

Diplomarbeit an der Universität Ulm
Fakultät für Informatik
Abteilung Datenbanken und Informationssysteme

Workflowbasierte Unterstützung von sich ändernden Entwicklungsprozessen – Anforderungen und Lösungsansätze –



Florian Breier

Februar 2006

1. Gutachter: Prof. Dr. Peter Dadam, Universität Ulm
2. Gutachter: Prof. Dr. Manfred Reichert, Universität Twente

für alle, die nicht an mir verzweifelt sind

Kurzfassung

Innovationen der Automobilindustrie beruhen immer mehr auf elektronischen Steuergeräten und Software. Verteilte Teams entwickeln diese in drei bis fünfjährigen Projekten. Dabei verwendete Prozesse zur Unterstützung und Koordination werden in Informationssystemen umgesetzt. Das dynamische Umfeld führt zu sich ständig ändernden Bedingungen, an die sich das Projekt, die Prozesse, die Mitarbeiter und letztendlich auch das Informationssystem anpassen müssen.

Die Grundlage und den Schwerpunkt dieser Arbeit bildet die Untersuchung von Prozessen im Rahmen der Entwicklung eingebetteter Steuergeräte. Vor allem Änderungen und deren Ursachen werden aufgezeigt. Besonders das Problem- und Änderungsmanagement kann in seiner Entwicklung über acht Projekte hinweg verglichen werden.

Darauf aufbauend wird eine Kategorisierung entwickelt, um die gefundenen Änderungen einordnen zu können. Die Kategorisierung beschreibt die Änderungen in zwei Dimensionen: nach Ursache und Planbarkeit. Diese beiden Teile sind breit angelegt, um die Herausforderungen des Anforderungsgebiets aufzuzeigen.

Die gemachten Beobachtungen werden im Folgenden vertieft. Die sich aus den Fallbeispielen ergebenden Anforderungen werden an den Fähigkeiten des Prozessmanagementsystems ADEPT gespiegelt. Punktuell wird gezeigt, wie die Flexibilität realisiert werden kann, die sich aus den Fallbeispielen ergibt.

Benutzer reagieren flexibel auf sich ändernde Rahmenbedingungen. Dieses mentale Modell der Benutzer bildet die Vorlage, um die gewünschte Verhaltensweise in prozessbasierten Informationssystemen, am Beispiel ADEPT, abbilden zu können.

Inhaltsverzeichnis

Tabellenverzeichnis	8
Abbildungsverzeichnis	9
1 Einleitung	11
1.1 Problembeschreibung	11
1.2 Aufgabenstellung	11
1.3 Gewählter Lösungsansatz	12
2 Übersicht der relevanten Grundlagen	15
2.1 Grundlagen aus der Organisationstheorie	15
2.2 Informationstechnische Umsetzung von Prozessen	17
2.3 Einschränkungen der Umsetzung	18
3 Fallbeispiele	21
3.1 Übersicht	21
3.2 Entwicklungsvorgehen	22
3.2.1 Beschreibung	22
3.2.2 Weiterentwicklung	23
3.2.3 Sonderfunktionen	24
3.3 Änderungsmanagement	25
3.3.1 Allgemein	25
3.3.2 Anmerkungen zu Projekten	26
3.3.3 Fallbeispiele	27
3.3.4 Auswertung	31
3.4 Freigabemangement	38
3.5 Zusammenfassung	40
4 Kategorisierung der Änderungen	41
4.1 Entwicklung der Kategorisierung	41
4.1.1 Änderungen aus Perspektive eines Prozess-Management-Systems	41
4.1.2 Änderungen aus Organisationssicht	42
4.1.3 Änderungen aus Sicht eines Referenzprozessmodells	44
4.1.4 Anpassung und Erweiterung	45
4.2 Einordnung der Beispiele	48
4.2.1 Projektbezogene Änderungen	48
4.2.2 Änderungen der Aufbauorganisation	48

4.2.3	Technologische Änderungen	49
4.2.4	Kulturelle Änderungsgründe	50
4.2.5	Strategische Änderungen	51
4.2.6	Rechtliche Änderungen	52
4.3	Zusammenfassung	52
5	Zusammenfassung der Anforderungen	55
5.1	Vor Projektbeginn	55
5.2	Zur Projektlaufzeit	55
6	Analyse der Umsetzbarkeit in ADEPT	57
6.1	Das Prozessmanagementsystem ADEPT	57
6.1.1	Prozess-Meta-Modell	57
6.1.2	Organisationsmodell	59
6.1.3	Abweichung vom Standardablauf	60
6.2	Umsetzung	62
6.2.1	ADEPT und die Kategorisierung	65
6.3	Zusammenfassung	68
7	Erweiterungsvorschläge am Beispiel von ADEPT	69
7.1	Integration semantischer Informationen und Prozessänderungen	69
7.2	Szenarien	71
7.2.1	Instanz anhalten	72
7.2.2	Instanz beenden	73
7.2.3	Aktivität validieren	75
7.3	Umsetzung der Szenarien	76
7.4	Zusammenfassung	79
8	Zusammenfassung und Ausblick	81
8.1	Zusammenfassung	81
8.2	Ausblick	82
	Literaturverzeichnis	83
A		85
B	Prozessgraphen	89
B.1	Vertraulich	89

Tabellenverzeichnis

3.1	Übersicht der untersuchten Änderungsmanagement Prozesse	27
3.2	Übersicht der Rollen pro Projekt	32
3.3	Verwendung von Aktivitäten in den einzelnen Projekten	34
3.4	Bearbeiterzuordnung einzelner Aktivitäten in Abhängigkeit des Projekts	35
3.5	Projektspezifische Prüfzyklen	36
3.6	Projektspezifischer Prozessabbruch	37
4.1	Vergleich Kategorisierungen	45
4.2	Kategorisierung von Änderungen	47
6.1	Definition von Bearbeiterzuordnungen	60

Abbildungsverzeichnis

1.1	Lösungsansatz	13
2.1	Technologies in Practice	19
3.1	Überblick der Entwicklung eingebetteter Steuergeräte	21
3.2	Prinzipieller Ablauf und Aktivitäten des Änderungsmanagements	25
3.3	Zustandsübergänge ÄM(PVCS)	30
6.1	Kontrollflussflussstrukturen in ADEPT	58
6.2	Sequenz in ADEPT	58
6.3	Datenflussmodellierung in ADEPT	59
6.4	Planbarer Vorwärtssprung	61
6.5	Informationen nachfordern	63
6.6	Informationen nachfordern (dynamischer Fall)	64
6.7	Ausnahmepfad(Abkürzung)	66
7.1	Graphmarkierung für angehaltene/ unterbrochene Aktivitäten	73
7.2	Instanzbearbeitung stoppen	74
7.3	Graphmarkierung zur Validierung einer Aktivität	75
7.4	Arten der Prozesswiederaufnahme	77
7.5	Gestoppte Instanz fortsetzen	78
7.6	Validierung - Markierung setzten	78
7.7	Validierung - Ablauf	80
A.1	Das ADEPT Organisations-Metamodell	85
B.1	Notation um Änderungsmanagement Prozesse darzustellen	90
B.2	ÄM(M)	91
B.3	ÄM(G)	91
B.4	ÄM(StgV)	91
B.5	ÄM(SP)	91
B.6	ÄM(EL)	91
B.7	Änderungsmanagement ÄM(Gv1) Teil1	91
B.8	Änderungsmanagement ÄM(Gv1) Teil2	91
B.9	Änderungsmanagement ÄM(Gv2) Teil1	91
B.10	Änderungsmanagement ÄM(Gv2) Teil2	91
B.11	Änderungsmanagement PVCS:Entwicklung	91
B.12	Prozess für Sonderfunktionen	92

Abbildungsverzeichnis

B.13 Freigabeprozess Version 1	92
B.14 Freigabeprozess Version 2	92
B.15 Freigabeprozess Version 3	92
B.16 Umsetzung von ÄM(EL) in ADEPT	92
B.17 Umsetzung von ÄM(EL) in ADEPT mit Stopp-Aktivität	92

Kapitel 1

Einleitung

1.1 Problembeschreibung

In der Automobilindustrie vollzieht sich ein Wandel. Der Anteil von Software und elektronischen Steuergeräten in Fahrzeugen wird immer größer. Bis zu 90 % der Innovationen eines Fahrzeugs werden demnächst auf Software und elektronischen Steuergeräten basieren. Projekte zur Entwicklung eingebetteter Steuergeräte mit dazugehöriger Software gewinnen daher an Bedeutung. Diese Projekte dauern zwischen drei und fünf Jahre. Bei der Entwicklung werden Prozesse wie das Konfigurations-, Problem- und Änderungsmanagement verwendet. Diese Prozesse werden immer wieder verwendet. Beim Vergleich dieser Prozesse zwischen den jeweiligen Entwicklungsprojekten stellt man Folgendes fest: die verwendeten Prozesse ähneln sich zwar, sind dabei jedoch niemals gleich. Änderungen an den Prozessen finden sowohl im Laufe des Projektfortschritts, als auch zwischen einzelnen Projekten statt. Komplexe Randbedingungen, welche nicht immer sofort ersichtlich sind, beschränken die Änderungen. Diese Prozesse existieren jedoch nicht nur auf dem Papier. Informationssysteme werden eingesetzt, um die Prozesse zugänglich zu machen. Die Prozessbeteiligten werden so bei ihrer Kernaufgabe, der Entwicklung, unterstützt. Wie oben beschrieben sind die Prozesse fortlaufenden Änderungen unterworfen. Die verwendeten Informationssysteme sollten diese Änderungen zeitnah widerspiegeln. Bisherige Informationssysteme zur Prozessunterstützung können dies nur unzureichend leisten.

1.2 Aufgabenstellung

In dieser Diplomarbeit werden zunächst Fallbeispiele aus der Automobilindustrie mit sich entwickelnden Prozessen aufbereitet und dargestellt. Von dieser Grundlage ausgehend werden dann diese Prozesse in ihrer zeitlichen Entwicklung analysiert und verglichen. Ähnlichkeiten und Unterschiede dieser Prozesse bezüglich ihrer Entwicklung werden aufgezeigt. Ziel dieser Analyse ist es, Muster und Abhängigkeiten aufzudecken, sowie deren Ursachen zu ermitteln. In einem nächsten Schritt werden die gefundenen Regelmäßigkeiten kategorisiert und in einer verallgemeinerten Form dargestellt. Darauf aufbauend werden Anforderungen an prozessorientierte Informationssysteme gestellt. Dabei wird jedoch nicht auf ein existierendes System Bezug genommen, sondern von einem idealen System ausgegangen. Ziel dieser Analyse ist es,

Anforderungen an ein prozessorientiertes Informationssystem zu definieren. Diese Anforderungen basieren auf den untersuchten, real existierenden Entwicklungsprozessen. In einem weiteren Schritt soll untersucht werden, inwieweit sich diese Anforderungen mit den Mitteln von Prozesstechnologien umsetzen lassen. Als technologische Grundlage wird dazu das ADEPT-System verwendet. Welche Anforderungen lassen sich direkt, welche über Umwege und welche gar nicht umsetzen? Exemplarisch soll versucht werden, Anforderungen für nicht oder nur schlecht darstellbare Prozessadaptionen zu formalisieren und näher zu untersuchen.

Die Arbeit verfolgt mehrere Ziele. Zuerst soll ein tieferes Verständnis der Anwendung und vor allem der Veränderung von Prozessen und Prozesstechnologien in der Entwicklung eingebetteter Systeme erreicht werden. Des Weiteren sollen Probleme in der Anwendung von Prozesstechnologie erkannt werden, um daraus Anforderungen an prozessorientierte Informationssysteme abzuleiten. Um diese Ziele zu erreichen, wurde das in Abbildung 1.1 auf der nächsten Seite dargestellte und in Abschnitt 1.3 beschriebene Vorgehen gewählt.

1.3 Gewählter Lösungsansatz

Um die Ziele dieser Arbeit zu erreichen, wurde der in Abbildung 1.1 auf der nächsten Seite dargestellte Lösungsansatz gewählt.

In *Kapitel 2 auf Seite 15* werden zuerst die relevanten Grundlagen erarbeitet und dargestellt. Diese bilden den Rahmen dieser Arbeit. Auf den Grundlagen der Organisationstheorie werden Prozesse eingeführt. Deren informationstechnische Umsetzung, sowie deren Grenzen und Einschränkungen, werden daraufhin behandelt.

Kapitel 3 auf Seite 21 beschreibt die Ausgangsbasis dieser Arbeit. Prozesse in Entwicklungsprojekten werden beobachtet. Bei den Prozessen sollen Änderungen und deren Gründe und Ursachen festgehalten werden. Dieses Kapitel bildet einen Teil des Schwerpunkts dieser Arbeit. Die aufbereiteten Prozesse tragen letztendlich auch zu einem besseren Verständnis für den Einsatz von Prozesstechnologien in dem gewählten Umfeld bei.

Die direkte Beobachtung der Prozesse und der Verwendung und Umsetzung in Entwicklungsprojekten sind im zeitlichen Rahmen dieser Arbeit jedoch nicht möglich. Daher werden die Prozesse indirekt beobachtet. Bei DaimlerChrysler zugängliche Dokumente über in Entwicklungsprojekten eingesetzte Prozesse bilden daher den Ausgangspunkt. Daraus werden die Prozesse entnommen, sowie Änderungen, die an diesen stattgefunden haben. Änderungen an Prozessen werden dabei auf die Änderungsursachen hin untersucht. Parallel dazu werden Gespräche mit Prozessexperten von DaimlerChrysler geführt. Ihre Erfahrung dient dazu, die obigen Analysen zu ergänzen und abzurunden. Um Unklarheiten in den Prozessdokumentationen klären zu können, werden ebenfalls die Experten befragt. Durch die Einbeziehung der Prozessexperten findet auch eine erste Kontrolle der aus der Prozessdokumentation gewonnenen Beobachtungen statt. Die vorhandene Dokumentation wird, in Kombination mit den Erfahrungen der Prozessexperten auch dazu verwendet, Probleme,

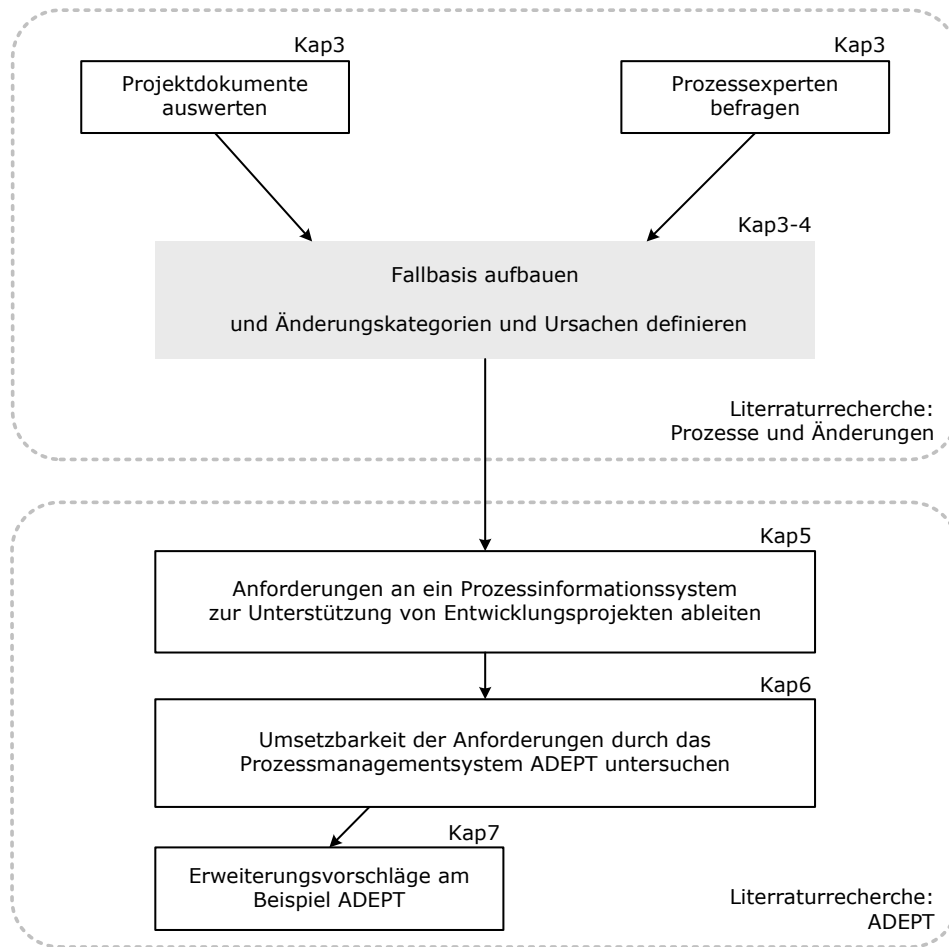


Abbildung 1.1: verwendeter Lösungsansatz; grau hinterlegte Teile definieren den Schwerpunkt der Arbeit

welche im Zusammenhang mit dem Einsatz von prozessbasierte Informationssystemen entstehen, aufzudecken. Diese beiden Schritte werden zum Aufbau einer Fallbasis verwendet.

Kapitel 4 auf Seite 41 ordnet die bisher gesammelten Erkenntnisse. Probleme und wiederkehrende Muster sollen dadurch identifiziert werden. Die dabei gewonnenen Ergebnisse werden nach Änderungskategorien bzw. Ursachen aufgeschlüsselt. Diese Kategorisierung soll später dazu dienen, auf bestimmte Änderungsmuster geeignet reagieren zu können. Dazu werden Kategorien erweitert und angepasst, welche sich in der Literatur über Informationssysteme oder Prozessmanagementsystemen finden.

Kapitel 5 auf Seite 55 enthält Anforderungen an ein Prozessinformationssystem. Diese Anforderungen basieren auf den aus den Fallbeispielen gewonnenen Erkenntnissen. Die Benutzer eines solchen prozessbasierte Informationssystems stehen im Mittelpunkt dieser Betrachtung. Welchen Anforderungen muss ein prozessbasierten

Informationssystemen genügen, um seine Benutzer bei ihrer Aufgabe, hier die Entwicklung eingebetteter Software, zu unterstützen. Dabei muss es von seinen Benutzern als nützlich betrachtet werden und nicht als zusätzliche Belastung im Arbeitsalltag. Diese Anforderungen orientieren sich noch nicht an einem konkreten Prozessmanagementsystem. Dieser Schritt wird erst in Kapitel 6 durchgeführt. Vielmehr wird von einem konkreten System abstrahiert und ein ideales System angestrebt. Diese Anforderungen können verwendet werden, um Prozessinformationssysteme bezüglich ihrer Tauglichkeit für Entwicklungsprojekte zu bewerten. Daher bilden diese einen weiteren Schwerpunkt dieser Arbeit.

Kapitel 6 auf Seite 57 enthält den Versuch, die in Kapitel 4 auf Seite 41 aufgestellten Anforderungen auf ein konkretes Prozessmanagementsystem abzubilden. Dabei wird als Grundlage das Prozessmanagementsystem ADEPT¹ eingesetzt. Es wird untersucht, welche Anforderungen sich mit den Fähigkeiten von ADEPT umsetzen lassen. Ist dies nicht möglich, wird versucht, dies so weit wie möglich zu tun. Durch diese Umsetzungsanalyse wird ermittelt, welche Fähigkeiten eines Prozessmanagementsystems benötigt werden und inwiefern diese vorhanden sind. Es werden also die idealen Anforderungen an einem existierenden System gespiegelt. Das Ergebnis ist eine Analyse des Stands der Technik. Die Ausdrucksmöglichkeit und Fähigkeiten von ADEPT sind das Ergebnis einer Literaturrecherche, sowie Diskussionen mit der ADEPT-Forschungsgruppe.

Kapitel 7 auf Seite 69 zeigt auf, wie die benötigte Flexibilität aus Benutzersicht dargestellt werden kann. Kann die Arbeitsweise der Benutzer nicht adäquat in den prozessbasierten Systeme abgebildet werden, wird dies näher untersucht. Welche Erwartungen stellt der Benutzer an das System, um welche Ziele zu erreichen. Dies basiert auf den untersuchten Fallbeispielen und der Umsetzbarkeitsanalyse. Ziel ist dabei nicht ein konkretes System (ADEPT) zu erweitern, sondern Arbeitsweisen der Benutzer so konkret wie möglich darzustellen.

Kapitel 8 auf Seite 81 rundet diese Arbeit ab. Es fasst die Ergebnisse der vorliegenden Arbeit zusammen und gibt einen Ausblick auf Fragen, die sich durch diese Arbeit ergeben haben oder nicht beantwortet werden konnten.

¹mehr zu ADEPT in Kapitel 6.1 auf Seite 57

Kapitel 2

Übersicht der relevanten Grundlagen

Dieses Kapitel führt den Begriff des (Geschäfts-) Prozesses, sowie seiner informationstechnischen Umsetzung ein. Geschäftsprozesse werden in Abschnitt 2.1 als Teil der betriebswirtschaftlichen Organisationstheorie eingeführt und einer funktionalen Organisation gegenübergestellt. Darauf aufbauend stellt Abschnitt 2.2 die informationstechnische Realisierung von Prozessen dar. Grundlagen des Prozessmanagements werden skizziert. Ergänzend zeigt Abschnitt 2.3 Grenzen und Einschränkungen auf, welche die Umsetzung von Prozessen durch Informationstechnologie limitieren. Dazu wird eine Theorie herangezogen, welche die Rolle von Informationstechnologie in Organisationen beschreibt.

2.1 Grundlagen aus der Organisationstheorie

Die betrieblichen Abläufe eines Unternehmens werden nach einer bestimmten Ordnung durchgeführt. Die Einführung und Durchsetzung dieser Ordnung wird unter dem Begriff Organisation zusammengefasst [Wöh00].

Eine klassische Form der Organisation ist die *Funktionsorientierung* oder *funktionale Organisation*. Betriebliche Faktoren wie Produktion oder Personalwesen bestimmen die letztendlich resultierende Ordnung. Es wird zwischen einer Aufbau- und einer Ablauforganisation unterschieden.

Definition Funktionsorientierung: Die Gliederung eines Unternehmens nach betrieblichen Funktionsbereichen bezeichnet man als Funktionsorientierung oder Funktionsorganisation. Der Weg der Aufträge und Weisungen wird von der Art der betreffenden Aufgaben bestimmt [Wöh00].

Als Erstes wird die Aufbaustruktur festgelegt. Dazu werden gleiche oder ähnliche Aufgaben, welche sich aus den betrieblichen Funktionen ergeben, zu einer Stelle zusammengefasst. Die Stelle bildet den Grundbaustein der Organisation und wird durch Personen besetzt, deren Tätigkeitsgebiet sich dann aus den Aufgaben der Stelle ableiten. Gleichartige Stellen werden zusammengeführt und bilden die funktionalen Bereiche des Unternehmens. Die sich daraus ergebende Arbeitsteilung nach dem Verrichtungsprinzip ergibt eine funktionale Gliederung. Die Organisation enthält Bereiche wie Personalwesen und Beschaffung. Diese sind stark hierarchisch geprägt. Die Gesamtzielsetzung geht verloren und die Erfüllung, der durch den funktionalen Bereich vorgegebenen Aufgaben, tritt in den Vordergrund.

Die Ablauforganisation gestaltet die durchzuführenden Arbeitsprozesse. Sie wird nach der Aufbauorganisation durchgeführt. Handlungsvorgänge werden innerhalb der bereits vorhandenen Aufbauorganisationsstruktur angeordnet. Durch dieses Vorgehen dominiert die Aufbauorganisation die von ihr abhängige Ablauforganisation. Alle hier erläuterten Informationen zur Funktionsorientierung sind im Wesentlichen aus [AS98] entnommen.

Durch die Optimierung der einzelnen Teilaufgaben kann nicht mehr schnell und flexibel auf Änderungen der Rahmenbedingungen reagiert werden. Technologische oder gesetzliche Änderungen können nur langsam umgesetzt werden. Dynamische Märkte sowie die sich ständig wandelnde Infrastruktur der Unternehmen sind Gründe für eine Ausrichtung hin zu einer prozessorientierten Organisation [Mue04]. Diese orientiert sich nicht mehr an den funktionalen Bedürfnissen des Unternehmens, sondern zielorientiert entlang der Wertschöpfungskette.

Definition Prozessorientierung: Ein Unternehmen ist ein System aus miteinander verbundenen Geschäftsprozessen. Ein Geschäftsprozess ist ein Netzwerk aus Ressourcen und Aufgaben, die zur Erstellung einer genau definierten, messbaren Leistung für einen bestimmten Prozesskunden oder zur Herstellung eines bestimmten Zielzustands notwendig sind. Die einzelnen Aufgaben sind über einen durchgängigen Leistungs- bzw. Informationsfluss miteinander verbunden und stehen in definierten Reihenfolgebeziehungen zueinander. Die Abwicklung eines Falls, das heißt einer konkreten Ausprägung eines Geschäftsprozesses, wird durch ein Startereignis ausgelöst [MS96].

Die Organisationsstruktur wird also an den Bedürfnissen des Prozesses ausgerichtet und nicht umgekehrt. Dies grenzt auch die Prozessorientierung gegenüber der Ablauforganisation der Funktionsorientierung ab: Abläufe werden dort durch die Organisationsstruktur der vorherigen Aufbauorganisation vorgegeben. Der zentrale Begriff der Prozessorientierung ist somit der des Geschäftsprozesses:

Definition Geschäftsprozess: Der Prozess ist eine logisch zusammenhängende Kette von Arbeitsvorgängen, die auf das Erreichen eines bestimmten Zieles ausgerichtet sind. Ausgelöst durch ein definiertes Ereignis wird ein Input durch den Einsatz materieller und immaterieller Güter unter Beachtung bestimmter Regeln und der verschiedenen unternehmensinternen und -externen Faktoren zu einem Output transformiert. Der Prozess ist in ein System von umliegenden Prozessen eingegliedert, kann jedoch als eine selbstständige, von anderen Prozessen isolierte Einheit, die unabhängig von Abteilungs- und Funktionsgrenzen ist, betrachtet werden [SK96].

Diese Definition umfasst auch unternehmensübergreifende Prozesse. Prozesse stehen in gegenseitigen Wechselbeziehungen zueinander. Ein Prozess ist keine atomare Einheit. Vielmehr lassen sich Prozesse in weitere Prozessfragmente oder Teilprozesse untergliedern. Diese sind ebenfalls in weitere Teilprozesse gliederbar. Diese strikt

getrennte Darstellung der Funktions- und Prozessorientierung wird in dieser Reinform in der Praxis nicht auftreten. Dennoch ist diese Darstellung nützlich, um die aktuelle Fokussierung der Organisationstheorie (nach [Mue04]) auf Geschäftsprozesse zu verdeutlichen.

2.2 Informationstechnische Umsetzung von Prozessen

Die Informationssysteme der Unternehmen müssen diesen Wandel weg von der Funktionsorientierung, hin zur Prozessorientierung widerspiegeln. Prozessorientierte Informationssysteme realisieren die Prozessorientierung. Die zugrunde liegende technologische Grundlage wird als *Prozess-Management-System (PMS)* bezeichnet. Diese Begriffe werden in [DRR04] mit einer Analogie zu Datenbank-Management-Systemen (DBMS) erklärt. Ein DBMS verwaltet Meta-Daten über Daten: die Relationen-Schemata. Es manipuliert Instanzdaten der Relationen-Schemata, indem es Datentupel einfügt, löscht oder ändert. Die Relationen-Schemata werden durch das DBMS geändert und die Änderungen auf die Dateninstanzen propagiert. Analog dazu verhält sich ein PMS. Es verwaltet Meta-Daten über Prozesse: die Prozess-Schemata. Das PMS ändert den Prozess-Status oder Prozessinstanzvariable und somit die Prozess-Instanz. Die Prozessschemata werden durch das PMS geändert und Änderungen auf die laufenden Prozess-Instanzen propagiert. Beide bilden die technologische Grundlage, um Informationssysteme zu realisieren. Durch ihren Bezug zu einem Anwendungskontext grenzen sie sich von den anwendungsneutralen Managementsystemen ab. Informationssysteme, denen ein explizites Prozessmodell zugrunde liegt, werden als *prozessorientierte Informationssysteme* bezeichnet¹.

Der von einem PMS auszuführende Geschäftsprozess wird in einem *Prozess-Modell* (Schema, Vorlage) beschrieben. Darin werden Tätigkeiten (Aktivitäten), deren Ausführungsreihenfolge und -bedingungen (Kontrollflussaspekt) sowie Informationsflüsse dargestellt. Ebenfalls geregelt wird, wer welche Regeln bearbeiten kann und darf. Das Schema ist ein Modell, ein Abbild des Geschäftsprozesses. Als solches umfasst es alle für die Darstellung von Arbeitsprozessen relevante Daten.

