



FAKULTÄT FÜR **!INFORMATIK**

# Entwurf und Entwicklung eines webbasierten Prozessmanagement- Systems am Fallbeispiel der IT-Abteilung eines mittelständischen Unternehmens

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Medizinische Informatik**

eingereicht von

**Andreas Reitter**

Matrikelnummer 0405027

an der  
Fakultät für Informatik der Technischen Universität Wien

Betreuung:  
*Betreuer: Thomas Grechenig*  
*Mitwirkung: Mario Bernhart*

Wien, 16. 09. 2009

\_\_\_\_\_  
(Unterschrift Verfasser/in)

\_\_\_\_\_  
(Unterschrift Betreuer/in)



**MASTERARBEIT**

**Entwurf und Entwicklung eines  
webbasierten Prozessmanagement-  
Systems am Fallbeispiel der  
IT-Abteilung eines mittelständischen  
Unternehmens**

Ausgeführt am Institut für  
Rechnergestützte Automation  
Forschungsgruppe Industrial Software (INSO)  
der Technischen Universität Wien

unter Anleitung von  
**Thomas Grechenig**

Assistenzbetreuung  
**Mario Bernhart**

durch

**Andreas Reitter**

Wagnerstraße 29/23/1  
2371 Hinterbrühl

Wien, am 16. September 2009

.....

# Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, am 16. September 2009

.....

(Andreas Reitter)

# Danksagung

Zunächst danke ich meinen Eltern dafür, dass sie mir das Studium ermöglicht haben, mich fortwährend im Laufe meines Studiums unterstützt haben und dass sie mir stets mit Rat und Tat zu Seite gestanden sind.

Meiner Freundin danke ich für das Korrekturlesen und dass sie es geschafft hat, mich immer wieder zu motivieren diese Arbeit zu schreiben. Genauso möchte ich mich bei meiner Tante Gertrude bedanken, welche mir beim Verbessern der Arbeit sehr geholfen hat.

# Kurzfassung

Diese Arbeit befasst sich mit dem Problem, dass Arbeitsabläufe in Softwareerzeugenden Organisationen oft immer komplexer werden. Mangels vollständiger Dokumentation kann es vor allem bei neu aufgenommenen Mitarbeitern vorkommen, dass einzelne oftmals wichtige Arbeitsschritte nicht durchgeführt werden. Dies betrifft in den meisten Fällen Arbeitsschritte, deren Nichtdurchführung für den Prozess/das Projekt keinerlei unmittelbare Auswirkung haben, rechtlich aber verbindlich durchzuführen sind. Ein Beispiel hierfür ist die Dokumentation von durchgeführten Tätigkeiten auf Kundenservern.

Die Einführung eines Workflow-Management-Systems kann diesem Problem entgegenwirken. Dieser Masterarbeit zeigt auf, wie Prozesse erfasst, analysiert, dokumentiert und in ein dynamisches System zur Durchführung eingebunden werden können. Besonderer Wert wird darauf gelegt, wie die Anwender bei ihrer täglichen Arbeit unterstützt werden können, so dass sie ihre Arbeitsaufgaben schneller bzw. mit besserer Qualität durchführen können. Anhand des Fallbeispiels einer IT-Abteilung eines mittelständischen Unternehmens wird ein webbasiertes Workflow-Management-System, welches diese Aufgaben bewältigen soll, entworfen und entwickelt.

Im Zuge der Einführung des Workflow-Management-Systems in der Organisation konnten Zuständigkeiten klar definiert, Grobabläufe für Prozesse vorgegeben und die Erstellung einer ausführlichen Dokumentation zu den Prozessen veranlasst werden. Das implementierte Workflow-Management-System wurde in ein bestehendes Zeitdokumentationssystem integriert, um die Akzeptanz der Nutzer möglichst hoch zu halten. Mit Hilfe der von Workflow-Management-System und Zeitdokumentationssystem erhobenen Zeitmarken können nun Auswertungen über die Prozessleistung generiert werden, welche der Verbesserung der Arbeitsabläufe dienen und zur Planung von zukünftigen Projekten beitragen. Als Abschluss der Arbeit wurde ein Fragebogen entworfen, mit Hilfe dessen eine Evaluierung der Arbeit durchgeführt wurde.

# Abstract

This diploma thesis deals with the problem that workflows of software developing organisations are getting more and more complicated. Because of the absence of exhaustive documentation especially new employees are in danger of forgetting necessary work steps. Usually work steps which are legally important and do not have an immediate effect on the process or the project are affected. The documentation of activities done on a server belonging to a customer is an example therefore.

This lapse may be prevented by using a workflow-management-system. This diploma thesis shows how to create documentation, gather and analyze processes and how to integrate these processes into a dynamic system for the workflow execution. There is a particular focus on supporting users at their daily work in a way that makes them do their work faster or with a higher quality respectively. Based the case study of the IT-Department of a medium-sized organisation a workflow-management-system which should fulfill these tasks will be designed and engineered.

