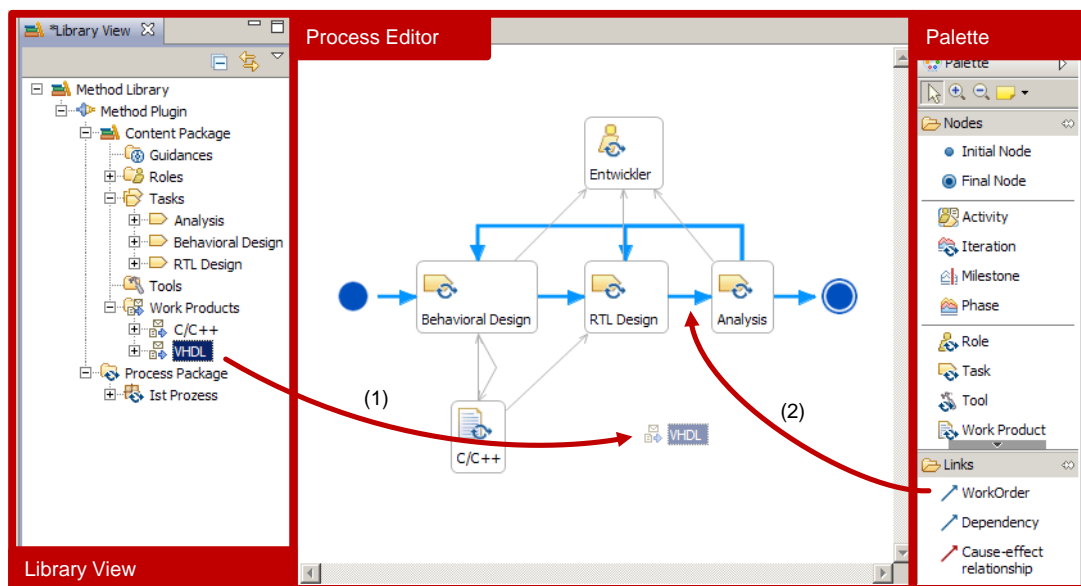


Abbildung 5.3 Beschreibung des konkreten Entwicklungsprozesses

5.3.3 Identifizierung und Beschreibung von Wirkzusammenhängen

Wirkzusammenhänge zwischen Prozesselementen beziehungsweise Eigenschaften von Prozesselementen werden ebenso wie zuvor die Beziehungen zwischen Prozesselementen beschrieben. Abbildung 5.4 zeigt den modellierten Entwicklungsprozess im entsprechenden Editor mit identifizierten Wirkzusammenhängen. Zur Beschreibung von Wirkzusammenhängen steht in der Palette die Beziehung *Cause-effect relationship* zur

Verfügung. Neben der damit definierten strukturellen Beschreibung von Wirkzusammenhängen wird zudem die textuelle Beschreibung eines Wirkzusammenhangs ermöglicht, um entsprechende Annahmen oder Wissen zu beschreiben.

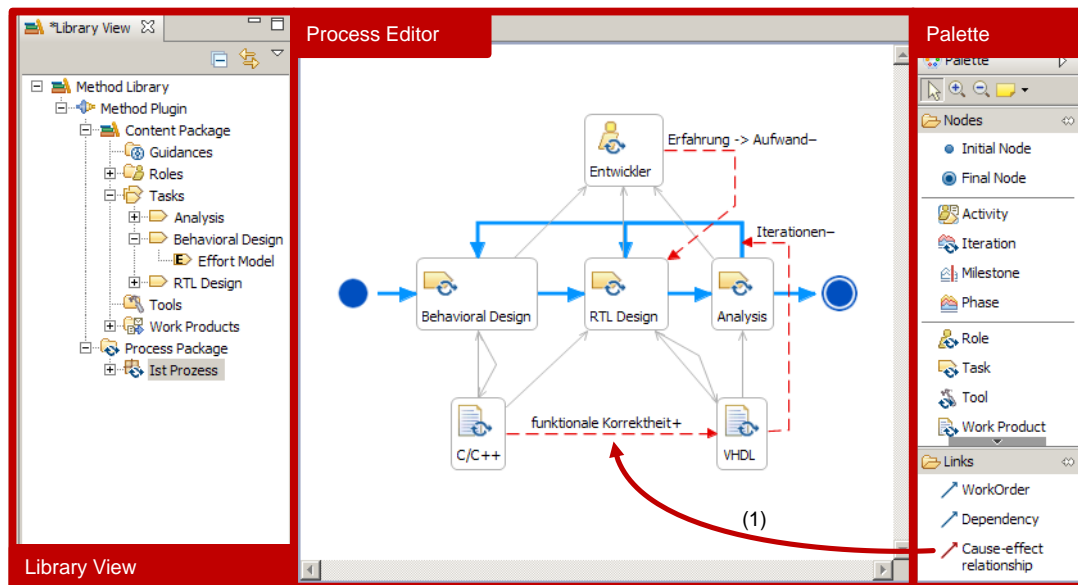


Abbildung 5.4 Beschreibung von Wirkzusammenhängen im Entwicklungsprozess

Bei Bedarf können beschriebene Wirkzusammenhänge ausgeblendet werden, um eine bessere Übersichtlichkeit des Prozessdiagramms zu gewährleisten.

5.3.4 Sammlung und Analyse empirischer Daten und Quantifizierung von Wirkzusammenhängen

Die Sammlung und Analyse empirischer Daten steht sowie auch die Modellbildung auf Basis entsprechender Daten nicht im Fokus der vorgestellten prototypischen Werkzeugunterstützung. Vielmehr erlaubt der Prototyp durch entsprechende Schnittstellen die Erweiterbarkeit und Anbindung entsprechender Werkzeuge.

Dennoch ist die Beschreibung von Verhaltensmodellen als mathematische Funktionen definierter Eigenschaften von Prozesselementen möglich. Dabei werden Verhaltensmodelle der Definition einer Aufgabe zugeordnet, wie bereits in Abbildung 5.2 zu sehen ist. Durch Doppelklick auf ein entsprechendes Verhaltensmodell in der Definition wird dieses zur Bearbeitung geöffnet. Abbildung 5.5 zeigt exemplarisch die Beschreibung eines Verhaltensmodells für den Aufwand der Aufgabe Behavioral Design.

Im linken Bereich wird das Verhaltensmodell auf Basis der Eigenschaften assoziierter Prozesselemente und erweiterbaren Funktionen beschrieben. Dabei stehen nur Eigenschaften von Prozesselementen zur Verfügung, welche strukturell mit der betrachteten Aufgabe in Beziehung stehen. Eigenschaften und Funktionen werden dem Verhaltens-

modell durch Doppelklick auf ein entsprechendes Element im rechten Baum hinzugefügt.

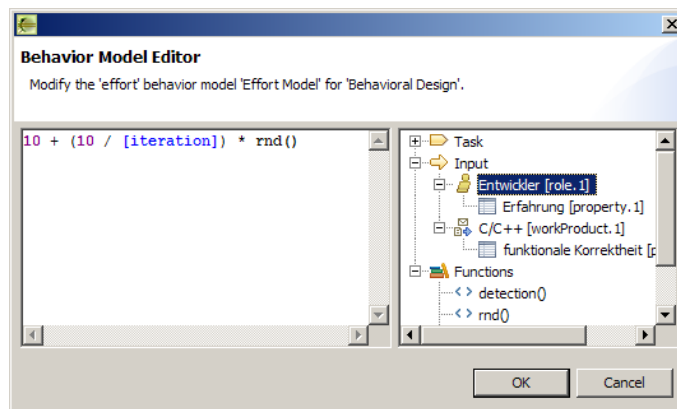


Abbildung 5.5 Modellierung von Verhaltensmodellen

Die Definition einer Aufgabe kann mehrere Verhaltensmodelle besitzen, die dann von einer entsprechenden Aufgabe im konkreten Modell eines Entwicklungsprozesses wiederverwendet werden können. Zudem werden neben dieser Implementierung zur Unterstützung von mathematischen Verhaltensmodellen Schnittstellen angeboten, um weitergehende Modelle, beispielsweise auf Basis von Zustandsdiagrammen, einzubinden und zu verwenden.

Gegenüber den Verhaltensmodellen zur Beschreibung von Wertänderungen, Aufwand und Kosten stehen Verhaltensmodelle zur Beschreibung des Entscheidungsverhaltens für folgende Aufgaben. Diese werden für das Modell eines Entwicklungsprozesses über die Struktur der *Library View* definiert. Der Prototyp erlaubt dabei die Modellierung von Entscheidungen auf Grundlage von definierten Zielen an Eigenschaften von Design Artefakten sowie auch die Modellierung von Übergangswahrscheinlichkeiten von einer Aufgabe zu einer anderen. Abbildung 5.6 veranschaulicht die strukturierte Definition eines Entscheidungsmodells für die *Analysis* Aufgabe (1) auf Basis eines Ziels an die Eigenschaft *funktionale Korrektheit* des *VHDL* Artefakts (2).

Entscheidungsmodelle für Ziele und probabilistisches Verhalten können auch miteinander kombiniert werden. In dem gezeigten Beispiel wird die Eigenschaft *funktionale Korrektheit* gegen das definierte Ziel geprüft. Wenn dieses Ziel erfüllt werden kann, wird der Prozess beendet (3), ansonsten wird probabilistisch ausgewählt, welche Aufgabe als nächstes folgt (4).

5.3.5 Validierung des Modells

Die prototypische Umsetzung ermöglicht die Validierung des entwickelten Systems über mehrere Wege. Zum einen eignet sich insbesondere die Beschreibung des kon-

kreten Entwicklungsprozesses und abgebildeten Wirkzusammenhängen zur Diskussion mit Experten des Entwicklungsprozesses. Auf der anderen Seite werden eine schrittweise Simulation sowie auch eine Durchführung mehrfacher Simulationsläufe modellierter Entwicklungsprozesse unterstützt.

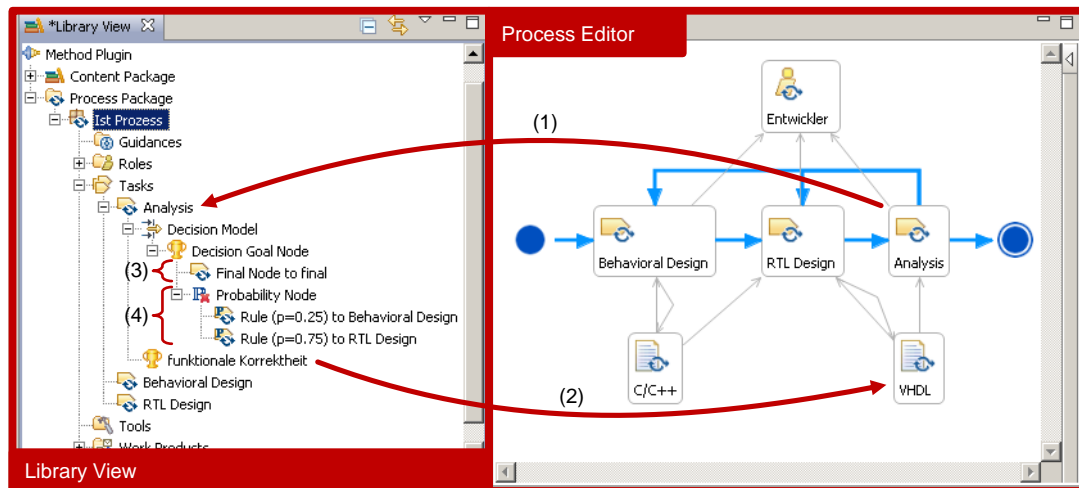


Abbildung 5.6 Beschreibung eines Entscheidungsmodells

Abbildung 5.7 zeigt die schrittweise Simulation eines modellierten Entwicklungsprozesses, welche über die *Simulation View* gesteuert werden kann. Verfügt ein Modell über modellierte Aktivitäten, so lassen sich diese auch einzeln simulieren und untersuchen. Der aktuelle Schritt im Ablauf des Entwicklungsprozesses wird rot im Diagramm eingefärbt. Die *Value View* auf der rechten Seite der Abbildung ermöglicht die Beobachtung aktueller Eigenschaftswerte von Prozesselementen sowie deren Verlauf über die Zeit. Auf Basis der Eigenschaftswerte und deren Verlauf über die Zeit kann das mögliche Verhalten des modellierten Entwicklungsprozesses untersucht werden. Anzumerken ist dabei, dass das beobachtete Verhalten bei probabilistischen Wirkzusammenhängen nicht deterministisch ist. Die schrittweise Simulation liefert daher nur ein mögliches Szenario von vielen.

Ergänzend zur schrittweisen Simulation ermöglicht die mehrfache Simulation des modellierten Entwicklungsprozesses die statistische Untersuchung des Prozessverhaltens und zeigt damit mehr oder weniger wahrscheinliche Szenarien auf. Abschnitt 5.3.7 geht auf die Werkzeugunterstützung zur mehrfachen Simulation ein.

5.3.6 Beschreibung der neuen Methode und virtuelle Integration der neuen Methode

Die Beschreibung der neuen Methode erfolgt analog zur Definition des Entwicklungsprozesses mit Hilfe der bereits beschriebenen Editoren. So werden neue Aufgaben und Beziehungen und entsprechende Verhaltensmodelle definiert. Sind ergänzende

Verhaltensmodelle zum bereits beschriebenen Entwicklungsprozess notwendig, so können diese den entsprechenden Aufgaben hinzugefügt werden.

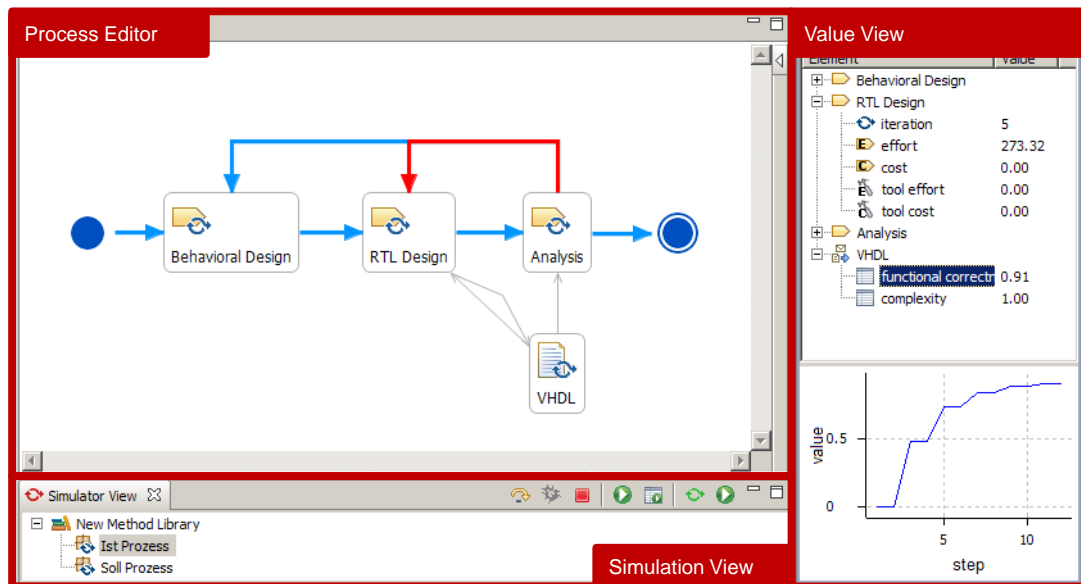


Abbildung 5.7 Schrittweise Simulation eines modellierten Entwicklungsprozesses

Die Integration der neuen Methode beziehungsweise der Prozessänderung erfolgt ebenfalls analog zur Beschreibung von Entwicklungsprozessen. So wird dabei auf die Definition entsprechender Prozesselemente zurückgegriffen.

Mit der Beschreibung der Prozessänderung stehen nun verschiedene Prozessalternativen zur Verfügung, die im folgenden Schritt durch Simulation weiter untersucht werden.

5.3.7 Simulation der Entwicklungsprozesse und Bewertung des Einflusses der neuen Methode

Die mehrfache Simulation eines Entwicklungsprozesses erlaubt die statistische Auswertung des modellierten Prozessverhaltens sowie die Bewertung der Differenz verschiedener Prozessalternativen. Die Anzahl der durchzuführenden Simulationsläufe kann dabei benutzerdefiniert eingestellt werden. In der Regel liefern bereits 1000 Läufe ein charakteristisches Bild der modellierten Entwicklungsprozesse.

Abbildung 5.8 zeigt einen Ausschnitt der implementierten Analysen, hier für den notwendigen Aufwand der simulierten Entwicklungsprozesse. Es stehen weitere Analysen etwa zur Verteilung des Aufwands über die verschiedenen Aufgaben, über die Eigenschaften von Prozesselementen sowie Break-even Analysen zur Bewertung des Mehrwerts einer Prozessänderung zur Verfügung. Zudem werden Schnittstellen angeboten, um weitere Analysen auf Basis der während einer Simulation gesammelten Daten zu erlauben.

Um entsprechende Analysen auszuführen werden zunächst die zu untersuchenden Entwicklungsprozesse im oberen linken Teil ausgewählt. Dann können im unteren linken Teil der Abbildung eine zur Verfügung stehende Analyse für Prozesse, einzelne Aufgaben oder andere Prozesselemente, wie etwa Design Artefakte, gewählt werden. Im Anschluss an diese Auswahl aktualisieren sich entsprechende Auswertungen im rechten Teil der Abbildung.

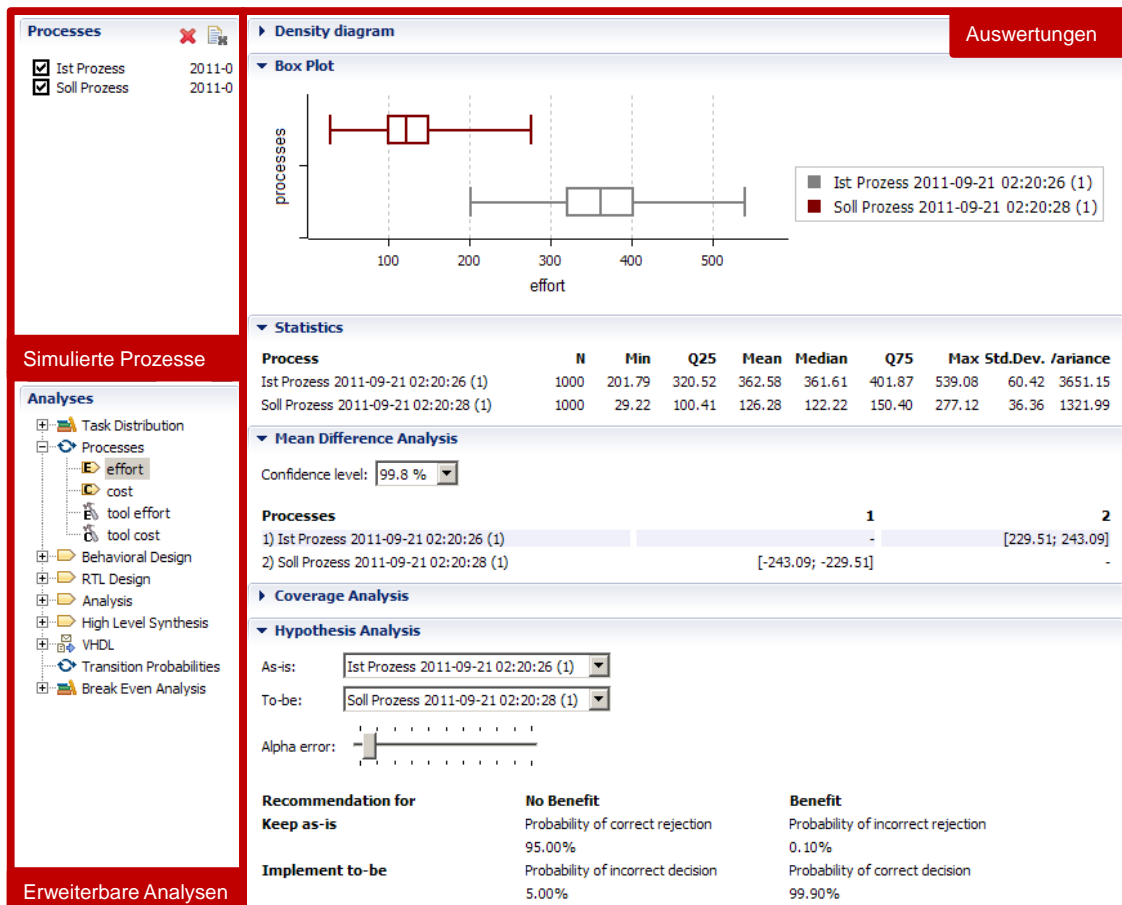


Abbildung 5.8 Analyse und Bewertungen für mehrfache Simulationen

5.3.8 Sensitivitätsanalyse

Standardmäßig erfolgt die Simulation von Entwicklungsprozessen beginnend mit den in der Modellierung gesetzten Eigenschaftswerten. Für Sensitivitätsanalysen müssten diese Eigenschaftswerte in der entsprechenden Definition geändert werden. Um dieses handhabbarer zu ermöglichen wurde ein *Wizard* implementiert, mit dem verschiedene Einstellungen für die Simulation eines Entwicklungsprozesses vorgenommen werden können. Abbildung 5.9 illustriert diese Möglichkeiten exemplarisch. Dabei wird eine Variation der Eigenschaftswerte für die Erfahrung der modellierten Rolle eines Entwicklers durchgeführt. Hierbei werden die Grenzen sowie die Schrittweite für die

Sensitivitätsanalyse angegeben. Entsprechend der gezeigten Konfiguration würden bei dieser Einstellung 1000 x 11 (siehe 1 und 2) Simulationsläufe für den modellierten Entwicklungsprozess durchgeführt, welche dann in der Analyse miteinander verglichen und bewertet werden können.