Die Grundlage dieser Beschreibung ist das *Prozess-Metamodell* (Beschreibungssprache). Darin sind die Konstrukte festgelegt, mit denen die Prozessaspekte beschreiben werden. Dies wird durch grafische Beschreibungssprachen realisiert. Rechtecke stellen Aktivitäten und Pfeile die Reihenfolgebeziehung zwischen ihnen dar. Petrinetze (siehe [Aal98]) oder blockbasierte Ansätze (z. B. ADEPT 6.1 auf Seite 57) sind Beispiele für Beschreibungssprachen. Neben den Kontroll- und Datenflüssen sind auch zeitliche Aspekte und organisatorische Regelungen beschreibbar. Prozessschritte (Aktivitäten, Funktionen) werden darin ebenfalls definiert. Man unterscheidet zwischen automatischen und manuellen Aktivitäten. Automatische Aktivitäten erfordern keine Benutzerinteraktion und werden durch das PMS ausgeführt. Manuelle Aktivitäten werden durch prozessbeteiligte Personen ausgeführt. Das PMS bietet dazu allen in Frage kommenden Personen die Aktivität in ihrer Arbeitsliste an. Eine Person startet

¹engl. Process Aware Information System (PAIS); vergleiche [DAH05]

dann die Aktivität, wie die Bearbeitung eines Dokuments oder das Ausfüllen einer Bildschirmmaske. Die Aktivität wird dann aus allen anderen Arbeitslisten entfernt, da zur Bearbeitung einer Aktivität mehrere Personen in Frage kommen können.

Die Aufbauorganisation der Organisation wird in einem *Organisationsmodell* beschrieben. In ihr spiegelt sich die Aufbauorganisation der Organisation wieder. Ausführungs-, Zugriffs- und Änderungsrechte werden auf seiner Grundlage festgelegt. Damit lassen sich Bearbeiterzuordnungen festlegen, d. h. welche Teilmenge der Personen für die Ausführung einer Aktivität in Frage kommt. Eine Bearbeiterzuordnung bezieht sich auf konkrete Personen oder andere organisatorische Einheiten, wie Abteilung oder Arbeitsgruppe. Häufig werden Aktivitäten bestimmten Rollen zugeordnet. Eine Rolle – z. B. Experte oder Softwareentwickler – fasst Fähigkeiten und Kompetenzen zusammen. Ist eine Aktivität zur Ausführung bereit, erscheint sie in den Arbeitslisten der Personen, die mittels der Bearbeiterzuordnung ermittelt wurden.

Eine *Prozess-Instanz* ist eine konkrete Ausprägung eines Prozess-Schemas. Beschreibt ein Prozessschema den Test einer Steuergeräts, dann beschreibt eine Instanz davon, den konkreten Test des Steuergeräts zur Airbagauslösung. Von einem Schema können mehrere Tausend Instanzen gleichzeitig aktiv sein. Eine Instanz kann bis zu mehrere Monate lang aktiv sein. Ein konkretes Beispiel eines PMS und seine Prozessmetamodells findet sich in Kapitel 6.1 auf Seite 57.

2.3 Einschränkungen der Umsetzung

Der informationstechnologischen Umsetzung von Geschäftsprozessen sind Grenzen gesetzt. Orlikowskis Theorie Technologies-in-Practice wird herangezogen, um diese Einschränkungen aufzuzeigen. Diese Theorie wurde von Kleinert, (in [Kle04]), auf die Einführung bzw. Veränderung eines bestehenden Geschäftsprozesses durch ein prozessorientiertes Informationssystem² angewandt. Zunächst wird Orlikowskis Theorie Technologies-in-Practice angerissen und danach die daraus gewonnenen Erkenntnisse dargestellt.

Situiert handelnde Akteure bilden die unterste Ebene. Der Kontext bestimmt und ermöglicht das Handeln der Akteure. Sie werden durch den Kontext beeinflusst, gleichzeitig verändern sie ihn jedoch auch durch ihr Handeln. Der Kontext, als mittlere Ebene, lässt sich durch genau drei Klassen von handlungsbeeinflussenden Faktoren bestimmen: Deutungsschemata, Normen und Ressourcen. Deutungsschemata, wie Annahmen und Wissen, beeinflussen wesentlich die Handlungen der Akteure. Normen sind z. B. Gesetze, sowie Verhaltensregeln und Unternehmenskultur. Ressourcen, wie z. B. Technologien und Macht, bilden die letzte Kategorie. Als Ergebnis entsteht die oberste Ebene: die soziale Struktur. Die Organisation, als soziales System, ist

²Kleinert geht von einem Prozess unterstützendem Informationssystem aus. Darunter versteht er ein Informationssystem, welches einen Geschäftsprozess unterstützt. Von der verwendeten technologischen Grundlage abstrahiert er dabei. Da in dieser Arbeit von einem Prozessmanagement-System als technologischer Grundlage ausgegangen wird, sind seine Ergebnisse übertragbar.

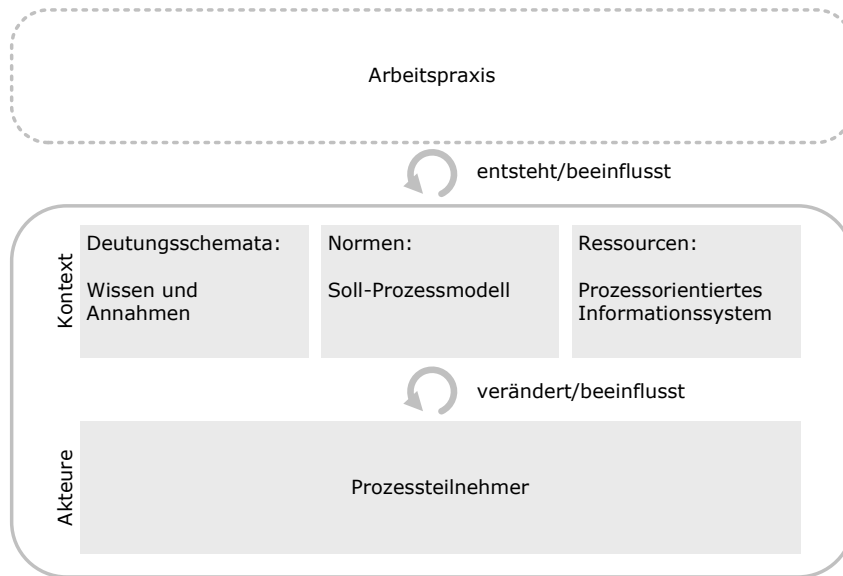


Abbildung 2.1: Technologies in Practice: Die Rolle prozessorientierter Informationssysteme in einer Organisation

nicht-materiell. Sie entsteht aus den Handlungen der Akteure im Kontext und ist auch nur in ihren Handlungen beobachtbar. Der Einfluss der Informationstechnologie wird somit über die Nutzungsweise definiert. Der Einfluss des Informationssystems ist somit nicht deterministisch. Er entsteht abhängig vom jeweiligen Handlungskontext (nach [Kle04]). Diese Ebenen sind in Abbildung 2.1 dargestellt.

Im Rahmen von prozessorientierten Informationssystemen ergibt sich das in Abbildung 2.1 dargestellte Bild. Die Akteure sind (primär) die Prozessteilnehmer. Der geplante Geschäftsprozess (Soll-Prozess) ist eine Norm, prozessorientierte Informationssystem eine Ressource. Das Wissen und die Annahmen der Prozessteilnehmer bzgl. des Prozesses bilden das Deutungsschema. Dies basiert z. B. auf Schulungen der Prozessbeteiligten oder ihrem Wissen wie eine Aufgabe zu erledigen ist. Dadurch wird der Kontext gebildet. Aus den letztendlich instanziierten Ressourcen und Regeln ergibt sich die Arbeitspraxis. Sie enthält alle Aktivitäten, die durchgeführt wurden, um den Soll-Prozess zu erfüllen. Kleinert leitet daraus die im Folgenden vorgestellten Eigenschaften für den Anwendungsbereich von prozessorientierten Informationssystemen ab:

- Der durch ein prozessorientiertes Informationssystem unterstützte Soll-Prozess stellt nur eine Handlungsvorlage dar. Die daraus entstehende Arbeitsweise ist nicht vorhersehbar. (Emergenz)
- Aufgrund der Immaterialität der organisatorischen Abläufe sind diese nur in und durch die Nutzung des prozessorientierten Informationssystems, sowie den Handlungen der Prozessteilnehmer beobachtbar. (Immaterialität)

- Die Arbeitspraxis ist einem ständigen Wandel unterworfen. Jede Handlung verändert den Kontext. Dieser ist Ausgangspunkt nachfolgender Handlungen und verändert somit die daraus resultierende Arbeitspraxis. (Rekursivität)

Kleinert geht nicht von einem Optimum bzgl. des Prozesses aus. Damit kann auch das prozessorientierte Informationssystem nie optimal sein. Sollte es eines geben, müsste sich dies nicht notwendigerweise auf die Arbeitspraxis auswirken. Des Weiteren kann ein Optimum nur von zeitlich begrenzter Dauer sein. Es können sich zwar bestimmte Arbeitsweisen festsetzen, hinreichend starke Kontextänderungen sowie die Rekursivität des Anwendungsgebiets, würden dennoch zu Änderungen führen. Somit muss ständig weiter nach Verbesserungen gesucht oder hinzugelernt werden. Es muss also weder der Prozess noch das prozessorientierte Informationssystem auf ein mögliches Optimum hin ausgerichtet werden. Vielmehr sollte der kontinuierliche und niemals beendete Lernprozess im Spannungsfeld zwischen Management und Arbeitspraxis im Zentrum stehen.

Der durch das prozessorientierte Informationssystem zu unterstützende Soll-Prozess ist also nicht statisch. Hinreichend starke Kontextänderungen machen es nötig, den Soll-Prozess zu ändern. Das Schema des Prozesses muss sich diesen veränderten Bedingungen anpassen. Der Prozess ist nur eine Handlungsvorlage, ein Plan. Treten Änderungen des vorhergesehenen Kontexts auf, muss flexibel von diesem Plan abgewichen werden können. Der Prozess muss an diesen konkreten Fall angepasst werden. Bezogen auf prozessorientierte Informationssysteme bedeutet dies: Sowohl das Schema als auch die Prozessinstanz müssen flexibel änder- und anpassbar sein. Änderungen des Kontexts und deren Auswirkungen werden in Kapitel 3 auf der nächsten Seite näher untersucht. Kapitel 6.1 auf Seite 57 stellt ein Prozessmanagementsystem vor, das versucht die benötigte Flexibilität zu realisieren.

Kapitel 3

Fallbeispiele

In diesem Kapitel werden die Beobachtungen bei der Entwicklung eingebetteter Steuergeräte dargestellt. Ein erster Überblick und der Zusammenhang zwischen den einzelnen Bereichen wird in 3.1 dargestellt. Das Vorgehensmodell zur Entwicklung eingebetteter Steuergeräte wird in 3.2 gezeigt. Näher erläutert wird das Änderungsmanagement in 3.3, das Freigabemanagement in 3.4 sowie die Realisierung von Sonderfunktionen in 3.2.3.

3.1 Übersicht

Das Vorgehen zur Entwicklung eingebetteter Steuergeräte wird im automobilen Umfeld beobachtet. Abbildung 3.1 stellt diesen Rahmen dar. Eine neues Steuergerät, z. B. für eine Motorsteuerung, soll entwickelt werden. Das Projekt zur Entwicklung wird vom Anfang des Projekts bis zum Beginn der Serienproduktion ungefähr drei bis fünf Jahre dauern. Vereinfacht dargestellt läuft die Entwicklung in drei Phasen ab: A-, B- und C-Muster Entwicklung. Parallel dazu finden phasenübergreifende Aktivitäten, sogenannte Querschnittsprozesse statt. Das A-Muster wird entwickelt, um das Konzept, die neue Idee, die hinter der Entwicklung steht, zu überprüfen. Ist das Muster konzepttauglich, wird das B-Muster entwickelt. Darin wird das Steuergerät weiterentwickelt, um in der Serienproduktion verwendet zu werden. Ist es

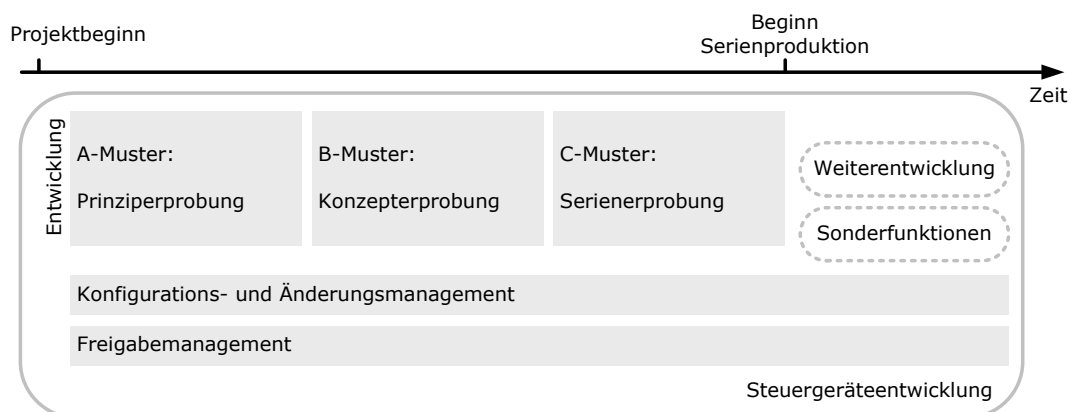


Abbildung 3.1: Überblick der Entwicklung eingebetteter Steuergeräte

serientauglich, wird mit der C-Muster Entwicklung das Vorgehen bis zum Beginn der tatsächlichen Serienproduktion abgeschlossen. Mit dem Beginn der Serienproduktion ist die Entwicklung jedoch noch nicht abgeschlossen. In der realen Verwendung erkannte Probleme müssen behoben werden. Erkannte Verbesserungsmöglichkeiten fließen in eine Weiterentwicklung ein. Spezielle Kundenwünsche und Sonderanfertigungen werden im Sonderfahrzeugbau berücksichtigt. Dazu werden Sonderfunktionen realisiert.

Parallel dazu finden die Querschnittsprozesse des Konfigurations- und Änderungs-, sowie des Freigabemanagements statt. Das Konfigurationsmanagement verwaltet die Konfiguration eines Produkts aus seinen Bestandteilen mit dem jeweiligen Versions- und Änderungsstand. Während der gesamten Entwicklung wird das Steuergerät überarbeitet und verändert. Auch danach müssen aufgetretene Probleme erfasst werden. Das Änderungsmanagement verwaltet und verfolgt die Änderungen und Probleme. Das Konfigurations- und das Änderungsmanagement sind eng miteinander verbunden. Das Freigabemanagement gibt Bauteile zur Verwendung frei, falls diese bestimmte Kriterien erfüllen. Unterschiedliche Bauteile dürfen sich an ihrem späteren Verwendungsort im Fahrzeug nicht räumlich überlappen oder kollidieren. Auch muss sichergestellt werden, dass die einzelnen Bauteile miteinander funktionieren.

Dieser Blick auf die Entwicklung eingebetteter Steuergeräte ist stark vereinfacht. Er soll nur die beobachteten Prozesse miteinander verbinden und in einen größeren Zusammenhang stellen. Nicht betrachtet werden hier weitere Arbeitsabläufe wie Qualitätssicherung, Einkauf und Produktmanagement.

Aufgrund der zur Verfügung stehenden Daten wird ein sehr detaillierter und tiefgehender Blick auf das Änderungsmanagement in 3.3 auf Seite 25 möglich. Es werden die Prozesse aus unterschiedlichen Projekten und teilweise deren Entwicklung innerhalb eines Projekts verfolgt. Auch verwendete Informationssysteme werden berücksichtigt. Das Konfigurationsmanagement wird dagegen nicht weiter angesprochen. Das Freigabemanagement vervollständigt den Blick auf die Querschnittsprozesse 3.4 auf Seite 38. Das allgemeine Entwicklungsvorgehen und seine Auswirkungen auf die Querschnittsprozesse werden in 3.2 aufgezeigt. Als Teil davon wird auch die Sonderentwicklung 3.2.3 auf Seite 24 sowie die Weiterentwicklung 3.2.2 auf der nächsten Seite dargestellt.

3.2 Entwicklungsvorgehen

3.2.1 Beschreibung

Eingebettete Steuergeräte werden nach einem Vorgehensmodell entwickelt. Es ist grob an das V-Modell angelehnt (vergleiche [Bun]). Das V-Modell des Bundes ist ein Leitfaden, um Entwicklungsprojekte zu planen und durchzuführen. So weit dies für das Verständnis der Arbeit notwendig ist, wird es im Folgenden erläutert. Das Entwicklungsvorgehen definiert einen Prozess, um definierte Ergebnisse zu erhalten. Entsprechend der Definition in 2.1 auf Seite 15 ist es somit ein Geschäftsprozess. Er beeinflusst die anderen in diesem Kapitel dargestellten Prozesse und bildet deren

Rahmen.

Drei Phasen werden definiert: A-, B- und C-Muster Entwicklung. Die fortschreitende Benennung des Steuergeräts spiegelt seinen Entwicklungsstand wieder. Das C-Muster ist weiter entwickelt und reifer als das B- und A-Muster. Jede Phase baut auf der Vorherigen auf, verfolgt aber ein anderes Ziel.

Das A-Muster dient der Prinzipierprobung. Die neue Entwicklungsidee wird auf ihre Tragfähigkeit hin überprüft. Entwickelt wird auf einem Simulator. Am Ende dieser Phase befindet sich ein Prüfpunkt. Darin wird die Konzepttauglichkeit des Musters überprüft. Im positiven Fall beginnt die nächste Phase der B-Muster Entwicklung, im negativen Fall wird iteriert.

Das B-Muster dient der Konzepterprobung. Das Steuergerät wird weiterentwickelt, bis es auf die Serientauglichkeit überprüft werden kann. Entwickelt wird dabei nicht mehr auf einem Simulator, sondern auf der Zielhardware. Eine zusätzliche Aktivität wird daher notwendig: Initialisierung der Signale und Kenngrößen des B-Musters. Auch der Hardware-In-the-Loop-Test ist jetzt erst möglich. Die Umgebung, die das Steuergerät einmal steuern soll, wird durch ein Modell dargestellt. Eingaben des Steuergeräts werden durch das Modell geliefert, Ausgaben in das Modell zurückgegeben. Am Ende dieser Phase wird überprüft, ob die Serientauglichkeit erreicht wird.

Das C-Muster dient der Serienerprobung. Am Ende dieser Phase beginnt die Serienproduktion des Steuergeräts und die Entwicklung ist abgeschlossen. Vorgegangen wird dabei wie beim B-Muster. Am Ende wird entschieden, ob die Serienproduktion beginnen kann, oder ob nachgebessert werden muss. Integrationstests können im späteren Gesamtsystem durchgeführt werden.

Die Entwicklung kann damit als inkrementell und iterativ beschrieben werden. Die einzelnen Phasen unterscheiden sich in ihrem Vorgehen und in den Zielen, die sie verfolgen. Das A-Muster wird in einer Simulationsumgebung entwickelt und getestet, das B-Muster in einer Laborumgebung und das C-Muster im späteren Fahrzeug. Je näher der Start der Serienproduktion rückt, umso mehr Prüfungen und Freigaben werden nötig.

3.2.2 Weiterentwicklung

Hat die Serienproduktion begonnen, ist die Entwicklung des Steuergeräts noch nicht abgeschlossen. Die Wartungsphase beginnt, in der Steuergeräte weiterentwickelt und an veränderte Bedingungen angepasst werden. Auch Probleme, die erst beim Einsatz auftreten, müssen jetzt behoben werden (Wartungsphase). Während Änderungen und Anpassungen auf geordnete Art und Weise vorgenommen und entwickelt werden müssen, gilt dies nicht für Probleme. Diese sollten möglichst schnell behoben werden können. Dies widerspricht aber dem Vorgehen in der C-Musterentwicklung, bei der viele Prüfungen und Freigaben nötig sind, um Änderungen einzubringen. Ein anderes Vorgehen wird also nötig. Aus dieser Beschreibung lassen sich bereits Konsequenzen für das Änderungsmanagement ableiten. Es muss die im Vergleich zur C-Muster-Entwicklung geänderten Anforderungen widerspiegeln. Für eine schnelle Problemlösung werden nur die notwendigsten Aktivitäten ausgeführt. Tests und

Freigaben werden hier problemspezifisch verwendet.

3.2.3 Sonderfunktionen

Die Anzahl softwarebasierter Funktionen im Fahrzeugbau steigt. Spezielle Kundenbedürfnisse und -wünsche werden, so weit dies möglich ist, mittels Software realisiert. Die Grundlage dafür bilden programmierbare Steuereinheiten, also eingebettete Steuergeräte. Der Prozess zur Entwicklung dieser Funktionen ist in Abbildung B.12 auf Seite 92 dargestellt. Aufgrund der Individualität der Funktionen werden diese erst nach der Bestellung durch den Kunden entwickelt. Das Fahrzeug wird produziert, während gleichzeitig seine Sonderfunktionen entwickelt werden. Daraus entsteht ein großer Zeitdruck. Der bisher verwendete Prozess der Funktionsentwicklung ist zwar reif (siehe Abbildung B.12(a) auf Seite 92), weist aber Probleme auf.

Die gewünschte Funktion wird entwickelt. Danach werden Testfälle spezifiziert und ein Review durch den Teammanager durchgeführt. Im Erfolgsfall wird danach ein Qualitätsreview sowie ein Systemtest durchgeführt. Werden bei einem der beiden Überprüfungen Mängel entdeckt, wird die Funktionsrealisierung überarbeitet. Abschließend wird die Funktion für das Fahrzeug bereitgestellt. Dazu wird der aktuelle Versionsstand festgehalten und auf das Steuergerät geladen. Folgende Probleme treten auf: Die Testspezifikationen sind an die Realisierung angelehnt, da sie erst nach dieser spezifiziert werden. Dadurch werden keine Fehler entdeckt. Die Testspezifikation entspricht der realisierten Funktion. Diese Arbeitspraxis hat sich festgesetzt, auch wenn die Testspezifikation eigentlich Fehler finden sollte. Die Realisierung wird durch eine manuelle Simulation überprüft. Das Manager-Review wird daher als eine Art erste Überprüfung durch die Implementierer angesehen und missbraucht. Die Qualität der Realisierung, welche in den Reviews geprüft wird, ist daher niedrig. Verzögerungen der Produktion sind die Folge. Um Verbesserungsmöglichkeiten zu finden, wurde eine Prozessanalyse durchgeführt. Als Hauptgrund für die schlechte Performance wurden die vielen Durchläufe der Implementierung und der Review-Schleife genannt. Viele Iterationen wurden benötigt, um eine stabile und qualitativ gesicherte Funktion zu erhalten. Dadurch wurde viel überflüssiger Kommunikationsaufwand betrieben.

Um diese Probleme zu beheben, wurde der Prozess wie in Abbildung B.12(b) auf Seite 92 dargestellt, überarbeitet. Testspezifikationen werden jetzt nicht mehr nach, sondern vor der Entwicklung der Sonderfunktion spezifiziert. Ist die Funktion realisiert (Aktivität **Funktion entwickeln** beendet), wird sie automatisch bezüglich der Testspezifikation geprüft. Nur im Erfolgsfall wird nun daraufhin das Manager Review durchgeführt. Schlägt der Test fehl, wird die Realisierung überarbeitet. Iterationen führen jetzt auch nicht direkt zu einer erneuten Realisierung, sondern die Testfallspezifikation wird überarbeitet.

Der verbesserte Prozess und das zugehörige Prozessinformationssystem wurden zunächst in einer Pilotphase bei ausgewählten Entwicklern getestet. Danach wurde es für alle Entwickler eingeführt. Die Verbesserung brachte folgende Ergebnisse: Die Durchlaufzeit des Prozesses wurde verkürzt. Funktionen werden nicht mehr so oft durch das Manager-Review oder danach abgelehnt. Das Manager Review verliert

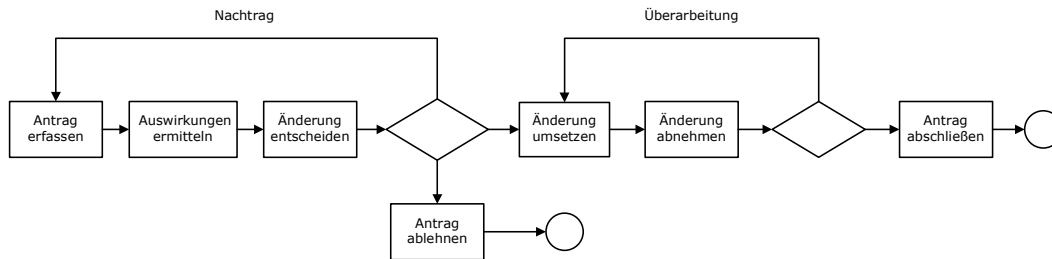


Abbildung 3.2: Prinzipieller Ablauf und Aktivitäten des Änderungsmanagements

dadurch an Bedeutung. Zur weiteren Prozessverbesserung soll es deshalb ersatzlos entfallen. Diese Verbesserung ist bereits geplant, aber noch nicht umgesetzt. Eine Iteration aufgrund eines Fehlers des Qualitätsreviews oder des System-Tests wird zum Ausnahmefall. Die gewünschten Ergebnisse wurden erzielt. Beim Scheitern der Pilotphase oder der Prozessänderungen wären Teile oder die gesamte Prozessänderung zurückgenommen worden. Der Erfolg der Änderungen ist nicht vorhersehbar. Messungen der Durchlaufzeit konnten erst die Wirksamkeit der Verbesserungen bestätigen. Das Manager Review sollte zwar auch entfallen, wird aber immer noch verwendet. Dies ist auf die Arbeitspraxis zurückzuführen, in der sich dieses Vorgehen eingeschliffen hat.

3.3 Änderungsmanagement

3.3.1 Allgemein

Ziel des Problem- und Änderungsmanagements ist es, Probleme und Änderungsanträge zu erfassen, Auswirkungen auf vereinbarte Projekt- und Systemleistungen zu bewerten, Änderungen zu beschließen und umzusetzen. Technologische Probleme, Fehlverhalten des Systems oder Veränderungen des Umfelds sind mögliche Ursachen für Änderungen.

Fehlermeldungen, Problemmeldungen und Verbesserungsvorschläge werden in Form von Änderungsanforderungen¹ oder -anträgen erfasst. Der Antrag und die damit zusammenhängende Änderung besitzen einen bestimmten Status. Dieser ergibt sich aus den getroffenen Entscheidungen und dem Fortschritt der Änderungsumsetzung. Anträge sind z. B. erfasst, in Bearbeitung oder abgeschlossen (weitere Status in B.11 auf Seite 91). Das Änderungsmanagement verwaltet und verfolgt die Anträge und deren Status (nach [Bun]).

Im Folgenden werden einzelne Tätigkeiten näher erläutert (siehe Abbildung 3.2). Als Erstes wird der Änderungswunsch erfasst (*Antrag erfassen*). Dazu beschreibt der Antragsteller das aufgetretene Problem. Weitere Informationen dienen der Identifikation des Antragstellers, des Projekts und der betroffenen Konfiguration. Der Antragsteller (in der Rolle als Auslöser) kann, aus seiner Sicht, noch die Dringlichkeit

¹daher werden im Folgenden nur noch die Begriffe Änderungen bzw. Änderungsanträge oder Anträge verwendet

der Änderung angeben. Im nächsten Schritt werden die Auswirkungen einer möglichen Änderung ermittelt (*Auswirkungen ermitteln*). Technische, finanzielle oder organisatorische Aspekte werden hier berücksichtigt. Danach folgt die Entscheidung über die Änderung (*Änderung entscheiden*). Änderungen werden, unter Berücksichtigung von Lösungsvorschlägen, entweder abgelehnt (*Änderung ablehnen*) oder angenommen. Ist eine Entscheidung aufgrund fehlender oder unklarer Informationen nicht möglich, werden diese mit dem Antragsteller zusammen nachgetragen (Nachtrag). Im Falle einer positiven Entscheidung wird die Änderung nun realisiert (*Antrag umsetzen*). Dies geschieht durch die Stelle mit dem notwendigen fachlichen und systemrelevanten Wissen (in der Rolle als Experte). Danach wird die Umsetzung geprüft (*Änderung abnehmen*). Erfüllt die Umsetzung die gewünschten Qualitätskriterien und setzt den Änderungswunsch um, wird sie abgenommen. Ansonsten wird die Änderung erneut umgesetzt. Die Umsetzung wird also überarbeitet. Danach wird der Änderungsantrag abgeschlossen (*Antrag abschließen*) und damit der Vorgang beendet.

Zwei Rollen spielen beim Änderungsmanagement noch eine Rolle: der Änderungsverantwortliche und die Änderungssteuerungsgruppe (engl. Change Control Board). Der Änderungsverantwortliche analysiert Änderungen, bewertet diese und erarbeitet Lösungsvorschläge und Empfehlungen. Die Änderungssteuerungsgruppe entscheidet bei wichtigen Änderungen und legt das weitere Vorgehen für eine oder mehrere zusammenhängende Änderungen fest. Sie besteht unter anderem aus Vertretern der Entwicklung, des Projektmanagements sowie evtl. Vertretern externen Firmen.

Das Änderungsmanagement erfasst und verwaltet zeitnah alle Änderungsanträge. Es informiert dadurch über den aktuellen Zustand und Bearbeitungsfortschritt aller erfassten Anträge. Betroffene werden über anstehende Aufgaben informiert. Letztendlich werden somit durch das Änderungsmanagement alle Änderungen und Entscheidungen dokumentiert und nachvollziehbar.