In the course of the implementation of the workflow-management-system at the organisation responsibilities and rough process structures were defined and the creation of an exhaustive documentation of the processes was arranged. In order to maintain the user satisfaction the workflow-management-system was integrated into an existing system for documenting working hours. Using the timestamps created by both systems it is now possible to create charts for the process performance which may be used to optimize the workflows and to allow resource planning's for future projects respectively. Finally an evaluation based on a self designed questionnaire took place.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Problemstellung . . . . .	1
1.2	Zielsetzung . . . . .	2
1.3	Aufbau der Arbeit . . . . .	3
<b>2</b>	<b>Grundlagen des Prozessmanagements</b>	<b>5</b>
2.1	Geschäftsprozessmanagement . . . . .	5
2.1.1	Prozess- und Workflowmanagement . . . . .	6
2.1.2	Workflow-Management-Systeme . . . . .	7
2.1.3	Rollen im BPM . . . . .	8
2.1.4	Gründe für Prozessmanagement . . . . .	11
2.1.5	Prozessorientierte Sichtweise einer Organisation . . . . .	12
2.1.5.1	Organisationsformen . . . . .	13
2.1.5.2	Organisationsstrukturen . . . . .	14
2.1.6	Prozesslebenszyklus . . . . .	16
2.1.7	Visualisierung von Prozessen . . . . .	18
2.1.7.1	Anforderungen an die Visualisierung . . . . .	18
2.1.7.2	Visualisierungsmethoden . . . . .	19
2.1.8	Optimierung von Prozessen (Die Vier Schritte Methode) . . . . .	22
2.1.8.1	Prozessidentifikation . . . . .	23
2.1.8.2	Ist-Analyse . . . . .	24
2.1.8.3	Soll-Konzeption . . . . .	27
2.1.8.4	Umsetzung der Verbesserungspotenziale . . . . .	28
2.2	IT Infrastructure Library . . . . .	30
2.2.1	IT Service Management . . . . .	31
2.2.2	Servicestrategie . . . . .	32
2.2.3	Serviceentwurf . . . . .	32
2.2.4	Serviceüberführung . . . . .	33
2.2.5	Servicebetrieb . . . . .	34
2.2.6	Kontinuierliche Serviceverbesserung . . . . .	34

<b>3</b>	<b>Beschreibung der Organisation des Fallbeispiels</b>	<b>35</b>
3.1	Szenario allgemein . . . . .	35
3.2	Produkte . . . . .	35
3.3	Struktur . . . . .	36
3.3.1	Projektteam . . . . .	36
3.3.2	Technikteam . . . . .	36
3.4	Eingesetzte Software . . . . .	37
<b>4</b>	<b>Optimierungsbedürfnisse der Organisation - Problemstellung</b>	<b>38</b>
4.1	Einschulung neuer Mitarbeiter . . . . .	38
4.2	Verteilte Organisation . . . . .	39
4.3	Projektstatus . . . . .	39
4.4	Unklare Richtlinien . . . . .	39
4.5	Nicht einheitliche Dokumentenablage . . . . .	39
4.6	Ansprechpartner für die Kunden . . . . .	40
4.7	Rechtliche Verbindlichkeiten . . . . .	40
4.8	Standardisiertes Arbeiten . . . . .	40
<b>5</b>	<b>Methodik des Erfassens der Prozesse und des Entwurfs des WMS</b>	<b>42</b>
5.1	Erfassen der Prozesse/Workflows . . . . .	42
5.1.1	Prozessteam Meeting 1 . . . . .	42
5.1.2	Prozessteam Meeting 2 . . . . .	43
5.1.3	Prozessteam Meeting 3 . . . . .	43
5.1.4	Prozessteam Meeting 4 . . . . .	43
5.2	Entwurf des Workflow-Management-Systems . . . . .	45
5.2.1	Prototyp . . . . .	45
5.2.2	Methodik der Implementierung . . . . .	45
<b>6</b>	<b>Erfassen der Workflows</b>	<b>46</b>
6.1	Workflow „Kundenwunsch“ . . . . .	46
6.1.1	Grund der Aufnahme in das WMS . . . . .	46
6.1.2	IST-Zustand . . . . .	47
6.1.3	Verbesserungspotenziale . . . . .	47
6.1.4	SOLL-Zustand . . . . .	47
6.1.4.1	Zusätzliche Aufwände vom IST- zum SOLL-Zustand	50
6.1.4.2	Verbesserungen vom IST- zum SOLL-Zustand .	50
6.1.5	Messgrößen . . . . .	50
6.2	Workflow „Produktaktualisierung“ . . . . .	50
6.2.1	Grund der Aufnahme in das WMS . . . . .	51
6.2.2	SOLL-Zustand . . . . .	51
6.2.3	Messgrößen . . . . .	55
6.3	Workflow „Neuer Mitarbeiter“ . . . . .	55