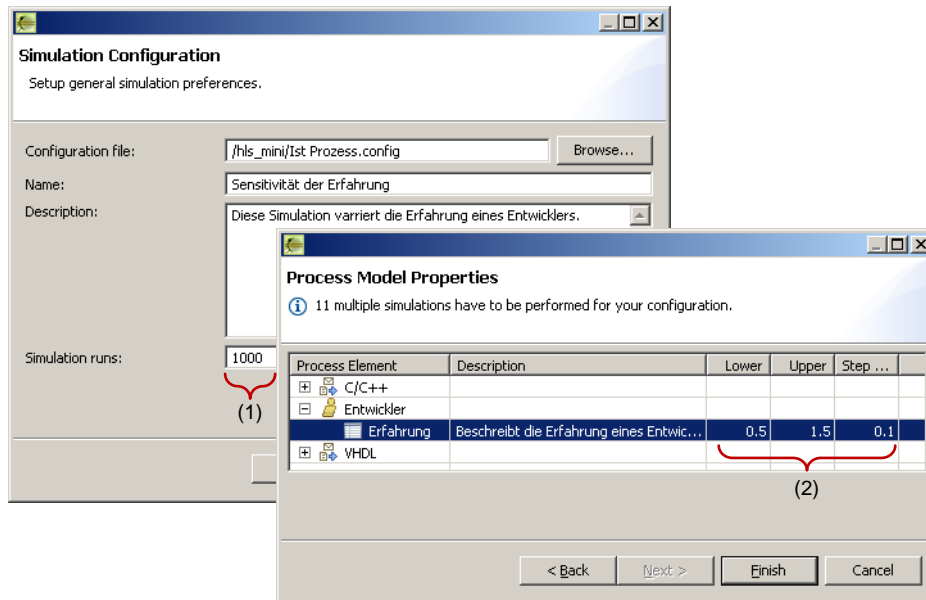


Abbildung 5.9 Wizard zur Konfiguration von Eigenschaftswerten und Simulationsläufen

5.4 Zusammenfassung

Der in diesem Kapitel vorgestellte Prototyp unterstützt bei der Modellierung von Methodenwissen, Entwicklungsprozessen und Wirkzusammenhängen. Darüber hinaus erlaubt er die Simulation von beschriebenen Entwicklungsprozessen und die Analyse des modellierter Entwicklungsprozesse, welches für die Einflussanalyse und Einflussbewertung Verwendung findet.

Die prototypische Umsetzung wurde für alle in der Evaluation, siehe Kapitel 6, betrachteten Anwendungsbeispiele zur Modellierung von Entwicklungsprozessen sowie zur Simulation, Analyse und Bewertung eingesetzt. Die Evaluation umfasst damit nicht nur die Anwendung der Methodik für unterschiedliche Anwendungsbeispiele sondern auch die Anwendung des entwickelten Prototyps. Erkenntnisse zu möglichen Verbesserungen der prototypischen Umsetzung wurden entsprechend bei der weiteren Implementierung beachtet.

Insgesamt deckt die prototypische Umsetzung die Anforderungen an die Modellierung von Entwicklungsprozessen, der Durchgängigkeit sowie die Analyse mittels entsprechender Kennzahlen zur Bewertung von Methodenänderungen ab. Im Abschnitt 6.6 werden die aufgestellten Anforderungen noch einmal detaillierter betrachtet.

6 Evaluierung

In diesem Kapitel wird ein Überblick über eine Reihe von Untersuchungen gegeben, die auf der einen Seite eine Basis für die Entwicklung von Methodik und Metamodell bildeten und auf der anderen Seite für die Evaluierung der Arbeit herangezogen wurden. Tabelle 6.1 zeigt eine Übersicht der betrachteten Anwendungen. In den rechten Spalten markiert ein Haken die Instanziierung des Metamodells, die Anwendung der Methodik und die Modellierung von Entwicklungsprozessen mit dem entwickelten Prototypen.

Tabelle 6.1 Übersicht betrachteter Untersuchungen und Anwendungsbeispiele

Untersuchung	Domäne	Meta-modell	Methodik	Prototyp
Gammatone Filterbank	Hardware	✓	(✓)	✓
High Level Synthesis	Hardware	✓	✓	✓
Software Inspections	Software	✓	(✓)	✓
Software Reviews	Software	✓	✓	✓
Lebensmittel Entwicklung	Lebensmittel-industrie	✓	✓	✓

Die erste Untersuchung diente als Vorstudie in Ergänzung zum Stand der Technik, um die grundlegenden Mechanismen von Entwicklungsprozessen zu verstehen und daraus Metamodell und Methodik zu entwickeln (Abschnitt 6.1). Im zweiten Anwendungsbeispiel wurde der Einfluss der High Level Synthesis auf die Hardware-Entwicklung untersucht (Abschnitt 6.2). Die dritte Untersuchung betrachtet den Einfluss von Software Inspections auf Basis von Daten in der Literatur (Abschnitt 6.3). Ein weiteres Praxisbeispiel aus der Domäne der Software-Entwicklung untersucht den Einfluss von Reviews auf die Entwicklung von Fehlern und damit den Aufwand der Entwicklung bei

einem großen Automobilzulieferer (Abschnitt 6.4). Das letzte Praxisbeispiel stammt aus der Lebensmittelindustrie zur Entwicklung eines neuen Lebensmittels (Abschnitt 6.5).

6.1 Gammatone Filterbank

Als Vorstudie und Ergänzung zum Stand der Technik dieser Arbeit wurde eine Untersuchung im Bereich der Hardware-Entwicklung durchgeführt (Koppe et al. 2011a). Das Ziel dabei war das Verstehen der grundlegenden Mechanismen von Entwicklungsprozessen sowie die Ableitung von Erkenntnissen für die Modellierung von Entwicklungsprozessen und die Durchführung von Untersuchungen zur Einflussanalyse.

Die Gammatone Filterbank ist eine psycho-akustische Filterbank eines Hörgeräts zur Unterstützung hörgeschädigter Menschen (Poppen et al. 2006). Für die Untersuchung wurde mit einem Experten der Entwicklungsprozess für ein existierendes Design der Gammatone Filterbank definiert. Der Experte übernahm auch die Aufgabe der Entwicklung. Die Verwendung eines bereits bekannten Designs gründet sich auf die folgenden Annahmen:

- es sind kaum unerwartete Ereignisse während der Entwicklung zu erwarten und
- gesammelte Daten über Produkteigenschaften und Prozessverhalten können relativ einfach auf Plausibilität geprüft werden.

Es wurden drei neue Anforderungen an das Design gestellt, die im Ergebnis des Entwicklungsprozesses erreicht werden sollten. Dabei beziehen sich diese Anforderung beziehungsweise Ziele auf Eigenschaften des Designs:

- eine Clock Frequenz von 200 MHz und eine Clock Period von 5 ns (timing),
- die Fläche des resultierenden Chips (area) soll bei maximal 1 mm² liegen und
- der Energiebedarf (power) soll bei maximal 20 mW liegen.

Bei den genannten Zielen ist das Timing ein hartes Ziel, welches unbedingt erreicht werden muss. Die beiden anderen Ziele sind weicher und werden als erfolgreich bewertet, wenn die Abweichung vom gegebenen Wert nicht größer als 5% ist.

Der Entwicklungsprozess besteht aus vier aufeinanderfolgenden Aufgaben. In der *Design* Aufgabe wird das Design bearbeitet und auf die Ziele hin optimiert. Durch die *Synthesis* Aufgabe wird das Design in eine Gatternetzliste übersetzt, die in *Place & Route* auf der Fläche ausgerichtet wird. Während der *Analysis* Aufgabe werden dann die Eigenschaften des Designs durch den Entwickler gegen die Ziele geprüft und entschieden, welche Aufgabe als nächstes durchgeführt werden soll. Dabei werden für jede Aufgabe bestimmte Werkzeuge verwendet.

Während der Entwicklung wurden durch den Entwickler die folgenden Dinge schriftlich protokolliert:

- die Abfolge der einzelnen Aufgaben,
- der Grund einer Iteration zu früheren Aufgaben,
- der Aufwand der menschlichen Arbeit für jede Aufgabe und Iteration sowie
- die Werkzeuglaufzeit für jede Aufgabe.

Ergebnisse

Tabelle 6.2 gibt eine Übersicht über die während der Untersuchung quantitativ erhobenen Produkt- und Prozessdaten. Dabei zeigt die Tabelle die für jede Iteration (I) angefallenen personenbezogenen Aufwände (a) und die jeweilige Werkzeuglaufzeit (b) über die einzelnen Aktivitäten im Entwicklungsprozess in Stunden. Auf der rechten Seite sind die am Ende jeder Iteration durch die Analysis Aufgabe festgestellten Eigenschaftswerte des Designs dargestellt.

Tabelle 6.2 Protokollierter Aufwand, Werkzeuglaufzeit und Eigenschaftswerte

I	Design		Synthesis		Place & Route		Analysis		area	timing	power
	a	b	a	B	a	b	a	b			
0	0,10	0,01	0,60	0,02	2,60	0,02	0,40	2,76	0,76	6,66	50,35
1	3,30	0,02	0,20	0,04	1,00	0,06	2,10	3,66	0,76	11,00	17,50
2			0,50	0,05	0,70	0,05	2,00	13,28	0,94	6,76	25,08
3	2,50	0,05	0,25	0,04	0,70	0,05	1,00	3,73	0,67	6,76	19,84
4			0,17	0,04	0,25	0,05	0,60	3,49	0,68	8,60	19,40
5			0,23	0,12	0,50	0,09	1,10	3,97	0,68	6,80	20,60
6					2,20	0,82	2,30	0,34	0,68	–	–
7	0,40	0,02	0,10	0,03	0,40	0,10	1,10	0,08	0,63	–	–
8					1,00	0,47	0,90	3,76	0,63	5,00	20,90

Abbildung 6.1 veranschaulicht den für jede Iteration kumulierten menschlichen Aufwand und die Werkzeuglaufzeit. Es zeigt sich, dass der Aufwand für die menschliche Arbeit über die Entwicklung von etwa 6 Stunden auf 2 Stunden sinkt, da in späteren Iterationen weniger Rücksprünge zu frühen Aufgaben durchgeführt werden. Dabei sinkt zudem der Aufwand je Aufgabe und Iteration.

Die Werkzeuglaufzeit bleibt in etwa konstant, wobei der Hauptaufwand durch die Simulation des Designs während der Analyse beigetragen wird. Dieses Verhalten lässt sich auf Basis des Protokolls durch zwei Punkte erklären:

- im Wesentlichen steigt die Qualität des Designs mit jeder Iteration an, wodurch die Wahrscheinlichkeit für eine Iteration zu frühen Aufgaben wie dem Design sinkt, was wiederum den Aufwand für eine Iteration reduziert und

- mit jeder Iteration steigt sowohl das Wissen über das Design als auch über die erfolgsversprechende Einstellung verwendeter Werkzeuge, um die definierten Ziele zu erreichen.

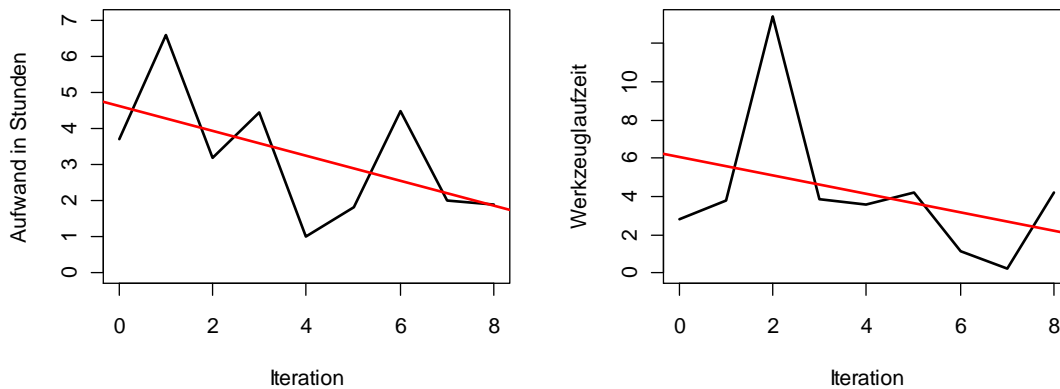


Abbildung 6.1 Verteilung des Aufwands über die verschiedenen Iterationen im Experiment

Aus diesen Ergebnissen können zwar keine verallgemeinernden quantitativen Aussagen über diesen oder ähnliche Entwicklungsprozesse abgeleitet werden. Es zeigt sich aber, welche Konzepte für eine Beschreibung von Entwicklungsprozessen und deren Verhalten notwendig sind. So beschreiben Aufgaben eine bestimmte Arbeit zur Bearbeitung oder Erstellung eines Artefakts. Aufgaben stehen mit anderen Aufgaben, Rollen und Werkzeugen in Beziehung. Zudem wird jedes Prozesselement durch eine Menge von Eigenschaften beschrieben. Für Artefakte sind dieses etwa area, timing und power. Solche Eigenschaften werden gegen bestimmte festgelegte Ziele geprüft. Die Überprüfung von Eigenschaften führt zu einer Entscheidung für das weitere Vorgehen im Entwicklungsprozess. So entscheidet der Entwickler am Ende der zweiten Iteration, dass eine Verbesserung des Designs eine entsprechende Überarbeitung notwendig ist. Der Aufwand und die Werkzeuglaufzeit sowie Eigenschaftswerte verändern sich über die Zeit und sind von zufälligen beziehungsweise unvorhergesehenen Ereignissen abhängig.

Anwendung

In der Anwendung wurde untersucht, ob sich das Anwendungsbeispiel durch das in dieser Arbeit entwickelte Modell und die Methodik ausdrücken lässt und ob es mit Hilfe des Prototyps modellierbar und analysierbar ist.

Auf Grund der beschränkten Datenbasis wurde als Ansatz eine stochastische Modellierung gewählt. Dabei verfügt jede Aufgabe über ein Aufwandsmodell, welches vereinfacht den durchschnittlichen Aufwand einer Aufgabe beschreibt. Die Analyse Aufgabe besitzt ein probabilistisches Entscheidungsmodell, das die folgende Aufgabe bestimmt. Der Entwicklungsprozess wurde 1000 Mal simuliert und liefert einen mittleren Gesamt-

aufwand von etwa 29 Stunden. Dieser Wert entspricht dem gemessenen Aufwand von 29,2 Stunden. Für Metamodell, Methodik und Prototyp lässt sich festhalten:

- Manuelle Protokolle sind für die Erfassung von Prozess- und Produktdaten geeignet. Nach Möglichkeit sind mehrere Projekte zu erfassen und Projekt-datenbanken anzulegen, um eine breitere Datenbasis zu erhalten. Dabei ist die Schwierigkeit eines Projekts am Produkt zu klassifizieren, um eine sinnvolle Datenzuordnung zu erhalten.
- Entwicklungsprozesse sind stochastisch mit Metamodell und Prototyp modellierbar.
- Die Simulation und Analyse des Entwicklungsprozesses liefert ein Bild über mehr oder weniger wahrscheinliches Prozessverhalten.

6.2 High Level Synthesis

Unter High Level Synthesis wird die automatische Transformation einer algorithmischen Verhaltensbeschreibung, zum Beispiel in C/C++ oder SystemC, in eine hardwarenahe Sprache wie VHDL verstanden (Coussy und Morawiec 2008). Ziel der High Level Synthesis ist die drastische Reduzierung des Entwicklungsaufwands und der damit verbundenen Entwicklungskosten für die manuelle Transformation einer Verhaltensbeschreibung in die entsprechende Hardware Beschreibungssprache.

Im Rahmen einer Fallstudie wurden durch Expertenbefragung zwei alternative Entwicklungsprozesse zur Entwicklung eines Filters zur Kantenerkennung für ein Nachtsichtgerät in Automobilen beschrieben. Im bisherigen Entwicklungsprozess wird eine Verhaltensbeschreibung in C/C++ durch einen Entwickler manuell in entsprechenden VHDL Code transformiert. Im alternativen Entwicklungsprozess wird die Methode der High Level Synthese eingesetzt, die eine Verhaltensbeschreibung in SystemC erfordert. Vergleiche dazu auch das Beispiel in Abschnitt 1.5.

Anhand der Methodik wurden auf Basis von Experteninterviews die beiden alternativen Entwicklungsprozesse mit ihren Aufgaben, Artefakten, Rollen und Werkzeugen sowie deren Beziehungen modelliert. Zur Untersuchung des Einflusses der High Level Synthesis (HLS) wurden zudem probabilistische Verhaltensmodelle für Wertänderungen, Aufwände und Entscheidungen modelliert.

Abbildung 6.2 zeigt die beiden für die Untersuchung modellierten Entwicklungsprozesse. Die Aktivität *Design VHDL* des existierenden Entwicklungsprozesses beinhaltet dabei zwei weitere Aufgaben zum Design des VHDL und der dazugehörigen Testbench. Der alternative Entwicklungsprozess besitzt im Vergleich dazu weitere Aufgaben zur Beschreibung der High Level Synthesis (siehe Rahmen). Entsprechend der Einführung der neuen Methode existieren damit auch weitere Rücksprünge zu anderen Aufgaben.