3.3.2 Anmerkungen zu Projekten

Projekte unterscheiden sich. Diese Unterschiede wirken sich auch auf die verwendeten Prozesse aus. Die zur Verfügung stehenden finanziellen Mittel und benötigten Mitarbeiter wirken sich auf die Prozesse aus. Neben dem Faktor der Projektgröße lassen sich einfach weitere ausmachen:

- Projektgröße: finanzielle Mittel, benötigte Mitarbeiter
- Projektaufbau: eine oder mehrere interne Organisationseinheiten (Abteilungen) in Kombination mit einen oder mehreren externen Zulieferern/Entwicklern
- Projektziele: z. B. wird ein A-, B-, oder C-Muster entwickelt?
- Erfahrung der Mitarbeiter

Damit kann kein Prozess allen Projekten gerecht werden. Wird nur ein Prozess oder ein PAIS für eine bestimmte Aufgabe verwendet, kann dies daher nur einen Kompromiss darstellen. Dieser Kompromiss ist aber für keine der Projekte ganz angemessen. Zu große Standardisierung oder Vereinheitlichung ist daher schädlich.

Abkürzung	Beschreibung	Art	Aufbauorganisation
ÄM(M)	Motorsteuergerät	Statisch	vorhanden
ÄM(G)	Getriebesteuerung	Statisch	vorhanden
ÄM(Gv1)	Getriebesteuerung	Prozessentwicklung	nein
ÄM(Gv2)	Getriebesteuerung	Prozessentwicklung	nein
ÄM(StgV)	Steuergeräteverbund	Statisch	vorhanden
ÄM(EL)	Bereich EL	Statisch	vorhanden
ÄM(SP)	Projekt SP	Statisch	vorhanden
ÄM(PVCS)	Änderungsmanagement in Serena Changeman Dimensions	Prozessentwicklung	nein

Tabelle 3.1: Übersicht der untersuchten Änderungsmanagement Prozesse

3.3.3 Fallbeispiele

Das Änderungsmanagement (ÄM²) wird über acht Projekte hinweg verglichen. Eine Übersicht der untersuchten Änderungsmanagement Prozesse wird in Tabelle 3.1 dargestellt. Bei fünf Projekten liegt der aktuell verwendete Prozess vor. Bei drei weiteren Prozessen ist zusätzlich die Entwicklung des Prozesses bis zur aktuell verwendeten Version verfügbar. ÄM(G), ÄM(Gv1) und ÄM(Gv2) sind jeweils Änderungsmanagement Prozesse, welche bei der Entwicklung einer Getriebesteuerung verwendet wurden. Jedes dieser Projekte verwendet einen eigenen Prozess. Sie entwickeln ähnliche Funktionen, verwenden dazu aber unterschiedliche Technologische Grundlagen. Weitere Ursachen für Projektunterschiede sind in Abschnitt 3.3.2 auf der vorherigen Seite skizziert. Individuelle Anpassungen sollten bei einem Vergleich hier besonders deutlich hervortreten.

Entwicklung von ÄM(Gv1) und ÄM(Gv2)

Der Änderungsmanagementprozess ÄM(Gv1) wird bei der Entwicklung von Getriebesteuerungen eingesetzt. Er wird damit im gleichen Umfeld wie ÄM(Gv2) und ÄM(G) eingesetzt. Änderungen am Prozess sind über einen Zeitraum von sechs Wochen beschrieben. Daraus lässt sich die Entwicklung des Prozesses und teilweise auch die Gründe, die zu den Änderungen führten, nachvollziehen. Die Entwicklung von ÄM(Gv1) wird in Abbildung B.7 auf Seite 91 und B.8 auf Seite 91 dargestellt. Die Entwicklung von ÄM(Gv2) wird in Abbildung B.9 auf Seite 91 und B.10 auf Seite 91 dargestellt. Neben den Gesprächen mit Prozessexperten beruht die Darstellung auch auf [JSS⁺04] und [JR04].

Die Entwicklung von ÄM(Gv1) wird durch die Abbildung B.7(a) auf Seite 91 als

²ÄM(x) bezeichnet das Änderungsmanagement von x. ÄM(G) ist beispielsweise das Änderungsmanagement eines Projekts zur Entwicklung einer Getriebesteuerung (abgekürzt G)

Ausgangspunkt und durch Abbildung B.7(b) auf Seite 91 und Abbildung B.8(c) auf Seite 91 als Zwischenpunkte beschreiben. Der Endpunkt wird in Abbildung B.8(d) auf Seite 91 gezeigt. Änderungen des Prozesses wurden in Treffen zwischen Prozessbeteiligten und Prozessexperten beschlossen und anschließend umgesetzt. In einem ersten Treffen, vor Projektbeginn, wurde der Prozess in Abbildung B.7(a) auf Seite 91 nach den Wünschen der Prozessbeteiligten angepasst (vgl. B.7(b) auf Seite 91). Die Ursachen dieser Änderungen können nicht mehr nachvollzogen werden. Ab diesem Zeitpunkt wird der Prozess in dem Projekt verwendet. Nach zwei Wochen fand ein Treffen statt um den Prozess zu bewerten. Die Prozessbeteiligten wurden gebeten Verbesserungsvorschläge und ihre positiven Erfahrungen zu sammeln. Die Prozessbeteiligten nannten folgende Vorteile der Unterstützung durch ein prozessorientiertes Informationssystem:

- Das PAIS ermöglicht die einheitliche Verwaltung der Anträge.
- Der Überblick über noch zu bearbeitende Anträge und deren Arbeitsfortschritt sowie die automatische Koordinierung der Prozessbeteiligten.

Diese Sicht der Prozessbeteiligten bestätigt die Nützlichkeit der Umsetzung des Änderungsmanagements in einem PAIS. Folgende Probleme wurden genannt:

- Die Aktivität **Antrag zuordnen** legt die Bearbeiterzuordnung der folgenden Schritte fest. Je nachdem, wo die Ursachen des Problems liegen (Software, Elektronik, Aktuator), wird die entsprechende Organisationseinheit, bzw. die Rolle mit den relevanten Fähigkeiten ausgewählt. Ein Antrag wird häufig falsch zugeordnet. Diese Zuordnung muss also aufgehoben und erneut getroffen werden.
- Ist ein Antrag nicht korrekt umgesetzt (**Umsetzung prüfen**), wird die Aktivität **Antrag umsetzen** erneut durchgeführt. Die Umsetzung wird überarbeitet. Dieses Verhalten soll nun geändert werden. Bevor der Antrag umgesetzt wird, soll der Antrag erneut klassifiziert und bewertet werden (**Antrag klassifizieren und bewerten**). Darin legt das Change Control Board fest, ob der Antrag wichtig genug ist, um sofort erneut umgesetzt zu werden, oder ob andere Anträge wichtiger sind. Der Experte soll zuerst die wichtigeren Anträge erledigen. Die Anträge werden in Bezug auf einen festgelegten Zeitpunkt (Meilenstein) priorisiert.

Diese Prozessvariante wurde nun für weitere vier Wochen eingesetzt. Danach fand erneut ein Treffen statt und ein Problem wurde identifiziert. Der Antrag enthält nicht genug Informationen, bzw. keine eindeutigen Informationen. Damit wird die Bearbeitung des Antrags erschwert, falsch durchgeführt oder ist gar nicht möglich. Daher sollen diese Unklarheiten durch den Antragsteller in **Antrag erstellen** geklärt werden. Diese Aktivität wird dann erneut ausgeführt (Iteration). Daraus ergibt sich der Prozess in Abbildung B.8(d) auf Seite 91. Danach sind keine Änderungen mehr dokumentiert.

Die Entwicklung von ÄM(Gv2) beginnt mit einem sehr einfachen Prozess mit **Antrag erstellen** und **Antrag abschließen**. Die Änderungsgründe und das Vorgehen der

Änderungen waren nicht dokumentiert. Mit der Hilfe eines Prozessexperten konnten Ursachen teilweise rekonstruiert werden. Die zweite Prozessvariante (Abbildung B.9(b) auf Seite 91) entspricht einer Prozessvariante des ÄM(Gv1) (Abbildung B.7(a) auf Seite 91). Nimmt man diese Variante als Ausgangspunkt, lässt sich die Entwicklung parallel verfolgen. Änderungen an diesen Prozessen führen zu Abbildung B.10(c) auf Seite 91. Diese sind identisch mit ÄM(Gv1) Variante 2. Die nächsten Prozessvarianten unterscheiden sich. Auch in ÄM(Gv2) tritt das Problem mit nicht ausreichend detaillierten Beschreibungen des Antrags auf. Die gewählte Lösung unterscheidet sich jedoch. Hier wird davon ausgegangen, dass fehlende Informationen erst bei der genaueren Analyse (**Antrag analysieren**) auffallen. Hier kann jedoch nicht einfach die Aktivität **Antrag erfassen** erneut ausgeführt werden, da sonst die Aktivität **Antrag zuteilen** auch erneut durchlaufen würde. Deren Ergebnisse sind aber noch gültig. Um unnötigen Doppelaufwand zu vermeiden, wird daher die Aktivität **Rücksprache** eingeführt. Scheitert eine der beiden Prüfungen der Umsetzung (**Umsetzung prüfen** und **Integration prüfen**) wird hier direkt wieder die Aktivität **Antrag umsetzen** ausgeführt. Es entsteht ein Prüfzyklus um die Aktivität, die den Antrag ausführt. Hier priorisiert das Change Control Board die Anträge nicht in jedem Prüfzyklus erneut.

ÄM(PVCS)

Der im Folgenden beschriebene Prozess ist mittels Serena Changeman Dimensions realisiert. Serena Changeman Dimension versteht sich als Process Driven Closed-Loop Change Management. Die hier betrachteten Prozesse dienen dem Problem- und Änderungsmanagement. Der Prozess wird dabei durch Zustände und Zustandsübergänge beschrieben, die ein Änderungsdokument durchläuft. Das Änderungsdokument definiert alle für einen Änderungsantrag relevanten Daten. Die Entwicklung des Zustandsautomaten liegt vollständig bis zur geplanten, aber noch nicht eingesetzten Version vor. Alle Versionen sind in B.11 auf Seite 91 abgebildet. Bei diesem Änderungsmanagement Prozess können vier Versionen des Prozesses untersucht werden. Durch eine Auswertung des zugrunde liegenden Systems können auch die tatsächlich aufgetretenen Statusübergänge ermittelt werden. Diese können von den definierten Statusübergängen abweichen: ein Benutzer mit den entsprechenden Rechten darf ein Änderungsdokument in jeden beliebigen Zustand setzen, Zustände können direkt in der Datenbank verändert werden.

Befindet sich ein Änderungsdokument im Status **deferred** (engl. zurückgestellt oder aufgeschoben), wird die Bearbeitung ausgesetzt. Dieser Zustand wird verwendet, wenn es zu dem entsprechenden Zeitpunkt nicht möglich ist, eine Entscheidung über das weitere Vorgehen zu treffen. Auch, wenn auf eine andere Entscheidung gewartet wird, oder entschieden worden ist, die Bearbeitung auf einen späteren Zeitpunkt zu verschieben. Befindet sich ein Änderungsdokument im Endstatus **killed**, wird dadurch verdeutlicht, dass der Antrag abgelehnt wurde. Damit unterscheidet er sich vom jeweiligen anderen Endzustand. In Abbildung B.11(a) auf Seite 91 ist **closed** der normale Endzustand. Der Zustand **killed** wird gewählt, wenn das Problem kein

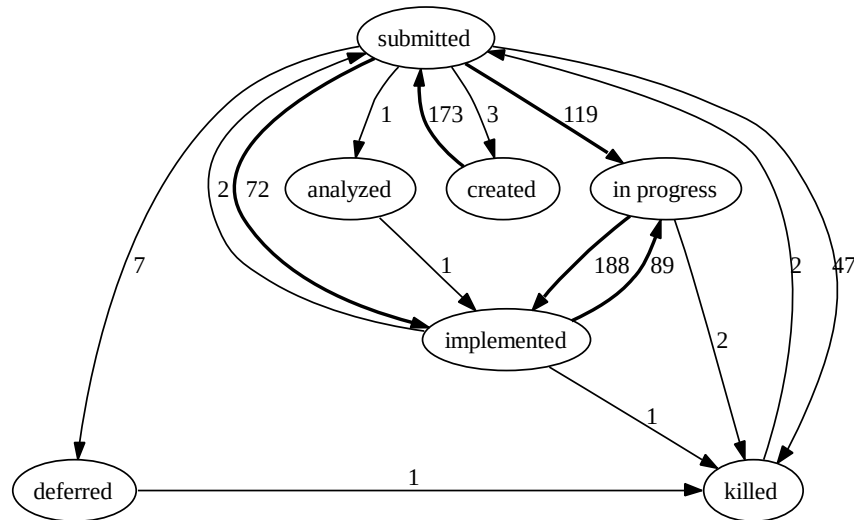


Abbildung 3.3: Real aufgetretene Zustandsübergänge beim ÄM(PVCS)

Problem ist, oder entschieden wird, nicht darauf zu reagieren. Dagegen bezeichnet der normale Endzustand, dass die Arbeit an der Änderung erfolgreich abgeschlossen ist. Die Bearbeitung einer Änderung wird also durch **deferred** unterbrochen, um später wiederaufgenommen zu werden und durch **killed** vorzeitig beendet.

In B.11(a) sind diese beiden Zustände nur von **submitted** erreichbar. Das vorübergehende Aussetzen des Ablaufs (**deferred**) und sein vorzeitiger Abbruch (**killed**) sind jedoch während des gesamten Ablaufs nützlich. Daher wurden die entsprechenden Zustandsübergänge eingefügt (ab B.11(b)). Nur aus dem Endzustand sollen sie nicht erreichbar sein, da dies semantisch keinen Sinn ergibt. **Deferred** und **killed** werden aus jedem Zustand erreichbar. Die Antragsbearbeitung kann somit zu jedem Zeitpunkt unterbrochen bzw. abgebrochen werden. Wird ein Änderungsantrag abgebrochen, sollen seine Daten nicht verloren gehen. Es ist wichtig zu dokumentieren, dass und warum er abgebrochen wurde. Auch der Zeitpunkt des Abbruchs ist interessant. Abbildung 3.3 stellt die tatsächlichen Statusübergänge dar. Grundlage ist dabei der Zustandsautomat B.11(c). Die neuen Zustände und Übergänge treten nicht so deutlich hervor, da der Zustandsautomat noch nicht lange verwendet wird. Der Endzustand ist **implemented**. Obwohl kein Übergang vorgesehen ist, wurden 89 Änderungsdokumente wieder nach **in progress** zurückgeschaltet. Ein bereits erfolgreich abgeschlossener Antrag wird erneut be- oder überarbeitet. Mögliche Gründe hierfür können eine fehlerhafte Umsetzung oder unerwünschte Nebenwirkungen sein.

Übergänge von **submitted** nach **implemented** sind nicht geplant, treten aber 72 mal auf. Dieser kommt vor, wenn eine Änderung eine Untermenge einer zweiten Änderung ist. Wird diese zweite Änderung umgesetzt, ist die Erste damit auch erledigt. Der Änderungsantrag muss also nicht weiterbearbeitet werden, sondern muss als abgeschlossen markiert werden. Der Übergang von **killed** nach **submitted** wird benötigt, um abgebrochene Anträge wieder bearbeiten zu können, beispielsweise wenn

die Entscheidung zum Abbruch von einer Person mit mehr Rechten revidiert wird.

Die Zustände `deferred` und `killed` können als Entscheidungszustände betrachtet werden, die parallel zu den Zuständen der normalen Bearbeitung stehen. Das Zustandsmodell sieht als normale Bearbeitungsreihenfolge `submitted` → `analyzed` → `ready for implementation` → `in progress` → `implemented` vor (vgl. B.11(c)). Dieser sequenzielle Ablauf ist in machen Fällen aber im Weg. Ein Entwickler (Experte) implementiert gerade einen Antrag. Beim Betrachten der anderen Anträge (`submitted`) erkennt er, dass ein weiterer Antrag Teil seines Antrag ist. Wenn sein Antrag also implementiert ist, ist damit auch der andere Antrag erledigt. Er könnte jetzt den Antrag durch jeden einzelnen Zustand schalten. Was passiert ist, dass er auf einer Papierliste vermerkt, dass der zweite Antrag auch erledigt ist. Ein inkonsistenter Zustand zwischen Realität und Prozess entsteht. Nach einiger Zeit akkumulieren sich solche Abweichungen. Um wieder einen konsistenten Zustand zu erreichen, wird der Zustand manuell in die zugrunde liegende Datenbank eingetragen. Neben der Fehleranfälligkeit dieser Arbeitsweise entstehen weitere Probleme. Auswertungen, z. B. wie lange die Anträge in jedem Zustand waren, werden dadurch wertlos. Der Prozess gibt nicht mehr die Realität wieder. Der Grund dafür ist Folgender: das prozessgestützte Informationssystem ermöglicht keine flexible Arbeitsweise. Dies tritt dann auf, wenn der Prozess nicht der Auslöser der Arbeit ist.

Wenn man die Zustandsübergänge einzelner Änderungsdokumente betrachtet, erhält man Anträge, die dem Prozess aus `submitted` → `in progress` → `implemented` folgen. Teilweise wird der Zyklus `in progress`/ `implemented` bis zu vier mal durchlaufen. Aber auch die Folge `submitted` → `implemented` tritt auf. Dieser Übergang ist eigentlich nicht vorgesehen, spiegelt aber die benötigte Flexibilität in der Bearbeitung der Änderungsanträge wieder.

Änderungen des Prozessschemas mussten von Hand vorgenommen werden. Beim Wechsel von Version B.11(a) auf Seite 91 auf B.11(b) auf Seite 91 wurden zwei Status entfernt: `created` und `closed`. Zu diesem Zeitpunkt befanden sich schon Änderungsdokumente in allen Status. Um das Prozessschema ändern zu können, durften sich keine Prozessinstanzen in einem der beiden Status befinden. Aus dem Status `created` konnte ein berechtigter Nutzer manuell in den nächsten Status schalten. Beim Endstatus musste die Änderung direkt in der Datenbank vorgenommen werden. Da zu diesem Zeitpunkt nur relativ wenige Instanzen existierten, konnte der manuelle Arbeitsaufwand bewältigt werden. Bei mehr Instanzen (einige tausend) wäre dieser Aufwand nicht mehr vertretbar gewesen. Mit fortlaufender Benutzung des Systems, erschwert dieses Änderungen.

3.3.4 Auswertung

Wertet man die obigen Prozesse, ihre Änderungen und ihre informationstechnische Umsetzung aus, lassen sich detaillierte Aussagen treffen. Das Änderungsmanagement kann als spezieller Geschäftsprozess charakterisiert werden (angelehnt an [Kle04]): er kann viele organisationsspezifische Varianten annehmen. Ein Projekt, als zeitliche begrenzte Organisation mit definiertem Ziel, erfüllt diese Bedingung ebenfalls.

Organisationseinheit	Änderungsmanagement (ÄM) des Projekts				
	ÄM(M)	ÄM(G)	ÄM(StgV)	ÄM(EL)	ÄM(SP)
Auslöser	•	•	•	•	•
Änderungs- Verantwortlicher	•		•	•	•
Experte	•	•	•	•	•
System-Experte		•			
System-Bereich			•		
Fahrversuch-Bereich			•		
Change Control Board	•	•	•	•	
Gesamt-Änderungs- Verantwortlicher		•			
Gesamt-Fahrzeug- Verantwortlicher			•		
Release Manager					•
Tester					•

Tabelle 3.2: Übersicht der Rollen pro Projekt

Alle betrachteten Änderungsmanagement-Prozesse wurden an das jeweilige Projekt angepasst. Jedes Projekt versucht die für sich geeignete Arbeitsweise zu finden. Gemeinsamkeiten entstehen aus einem Grundverständnis über die in diesem Kontext angebrachte Vorgehensweise.

Projektspezifische Varianten Insgesamt acht unterschiedliche Varianten eines Änderungsmanagement Prozesses können miteinander verglichen werden. Alle werden im gleichen Unternehmen eingesetzt. Unterschiede können daher, zumindest teilweise, auf unterschiedliche Projektcharakteristika zurückgeführt werden.

Tabelle 3.2 vergleicht das verwendete *Organisationsmodell* der Projekte miteinander. ÄM(Gv1) und ÄM(Gv2) verwenden ein zu ÄM(G) ähnliches Modell mit Änderungsverantwortlichem, Change Control Board und Experte. Dazu sind aber keine Details vorhanden, so dass hier auf eine nähere Betrachtung verzichtet wird. Rollen wie Auslöser, Änderungsverantwortlicher (ÄM(G) in Form des Gesamtänderungsverantwortlichen) und das Change Control Board werden über Projektgrenzen hinweg verwendet. ÄM(SP) definiert ein etwas anderes Organisationsmodell mit den Rollen Tester und Release Manager. ÄM(StgV) definiert zusätzlich den Systembereich, einen Fahrversuchsbereich sowie die Rolle Gesamtfahrzeugverantwortlicher. ÄM(G) einen Gesamtänderungsverantwortlichen und den Systemexperten. Diese speziellen

Rollen lassen sich durch Projekteigenschaften erklären. Die Abnahme der Umsetzung durch den Fahrversuchsbereich deutet darauf hin, dass dieser Prozess gegen Ende einer Entwicklung eingesetzt wird. Auch müssen die Anträge mit anderen Steuergeräteentwicklungen koordiniert werden. Daher trifft ein Verantwortlicher, der das gesamte Fahrzeug überwacht, hier Entscheidungen. Ebenso bei ÄM(G). Dadurch unterscheiden sich diese beiden Projekte von den anderen. Dort werden keine übergreifenden Rollen definiert. Sie sind nicht in einen größeren Kontext eingebettet, oder dieser wird nicht berücksichtigt. Diese Prozesse finden sich eher am Anfang des Entwicklungszyklus wieder.

Tabelle 3.3 auf der nächsten Seite stellt dar, welche *Aktivitäten* in welchen Projekten verwendet werden. Auffällig ist dabei die große Übereinstimmung der Aktivitäten. Nur in zwei Projekten werden einzigartige Aktivitäten verwendet. Bei ÄM(SP) sind dies: **Ablehnung entscheiden** und **Testinformationen bereitstellen**. Spezielle Aktivitäten von ÄM(EL) sind: **Auswirkungen ermitteln**, **Antrag stellen** und **Auswirkungen bewerten**. Alle anderen Aktivitäten werden in mindestens zwei (Bsp. **Zuteilung Entwicklungszyklus** in ÄM(M) und ÄM(EL)), größtenteils jedoch in fast allen verwendet (**Antrag umsetzen** oder **Umsetzung prüfen**). Aktivitäten lassen sich also in projektspezifische und allgemeine Aktivitäten unterteilen. Projektspezifische Aktivitäten sind jedoch selten.

Tabelle 3.4 auf Seite 35 stellt dann die Verbindung zwischen dem Organisationsmodell und den verwendeten Aktivitäten her: die *Bearbeiterzuordnung*. Da bei ÄM(Gv1) und ÄM(Gv2) keine Bearbeiterzuordnungen verfügbar sind, werden diese Projekte hier ignoriert. Wird eine Aktivität mehrmals von unterschiedlichen Teilen der Organisationseinheit ausgeführt (z. B. **Antrag ablehnen** in ÄM(EL) wird einmal vom Change Control Board und einmal vom Änderungsverantwortlichen bearbeitet), werden alle Organisationseinheiten aufgeführt. Ist eine Aktivität in jedem Projekt der gleichen Rolle zugeordnet (z. B. **Antrag erstellen** vom Auslöser oder **Antrag umsetzen** von Experten), wird sie nicht aufgeführt. Ebenso die Aktivitäten, die nur in einem Projekt verwendet werden.

Das Ergebnis der Auswertung ist folgendes: jedes Projekt besitzt ein eigenes Verständnis davon, wer für eine Aktivität verantwortlich ist und diese durchführen sollte. **Antrag zuordnen** wird in ÄM(G) vom Systemexperten, in ÄM(StgV) vom Gesamtfahrzeugverantwortlichen und in ÄM(SP) vom Auslöser bearbeitet.

Jedes Projekt hat eine unterschiedliche Auffassung davon, wie die Aufgabe des Änderungsmanagements am effizientesten zu erledigen ist. Oder historisch gewachsene Arbeitsweisen des Anwendungsbereichs oder der Prozessteilnehmer sollen im Änderungsmanagement, so gut es geht, weiter unterstützt werden (Aussage: „Bei uns macht man das so!“). Wie schon beobachtet, unterscheiden sich die verwendeten Aktivitäten nur wenig. Die Reihenfolge der Aktivitäten unterscheidet sich jedoch. Bei ÄM(G) und ÄM(StgV) wird zuerst **Antrag zuordnen** und danach **Antrag analysieren** durchgeführt. ÄM(SP) kehrt diese Reihenfolge um. Die Zielsetzung, zuerst den Antrag zuzuordnen und danach detaillierte Analysen durchzuführen oder genau andersherum, wird kurz skizziert. Wird der Antrag zuerst zugeordnet, erhält die Abteilung mit dem nötigen Fachwissen die Aufgabe, detaillierte Analysen durchzuführen. Die

Aktivität	Änderungsmanagement (ÄM) des Projekts				
	ÄM(M)	ÄM(G)	ÄM(StgV)	ÄM(EL)	ÄM(SP)
Antrag erfassen	•	•	•	•	•
Antrag analysieren	•	•	•		•
Antrag bewerten	•	•	•	•	•
Antrag ablehnen	•	•		•	•
Antrag umsetzen	•	•	•	•	•
Antrag abschließen	•	•	•	•	•
Antrag zuordnen		•	•		•
Antrag klassifizieren und bewerten		•			
Antrag stellen				•	
Rücksprache führen	•		•	•	•
Zuteilung Entwicklungszyklus	•			•	
Umsetzung prüfen	•	•	•	•	•
System abnehmen		•	•		
Auswirkungen ermitteln				•	
Auswirkungen bewerten				•	
Ablehnung entscheiden					•
Testinformationen bereitstellen					•

Tabelle 3.3: Verwendung von Aktivitäten in den einzelnen Projekten

Aktivität	Änderungsmanagement (ÄM) des Projekts				
	ÄM(M)	ÄM(G)	ÄM(StgV)	ÄM(EL)	ÄM(SP)
Antrag analysieren	Änderungs- Verantwortlicher	Experte	Änderungs- Verantwortlicher		Bug-Tracker
Antrag bewerten	Change-Control- Board		Change-Control- Board	Änderungs- Verantwortlicher	
Antrag ablehnen	Change-Control- Board	Change-Control- Board		Change-Control- Board; Änderungs- Verantwortlicher	Bug-Tracker
Antrag zuordnen		System-Experte	Gesamtfahrzeug- Verantwortlicher		Auslöser; Bug-tracker
Umsetzung prüfen	Änderungs- Verantwortlicher	Experte	System-Bereich	Änderungs- Verantwortlicher	Tester
System abnehmen		System-Experte	Fahrversuch- Bereich		
Antrag abschließen	Änderungs- Verantwortlicher	System-Experte; Gesamt-Änderungs- Verantwortlicher	Änderungs- Verantwortlicher	Änderungs- Verantwortlicher	Qualitäts-Manager Tester

Tabelle 3.4: Bearbeiterzuordnung einzelner Aktivitäten in Abhängigkeit des Projekts: nicht enthalten sind Aktivitäten die nur in einem Projekt verwendet werden, oder in jedem Projekt die selbe Bearbeiterzuordnung besitzen

Projekt	Prüfzyklus	
	Beginn	Ende
ÄM(M)	Zuteilung Entwicklungszyklus	Umsetzung prüfen
ÄM(G)	Antrag klassifizieren und bewerten	Umsetzung prüfen System abnehmen
ÄM(Gv1)	Antrag klassifizieren und bewerten	Umsetzung prüfen Integration prüfen
ÄM(Gv2)	Antrag umsetzen	Umsetzung prüfen Integration prüfen
ÄM(StgV)	Antrag umsetzen	Umsetzung prüfen System abnehmen
ÄM(EL)	Zuteilung Entwicklungszyklus	Umsetzung prüfen
ÄM(SP)	Antrag umsetzen	Umsetzung prüfen

Tabelle 3.5: Projektspezifische Prüfzyklen. Zwei Aktivitäten in der Spalte *Ende* bedeuten zwei Prüfzyklen, die zu der Aktivität in der Spalte *Beginn* zurückkehren

Zuordnung zu einer bestimmten Abteilung zu treffen ist jedoch nicht immer einfach. Diese Fehlzusordnungen erzeugen in den Abteilungen das Gefühl, unnötige Arbeit zu erledigen. Die Lösung ist genau das umgekehrte Vorgehen. Doch auch dieses ist nicht ohne Probleme. Wird der Antrag zuerst analysiert, muss dies von einem Experten durchgeführt werden, der sich in allen relevanten Bereichen auskennt. Diese spezialisierten Mitarbeiter bei einer ersten Analyse einzusetzen, nur um dann später gleich die richtige Abteilung auszuwählen, erzeugt viel zusätzlichen Arbeitsaufwand unter Verschwendung wichtiger Ressourcen.