6.3.1	Grund der Aufnahme in das WMS . . . . .	55
6.3.2	IST-Zustand . . . . .	55
6.3.3	Verbesserungspotenziale . . . . .	56
6.3.4	SOLL-Zustand . . . . .	56
6.3.4.1	Zusätzliche Aufwände vom IST- zum SOLL-Zustand	59
6.3.4.2	Verbesserungen vom IST- zum SOLL-Zustand .	59
6.3.5	Messgrößen . . . . .	60
6.4	Workflow „Anwendersupport“ . . . . .	60
6.4.1	Grund der Aufnahme in das WMS . . . . .	60
6.4.2	IST-Zustand . . . . .	60
6.4.3	Verbesserungspotenziale . . . . .	61
6.4.4	SOLL-Zustand . . . . .	61
6.4.4.1	Zusätzliche Aufwände vom IST- zum SOLL-Zustand	63
6.4.4.2	Verbesserungen vom IST- zum SOLL-Zustand .	63
6.4.5	Messgrößen . . . . .	64
<b>7</b>	<b>Anforderungen an das Workflow-Management-System</b>	<b>65</b>
7.1	Anforderungen . . . . .	65
7.1.1	Allgemeine Anforderungen . . . . .	65
7.1.2	Nichtfunktionale Anforderungen . . . . .	69
7.1.2.1	Funktionalität . . . . .	69
7.1.2.2	Zuverlässigkeit . . . . .	70
7.1.2.3	Benutzbarkeit . . . . .	70
7.1.2.4	Effizienz . . . . .	71
7.1.2.5	Änderbarkeit . . . . .	71
7.1.2.6	Übertragbarkeit . . . . .	71
7.1.2.7	Benutzerschnittstelle . . . . .	71
7.2	Build-or-Buy Analyse . . . . .	73
7.2.1	Track+ . . . . .	74
7.2.1.1	Vor- und Nachteile von Track+ . . . . .	74
7.2.2	Eigenentwicklung . . . . .	75
7.2.2.1	Vor- und Nachteile einer Eigenentwicklung . . .	75
7.2.3	Entscheidung . . . . .	75
<b>8</b>	<b>Entwicklung und Entwurf des Workflow-Management-Systems</b>	<b>79</b>
8.1	Architektur . . . . .	79
8.1.1	UML Diagramme . . . . .	80
8.1.2	MySQL Datenbanken . . . . .	83
8.1.2.1	WMS . . . . .	83
8.1.2.2	Zeitdokumentation . . . . .	84
8.1.3	Abläufe im WMS . . . . .	85
8.2	Implementierung . . . . .	86

8.2.1	Berechtigungskonzept . . . . .	86
8.2.2	Implementierung und Integration des WMS in das ZDS . .	87
8.2.2.1	Login . . . . .	88
8.2.2.2	Prozess starten . . . . .	89
8.2.2.3	Aufgabenliste . . . . .	90
8.2.2.4	Messung durchführen . . . . .	94
8.2.2.5	Verwaltung . . . . .	94
8.2.3	Eingesetzte Technologien . . . . .	96
8.2.3.1	JBoss jBPM 3.2.3 . . . . .	97
8.2.3.2	JBoss SEAM 2.1.1.GA . . . . .	103
8.2.3.3	JavaServer Faces . . . . .	107
8.2.3.4	JBoss Application Server 4.2.2.GA . . . . .	108
8.2.3.5	MySQL 5.0.67 Datenbank . . . . .	108
8.2.3.6	Eclipse 3.4.0 . . . . .	108
8.2.3.7	JFreeChart 1.0.12 . . . . .	108
8.2.4	Aufgetretene Probleme . . . . .	109
<b>9</b>	<b>Ergebnisse der Evaluierung</b>	<b>111</b>
9.1	Situation vor Einführung des WMS . . . . .	111
9.2	Situation nach Einführung des WMS . . . . .	112
9.3	Evaluierung der Benutzerschnittstelle . . . . .	112
<b>10</b>	<b>Zusammenfassung und Ausblick</b>	<b>113</b>
10.1	Umsetzung des Prozessmanagements . . . . .	113
10.2	Umsetzung WMS . . . . .	114
10.3	Ausblick . . . . .	114
<b>11</b>	<b>Begriffsdefinitionen</b>	<b>116</b>
<b>A</b>	<b>Quellcode</b>	<b>118</b>
A.1	Leeren der Datenbank . . . . .	118
A.2	SOLL-Zustand des Prozesses „Kundenwunsch“ . . . . .	119
A.3	SOLL-Zustand des Prozesses „Produktaktualisierung“ . . . . .	121
A.4	SOLL-Zustand des Prozesses „Neuer Mitarbeiter“ . . . . .	125
A.5	SOLL-Zustand des Prozesses „Anwendersupport“ . . . . .	128
<b>B</b>	<b>Fragebögen</b>	<b>130</b>
B.1	Ergebnisse des Fragebogens „Allgemeine Daten“ . . . . .	130
B.2	Ergebnisse des Fragebogens „Situation vor Einführung des WMS“	131
B.3	Ergebnisse des Fragebogens „Situation nach Einführung des WMS“	133
B.4	Ergebnisse des Fragebogens „Evaluierung der Benutzerschnittstelle“	135