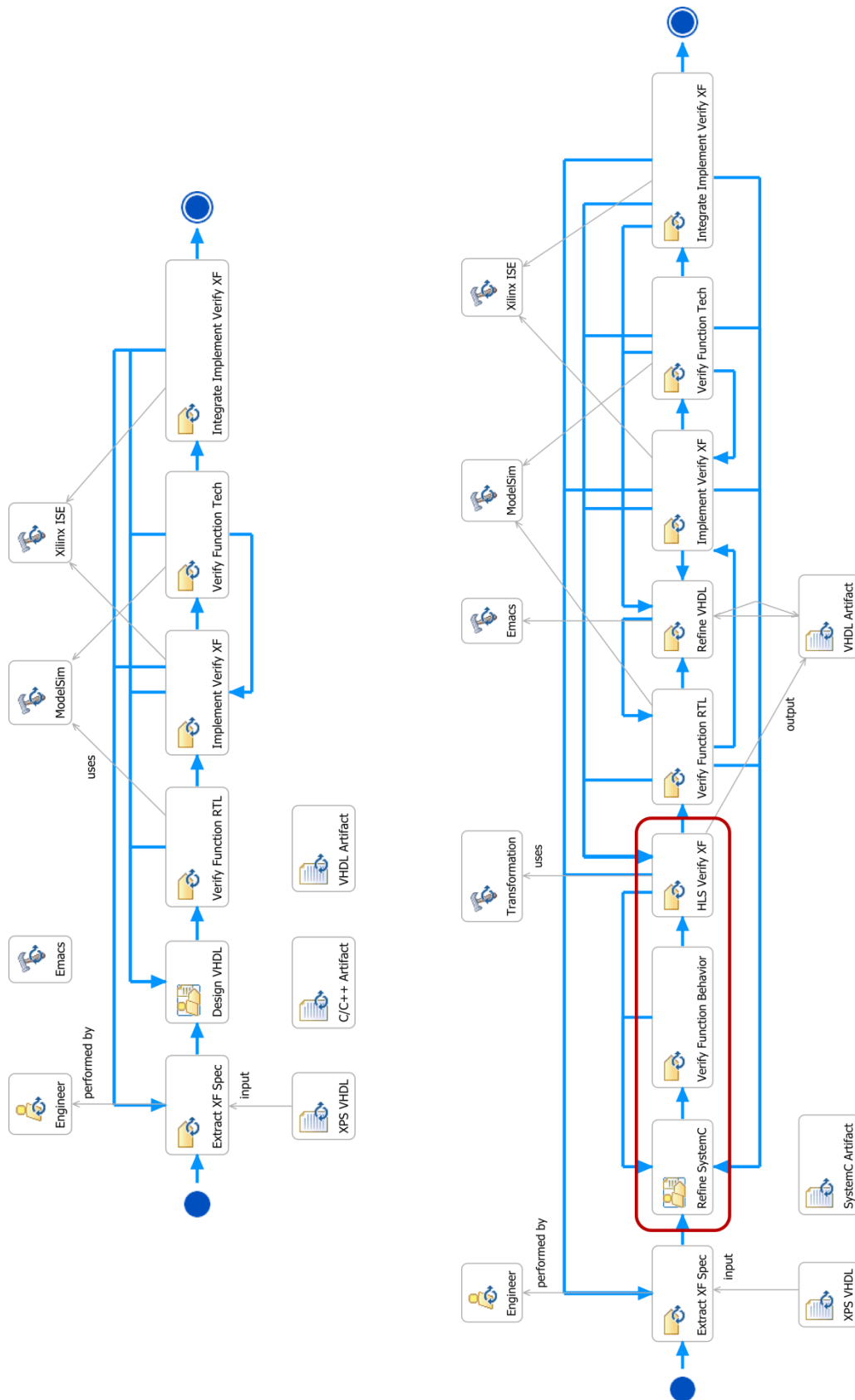


Abbildung 6.2 Modellierter Entwicklungsprozesse zur High Level Synthesis

Im Folgenden wird die Modellierung der Verhaltensmodelle am Beispiel der Aufgabe *Verify Function RTL* illustriert, wie sie auf Grundlage des Experteninterviews abgeleitet wurden. Der Aufwand wird demnach durch die Komplexität des Designs (c), Lerneffekte und unvorhersehbare Ereignisse beeinflusst. Zudem wird eine obere (u) und untere Grenze (l) für den Aufwand einer Iteration (beginnend mit 0) angenommen.

$$e = l \cdot c + \frac{u \cdot c \cdot rnd}{iteration + 1}$$

Dabei liefert rnd eine gleichverteilte Zufallszahl zwischen 0 und 1. Komplexität wird hier als in Stunden Aufwand bewertete Schwierigkeit verstanden. Das Modell wurde mit $l = 0,2$ und $u = 0,9$ als Grenzen für den zu erwartenden Aufwand instanziiert.

Neben der Beschreibung des notwendigen Aufwands wurde ein einfaches boolesches Entscheidungsmodell verwendet, welches die funktionale Korrektheit (fc) des Codes gegen ein Ziel (100%, Toleranz 5%) prüft und daraufhin die folgende Aufgabe wählt.

```
Wenn fc >= 95% dann
  Wähle Implement Verify XF
Sonst
  Wähle Design VHDL
```

Im Weiteren wird die funktionale Korrektheit durch ein probabilistisches Modell beschrieben, welches der vorgelagerten Aufgabe *Design VHDL* zugeordnet ist. So steigt die Wahrscheinlichkeit funktional korrekten Code zu erzeugen mit der Anzahl an Iterationen. Dabei sind durchaus auch Rückschläge zu erwarten.

$$fc = \min \left(rnd + \left(1 - \frac{1}{\exp \left(\frac{iteration}{2 \cdot c} \right)} \right), 1 \right)$$

Mit Hilfe von Simulation und Sensitivitätsanalysen wurden dann die ökonomischen Auswirkungen für unterschiedlich komplexe, das heißt unterschiedlich schwer zu implementierende, Designs untersucht. Dabei besitzt der betrachtete Kantenfilter eine Referenzkomplexität von 1, welche mit einem Experten festgelegt wurde und die Schwierigkeit der Entwicklung mit Blick auf den benötigten Entwicklungsaufwand beschreibt. Somit wird hier die Komplexität anderer Produkte anhand der Schwierigkeit der Implementierung des Kantenfilters durch Expertenbefragung bestimmt.

Abbildung 6.3 zeigt die Verteilung des Aufwands der beiden Entwicklungsprozesse für ein durch den Experten als relativ einfach eingestuftes Design, den Kantenfilter. Es ist deutlich zu erkennen, dass der Entwicklungsprozess mit der neuen Methode vom Aufwand her vor dem existierenden Entwicklungsprozess mit der manuellen Transformation liegt.

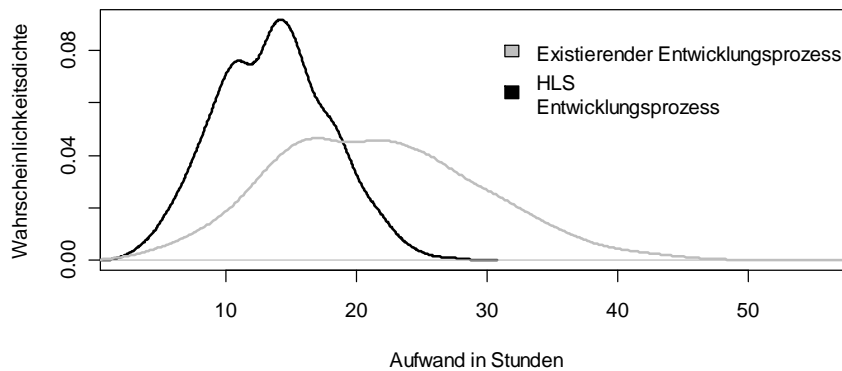


Abbildung 6.3 Verteilung des Aufwands für zwei alternative Entwicklungsprozesse

Für einen quantitativen Vergleich wurden die Konfidenzintervalle der mittleren Differenz beider Verteilungen für unterschiedlich komplexe Artefakte untersucht. Dabei beschreibt eine Komplexität von 1 den betrachteten Kantenfilter und eine Komplexität von 32 einen einfachen digitalen Signalprozessor, nach Einschätzung des Experten. Abbildung 6.4 zeigt die Differenz des Entwicklungsaufwands als Intervall zwischen dem 25%- und 75%-Quantil der Differenz für unterschiedlich komplexe Designs.

Mit einer angenommenen Investition von 100.000 € und einem Stundensatz von 100 € für einen Entwickler würde sich HLS für einfache Designs bei Betrachtung des mittleren 99,8%-Konfidenzintervalls erst nach über 110 durchgeführten Entwicklungen lohnen. Für schwierigere Designs rentiert sich die Methode bereits nach deutlich weniger durchgeführten Entwicklungen. So ergibt sich für eine Komplexität von 32 unter Betrachtung des 25% Quantils als untere Grenze einer breiten 50% Abdeckung für die Rentabilität

$$\frac{100.000 \text{ €}}{232 \text{ h} \cdot 100 \text{ €/h}} = 4,3$$

wobei der Mehrwert der High Level Synthesis in diesem Fall bei ca. 232 Stunden gegenüber der manuellen Entwicklung liegt. Entsprechend lohnt sich die Einführung der neuen Methode bereits nach vier durchgeführten Entwicklungen.

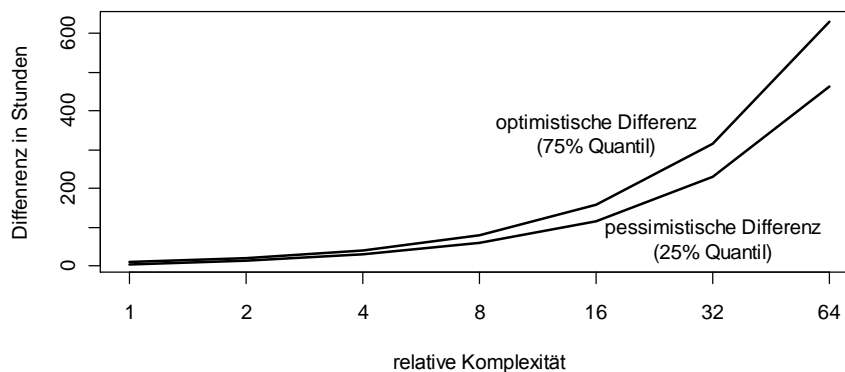


Abbildung 6.4 25% und 75% Quantile der Differenz des Aufwands zweier alternativer Entwicklungsprozesse und unterschiedlich komplexe Designs

Der Hypothesentest gibt Auskunft über die Sicherheit des Unterschieds der beiden betrachteten Entwicklungsprozesse. Für Designs der Komplexität 32 und einem angenommenen α -Fehler von 5% (Wahrscheinlichkeit, dass HLS eingeführt wird, aber keine Verbesserung bringt), wird bei einer Implementierung von HLS mit einer Wahrscheinlichkeit von 83,83% die korrekte Entscheidung getroffen.

Ergebnisse

- In einer ersten Version wurden die alternativen Entwicklungsprozesse in Matlab Simulink/Stateflow modelliert. Dieses erlaubte eine hohe Flexibilität auf Kosten der Modellierung von Entwicklungsprozessen und Übersichtlichkeit (Poppen et al. 2011). Es zeigte sich, dass die entwickelte Werkzeugunterstützung hier einen deutlichen Mehrwert mit Blick auf den Modellierungsaufwand und Beschreibung von Wissen für einen Entwicklungsprozess liefert.
- Der Aufwand für die Erhebung, Modellierung und Validierung der beiden Entwicklungsprozesse lag bei etwa 80 Stunden.
- Die Bildung von Verhaltensmodellen erfordert entsprechendes Wissen beziehungsweise Annahmen über Wirkzusammenhänge, welches mit Experten erhoben werden kann. Dabei beschränken sich die skizzierten Modelle auf Wahrscheinlichkeiten und gehen nicht in Details verwendeter Methoden ein.
- Mit Hilfe der Simulation lässt sich das modellierte Verhalten gut prüfen und auf die angenommenen Wirkzusammenhänge der Wirklichkeit kalibrieren.
- Sensitivitätsanalysen lassen sich schnell durch eine entsprechende Konfiguration der Simulation durchführen.

6.3 Software Inspections

Der Begriff Software Inspections beschreibt Methoden, welche die Qualitätssicherung in der Software-Entwicklung unterstützen sollen. Dabei zielen diese Methoden auf das frühe Erkennen von Fehlern beispielsweise in der Phase der Anforderungsentwicklung. Durch das Erkennen und Beheben entsprechender Fehler sollen teure Iterationen sowie Fehler in späteren Phasen, die auf nicht korrigierte Fehler zurückzuführen sind, vermieden werden. Hierdurch wird auch eine positive Auswirkung auf die Dimensionen Zeit und Kosten erwartet. In diesem Anwendungsbeispiel wird der mögliche Einfluss von Design und Code Inspections auf den Aufwand des Software-Entwicklungsprozess auf Basis von Daten aus der Literatur untersucht.

Briand et al. entwickeln einen Benchmark für unterschiedliche Inspections in der Software-Entwicklung (Briand et al. 1998). Dazu fassen die Autoren eine ganze Reihe von Quellen mit industriellen Daten zum Aufwand und zur Effektivität von entsprechenden

Inspections zusammen. Der Aufwand bezieht sich dabei auf das Finden und Beseitigen eines einzelnen Fehlers (Briand et al. 1997a). Die Effektivität beschreibt die Wahrscheinlichkeit, dass ein entsprechender Fehler gefunden wird. Tabelle 6.3 gibt eine Übersicht der bei Briand et al. gefundenen triangulären Wahrscheinlichkeitsverteilungen für Aufwand und Effektivität.

Tabelle 6.3 Verteilung der Effektivität unterschiedlicher Inspektionsmethoden und den mit ihnen verbundenen Aufwand zur Fehlerbeseitigung nach (Briand et al. 1998)

Methode	Effektivität in Prozent			Aufwand pro Fehler in PH		
	min	Median	max	min	Median	max
Design Inspections	25 %	57 %	84 %	0,58	1,58	2,90
Code Inspections	19 %	57 %	70 %	0,67	1,46	2,70
Test				4,50	6,00	17,00

Das Modell eines einfachen Entwicklungsprozesses für dieses Anwendungsbeispiel besteht aus den Aufgaben Design, Implementierung und Test. Durch die Methoden der Design und Code Inspections wird dieser Entwicklungsprozess um jeweils eine oder in Kombination um zwei Aufgaben erweitert. Neben dem bloßen Finden von Fehlern beinhalten die modellierten Aufgaben auch die Behebung gefundener Fehler. Abbildung 6.5 zeigt das Modell des Entwicklungsprozesses, der über Design und Code Inspections verfügt.

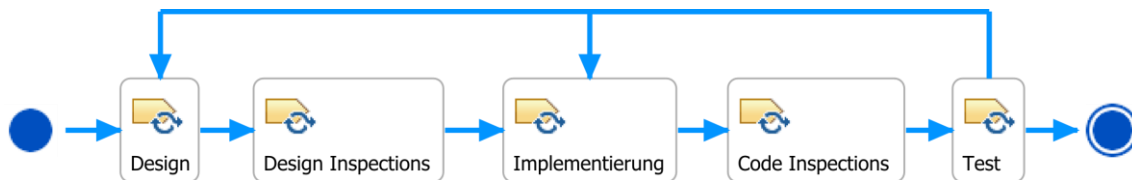


Abbildung 6.5 Software-Entwicklungsprozess mit Design und Code Inspections

Zur Kalibrierung des Entwicklungsaufwandes des Entwicklungsprozesses ohne Inspections werden die durch Boehm et al. festgestellten empirischen Daten des COCOMO Modells verwendet (Boehm et al. 2000c). Für die genannten Aufgaben ergibt sich eine relative Aufwandsverteilung von 30% für Design und Test sowie 40% für die Implementierung. Bezogen auf ein, nach COCOMO, mittelschweres und mittelgroßes Projekt umfassen diese Aufgaben zusammen einen Aufwand von etwa 18293 Stunden (Boehm et al. 2000a).

Entsprechend der genannten Daten nach Briand und Boehm wurden Aufwandmodelle modelliert, die in Abhängigkeit von gefundenen Fehlern den Aufwand einer Aufgabe bei jeder Iteration bestimmen. Die Entwicklung von Fehlern über die Zeit wurde durch

Added Value Modelle beschrieben, welche damit wiederum einen Einfluss auf die Aufwände besitzen.

Abbildung 6.6 gibt einen Überblick über das simulierte Prozessverhalten verschiedener Kombinationen von Inspections. Der Box Plot zeigt für den Entwicklungsprozess ohne Inspections im Mittel den erwarteten kalibrierten Aufwand und eine breite Streuung möglicher Entwicklungsaufwände. Die Einführung von Inspections reduziert hingegen den Entwicklungsaufwand deutlich. Die Kombination von Design Inspections und Code Inspections bietet dabei den größten Mehrwert gegenüber dem Entwicklungsprozess ohne Inspections.

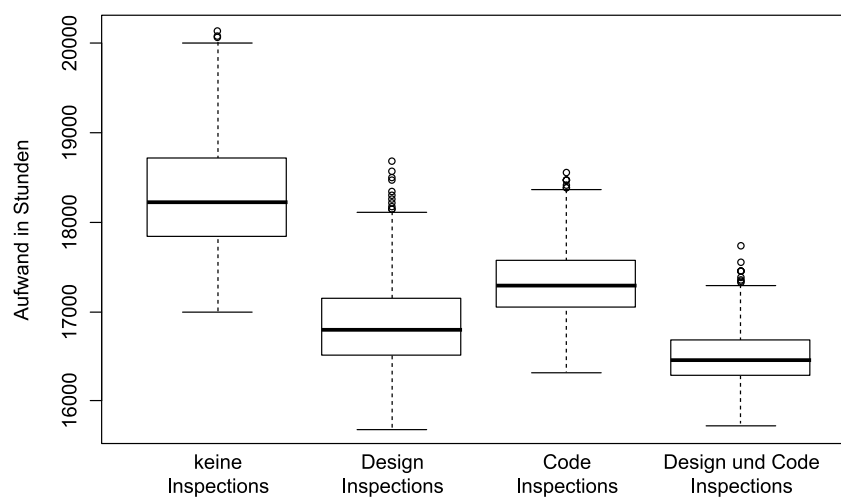


Abbildung 6.6 Vergleich alternativer Software-Entwicklungsprozesse ohne sowie mit Design beziehungsweise Code Inspections

Ergebnisse

- Auch auf Basis recht abstrakter Daten lassen sich tendenzielle Vorteile von alternativen Entwicklungsprozessen untersuchen.
- Einen großen Einfluss auf die dargestellten Ergebnisse besitzt die Anzahl der vorhandenen Fehler, die mittels Sensitivitätsanalysen untersucht werden kann.
- Der Aufwand für die Modellierung lag bei etwa vier Stunden.

Im folgenden Anwendungsbeispiel werden die Auswirkungen von Software Reviews in der Phase der Anforderungsentwicklung an einem Beispiel aus der Praxis untersucht.

6.4 Software Reviews

Reviews werden in der Software-Entwicklung zur Sicherung der Qualität von Artefakten eingesetzt. Durch Reviews sollen während der Entwicklung unbemerkte Fehler gezielt gefunden werden, um diese dann zu korrigieren bevor sie sich in die weitere Entwicklung einschleichen. Hiervon werden sich positive Auswirkungen auf die Produktqualität sowie auf die Entwicklungsdauer und die entsprechenden Kosten versprochen. So wird angenommen, dass die Wahrscheinlichkeit für spätere Iterationen und Korrekturmaßnahmen durch weniger verschleppte Fehler verringert wird.

Im Rahmen dieser Arbeit wurde die entwickelte Methodik für eine Untersuchung bei einem großen Automobilzulieferer für Hardware / Software Komponenten am Praxisbeispiel eingesetzt. Leider können hier keine Daten und detaillierte Ergebnisse veröffentlicht werden. Daher wird im Folgenden eine Charakterisierung der Untersuchung gegeben.