Ein weiterer Unterschied ist das gewünschte Verhalten, falls der Test der Umsetzung fehlschlägt. Das Ergebnis der Aktivität **Antrag umsetzen** wird durch eine oder mehrere Aktivitäten geprüft (**Umsetzung prüfen**, **System abnehmen**, **Integration prüfen**). Je nach Projekt unterscheidet sich die Anzahl der Prüfungen und wie nach dem Fehlschlagen der Prüfungen weiter vorgegangen wird. Im Folgenden wird die Kombination aus prüfender Aktivität und Aktivität, die nach dem Fehlschlagen der Prüfung durchgeführt wird, Prüfzyklus genannt. ÄM(M) besitzt einen Prüfzyklus mit **Umsetzung prüfen** und **Zuteilung Entwicklungszyklus**. ÄM(G) hat zwei Prüfzyklen aus **Umsetzung prüfen** und **System abnehmen**, die zu **Antrag klassifizieren und bewerten** zurückkehren. Tabelle 3.5 stellt die verwendeten Prüfzyklen in Abhän-

Projekt	Abbruchaktivität	Rolle
ÄM(M)	Antrag analysieren Antrag bewerten	Änderungsverantwortlicher Change Control Board
ÄM(G)	Antrag zuordnen	Systemexperte
ÄM(Gv1)	Antrag zuordnen Antrag analysieren	
ÄM(Gv2)	Antrag klassifizieren und bewerten	
ÄM(StgV)	Antrag analysieren	Änderungsverantwortlicher
ÄM(EL)	Antrag bewerten Auswirkungen ermitteln Auswirkungen bewerten	Änderungsverantwortlicher Change Control Board Change Control Board
ÄM(SP)	Antrag analysieren	Bug Tracker
ÄM(PVCS)	jede	jew. Bearbeiter

Tabelle 3.6: vorzeitiger Abbruch der Prozessbearbeitung

gigkeit des Projekts dar. Diese drücken jeweils eine unterschiedliche Semantik aus. Wird nach dem Fehlschlagen einer Prüfung direkt wieder versucht, diese umzusetzen (zu überarbeiten), genügt die Umsetzung nicht den gewünschten Qualitätskriterien, Anforderungen oder kann nicht mit anderen Anträgen integriert werden. Wird dagegen eine Entscheidungsaktivität ausgeführt, wie **Antrag klassifizieren und bewerten**, kann dies zum Abbruch der Bearbeitung führen. Der gewählte Lösungsansatz liefert nicht die gewünschten Ergebnisse oder würde den gesetzten Zeitrahmen überschreiten. Beide haben ihre Berechtigung. Tendenziell wird am Anfang des Entwicklungszyklus versucht, die Umsetzung zu überarbeiten. Schließlich soll ja die neue Funktionalität entwickelt und demonstriert werden. Gegen Ende wird eher erst entschieden werden müssen, ob eine Umsetzung noch lohnt oder gar den Projekterfolg gefährden würde.

In 3.3.3 auf Seite 29 hat sich gezeigt, dass es wünschenswert ist, den Prozess zu jedem beliebigen Zeitpunkt (vorzeitig) zu beenden. Die anderen beobachteten Änderungsmanagement Prozesse (ÄM(G) etc.) setzen dies nicht um. Ein vorzeitiges Beenden durch die Aktivität **Antrag ablehnen** und ein früheres Ausführen der Aktivität **Antrag abschließen** ist nicht in jedem möglichen Punkt vorgesehen. Es existieren bestimmte Aktivitäten, bei denen dies möglich ist. Diese Entscheidungspunkte werden von unterschiedlichen Bearbeitern ausgeführt. Wie sich in Tabelle 3.6 zeigt, unterscheidet sich diese Entscheidungsbefugnis von Projekt zu Projekt. In ÄM(StgV) darf diese Entscheidung nur der Änderungsverantwortliche treffen. ÄM(G) erlaubt dagegen den Abbruch durch den System-Experten und die normale Beendigung durch den Gesamtänderungsverantwortlichen.

Schlussfolgerung Obwohl diese Beobachtungen nur über acht Projekte hinweg gemacht worden sind und nur im Rahmen des Änderungsmanagements, lassen sich daraus Erkenntnisse ableiten. Um einen Prozess an die Bedürfnisse eines speziellen Projekts anzupassen, sind nicht viele projektspezifische Aktivitäten zu realisieren. Vielmehr lässt sich mit einem gemeinsamen Grundstock von Aktivitäten, ein großer Bereich, der von den Projekten gewünschten Funktionalitäten abdecken. Unterschiede liegen im Organisationsmodell und den Mitarbeiterzuordnungen. Darauf aufbauend lässt sich dann der von den Projekten gewünschte Kontrollfluss realisieren.

3.4 Freigabemangement

Jedes Bauteil bei der Entwicklung eines Fahrzeugs, bzw. seine geometrische Beschreibung, durchläuft eine Menge von Freigaben, um die Baubarkeit, Funktionsfähigkeit und Qualität des Fahrzeugs sicherzustellen. Die Serienfreigabe regelt, ob ein Bauteil bei der Serienproduktion verwendet werden darf. Bei der Packagingfreigabe wird überprüft, ob Bauteile an ihrem Bestimmungsort mit Nachbarbauteilen kollidieren oder Abstandsgrenzen nicht einhalten. Die Packagingfreigabe wird hier als Muster für einen Freigabeprozess verwendet, obwohl sie nicht direkt in der Entwicklung von Steuergeräten verwendet wird. Der Grundgedanke kann jedoch einfach übertragen werden.

Folgender Ablauf wird dabei verwendet. Der Konstrukteur meldet die Geometrie seines Bauteils zur Packagingfreigabe an. Die Meldungen werden alle an das Digital Mockup Team (DMU) geleitet. Dort werden die Meldungen koordiniert und an den zuständigen Packagingprüfer weitergeleitet. Der Packagingprüfer überprüft das Bauteil in seinem geometrischen Umfeld am geplanten Verwendungsort im Fahrzeug. Kollisionen und Abstandsverletzungen mit den räumlich umliegenden Nachbarbauteilen werden an die Fachabteilung und somit an den Konstrukteur gemeldet. Solange diese auftreten, wird das Bauteil nicht freigegeben.

Dieses Vorgehen wurde detailliert analysiert und umgesetzt. Ein erster Versuch mit einem prozessorientiertem Informationssystem wurde von den Anwendern jedoch abgelehnt. In einem zweiten Versuch wurde der Prozess durch eine konventionelle Implementierung realisiert. Die Prozesslogik wurde im Anwendungsprogramm direkt implementiert (vgl. Abbildung B.13 auf Seite 92). Andere von den Prozessteilnehmern verwendete Systeme (z. B. zur Speicherung der geometrischen Bauteilbeschreibungen) waren nicht integriert. Bauteile mussten daher von Hand in das System eingetragen werden. Inkonsistenzen zwischen den einzelnen Systemen waren die Folge. Trotz eines relativ einfachen Prozesses waren zahlreiche Änderungen nötig. Diese führten aufgrund der konventionellen Implementierung zu Termin und Qualitätsproblemen beim Hersteller des Systems.

Daraufhin wurde eine Lösung mit expliziter Modellierung der Prozesslogik gewählt. Dieses prozessbasierte Informationssystem wurde mit den bereits vorhandenen Systemen integriert, es tritt also kein Datenbruch mehr auf. Der Prozess wurde auch hier verändert. Die daraus entstehenden Prozessvarianten sind in den Abbildungen B.13,

B.14 und B.15 dargestellt. In diesem Fall sind die Gründe für die Änderungen bekannt. Sie liegen vor allem in der Übersicht der einzelnen Bauteilmeldungen. Die zweite Prozessvariante entsteht, um einen besseren Überblick über den Stand der Meldungen zu erhalten. Eine Verbesserung der Prozessüberwachung und -steuerung (Controlling) ist das Ziel. Soll ein Bauteil geprüft werden, ist es im Status **gemeldet**, bei der Bearbeitung **in Arbeit** und am Ende ist es **bearbeitet**. Diese drei bisher verwendeten Status zur Verbesserung der Übersicht werden auf sieben erweitert. Der Status **bearbeitet** wird in **in Ordnung** und **nicht in Ordnung** aufgeteilt. Die neuen Status **warten**, **abgelehnt** und **in Abstimmung** werden verwendet. Der Status **warten** bedeutet, dass noch Bauteile aus der Umgebung bewertet werden müssen, bevor eine Freigabe erfolgen kann, das aktuelle Bauteil aber abgenommen wurde. Diese Aufteilung wurde von den Anwendern abgelehnt. Sie bewerteten es als zu kompliziert. Status wurden unterschiedlich interpretiert und trugen zur Verwirrung bei. Daher wurde der Prozess wie in Abbildung B.15 auf Seite 92 vereinfacht. Die Abstimmung wurde gestrichen. Die Anzahl der Status wurde auf fünf reduziert. Bearbeitet der Packaging Prüfer gerade das Bauteil, ist aber noch zu keinem Ergebnis gelangt, wird der Status **warten** gesetzt. Seine Bedeutung wurde verändert, war aber immer noch nicht klar: wird auf Nachbarbauteile gewartet, oder auf eine Abstimmung mit dem Konstrukteur? Für den Konstrukteur ist auch nicht ersichtlich, wie weit sein Bauteil gerade ist (**warten** wird nur dem DMU-Team angezeigt). Er erhält nur eine negative oder positive Meldung am Ende der Prüfung.

Diese unterschiedlichen Ansprüche der Prozessbeteiligten an den Prozess führen zu einem Problem in Bezug auf die Prozessgestaltung. Jede Interessensgruppe hat unterschiedliche Ansprüche an die Darstellung des Prozesses. Bei der Überwachung und Auswertung soll dies möglichst detailliert sein. Der Konstrukteur will nur wissen, ob sein Bauteil abgenommen wird oder nicht und wie lange er noch auf die Entscheidung warten muss. Bei Kenntnissen des Prozesses interessieren ihn daher Zwischenstände. Das obere Management hingegen will nur wissen, wie viel Arbeit noch zu erledigen ist. Ein grober Überblick der abgeschlossenen und noch zu bearbeitenden Bauteilen ist ausreichend. Ein Projektleiter ist aber schon an detaillierten Ergebnissen interessiert. Dies stellt einen Zielkonflikt dar.

Die folgende Änderung wurde danach vorgenommen. Die manuelle Prüfung von Kollisionen anhand des digitalen Mockups ist sehr zeitaufwendig und arbeitsintensiv. Es wurde ein System entwickelt, um Kollisionen und Abstandsverletzungen automatisch zu erkennen. Dies hat mehrere Auswirkungen auf den Prozess. Die gewünschte Kontrolle der Bauteile wird nun automatisch durchgeführt und nur noch manuell nachkontrolliert. Diese Änderungen des Prozesses hatten aber noch weitere Auswirkungen. Bis zu diesem Zeitpunkt waren die Serienfreigabe und die Packagingfreigabe voneinander getrennt. Die Packagingfreigabe wurde in die Serienfreigabe integriert. Davon losgelöst wurde die Bewertung des Packages. Kollisionsberechnungen und die (automatische) Verwaltung der Ergebnisse wurden in einen separaten Prozess ausgegliedert. Abstandsverletzungen werden in das Problem und Änderungsmanagement übertragen. Dieses Beispiel zeigt, dass hinreichend starke Änderungen des Kontexts, in diesem Fall eine neu verfügbare Technologie, zu Änderungen eines reifen und lange

verwendeten Prozesses führen können.

3.5 Zusammenfassung

Dieses Kapitel zeigt, dass Änderungen die Regel und sich nicht verändernde Prozesse der Sonderfall sind. Nicht alle der tatsächlich aufgetretenen Änderungen können mit dieser nachträglichen Analyse wiedergegeben werden. Sie sind nicht dokumentiert und den Prozessexperten nicht bekannt. Auch fehlte den Prozessteilnehmern die Möglichkeit, Änderungen des Prozesses auf einfache Art und Weise durchzuführen. Die prozessbasierten Informationssysteme ermöglichten dies nicht. Änderungen wären notwendig gewesen, konnten aber nicht durchgeführt werden. Die beobachteten Änderungen wurden auf drei Ebenen durchgeführt: vor Projektbeginn (um Prozesse an das Projekt anzupassen) und während des Projekts. Die Änderungen während des Projekts konnten weiter in Änderungen unterteilt werden, die zyklisch immer wieder auftreten und in nicht vorhersehbare Änderungen. Die zyklisch auftretenden Prozessänderungen frustrieren die Prozessexperten. Sie haben den Eindruck, die gleichen Änderungen immer und immer wieder durchführen zu müssen. Die verwendeten Informationssysteme konnten diese Änderungen nicht adäquat unterstützen. Im nächsten Kapitel 4 auf der nächsten Seite wird versucht, die Änderungen weiter zu unterteilen. Kapitel 6 auf Seite 57 untersucht, wie gut ein adaptives Prozessmanagement System diese Änderungen unterstützen kann.

Kapitel 4

Kategorisierung der Änderungen

In diesem Kapitel wird eine Kategorisierung der gefundenen Änderungen und deren Ursachen vorgeschlagen. Abschnitt 4.1 entwickelt die Kategorisierung. In der Literatur gefundene Kategorien werden angepasst und um die Erkenntnisse dieser Arbeit erweitert. In Abschnitt 4.2 werden die gefundenen Änderungen aus Kapitel 3 in die zuvor entwickelten Kategorien eingeordnet. In Tabelle 4.2 auf Seite 47 werden die Kategorien mit typischen Beispielen des Kontexts dargestellt. In 4.3 werden diese Kategorien abschließend bewertet.

4.1 Entwicklung der Kategorisierung

Gründe für Änderungen werden in der Literatur aus drei Perspektiven beschrieben: Technische Aspekte aus Sicht eines Prozess-Management Systems (dargestellt in 4.1.1), aus Organisationssicht (dargestellt in 4.1.2 auf der nächsten Seite) und aus der Sicht eines Referenzprozessmodells (dargestellt in 4.1.3 auf Seite 44).

4.1.1 Änderungen aus Perspektive eines Prozess-Management-Systems

In [AJ00] werden Änderungen und deren Auswirkungen aus Sicht eines Prozess-Management-Systems beschrieben. Ursachen von Änderungen und die Änderungen selbst müssen verstanden werden, damit Prozess-Management-Systeme auf sich ändernde Bedingungen angemessen reagieren können. Die Ursachen von Änderungen werden in zwei Bereiche unterteilt: Änderungen, deren Ursachen innerhalb und Änderungen, deren Ursachen außerhalb des betrachteten (prozessbasierten Informations-) Systems liegen. Änderungen werden hauptsächlich durch Entwicklungen in der Umgebung (Kontext) des Informationssystems ausgelöst. Auslöser für Änderungen *außerhalb* des Systems sind drei Bereiche:

1. *Geschäftsumfeld*: Dies geschieht im Rahmen von Geschäftsprozessverbesserungen und -anpassungen (Business Process Reengineering). Auch sich ändernde Marktbedingungen durch neue Konkurrenten oder Produkte sowie individuelle Kundenanforderungen werden diesem Bereich zugeordnet.
2. *Rechtliches Umfeld*: Führt eine neue oder geänderte Gesetzgebung zu Änderungen, gehört sie in diese Kategorie. Gesetzlich vorgeschrieben Dieselfußfilter oder

die Verpflichtung der Hersteller zur Rücknahme ihrer Fahrzeuge sind Beispiele hierfür.

3. *Technologisches Umfeld*: Werden neue Technologien verfügbar, können diese z. B. zur Reduzierung des Wartungsaufwandes und zur Erhöhung der Verfügbarkeit führen. Auch die technische Infrastruktur kann sich ändern, wenn ein Dienstleister seine technischen Dienste einstellt.

Auslöser für Änderungen *innerhalb* des Systems entstehen durch Probleme des Systems selbst:

1. *Logische Design-Fehler*: Fehler, die während der Entwicklung des Systems gemacht wurden und zu Verklemmungen oder fehlender Datenversorgung von Aktivitäten führt.
2. *Technische Probleme*: Die Leistungsfähigkeit oder Zuverlässigkeit des Systems wird durch ausfallende Komponenten beeinträchtigt.

Diese Kategorien sind stark auf das PMS selbst ausgerichtet. Die genannten Beispiele zu technologischen Änderungen betreffen das PMS und seine technischen Komponenten. Ob der technologische Wandel inhaltliche Auswirkungen auf den Prozess hat, wird hier nicht (direkt) angesprochen. In Kapitel 3 wäre dies im Freigabemanagement die Einführung einer Prozessverbesserung (Automatisierung einer Aktivität), die außer dieser Automatisierung noch weitere Änderungen zur Folge hatte (die Integration bestimmter Aktivitäten in einen anderen Prozess).

Änderungen, die aus der Änderung der Aufbauorganisation resultieren, werden ebenfalls nicht erfasst. Es wird davon ausgegangen, dass Änderungen nur Auswirkungen auf die Organisationsperspektive haben und nicht durch diese selbst ausgelöst werden können. Wird eine Fremdfirma integriert (Änderung der Organisationsperspektive), ändert sich nicht nur die Aufbauorganisation. Verantwortlichkeiten ändern sich und spezielle Aktivitäten zur Koordinierung mit der Fremdfirma müssen berücksichtigt werden. (Aktivität **Analyse intern/extern**).

4.1.2 Änderungen aus Organisationssicht

In [WM00] werden Änderungsursachen genannt, die Organisationen betreffen. Organisationen werden durch fünf Änderungsquellen in ihrer Umgebung beeinflusst. Diese werden durch zwei¹ Quellen innerhalb der Organisation selbst ergänzt. Änderungsursachen der *Organisationsumgebung* sind:

1. *Technologisch*: Technologische Änderungsursachen beschäftigen sich mit dem Wissen über die Art und Weise, mit der Aufgaben erledigt werden können.

¹Es werden drei Änderungsursachen innerhalb der Organisation selbst genannt. Die Änderungsursache Politik wird hier nicht weiter beschrieben. Die Akzeptanz bzw. Ablehnung eines prozessbasierten Informationssystems wegen der dadurch veränderten Machtstruktur wird als Beispiel genannt. Diese Änderungsursache ist aber schon zu weit von den beobachtbaren Änderungen dieser Arbeit entfernt, um weiter berücksichtigt werden zu können

Bisher verwendete Ressourcen werden durch die neu verfügbaren ersetzt. Die Aufgabe, die mit der alten Ressource (oder auf die alte Art und Weise) durchgeführt worden ist, verschwindet jedoch meistens nicht. Sie passt sich den neuen Gegebenheiten an.

2. *Ökonomisch*: Zulieferer, Kosten, Kunden und andere allgemeine Marktbedingungen werden hier erfasst. Diese beeinflussen ein Aufgabengebiet hierarchisch von oben nach unten: Verschiebungen im ökonomischen Umfeld führen dazu, dass sich die Ziele und Aufgaben einer Organisation in Bezug auf ein Aufgabengebiet ändern. Dies führt dazu, dass die Organisation andere Lösungswege als die bisherigen bevorzugt. Aufgaben werden durch andere ersetzt oder nicht mehr so häufig benötigt.
3. *Gesellschaftlich*: Gesellschaftliche Änderungsursachen beschäftigen sich mit den Einstellungen, Überzeugungen, Gebräuchen, Wünschen und Erwartungen von Menschen. Das soziale Umfeld beeinflusst die Organisation und die von ihr durchgeführten Aufgaben: soziale (gesellschaftliche) Änderungen können technologische Änderungen hervorbringen. Als Beispiel wird die Einstellung gegenüber Krediten genannt. Deren Akzeptanz fördert die Verwendung von Kreditkarten und Organisationen müssen die Bezahlung per Kreditkarte in ihren Prozessen berücksichtigen.
4. *Politisch-Rechtlich*: Politisch rechtliche Änderungsursachen umfassen Gesetze, Normen sowie Behörden und deren Einfluss auf die Organisation. Diese können sowohl einschränken (erhöhte Grenzwerte) als auch neue Möglichkeiten eröffnen (Freigabe eines Frequenzbandes für Abstandsradar).
5. *Ethisch-Moralisch*: Ob das Vorgehen und Verhalten von Organisationen akzeptiert wird, hängt von ethisch moralischen Erwägungen ab. Diese bilden einen vernachlässigten Teil des Kontexts von Unternehmen. Sie haben nicht notwendigerweise unmittelbare Auswirkungen auf Informationssysteme. Das steigende Umweltbewusstsein führt zur Bevorzugung ökonomisch verträglicher Lösungen und ist auch der Grund für gesetzlich festgelegte Abgasgrenzwerte. Verhaltensweisen von Unternehmen, die dem nicht entsprechen, werden nicht akzeptiert und müssen geändert werden.

Änderungsursachen *innerhalb* der Organisation sind:

1. *Organisationsstruktur*: Die Organisationsstruktur wird als Verteilung von Verantwortlichkeiten zwischen Abteilungen und deren Stellen beschrieben. Damit wird die Aufbauorganisation der Organisation beschrieben.
2. *Kultur*: Die (Unternehmens-) Kultur ist eine Kombination der Ideologien, Annahmen und Werte innerhalb der Organisation. Die Ziele der Organisation, deren Umsetzung durch Werte und die Interaktion zwischen den einzelnen Mitarbeitern und ihrer Umgebung fallen hierunter.

Die Organisationsstruktur wird als weitere Änderungsursache genannt. Die Definition technologischer Änderungsursachen umfasst die beobachtbaren Änderungen dieser Arbeit. Insgesamt werden weniger technische Ursachen beschrieben und semantische Änderungen, die oben als Geschäftsprozessänderungen umrissen worden sind, weiter detailliert.

4.1.3 Änderungen aus Sicht eines Referenzprozessmodells

Im Rahmen einer Studie über Prozessvariabilität der Fraunhofer IESE² [MO04] werden Gründe angeführt, die zu Änderungen eines Referenzprozessmodells führen. Es wird zwischen externen/technologischen und internen/nicht technologischen Gründen unterschieden. Auf Änderungen bei der Verwendung des Prozesses wird nicht näher eingegangen. Externe/technologische Gründe sind:

1. Neue oder geänderte Softwareentwicklungstechniken, z. B. Werkzeuge und Methoden zur Unterstützung von Tests
2. neue Techniken der Prozessmodellierung
3. Neue oder geänderte Standards und Richtlinien zur Softwareentwicklung
4. Erfahrung, die zu neuen oder verbesserten Arbeitsabläufen führen, z. B. Empfehlungen der IEEE³

Interne/nicht technologische Gründe sind:

1. Neue oder geänderte Geschäftsregeln, als Folge geänderter Geschäftsziele
2. Erfahrungen aus abgeschlossenen Projekten, die zu neuen oder verbesserten Arbeitsabläufen führen
3. Neue Ansätze, um die Kooperation innerhalb von Teams zu verbessern

Die Ursachen und die daraus resultierenden Änderungen sind in Bezug auf das Referenzprozessmodell zu sehen. Individuelle Abweichungen vom Arbeitsablauf (Änderungen an einer Prozessinstanz) werden nicht berücksichtigt. Die aufgezählten Änderungsgründe benötigen keine unmittelbare Reaktion. Die neue Technik der Prozessmodellierung muss nicht einmal überhaupt eine Änderung des Prozesses erzwingen. Werden Änderungen eingeführt, können diese geplant und gezielt in das Prozessmodell integriert werden. Diese Änderungsgründe erweitern die beiden obigen Studien um diesen neuen Aspekt.

²IESE: Institut Experimentelles Software Engineering

³IEEE: Institute of Electrical and Electronics Engineers

diese Arbeit	[WM00]	[AJ00]	[MO04]
Projekt			
Aufbauorganisation	Organisationsstruktur		
Technologie	technologisch	technologisches Umfeld	
Kultur	gesellschaftlich (Unternehmens-) Kultur ethisch-moralisch		
Strategie	ökonomisch	Geschäftsumfeld	Änderungen Referenzprozessmodell
Politisch-Rechtlich	politisch-rechtlich	rechtliches Umfeld	

Tabelle 4.1: Vergleich der Kategorisierung dieser Arbeit mit der Kategorisierung aus [WM00], [AJ00] und [MO04]

4.1.4 Anpassung und Erweiterung

Obige Kategorisierung wird nun an die Bedürfnisse dieser Arbeit angepasst. Die beobachteten Änderungen und deren Ursachen sollen adäquat unterteilt werden können. Die daraus entstehende Kategorisierung wird in Tabelle 4.1 mit den oben dargestellten verglichen.

Zunächst wird die Perspektive festgelegt. Die Perspektive des Prozess-Management-Systems ist zu technisch. Das Informationssystem beschreibt lediglich die gewollte Vorgehensweise und ist nur eines der zur Verfügung stehenden Werkzeuge/ Ressourcen (vgl. 2.3 auf Seite 18). Daher werden die Änderungen aus der Sicht der Organisation betrachtet. Vor allem das Projekt als spezielle Organisation wird betrachtet. Dieses Vorgehen deckt sich mit den Erkenntnissen aus [WM00], dass bei der Entwicklung von Informationssystemen andere als nur technische Änderungsgründe berücksichtigt werden müssen. Die Betrachtung eines Referenzprozessmodells kann jedoch ohne Probleme in die Betrachtung der Änderungen aus Sicht der Organisation integriert werden.

Die Änderungskategorien werden in zwei Dimensionen beschrieben: Änderungsursache und Planbarkeit.

Änderungsursachen

1. *Projekt* ist eine neu eingeführte Kategorie. In ihr werden alle Änderungen zusammengefasst, die auf spezielle Projekteigenschaften (Charakteristiken)

Kapitel 4 Kategorisierung der Änderungen

zurückzuführen sind (und in keine der anderen Kategorien fallen). Änderungen des Prozesses müssen vorgenommen werden, weil sich zwei Projekte in speziellen Eigenschaften unterscheiden. Ändert sich eine dieser Charakteristiken, treten ebenfalls Änderungen auf.

2. *Aufbauorganisation*: wie oben beschrieben;
3. *Technologie*: wie oben beschrieben;
4. *Kultur* ist eine neu definierte Kategorie. Sie umfasst die oben beschriebenen gesellschaftlichen, ethisch-moralischen und kulturellen Änderungsgründe. Sie können im Rahmen dieser Arbeit nicht unterschieden werden. Obige Beschreibungen gehen zwar darauf ein, dass diese Änderungen wichtig sind. Genauere Angaben, wie sich diese ausdrücken, werden jedoch nicht gegeben. Gespräche mit Prozessexperten ergaben, dass diese Unterschiede bei der Prozessverwendung zwischen den Kulturen erkennen. Geert Hofstede liefert eine passende Theorie. Diese wird in Abschnitt 4.2.4 auf Seite 50 näher beschreiben.
5. *Strategie*: Hierunter werden alle Änderungen verstanden, die über einen längeren Zeitraum wirken. Die unter 4.1.3 auf Seite 44 definierten Gründe werden hier zusammengefasst. Erkenntnisse über ein verbessertes Vorgehen bei Softwaretests beispielsweise werden vorsichtig und schrittweise in vorhandene Prozesse integriert werden. Unmittelbarer Handlungsbedarf besteht hier nicht.
6. *Politisch-Rechtlich*: wie oben beschrieben;

Andere Kategorien konnten nicht beobachtet werden, so dass sie hier nicht weiter betrachtet werden. Im Rahmen der Erstellung der Fallbeispiele stellte sich die Frage, welche der Änderungen immer wieder auftreten und welche singulär sind, und zu welchem Zeitpunkt sie auftreten. Zu diesem Zweck wird eine weitere Kategorie eingeführt, die sich mit der Planbarkeit der Änderungen beschäftigt.

Planbarkeit Erfahrungen der Prozess- und Projektextperten zeigen, dass gleiche oder ähnliche Änderungen immer wieder vorgenommen werden müssen. Dieser Aspekt soll im Rahmen dieser Kategorie untersucht werden. Da die Kategorisierung aus Projektsicht erfolgt, wird die Planbarkeit auch auf dieses bezogen. Damit entstehen *planbare* und *nicht planbare* Änderungen. Bei den planbaren Änderungen kann noch zwischen Änderungen vor und zur *Projektlaufzeit* unterschieden werden. Änderungen vor Projektbeginn werden mit *a priori* gekennzeichnet. Die Kategorie planbar-zur Laufzeit enthält unter anderem auch die Änderungen, die in jedem Projekt zyklisch immer wieder auftreten.

Die sich daraus ergebende Kategorisierung der Fallbeispiele ist in Tabelle 4.2 auf der nächsten Seite dargestellt.

Ursache	Planbar	Nicht planbar	
	a priori	zur Laufzeit	
Projekt	projektspezifische Aktivitäten: bei externen Beteiligten Aktivität Analyse (intern/extern) verwenden	bis Beginn Serienproduktion immer mehr Prüfungen/Freigaben, danach weniger	Qualitätsprobleme: zusätzliche Prüfungen; Überspringen von Aktivitäten
Aufbauorganisation	Neuer Partner: Steuergerät auch für diesen entwickeln	Personenanzahl steigt Bearbeiterzuordnung ändert sich	Kapazitätsengpass: neuer Zulieferer muss integriert werden
Technologie		Wechsel der Entwicklungsplattform: von Simulator über Prüfstand zu Fahrzeug	Automatisierung von Aktivitäten: automatisierte Tests
Kultur	Internationalisierung der Mensch-Maschine Schnittstelle		Hofsteedes Cultural Dimensions: Organisation hierarchisch oder flach
Strategie		geplante Einführung von Prozessverbesserungen	Sensor durch Berechnung ersetzen (Kosten): Hardware- durch Software-Ersetzen
Politisch-Rechtlich			strengere Abgaskennwerte: auf alten Werten basierende Aktivitäten validieren

Tabelle 4.2: Kategorisierung von Änderungen mit Beispielen

4.2 Einordnung der Beispiele

Hier werden nun die in Kapitel 3 dargestellten Beispiele eingeordnet. In Tabelle 4.2 wird dies in einer Übersicht dargestellt.