# Abbildungsverzeichnis

2.1	Integriertes Geschäftsprozess- und Workflowmanagement [Gad07, S. 2] . . . . .	7
2.2	Generische Struktur einer Workflow-Software [Hol95, S. 13] . . . . .	8
2.3	Workflow Reference Model [Hol95, S. 20] . . . . .	9
2.4	Rollen im Prozessmanagement [Gad07, S. 4] . . . . .	9
2.5	Schematische Darstellung eines Einliniensystems . . . . .	13
2.6	Schematische Darstellung eines Mehrliniensystems . . . . .	14
2.7	Schematische Darstellung eines Stabliniensystems . . . . .	14
2.8	Schematische Darstellung einer funktionalen Organisation . . . . .	15
2.9	Schematische Darstellung einer divisionalen Organisation . . . . .	15
2.10	Schematische Darstellung einer Matrixorganisation [Büh04, S. 170] . . . . .	16
2.11	Schematische Darstellung einer prozessorientierten Organisation [GA04] . . . . .	17
2.12	Prozesslebenszyklus . . . . .	18
2.13	Prozesslandkarte [Len05, S. 33] . . . . .	20
2.14	Übersicht über diagrammbasierte Methoden [Gad07, S. 81] . . . . .	20
2.15	Pfeilformdarstellung eines Prozesses . . . . .	21
2.16	Ereignisgesteuerte Prozesskette aus [Sta06, S. 70] . . . . .	22
2.17	UML 2.1 Aktivitätsdiagramm aus [Bal08, S. 88] . . . . .	22
2.18	Prozessabgrenzung [Wag06, S. 99] . . . . .	23
2.19	Vorgehen bei der Wertschöpfungsanalyse . . . . .	25
2.20	Gestaltungsmaßnahmen der Ablaufanalyse [WP07, S. 154] . . . . .	27
2.21	Phasenmodell von Veränderungsprozessen [KM06, S. 11] . . . . .	30
2.22	Die ITIL Rahmenstruktur [Olb08] . . . . .	31
5.1	Prozessteam Meetings (PTM) . . . . .	44
6.1	SOLL-Zustand des Prozesses „Kundenwunsch“ . . . . .	48
6.2	SOLL-Zustand des Prozesses „Produktaktualisierung“ . . . . .	52
6.3	SOLL-Zustand des Prozesses „Neuer Mitarbeiter“ . . . . .	57
6.4	SOLL-Zustand des Prozesses „Anwendersupport“ . . . . .	62
7.1	Gantt-Diagramm [Ang05, S. 72] . . . . .	67
7.2	Entwurf der Aufgabenliste . . . . .	72

7.3	Entwurf der Aufgabedetails . . . . .	73
7.4	Details eines Vorgangs in Track+ . . . . .	78
8.1	Architektur des WMS . . . . .	81
8.2	Paketdiagramm des WMS . . . . .	81
8.3	Klassendiagramm des WMS . . . . .	82
8.4	Diagramm der Datenbank des WMS . . . . .	85
8.5	Diagramm der Datenbank des Zeitdokumentationssystems . . . . .	86
8.6	Module des Frameworks JBoss jBPM [JBo08a] . . . . .	98
8.7	Grafischer Designer für jPDL (integriert in Eclipse) . . . . .	99
8.8	Übersicht über Java EE 5.0 [JBo08b] . . . . .	104

# Tabellenverzeichnis

7.1	Erfüllung der Anforderungen durch Track+ bzw. jBPM und SEAM. Gew.: Gewichtung in Prozent; Erf.: Grad der Erfüllung (-1 bis 1) .	76
B.1	Ergebnisse des Fragebogens „Allgemeine Daten“ . . . . .	130
B.2	Ergebnisse des Fragebogens „Situation vor Einführung des WMS“	133
B.3	Ergebnisse des Fragebogens „Situation nach Einführung des WMS“	135
B.4	Ergebnisse des Fragebogens „Evaluierung der Benutzerschnittstelle“	137

# Listings

8.1	Liste aller Links und Parameter zur Steuerung von WMS-Funktionen	87
8.2	Neu laden des Hauptfensters des ZDS	88
8.3	Aufrechterhalten der Anmeldung am WMS	88
8.4	iFrame für Schnittstelle zu WMS	88
8.5	Externer Login	89
8.6	Prozess starten	89
8.7	Anzeige aller Prozessdefinitionen	89
8.8	Starten einer Prozessinstanz	90
8.9	Auslesen der Prozessaufgaben für einen Anwender	91
8.10	Anzeige paralleler Prozessaufgaben	93
8.11	Erstellen eines GANTT-Diagrammes	94
8.12	Status der Prozessinstanzen	95
8.13	jPDL Prozess Demo	100
A.1	Leeren der Datenbank	118
A.2	Prozess „Kundenwunsch“	119
A.3	Prozess „Produktaktualisierung“	121
A.4	Prozess „Neuer Mitarbeiter“	125
A.5	Prozess „Anwendersupport“	128

# Kapitel 1

## Einleitung

### 1.1 Problemstellung

Um sich als Organisation im Wettbewerb durchzusetzen, ist es notwendig, auf Veränderungen des Marktes flexibel und rasch zu reagieren, sich auf seine Kernkompetenzen zu konzentrieren und diese besser als die Konkurrenz zu beherrschen und sich rechtzeitig auf Veränderungen (interne sowie externe) einzustellen [Wag06, S. 1]. Um dies zu erreichen, sollten die organisationsinternen Prozesse ressourcenoptimal durchgeführt werden, so dass sich die Organisation gänzlich auf die Kundenbedürfnisse konzentrieren zu kann. Dies kann unter anderem durch die Einführung eines Prozessmanagement-Systems geschehen. Vor der Einführung eines solchen Systems sind die Arbeitsabläufe oft ungenügend genau dokumentiert oder nur unzureichend zwischen den Mitgliedern verschiedener Abteilungen abgestimmt, Was vor allem bei Organisationen im Dienstleistungsbereich negative Auswirkungen auf die Kundenzufriedenheit hat. [WP07, S. 53]

Organisationen haben eine bestimmte Struktur, sie sind meistens in mehrere Arbeitsgruppen oder Abteilungen unterteilt, welche zwar eigene Aufgabenbereiche haben, dennoch alle zusammenarbeiten. Die Abteilungen sind von einer guten Zusammenarbeit zwischen den einzelnen Teams abhängig. Verändert sich die Organisation, wird beispielsweise immer wieder neues Personal aufgenommen oder kommen für die Mitarbeiter neue Aufgabenbereiche hinzu, so kann die Zusammenarbeit unter Umständen erschwert werden, wenn Verantwortlichkeiten von Anfang an nicht klar definiert bzw. die Mitarbeiter nicht ausreichend geschult werden.