Das Ziel der Untersuchung lag auf der Analyse und Bewertung des Einflusses unterschiedlicher Review Methoden auf die Anzahl gefundener Fehler und den damit einhergehenden Aufwänden. Dabei wurde der Fokus auf die Phase der Anforderungsentwicklung gelegt und die Auswirkungen auf den weiteren Entwicklungsprozess untersucht.

Auf Basis einer repräsentativen Menge von Daten konnten solche Aufgaben identifiziert werden, welche entsprechend gefundener Fehler in den Anforderungen einen Rückschritt zur Phase der Anforderungsentwicklung bedingen. Für diese Aufgaben wurden Added Value und Decision Modelle definiert, um die Anzahl auffindbarer Fehler und die Entscheidung für einen Rücksprung zur Anforderungsentwicklung abzubilden.

Zur Modellierung von Wirkzusammenhängen der verwendeten Review Methode wurden Erkenntnisse aus der Literatur herangezogen und diese mit repräsentativen Daten aus dem Praxisbeispiel verglichen. In der Konsequenz wurden entsprechende Added Value Modelle definiert, welche das Verhalten betrachteter Review Methoden beschreiben. Im Folgenden werden die gesammelten Erkenntnisse zusammengefasst.

In der Literatur wird grundsätzlich ein Zusammenhang zwischen der Anzahl an Mitarbeitern und der Wahrscheinlichkeit einen Fehler zu finden angenommen, vergleiche (Laitenberger et al. 1999) und (Boodoo et al. 2000). Demnach steigt mit der Anzahl eingesetzter Mitarbeiter auch die Wahrscheinlichkeit Fehler in einem Artefakt zu finden. Dieser Zusammenhang flacht mit steigender Anzahl von Mitarbeitern ab.

Nach einer Untersuchung von Laitenberger et al. hängt die Anzahl gefundener Fehler wesentlich vom eingesetzten Vorbereitungsaufwand für Reviews ab, wobei der Umfang des Artefakts keinen wesentlichen Einfluss auf die Anzahl gefundener Fehler zu haben scheint (Laitenberger et al. 1999). Auch Briand et al. stellen einen entsprechenden Zusammenhang zwischen dem Vorbereitungsaufwand und der Effektivität zum Auf-

finden von Fehlern durch Reviews fest (Briand et al. 1997b). Dabei führt ein größerer Vorbereitungsaufwand zu mehr gefundenen Fehlern. Dieser Zusammenhang flacht sich allerdings mit zunehmendem Aufwand durch verschiedene angenommene Effekte ab:

- Ermüdungserscheinungen,
- Mangel an Motivation und
- höhere Komplexität von umfangreicheren Artefakten.

Zudem wird die Anzahl gefundener Fehler bei großen Artefakten durch eine weniger genaue Begutachtung reduziert: mehr Informationen müssen in kürzerer Zeit geprüft werden, vergleiche (Briand et al. 1998), (Christenson et al. 1990). Die optimale Review Rate liegt bei etwa einer Seite pro Stunde (Gilb und Graham 1993), (Rösler 2008). Selbst bei einer optimalen Review Rate werden demnach allerdings nur etwa 50% der Fehler gefunden.

Aus weiteren Daten zu eingesetzten Reviews im Praxisbeispiel konnte bestätigend abgeleitet werden, dass im Wesentlichen der eingesetzte Vorbereitungsaufwand einen Einfluss auf die Anzahl der gefundenen Fehler besitzt. Bei umfangreicheren Anforderungsdokumenten ist demnach ein Rückgang gefundener Fehler zu erwarten. Dabei wurde kein signifikanter Zusammenhang zwischen der Anzahl der eingesetzten Mitarbeiter und gefundener Fehler pro Stunde festgestellt.

Die grundsätzlichen Ergebnisse werden im Folgenden genannt.

Ergebnisse

- Die Methodik lässt sich erfolgreich in der Praxis einsetzen, um den Einfluss von Änderungen in Entwicklungsprozessen zu untersuchen und zu bewerten.
- Der Aufwand für die vorgestellte Modellierung lag bei etwa 20 Stunden. Der wesentliche Aufwand liegt in der Datensammlung und Datenanalyse mit der Modellbildung und Kalibrierung (etwa 60 Stunden). Die automatisierte Simulation und Bewertung hat keinen wesentlichen Einfluss auf den Aufwand.
- Die Methodik erlaubt die Modellierung von Entwicklungsprozessen und unterschiedlichen Szenarien in verschiedenen Abstraktionen. Im vorliegenden Praxisbeispiel wurden probabilistische Modelle wie auch Übergangswahrscheinlichkeiten auf Basis gesammelter Daten verwendet.
- Insgesamt wurden im Praxisbeispiel neue und praxisbezogene Erkenntnisse zu Reviews im Entwicklungsprozess auf Basis der Methodik gesammelt, die auch durch den Experten als interessant und bedeutend beurteilt wurden.

6.5 Lebensmittel Produktentwicklung

Die Produktentwicklung in der Lebensmittelindustrie ist von immer kürzer werdenden Produktlebenszyklen und der Herausforderung laufend neue innovative oder verbesserte Produkte auf den Markt zu bringen geprägt. Dazu sind insbesondere die Entwicklung und die Auswahl einzelner Zutaten sowie die Kombination solcher Zutaten zu Rezepten eine Aufgabe der Forschung und Entwicklung in der Lebensmittelindustrie. So soll etwa der Geschmack eines Produkts einen besonderen Charakter haben, die Textur, die Viskosität oder auch die Haltbarkeit verbessert werden.

Das vorliegende Praxisbeispiel betrachtet einen Entwicklungsprozess zur Entwicklung eines neuen Puddings durch eine Variation von Zutaten, insbesondere der eingesetzten Stärke. Der existierende Entwicklungsprozess wurde mit einer Expertin aus der Anwendungsdomäne erhoben. Dabei wurde das Modell interaktiv modelliert und diskutiert. Nach einer Aufbereitung des Modells wurde ein weiteres Interview mit Experten des Prozesses durchgeführt, auf dessen Basis Aufwände und Übergangswahrscheinlichkeiten modelliert wurden.

Abbildung 6.7 zeigt den modellierten Entwicklungsprozess. Insbesondere sind hierbei die Aufgaben zur Entwicklung des Produkts auf Grundlage einer Basismischung und die komplett neue Entwicklung des Produkts durch eine sogenannte Eigenmischung zu unterscheiden.

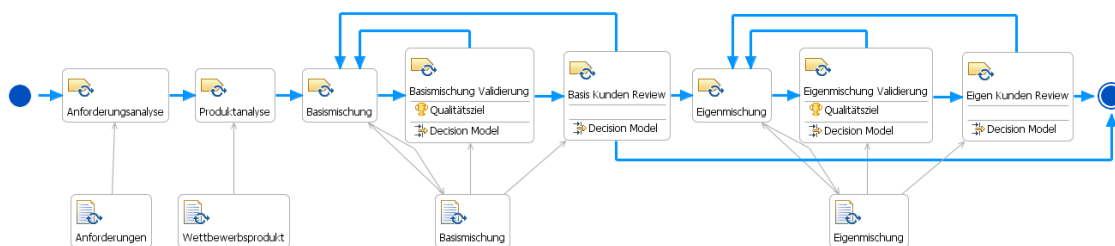


Abbildung 6.7 Entwicklungsprozess aus der Lebensmittelindustrie

Die Aufgaben der Basismischung sind durch eine bereits existierende Rezeptur gekennzeichnet, in der eine oder wenige Zutaten variiert werden, um bestimmte Qualitätseigenschaften zu erreichen. Verläuft die Basismischung mit ihrer Validierung und dem Kunden Review erfolgreich, ist der Entwicklungsprozess beendet und das Rezept schafft seinen Weg in die Produktion.

Ergibt die Basismischung kein zufriedenstellendes Ergebnis, so wird eine Eigenmischung entwickelt. Hierbei werden grundsätzlich mehrere Zutaten variiert, welches eine wesentliche Herausforderung der Entwicklung darstellt. So können verschiedene Wechselwirkungen zwischen den einzelnen Komponenten auftreten – die Entwicklung lässt sich als „Trial and Error“ also Versuch und Irrtum charakterisieren.

Abbildung 6.8 gibt einen Überblick über die mittleren Aufwände zur Entwicklung eines Puddings mit diesem Entwicklungsprozess in Stunden. Offensichtlich tragen die Aufgaben der Eigenmischung maßgeblich zum Gesamtaufwand bei, wenn diese erforderlich sind. Im Gegensatz dazu sind die Aufwände der Aufgaben der Anforderungserhebung und Produktanalyse unkritisch.

Im Folgenden wurde untersucht, welchen Einfluss die Übergangswahrscheinlichkeit vom Kunden Review der Basismischung auf den Gesamtaufwand besitzt und ab wann sich ein Werkzeug lohnen würde, welches auf Basis gegebener Zutatengruppen erfolgversprechende Rezepte generiert.

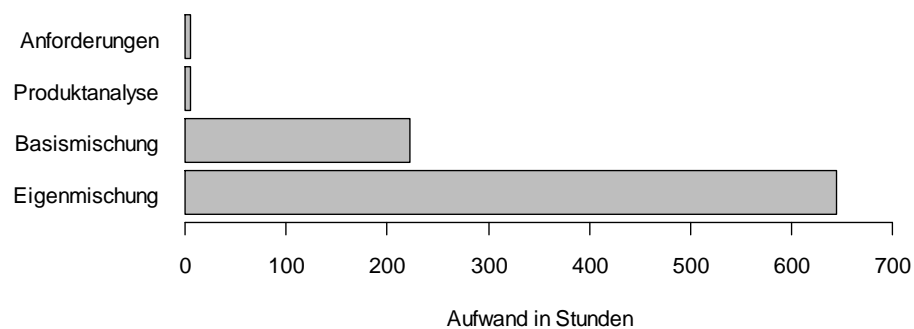


Abbildung 6.8 Mittlere Aufwandsverteilung für unterschiedliche Aufgaben

Abbildung 6.9 veranschaulicht den Einfluss modellierten Rezeptwissens und der Übergangswahrscheinlichkeit vom Review der Basismischung zu den Aufgaben der Eigenmischung. Aus der Abbildung lässt sich erkennen, dass sich der Gesamtaufwand insbesondere für höhere Übergangswahrscheinlichkeiten deutlicher reduziert.

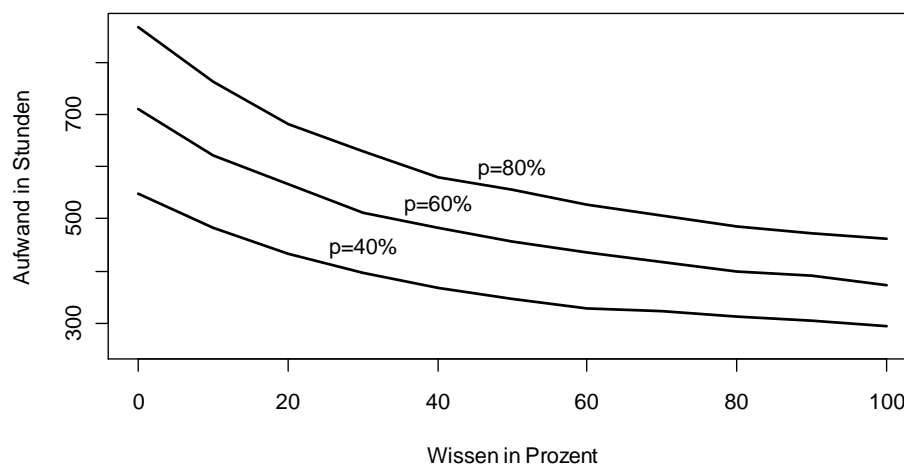


Abbildung 6.9 Einfluss von modellierten Rezeptwissen und Übergangswahrscheinlichkeiten

Auf Basis des Experteninterviews wird eine Investition von 75.000 Euro und durchschnittliche Kosten von 75 Euro pro Stunde angenommen. Zudem wird Rezeptwissen kontinuierlich über verschiedene Projekte aufgebaut und bedarf nach Expertenschätz-

ung mindestens ein Jahr. Für eine wahrscheinliche Übergangswahrscheinlichkeit von 60% zeigt Tabelle 6.4 die Anzahl durchzuführender Entwicklungen in Abhängigkeit von aufgebautem Wissen für die pessimistische Grenze des 99,8%-Konfidenzintervalls des Mittelwertes.

Tabelle 6.4 Durchzuführende Entwicklungen bis zum Mehrwert eines neuen Werkzeugs

Rezept-Wissen	10%	20%	30%	40%	50%
Durchzuführende Entwicklungen	15,29	7,86	5,62	4,66	4,16

Ergebnisse

- Auch wenn in der Praxis ständig nach einem bestimmten Vorgehen gearbeitet wird, so war dieses im Anwendungskontext doch nicht explizit beschrieben oder dokumentiert. Die interaktive Modellierung im Expertengespräch war hier sehr gut zur Erfassung des Entwicklungsprozesses geeignet. Der Aufwand für diese Modellierung lag bei etwa 4 Stunden.
- Der Aufwand für ein weiteres Interview zu Aufwänden, Übergangswahrscheinlichkeiten und Verhalten, deren Modellierung sowie die Validierung mit der Expertin lag bei etwa 8 Stunden.
- Die mit der Methodik erreichte Modellierung sowie der mögliche Einfluss von Änderungen im Entwicklungsprozess stellt eine erste Basis für weitere Untersuchungen zur Verfügung und wurde durch die Expertin als äußerst wertvoll bewertet.
- Die Experten konnten für die Notwendigkeit einer Dokumentation von Entwicklungsprozessen in der Lebensmittelindustrie sowie der Sammlung und Analyse von Prozess- und Produktdaten erfolgreich sensibilisiert werden.

6.6 Zusammenfassung und Bewertung

In den zurückliegenden Abschnitten wurden verschiedene Anwendungsbeispiele der Methodik aus unterschiedlichen Domänen beschrieben. Die Beispiele zeigen die grundsätzliche Anwendbarkeit des entwickelten Ansatzes. Mit Blick auf die Untersuchung von Reviews in der Phase der Anforderungsentwicklung bei einem großen Automobilzulieferer sowie in der Domäne der Lebensmittelindustrie konnte die entwickelte Methodik im Praxiskontext eingesetzt werden.

Qualitativ lässt sich festhalten, dass durch die Anwendung der Methodik mit der Modellierung und Diskussion von Entwicklungsprozessen ein deutlicher Erkenntnisgewinn zu erwarten ist. So wird in der Praxis zwar nach einem bestimmten Vorgehen gearbeitet, dieses ist aber nicht immer dokumentiert. Selbst, wenn eine grundsätzliche

Dokumentation des Entwicklungsprozesses mit Methodenbeschreibungen und Guidelines existiert, so fehlt es dort doch noch an einer Beschreibung von Wirkzusammenhängen. Neue Erkenntnisse beziehen sich insbesondere auf die Beziehungen zwischen Prozesselementen sowie auf die erwarteten Wirkzusammenhänge. Der entwickelte Ansatz leistet hierzu einen Beitrag.

Weitere Erkenntnisse über einen modellierten Entwicklungsprozess werden durch die Datensammlung, Datenanalyse sowie Modellbildung gewonnen. So wurden etwa im Praxisbeispiel der Software Reviews initial vermutete Wirkzusammenhänge auf Grundlage einer breiten Datenbasis untersucht und modelliert. Dank der Simulation können verschiedene Szenarien des Prozessverhaltens auf Basis von modellierten Eigenschaften und Verhaltensmodellen studiert werden. Die Ergebnisse der Einflussanalyse und Einflussbewertung liefern auch hier einen Mehrwert zum Verständnis von Entwicklungsprozessen, wie auch die Praxisbeispiele gezeigt haben.

Reflektion der Anforderungen

Im Folgenden werden die formulierten Anforderungen auf ihre Erfüllung durch den vorgestellten Ansatz kritisch diskutiert und anschließend die Erreichung der aufgestellten Ziele geprüft.

Anforderung 1. Kennzahlen. Der mit dieser Arbeit vorgestellte Ansatz erlaubt die Bewertung von Methodenänderungen in Entwicklungsprozessen über bekannte Kennzahlen der Investitionsrechnung. Dabei kann die monetäre Bewertung über den Aufwand oder direkt über modellierte Kosten erfolgen. Neben den Kennzahlen zu Rentabilität und Amortisation werden Methoden wie Konfidenzintervall, Abdeckungsintervall und Hypothesentest eingeführt, welche eine weitere Bewertung der Differenz alternativer Entwicklungsprozesse auf Basis statistischer Analysen erlauben. Die betrachteten Kennzahlen sind nicht notwendigerweise vollständig, so dass eine Erweiterung im Anwendungskontext denkbar ist.

Anforderung 2. Unsicherheit. Der Aspekt der Unsicherheit ist zentral für die Untersuchung von Entwicklungsprozessen. Der entwickelte Ansatz erlaubt die Beschreibung von angenommenen Wirkzusammenhängen und darauf basierend die Modellierung von Verhaltensmodellen für Aufwände, Kosten, Werteänderungen auf Basis von Eigenschaften beteiligter Prozesselemente sowie Entscheidungsverhalten. Dabei wird die probabilistische Abbildung solcher Verhaltensmodelle über parametrisierte Wahrscheinlichkeitsfunktionen unterstützt.

Anforderung 3. Analyse. Die entwickelte Methodik sowie die prototypische Umsetzung erlauben die Analyse und Aufbereitung von Daten, die während der Simulation von Entwicklungsprozessen gesammelt werden. Dabei kommen grundlegende Methoden der Statistik für eine numerische Darstellung zum Einsatz. Darüber hinaus unterstützt der

Prototyp die Aufbereitung entsprechender Ergebnisse durch Histogramme, Dichtediagramme sowie Box Plots. Grundsätzlich sind hier Erweiterungen von Analysen denkbar, die im Prototyp durch entsprechende Schnittstellen vorgenommen werden können.

Anforderung 4. Modellierung. Für die Modellierung von Methodenwissen, Entwicklungsprozessen und Wirkzusammenhängen wurde ein konzeptionelles Metamodell entwickelt, welches auf dem Modellierungsstandard SPEM 2.0 basiert. Die prototypische Umsetzung erlaubt dementsprechend die Beschreibung von Methoden, Prozessen und Verhaltensmodellen, welche direkt für die Simulation modellierter Entwicklungsprozesse verwendet werden.

Anforderung 5. Durchgängigkeit. Im Stand der Technik wurde die Trennung zwischen der Modellierung von Methodenwissen und Entwicklungsprozessen und der Modellierung von Simulationsmodellen deutlich. Der in dieser Arbeit entwickelte Ansatz erlaubt die Beschreibung entsprechenden Wissens und die Erweiterung um entsprechende Modelle, so dass eine automatische Überführung in ausführbare Simulationsmodelle ermöglicht wird. Diese Übersetzung ist für den Anwender der prototypischen Umsetzung transparent. Somit wird eine Durchgängigkeit von der Modellierung bis hin zur Einflussanalyse und Einflussbewertung erreicht.

Anforderung 6. Wiederverwendung. Durch die Methodik und das beschriebene Metamodell ist grundsätzlich eine Wiederverwendung von Methodenwissen und Wissen über Entwicklungsprozesse möglich. Dabei können definierte Prozesselemente mit ihren modellierten Verhaltensmodellen in Modellen von Entwicklungsprozessen wiederverwendet werden. In der prototypischen Umsetzung sind Verhaltensmodelle Teile von Aufgaben und somit von diesen abhängig. Eine erweiterte Wiederverwendung über verschiedene Aufgaben hinweg würde hier eine noch generischere Definition von Verhaltensmodellen erfordern. In den untersuchten Anwendungsbeispielen war dieses allerdings nicht notwendig, so dass die Anforderung als weitestgehend erfüllt bewertet werden kann.