4.2.1 Projektbezogene Änderungen

Zu einem gegebenen Prozess existieren mehrere projektspezifische Varianten. Die Änderungen können auf Eigenschaften des speziellen Projekts zurückgeführt werden. Projektbezogene Änderungen waren in den Fallbeispielen eine der Hauptänderungsgründe. Prozessexperten und Systementwickler mussten vorhandene Prozesse und prozessbasierte Informationssysteme mit viel Aufwand anpassen. Diese Anpassungen nahmen viel Zeit in Anspruch und verzögerten den Einsatz des Systems. Auch nach den initialen Anpassungen an die Projektbedürfnisse (planbar-a priori) traten weiterhin Änderungswünsche auf. Prozesse müssen immer wieder an sich verändernde Bedürfnisse des Projekts angepasst werden. Diese Änderungen treten innerhalb des Projekts und über Projektgrenzen immer wieder in ähnlicher Form auf. Diese sich wiederholenden Prozessänderungen sind planbar - zur Laufzeit.

planbar-a priori Hierunter fallen Änderungen des Prozesses, die sich aus den Eigenschaften des Projekts ergeben. Die Aktivität **Analyse intern/extern**, in der untersucht wird, ob der externe Zulieferer oder die interne Entwicklung für ein Problem zuständig ist, kann nur in Projekten mit externen Zulieferern überhaupt sinnvoll verwendet werden. Die unterschiedliche Verwendung von Aktivitäten in unterschiedlichen Projekten zeigt Tabelle 3.3 auf Seite 34.

planbar-zur Laufzeit Zu Beginn eines Projekts finden wenig Prüfungen und Freigaben statt. Die Entwickler sind noch freier in ihrem Vorgehen und ihren Entscheidungen. Je näher der Beginn der Serienproduktion rückt, desto mehr Prüfungen werden eingeführt, um die Qualität des Projekts zu gewährleisten. Im C-Muster wird ein Fahrversuchstest durchgeführt, zu Beginn des Projekts nicht. Unterschiedliche Prüfzyklen werden in Tabelle 3.5 auf Seite 36 dargestellt.

nicht planbar Treten in einem Projekt Qualitätsprobleme auf, ist eine mögliche Reaktion darauf, zusätzliche Prüfungen durchzuführen. Diese Prüfungen müssen dann auf alle laufenden Prozessinstanzen propagiert werden (so weit möglich). Auch bei einzelnen Instanzen können Änderungen an dem Prozess nötig sein. Diese sind vom konkreten Kontext und somit vom Projekt abhängig.

4.2.2 Änderungen der Aufbauorganisation

Die Aufbauorganisation eines Projekts bzw. der sie umgebenden Organisation ist auch ständigen Änderungen unterworfen.

planbar-a priori Ein Referenzprozess ist für eine bestimmte Organisationsstruktur ausgelegt. Ist von Anfang an bekannt, dass an diesem speziellen Projekt eine weitere, normalerweise nicht beteiligte Organisationseinheit beteiligt wird, müssen ebenfalls Änderungen am Organisationsmodell vorgenommen werden. Jedes Projekt bevorzugt auch ein anderes Organisationsmodell. Projektspezifische Organisationsmodelle werden in Tabelle 3.2 auf Seite 32 gezeigt. Darauf aufbauend ordnen Projekte Aktivitäten anderen Bearbeitern zu. Dies ist in Tabelle 3.4 auf Seite 35 zu sehen.

planbar-zur Laufzeit Üblicherweise wird ein Projekt mit möglichst wenig Mitarbeitern begonnen. Diese haben dann mehrere Rollen inne. Im Verlauf des Projekts steigt die Anzahl Personen und Rollen werden neu verteilt. Ebenso sind bestimmte Teile der Organisation nur zu bestimmten Zeiten des Projekts involviert. Die Abteilung, die Fahrzeugversuche durchführt, wird erst gegen Ende der Entwicklung eines Steuergeräts damit beschäftigt sein.

Einem Mitarbeiter sind mehrere Rollen zugeordnet. Er meldet sich am System mit einer der Rollen an und kann dann die ihm zugewiesenen Aktivitäten bearbeiten. Will er die Aktivitäten der anderen Rolle ausführen, muss er sich mit dieser Rolle am System anmelden. Hat z. B. bei ÄM(G) ein Mitarbeiter die Rollen Experte und Systemexperte inne, folgen die zu bearbeitenden Aktivitäten direkt aufeinander. Entscheidet er in seiner Rolle als Systemexperte, dass keiner der anderen Experten sondern er für den Antrag verantwortlich ist, (**Antrag zuordnen**) muss er sich als Systemexperte abmelden und als Experte anmelden, um den an sich selber zugewiesenen Antrag zu bearbeiten. Die in den Fallbeispielen untersuchten Systeme erforderten diese. Eine mögliche Lösung ist, dass ein Mitarbeiter, wenn er sich am System anmeldet, Arbeitslisten für jede seiner Rollen angezeigt bekommt.

nicht planbar In einem Projekt tritt ein Kapazitätsengpass auf. Daher wird eine Fremdfirma beauftragt, Teile der Entwicklung zu übernehmen. Neben den Anpassungen des Organisationsmodells müssen weitere Änderungen vorgenommen werden. Fremdfirmen müssen überwacht und kontrolliert werden. Entsprechende Aktivitäten müssen eingeplant werden. Auch sind sensible Daten vor dem Zugriff der Fremdfirma zu schützen.

4.2.3 Technologische Änderungen

planbar-a priori kein Beispiel gefunden

planbar-zur Laufzeit Im Rahmen des Entwicklungsprozesses für eingebettete Steuergeräte findet je nach Phase eine andere technologische Grundlage als Entwicklungsplattform Verwendung. Das A-Muster wird auf einem Simulator entwickelt, die anderen Muster auf der Zielhardware. In jedem Projekt ist diese Entwicklung immer wieder die gleiche und tritt zur Projektlaufzeit auf. Auch die Auswirkungen dieses Wechsels sind bekannt. Tests auf einem Simulator enthalten andere Aktivitäten als Tests eines realen Steuergeräts in einem Versuchsaufbau. Diese Änderungen sind stark

mit dem zugrunde liegenden Entwicklungsprozess und dessen Phasen verbunden. Dies ist in Abschnitt 3.2 auf Seite 22 beschrieben.

nicht planbar In diese Kategorie fallen die Automatisierung von Aufgaben wie im Falle der Sonderentwicklung (vgl. 3.2.3 auf Seite 24) oder des Freigabemanagements (vgl. 3.4 auf Seite 38). Manuelle Aktivitäten werden (teilweise) automatisiert. Wird eine solche Automatisierung verfügbar, oder ist technologisch ausgereift, wird sie in den Projekten eingesetzt. Es ist jedoch nicht vorhersehbar, wann und ob dies der Fall ist. Daher sind diese nicht planbar, müssen jedoch von den Projekten berücksichtigt werden. Die Auswirkungen dieser Änderungen sind ebenfalls nicht vorhersehbar. Nicht nur die Aktivitäten selber werden ausgetauscht, auch die Arbeitsweise selber ändert sich.

4.2.4 Kulturelle Änderungsgründe

In einer globalisierten Wirtschaft, mit länderübergreifenden weltweiten Konzernen spielen kulturelle Unterschiede eine große Rolle und müssen daher berücksichtigt werden. Geert Hofstede fasste kulturelle Unterschiede in fünf kulturelle Dimensionen [HH05]. Eine Befragung von IBM Mitarbeitern in 70 Ländern führte zu einer Rangordnung der jeweiligen Dimensionen. Dadurch wird es möglich Länder quantitativ miteinander zu vergleichen. Aussagen über Unterschiede zwischen einzelnen Ländern und vor allem die Größe der Unterschiede können damit getroffen werden. Die fünf Dimensionen sind:

- Machtdistanz (Power Distance): Hier wird die soziale Ungleichstellung und die Reaktion darauf beschrieben. Dies ist die, auf der Machtstruktur beruhende, Rangordnung.
- Individualismus/Kollektivismus (Individualism vs. Collectivism): Kulturen können individuelle oder gemeinschaftliche Züge tragen. Bei gemeinschaftlichen Kulturen dominieren familiäre Bindungen und Gruppen. Individualistische Kulturen werden durch ein starkes Ich-Gefühl gerägt.
- Männlichkeit/Weiblichkeit (Masculinity vs. Femininity): Kulturen können eher männlich oder weiblich geprägt sein.
- Vermeidung von Unsicherheiten (Uncertainty Avoidance): Kulturen haben einen anderen Umgang mit Unsicherheiten. Zu viel Freiheiten und Entscheidungsmöglichkeiten führen entweder zu Verunsicherung oder werden vorrausgesetzt und bevorzugt.
- Langfristige/kurzfristige Orientierung (Long- vs. Short-Term Orientation): Der kulturelle Hintergrund bestimmt die Orientierung an kurzfristigen oder langfristigen Zielen und Werten. Personen richten ihr Handeln und Denken danach.

Mögliche Auswirkungen dieser Dimensionen werden unter unplanbar angerissen.

planbar-a priori Bereits vor Projektbeginn steht fest, welche Nationen an dem Projekt beteiligt sein werden. Eine Lokalisierung der Mensch-Maschine Schnittstelle des prozessgestützten Informationssystems ist daher nötig. Die Verwendung der Landessprache ist hierbei noch der einfachere Aspekt. Vieles deutet darauf hin, dass kulturelle Unterschiede auf die Mensch-Maschine Schnittstelle weitreichende Folgen besitzen (vgl. [MA01]). Diese beschränken sich nicht nur auf die Sprache, sondern z. B. auch auf bevorzugte Darstellungs- oder Navigationsformen. Da diese jedoch nicht so einfach zu erfassen sind, wird hier auf eine weitere Darstellung verzichtet.

planbar-zur Laufzeit kein Beispiel gefunden

nicht planbar Nach Hofstede's Dimensionen lassen sich einzelne Kulturen gezielt miteinander vergleichen. Allein auf diesen Unterschieden lassen sich jedoch keine konkreten Änderungen vorhersagen. Betrachtet man einen Prozess als Anleitung zur Arbeit, wird schnell klar, dass die in einer Kultur geltenden Normen diese beeinflussen und somit zu Änderungen des Prozesses führen können. Ein Prozess, der erfolgreich in einer westlichen Kultur wie Deutschland eingesetzt wird, muss nicht zum gleichen Erfolg in einer asiatischen Kultur wie Japan führen. Änderungen sind die Folge.

Eine der Dimensionen und ihre möglichen Auswirkungen auf den Prozess werden kurz erläutert: Die Power Distance befasst sich mit der Machtdistanz zwischen Angestellten und Vorgesetzten. Kulturen mit einem großen Power Distance Index besitzen eine streng hierarchische Rangordnung innerhalb der Organisation in der von Angestellten viel Disziplin erwartet wird. Kulturen mit kleinem Power Distance Index leben eine flache Hierarchie in die Angestellte viel Eigeninitiative einbringen. Daraus wird unmittelbar der Zusammenhang mit der Aufbauorganisation klar. Daneben können aber auch Schlussfolgerungen in Bezug auf die benötigte Flexibilität des Prozesses gemacht werden. Wird von einem Angestellten Eigeninitiative erwartet, muss er auch die Möglichkeit haben, vom vorgesehen Prozessablauf abzuweichen. Eng damit verknüpft ist, in diesem Zusammenhang, die Uncertainty Avoidance (Vermeidung von Unsicherheiten). Kulturen mit einem hohen Wert bevorzugen strengere Vorgaben in Bezug auf was getan werden sollte. Zu viele Freiheiten oder Auswahlmöglichkeiten sind nicht erwünscht. Der Prozess sollte dies widerspiegeln.

4.2.5 Strategische Änderungen

planbar-a priori kein Beispiel gefunden

planbar-zur Laufzeit Prozessverbesserungen werden nicht schlagartig eingeführt. Ihre Einführung wird sorgfältig geplant. Prozessverbesserungen werden zuerst an ausgewählten Benutzern getestet, bevor diese für alle verfügbar gemacht werden. Die Verbesserung an sich wird stufenweise eingeführt. Die Verbesserung des Entwicklungsprozesses für Sonderfunktionen 3.2.3 auf Seite 24 beispielsweise. Diese Verbesserungen werden nicht nur zwischen einzelnen Projekten eingeführt, sondern auch während

der Projekte. Verbesserungen sollen diesen so schnell wie möglich verfügbar gemacht werden.

nicht planbar Im Kontext des Prozesses werden Entscheidungen getroffen die dieses und das PAIS beeinflussen, aber nicht von ihm beeinflusst werden. Im Projekt soll ein Wert durch einen Sensor gemessen werden. Der Prozess und auch andere Entscheidungen basieren auf dieser Annahmen (Anforderung). Aus Kostengründen, oder anderen Gründen, die dem Projekt nicht bekannt sein müssen, werden diese durch eine Berechnung ersetzt. Dies hat nun offensichtliche und versteckte Auswirkungen auf den Prozess. Bisher war ein hardwarebasierter Test des Sensors geplant. Dieser entfällt und wird durch eine Simulation ersetzt. Auch muss dieser erst entwickelt werden. Wurden im Projekt Entscheidungen getroffen, die direkt oder indirekt auf der Annahme eines Sensors beruhen, müssen diese überprüft werden. Treffen diese nicht mehr zu, müssen Aktivitäten erneut durchgeführt werden.

4.2.6 Rechtliche Änderungen

Beispiele hierfür sind Einführung oder Änderungen von Abgasgrenzwerten. Oftmals ist es bekannt, dass neue Regulierungen erlassen werden, deren genaue Auswirkungen können jedoch erst abgeschätzt werden, wenn der genaue Inhalt bekannt ist.

planbar-a priori kein Beispiel gefunden

planbar-zur Laufzeit kein Beispiel gefunden

nicht planbar Die Auswirkungen auf ein Projekt müssen aber dennoch berücksichtigt werden. Betrifft die Regelung das zu entwickelnde Produkt, so müssen diese Regelungen noch zur Projektlaufzeit berücksichtigt werden. Damit wird das Projekt beeinflusst. Welche Auswirkungen hat dies aber auf Prozesse? Instanzen, die implizit oder explizit von den Regelungen betroffen sind, müssen neu bewertet werden.

4.3 Zusammenfassung

Änderungen, die im Umfeld von Entwicklungsprojekten auftreten, wurden in diesem Kapitel kategorisiert. Sie können in zwei Dimensionen nach Ursache und Planbarkeit unterteilt werden. Die Ursache beschreibt aus welcher Richtung Änderungen an Prozesse zu erwarten sind und immer wieder auftreten. Diese können dann planbar oder nicht planbar sein. Für nicht planbare Änderungen hilft die Kategorisierung zu erkennen, welche Mechanismen nötig sind, um darauf flexibel reagieren zu können. Treten Änderungen immer wieder auf, sind also planbar, müssen diese einfach immer wieder durchführbar sein. Darunter fallen Änderungen, die vor Beginn eines Projekts immer wieder an Prozessen auftreten, als auch Änderungen, die sich zyklisch zur Projektlaufzeit wiederholen. Die Kategorisierung dieser Arbeit kann alle aufgetretenen

Änderungen erfassen. Soll ein PAIS eingesetzt werden, um die beobachteten Prozesse zu unterstützen, muss es alle auftretenden Änderungen der Kategorien abdecken.

Kapitel 5

Zusammenfassung der Anforderungen

Aus den Fallbeispielen und deren anschließenden Kategorisierung lassen sich Anforderungen an ein prozessbasiertes Informationssystem ableiten. Diese sind stark an die Fallbeispiele angelehnt und verbinden die Fallbeispiele mit der punktuellen Umsetzbarkeitsanalyse in ADEPT. Betrachtet wird dabei die benötigte Flexibilität. Nicht weiter betrachtet werden hingegen Usability-, Verfügbarkeits- und Wartbarkeitsaspekte.

Aus den Fallbeispielen und der Kategorisierung geht hervor, dass Änderungen an Prozessen an der Tagesordnung stehen. Ein projektspezifischer Prozess sollte so einfach wie möglich in einem PAIS umgesetzt werden können. Dieses PAIS sollte aber seinen Benutzern weiterhin die benötigte Flexibilität bieten.

5.1 Vor Projektbeginn

Die Fallbeispiele haben gezeigt, dass selbst bei einem wohlverstandenen einfachen Prozess wie dem Problem- und Änderungsmanagement, ein PAIS bzw. ein Prozess die Bedürfnisse mehrere Projekte erfüllen kann. Ein angepasster Prozess und darauf basierendes PAIS müssen möglich sein. Diese Anpassungen müssen zeit- und kosteneffektiv machbar sein. Ein speziell an die Bedürfnisse des Projekts angepasstes PAIS darf nicht erst nach Projektbeginn zur Verfügung stehen. Es konnte beobachtet werden, dass Prozesse zu Beginn einfach sein können und erst mit steigender Erfahrung der Prozessbeteiligten komplizierter werden müssen.

Auch die zugrunde liegende Organisationsstruktur ist nicht identisch. Die initiale Organisationsstruktur und die darauf aufbauenden Mitarbeiterregeln sollten einfach geändert werden können und immer einen konsistenten Zustand ergeben. Darauf aufbauend müssen auch die Mitarbeiterregeln einfach angepasst werden können. In Tabelle 3.4 auf Seite 35 werden die Zuordnung von Aktivitäten je Projekt dargestellt. Darin kann man einfach die nötigen Anpassungen ablesen.

5.2 Zur Projektlaufzeit

Betrachtet man die Änderungen, die zur Projektlaufzeit auftreten, kann man Änderungen des Prozessschemas, einzelner Prozessinstanzen, des Organisationsmodells und der Mitarbeiterregeln feststellen.

In 3.3.3 auf Seite 29 konnte ein Status nur gelöscht werden, falls sich keine Instanz mehr in dem entsprechenden Status befindet. Insbesondere gilt dies auch für bereits

beendete Instanzen. Großer manueller, bei vielen Instanzen nicht mehr zu leistender, Aufwand war nötig, um das Schema trotzdem ändern zu können. Dies widerspricht aber dem gewünschten Verhalten aus Benutzersicht. Das Schema muss geändert werden können, ohne nach dem alten Schema laufende Instanzen ändern zu müssen. Dies darf nicht zu einem inkonsistenten Systemzustand führen. Neue Instanzen sollten nach dem neuen Schema gestartet werden können. Für bereits gestartete Instanzen gibt es Anwendungsfälle, in denen es vorteilhaft ist, diese nach dem alten Schema zu beenden. Es gibt aber auch Fälle, in denen versucht werden sollte, bereits gestartete Instanzen, so weit wie möglich, nach dem neuen Schema laufen zu lassen.

Die Erfahrung der Prozessexperten hat gezeigt, dass Änderungen an den Prozessschemas und -instanzen sich zwischen den Projekten ähneln und wiederholen. Ein Prozessschema durchläuft von Projektanfang bis -ende immer wieder die gleichen Änderungen und Änderungen zwischen den Projekten am Schema und an Prozessinstanzen gleichen sich ebenfalls. Dies wurde von keinem der untersuchten Systeme unterstützt. Einmal gemachte Änderungen, sowohl am Schema als auch an Instanzen, sollten wieder verwendet werden können. Dies zeigt sich besonders am Wechsel zwischen den Phasen des Entwicklungsprozesses. Ist die A-Muster Entwicklung beendet, müssen Änderungen am Prozess vorgenommen werden, um den Anforderungen der B-Muster Entwicklung gerecht zu werden (vgl. Abschnitt 3.2 auf Seite 22). Diese Änderungen sind immer die gleichen und werden in jedem neuen Projekt erneut benötigt.

Eine Abweichung vom vorhergesehenen Ablauf bei einzelnen Prozessinstanzen muss möglich sein. Die in den Fallbeispielen betrachteten Systeme machten es nötig, die Systemmechanismen zu umgehen. Inkonsistenzen waren die Folge. Ein PAIS zur Unterstützung von Entwicklungsprojekten sollte Abweichungen also ermöglichen und immer einen konsistenten Zustand des Prozesses zur Folge haben. Die Änderungen, aus der Sicht eines Mitarbeiters, sind z. B. das Einfügen oder Löschen eines Prüfzyklusses. Auch das Überspringen oder Vorziehen einzelner Aktivitäten fällt darunter.

Da sich sowohl die Anzahl der Projektbeteiligten, als auch die beteiligten internen und externen Organisationseinheiten ändern, sollten diese im Informationssystem einfach nachgeführt werden können. Neben den Änderungen des Organisationsmodells wandeln sich auch die Mitarbeiterregeln.

Kapitel 6

Analyse der Umsetzbarkeit in ADEPT

In diesem Kapitel wird untersucht, wie gut sich die bisher dargestellten Prozessbeispiele in ADEPT umsetzen lassen. Die Bewertung erfolgt aus der Sicht eines Benutzers und seinen Ansprüchen an das System. In Abschnitt 6.1 erfolgt eine Einführung in das Prozess-Management-System ADEPT. Probleme, die bei der Umsetzung auftraten, werden begleitend besprochen.

6.1 Das Prozessmanagementsystem ADEPT

ADEPT ist ein Prozessmanagementsystem. ADEPT steht für Application Development Based on Encapsulated Pre-Modeled Process Templates. Es wird an der Universität Ulm von der Abteilung Datenbanken und Informationssysteme entwickelt. Es dient als Basis, um prozessbasierte Informationssysteme zu realisieren. Durch ein einfaches aber ausdrucksstarkes Prozessmetamodell sollen reale betriebliche Abläufe realisiert werden. Das Prozessmetamodell erlaubt eine grafisch anschauliche Darstellung des Prozessmodells. Neben dynamischen Ablaufänderungen (ad-hoc) werden auch Änderungen des Prozessschemas (Schemaevolution) und die anschließende Übertragung auf bereits gestartete Instanzen unterstützt. Korrektheitskriterien werden automatisch zur Modellierungs- und Laufzeit geprüft. Dabei wurde Wert auf die effiziente Überprüfung zur Modellier- und Laufzeit gelegt. Dabei wird neben Verklemmungsfreiheit, Erreichbarkeit und Terminierbarkeit auch der Datenfluss überprüft. Erst durch die effiziente Prüfung zur Laufzeit wird sichergestellt, dass ad-hoc Änderungen syntaktisch korrekt sind und vom Benutzer akzeptiert werden. ADEPT ruft mit Aktivitäten verknüpfte Anwendungskomponenten auf. Es versorgt sie mit benötigten Eingabedaten und übernimmt Ausgabedaten. Im Rahmen des PMS auftretende Ereignisse wie Start- und Endzeiten und Bearbeiter werden in einer Ablaufhistorie gespeichert. Änderungen werden in einer Änderungshistorie mitgeführt.

6.1.1 Prozess-Meta-Modell

Die Kontrollflussmodellierung basiert auf einem blockstrukturierten, graphbasierten Ansatz. Durch die regelmäßige Blockstrukturierung werden Kontrollstrukturen (Sequenz, Verzweigung, Schleife) auf Blöcke mit eindeutiger Start- und End-Aktivität abgebildet. Blöcke dürfen sich nicht überschneiden, können aber beliebig ineinander verschachtelt sein. Damit besitzt ein Prozess auch eine eindeutige Start- und

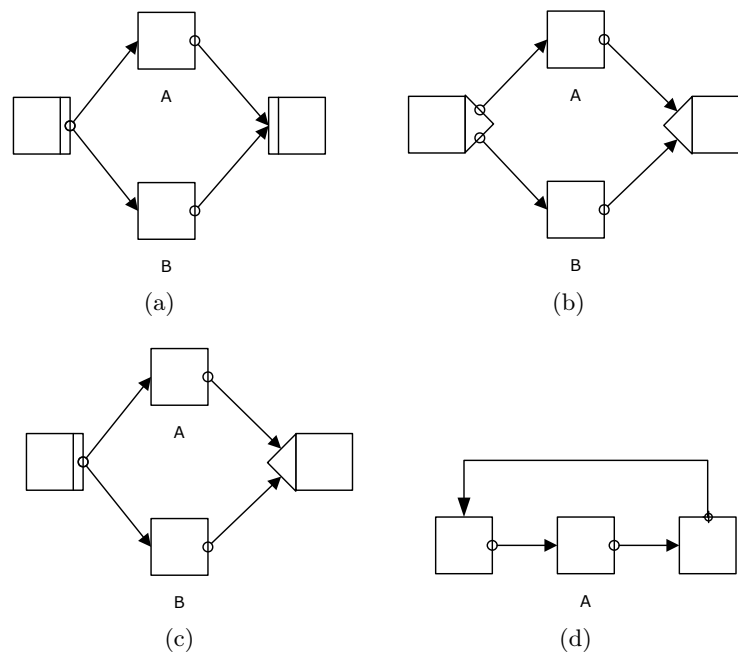


Abbildung 6.1: Kontrollflussstrukturen in ADEPT: (a) parallele Verzweigung, (b) bedingte Verzweigung, (c) parallele Verzweigung mit finaler Auswahl und (d) eine Schleife

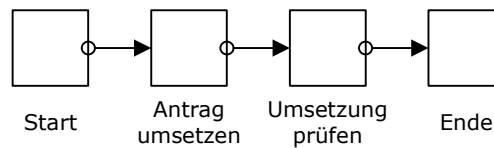


Abbildung 6.2: Sequenz in ADEPT

Endaktivität. Der Prozess besteht aus ineinander geschachtelten Blöcken, die durch Kanten verbundene Aktivitäten enthalten. Abbildung 6.2 zeigt in ADEPT mögliche Kontrollstrukturen. In ADEPT wird nicht nur der Kontrollfluss sondern auch der Datenfluss explizit dargestellt. Globale Prozessvariablen (Datenelemente) verwalten den Datenkontext. Jedes Datenelement enthält einfache oder komplexe Datenstrukturen. Der Datenfluss zwischen den Aktivitäten wird über diese Datenelemente dargestellt. Die Ausgabedaten einer Aktivität werden über Schreibkanten in ein Datenelement geschrieben. Nachfolgende Aktivitäten lesen über eine Lesekante den Wert aus dem Datenelement. Ein korrekter Datenfluss ist sichergestellt, wenn vor jedem Lesezugriff mindestens ein Schreibzugriff auf ein Datenelement erfolgt ist. In Abbildung 6.3 wird ein Lese- und Schreibzugriff auf ein Datenelement skizziert.

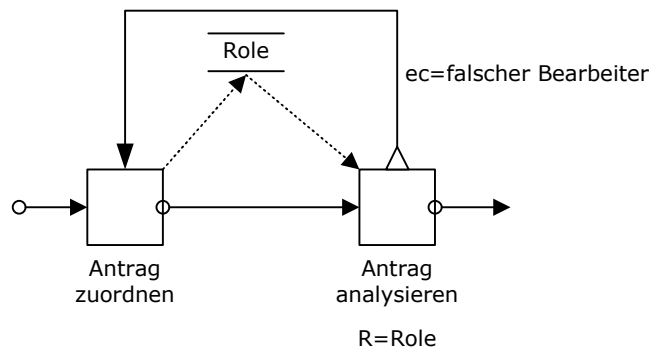


Abbildung 6.3: Die Modellierung des Datenfluss in ADEPT; dynamische Bearbeiterzuordnungen aus Datenelementen und die Verwendung einer Fehlerkante

6.1.2 Organisationsmodell

In [Spa01] wird das Organisations-Metamodell von ADEPT vorgestellt (vgl. Abbildung A.1 im Anhang). Basierend auf seinen Entitäten wie Rolle, Stelle, Arbeitsgruppe oder Organisationseinheit wird ein konkretes Organisationsmodell erzeugt. Ausführungs-, Zugriffs- und Änderungsrechte werden auf seiner Grundlage mit Bearbeiterregeln ausgedrückt. Die kleinste Einheit einer Bearbeiterregel ist eine Bearbeiterzuordnung. Soll eine Aktivität **Antrag umsetzen** durch die Rolle Experte ausgeführt werden, wird dies durch die Bearbeiterzuordnung $R = \text{Experte}$ dargestellt. Soll ausgedrückt werden, dass der Experte auch noch die Fähigkeiten über Echtzeitbetriebssysteme haben soll, entsteht daraus ein Bearbeiterausdruck der zwei Bearbeiterzuordnungen immer mittels UND verbindet: $(R = \text{Experte}) \text{ AND } (F = \text{Echtzeitbetriebssysteme})$. Kombiniert man diese Bearbeiterausdrücke mittels ODER, UND_NICHT entsteht eine Bearbeiterregel wie: $(R = \text{Experte} \text{ AND } F = \text{Echtzeitbetriebssysteme}) \text{ OR } (OE = \text{Systembereich})$. In den Bearbeiterzuordnungen können auch Datenelemente des Datenkontexts referenziert werden. Dadurch können Bearbeiterzuordnungen zur Laufzeit verändert werden, indem z. B. die gewünschte Rolle in ein Datenelement geschrieben wird und dieses in der Bearbeiterzuordnung referenziert wird. In Abbildung 6.3 wird in der Aktivität **Zuordnung festlegen** definiert, wer die nächste Aktivität ausführt. Durch diese Bearbeiterregeln wird so festgelegt, wer die eine Aktivität in seiner Arbeitliste angeboten bekommt. Dies können auch mehrere Mitarbeiter sein.