Eine gründliche Schulung der Mitarbeiter ist vor allem dann notwendig, wenn ein Mitarbeiter gleichzeitig an mehreren, voneinander grundlegend verschiedenen Projekten arbeitet. Da sich die Produkte voneinander unterscheiden und somit die Arbeitsabläufe sehr unterschiedlich sein können, kann es vorkommen, dass einzelne Arbeitsschritte, welche keine unmittelbare Auswirkung auf den Kunden

haben (z.B.: die rechtliche Verbindlichkeit zur Dokumentation eines bestimmten Vorgangs) schlichtweg vergessen werden.

Bei der Einschulung von neuen Mitarbeitern können Probleme entstehen, sofern die einzelnen Arbeitsabläufe nicht genau dokumentiert sind. Diese übernehmen zwangsweise die Arbeitsmethodik der sie einschulenden Kollegen, wodurch falsches Verhalten bzw. fehlende Abläufe bei Arbeitsschritten weitergegeben werden. Auf diese Art und Weise sind Fehler unvermeidbar. Aus der fehlenden Dokumentation der Arbeitsabläufe ergibt sich auch, dass Verantwortlichkeiten oft nicht oder nicht ausreichend klar definiert sind. Hat ein Mitarbeiter bei einem Arbeitsschritt ein Problem, wird dies häufig an den Projektleiter/Abteilungsleiter und nicht an die eigentlich verantwortliche Person kommuniziert. Dies führt dazu, dass der Projektleiter/Abteilungsleiter mit Angelegenheiten konfrontiert wird, welche ihn von seiner eigentlichen Arbeit ablenken.

Da die Personengruppe aus der Managementebene meist genaue Vorstellungen hat, wie Arbeiten erledigt werden sollten, kann es vorkommen, dass Konfliktpotenziale zwischen Management und den Mitarbeitern entstehen, falls ein Mitarbeiter (mangels Dokumentation der Arbeitsschritte) Arbeitsaufträge anders abwickelt, als es vom Management gewünscht wird. Auch das Arbeitsklima unter den Mitarbeitern kann darunter leiden, da aufgrund der unterschiedlichen Anzahl der Arbeitsschritte pro Prozess manche Mitarbeiter den Arbeitsprozess schneller bearbeiten als andere und diese somit als langsam gelten, obwohl das Produkt ihrer Arbeit in der Wertigkeit sehr hoch sein kann. Alle diese Unstimmigkeiten innerhalb einer Organisation wirken sich negativ auf die Zufriedenheit der internen und externen Kunden aus.

Die reine Dokumentation der Arbeitsschritte ist für eine Verbesserung der Zusammenarbeit zwischen Arbeitsgruppen sowie die Steigerung der Kundenzufriedenheit nicht ausreichend. Zusätzlich zu der Dokumentation der Arbeitsschritte muss der gesamte Prozess, der die Arbeitsschritte umgibt, analysiert und optimiert werden. Weiters ist es notwendig, dass die Mitarbeiter bei ihrer täglichen Arbeit Zugriff auf dieses Wissen haben, um es effektiv anwenden zu können.

## **1.2 Zielsetzung**

Diese Masterarbeit zeigt auf, welche Schritte notwendig sind, um ein Workflow--Management-System (WMS) in einer Organisation einzuführen. Unter anderem wird beantwortet, welcher Aufwand für die Einführung zu erwarten ist, wie die Arbeitsabläufe zu erfassen sind, wie die Arbeitsabläufe optimal erfasst werden

können, um sie später in ein dynamisches System <sup>1</sup> einbinden zu können und wie hoch die Akzeptanz der Mitarbeiter ist.

Die Hypothese ist, dass in Organisationen bei den meisten abteilungsübergreifenden Arbeitsabläufen sehr viel Zeit bei der Kommunikation zwischen den verschiedenen Abteilungen verloren geht, da oft nicht klar ist, ob bestimmte Arbeitsschritte bereits abgeschlossen sind oder nicht. Daher soll untersucht werden, wie diese überflüssige Kommunikation einerseits durch die Verwendung eines WMS, andererseits durch die Optimierung der Arbeitsprozesse vermieden werden kann.

Die Ziele dieser Masterarbeit gliedern sich in zwei Teilbereiche auf. Erstens, Ziele welche durch die Umsetzung eines Prozessmanagements erreicht werden können, zweitens Ziele welche mit Hilfe des entwickelten Workflow-Management-Systems erreicht werden können.

Auf der Seite des Prozessmanagements soll sichergestellt werden, dass Prozesse optimal aufeinander abgestimmt, vollständig erfasst und dokumentiert sind, um einerseits eine mögliche ISO Zertifizierung vorzubereiten und andererseits die Mitarbeiter bei ihrer täglichen Arbeit zu unterstützen. Die Unterstützung der Mitarbeiter bei der täglichen Arbeit soll mit Hilfe eines Workflow-Management-Systems erreicht werden. Dadurch soll u.a. der Kommunikationsaufwand zwischen den Mitarbeitern gesenkt werden. Das WMS soll dafür sorgen, dass Prozesse vollständig abgearbeitet werden bzw. dass Änderungen im Ablauf zumindest dokumentiert sind. Mit dem WMS soll es dem Management möglich sein, die Auslastung der Mitarbeiter zu überwachen sowie eine regelmäßige Berichterstattung über die Performance der Prozesse zu generieren.