Anforderung 7. Wirkzusammenhänge. Die Methodik erlaubt die Beschreibung von Wirkzusammenhängen mit dem Ziel der Kommunikation sowie die Modellierung solcher Wirkzusammenhänge als Verhaltensmodelle. Weder die Methodik noch das Metamodell begrenzt die Art eines entsprechenden Verhaltensmodells. Neben probabilistischen Verhaltensmodellen, welche auf Basis von Eigenschaften von Prozesselementen parametrisiert werden können, werden auch Übergangswahrscheinlichkeiten zwischen Aufgaben unterstützt, wie sie etwa in Markov-Ketten verwendet werden. Die prototypische Umsetzung unterstützt die Modellierung von mathematischen Verhaltensmodellen verschiedener Abstraktionsebenen. Definierte Schnittstellen erlauben hier

zudem die Einbindung anderer Arten von Verhaltensmodellen beispielsweise auf Basis von Zustandsautomaten.

Anforderung 8. Methodik. Die in dieser Arbeit entwickelte Methodik leitet den Anwender der Methodik von der Auswahl von Zielen, die mit der Einflussanalyse und Einflussbewertung verbunden sind, über die Modellierung von Entwicklungsprozessen, Wirkzusammenhängen und Verhaltensmodellen bis hin zur Auswertung simulierter Entwicklungsprozesse. Somit wird ein umfassendes Instrument für die Untersuchung von Entwicklungsprozessen und damit zur Einflussanalyse und Einflussbewertung von Methodenänderungen zur Verfügung gestellt, welches entsprechend für den spezifischen Anwendungskontext zu instanziiieren ist. Dabei ist die Methodik in weiteren Praxisbeispielen zu untersuchen und gegebenenfalls weiter zu entwickeln.

Anforderung 9. Verfeinerung. Der vorgestellte Ansatz erlaubt grundsätzlich eine Abstraktion sowie eine schrittweise Verfeinerung von Modellen zu Prozesselementen sowie Wirkzusammenhängen beziehungsweise Verhaltensmodellen. So liegt der Fokus der Modellierung zunächst auf kritischen Prozesselementen und Wirkzusammenhängen, die detailliert beschrieben werden können. Andere Zusammenhänge können dabei etwa zunächst mit einfachen Übergangswahrscheinlichkeiten beschrieben werden und sukzessive verfeinert werden, wenn dieses notwendig erscheint. Die Auswahl, welche Elemente wie detailliert beschrieben werden, hängt dabei vom Anwendungskontext ab.

Anforderung 10. Werkzeugunterstützung. Mit der prototypischen Umsetzung wird eine entsprechende Werkzeugunterstützung für die Anwendung der entwickelten Methodik geliefert. Der Prototyp erlaubt die Modellierung von Methodenwissen und Entwicklungsprozessen sowie Wirkzusammenhängen und Verhaltensmodellen. Die Simulation von Entwicklungsprozessen erfolgt für den Anwender des Prototyps transparent. Hier sind keine manuellen Transformationen durchzuführen, um ein ausführbares Modell des Entwicklungsprozesses zu erstellen. Der Prototyp unterstützt die Simulation sowie die Einflussanalyse und Bewertung. Da weitere Arten von Modellierungen und Analysen denkbar sind, wurden Schnittstellen definiert, welche eine entsprechende Erweiterung des Prototyps an verschiedenen Stellen erlauben.

Als Ergebnis der Diskussion der aufgestellten Anforderungen kann festgehalten werden, dass diese zum Großteil durch den vorgestellten Ansatz erfüllt werden konnten. Fehlt es in einem speziellen Anwendungskontext dennoch an Kennzahlen, Analysen oder Werkzeugen in der prototypischen Umsetzung, können diese durch definierte Schnittstellen erweitert werden. Die Methodik sowie das vorgestellte Metamodell machen hier keine wesentlichen Einschränkungen und erlauben die Anpassbarkeit auf verschiedene Untersuchungskontexte.

Reflektion der Ziele

Im Folgenden werden nun die eingangs aufgestellten Ziele reflektiert.

Ziel 1. Stand der Technik. In Kapitel 2 wurden Methoden und Ansätze vorgestellt, die für eine Einflussbewertung von Methodenänderungen in Entwicklungsprozessen herangezogen werden können. Methoden der Leistungs- und Nutzenbewertung können bei Investitionsvorhaben unterstützen, abstrahieren allerdings von den zugrunde liegenden Entwicklungsprozessen. Um eine Beschreibung von Entwicklungsprozessen und Methodenwissen zu erlauben, wurden Methoden zur Modellierung von Projekten, Prozessen und Methoden untersucht. Mit Blick auf eine Analyse von Entwicklungsprozessen wurde der Zusammenhang von analytischen Methoden und einer Beschreibung von Wirkzusammenhängen diskutiert, bevor stochastische und simulative Ansätze zur Analyse von Prozessen vorgestellt wurden. Der Stand der Technik wurde damit umfassend für den anvisierten Lösungsansatz untersucht.

Ziel 2. Methodik. In Kapitel 3 wurde eine Methodik für die Einflussanalyse von Methodenänderungen in Entwicklungsprozessen entwickelt. Die Methodik leitet von der Definition von Zielen an eine Einflussanalyse, über die Modellierung von Entwicklungsprozessen und Wirkzusammenhängen bis hin zu einer Bewertung des Mehrwerts einer entsprechenden Prozessänderung. Die Methodik wurde dabei an einem beispielhaften Entwicklungsprozess illustriert und in verschiedenen Anwendungsbeispielen in Kapitel 6 erprobt. Zudem wurde die Idee der Methodik in mehreren Konferenzbeiträgen erfolgreich veröffentlicht und der wissenschaftlichen Gemeinschaft vorgestellt (Koppe et al. 2010c), (Koppe et al. 2011a), (Koppe et al. 2011b), (Poppen et al. 2011).

Ziel 3. Metamodell. Für die Beschreibung von Methodenwissen, Entwicklungsprozessen und Wirkzusammenhängen wurde auf Basis des Software & Systems Process Engineering Meta-Models (SPEM 2.0), einem Standard der Object Management Group, ein entsprechendes Metamodell spezifiziert (Kapitel 4). Das Metamodell erlaubt damit die formalisierte Beschreibung von Entwicklungsprozessen und Verhaltensmodellen, welche für eine Simulation von Entwicklungsprozessen verwendet werden können. Die Auswahl von SPEM wurde auf Basis des erhobenen Stands der Technik zur Beschreibung von Entwicklungsprozessen und der Erfüllung gesetzter Anforderungen getroffen (Koppe et al. 2010b).

Ziel 4. Prototyp. Es wurde eine prototypische Umsetzung auf Basis des Metamodells implementiert, welche bei der Anwendung der Methodik unterstützt (Kapitel 5). Die prototypische Umsetzung unterstützt die Modellierung von Methodenwissen sowie die Beschreibung von Entwicklungsprozessen, Beziehungen zwischen Prozesselementen, Wirkzusammenhängen und Verhaltensmodellen. Neben der der Modellierung wird die

Simulation von Entwicklungsprozessen sowie die Einflussanalyse und Einflussbewertung unterstützt (Koppe et al. 2010a).

Ziel 5. Evaluierung. Für die Evaluierung des Ansatzes wurden mehrere Anwendungsbeispiele aus unterschiedlichen Anwendungsdomänen herangezogen (Kapitel 6). Diese Anwendungsbeispiele illustrieren die grundsätzliche Anwendbarkeit des entwickelten Ansatzes auf verschiedene Probleme der Einflussanalyse. Insgesamt liefert bereits die Modellierung von Entwicklungsprozessen mit ihren Wirkzusammenhängen einen Erkenntnisgewinn. Darüber hinaus sind die gewählten Analysen für eine Bewertung des (monetären) Mehrwerts einer Änderung in Entwicklungsprozessen hilfreich.

Die Evaluierung zeigt also im Wesentlichen die Erreichung der aufgestellten Anforderungen und Ziele und damit die Anwendbarkeit des vorgestellten Ansatzes mit einem entsprechenden Erkenntnisgewinn auch in Praxisbeispielen. Der folgende Abschnitt 6.7 fasst dazu grundsätzliche Anwendungsbedingungen der Methodik zusammen.

Die Frage, ob eine angenommene Bewertung des Mehrwert neuer Methoden, sich in der Wirklichkeit genauso zeigt, wie sie durch eine Simulation vorhergesagt wurde, kann hier allerdings nicht gezeigt werden. Eine entsprechende Evaluierung setzt zum Einen eine anwendungsspezifische und adäquate Modellierung des Entwicklungsprozesses und zum Anderen die mehrfache Umsetzung einer entsprechenden Prozessänderung in der Praxis voraus. Abschließend kann eine Bewertung also nur erfolgen, wenn auch empirische Daten für eine Änderung im Anwendungskontext zur Verfügung stehen. Dies liegt allerdings außerhalb des Fokus dieser Arbeit.

6.7 Anwendungsbedingungen der Methodik

Aus den betrachteten Anwendungsbeispielen und den Ergebnissen der Evaluierung lassen sich verschiedene Klassen für die Anwendbarkeit des in dieser Arbeit entwickelten Ansatzes anhand des Charakters beziehungsweise der Reife von Entwicklungsprozessen ableiten, welche im Folgenden diskutiert werden.

Eine Basis dazu bilden Hinweise, für die Durchführung von Simulationsstudien (Banks et al. 2009). So ist der Einsatz von Simulation nicht erforderlich, wenn Einflussanalysen einfach auch manuell durchgeführt werden können, die Auswirkungen einer Änderung im Entwicklungsprozess oder das zu erwartende Ergebnis der Einflussanalyse also offensichtlich sind. Weiterhin erfordert der Aufbau von Wirkzusammenhängen die Analyse gesammelter Daten oder die Extraktion von Erfahrungen hin zu Verhaltensmodellen. Stehen entsprechende Daten für den Aufbau von Modellen und deren Validierung nicht zur Verfügung, können lediglich mögliche Tendenzen einer Methodenänderung aufgezeigt werden. Schließlich schränkt die Komplexität des betrachteten Systems beziehungsweise seiner Teile die Modellierbarkeit und Anwendbarkeit von

Simulation ein. Lässt sich das Verhalten eines Entwicklungsprozesses nicht mit absehbarem Aufwand geeignet abstrahieren und modellieren, können keine sinnvollen Einflussanalysen durchgeführt werden.

Ergänzend zu diesen eher allgemeinen Hinweisen können nun Kriterien für die Anwendbarkeit der entwickelten Methodik formuliert werden (Tabelle 6.5). Die genannten Kriterien werden bestimmten Anwendungsklassen zugeordnet, welche dabei auch in Beziehung zu Reifegradmodellen wie etwa dem Capability Maturity Model Integration (CMMI) for Development (CMU/SEI 2006), ohne jedoch eine entsprechende Zertifizierung für die Anwendung der Methodik vorauszusetzen.

Tabelle 6.5 Kriterien zur Anwendung der Methodik

Nr	Kriterium
K01	Die Aufgaben im Entwicklungsprozess sind klar unterscheidbar, so dass es nicht zu einer Überschneidung von Tätigkeiten kommt. Damit sind Aufgaben definiert und beschreiben für jede Durchführung vergleichbare Inhalte.
K02	Der Aufwand zur Durchführung einer Aufgabe im Entwicklungsprozess kann bestimmt werden. Neben dem mittleren Aufwand ist auch die Varianz des Aufwands von Interesse.
K03	Für den Ablauf von Aufgaben können Übergangswahrscheinlichkeiten auf Basis von Daten und Erfahrungen abgeleitet werden.
K04	Die zur Durchführung einer Aufgabe beziehungsweise Anwendung einer Methode erforderlichen Prozesselemente und deren Ergebnisse sind definiert – insbesondere sind dieses Design Artefakte.
K05	Die wesentlichen Einflussfaktoren (Eigenschaften) von Eingaben einer Aufgabe beziehungsweise Methode sowie die durch eine Methode beeinflussten Eigenschaften von Ausgaben sind klar bestimmbar.
K06	Das Verhalten einer Methode unterliegt einer durch Wirkzusammenhänge beschreibbaren Gesetzmäßigkeit, bei der bestimmte eingehende Eigenschaften in bestimmte ausgehende Eigenschaften transformiert werden.
K07	Im Anwendungskontext kann eine Vergleichbarkeit von Design Artefakten in Bezug auf die Anwendung einer bestimmten Methode erreicht werden. Hier gehen die betrachteten Eigenschaften und die akzeptierte Varianz modellierten Verhaltens ein.

Unstrukturierte Entwicklungsprozesse

Entwicklungen bei denen Aufgaben nicht differenziert werden können (K01) und auch der Aufwand (K02) sowie der Ablauf von Aufgaben (K03) nicht bestimmt werden können, werden hier als unstrukturiert bezeichnet und sind für Einflussanalysen nicht

geeignet. Dieses entspricht einer initialen Reife von Entwicklungsprozessen, bei der Entwicklungen ad-hoc oder chaotisch durchgeführt werden und der Erfolg eher zufällig und nicht wiederholbar ist. Häufig werden bei solchen Entwicklungen die geplante Entwicklungsdauer und das geplante Budget überschritten (CMU/SEI 2006).

Im Rahmen dieser Arbeit wurden Interviews mit Teilnehmern einer studentischen Projektgruppe aus der Informatik an der Universität Oldenburg durchgeführt. Hier war es nicht möglich, Aufgaben, Abläufe, Aufwände und Design Artefakte sinnvoll zu bestimmen, um darauf aufbauend Einflussanalysen durchzuführen. Die Methodik kann in solchen Fällen also keinen Beitrag zur Einflussanalyse leisten.

Strukturierte Entwicklungsprozesse

Mit der Erfüllung der ersten drei Kriterien lassen sich Entwicklungsprozesse als strukturiert bezeichnen. Strukturierte Entwicklungsprozesse erlauben ein erstes Verstehen sowie den Aufbau von Prozess- und Methodenwissen. Mit Blick auf CMMI handelt es sich mindestens um eine Reife der zweiten Stufe „Managed“ bei der Entwicklungen auf Basis einfacher Prozessbeschreibungen durchgeführt werden und Meilensteine definiert sind. Darüber hinaus sind die für die Anwendung einer Methode erforderlichen Eingaben sowie deren Ergebnisse definiert (K04). Bei einer weiteren Reife werden bereits explizit Ziele, Eingaben, Ausgaben, Methoden und Rollen beschrieben. Ergänzend dazu sind die Zusammenhänge zwischen Aufgaben bekannt und auch existieren Produkt- und Prozesskennzahlen (CMU/SEI 2006). Für noch reifere Entwicklungsprozesse werden bereits regelmäßig quantitative Daten erfasst, welche für Einflussanalysen verwendet werden können.

Entsprechend bildet diese Klasse die Basis für die Anwendung der in dieser Arbeit entwickelten Methodik. Die zu Beginn dieses Kapitels vorgestellten Anwendungsbeispiele verfügen alle über ein strukturiertes Vorgehen. Im Folgenden lassen sich bestimmte Fälle strukturierter Entwicklungsprozesse unterscheiden.

Strukturierte Entwicklungsprozesse und einfaches Erfahrungswissen

Existiert einfaches Erfahrungswissen zu Aufwänden und Abläufen zwischen Aufgaben im betrachteten Entwicklungsprozess, so lassen sich Einflussbewertungen bereits über eine stochastische Modellierung nach dem Ansatz der **Markov-Ketten** vornehmen. Eine entsprechende Anwendung ist bei Entwicklungsprozessen vorzunehmen, bei denen Einflussfaktoren und Eigenschaften von Prozesselementen beziehungsweise Wirkzusammenhänge nicht beschrieben werden können (K05).

Dabei werden einfache Übergangswahrscheinlichkeiten und Aufwände je Iteration bestimmt. Bei der klassischen Anwendung von Markov-Ketten kann zur Ermittlung des Einflusses einer Änderung eine analytische Lösung oder ein simulativer Ansatz

verwendet werden (Abschnitt 2.7). Es ist klar, dass bei einer entsprechenden Modellierung beobachtbares Verhalten abgebildet wird aber noch keine erklärenden Wirkzusammenhänge beschrieben werden.

Eine entsprechende Modellierung wurde im ersten Anwendungsbeispiel „Gammatone Filterbank“ (Abschnitt 6.1) sowie im Praxisbeispiel „Lebensmittel Produktentwicklung“ (Abschnitt 6.5) vorgenommen.

Strukturierte Entwicklungsprozesse, erweitertes Erfahrungswissen und Daten

Steht ein erweitertes Erfahrungswissen über mehrere Experten eines Entwicklungsprozesses oder eine konkrete Datenbasis für einen Entwicklungsprozess zur Verfügung, so lassen sich Abläufe auf Basis von Entscheidungsmodellen beschreiben. Weiterhin können Wirkzusammenhänge zwischen Prozesselementen und die Änderung von Eigenschaften von Design Artefakten modelliert werden. Hierbei sind insbesondere Eigenschaften von Interesse, welche den Aufwand von Aufgaben beeinflussen oder die Qualität von Design Artefakten beschreiben. Um entsprechende **Wirkzusammenhänge** zu beschreiben und damit Abhängigkeiten zwischen Eigenschaften zu quantifizieren, ist eine gewisse Gesetzmäßigkeit (K06) im Verhalten eingesetzter Methoden notwendig.

- Das Entscheidungsverhalten für folgende Aufgaben muss sich aus den betrachteten Eigenschaften von Design Artefakten und den an sie gestellten Anforderungen beziehungsweise Zielen ergeben.
- Die Veränderung von Eigenschaften durch eine Methode muss nachvollziehbar und damit vorhersagbar sein. Dabei wird eine Varianz von Eigenschaftswerten in anwendungsspezifischen Grenzen explizit gestattet.
- Veränderungen von Eigenschaften basieren auf anderen Eigenschaften, die als Eingabe in ein Modell verstanden werden können. Für entsprechende eingehende Eigenschaftswerte ist das Verhalten einer Methode definierbar.

Im Anwendungsbeispiel „Software Inspections“ (Abschnitt 6.3) wurde die Produktqualität als die Anzahl an Fehlern modelliert, welche einen Einfluss auf notwendige Iterationen und damit auf den Aufwand besitzt. Dabei wurde die Anzahl der Fehler über trianguläre Wahrscheinlichkeitsverteilungen der Effektivität verschiedener Methoden bestimmt. Im Praxisbeispiel „Software Reviews“ (Abschnitt 6.4) wurde eine ähnliche Modellierung über die Anzahl der Fehler vorgenommen. Bei diesen Beispielen bedingt die Anzahl gefundener Fehler eine Überarbeitung der betrachteten Design Artefakte, durch welche die Qualität (Reduzierung der Fehler) steigt, allerdings auch Aufwand für Überarbeitungen investiert werden muss.