Die Organisationsstruktur ist auch Änderungen unterworfen. Diese sind in 4.2.2 beschrieben. In [WC02] werden, basierend auf dem ADPET Organisationsmodell, Änderungsoperationen auf den Entitätstypen (Rolle, Organisationseinheit, Stelle etc.) dargestellt. Konkret werden das Einfügen, Löschen und die Veränderung von Entitäten unterstützt. Wo dies sinnvoll ist, werden auch komplexere Operationen wie Teilen, Vereinigen und Verschieben von Entitäten unterstützt. Betrachtet werden dabei Veränderungen des Organisationsmodells, daraus notwendig werdende Veränderungen der Bearbeiterregeln und evtl. notwendige Anpassungen der Arbeitslisten.

6.1.3 Abweichung vom Standardablauf

ADEPT unterstützt sowohl planbare als auch nicht planbare Abweichungen vom Standardablauf. Planbare Abweichungen sind bereits zur Modellierungszeit bekannt. Sie können daher im Prozessschema berücksichtigt werden. Nicht planbare Abweichungen, so genannte ad-hoc Abweichungen, treten erst zur Laufzeit auf. Zu diesem Zeitpunkt muss dann flexibel reagiert werden. Sie können daher nicht im Schema berücksichtigt werden. ADEPT bietet beide Möglichkeiten an (nach [Rei00]).

planbare Abweichungen Für planbare Abweichungen stellt ADEPT unter anderem folgende Konstrukte bereit:

- Vorwärtssprünge mit/ohne Nachholen der übersprungenen Aktivitäten
- Rücksprünge über Fehlerkanten
- Normale und Ausnahmeaktivitäten über Knotenprioritäten und Priorisierungskanten

In Abbildung 6.4 auf der nächsten Seite wird ein Vorwärtssprung mit einer shortcut-Kante dargestellt. Nach Beendigung der Aktivität **Antrag analysieren** sind zwei Aktivitäten selektierbar: die Aktivität **Antrag klassifizieren und bewerten** des normalen Ablaufs und die Aktivität am Ziel des shortcuts (**Antrag umsetzen**). Wird jetzt die Aktivität **Antrag umsetzen** zum Starten selektiert, wird die andere Aktivität übersprungen. Ansonsten wird der Kontrollfluss weiterverfolgt.

In ADEPT kann zwischen normalen und Ausnahmeaktivitäten unterschieden werden. Dazu werden Aktivitäten Ausführungsprioritäten zugeordnet. Durch einen neuen Kantentyp, der Priorisierungskante, kann die Priorität einer Aktivität geändert werden. Durch die Priorität *regular* (engl. normal) wird der normale Ausführungspfad gekennzeichnet. Durch *exceptional* (engl. ausnahmsweise) können ausnahmebedingte Abweichungen dargestellt werden.

Ad-hoc Abweichungen Ad-hoc Abweichungen werden in ADEPT durch Änderungsoperationen unterstützt. Änderungsoperationen sind auszugsweise:

Bearbeiterzuordnung	Bedeutung
$R = x$	Benutzer ist der Mitarbeiter x
$F = x$	Benutzer hat die Fähigkeit x
$OE = x$	Benutzer ist Mitglied der Organisationseinheit x
$AG = x$	Benutzer ist Mitglied der Arbeitsgruppe x
$S = x$	Benutzer besetzt die Stelle x

Tabelle 6.1: Definition von Bearbeiterzuordnungen

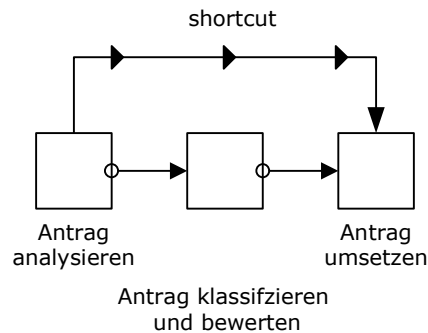


Abbildung 6.4: Planbarer Vorwärtssprung in ADEPT. Die Aktivität Antrag klassifizieren und bewerten wird übersprungen und nicht nachgeholt

- Hinzufügen, Löschen oder Verschieben von Aktivitäten und Blockstrukturen
- Einfügen einer Aktivität zwischen zwei Aktivitätsmengen
- Hinzufügen und Entfernen von Datenelementen und Datenflusskanten

Auf der Grundlage dieser und weiterer Änderungsoperationen lassen sich komplexere Operationen zusammensetzen. Das Einfügen einer Aktivität zwischen zwei Aktivitätsmengen ist ein Beispiel einer komplexen Operation. Auch diese führen nicht zu einer Verletzung der Korrektheit oder Konsistenz des Prozesses. Die regelmäßige Blockstrukturierung bleibt erhalten. Verklebungen werden verhindert und die Datenversorgung sichergestellt. Graphmarkierungen verletzen auch nach der Änderung keine der für sie geltende Regeln. Keine Aktivität kann so aktivierbar sein, wenn sie im Ablaufgraphen vor einer beendeten Aktivität liegt. Durch die Graphmarkierungen wird auch sichergestellt, dass keine Änderungen in abgeschlossenen oder nicht mehr erreichbaren Bereichen des Prozessgraphen vorgenommen werden.

Schemaevolution Änderungen können auf Prozessinstanzen und auf Prozessschemas vorgenommen werden. Bisher wurden nur Änderungen einzelner Prozessinstanzen besprochen. Im Zuge von Prozessverbesserungen oder der Einführung neuer Aktivitäten (vgl. 3.2.3 auf Seite 24) werden Änderungen des Prozessschemas nötig. Die Änderungsoperationen, die auf dem Prozessschema durchgeführt werden können, sind die gleichen, wie die obigen Instanzänderungsoperationen. Im einfachsten Fall bedeutet dies, dass bereits gestartete Prozessinstanzen nach dem alten Schema weiterlaufen und neu gestartet das neue Schema verwenden. Die alte und die neue Version des Schemas werden nebeneinander verwendet, bis alle Prozessinstanzen des alten Schemas beendet sind. Sollen die Schemaänderungen aber auch bei bereits laufenden Instanzen verwendet werden, so spricht man von Schemaevolution: Prozessinstanzen des alten Schemas sollen auf das neue migriert werden. Bei der Übertragung der Schemaänderungen auf laufende Prozessinstanzen muss man neben dem Ausführungszustand der Instanz auch beachten, ob die Instanz schon individuell verändert worden

ist. Vor allem sollte es möglich sein, auch auf dem neuen Schema wieder Instanzänderungen durchzuführen. Schemaevolution und Prozessinstanzänderungen schließen sich nicht gegenseitig aus. Ziel ist es auch hier, die Korrektheits- und Konsistenzkriterien zu erhalten und die Migration einer großen Anzahl von Instanzen automatisch und effizient durchzuführen. Details sowie formale Grundlagen finden sich in [RRD04a], [RRD04b]. Die Schemaevolution in ADEPT kann Instanzen ohne Änderung und auch mit Änderungen auf ein neues Schema migrieren. Bei Instanzen mit Prozessänderungen muss unterschieden werden, ob die Änderungen sich überlappen oder nicht. Überlappen sich die Änderungen nicht, ist ebenfalls eine Schemaevolution möglich. Bei sich überlappenden Änderungen ist dies nicht in jedem Fall möglich. Auf eine detaillierte Darstellung wird hier verzichtet.

6.2 Umsetzung

ADEPT wird nun verwendet, um die ausgewählte Aspekte der Fallbeispiele und der Kategorisierung zu untersuchen. Ein vollständiges und einsetzbares prozessbasiertes Informationssystem ist nicht das Ziel. Anforderungen der Anwender sollen an den Fähigkeiten von ADEPT gespiegelt werden.

In Abbildung B.16 auf Seite 92 ist ein Änderungsmanagement Prozess in ADEPT dargestellt. Die Aktivitäten sind wie in B.6 auf Seite 91 benannt. Die Bearbeitungsordnung wird hier nicht extra aufgeführt und entspricht der Darstellung in B.6. Ein Prüfzyklus wurde als Schleife um die Aktivitäten **Zuteilung Entwicklungszyklus**, **Antrag umsetzen** und **Umsetzung prüfen** realisiert. Aufgrund der Blockstrukturierung besitzt ADEPT immer genau einen definierten Start und End-Knoten. Der Prozess wird beendet, indem die End-Aktivität ausgeführt wird. Soll nach einer Aktivität der Prozess beendet werden, ohne dass weitere Aktivitäten nach dieser ausgeführt werden, muss ein Pfad zum Endknoten bestehen. Dies soll in den Aktivitäten **Antrag bewerten**, **Auswirkungen ermitteln**, **Auswirkungen bewerten** möglich sein. Wird die Bearbeitung des Antrag in einer dieser Aktivitäten abgelehnt, soll **Antrag ablehnen** ausgeführt werden. Wird er hingegen umgesetzt, **Antrag abschließen**. Dies wird durch eine bedingte Verzweigung erreicht. Entweder die Bearbeitung wird fortgeführt, oder vorzeitig beendet. Betrachtet man den Graphen, ist dieses Verhalten nicht sofort ersichtlich.

In den folgenden Abschnitten wird anhand ausgewählter Beispiele dargestellt, wie sich mit einfachen Mitteln von ADEPT die in den Prozessen benötigte Flexibilität erreichen lässt.

Prüfzyklus abbrechen Soll die Bearbeitung von innerhalb des Prüfzyklusses eingestellt werden wie in $\ddot{A}M(G)$ (vgl. Abbildung B.3 auf Seite 91), ist dieses Vorgehen nicht möglich. Die Möglichkeit der erneuten Entscheidung über das weitere Vorgehen in jeder Iteration des Prüfzyklus ist aber erwünscht. Schlägt eine Umsetzung fehl, kann nicht immer festgestellt werden, warum die Umsetzung fehlschlägt. Es muss erst festgestellt werden, ob es eine unklare Beschreibung des ursprünglichen Antrags war,

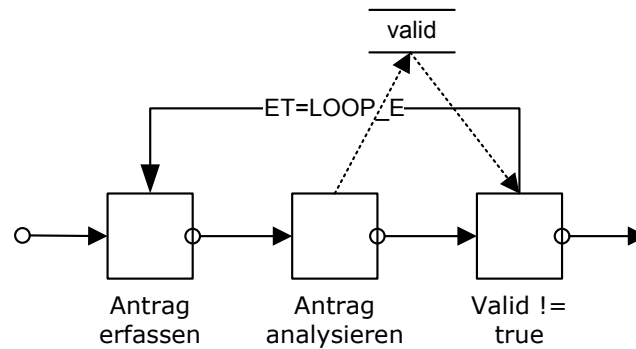


Abbildung 6.5: Informationen nachfordern

ein falscher Lösungsweg eingeschlagen wurde oder einfach nur die Umsetzung selbst Fehler aufweist. Bei der Prozessmodellierung ist also noch nicht klar definierbar, ab wo der Prozess iteriert werden soll. Diese Art der Vorgehensweise kann nicht ausreichend in ADEPT dargestellt werden. Sie wird daher im nächsten Kapitel näher untersucht.

Überspringen nicht benötigter Aktivitäten Ist die Ausführung von Aktivitäten nicht in jedem Fall nötig, lassen sich diese überspringen. Die übersprungenen Aktivitäten können dann entweder später noch zur Ausführung kommen (Überspringen mit Nachholen) oder nicht (Überspringen ohne nachholen). Ist bereits bei der Modellierung bekannt, welche Aktivitäten übersprungen werden sollen, kann dies vormodelliert werden. Da dies nicht in jedem Fall möglich ist und nicht alle Abweichungen vormodelliert werden können/sollen, lassen sich diese Vorwärtssprünge auch dynamisch zur Laufzeit anwenden.

Abbildung 6.4 auf Seite 61 stellt einen vormodellierten Vorwärtssprung dar. Ist die Aktivität **Antrag erfassen** beendet, wird sowohl **Antrag klassifizieren und bewerten** als auch **Antrag umsetzen** aktiviert. Wird dann **Antrag umsetzen** zum Starten selektiert, werden alle Aktivitäten dazwischen ausgelassen (hier **Antrag klassifizieren und bewerten**). Vorwärtssprünge lassen sich auch dynamisch zur Laufzeit realisieren. Aktivitäten, die noch nicht aktiviert sind, lassen sich als Sprungziel auswählen (bereits laufende Aktivitäten müssen daher abgebrochen und zurückgesetzt werden). Der Zielknoten muss auch deterministisch erreichbar sein, darf also z. B. nicht in dem Ast einer alternativen Verzweigung liegen. Der Benutzer muss sich um diese Bedingungen nicht kümmern. Das System bietet ihm nur anspringbare Aktivitäten zur Auswahl an. Ein Prüfzyklus, der eine Schleife darstellt, lässt sich so abbrechen. Als Sprungziel muss dazu eine Aktivität außerhalb der Schleife gewählt werden.

Durch das vorhergeplante oder dynamische Überspringen von Aktivitäten lässt sich vom vorgeplanten Ablauf abweichen, wenn der Prozess, wie in 3.3.3 auf Seite 29 beschreiben, nicht der Auslöser des Ablaufs ist.

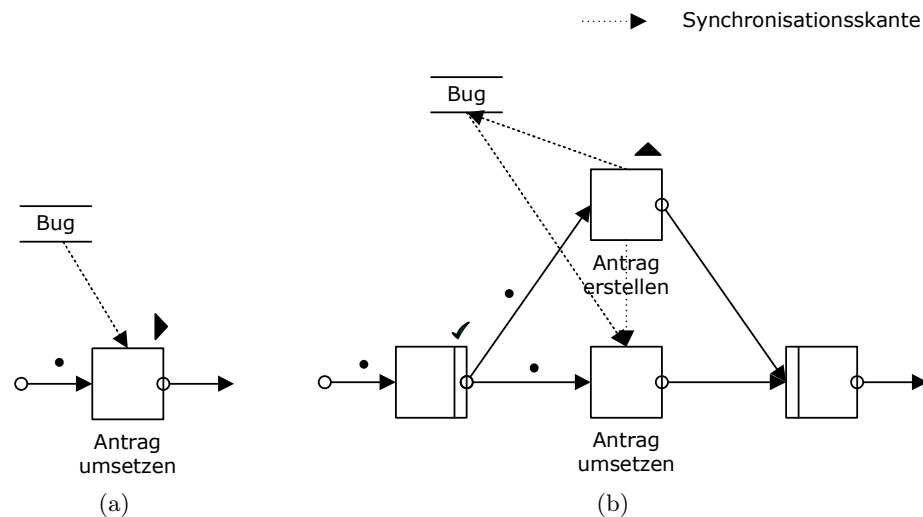


Abbildung 6.6: Informationen nachfordern (dynamischer Fall): (a) zeigt einen Ausschnitt des Prozessgraphen vor, (b) nach den Einfügeoperationen.

Informationen nachfordern Informationen müssen immer dann nachgefordert werden, wenn in einer Vorgängeraktivität ein Sachverhalt beschrieben wird, diese Beschreibung aber nicht ausreicht, um in der Nachfolgeraktivität damit zu arbeiten. Im einfachsten Fall sind das zwei direkt aufeinander folgende Aktivitäten. Ist zusätzlich bekannt, dass dies häufig der Fall ist, lässt sich dies einfach vormodellieren.

Dies beruht auf der Entwicklung von $\ddot{A}M(Gv1)$ und $\ddot{A}M(Gv2)$ in Abschnitt 3.3.3 auf Seite 27. In $\ddot{A}M(Gv1)$ wird mit einer expliziten **Rücksprache** Aktivität begonnen. Diese wird dann entfernt, in der letzten Variante des Prozesses aber durch eine Schleife über die Aktivitäten **Antrag erfassen** und **Antrag zuordnen** substituiert. Auch $\ddot{A}M(Gv2)$ besitzt die Aktivität **Rücksprache**, entfernt diese, führt sie dann aber doch wieder ein. Auch $\ddot{A}M(M)$, $\ddot{A}M(StgV)$, $\ddot{A}M(SP)$ und $\ddot{A}M(EL)$ sehen Aktivitäten vor, um fehlende Informationen zu ergänzen. Dies zeigt die Wichtigkeit einer flexiblen Behandlung dieses Falls.

In Abbildung 6.5 auf der vorherigen Seite ist dies dargestellt. Eine einfache Schleife wird solange ausgeführt bis die Informationen in der gewünschten Detailtiefe bereitstehen. Sind die beiden Aktivitäten nicht direkt aufeinander folgend und/oder tritt dieser Fall nicht so oft auf, lässt sich durch die dynamischen Änderungsoperationen eine Lösung modellieren. Diese wird in Abbildung 6.6 dargestellt. Dies tritt auf, wenn in $\ddot{A}M(M)$ bei der Aktivität **Antrag bewerten** festgestellt wird, dass in der Beschreibung des Antrags **Antrag erfassen** Informationen fehlen, um eine Bewertung festzulegen oder in **Antrag implementieren** Detailinformationen des Antragstellers eingeholt werden müssen. Er muss also seine Aktivität **Antrag erstellen** überarbeiten. Dazu wird **Antrag umsetzen** zurückgesetzt und **Antrag erstellen** parallel dazu eingefügt. Zusätzlich wird noch eine einfache Sync-Kante von **Antrag erstellen** nach **Antrag**

umsetzen eingefügt. Diese Kante besagt, dass **Antrag umsetzen** erst dann aktiviert/gestartet werden darf, wenn **Antrag erstellen** beendet ist oder nicht mehr zur Ausführung kommt. Die Aktivität **Antrag erstellen** (die dem gleichen Bearbeiter wie das Erste **Antrag erstellen**) wird sofort durch den Bearbeiter ausführbar. **Antrag umsetzen** hingegen erst, wenn die dazu parallele Aktivität beendet wird. Abbildung 6.6(b) auf der vorherigen Seite zeigt einen Ausschnitt des Prozessgraphen vor, Abbildung 6.6(b) nach den Einfügeoperationen.

Damit wird allerdings noch nicht ganz das gewünschte Verhalten erzielt. **Antrag umsetzen** sollte nur unterbrochen werden und sobald **Antrag erstellen** beendet ist, wieder aktivierbar werden. Durch das Zurücksetzen der Aktivität in obiger Darstellung gehen evtl. Arbeitsergebnisse verloren, wenn der Informationsmangel nicht sofort entdeckt wird.

Schnelle Implementierung Hier wird nun eine Möglichkeit vorgestellt, um z. B. nach Beginn der Serienproduktion Probleme auf schnelle Art und Weise zu beheben ohne dazu das prozessbasierte Informationssystem zu umgehen (Kategorie Projekt/planbar-zur Laufzeit). Bei der Verwendung von PVCS trat die Folge auf, dass ein Antrag eingereicht wurde und direkt danach implementiert wurde (als Teil einer anderen Problemlösung o. Ä.). Auch dieses Beispiel, in dem das prozessbasierte Informationssystem nicht Auslöser von Handlungen ist, sondern den in der Realität anstehenden Aufgaben folgen muss (oder nicht verwendet wird), zeigt welche Art der Flexibilität benötigt wird. Es beruht auch auf der Beschreibung in Kapitel 3.2.2 auf Seite 23.

Dazu wird parallel zum bisher geplanten Ablauf wie in Abbildung B.16 auf Seite 92 ein paralleler Ausnahmepfad angeboten. Dieser realisiert die schnelle Implementierung. Der Prozess wird wie in Abbildung 6.7 auf der nächsten Seite dargestellt angepasst. Jedem Aktivitätsknoten wird eine Ausführungspriorität zugeordnet. Reguläre Arbeitsschritte werden mit REGULAR, Ausnahmeschritte mit EXCEPTIONAL gekennzeichnet. Aus Benutzersicht stellt sich der Ablauf nun so dar. Nach Beendigung der Aktivität **Antrag erstellen** wird sowohl die Aktivität **Antrag bewerten** als auch die Aktivität **Dringende Problembehebung** aktiviert. Die dringende Fehlerbehebung wird in der Benutzerschnittstelle als Ausnahmeaktivität gekennzeichnet. Sie können in einer zweiten Arbeitsliste erscheinen oder farblich gekennzeichnet sein. Dieser parallele Zweig ist dann jedoch nicht auf diese eine Aktivität beschränkt. Durch die Anwendung der Änderungsoperationen lassen sich Aktivitäten wie **Umsetzung prüfen** dynamisch einfügen, um so den konkreten Bedürfnissen gerecht werden zu können. Die problemspezifische Anpassung des Ausnamezweigs ist somit gegeben. Sie kann mit – und nicht wie bisher um das System herum – realisiert werden.

6.2.1 ADEPT und die Kategorisierung

In Kapitel 4 auf Seite 41 sind Kategorien nötiger Änderungen untersucht worden. Hier wird nun kurz dargestellt, mit welchen Mechanismen diese unterstützt werden können bzw. welche dafür nötig sind.

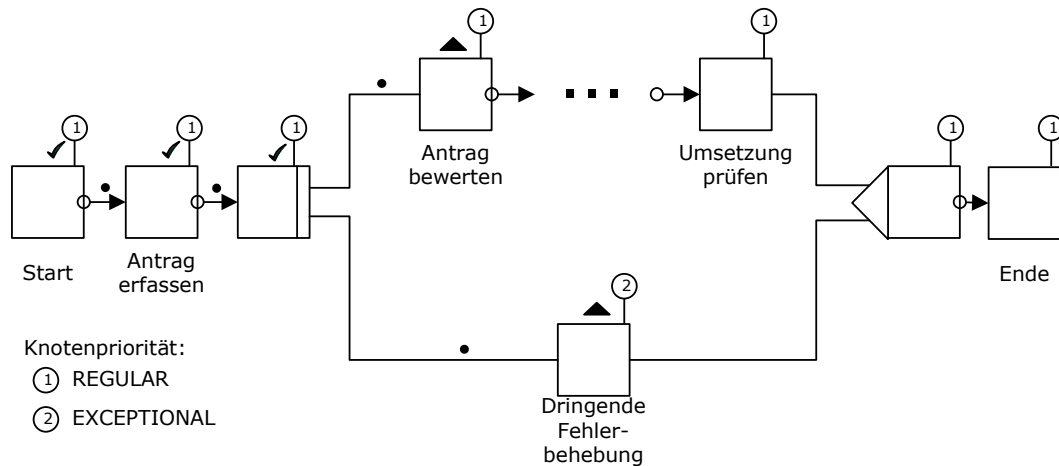


Abbildung 6.7: Ausnahmepfad in ADEPT, um einen schnellen alternativen Implementierungspfad anzubieten. Aus Benutzersicht ist dies eine Abkürzung des normalen, mit vielen Prüfungen versehenen Pfades.

Projekt Änderungen, die vor Projektbeginn durchgeführt werden müssen, machen einen Großteil der Änderungswünsche aus, mit denen sich die Prozessexperten konfrontiert sehen. Wie sich gezeigt hat, bestehen die Änderungen selten aus komplett neu zu erstellenden Aktivitätsvorlagen. Ausgehend von einem gegebenen Prozessmodell lassen sich in ADEPT eine Vielzahl von Anpassungen auf einfache Art und Weise erledigen. Durch das Verschieben und Löschen von Aktivitäten lässt sich bereits ein Großteil der gewünschten Änderungen realisieren. Dabei werden diese Änderungen auch zur Modellierungszeit durch die semantisch hoch stehenden Änderungsoperationen unterstützt. Für Änderungen sind also keine zu großen Kenntnisse nötig. Auch die Korrektheit wird gewährleistet. Andererseits ist es auch denkbar, projektspezifische Prozesse nicht ausgehend von einem vorgegebenen Prozessmodell anzupassen, sondern dies basierend auf einzelnen Komponenten zusammenzubauen: durch die komponentenbasierte Entwicklung prozessorientierter Informationssysteme. Dieses Ziel hat sich das AristaFlow Projekt gesetzt. In AristaFlow wird ADEPT als Grundlage verwendet (mehr in [AAD⁺04])

Änderungen zur Laufzeit, die vorhersehbar sind, können durch Mechanismen der Schemaevolution unterstützt werden. Es ist nicht mehr nötig, ein Prozessmodell zu erstellen, das einen Kompromiss zwischen den Anforderungen zu Beginn des Prozesses und denen kurz vor Ende darstellt. Vor allem Querschnittsprozesse wie die des Änderungsmanagements werden so besser unterstützt. Es können mehrere Versionen des Schemas erstellt werden und diese dann sukzessive per Schemaevolution verwendet werden. Auch bereits laufende Instanzen können dann nach dem neuen Schema laufen.

Nicht planbare Änderungen, die die Abweichung vom vorgesehenen Fall notwendig machen, werden von ADEPT unterstützt. Sowohl Änderungen einzelner Instanzen als

auch die Änderung des Schemas sind möglich. Treten Qualitätsprobleme auf, sind so zusätzliche Prüfungen in das Schema integrierbar und auf bereits laufende Instanzen propagierbar. Diese Möglichkeiten gehen deutlich über die Fähigkeiten der bisher verwendeten System hinaus, bei denen diese Änderungen entweder arbeitsintensiv waren (a-priori) oder gar nicht unterstützt wurden.

Aufbauorganisation ADEPT bietet auch die Möglichkeit, die Änderungen der Aufbauorganisation zu berücksichtigen, wie in 6.1.2 auf Seite 59 angedeutet.

Technologie Planbare Änderungen zur Projektlaufzeit, vor allem immer wiederkehrende, wie die Änderung der Entwicklungsplattform, werden unterstützt. Die Grundvoraussetzung, um Änderungen des Schemas zur Projektlaufzeit immer wieder durchzuführen, ist eine systematische und konsistenzhaltene Möglichkeit dies zu tun: Schemaevolution. Änderungen können dann durch mehrere versionierte Schemas realisiert werden. Ebenso anbieten würde sich hier ein Mechanismus wie in Abschnitt 7.1 auf Seite 69 beschreiben. Mehrere Änderungsoperationen, die nötig sind, um einen bestimmten Prozess von der A-Muster auf die B-Muster Entwicklung umzustellen, können so einheitlich zugänglich gemacht werden.

Ist eine neue Technologie ausgereift genug, um z. B. Tests automatisiert ablaufen zu lassen, muss eine neue Aktivitätenvorlage realisiert werden. Diese versorgt dann die Anwendung/Funktion zur automatischen Testdurchführung mit Eingabeparametern und übernimmt die Ausgabeparameter. Das Schema muss dann anschließend geändert werden, um die neue Aktivität einzubinden und kann dann mittels Schemaevolution propagiert werden.

Kultur Die apriori Änderung der Mensch-Maschine Schnittstelle betrifft weniger das PMS, als das PAIS, mit dem der Benutzer letztendlich arbeitet. Beispielhaft sind in 4.2.4 auf Seite 50 notwendige Änderungen genannt, die hier jedoch nicht weiter vertieft werden sollen.

Die nicht planbaren Änderungen sind hier von größerem Interesse. Es lässt sich nur schwer abschätzen, wie genau sich ein Prozess zwischen den Kulturen verändern muss, um angemessen zu bleiben. Für Prozess/ prozessbasiertes Informationssystem, das in Europa von den Benutzern akzeptiert wird und funktioniert, muss nicht das Gleiche in Asien gelten. Vielmehr werden Anpassungen nötig sein, die das PMS unterstützen muss. Eine hierarchische Denkweise könnte sich in einer anderen Organisationsstruktur und klar vorgegebenen Aktivitäten niederschlagen.

Strategie Die geplante Einführung von Prozessverbesserungen und deren evtl. notwendige Rücknahme (planbar - zur Laufzeit) werden ebenfalls durch Mechanismen der Schemaevolution ermöglicht.

Bei nicht planbaren Änderungen wurde ein Beispiel gefunden, bei dem strukturelle Änderungen nicht ausreichen, um auf die neue Situation flexibel reagieren zu können. Wird ein Hardwarensensor durch einen virtuellen Sensor ersetzt, betrifft dies nicht nur

die Struktur (Hardwaretest durch Simulation ersetzen), sondern auch Entscheidungen. Bei diesen muss überprüft werden, ob sie im neuen Kontext noch Gültigkeit besitzen. Dieses Themengebiet wird in Abschnitt 7.2 auf Seite 71 vertieft.

Politsch-Rechtlich Diese Kategorie kann zwar als Ursache für Änderungen ausgemacht werden, die sich daraus ergebenden Änderungen unterscheiden sich jedoch nicht von den bisherigen. Daher wird hier nicht weiter darauf eingegangen.

6.3 Zusammenfassung

Die von Entwicklungsprojekten benötigte Flexibilität wird auf ein konkretes Prozess-Management-System abgebildet. Durch die Verwendung eines adaptiven Prozess-Management-Systems, wie ADEPT, lässt sich zeigen wie die Beschränkungen konventioneller Systeme umgangen werden können. Ein PAIS kann somit eine flexiblere Arbeitsweise bieten, anstatt die Arbeit zu beschränken und zu behindern.