Durch das Erreichen dieser Ziele soll garantiert werden, dass die Mitarbeiter einer Organisation die Arbeitsabläufe tatsächlich befolgen, um gleichzeitig die Produktivität der Organisation sowie die Kundenzufriedenheit zu steigern.

Anhand eines Fallbeispiels soll aufgezeigt werden, wie Prozesse erfasst und ein Workflow-Management-System implementiert werden kann, welches Maßnahmen zur Erreichung der Ziele dieser Arbeit umsetzt.

### **1.3 Aufbau der Arbeit**

In Kapitel 2 werden die Grundlagen des Prozess- bzw. Workflowmanagements erläutert. Darüber hinaus wird ein kurzer Einblick in die IT Infrastructure Library (ITIL) gegeben.

Im darauf folgenden Kapitel 3 wird der Kontext des Fallbeispiels beschrieben und ein kurzer Überblick über die Organisation gegeben.

---

<sup>1</sup>Ein System, welches die Benutzer aktiv unterstützt, ihnen beispielsweise nur Informationen anzeigt, welche zum aktuellen Zeitpunkt benötigt werden.

Kapitel 4 beschreibt die Optimierungsbedürfnisse der Organisation aus Kapitel 3.

Kapitel 5 behandelt die Methodik, wie im Kontext der Organisation Prozesse erfasst werden können bzw. wie bei dem Entwurf und der Implementierung des Workflow-Management-Systems vorgegangen wird.

Eine Übersicht über die erhobenen Workflows wird im anschließenden Kapitel 6 gegeben. Es wird jeweils u.a. der ideale SOLL-Zustand der Workflows beschrieben sowie sämtliche für die Erreichung des SOLL-Zustandes notwendigen zusätzlichen Aufwände.

Anschließend werden im Kapitel 7 die für die Implementierung des Workflow-Management-Systems notwendigen Anforderungen geschildert. Zusätzlich wird eine Evaluierung zwischen zwei Softwaresystemen durchgeführt. Dies dient der Entscheidung, welches der beiden Systeme die Basis des WMS bilden soll.

Im Kapitel 8 werden der Entwurf sowie die Implementierung des WMS beschrieben. Es wird ein Überblick über die verwendete Architektur, die Implementierung selbst, die eingesetzten Technologien und eine Auswahl der aufgetretenen Probleme gegeben.

Kapitel 9 beinhaltet die Ergebnisse der Evaluierung dieser Arbeit.

Schließlich werden in Kapitel 10 die Ergebnisse dieser Arbeit zusammengefasst. Weiters wird ein Ausblick auf mögliche aufbauende Arbeiten gegeben.

## Kapitel 2

# Grundlagen des Prozessmanagements

Dieses Kapitel gibt einen Überblick über das Prozess- bzw. Workflowmanagement sowie die IT Infrastructure Library (ITIL). Es werden die Grundlagen dieser Themenbereiche erklärt.

Der Begriff Prozessmanagement wird in der Literatur, je nach Schwerpunktsetzung, unterschiedlich definiert. Das Spektrum reicht von einer operativen Betrachtung (v.a. der Steigerung der Kundenzufriedenheit) bis hin zur prozessorientierten Unternehmensführung. Außerdem werden in der Literatur für unterschiedliche Branchen speziell angepasste Prozessmanagementschemata vorgestellt.[Fin03, S. 26]

In dieser Arbeit wird Prozessmanagement hauptsächlich von dem operativen Standpunkt betrachtet, auch wenn teilweise auf Aspekte des prozessorientierten Managements eingegangen wird.

### 2.1 Geschäftsprozessmanagement

Organisationen sind vielen, sich ständig ändernden Herausforderungen ausgesetzt, deren Bewältigung die Schaffung geeigneter Prozesse voraussetzt. Das Geschäftsprozessmanagement hilft bei der Bewältigung dieser Herausforderungen und ist somit für die Erhaltung und Verbesserung der Wettbewerbsfähigkeit der Organisation wichtig [All05, S. 4].

Das Geschäftsprozessmanagement (Business Process Management (BPM)/ Prozessmanagement) zielt auf die systematische Gestaltung, Überwachung und Weiterentwicklung der Geschäftsprozesse eines Unternehmens ab [All05, S. 12]. Ein (Geschäfts-)Prozess ist eine Abfolge von Funktionen (Aktivitäten) zur Erfüllung

einer betrieblichen Aufgabe, wobei die Leistung in Form von Informations- und/oder Materialtransformation erbracht wird [All05, S. 8]. Beim Prozessmanagement soll zu jedem Zeitpunkt klar sein, welche Person (oder welches IT Produkt) auf welche Art und Weise eine Aufgabe durchführt [Kra08, S.11]. Um die Leistungsfähigkeit der Prozesse zu messen, werden Messgrößen definiert und in geregelten Abständen Kennzahlen erhoben. Basierend auf diesen Kennzahlen können Optimierungspotenziale erkannt und in einem weiteren Schritt realisiert werden.