Weitere Anwendungsbeispiele sind etwa

- **Modellgetriebene Entwicklung:** durch automatische Generation von Quellcode aus UML Diagrammen oder auch aus Ecore Modellen soll der Aufwand für die manuelle Implementierung reduziert werden. Dazu ist Aufwand zur Definition von Modellen notwendig. Zu beachten sind: Kosten für Anschaffung und Schulungen, Aufwand notwendiger Aufgaben, Anzahl der Abhängigkeiten zwischen Konzepten, Änderbarkeit von Konzepten und Auswirkungen auf bereits generierten Code, Anzahl generierter Klassen, Wartbarkeit generierten Codes, Wiederverwendbarkeit von Komponenten.
- **Qualitätssicherung:** eine neue Methode soll frühzeitig potentielle Fehler im Design finden, um teure Überarbeitungen zu vermeiden. Zu beachten sind: Kosten für Anschaffung und Schulung, aktuell vorhandene und gefundene Fehler, gefundene Fehler durch neue Methode, reduzierter Aufwand je Fehler und Aufgabe. Weiter können Fehlerarten und deren Auswirkungen / Kritikalität unterschieden werden.
- **CAD / CAM:** CAD Modelle werden mit Technologie-Informationen angereichert, um entsprechende Bauteile automatisiert zu fertigen und dadurch den Entwicklungsaufwand zu reduzieren. Zu beachten sind: Kosten für Anschaffung und Schulung, Aufwand für Annotation von Technologie-Informationen, Wiederverwendbarkeit von (Teil-) Modellen, Anzahl der Fehler bei manueller NC Programmierung, Anzahl der Fehler bei generierten NC Programm, Aufwand zur Behebung von Fehlern (Kollisionen, falsche Werkzeugbewegungen, Programmierfehler).

Strukturierte Entwicklungsprozesse mit hoch variablen Abläufen

Entwicklungsprozesse mit hoch variablen beziehungsweise unsicheren Abläufen sind durch eine „zufällige“ Auswahl der folgenden Aufgaben gekennzeichnet. Das Entscheidungsverhalten, um von einer Aufgabe zu einer anderen zu gehen, kann also nicht mit einem gegebenen Aufwand für die Erfassung von Daten und die Modellierung des betrachteten Entwicklungsprozesses beschrieben werden – beeinflussende Eigenschaften oder Störgrößen wurden nicht erkannt.

Um Einflussanalysen durchführen zu können bedarf es einer **Abstraktion**: die Abläufe im Entwicklungsprozess werden durch Übergangswahrscheinlichkeiten beschrieben. Damit wird im Wesentlichen die beobachtbare Häufigkeit für die Auswahl eines bestimmten Ablaufs beschrieben. Dieses steht im Gegensatz zur Erklärung des Prozessverhaltens über Wirkzusammenhänge, erlaubt aber dennoch tendenzielle Aussagen über Änderungen im Entwicklungsprozess.

Strukturierte Entwicklungsprozesse mit hoch variablen Eigenschaften

Es existieren Methoden, deren Auswirkungen auf Eigenschaften von Design Artefakten nur begrenzt prognostiziert werden können und daher nur schwierig oder nicht als Wirkzusammenhänge modelliert werden können. Dabei können zwei Fälle unterschieden werden

- die Methode liefert in gegebener Zeit für ein gegebenes Design mit bestimmten Eigenschaften kein Ergebnis oder
- die aus der Anwendung der Methode resultierenden Eigenschaften assoziierter Design Artefakte sind hoch variabel.

Ein solches Verhalten lässt sich bei insbesondere bei Methoden in der Elektronik Entwicklung beobachten, bei denen kleine Änderungen an einem eingehenden Design und bestimmte Methoden Parameter große Auswirkungen auf das Ergebnis besitzen. In Abschnitt 6.2 wurde das Beispiel „High Level Synthesis“ (HLS) eingeführt. HLS ist eine Methode, welche automatisiert aus einer existierenden Verhaltensbeschreibung eine hardwarenahe Beschreibung erzeugt. Wesentliche Eigenschaften des resultierenden Designs an welche bestimmte Anforderungen gestellt werden, etwa timing, area und power, können vorab nur heuristisch ermittelt werden. Im Anwendungsbeispiel wurde daher auf eine detaillierte Abbildung verzichtet und variierende Übergangswahrscheinlichkeiten verwendet.

Durch die inhärente Komplexität solcher Methoden (welche Eigenschaften des Designs haben alles Auswirkungen auf das Ergebnis?) sind Wirkzusammenhänge für eine detaillierte erklärende Beschreibung „aller“ Einflussgrößen hier weniger geeignet. Eine Abbildung zur Einflussanalyse erfordert hier daher eine **Abstraktion** zu Wahrscheinlichkeitsverteilungen von Eigenschaften und begrenzt die Anwendbarkeit erstellter Modelle auf bestimmte im Anwendungskontext zu definierende gleichartige Design Klassen. Die Ähnlichkeit zwischen Design Artefakten ist ein Forschungsthema, welches im Rahmen dieser Arbeit nicht behandelt wurde. Ein weiteres Forschungsthema ist der Aufbau möglichst detaillierter Modelle von Methoden entsprechender Design Klassen.

7 Zusammenfassung und Ausblick

Die Entwicklung neuer und innovativer Produkte stellt eine immer komplexer werdende Aufgabe dar. Gleichzeitig steigen die Anforderungen an die Entwicklungsprozesse zur Reduktion des Aufwandes und der Kosten sowie der Verbesserung der Produktqualität, um Produkte gut im Markt positionieren zu können.

Neue Methoden und Werkzeuge versprechen große Potentiale zur Optimierung von Entwicklungsprozessen. Doch, ob sich eine neue Methode im konkreten Anwendungskontext eines Unternehmens wirklich lohnt beziehungsweise wie der Einfluss einer neuen Methode auf einen Entwicklungsprozess zu bewerten ist, ist nicht einfach zu beantworten. So ist eine vom Entwicklungskontext losgelöste Evaluierung von Methoden oft nicht ausreichend und auch kostspielige Pilotprojekte liefern in der Regel nur ein mögliches Bild des Einflusses von vielen.

In dieser Arbeit wird eine Methodik entwickelt (Kapitel 3), welche den Einfluss von Methodenänderungen in Entwicklungsprozessen quantitativ bewertet. Dadurch wird die strategische Entscheidungsfindung für die Auswahl neuer Methoden für angestrebte Prozessoptimierungen unterstützt. Die vorgestellte Methodik gliedert sich im Wesentlichen in die folgenden Punkte:

- Definition der Ziele der Einflussanalyse
- Beschreibung des Entwicklungsprozesses
- Beschreibung von Wirkzusammenhängen
- Sammlung und Analyse empirischer Daten und Quantifizierung von Wirkzusammenhängen
- Validierung des Modells
- Beschreibung der neuen Methode und virtuelle Integration der neuen Methode un
- Simulation der Entwicklungsprozesse und Bewertung des Einflusses der neuen Methode

Zur Beschreibung von Methodenwissen und Entwicklungsprozessen, sowie der Beziehungen zwischen Prozesselementen und Wirkzusammenhängen wird ein existierender Standard erweitert (Kapitel 4). Eine prototypische Umsetzung (Kapitel 5) unterstützt bei

der Modellierung entsprechender Modelle. Mit Hilfe von Simulation alternativer Entwicklungsprozesse wird dann der mögliche Mehrwert einer Prozessänderung bewertet.

Zur Evaluierung (Kapitel 6) des entwickelten Ansatzes werden mehrere Anwendungsbeispiele aus den Domänen der Software-Entwicklung, Hardware-Entwicklung und Lebensmittelindustrie betrachtet. Dabei wird die jeweilige Untersuchung skizziert und die resultierenden Ergebnisse zusammengefasst. Abschließend wird die Erreichung aufgestellter Anforderungen und Ziele an diese Arbeit diskutiert. Dabei wurden die gesetzten Ziele dieser Arbeit insgesamt erreicht, so dass eine Methodik für die Einflussanalyse und Einflussbewertung von Methodenänderungen in Entwicklungsprozessen zur Verfügung steht.

Für die weitere Forschung im Bereich der Einflussanalyse und Einflussbewertung von Methodenänderungen können im Wesentlichen zwei Perspektiven unterschieden werden. Auf der einen Seite steht die Anwendung und Weiterentwicklung der in dieser Arbeit vorgestellten Methodik. Ergänzend dazu sind Erweiterungen beziehungsweise eine Weiterentwicklung des vorgestellten Prototyps zu einer ausgereiften Lösung denkbar.

Es wurden verschiedene Anwendungsbeispiele der Methodik in dieser Arbeit vorgestellt. Als aktuelle Untersuchung zum Zeitpunkt dieser Arbeit bietet sich die weitere Bearbeitung des Themas der Reviews an. Hier können unterschiedliche Review Methoden im Praxiskontext eines großen Automobilzulieferers mit ihren Auswirkungen auf den gesamten Entwicklungsprozess weiter untersucht werden. Hieraus würden sich weitere Erkenntnisse zum Einsatz und Erfolg von Review Methoden in der Praxis ergeben. Zudem wird die hier entwickelte Methodik im Projekt COMPLEX (COdesign and power Management in PPlatform-based design space EXploration), gefördert durch das siebte Framework Programm der Europäischen Kommission, eingesetzt, um den potentiellen Mehrwert entsprechender Methodenänderungen im Entwicklungsprozess zu bewerten. Weiterhin können durch eine stetige Anwendung der Methodik Erfahrungen zu Entwicklungsprozessen und Wirkzusammenhängen gesammelt werden, die gegebenenfalls für eine Weiterentwicklung der Methodik herangezogen werden können.

In diesen Zusammenhang steht auch der Aufbau einer Wissensbasis für Wirkzusammenhänge in Entwicklungsprozessen. So könnten auf Basis von Erfahrungen zunächst relativ generische Wirkzusammenhänge spezifiziert und aus der Literatur gesammelt werden, welche dann für bestimmte Anwendungen konkretisiert werden. Durch ein so aufgebautes Wissen könnte die Auswahl und Modellierung von konkreten Wirkzusammenhängen und Verhaltensmodellen für die Einflussanalyse und Einflussbewertung problemspezifisch unterstützt werden. Zudem sind, auf Basis einer solchen Datenbank von Wirkzusammenhängen, Systeme denkbar, welche den verschiedenen

Beteiligten in Entwicklungsprozessen bei der täglichen Arbeit etwa in der Auswahl sinnvoller Folgeschritte auf Basis aktueller Prozess- und Produktdaten helfen.

Eine weitere Fragestellung ergibt sich mit Blick auf eine automatische Optimierung von Entwicklungsprozessen. Es existieren verschiedene Ansätze, welche eine Optimierung auf Basis der Beziehungen zwischen den Aufgaben eines Entwicklungsprozesses vornehmen. Interessant wäre hier etwa eine Erweiterung verwendeter struktureller Informationen um definierte oder angenommene Wirkzusammenhänge, um dadurch eine weitere Untersuchung entsprechender Veränderungen und deren Auswirkungen zu erlauben.

Die prototypische Umsetzung fokussiert in dieser Arbeit auf eine Unterstützung der Modellierung von Entwicklungsprozessen und deren Simulation sowie Auswertung mit dem Ziel der Bewertung des Einflusses von Methodenänderungen. Ausdrücklich ausgeschlossen waren die Betrachtung von Werkzeugen zur Sammlung und Analyse von Daten sowie darauf basierende Modellbildungen. Hierzu existieren verschiedene Ansätze und technische Lösungen oder Prototypen, welche mit der vorgestellten Lösung integriert werden könnten, um damit eine weitere Durchgängigkeit im Entwicklungsmanagement zu erreichen.

Literaturverzeichnis

Abdel-Hamid, T. K. und Madnick, S. E. (1989): Lessons learned from modeling the dynamics of software development. *Communication ACM*. Band 32. Ausgabe 12. S. 1426-1438. doi:10.1145/76380.76383.

Abdel-Hamid, T. K. und Madnick, S. E. (1991): *Software project dynamics: an integrated approach*. Prentice Hall, Englewood Cliffs. ISBN: 0-13822-0409.

von Ahsen, A., Heesen, M. und Kuchenbuch, A. (2009): *Grundlagen der Bewertung von Innovationen im Mittelstand*. In: von Ahsen, A. (Hrsg.) *Bewertung von Innovationen im Mittelstand*. 1. Auflage. Springer, Berlin. ISBN: 3-64201-6995.

Allweyer, T. (2009): *BPMN 2.0 - Business Process Model and Notation: Einführung in den Standard für die Geschäftsprozessmodellierung*. 2. Auflage. Books on Demand. ISBN: 3-83912-1345.

An, L. und Jeng, J.-J. (2005): On developing system dynamics model for business process simulation. In: *Proceedings of the 37th Conference on Winter Simulation*. Winter Simulation Conference, Orlando, Florida. S. 2068-2077. ISBN: 0-78039-5190.

Balazova, M. (2004): *Methode zur Leistungsbewertung und Leistungssteigerung der Mechatronikentwicklung*. Dissertation. Universität Paderborn, Paderborn.

Banks, J. (1998): *Handbook of simulation: principles, methodology, advances, applications, and practice*. John Wiley & Sons, New York. ISBN: 0-47113-4031.

Basili, V. R. (1985): *Quantitative evaluation of software methodology*. Technical Report. University of Maryland. College Park, Maryland.

Basili, V. R. (2007): The role of controlled experiments in software engineering research. In: *Proceedings of the 2006 International Conference on Empirical Software Engineering Issues: critical assessment and future directions*. Springer-Verlag, Berlin, Heidelberg. S. 33–37. ISBN: 978-3-54071-3005.

Basili, V. R., Caldiera, G. und Rombach, H. D. (1994): The Goal Question Metric Approach. In: Marciniak, J. (Hrsg.) *Encyclopedia of Software Engineering*. Band 1. John Wiley & Sons.

Batinic, B. und Appel, M. (2008): *Medienpsychologie*. 1. Auflage. Springer, Berlin. ISBN: 3-54046-8943.

Birkhölzer, T., Madachy, R., Pfahl, D., Port, D., Beitinger, H., Schuster, M. und Olkov, A. (2010): *SimSWE – A Library of Reusable Components for Software Process Simulation*. In: Münch, J., Yang, Y., und Schäfer, W. (Hrsg.) *New Modeling Concepts for Today's Software Processes*. Band 6195. Springer, Berlin. S. 321-332. ISBN: 978-3-64214-3465.

- Boehm, B., Abts, C., Brown, A. W., Chulani, S., Clark, B., Horowitz, E., Madachy, R., Reifer, D. und Steece, B. (2000a): COCOMO II Model Definition Manual, Version 2.1. Technical Report. Center for Software Engineering, University of Southern California.
- Boehm, B., Abts, C., und Chulani, S. (2000b): Software development cost estimation approaches – A survey. In: *Annals of Software Engineering*. Band 10. Ausgabe 1. S. 177-205. doi:10.1023/A:1018991717352.
- Boehm, B. W., Brown, J. R. und Lipow, M. (1976): Quantitative evaluation of software quality. In: *Proceedings of the 2nd International Conference on Software Engineering*. IEEE Computer Society Press, San Francisco, California, USA. S. 592-605.
- Boehm, B. W., Clark, Horowitz, Brown, Reifer, Chulani, Madachy, R. und Steece, B. (2000c): *Software Cost Estimation with Cocomo II with Cdrom*. Prentice Hall PTR, Upper Saddle River, NJ, USA. ISBN: 0-13026-6922.
- Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R. und Selby, R. (1995): Cost models for future software life cycle processes: COCOMO 2.0. In: *Annals of Software Engineering*. Band 1. Ausgabe 1. S. 57-94. doi:10.1007/BF02249046.
- Boodoo, S., El Emam, K., Laitenberger, O. und Madhavji, N. (2000): *The Optimal Team Size for UML Design Inspections*. Technical Report. National Research Council Canada.
- Bortz, J. und Döring, N. (2006): *Forschungsmethoden und Evaluation: für Human- und Sozialwissenschaftler*. 4., überarb. Auflage. Springer, Berlin. ISBN: 3-54033-3053.
- Bossel, H. (2004): *Systeme, Dynamik, Simulation: Modellbildung, Analyse und Simulation komplexer Systeme*. Books on Demand, Norderstedt. ISBN: 3-83340-9843.
- Briand, L., El Emam, K., Laitenberger, O. und Fussbroich, T. (1997a): Using simulation to build inspection efficiency benchmarks for development projects. Technical Report. doi: 10.1.1.31.3739.
- Briand, L., El Emam, K., Laitenberger, O. und Fussbroich, T. (1998): Using simulation to build inspection efficiency benchmarks for development projects. In: *Proceedings of the 20th International Conference on Software Engineering*. IEEE Computer Society, Washington, DC, USA. S. 340–349. ISBN: 0-81868-3686.
- Briand, L. C., Laitenberger, O., und Wieczorek, I. (1997b): *Building Resource and Quality Management Models for Software Inspections*. Technical Report. Fraunhofer IESE, Kaiserslautern.
- Brinkkemper, S. (1996): Method engineering: engineering of information systems development methods and tools. In: *Information and Software Technology*. Band 38. Ausgabe 4. Elsevier. S. 275-280. doi:10.1016/0950-5849(95)01059-9.

- Brinkkemper, S., Lyytinen, K., und Welke, R. J. (1996): Method engineering: principles of method construction and tool support. In: Proceedings of the IFIP TC8, WG8.1/8.2 Working Conference on Method Engineering, Atlanta, USA. Springer. ISBN: 978-0-41279-7507.
- Browning, T. R. (1998): Modeling and analyzing cost, schedule, and performance in complex system product development. Dissertation. Massachusetts Institute of Technology. Massachusetts, USA.
- Browning, T. R. und Eppinger, S. D. (2002): Modeling impacts of process architecture on cost and schedule risk in product development. In: IEEE Transactions on Engineering Management. Band 49. Ausgabe 4. S. 428-442. doi:10.1109/TEM.2002.806709.
- Browning, T. R., Deyst, J. J., Eppinger, S. D., und Whitney, D. E. (2002): Adding value in product development by creating information and reducing risk. IEEE Transactions on Engineering Management. Band 49. Ausgabe 4. S. 443-458.
- Brugger, R. (2005): IT-Projekte strukturiert realisieren: Situationen analysieren, Lösungen konzipieren – Vorgehen systematisieren, Sachverhalte visualisieren – UML und EPKs nutzen. 2. überarb. und erw. Auflage. Vieweg. ISBN: 3-83480-1186.
- Bryman, A. (2008): Social Research Methods. 3. Auflage. Oxford University Press. ISBN: 0-19920-2958.
- Burghardt, M. (2006): Projektmanagement: Leitfaden für die Planung, Überwachung und Steuerung von Entwicklungsprojekten. 7. wesentlich überarb. und erw. Auflage. Publicis Corporate Publishing. ISBN: 3-89578-2742.
- Buschermöhle, R. und Oelerink, J. (2008): Rich meta object facility formal integration platform: syntax, semantics and implementation. In: Innovations in Systems and Software Engineering. Band 4. Ausgabe 3. S. 249-257. doi:10.1007/s11334-008-0057-0.
- Cho, S.-H. und Eppinger, S. D. (2001): Product Development Process Modeling Using Advanced Simulation. In: Proceedings of Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Pittsburgh, Pennsylvania, USA.
- Choi, K. und Bae, D.-H. (2009): Dynamic project performance estimation by combining static estimation models with system dynamics. In: Information and Software Technology. Band 51. Ausgabe 1. Butterworth-Heinemann. S. 162–172. doi:10.1016/j.infsof.2008.03.001.
- Choi, K., Bae, D. und Kim, T. (2006): An Approach to a Hybrid Software Process Simulation using the DEVS Formalism. In: Software Process: Improvement and Practice. Band 11. Ausgabe 4. Wiley & Sons. S. 373-383. doi:10.1002/spip.284.