Es wurde gezeigt, durch welche Mechanismen ein Prozess-Management-System die in den Fallbeispielen und der Kategorisierung dargestellte Flexibilität unterstützen kann. Nicht alle Möglichkeiten, die ADEPT bietet, entsprechen den Denkmustern der Prozessbeteiligten.

Kapitel 7

Erweiterungsvorschläge am Beispiel von ADEPT

In diesem Kapitel werden Erweiterungsvorschläge am Beispiel von ADEPT dargestellt. Darin wird die benötigte Prozessflexibilität in der Denkweise der Benutzer auf ADEPT abgebildet. Abschnitt 7.1 beschreibt, wie Änderungen um semantische Informationen angereichert werden können. In Abschnitt 7.2 werden grundlegende Szenarien vorgestellt, um den Ausgangspunkt darzustellen. Dann werden in 7.2.1 und 7.2.2 Grundlagen der späteren Abbildung der Szenarien geschaffen. Diese werden dann in Abschnitt 7.3 dargestellt. Abschnitt 7.4 fasst die gewonnen Erkenntnisse zusammen.

7.1 Integration semantischer Informationen und Prozessänderungen

Die in dieser Arbeit vorgestellten Änderungen und deren Kategorisierung beruhen auf dem Wissen der Prozessexperten und einer post mortem Analyse der Prozesse. Die Gründe und der Kontext von Änderungen waren im nachhinein nur schwer zu rekonstruieren. Eine Integration von ADEPT und CBRFlow verspricht diese Situation zu verbessern. Die Darstellung ist im Wesentlichen aus [WRWR05] und [RWRW05] entnommen. Prozessinstanzänderungen werden um semantische Informationen ergänzt. Diese enthalten den Kontext und den Grund der Änderung.

Der Ansatz basiert auf fallbasiertem Schließen (engl. Case-Based-Reasoning; CBR). Es ist ein maschinelles Lernverfahren zur Problemlösung durch Analogieschluss. Neue Probleme werden durch die Anwendung vergangener Erfahrungen gelöst. Kontinuierliches Lernen wird somit ermöglicht. Informationen zu dem Problem und seiner Lösung werden gespeichert und sind so sofort verfügbar, um zukünftige Probleme zu lösen. Dialogorientiertes fallbasiertes Schließen (Conversational Case-Based-Reasoning; CCBR) ist eine Erweiterung des fallbasierten Schließens. Der Benutzer wird aktiv am Finden der Lösung zu einem gegebenen Problem beteiligt. Er muss das Problem nicht vollständig spezifizieren können, sondern wird durch das System durch Frage-Antwort-Paare zu möglichen Lösungen geführt. CCBR kann damit auch eingesetzt werden, wenn das Problem nicht vollautomatisch gelöst werden kann. Das zentrale Element des Ansatzes ist die Fallbasis. Es enthält eine Menge von Fällen. Ein Fall beschreibt

eine konkrete ad-hoc Änderungen einer Prozessinstanz. Es besteht aus einer Problembeschreibung, einer Menge von Frage-Antwort Paaren und einem Lösungsteil. Der Lösungsteil beschreibt die durchzuführende ad-hoc-Modifikation durch eine Menge von Änderungsoperationen (z. B. Einfügen, Löschen von Aktivitäten). Durch die Frage-Antwort-Paare wird der Grund der Änderung beschrieben. Der Lösungsteil beschreibt den genauen Kontext des Prozesses, in dem die Änderung durchgeführt wurde. Es wird auch noch vermerkt wie oft ein Fall angewendet wurde. Für jedes Prozessschema existiert eine eigene Fallbasis.

Dadurch wird es möglich das Wissen über ad-hoc Instanzänderungen zu bewahren und in ähnlichen, zukünftigen Situationen wieder zu verwenden. Muss der Benutzer vom vordefinierten Ablauf abweichen, wird er dabei durch die Fallbasis unterstützt. Aus einer Menge dargestellter Fragen beantwortet er einige in beliebiger Reihenfolge. Die CCBK-Komponente ermittelt daraus eine Menge ähnlicher Fälle und stellt diese nach Relevanz sortiert dar. Zusätzlich lässt sich ein Filter angeben, der eine Änderungsoperation und/oder ein Änderungsobjekt enthält. Dieser Filter hilft dem Nutzer dabei, die Anzahl der Fälle einzuschränken und den gewünschten Fall zu finden. Erinnert sich der Benutzer vage daran, dass die Lösung seinen Problems etwas mit dem Einfügen der Aktivität **Analyse intern/extern** zu tun hat, kann er gezielt danach suchen. Verwendet er dann einen der Fälle, wird der Zähler des Falls um eins erhöht. Kann er keinen passenden Fall finden, muss er einen neuen definieren. Ad-hoc Änderungen, wie oben beschrieben, müssen dann noch um die Informationen für die Fallbasis erweitert werden (Problembeschreibung etc.). Es erfordert jedoch mehr Wissen, einen neuen Fall zu definieren, als einen alten wieder zu verwenden. Diese Art der ad-hoc Änderung versteckt die Komplexität von ad-hoc Änderungen vor dem Benutzer. Sie entspricht auch eher seinem mentalen Modell. Er kennt sein Problem und evtl. einen Teil der Lösung und kann dadurch den entsprechenden Fall finden. Wissen das nötig ist, um eine Menge von Änderungsoperationen anzuwenden wird so verringert. Die semantisch hoch stehenden Änderungsoperatoren, bewegen sich damit näher in Richtung des Anwendungskontexts.

Es existiert auch ein Mechanismus, um die Qualität der Fälle zu garantieren. Dazu wird ein Reputationsmechanismus verwendet. Die Reputation eines Falles zeigt an, wie erfolgreich die Verwendung eines Falls bei der Lösung des Problems war. Wird ein Fall aus der Fallbasis verwendet, erhält der Benutzer automatische eine Bewertungsaktivität in seine Arbeitsliste. Darin kann ein optionaler Kommentar sowie eine positive (1), neutrale (0) oder negative (-1) Bewertung angegeben werden. Aus der Differenz der Anzahl der positiven und negativen Bewertungen ergibt sich der Reputationswert eines Falles. Dieser Reputationswert wird dann bei der Wiederverwendung der Fälle mit angezeigt. Stark Negativ bewertete Fälle können dann vom Prozessexperten deaktiviert werden. Sie sollten nicht gelöscht werden, da sie nicht erfolgreiche Lösungswege dokumentieren. Dadurch kann die semantische Korrektheit der Fälle sichergestellt werden, die syntaktische Korrektheit wird schon durch ADEPT selber sichergestellt.

Die Fallbasis kann auch dazu verwendet werden das Prozessschema zu ändern. Dies beruht auf der Erfahrung, dass Änderungen des Prozessschemas auf vorhergehende ad-hoc Änderungen zurückzuführen sind. Der Prozessexperte erhält durch die

Fallbasis eine größere Unterstützung als dies ohne möglich ist. Er erfährt nicht nur den genauen Kontext und die durchzuführenden Änderungen (Lösungsteil), sondern auch die Ursache der Änderung. Die Reputation eines Falles und die Anzahl seiner Verwendung sind Indikatoren, welche Fälle in das Schema übernommen werden sollen. Schemaänderungen müssen jedoch nicht identisch mit den ad-hoc Änderungen sein. Ad-hoc Änderungen beziehen sich auf einen konkreten Fall, während das Schema den allgemeinen abdeckt. Wird als ad-hoc Änderungen die Aktivität **Analyse intern/extern** eingefügt, so wird sie im Prozessschema als bedingte Verzweigung eingeführt, falls externe Mitarbeiter beteiligt sind. Wird ein Schema geändert, so kann auch die dazugehörige Fallbasis übernommen werden. Die darin enthaltenen Fälle müssen jedoch (halbautomatisch) durch den Prozessexperten angepasst werden. Fälle, die in der Schemaänderung enthalten sind, müssen nicht übernommen werden. Fälle, die sich auf einen Teil des Schemas beziehen, der nicht mehr vorhanden ist, dürfen nicht übernommen werden.

7.2 Szenarien

Wie sich in 3.3.3 auf Seite 29 gezeigt hat, ist es vorteilhaft, die Bearbeitung eines Prozesses jederzeit unterbrechen zu können (um sie genau an diesem Punkt später wieder fortzusetzen) und die Bearbeitung frühzeitig zu beenden. Wird die Bearbeitung einer Instanz frühzeitig beendet, sollen die Ergebnisse der bisherigen Arbeit nicht verloren gehen. Die getroffenen Entscheidungen müssen nachvollziehbar sein. Würden die Auswirkungen der Aktivitäten zurückgenommen, wäre dies nicht der Fall. Wichtiger ist jedoch: Entscheidungen werden revidiert. Änderungen des Kontexts können dazu führen, dass getroffene Entscheidungen hinfällig werden und überdacht werden müssen. Eine getroffene Entscheidung erweist sich als falsch und muss zurückgenommen werden. Dies wird nun im Kontext des Änderungsmanagements, vor allem in Bezug auf (frühzeitig) beendete Prozessinstanzen betrachtet.

Ein Projekt zur Entwicklung eines eingebetteten Steuergeräts wird gestartet. Dabei wurde beschlossen, einen Hardwarensensor zur Messung bestimmter Werte zu verwenden. Das Projekt wird dann gestartet. Es wird ein projektspezifischer Prozess aufgesetzt, bei dem Aktivitäten zur Prüfung der Sensorfunktionen in einem Prüfstand vorgesehen sind. Entscheidungen werden im Projektverlauf getroffen, basierend auf der Annahme dieses Hardwarensensors. Nach einem Jahr wird beschlossen, den Hardwarensensor durch einen virtuellen Sensor zu ersetzen. Dies geschieht z. B. aus Kostengründen. Dies hat weit reichende Auswirkungen auf das Projekt. Betrachtet man die Auswirkungen aus Sicht des Änderungsmanagement, ergibt sich Folgendes: neben den Änderungen am Testvorgehen – Hardwaretests müssen durch Simulationen ersetzt werden – hat dies auch Auswirkungen auf bereits beendete und noch laufende Instanzen des Änderungsmanagements.

In den Szenarien wird dargestellt, wie ein Benutzer gerne handeln würde. Sein oberstes Ziel ist es dabei immer, so viel seiner Arbeit wie möglich wieder zu verwenden. Wie man sieht, soll hier flexibel auf eine Änderung des Kontexts reagiert werden, der

nicht notwendigerweise strukturelle Änderungen nach sich zieht. Szenario 3 bezieht sich auf Prüfzyklen. Diese wurden in Abschnitt 3.3.4 auf Seite 32 und Tabelle 3.5 auf Seite 36 dargestellt. Prüfzyklen wurden in ADEPT mittels Schleifen modelliert.

Szenario 1 Aufgrund der Änderung im Kontext (der Verwendung des virtuellen Sensors) muss eine bereits beendete Prozessinstanz neu bearbeitet werden. Es steht fest, dass Teile der Bearbeitung nicht mehr gültig sind und neu bearbeitet werden müssen. Welche Teile dies genau sind, kann nicht genau vorherbestimmt werden. Im Falle des Änderungsmangements muss bei einem beendeten Antrag überprüft werden, ab welchem Zeitpunkt er erneut bearbeitet werden muss. Aus einer anderen Perspektive betrachtet: welche Teile der Bearbeitung müssen zurückgenommen werden?

Szenario 2 Eine Prozessinstanz wurde an einem Entscheidungspunkt abgebrochen. Der Abbruchgrund wurde dokumentiert. Durch die Änderung des Kontexts wird klar, dass der Abbruchgrund nicht mehr zutrifft. Die Instanz soll an genau diesem Abbruchpunkt fortgesetzt werden. In der Aktivität **Antrag klassifizieren und bewerten** durch das Change Control Board wurde beschlossen, einen Antrag nicht weiter zu bearbeiten. Er soll an genau dem Punkt, an dem er abgelehnt wurde, weitergeführt werden. Dieser Fall trifft auch zu, wenn zu einem späteren Zeitpunkt klar wird, dass die getroffenen Entscheidung falsch war. Eine getroffene Entscheidung soll somit revidiert werden.

Szenario 3 Es sind zwei Prüfzyklen ineinander geschachtelt wie bei ÄM(G) (vgl. Abbildung B.3 auf Seite 91). Bei der Prüfaktivität **System abnehmen** wird festgestellt, dass die Umsetzung sich nicht in das Gesamtsystem integrieren lässt. In den meisten Fällen ist es sinnvoll, danach zur Aktivität **Antrag klassifizieren und bewerten** zurückzukehren. In diesem Fall wäre es jedoch besser, direkt **Antrag umsetzen** noch mal durchzuführen.

7.2.1 Instanz anhalten

In diesem Abschnitt wird das Anhalten des Ablaufs einer Prozessinstanz besprochen. Es beruht auf den Erkenntnissen in 3.3.3 auf Seite 29. Darin wird gefordert, dass der Ablauf jederzeit unterbrochen werden muss. Dies wurde durch den Status **deferred** ausgedrückt. Damit konnte ausgedrückt werden, dass die weitere Bearbeitung dieses Änderungsantrags verschoben wird (auf die nächste Softwareversion beispielsweise) oder auf andere Entscheidungen/Anträge gewartet werden muss. Ein Antrag in diesem Status soll weiterbearbeitet werden, nur nicht in diesem Augenblick.

Befindet sich ein Antrag in **deferred** ist nicht ersichtlich, wie weit der Antrag schon bearbeitet wurde. Ob er erst erfasst, **submitted** oder schon bereit zur Umsetzung in **ready for implementation** ist, lässt sich daraus nicht erkennen. ADEPT bietet die Möglichkeit, die Bearbeitung eines Prozesses jederzeit zu unterbrechen (falls der Benutzer die benötigten Rechte besitzt). Die Instanz kann manuell fortgesetzt werden

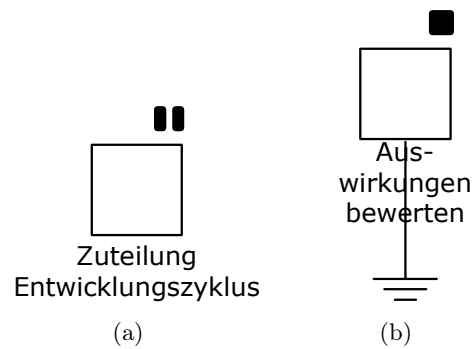


Abbildung 7.1: Graphmarkierung für Angehaltene/Unterbrochene Aktivitäten: (a) zeigt die eine angehaltene Aktivität, (b) eine Aktivität, bei der die Instanz vorzeitig beendet wurde.

oder zu einem definierbaren Zeitpunkt. Damit lassen sich die beiden gewünschten Verhaltensweisen erzielen.

1. Bearbeitung des Änderungsantrags auf einen späteren Meilenstein oder spätere Version des Produkts verschieben. Es ist zeitlich bekannt, wann der neue Meilenstein bearbeitet wird. Instanzen können für den notwendigen Zeitraum suspendiert werden. Wird dann der neue Meilenstein bearbeitet stehen sie wieder zur Bearbeitung bereit.
2. Bearbeitung des Änderungsantrags anhalten und warten. In diesem Fall ist nicht klar, zu welchem Zeitpunkt die Instanz weiterbearbeitet werden kann. Sie muss daher von Hand aufgeweckt werden.

Dieses Verhalten muss, bis auf die Rechte, nicht explizit modelliert werden. Aufgrund der Graphmarkierungen wird auch sofort ersichtlich, wie weit die Instanz schon bearbeitet wurde. Das Projektmanagement kann daher den noch nötigen Aufwand für einen Änderungsantrag besser abschätzen. Dies könnte durch eine spezielle Graphmarkierung noch verstärkt werden. Diese wird in Abbildung 7.1 dargestellt.

7.2.2 Instanz beenden

Hier wird nun versucht, das Verhalten des Status **killed** von 3.3.3 auf Seite 29 abzubilden. Es wird auf Modellierungsebene ein neues Konstrukt eingeführt. Wie dieses in das ADEPT-Basismodell überführt wird, wird nicht weiter betrachtet. Es dient eher dazu, um die Anforderungen aus Benutzersicht konkret darstellen zu können. Ziel ist es, die Bearbeitung einer Instanz zu beenden, und zwar vor dem eigentlichen Ende des vorhergesehenen Ablaufs. Im Kontext des Änderungsmanagements wird die vorzeitige Beendigung des Ablaufs klar. Vor der eigentlichen, arbeitsintensiven Aufgabe, der Aktivität **Antrag umsetzen**, findet eine genaue Analyse des Antrags an mehreren

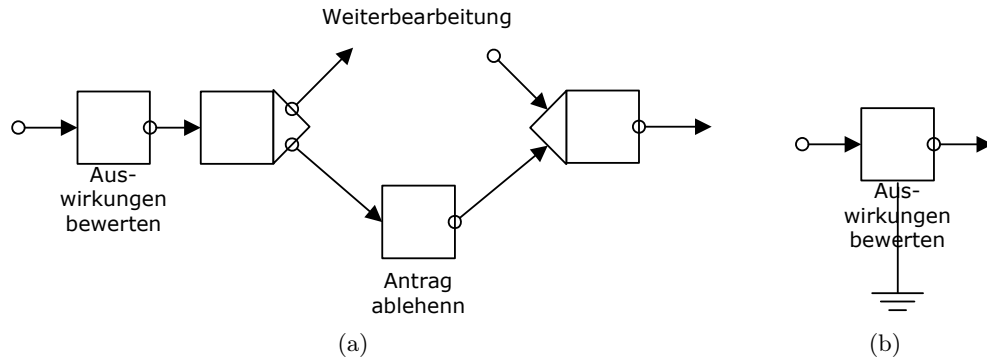


Abbildung 7.2: Instanz beenden: (a) zeigt die Beendigung mittels alternativem Pfad, (b) durch einen neuen Knotentyp

Punkten statt. An diesen Entscheidungspunkten wird entschieden, ob die zeit-/ kosten-/ personalintensive Umsetzung stattfindet. Die Umsetzung wird abgelehnt, wenn diese zu teuer wird oder z. B. eine nicht benötigte Komfortfunktion darstellt. Daher ist es im Sinne des Projekts, die Bearbeitung eines Antrags möglichst früh abubrechen, falls bestimmte Kriterien erfüllt sind. Damit wird unnötiger Aufwand vermieden. Aus der Sicht eines Projekts ist deshalb auch sinnvoll, nach einer nicht erfolgreichen Umsetzung eines Antrags die Prozessinstanz zu beenden. Eine Entscheidung zur Umsetzung kann nicht bedeuten, dass diese Umsetzung unter allen Umständen durchgeführt werden muss. Im Falle einer Kernfunktionalität des Projekts ist dies sinnvoll. In der Mehrzahl der Fälle muss nach einer fehlgeschlagenen Prüfung aber abgebrochen werden können. Mit anderen Worten: auch innerhalb einer Schleife muss eine Instanz vorzeitig beendet werden können.

Dies bedeutet jedoch nicht, dass die Auswirkungen der bisherigen Prozessbearbeitung zurückgenommen werden müssen. Die Entscheidung und bisherige Arbeiten müssen dokumentiert und erhalten bleiben. Wie in 3.3.3 auf Seite 29 gezeigt, werden auf diese Art abgebrochene Instanzen doch wieder fortgesetzt und zwar genau am Abbruchpunkt. Hinreichend starke Änderungen des Projektskontexts machen die Revidierung einer Entscheidung nötig. Ändert sich die Entscheidungsgrundlage, ändert sich auch die Entscheidung.

Bezeichnen wir im Folgenden eine Aktivität, in der die oben beschriebene Entscheidung getroffen werden kann, als Stopp-Aktivität. In Abbildung 7.1 auf der vorherigen Seite ist dafür ein Erdungssymbol vorgesehen. In Abbildung B.16 auf Seite 92 wurde diese Art des Stopps mit einem alternativen Pfad modelliert. Dass der alternative Pfad eigentlich eine Ende der Bearbeitung bedeutet, ist nicht sofort ersichtlich. Auch führen viele definierte Stopp-Aktivitäten zu einer tiefen Schachtelung der alternativen Pfade. Abbildung B.17 auf Seite 92 stellt die gleiche Bedeutung nun mit einer Stopp-Aktivität dar (unter der Annahme, dass diese, wie oben gefordert, auch in Schleifen anwendbar ist). Abbildung 7.2 stellt die Darstellung mittels alternativem Pfad und neuer Stopp-Aktivität nebeneinander dar.

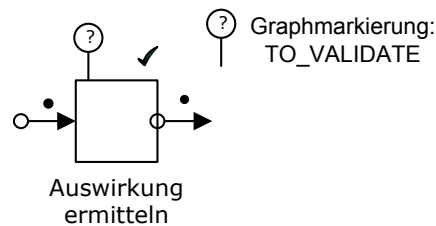


Abbildung 7.3: Graphmarkierung zur Validierung einer Aktivität

Damit erkennt man beim Blick auf das Prozessschema, bei welcher Aktivität ein vorzeitiger Abbruch erfolgen kann. Auch eine Graphmarkierung hilft bei einer Prozessinstanz zu erkennen, welche Teile noch bearbeitet wurden und welche nicht mehr. In der Anwendungsdomäne, laufen an Entscheidungspunkten alle parallel laufenden Stränge zusammen. Im Normalfall müssen also nur die Daten der aktuellen Aktivität soweit wie möglich gesichert werden, damit evtl. bereits durchgeführte Arbeit nicht verloren geht. Existieren doch parallele Aktivitäten, so gilt für diese das selbe: sie werden mit der Stopp-Markierung versehen und ebenfalls sofort beendet. Wenn die Entscheidung getroffen wird, die Bearbeitung abubrechen, sollen keine weiteren Ressourcen verschwendet werden.

7.2.3 Aktivität validieren

Im Folgenden wird definiert, was es bedeutet, eine Aktivität zu validieren. Dies wird benötigt, um festzustellen, welche Aktivitäten nach einer Kontextänderung noch gültig sind. Die Aktivität muss bereits einmal erfolgreich abgeschlossen worden sein. Damit sind sowohl die Ein- als auch die Ausgaben der Aktivität festgelegt. Bei der Validierung einer Aktivität soll festgestellt werden, ob diese Aktivität in dem jetzt gültigen Kontexts des Projekts immer noch so ausgeführt werden würde. Trifft dies zu, wird sie validiert, falls nicht muss sie erneut ausgeführt werden.

Um dem PMS mitzuteilen, dass eine bereits abgeschlossene Aktivität validiert werden soll, wird eine neue Graphmarkierung eingeführt. Diese wird in Abbildung 7.3 dargestellt. Sie ist nur auf Aktivitäten anwendbar, die bereits abgeschlossen sind. Wird die Aktivität validiert, dann kann die Markierung entfernt werden. Wird die Aktivität nicht validiert, kommen die weiter unten beschriebenen Mechanismen zum Einsatz.

Ablauf für eine Aktivität Idealerweise erhält der Mitarbeiter, der Aktivität zum ersten Mal ausgeführt hat, dieses auch zur Validierung wieder. Ist dieser konkrete Mitarbeiter nicht mehr verfügbar (aufgrund aufbauorganisatorischer Änderungen wie seinem Ausscheiden aus dem Projekt) wird die Bearbeiterregel erneut aufgelöst. Durch die spezielle Markierung wird es möglich, die zu validierende Aktivität speziell darzustellen. Bei der Gestaltung der Benutzerschnittstelle kann dabei z. B. eine eigene Arbeitliste o. Ä. dafür vorgesehen werden. Parallel dazu müssen ihm Informationen

bereitgestellt werden, warum er diese Aktivität erneut bearbeiten muss. Diese Informationen sollten ebenfalls im PAIS dargestellt werden, damit zusammengehörende Informationen auch zusammen dargestellt werden. Der Mitarbeiter kann nun entweder die Aktivität validieren oder nicht. Bestätigt er, dass die Ergebnisse der Aktivität noch korrekt sind, endet der Ablauf hier für ihn. Sind sie nicht mehr gültig, muss die Aktivität erneut durchgeführt werden, sprich sie wird zurückgesetzt. Sie wird dann sofort ausführbar. Im nächsten Abschnitt wird erläutert, wie die Validierung von Aktivitäten dazu verwendet werden kann, um die in Abschnitt 7.2 auf Seite 71 beschriebene Szenarien umzusetzen.

Ablauf für mehrere Aktivitäten Im oberen Abschnitt wurde definiert, wie eine Aktivität validiert wird. Im Folgenden wird nun dargestellt, wie sich das PMS verhält, wenn eine Aktivität mit `TO_VALIDATE` markiert ist. Grundsätzlich können nur Aktivitäten mit `TO_VALIDATE` markiert sein, die bereits beendet worden sind. Sie sind daher `COMPLETED`. Das PMS muss dem durch die Ausführungshistorie bereits definiertem Ablauf, folgen. Trifft es auf eine zu validierende Aktivität, geht es wie oben beschreiben vor. Die Aktivität kann entweder validiert werden oder nicht.

Wird sie validiert, wird mit allen weiteren in der Ausführungshistorie vorhandenen, mit `TO_VALIDATE` markierten Aktivitäten ebenso verfahren. Ist keine weitere Aktivität mehr vorhanden, ist das Ende erreicht. Das PMS kann in gewohnter Weise weiterverfahren. War die Instanz beendet oder gestoppt, gilt dies nach der Validierung immer noch. Sind danach noch Aktivitäten zur Ausführung selektiert, wird ganz normal weiterverfahren.

Wird die Aktivität nicht validiert, wird sie aktivierbar (da der Validierer nicht notwendigerweise auch derjenige ist, der nach der Bearbeiterregel dafür vorgesehen ist). Alle anderen Aktivitäten danach müssen also nicht mehr validiert werden. Im neuen Kontext besitzen sie keine Gültigkeit mehr. Aus Benutzersicht beginnt die Bearbeitung dieser Instanz ab dieser Aktivität erneut.

7.3 Umsetzung der Szenarien

Es sind nun alle Mittel vorhanden, um die oben beschriebenen Szenarien umsetzen zu können. Die drei erkannten Fälle, die sich auf die Wiederaufnahme beendeter Instanzen beziehen, sind in Abbildung 7.4 auf der nächsten Seite dargestellt.

Der erste Fall umfasst das erste Szenario: eine Instanz wurde über die spezielle Stopp-Aktivität beendet. Bei dieser Aktivität soll die Bearbeitung auch wieder fortgesetzt werden.

Im Fall 2.1 und 2.2 kann dagegen nicht bei der Stopp-Aktivität fortgesetzt werden. Es muss erst festgestellt werden, welche der Aktivitäten im neuen Kontext noch gültig sind. Im Fall 2.1 wirkt sich die Kontextänderung auf ein Datum im Datenkontext der Instanz aus. Oder es kann genau festgestellt werden, dass ein bestimmtes Datum davon betroffen ist. Es muss daher ab dem letzten Schreibzugriff auf dieses Datum überprüft werden. Ist das nicht möglich (Fall 2.2), müssen alle Aktivitäten vom Start

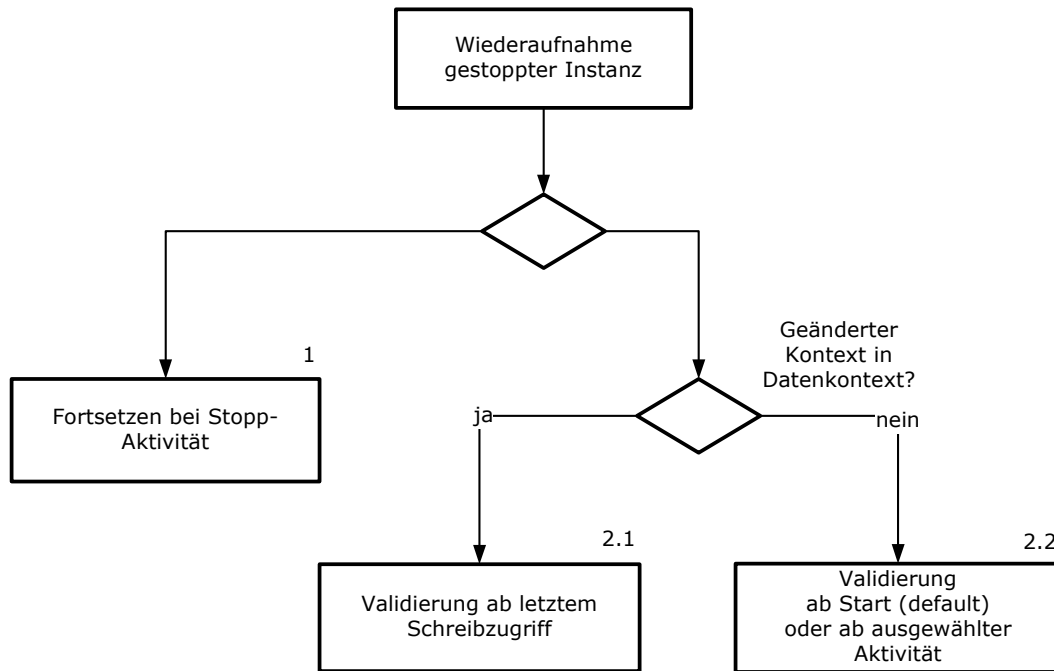


Abbildung 7.4: Wiederaufnahme gestoppter Prozessinstanzen. Es können drei Fälle unterschieden werden.

bis zur letzten ausgeführten Aktivität validiert werden. Alternativ dazu ist es möglich die erste zu validierende Aktivität manuell zu markieren (z. B. aus einer Liste mit Aktivitäten).