### **2.1.1 Prozess- und Workflowmanagement**

Ein Workflow bezeichnet die Automation von Prozeduren, bei der Dokumente, Informationen oder Aufgaben zwischen verschiedenen Benutzern, basierend auf zuvor definierten Regeln, ausgetauscht werden, um ein Geschäftsziel zu erreichen oder zur Erreichung dessen beizutragen. [Hol95, S. 6]

Das Geschäftsprozessmanagement in einer Organisation umfasst mehrere Ebenen (Abbildung 2.1). Auf der strategischen Ebene werden die Geschäftsfelder sowie die dazugehörigen Erfolgsfaktoren der Organisation betrachtet. Auf dieser Ebene findet die Entwicklung von Strategien statt, welche in die Gestaltung der Geschäftsprozesse einfließt. In der fachlich-konzeptionellen Ebene werden aus der Strategieentwicklung Ziele abgeleitet und mittels Prozessmanagement im Unternehmen etabliert. Das Geschäftsprozessmanagement reicht von der Definition der Prozesse und der Abgrenzung der Prozesse voneinander, über die Modellierung der idealen Soll-Prozesse, bis hin zur Durchführung derselben. Auf der darunter liegenden operativen Ebene werden die Workflows mit Hilfe eines Workflow-Management-Systems durchgeführt. Die Workflowmodellierung orientiert sich an der Prozessmodellierung<sup>2</sup>, wobei der Prozess um zusätzliche Informationen ergänzt wird, welche die semiautomatische Ausführung im WMS ermöglichen. Bei der Workflowausführung werden Prozessinstanzen erzeugt und anhand des modellierten Prozessmodells ausgeführt. Um die Leistung der Workflows quantifizieren zu können, ist es notwendig, Messgrößen zu erfassen und auszuwerten. Dies geschieht im Rahmen des Prozessmonitorings. \*Die daraus entstehenden Ergebnisse haben einen Einfluss auf die Organisationsgestaltung. Die Anwendungssystemgestaltung orientiert sich an den Bedürfnissen des Workflowmanagements. [Kra08, S. 4], [Gad07, S. 1 - 3]

---

<sup>2</sup>= Die Abstraktion und meist grafische Darstellung von Geschäftsprozessen

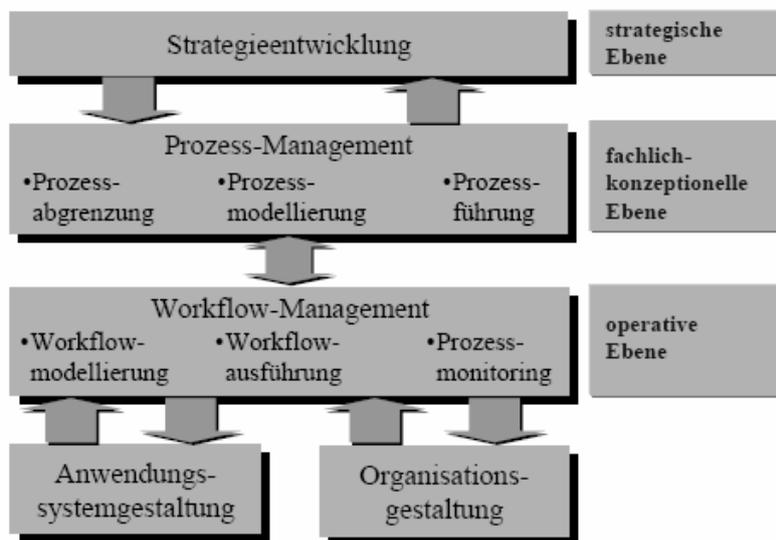


Abbildung 2.1: Integriertes Geschäftsprozess- und Workflowmanagement [Gad07, S. 2]

## 2.1.2 Workflow-Management-Systeme

Ein Workflow-Management-System unterstützt die automatische Ausführung von Geschäftsprozessen, indem es die korrekte Abfolge von Aufgaben ausführt, sowie dabei die richtigen Personen oder IT-Ressourcen den Aufgaben zuordnet. Das bedeutet, dass verschiedene Softwareprodukte in einer bestimmten, zuvor definierten Reihenfolge, basierend auf einer Workflow-Logik (Prozessdefinition), aufgerufen werden. [Hol95, S. 6] Da Geschäftsprozesse sehr unterschiedlich sein können (Durchlaufzeit, Komplexität), ist je Einsatzgebiet (verteilte Organisation, kleine Organisation, ...) ein anders ausgelegtes WMS notwendig. Die Prinzipien, auf denen die WMS basieren, sind jedoch meistens sehr ähnlich. Die WFMC<sup>3</sup> hat ein Modell eines Workflow-Management-Systems entwickelt, welches versucht möglichst viele verschiedene WMS-Implementierungen sowie Anwendungsgebiete zu vereinen.