- Christenson, D. A., Huang, S. T. und Lamperez, A. J. (1990): Statistical quality control applied to code inspections. In: IEEE Journal on Selected Areas in Communications. Band 8. Ausgabe 2. S. 196-200. doi:10.1109/49.46872.
- Collofello, J., Rus, I., Chauhan, A., Houston, D., Sycamore, D., und Smith-Daniels, D. (1998): A System Dynamics Software Process Simulator for Staffing Policies Decision Support. In: Proceedings of the 31th Hawaii International Conference on System Sciences. Kohala Coast, HI, USA.
- Corsten, H. und Corsten, H. (2000): Projektmanagement. Oldenbourg Wissenschaftsverlag. ISBN: 978-3-48625-2521.
- Coussy, P. und Morawiec, A. (2008): High-Level Synthesis. Springer Netherlands, Dordrecht. ISBN: 978-1-40208-5871.
- Creswell, J. W. (2002): Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. 2. Auflage. SAGE Publications. ISBN: 0-76192-4418.
- Deissenboeck, F. und Pizka, M. (2007): The Economic Impact of Software Process Variations. In: Software Process Dynamics and Agility. Band 4470. S. 259-271. doi:10.1007/978-3-540-72426-1_22.
- Deissenboeck, F. und Pizka, M. (2008): Probabilistic Analysis of Process Economics. In: Software Process: Improvement and Practice. Band 13. Ausgabe 1. John Wiley & Sons. doi: 10.1002/spip.362.
- Dejaeger, K., Verbeke, W., Martens, D. und Baesens, B. (2011): Data Mining Techniques for Software Effort Estimation – A Comparative Study. IEEE Transactions on Software Engineering. doi:10.1109/TSE.2011.55.
- Donzelli, P. (2006): A Decision Support System for Software Project Management. In: IEEE Software. Band 23. Ausgabe 4. S. 67-75. doi:10.1109/MS.2006.90.
- Donzelli, P. und Iazeolla, G. (2001): A hybrid software process simulation model. In: Software Process: Improvement and Practice. Band 6. Ausgabe 2. Wiley & Sons. S. 97-109. doi:10.1002/spip.138.
- Easterbrook, S., Singer, J., Storey, M.-A. und Damian, D. (2008): Selecting Empirical Methods for Software Engineering Research. In: Shull, F., Singer, J., und Sjøberg, D.I.K. (Hrsg.) Guide to Advanced Empirical Software Engineering. Springer London. S. 285-311. ISBN: 978-1-84800-0445.
- Ebert, C. und Dumke, R. (2007): Software measurement: establish, extract, evaluate, execute. Springer, Berlin. ISBN: 978-3-54071-6488.
- Ehrlenspiel, K. (2009): Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit. 4. Auflage. Hanser Verlag. ISBN: 3-44642-0134.

- Eppinger, S. D., Whitney, D. E., Smith, R. P. und Gebala, D. A. (1994): A model-based method for organizing tasks in product development. In: *Research in Engineering Design*. Band 6. Ausgabe 1. S. 1-13. doi:10.1007/BF01588087.
- Ester, M. und Sander, J. (2000): *Knowledge Discovery in Databases: Techniken und Anwendungen*. 1. Auflage. Springer, Berlin. ISBN: 3-54067-3288.
- Fahrmeir, L., Kneib, T. und Lang, S. (2009): *Regression – Modelle, Methoden und Anwendungen*. Springer, Berlin, Heidelberg. ISBN: 978-3-64201-8367.
- Faulkner, X. (2000): *Usability Engineering*. Palgrave Macmillan. ISBN: 0-33377-3217.
- Fenton, N. E. und Neil, M. (2000): *Software Metrics – Roadmap*. In: *Proceedings of the Conference on The Future of Software Engineering*. ACM, Limerick, Ireland. S. 357-370. ISBN: 1-58113-2530. doi:10.1145/336512.336588.
- Forrester, J. W. (1958): *Industrial Dynamics – a major breakthrough for decision makers*. In: *Harvard Business Review*. Band 36. Ausgabe 4. S. 37-66.
- Frankenberger, E. (1997): *Arbeitsteilige Produktentwicklung: empirische Untersuchung und Empfehlungen zur Gruppenarbeit in der Konstruktion*. Technische Hochschule Darmstadt, Düsseldorf.
- Garousi, V., Khosrovian, K. und Pfahl, D. (2009): A customizable pattern-based software process simulation model: design, calibration and application. In: *Software Process: Improvement and Practice*. Band 14. Ausgabe 3. S. 165-180. doi:10.1002/spip.411.
- Gau, T. (2006): *UMA und EPF: Einführung und Anwendung in der Praxis*. In: *Objekt Spektrum*. Ausgabe 06/2006.
- Gausemeier, J., Plass, C. und Wenzelmann, C. (2009): *Zukunftsorientierte Unternehmensgestaltung: Strategien, Geschäftsprozesse und IT-Systeme für die Produktion von morgen*. 1. Auflage. Hanser Fachbuch. ISBN: 3-44641-0554.
- Gilb, T. und Graham, D. (1993): *Software inspection*. Addison-Wesley, Wokingham, England. ISBN: 0-20163-1814.
- Götze, U. (2006): *Investitionsrechnung: Modelle und Analysen zur Beurteilung von Investitionsvorhaben*. 5. Auflage. Springer, Berlin. ISBN: 3-54028-8171.
- Grinstead, C. M. und Snell, J. L. (1997): *Introduction to probability*. 2. Auflage. American Mathematical Society. ISBN: 0-82180-7498.
- Große Austing, S. und Hahn, A. (2010): *Complexity Measurement of Product Models*. In: *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*. SciTePress. S. 404 – 407. ISBN: 978-9-89842-5294.

- Hassine, A. und Barke, E. (2008a): Towards Simulation of Chip Design Processes: The Request Service Model. In: Proceedings of the International Conference on Modelling and Simulation. Quebec, Canada.
- Hassine, A. und Barke, E. (2008b): On Modeling and Simulating Chip Design Processes: The RS Model. In: IEEE International Engineering Management Conference. Estoril, Portugal. S. 81-85. doi: 10.1109/IEMCE.2008.4617958.
- Haumer, P. (2007): Eclipse Process Framework Composer - Part 1: Key Concepts. <http://www.eclipse.org/epf/general/EPFComposerOverviewPart1.pdf> (2011-11-10)
- Häusler, S., Blaschke, J., Sebeke, C., Rosenstiel, W. und Hahn, A. (2009): Project Control System to Track and Optimize Chip Design Projects. In: Proceedings of the World Congress on Engineering and Computer Science. Band 2. San Francisco, USA. S. 1113-1118. ISBN: 978-9-88182-1027.
- Hausmann, K. (2008): Perimeter – Performanzmessung in der Produktentwicklung auf Basis semantisch integrierter Produktmodelle. Dissertation. Universität Oldenburg.
- Heym, M. (1993): Methoden-Engineering: Spezifikation und Integration von Entwicklungsmethoden für Informationssysteme. Dissertation. Universität St. Gallen.
- Heym, M. (1995): Prozeß- und Methoden-Management für Informationssysteme: Überblick und Referenzmodell. Springer, Berlin. ISBN: 3-54056-7895.
- Heym, M. und Osterle, H. (1992): A semantic data model for methodology engineering. In: . Proceedings of the 5th International Workshop on Computer-Aided Software Engineering. Montreal, Quebec, Canada. S. 142-155. doi:10.1109/CASE.1992.200144.
- Hinrichs, N., Hassine, A. und Barke, E. (2009): ADRENALIN – Simulation Chip Design Processes. In: Proceedings of the Design, Automation & Test in Europe Conference. Nice, Frankreich.
- Höfer, T., Przyrembel, H. und Verleger, S. (2004): New evidence for the theory of the stork. In: Paediatric and Perinatal Epidemiology. Band 18. Ausgabe 1. S. 88-92.
- Hummel, T. und Malorny, C. (2002): Total Quality Management. 4. Auflage. Hanser Verlag, München. ISBN: 978-3-44621-8635.
- IBM (2006): IBM Rational Method Composer. ftp://ftp.software.ibm.com/software/rational/web/datasheets/rmc_ds.pdf (2011-11-10)
- IDEF (1993): IDEF0 Integration Definition for Function Modeling. <http://www.idef.com/pdf/idef0.pdf> (2011-11-10)
- Johnson, E. W. (1996): Analysis and Refinement of Iterative Design Processes. Dissertation. University of Notre Dame, Indiana, USA.

- Johnson, E. W. und Brockman, J. B. (1996): Sensitivity analysis of iterative design processes. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design. IEEE Computer Society, San Jose, California, USA. S. 142-145. ISBN: 0-81867-5977.
- Johnson, E. W. und Brockman, J. B. (1998): Measurement and analysis of sequential design processes. ACM Transactions on Design Automation of Electronic Systems. Band 3. Ausgabe 1. ACM, New York, USA. S. 1-20. doi:10.1145/270580.270581.
- Johnson, E. W., Castillo, L. A. und Brockman, J. B. (1996): Application of a Markov model to the measurement, simulation, and diagnosis of an iterative design process. In: Proceedings of the 33rd annual Design Automation Conference. ACM, Las Vegas, Nevada, USA. S. 185-188. ISBN: 0-89791-7790. doi:10.1145/240518.240553.
- Kaplan, R. S. und Norton, D. P. (1997): Balanced Scorecard – Strategien erfolgreich umsetzen. Schäffer-Poeschel Verlag, Stuttgart. ISBN: 978-3-79101-2032.
- Karlsson, F. (2003): Meta-Method for Method Configuration: A Rational Unified Process Case. Dissertation. Linköping University, Schweden.
- Kastens, U. und Büning, H. K. (2008): Modellierung: Grundlagen und formale Methoden. Hanser Verlag. ISBN: 978-3-44641-5379.
- Keller, G., Nüttgens, M. und Scheer, A.-W. (1992): Semantische Prozeßmodellierung auf der Grundlage „Ereignisgesteuerter Prozeßketten (EPK)“. In: Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), Universität des Saarlandes. Band 89.
- Kellner, M. I., Madachy, R. J. und Raffo, D. M. (1999): Software process simulation modeling: Why? What? How? In: Journal of Systems and Software. Band 46. Ausgabe 2/3.
- Kessler, E. H. und Bierly, P. E. (2002): Is Faster Really Better? An Empirical Test of the Implications of Innovation Speed. In: IEEE Transactions on Engineering Management. Band 49. Ausgabe 1. doi:10.1109/17.985742.
- Khosrovian, K., Pfahl, D. und Garousi, V. (2008a): GENSIM 2.0: A Customizable Process Simulation Model for Software Process Evaluation. In: Making Globally Distributed Software Development a Success Story. S. 294-306. doi:10.1007/978-3-540-79588-9_26.
- Khosrovian, K., Pfahl, D. und Garousi, V. (2008b): GENSIM 2.0: A Customizable Process Simulation Model for Software Process Evaluation. Technical Report. University of Calgary and Simula Research Laboratory.
- Khosrovian, K., Pfahl, D. und Garousi, V. (2008c): Calibrating a Customizable System Dynamics Simulation Model of Generic Software Development Processes. Technical Report. University of Calgary and Simula Research Laboratory.

- Kitchenham, B. A. (1996a): DESMET: A method for evaluating Software Engineering methods and tools. Technical Report. University of Keele, Keele, England. ISSN: 1353-7776. <http://www.osel.co.uk/desmet.pdf> (2011-11-10)
- Kitchenham, B. A. (1996b): Evaluating Software Engineering Methods and Tools - Part 1: The Evaluation Context and Evaluation Methods. In: SIGSOFT Software Engineering Notes. Band 21. Ausgabe 1. S. 11-14. doi:10.1145/381790.381795.
- Kohn, W. (2005): Statistik – Datenanalyse und Wahrscheinlichkeitsrechnung. Springer, Berlin. ISBN: 3-54021-6774.
- Kontio, J., Bragge, J. und Lehtola, L. (2008): The Focus Group Method as an Empirical Tool in Software Engineering. In: Shull, F., Singer, J., und Sjøberg, D.I.K. (Hrsg.) Guide to Advanced Empirical Software Engineering. Springer, London. S. 93-116. ISBN: 978-1-84800-0445.
- Koppe, R., Häusler, S., Buschermöhle, R. und Hahn, A. (2010a): Process Change Impact Analysis Tool. In: Proceedings of the 1st International Conference on Modelling and Management of Engineering Processes. Cambridge.
- Koppe, R., Häusler, S., große Austing, S., Poppen, F. und Hahn, A. (2011a): Economic Impact Estimation of New Design Methods. In: Proceedings of the International Conference on Engineering Design. Kopenhagen.
- Koppe, R., Häusler, S., Poppen, F., und Hahn, A. (2010b): Process Model Based Methodology for Impact Analysis of new Design Methods. In: Heisig, P., Vajna, S., und Clarkson, P.J. (Hrsg.) Modelling and Management of Engineering Processes. 1. Auflage. Springer, London. S. 53-64. ISBN: 978-1-84996-1981.
- Koppe, R., Häusler, S., Poppen, F. und Hahn, A. (2011b): Impact Estimation for Design Flow Changes. In: Proceedings of the eda Workshop. Dresden.
- Koppe, R., Häusler, S. und Hahn, A. (2010c): Ein Modell zur Einflussanalyse von Änderungen in Entwicklungsprozessen. In: Informatik 2010, Lecture Notes in Informatics. Band 2. Gesellschaft für Informatik, Leipzig. S. 639-644. ISBN: 978-3-88579-2697.
- Krallmann, H. (2007): Systemanalyse im Unternehmen: prozessorientierte Methoden der Wirtschaftsinformatik. Oldenbourg Wissenschaftsverlag. ISBN: 978-3-48658-4462.
- Kröll, M. (2008): Methode zur Technologiebewertung für eine ergebnisorientierte Produktentwicklung. Dissertation. Universität Stuttgart, Stuttgart.
- Laitenberger, O., Leszak, M., Stoll, D. und Emam, K. E. (1999): Quantitative Modeling of Software Reviews in an Industrial Setting. In: IEEE International Symposium on Software Metrics. IEEE Computer Society, Los Alamitos, CA, USA. doi:10.1109/METRIC.1999.809752.

- Lakey, P. B. (2003): A Hybrid Software Process Simulation Model for Project Management. In: Proceedings of the 4th International Workshop on Software Process Simulation Modeling (ProSim). Portland, Oregon.
- Lang, C. (2004): Organisation der Software-Entwicklung. DUV. ISBN: 978-3-82448-1132.
- Law, A. M. und Kelton, W. D. (1991): Simulation modeling and analysis. 2. Auflage. McGraw Hill, New York, USA. ISBN: 0-07036-6985.
- Law, A. M. und McComas, M. G. (1989): Pitfalls to avoid in the simulation of manufacturing systems. In: Industrial Engineering. Band 21. Ausgabe 5. S. 28-31.
- Lemieux, C. (2009): The Monte Carlo Method. In: Monte Carlo and Quasi-Monte Carlo Sampling. Springer, New York, USA. S. 1-39. ISBN: 978-0-38778-1648.
- Lindemann, U. (2007): Methodische Entwicklung technischer Produkte. Springer, Berlin Heidelberg. ISBN: 978-3-54037-4350.
- Lloyd, S. (2001): Measures of complexity: a nonexhaustive list. In: IEEE Control Systems Magazine. Band 21. Ausgabe 4. S. 7-8. doi:10.1109/MCS.2001.939938.
- Luenberger, D. G. (2003): Linear and nonlinear programming. 2. Auflage. Kluwer Academic Publishers, Bosten, USA. ISBN: 1-40207-5936.
- Madachy, R. J. (1996): System dynamics modeling of an inspection-based process. In: Proceedings of the 18th International Conference on Software Engineering. IEEE Computer Society, Berlin. S. 376-386. ISBN: 0-81867-2463.
- Madachy, R. J. (2008): Software process dynamics. Wiley-IEEE. ISBN: 978-0-47127-4551.
- Malhotra, R. und Jain, A. (2011): Software Effort Prediction using Statistical and Machine Learning Methods. In: International Journal of Advanced Computer Science and Applications. Band 2. Ausgabe 1. S. 1451-1521.
- Martin, R. H. und Raffo, D. (2000): A model of the software development process using both continuous and discrete models. In: Software Process: Improvement and Practice. Band 5. Ausgabe 2-3. S. 147-157.
- Martin, R. und Raffo, D. (2001): Application of a hybrid process simulation model to a software development project. In: Journal of Systems and Software. Band 59. Ausgabe 3. S. 237-246. doi:10.1016/S0164-1212(01)00065-6.
- Mayer, R. (1992): IDEF1 Information Modeling Method. Technical Report. Knowledge Based Systems, College Station, Texas, USA.

- Mayer, R., Menzel, C., Painter, M., deWitte, P., Blinn, T., und Perakath, B. (1995): IDEF3 Process Description Capture Method. Knowledge Based Systems, College Station, Texas, USA.
- Meintrup, D. und Schäffler, S. (2005): Stochastik: Theorie und Anwendungen. 1. Auflage. Springer, Berlin. ISBN: 3-54021-6766.
- Miller, R. W. (1965): Zeit-Planung und Kosten-Kontrolle durch PERT – Ein Leitfaden für die Anwendung in Entwicklung und Fertigung. Decker, Hamburg.
- Mohagheghi, P., Anda, B. und Conradi, R. (2005): Effort estimation of use cases for incremental large-scale software development. In: Proceedings of the 27th International Conference on Software Engineering. ACM, St. Louis, MO, USA. S. 303-311. ISBN: 1-59593-9632. doi:10.1145/1062455.1062516.
- Morgan, D. L. (1998): Planning Focus Groups. Band 2. Sage Publications. ISBN: 0-76190-817X.
- Müller, M. (2007): Analyzing Software Quality Assurance Strategies through Simulation – Development and Empirical Validation of a Simulation Model in an Industrial Software Product Line Organization. Dissertation. Technische Universität Kaiserslautern.
- Müller, M. und Pfahl, D. (2008): Simulation Methods. In: Shull, F., Singer, J., und Sjøberg, D.I.K. (Hrsg.) Guide to Advanced Empirical Software Engineering. Springer, London. S. 117-152. ISBN: 978-1-84800-0445.
- Münch, J., Pfahl, D., und Rus, I. (2005): Virtual Software Engineering Laboratories in Support of Trade-off Analyses. In: Software Quality Journal. Band 13. Ausgabe 4. S. 407-428. doi:10.1007/s11219-005-4253-y.
- Neumann, K. (1979): Recent advances in temporal analysis of GERT networks. Zeitschrift für Operations Research. Band 23. Ausgabe 5. S. 153-177. doi:10.1007/BF01919481.
- Nicolai, W. (1980): On the temporal analysis of special GERT networks using a modified Markov renewal process. Zeitschrift für Operations Research. Band 24. Ausgabe 7. S. 263-272. doi:10.1007/BF01919904.
- OMG (2008): Software & Systems Process Engineering Meta-Model Specification. www.omg.org/cgi-bin/doc?formal/08-04-01.pdf (2011-11-10)
- OMG (2009a): Unified Modeling Language, Infrastructure. <http://www.omg.org/docs/formal/09-02-04.pdf> (2011-11-10)
- OMG (2009b): Business Process Model and Notation (BPMN). <http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF> (2011-11-10)

-
- OMG (2011): Business Process Model and Notation (BPMN), Version 2.0. <http://www.omg.org/cgi-bin/doc?dtc/10-06-04> (2011-11-10)
- Ophey, L. (2005): Entwicklungsmanagement – Methoden in der Produktentwicklung. Springer, Berlin. ISBN: 3-54020-6523.
- Ordonez, M. J. und Haddad, H. M. (2008): The State of Metrics in Software Industry. In: 3rd International Conference on Information Technology: New Generations. IEEE Computer Society, Los Alamitos, CA, USA. S. 453-458. ISBN: 978-0-76953-0994. doi:10.1109/ITNG.2008.106.
- Park, S., Choi, K., Yoon, K. und Bae, D.-H. (2007): Deriving Software Process Simulation Model from SPEM-based Software Process Model. In: Proceedings of the 14th Asia-Pacific Software Engineering Conference. IEEE Computer Society. S. 382-389. ISBN: 0-76953-0575.
- Park, S., Kim, H., Kang, D. und Bae, D.-H. (2008): Developing a Simulation Model Using a SPEM-Based Process Model and Analytical Models. In: Advances in Enterprise Engineering I. S. 164-178. doi:10.1007/978-3-540-68644-6_12.
- Pfahl, D. (2001): An integrated approach to simulation-based learning in support of strategic and project management in software organisations. Dissertation. Universität Kaiserslautern.
- Pfleeger, S. L. und Kitchenham, B. A. (2001): Principles of survey research part 1: turning lemons into lemonade. In: SIGSOFT Software Engineering Notes. Band 26. Ausgabe 6. S. 16-18. doi:10.1145/505532.505535.
- Pillai, K. und Sukumaran Nair, V. S. (1997): A model for software development effort and cost estimation. In IEEE Transactions on Software Engineering. Band 23. Ausgabe 8. S. 485-497. doi:10.1109/32.624305.
- Poppen, F., Koppe, R., Hahn, A., und Grüttner, K. (2011): Impact Simulation of Changes to Development Processes: An ESL Case Study. In: Proceedings of Forum on Specification & Design Languages, Oldenburg.
- Poppen, F., Schulte, M. und Nebel, W. (2006): Power Optimised Digital Filterbank as Part of a Psychoacoustic Human Hearing Model. In: Proceedings of the SNUG Europe.
- Pritsker, A. A. B. (1966): GERT – Graphical Evaluation and Review Technique. Technical Report. National Aeronautics and Space Administration. http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19670022025_1967022025.pdf (2011-11-10)
- Pritsker, A. A. B. und Whitehouse, G. E. (1966): GERT – Graphical Evaluation and Review Technique – Part II Probabilistic and Industrial Engineering Applications. In: Journal of Industrial Engineering. Ausgabe 17. S. 293-301.