Im ersten Szenario müssen somit keine Aktivitäten validiert werden. Im Fall 2.1 und 2.2 jedoch schon. Durch obige Unterscheidung wird auch festgelegt, welche Aktivitäten der Ausführungshistorie validiert werden müssen: im Fall 2.2 vom Ende der Ausführungshistorie alle bis zum Anfang (außer es wurde eine manuell festgelegt, dann ist diese die letzte). Im Fall 2.1 alle vom Ende der Ausführungshistorie bis zu der Aktivität, welche das Datum als erstes schreibt. Es ist die erste, da die Ausführungshistorie von hinten her, markiert wird.

Der erste Fall wird in Abbildung 7.5 auf der nächsten Seite gezeigt. In diesem Fall müssen keine Aktivitäten validiert werden. Die Aktivität aus der die Aktivität beendet wurde wird wieder aktiviert (und mit den bereits vorhandenen Daten versorgt).

Welche Aktivitäten validiert werden müssen, zeigt Abbildung 7.6 auf der nächsten Seite.

Ein konkretes Beispiel wird in Abbildung 7.7 auf Seite 80 erläutert. Der Ausgangspunkt ist 7.7(a) auf Seite 80. Alle Aktivitäten bis zur letzten wurden zur Validierung markiert. Von der Startaktivität aus werden jetzt alle Aktivitäten ihren Bearbeitern zur Validierung vorgelegt. Werden alle Aktivitäten validiert, ergibt sich das Bild in 7.7(b). Dies bedeutet, dass alle Aktivitäten noch gültig sind. Die Prozessinstanz wird

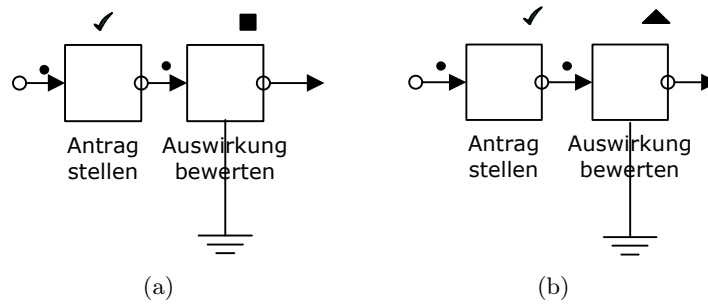


Abbildung 7.5: Gestoppte Instanz fortsetzen, Fall 1: (a) zeigt die Ausgangssituation. Der Prozess wurde in **Auswirkung bewerten** gestoppt. In (b) wird die Instanz an dieser Aktivität fortgesetzt

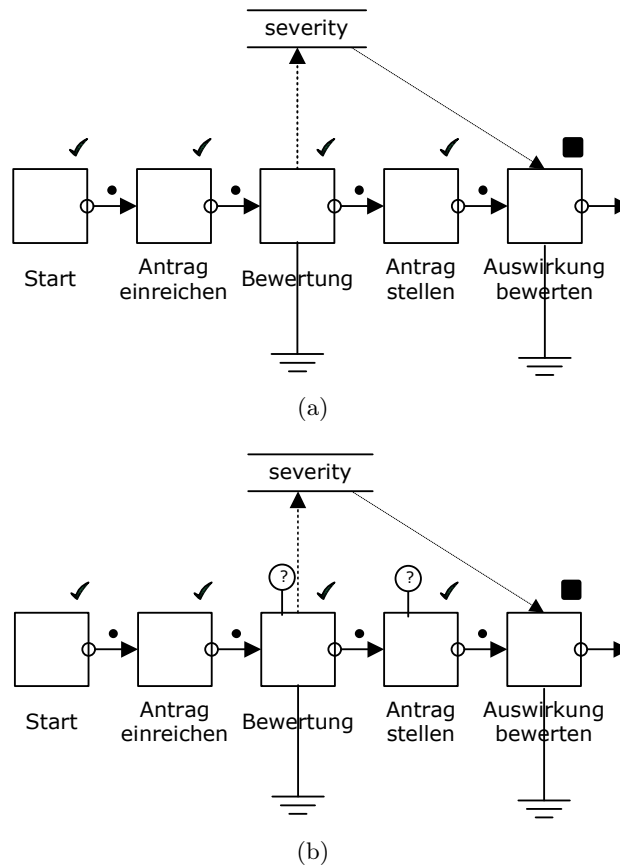


Abbildung 7.6: Validierung einer Instanz, Fall 2.1: (a) zeigt die Ausgangssituation. In (b) wurden die zu validierenden Aktivitäten markiert.

bei der Aktivität **Auswirkung bewerten** weiter bearbeitet. Wird eine der Aktivitäten nicht validiert, werden alle Aktivitäten danach zurückgesetzt (Rollback). Dieser Fall ist in Abbildung 7.7(c) auf der nächsten Seite dargestellt. Die Aktivität, die nicht validiert werden kann, ist die erste die erneut bearbeitet werden muss. Hier konnte die Aktivität **Antrag einreichen** validiert werden. Nur die **Bewertung** ist im neuen Kontext nicht mehr gültig und muss somit erneut durchgeführt werden. Alle Aktivitäten danach müssen nicht mehr validiert werden, sondern werden zurückgesetzt.

Aus diesen Szenarien und Fällen lassen sich weitere ableiten. Auch bei gerade laufenden Instanzen muss überprüft werden können, ob die bisherigen Annahmen noch gültig sind. Darin muss noch zur Laufzeit überprüft werden ob die bisherigen Aktivitäten noch gültig sind. Dies lässt sich jedoch auf den zweiten Fall abbilden. Die aktuell gerade zu bearbeitenden Aktivitäten werden suspendiert. Dann werden die Vorgängeraktivitäten wie oben beschrieben markiert und validiert. Aus Benutzersicht handelt es sich um eine Art Schleife mit variabel zur Laufzeit definierter Anfangsaktivität. Dies ist die erste Aktivität, die nicht validiert wird. Die suspendierte Aktivität bildet das Schleifenende.

Damit ist das dritte Szenario noch nicht abgedeckt. Dieses beschreibt einen anderen Fall. Obwohl auch hier Teile neu bearbeitet werden müssen, sollen hier keine Teile zurückgesetzt werden. Vielmehr soll hier nur der Anfang einer Schleife variabel zur Laufzeit ausgewählt werden können. In Tabelle 3.5 auf Seite 36 sind unterschiedliche Fälle für Prüfzyklen aufgelistet. In $\ddot{A}M(G)$ (vgl. Abbildung B.3 auf Seite 91) und beim $\ddot{A}M(PVCS)$ (vgl. Abschnitt 3.3.3 auf Seite 29) ergab sich die Notwendigkeit, den Schleifenrücksprung flexibel bestimmen zu können. Der Prozessexperte bezeichnete dies als „wahlfreien Rücksprung“. Dies soll hier nicht weiter vertieft werden.

7.4 Zusammenfassung

Die zu validierenden Aktivitäten müssen nicht von den in den Bearbeiterregeln vorgesehenen Mitarbeitern bearbeitet werden. Auch eine andere Bearbeiterzuordnung würde, unter Berücksichtigung der Ausführungsrechte, Sinn ergeben. Man kann alle zu validierende Aktivitäten einem Mitarbeiter zuordnen. Dieser validiert dann eine Aktivität nach der anderen. Erst wenn er eine Aktivität nicht validiert, wird diese wieder dem eigentlich dafür vorgesehen Mitarbeiter vorgelegt. Dieser bearbeitet dann die Aktivität. Damit kann das weitere Vorgehen von einer Person bestimmt werden.

Die in diesem Kapitel beschriebenen Änderungen führen zu einer weiteren Stufe der Flexibilität. Sie erfordern keine strukturelle Änderung des Prozesses wie durch ad-hoc Änderungen oder die Schemaevolution. Sie ermöglichen es, flexibel auf Änderungen des Kontexts einzugehen. Es lässt sich vom System unterstützt überprüfen, welche Aktivitäten trotz einer Kontextänderung noch Gültigkeit besitzen. Auch einmal getroffene Entscheidungen lassen sich einfach zurücknehmen. Diese Möglichkeiten sind in Reaktion auf reale Bedürfnisse der Anwender hin angedacht worden.

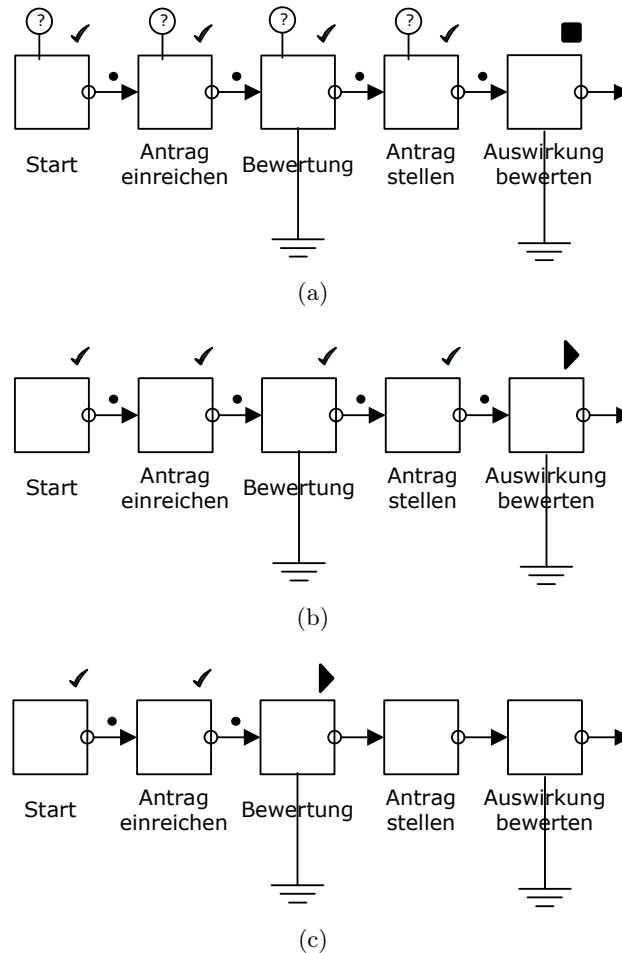


Abbildung 7.7: Validierung einer Instanz, Fall 2.2: Alle Aktivitäten vom Start bis zur Endaktivität müssen validiert werden. Dies wird in (a) dargestellt. In (b) wurden alle Aktivitäten validiert, die Bearbeitung wird bei der Endaktivität fortgesetzt. In (c) wurde die **Bewertung** nicht validiert. Daher wird diese Aktivität ausgeführt und alle nachfolgenden zurückgenommen.

Kapitel 8

Zusammenfassung und Ausblick

Dieses Kapitel bildet den Abschluss dieser Arbeit. In Abschnitt 8.1 werden die Ergebnisse dieser Arbeit noch einmal zusammengefasst. Abschnitt 8.2 beschreibt offene gebliebene Fragen und Ansatzpunkte für weitere Arbeiten.

8.1 Zusammenfassung

Um die Herausforderungen des Anwendungsgebiets aufzuzeigen, werden Prozesse zur Entwicklung eingebetteter Steuergeräte untersucht. Die Ursachen und Auswirkungen von Änderungen werden in einer post mortem Analyse betrachtet. Besonders Prozesse des Problem- und Änderungsmanagements können dabei detailliert untersucht werden. Aus dem Vergleich von acht unterschiedlichen Projekten konnte gefolgert werden, worin sich die Prozesse unterscheiden. Die Aktivitäten sind größtenteils gleich. Nur ihre Anordnung (Kontrollfluss) und die Bearbeiterzuordnung sowie das zugrunde liegende Organisationsmodell unterscheiden sich. Änderungen am Prozessschema und an einzelnen Prozessinstanzen sind die Regel.

Die gefundenen Änderungen werden in Kategorien eingeteilt. Es können dabei zwei Dimensionen unterschieden werden: die Ursache und die Planbarkeit der Änderung. Als Ursachen werden Änderungen der Aufbauorganisation, technologische, kulturelle, strategische und politisch-rechtliche identifiziert. Projektspezifische Anpassungen verursachen einen Großteil der Änderungen in der Praxis und werden daher als neue Kategorie mit aufgenommen. Diese Ursachen können dann noch hinsichtlich ihrer Planbarkeit unterschieden werden. Planbare Änderungen können vor Projektbeginn oder während des Projekts auftreten. Damit werden vor allem auch zyklische Änderungen zur Projektlaufzeit, die in jedem Projekt erneut auftreten, erfasst. Daneben werden noch nicht planbare Änderungen unterschieden.

Aus den Fallbeispielen und der Kategorisierung lassen sich Anforderungen an ein Informationssystem zur Unterstützung von Entwicklungsprojekten ableiten. Änderungen müssen vor Projektbeginn und während des Projektverlaufs unterstützt werden. Dabei sollte das System Änderungen sowohl des Schemas, als auch einzelner Instanzen zulassen. Auch Änderungen am Organisationsmodell müssen gewährleistet werden. Dabei dürfen Änderungen nicht zu einem inkonsistenten Zustand führen.

Danach wird skizziert, wie mit den Mitteln von ADEPT, einem Prozessmanagementsystem, flexible prozessbasierte Informationssysteme realisiert werden können.

Mit den Mitteln der Schemaevolution sowie der Möglichkeit, dynamisch, zur Laufzeit vom vorgesehenen Ablauf abzuweichen, werden einzelne Beispiele näher dargestellt.

Veränderungen des Kontexts führen nicht nur zu strukturellen Änderungen des Prozesses. Im Laufe von Prozessinstanzen getroffene Annahmen und Entscheidungen müssen überprüft und revidiert werden. Aus Benutzersicht müssen somit bereits abgeschlossene Aktivitäten validiert werden. Kann in der Ausführungshistorie eine Aktivität nicht validiert werden, müssen alle anderen Aktivitäten danach zurückgesetzt werden. Ab diesem Punkt muss die Prozessinstanz neu bearbeitet werden. Daneben wurden noch weiter, aus Benutzersicht notwendige Operationen dargestellt: eine frühzeitig beendete Prozessinstanz an genau diesem Punkt wieder fortzusetzen. Auch ist es notwendig den Startpunkt einer Schleife dynamisch zur Laufzeit bestimmen zu können.

8.2 Ausblick

In dieser Arbeit wurde versucht, einen Teil der Fallbeispiele exemplarisch an den Fähigkeiten eines Prozessmanagement Systems zu spiegeln. Es konnten jedoch nur einige, wenige Beispiele konzeptionell untersucht werden. Um den empirischen Ansatz dieser Arbeit zu vervollständigen, wäre es notwendig, in Zusammenarbeit mit den späteren Benutzern, ein prozessbasiertes Informationssystem zu realisieren. Die Erstellung eines einsetzbaren Informationssystems, das allen Anforderungen, z. B. des Problem- und Änderungsmanagements, gerecht wird, liegt außerhalb der Möglichkeiten dieser Arbeit. Aus der möglichen Beobachtung seiner Benutzung lassen sich die Erkenntnisse dieser Arbeit widerlegen, erweitern oder bestätigen.

Wird ein PAIS mit der benötigten Flexibilität realisiert, stellt dies wiederum den Ausgangspunkt weiterer Beobachtungen dar. Diese neuen Möglichkeiten werden die Arbeitspraxis verändern, da sie einerseits nicht mehr so einschränkend sind und andererseits neue Arbeitsweisen überhaupt erst ermöglichen.

Neben der soeben beschriebenen Vervollständigung des empirischen Ansatzes dieser Arbeit, ergibt sich noch ein weiterer Ansatzpunkt für weitere Untersuchungen: die systematische Aufzeichnung, Verwaltung und Wiederverwendung von Prozessänderungen, deren Ursachen und Zusammenhänge. Ein erster Schritt dafür wurde mit der CCBR Erweiterung von ADEPT in Abschnitt 7.1 auf Seite 69 vorgestellt. Dies stellt jedoch nur den Ausschnitt eines Prozessschemas dar. Quer- und Wechselbeziehungen mit weiteren Prozessen und insbesondere mit Prozessen, die nicht in PAIS umgesetzt sind, werden dadurch nicht erfasst. Diese stellen aber eine weitere Quelle von Wissen dar. Wird es nicht erfasst, ist es nur in den Köpfen weniger Personen vorhanden und kann einfach verloren gehen. Um dieses Wissen nutzen zu können, muss es erfasst werden.

Literaturverzeichnis

- [AAD⁺04] ACKER, Hilmar ; ATKINSON, Colin ; DADAM, Peter ; RINDERLE, Stefanie ; REICHERT, Manfred: Aspekte der komponentenorientierten Entwicklung adaptiver prozessorientierter Unternehmenssoftware. In: *AKA*, 2004, S. 7–24
- [Aal98] AALST, W.M.P. van der: The Application of Petri Nets to Workflow Management. In: *The Journal of Circuits, Systems and Computers* 8 (1998), Nr. 1, S. 21–66
- [AJ00] AALST, W.M.P. van der ; JABLONSKI, S.: Dealing with Workflow Change: Identification of Issues and Solutions. In: *International Journal of Computer Systems, Science, and Engineering* 15 (2000), Nr. 5, S. 267–276
- [AS98] ALEXANDER, D. ; SCHWICKERT, A. C.: *Geschäftsprozeß- und Funktionsorientierung - Ein Vergleich (Teil1)*. Lehrstuhl für Allg. BWL und Wirtschaftsinformatik, 1998
- [Bun] BUNDESREPUBLIK DEUTSCHLAND: *V-Modell® XT Teil 6: V-Modell-Referenz Aktivitäten*. <http://www.v-modell-xt.de/>. PDF-Dokument. – Online-Ressource, Abruf: 10.07.2005. – Version 1.1.0
- [DAH05] DUMAS, M. ; AALST, W.M.P. van der ; HOFSTEDE, A.H.M. ter: *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley & Sons, 2005
- [DRR04] DADAM, P. ; REICHERT, M. ; RINDERLE, S.: Von funktionsorientierten zu prozessorientierten Informationssystemen - Herausforderungen und Lösungsansätze -. In: *Tagungsband der 17. Deutschen Oracle-Anwender-Konferenz* (2004). – Keynote
- [HH05] HOFSTEDE, Geert ; HOFSTEDE, Gert J.: *Cultures and Organizations - Software of the Mind*. 2nd Edition. McGraw-Hill, 2005. – ISBN 0-07-143959-5
- [JR04] JAUFMAN, O. ; REICHERT, M.: *Eliciting Suitable Process Model for Software Development*. 2004. – interner Bericht
- [JSS⁺04] JAUFMAN, O. ; STUPPERICH, M. ; SCHNEIDER, K. ; DOLD, A. ; KLEINER, N.: *A learning Support for the Definition of Process Flexibility*. 2004. – QUATIC 2004)
- [Kle04] KLEINER, N.: *Verbesserungsstrategien für den Workflow-Designprozess*, Universität Ulm, Dissertation, Mai 2004
- [MA01] MARCUS, Aaron ; ASSOCIATES, Inc.: *Cultural Dimensions and Global Web Design: What? So What? Now What?* Aaron Marcus and Associates, Inc., 2001
- [MO04] MÜNCH, J. ; OCAMPO, A.: *Software Process Variability: Concept and Approaches*. Fraunhofer IESE, 2004
- [MS96] MAURER, G. ; SCHWICKERT, A. C.: *Von der Prozeßorientierung zum Workflow Management Teil1*. Lehrstuhl für Allg. BWL und Wirtschaftsinformatik, 1996
- [Mue04] MUEHLEN, M. Zur: *Workflow-based Process Controlling: Foundation, Design and Application of workflow-driven Process Information Systems*. Logos, Berlin, 2004
- [Rei00] REICHERT, M.: *Dynamische Ablaufänderungen in Workflow-Systemen*, Universität Ulm, Dissertation, Mai 2000
- [RRD04a] RINDERLE, Stefanie ; REICHERT, Manfred ; DADAM, Peter: Disjoint and Overlapping Process Changes: Challenges, Solutions, Applications. In: *CoopIS/DOA/ODBASE (1)*, 2004, S. 101–120

Literaturverzeichnis

- [RRD04b] RINDERLE, Stefanie ; REICHERT, Manfred ; DADAM, Peter: On Dealing with Structural Conflicts between Process Type and Instance Changes. In: *Business Process Management*, 274–289
- [RWRW05] RINDERLE, Stefanie ; WEBER, Barbara ; REICHERT, Manfred ; WILD, Werner: Integrating Process Learning and Process Evolution - A Semantics Based Approach. In: *Business Process Management*, 252–267
- [SK96] SCHWICKERT, A. C. ; KIM, A.: *Der Geschäftsprozeß als formaler Prozeß - Definition, Eigenschaften, Arten*. Lehrstuhl für Allg. BWL und Wirtschaftsinformatik, 1996
- [Spa01] SPARR, S.: *Ausführungs-, Zugriffs- und Änderungsrechte in adaptiven Prozess-Management-Systemen*. Universität Ulm/Deutschland, Fakultät für Informatik, Diplomarbeit, 2001
- [WC02] WIEDEMTH-CATRINESCU, U.: *Evolution von Organisationsmodellen in Workflow-Management-Systemen*. Universität Ulm/Deutschland, Fakultät für Informatik, Diplomarbeit, 2002
- [Wöh00] WÖHE, Günter: *Einführung in die Allgemeine Betriebswirtschaftslehre*. München : Verlag Franz Vahlen GmbH, 2000
- [WM00] WILD, Peter J. ; MACREDIE, Robert D.: *On Change and Tasks*. August 2000
- [WRWR05] WEBER, Barbara ; RINDERLE, Stefanie ; WILD, Werner ; REICHERT, Manfred: CCB-Driven Business Process Evolution. In: *ICCB*, 610–624

Anhang A

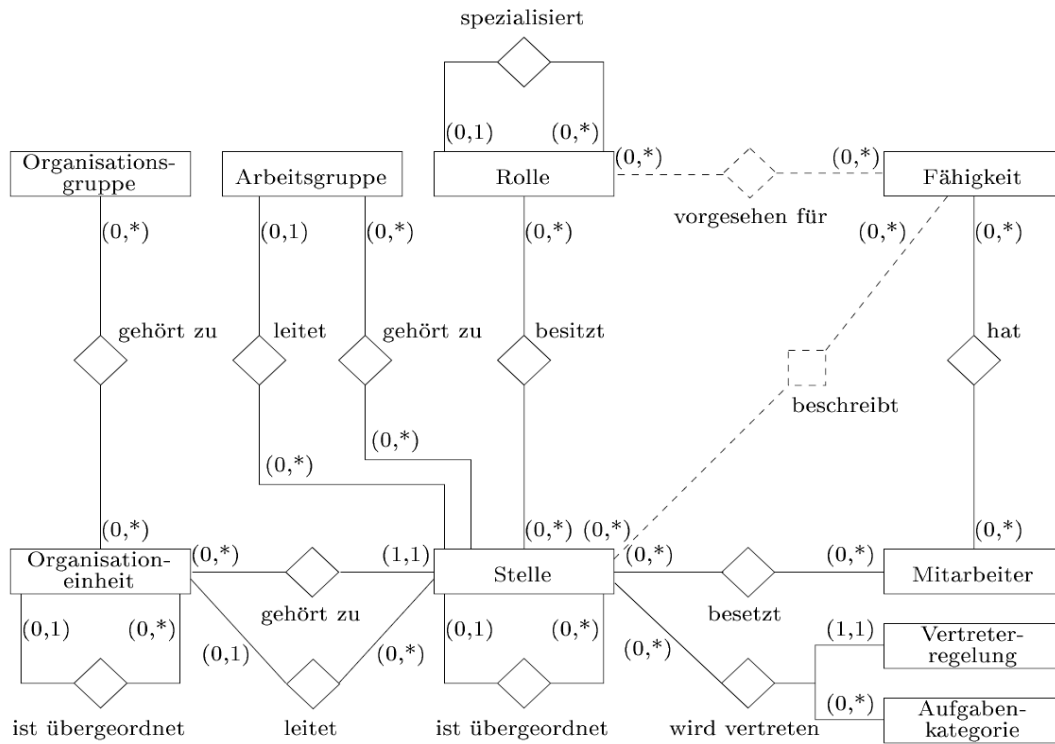


Abbildung A.1: Das ADEPT Organisations-Metamodell

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den 01. Februar 2006
Florian Breier

Anhang B

Prozessgraphen

B.1 Vertraulich

Der Anhang B ist vertraulich. Nur die interne Verwendung in der Abteilung Datenbanken und Informationssysteme der Universität Ulm sowie der Abteilung REI/SP von DaimlerChrysler ist erlaubt. Das Kopieren, Vervielfältigen oder eine andere Nutzung der hier dargestellten Inhalte ist nicht erlaubt.

Anmerkung zur Prozessdarstellung und Auswertung

Notation Die verwendete Notation wird in Abbildung B.1 dargestellt. Auf die Verwendung eines expliziten Startknotens wurde verzichtet. Im Falle des Änderungsmanagements ist dies immer **Antrag erfassen**. Das Organisationsmodell und die Mitarbeiterzuordnung sind, soweit vorhanden, auch dargestellt. Jede Zeile des Diagramms stellt eine bestimmte Rolle oder andere Entität des Organisationsmodells dar. Eine Aktivität wird einem Mitarbeiter zugewiesen, indem sie in der entsprechenden Zeile dargestellt wird. In den Diagrammen können mehrere Endknoten dargestellt sein. Dies soll verdeutlichen, aus welchen Aktivitäten heraus der Prozess (vorzeitig) beendet werden kann. Dadurch kann auch auf Anhalt erkannt werden, wer für das Prozessende verantwortlich war: derjenige, in dessen Reihe sich der Endknoten befindet.

Anpassungen Die Prozessgraphen wurden durch den Autor angepasst. Sie entsprechen nicht mehr den gefundenen Originalen. Prozessgraphen waren mit unterschiedlichen Notationen dargestellt. Die Notation wurde vereinheitlicht. Dabei wurde versucht, so nah wie möglich am Original zu bleiben, um die durch den Prozess ausgedrückte Semantik nicht zu verfälschen. Verwendete Begriffe wurden auch angepasst. Wurde in den Prozessen von Fehlern gesprochen, wird dies in dieser Arbeit auf Änderungen abgebildet. Der Fehlerverantwortliche wird so zum Änderungsverantwortlichen und das Failure Control Board zum Change Control Board. Der Begriff der Änderung wurde schon von einigen Projekten verwendet. Durch diese Anpassung wird eine einheitliche Begriffsbildung über die Projektgrenzen hinweg etabliert. Auch entfällt die Diskussion, was nun ein Fehler und was eine Änderung ist. Aktivitäten mit unterschiedlichen Bezeichnungen aber gleichem Inhalt wurden auch vereinheitlicht. Es wird so immer von **Antrag erfassen** geredet und andere Begriffe wie **Antrag erstellen** oder **Antrag einreichen** so weit wie möglich vermieden. Die Anpassungen erfolgten in Zusammenarbeit mit den Prozessexperten. Aussagen werden dadurch nicht verfälscht.

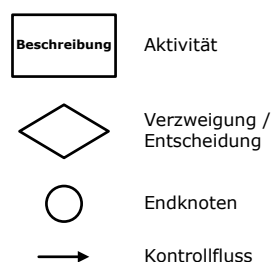


Abbildung B.1: Notation um Änderungsmanagement Prozesse darzustellen

Abbildung B.2: ÄM(M)

Abbildung B.3: ÄM(G)

Abbildung B.4: ÄM(StgV)

Abbildung B.5: ÄM(SP)

Abbildung B.6: ÄM(EL)

(a) (b)

Abbildung B.7: Änderungsmanagement ÄM(Gv1) Teil 1 (a) zeigt den Ausgangspunkt der Entwicklung, (b) die zweite Prozessvariante

(c) (d)

Abbildung B.8: (c) die dritte und (d) die letzte Prozessvariante

(a) (b)

Abbildung B.9: Änderungsmanagement ÄM(Gv2) Teil 1 (a) zeigt den Ausgangspunkt der Entwicklung, (b) die zweite Prozessvariante

(c) (d)

Abbildung B.10: Änderungsmanagement ÄM(Gv2) Teil 2 (c) die dritte und (d) die letzte Prozessvariante

(a) (b)

(c) (d)

Abbildung B.11: Entwicklung des Änderungsmanagements in PVCS: (a) zeigt den Ausgangspunkt der Entwicklung, (b) die erste eingesetzte Variante, (c) zeigt die aktuell verwendete Variante und (d) die geplante Prozessänderung

(a) (b)

Abbildung B.12: Prozess, um Sonderfunktionen zu realisieren, (a) zeigt den Prozess vor und (b) nach einer Prozessverbesserung

Abbildung B.13: Freigabeprozess Version 1

Abbildung B.14: Freigabeprozess Version 2

Abbildung B.15: Freigabeprozess Version 3

Abbildung B.16: Umsetzung von $\ddot{A}M(EL)$ in ADEPT

Abbildung B.17: Umsetzung von $\ddot{A}M(EL)$ in ADEPT mit Stopp-Aktivität