- Pruscha, H. (2006): Statistisches Methodenbuch. Springer, Berlin, Heidelberg. ISBN: 978-3-54026-0066.
- Putnam, L. H. (1978): A General Empirical Solution to the Macro Software Sizing and Estimating Problem. IEEE Transactions on Software Engineering. Band 4. Ausgabe 4. doi: 10.1109/TSE.1978.231521.
- Putnam, L. H. und Myers, W. (1991): Measures for Excellence: Reliable Software on Time, Within Budget. Prentice Hall PTR. ISBN: 978-0-13567-6943.
- Raffo, D. (1993): Evaluating the impact of process improvements quantitatively using process modeling. In: Proceedings of the Conference of the Centre for Advanced Studies on Collaborative Research: Software Engineering. Band 1. IBM Press, Toronto, Ontario, Canada. S. 290-313.
- Raffo, D., Ferguson, R., Setamanit, S.-on und Sethanandha, B. (2007): Evaluating the Impact of the QuARS Requirements Analysis Tool Using Simulation. In: Software Process Dynamics and Agility. Band 4470. S. 307-319. doi:10.1007/978-3-540-72426-1_26.
- Raffo, D., Ferguson, R., Setamanit, S.-on und Sethanandha, B. (2008): Evaluating the impact of requirements analysis tools using simulation. In: Software Process: Improvement and Practice – Advances in Software Process Improvement. Band 13. Ausgabe 1. Wiley & Sons, New York, USA. S. 63-73.
- Raffo, D., Kaltio, T., Partridge, D., Phalp, K. und Ramil, J. F. (1999a): Empirical Studies Applied to Software Process Models. In: Empirical Software Engineering. Band 4. Ausgabe 4. S. 353-369. doi:10.1023/A:1009817721252.
- Raffo, D. M., Vandeville, J. V. und Martin, R. H. (1999b): Software process simulation to achieve higher CMM levels. In: Journal of Systems and Software. Band 46. Ausgabe 2-3. Elsevier. S. 163-172. doi:10.1016/S0164-1212(99)00009-6.
- Rosenberg, J. (2008): Statistical Methods and Measurement. In: Shull, F., Singer, J. und Sjøberg, D.I.K. (Hrsg.) Guide to Advanced Empirical Software Engineering. Springer, London. S. 155-184. ISBN: 978-1-84800-0445.
- Rösler, P. (2008): Warum Prüfen 50-mal länger dauert als Lesen und andere Überraschungen aus der Welt der Software-Reviews. Gesellschaft für Informatik, Reginalgruppe Rhein-Main. http://www.reviewtechnik.de/Reviewtechnik_GI_080923.pdf (2011-11-10)
- Ross, D. T. (1977): Structured Analysis (SA) – A Language for Communicating Ideas. In: IEEE Transactions on Software Engineering. Band SE-3. Ausgabe 1. S. 16-34.

- Ruiz, M., Ramos, I. und Toro, M. (2002): A Dynamic Integrated Framework for Software Process Improvement. In: *Software Quality Journal*. Band 10. Ausgabe 2. S. 181-194. doi:10.1023/A:1020580008694.
- Rus, I. (1998): Modeling the Impact on Cost and Schedule of Software Quality Engineering Practices. Dissertation. Arizona State University, USA.
- Rus, I. und Collofello, J. (2001): Integrating Process Simulation and Reliability Models. In: *Journal of Defense Software Engineering*. <http://www.crosstalkonline.org/storage/issue-archives/2001/200101/200101-Rus.pdf> (2011-11-10)
- Rus, I., Biffl, S. und Halling, M. (2002): Systematically combining process simulation and empirical data in support of decision analysis in software development. In: *Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering*. ACM, Ischia, Italien. S. 827-833. ISBN: 1-58113-5564. doi:10.1145/568760.568901.
- Rus, I., Collofello, J. und Lakey, P. (1999): Software process simulation for reliability management. In: *Journal of Systems and Software*. Band 46. Ausgabe 2-3. S. 173-182. doi:10.1016/S0164-1212(99)00010-2.
- Rus, I., Neu, H. und Münch, J. (2003): A Systematic Methodology for Developing Discrete Event Simulation Models of Software Development Processes. In: *ProSim Workshop on Software Process Simulation Modelling*. Maryland, USA.
- Sachs, L. und Hedderich, J. (2006): *Angewandte Statistik – Methodensammlung mit R*. 12. vollst. neu bearb. Auflage. Springer, Berlin. ISBN: 3-54032-1608.
- Salant, P. und Dillman, D. A. (1994): *How to conduct your own survey*. Wiley & Sons. ISBN: 978-0-47101-2733.
- Schabacker, M. (2001): *Bewertung der Nutzen neuer Technologien in der Produktentwicklung*. Dissertation. Otto-von-Guericke-Universität Magdeburg.
- Scheer, A.-W. und Jost, W. (2002): *ARIS in der Praxis*. 1. Auflage. Springer, Berlin. ISBN: 978-3-54043-0292.
- Scheer, A.-W., Boczanski, M., Muth, M., Schmitz, W.-G. und Segelbacher, U. (2006): *Prozessorientiertes Product Lifecycle Management*. Springer, Berlin. ISBN: 3-54028-4028.
- Schulz, T., Radliński, Ł., Gorges, T. und Rosenstiel, W. (2010): Defect cost flow model: a Bayesian network for predicting defect correction effort. In: *Proceedings of the 6th International Conference on Predictive Models in Software Engineering*. ACM, New York, USA. S. 16:1–16:11. doi:10.1145/1868328.1868353.

- Seaman, C. B. (2008): Qualitative Methods. In: Shull, F., Singer, J., und Sjøberg, D.I.K. (Hrsg.) Guide to Advanced Empirical Software Engineering. Springer, London. S. 35-62. ISBN: 978-1-84800-0445.
- SEI (2006): CMMI for Development, Version 1.2. Software Engineering Institute, Carnegie Mellon University. <http://www.sei.cmu.edu/library/abstracts/reports/06tr008.cfm> (2011-11-10)
- Singer, J., Sim, S. E. und Lethbridge, T. C. (2008): Software Engineering Data Collection for Field Studies. In: Shull, F., Singer, J., und Sjøberg, D.I.K. (Hrsg.) Guide to Advanced Empirical Software Engineering. Springer, London. S. 9-34. ISBN: 978-1-84800-0445.
- Smith, R. P. und Eppinger, S. D. (1997): A Predictive Model of Sequential Iteration in Engineering Design. In: Management Science. Band 43. Ausgabe 8. S. 1104-1120.
- Solomon, S. L. (1983): Simulation of waiting-line systems. Prentice Hall, Englewood Cliffs. ISBN: 0-13810-0446.
- Sommerville, I. (2007): Software engineering. 8. Auflage. Pearson Education. ISBN: 978-0-32131-3799.
- Steward, D. V. (1981): The Design Structure System – A Method for Managing the Design of Complex Systems. In: IEEE Transactions on Engineering Management. Band EM-28. Ausgabe 3. S. 71-74.
- Strickmann, J. (2008): Analysemethoden zur Bewertung von Entwicklungsprojekten – Ein integriertes semantisches Modell von Projekt- und Produktdaten zur Bewertung der Entwicklungsleistung im Projektcontrolling. Dissertation. Universität Oldenburg.
- Töpfer, A. (2007): Six Sigma: Konzeption und Erfolgsbeispiele für praktizierte Null-Fehler-Qualität. Springer, Berlin. ISBN: 978-3-54048-5919.
- Ulgen, O. M., Gunal, A. und Shore, J. (1996): Pitfalls of Simulation Modeling and How to Avoid Them by Using a Robust Simulation Methodology. In: Proceedings of the Auto Simulations' Symposium. AutoSimulations Inc., Bountiful, Utah.
- Velden, M. (1982): Die Signalentdeckungstheorie in der Psychologie. Kohlhammer, Stuttgart. ISBN: 3-17004-9364.
- Wernick, P. und Hall, T. (2004): A Policy Investigation Model for Long-term Software Evolution Processes. In: The 5th International Workshop on Software Process Simulation and Modeling. Edinburgh, England.
- Wickens, C. D. und Hollands, J. G. (1999): Engineering Psychology and Human Performance. 3. Auflage. Prentice Hall. ISBN: 0-32104-7117.
- Witten, I. H., Frank, E., und Hall, M. A. (2011): Data Mining: Practical Machine Learning Tools and Techniques. 3. Auflage. Morgan Kaufmann. ISBN: 0-12374-8569.

Yin, R. K. (1994): Case Study Research: Design and Methods. 4. Auflage. Sage Publications. ISBN: 0-80395-6622.

Zhang, H. und Kitchenham, B. (2006): Semi-quantitative Simulation Modeling of Software Engineering Process. In: Software Process Change. Band 3966. S. 242-253. doi: 10.1007/11754305_27.

Zhang, H., Jeffery, R. und Zhu, L. (2008a): Investigating test-and-fix processes of incremental development using hybrid process simulation. In: Proceedings of the 6th International Workshop on Software Quality. ACM, Leipzig. S. 23-28. ISBN: 978-1-60558-0234. doi:10.1145/1370099.1370105.

Zhang, H., Keung, J., Kitchenham, B., und Jeffery, R. (2008b): Semi-quantitative Modeling for Managing Software Development Processes. In: Australian Software Engineering Conference. IEEE Computer Society, USA. S. 66-75. doi: 10.1109/ASWEC.2008.4483194.

Zhang, H., Kitchenham, B., und Pfahl, D. (2008c): Software process simulation over the past decade: trends discovery from a systematic review. In: Proceedings of the 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. ACM, Kaiserslautern. S. 345-347. ISBN: 978-1-59593-9715. doi:10.1145/1414004.1414077.

Anhang A – Leitfaden zur Erhebung von Entwicklungsprozessen

Der Leitfaden zur Erhebung von Entwicklungsprozessen gibt eine Übersicht über zielgerichtete Fragen, die bei der Erfassung der Aufgaben, Abläufe, Artefakte, Rollen und Werkzeuge eines Entwicklungsprozesses helfen sollen. Dabei kann der Leitfaden grundsätzlich etwa für Interviews, Fokusgruppen und Workshops eingesetzt werden. Neben der Umsetzung des Leitfadens durch Befragung oder reine Diskussion können die vorgestellten Fragen auch im Rahmen einer Pinnwandmoderation oder auch Metaplan-Methode verwendet werden. Die im Folgenden angegebene Dauer ist ein grober Richtwert, der je nach Umfang des betrachteten Entwicklungsprozesses und der Anzahl der beteiligten Personen abzuschätzen ist. So wird beispielsweise ein Interview mit einem Experten zu einem „überschaubaren“ Entwicklungsprozess weniger Zeit in Anspruch nehmen als eine Fokusgruppe zu einem „umfangreichen“ Entwicklungsprozess.

Begrüßung

Guten Tag! Mein Name ist [Name des Interviewers / Moderators] von [Organisation] in [Ort]. Mit unserer Studie möchten wir den Einfluss von Methodenänderungen in Entwicklungsprozessen untersuchen. Dazu möchte ich Ihnen einige Fragen zum aktuellen Entwicklungsprozess beziehungsweise zu einer Änderung des Entwicklungsprozesses stellen. Unser heutiges Treffen wird etwa [1 bis 4 Stunden] dauern.

Leitfragen

- 1. Welche Informationen wurden Ihnen im Vorfeld zu unserer Untersuchung gegeben?**
Sind den Beteiligten die Ziele der Untersuchung noch nicht ganz klar, sollten die Ziele und die Methodik noch einmal kurz erläutert werden.
- 2. Haben Sie noch Fragen oder können wir beginnen?**
Sind noch Fragen offen, sollten diese nun zunächst geklärt werden.
- 3. Welche Ziele sollen mit der Untersuchung des Entwicklungsprozesses und der Einflussanalyse erreicht werden?**
Die grundsätzlichen Ziele sollten im Vorfeld bereits geklärt sein. Ansonsten ist der Fokus auf den Aufwand beziehungsweise die Kosten oder auf die Untersuchung der Produktqualität zu legen. Siehe Abschnitt 3.4.

4. Welches sind die wesentlichen Aufgaben in Ihrem Entwicklungsprozess?

Fertigen Sie eine Liste der Aufgaben im Entwicklungsprozess an. Steht eine Tafel / Whiteboard / Pinnwand zur Verfügung bietet sich an die Aufgaben jeweils geeignet zu notieren und bereits in ihrem normalen Ablauf zu sortieren. Dieses ermöglicht eine gute Grundlage zur Diskussion entsprechender Aufgaben. Siehe Abschnitt 3.5.

5. Gibt es zwischen den genannten Aufgaben Rücksprünge und von welcher Aufgabe zur welcher anderen Aufgabe?

Notieren Sie sich die Beziehungen zwischen den Aufgaben. Wenn Sie Tafel / Whiteboard / Pinnwand verwenden, dann können Sie die Abläufe direkt einzeichnen und diskutieren. Siehe Abschnitt 3.5.

6. Wie wahrscheinlich sind die festgestellten Rücksprünge im Verhältnis zueinander?

Notieren Sie die jeweiligen Wahrscheinlichkeiten an den Iterationen von einer Aufgabe zu den übrigen nachfolgenden Aufgaben als Übergangswahrscheinlichkeiten. Auch qualitative Aussagen wie „häufig“ oder „selten“ sind für eine erste Beschreibung sinnvoll.

7. Welche Design Artefakte sind notwendige Eingaben oder wichtige Ausgaben der gesammelten Aufgaben?

Notieren Sie Design Artefakte mit ihren Beziehungen, wenn die Teilnehmer einen wesentlichen Einfluss auf die gesetzten Ziele vermuten. Siehe Abschnitt 3.5.

8. Welche Eigenschaften der Design Artefakte beeinflussen wesentlich die gesetzten Ziele?

Notieren Sie die identifizierten Eigenschaften zu den Design Artefakten und lassen Sie diese durch die Teilnehmer, wenn möglich, priorisieren. Siehe Abschnitt 3.5 und 3.6.

9. Durch welche Rollen werden die gesammelten Aufgaben primär durchgeführt?

Notieren Sie Rollen mit ihren Beziehungen, wenn die Teilnehmer einen wesentlichen Einfluss auf die gesetzten Ziele vermuten. Siehe Abschnitt 3.5.

10. Welche Eigenschaften dieser Rollen beeinflussen wesentlich die gesetzten Ziele?

Notieren Sie die identifizierten Eigenschaften zu den Rollen und lassen Sie diese durch die Teilnehmer, wenn möglich, priorisieren. Siehe Abschnitt 3.5 und 3.6.

11. Welche Werkzeuge sind für die gesammelten Aufgaben notwendig?

Notieren Sie Werkzeuge mit ihren Beziehungen, wenn die Teilnehmer einen wesentlichen Einfluss auf die gesetzten Ziele vermuten. Siehe Abschnitt 3.5.

12. Welche Eigenschaften dieser Werkzeuge beeinflussen wesentlich die gesetzten Ziele?

Notieren Sie die identifizierten Eigenschaften zu den Werkzeugen und lassen Sie diese durch die Teilnehmer, wenn möglich, priorisieren. Siehe Abschnitt 3.5 und 3.6.

13. Welche Prozesselemente beziehungsweise Eigenschaften beeinflussen sich gegenseitig und wie bewerten Sie den Einfluss?

Notieren Sie die genannten Beziehungen als Wirkzusammenhänge zwischen den Prozesselementen. Lassen Sie die Teilnehmer die erfassten Wirkzusammenhänge auf ein jeweiliges Prozesselement oder eine Eigenschaft priorisieren, wenn möglich. Siehe Abschnitt 3.6.

14. Wie oft werden die gesammelten Aufgaben Ihrer Erfahrung nach wiederholt?

Notieren Sie die Anzahl der minimalen / wahrscheinlichen / maximalen Iterationen einer Aufgabe. Sie können im Anschluss an das Treffen die Anzahl der Iterationen gegen die angenommenen Übergangswahrscheinlichkeiten prüfen, indem der Entwicklungsprozess mit Übergangswahrscheinlichkeiten Markov-analysiert oder simuliert wird.

15. Wie viel Aufwand kostet ein Durchlauf einer Aufgabe durchschnittlich?

Notieren Sie den minimalen / wahrscheinlichen / maximalen Aufwand für eine Iteration einer Aufgabe. Bevor Sie die Aufwände bestimmen lassen, diskutieren Sie eine Referenz-Komplexität im Sinne von wie viel Aufwand bedeutet ein XY-Artefakt.

16. Welche Änderungen ergeben sich durch die neue Methode?

Beziehen Sie die Fragen 1 bis 15 auf die Änderung im Entwicklungsprozess. Notieren Sie sich neue oder sich ändernde Aufgaben, Abläufe, Rollen und Werkzeuge. Lassen Sie sich ändernde und neue Wirkzusammenhänge diskutieren. Siehe Abschnitt 3.10.

Abschluss

Vielen Dank für Ihre Mitarbeit. Wir werden die heute mit Ihnen gesammelten Informationen als Entwicklungsprozesse modellieren und kommen bei Fragen gezielt auf Sie zurück. Nachdem wir die alternativen Entwicklungsprozesse modelliert haben, werden wir Ihnen diese vorstellen und mit Ihnen diskutieren. Gibt es von Ihrer Seite aus noch Fragen zum heutigen Treffen?

Folgende Schritte

- Die erfassten Informationen sind als alternative Entwicklungsprozesse zu modellieren und mit Experten abzustimmen.
- Aus den gesammelten Eigenschaften und Wirkzusammenhängen sind Hypothesen zu Verhaltensmodellen abzuleiten. Diese sind gegebenenfalls mit entsprechenden Experten zu validieren.
- Zur Quantifizierung der Wirkzusammenhänge und Verhaltensmodelle sind Konzepte zu erarbeiten, welche Daten sinnvoller Weise gesammelt und analysiert werden können.