Eine Darstellung eines WMS ist in Abbildung 2.2 zu finden. Daraus ist abzulesen, wie die Interaktion zwischen WMS, Benutzern sowie externen Anwendungen erfolgen kann. Aus dieser Struktur lässt sich das Workflow Reference Model (Abbildung 2.3, [Hol95, S. 20]) ableiten:

- Es besteht aus einem grafischen Editor, mit welchem eine Prozessdefinition erstellt werden kann (*Process Definition Tools*),

<sup>3</sup>Workflow Management Coalition: <http://wfmc.org>



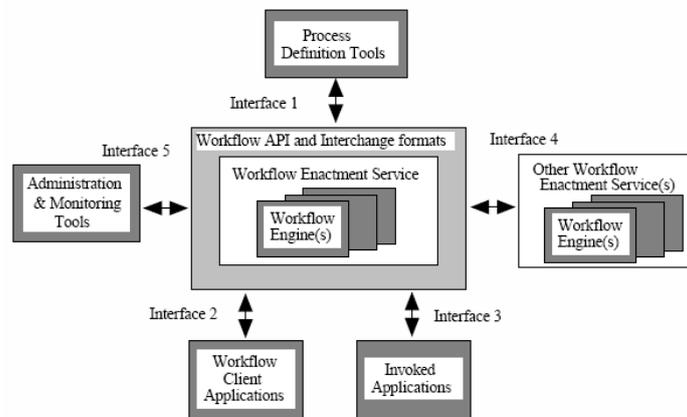


Abbildung 2.3: Workflow Reference Model [Hol95, S. 20]

Personen bekannt sein sollten. Die Einführung eines Prozessmanagements (PM) ist auch ohne jegliche Kenntnis dieser Rollen möglich, jedoch besteht die Gefahr, dass wichtige Rollen/Personen übersehen werden. Besonders bedeutsam ist es, die Rollen richtig (mit den am besten geeigneten Personen) zu besetzen. [Wal01, S. 2] In dieser Arbeit werden die Rollenbegriffe, wie sie von *Gadatsch* ([Gad07, S. 4 ff.]) definiert wurden, verwendet.

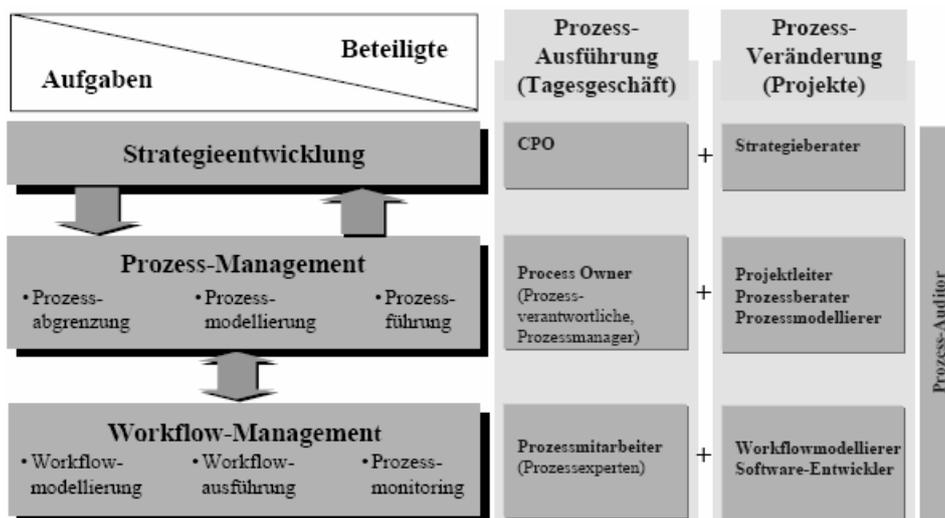


Abbildung 2.4: Rollen im Prozessmanagement [Gad07, S. 4]

Laut *Gadatsch* existieren im Geschäftsprozessmanagement folgende Rollen:

**Chief Process Officer (CPO)** Diese Person ist für die Steuerung und Optimierung

der Geschäftsprozesse verantwortlich. In großen Organisationen mit mehreren Abteilungen/Arbeitsgruppen hat diese Rolle die Aufgabe, ein übergeordnetes Unternehmensprozessmodell zu entwickeln, welches u.a. folgende Punkte beinhaltet:

- Die Definition der Rollen des Geschäftsprozessmanagements und die organisatorische Einordnung dieser.
- Die Vorgehensweise beim Prozesscontrolling und bei der Prozessoptimierung.
- Den Ablauf der Prozessidentifikation.

Vgl. [SS07, S. 155]

**Process Owner** Diese Rolle wird in der Literatur auch Prozessverantwortlicher oder Prozessmanager genannt. Der Hauptaufgabenbereich ist die laufende Steuerung und Optimierung der Geschäftsprozesse.

**Prozessmitarbeiter** [SS07, S. 164] ist der Ansicht, dass den Prozessmitarbeitern die wichtigste Rolle im Geschäftsprozessmanagement zukommt: *„Sie führen nicht nur fachspezifische Aufgaben auf der operativen Prozessebene aus, sondern sie sind auch die wichtigsten Initiatoren und Realisierer von Prozessverbesserungen“*. Prozessmitarbeiter sind keine gesteuerten Objekte, sondern gestaltende Subjekte.

In vielen Organisationen werden den Prozessmitarbeitern zusätzliche Handlungsspielräume eingeräumt. Dadurch ist es für sie möglich, auf freiwilliger Basis Verantwortung zu übernehmen und schneller sowie angemessener auf unerwartete Situationen zu reagieren. Durch dieses Handeln, welches zusätzlich zu ihren eigentlichen Aufgaben erfolgt, ändert sich das Anforderungsprofil an die Mitarbeiter, da es um Prozess Know-How ergänzt wird. Darüber hinaus müssen Prozessmitarbeiter mittels Personalentwicklung auf eine erweiterte Verantwortung und intensivere Kommunikation vorbereitet werden. Vgl. [SS07, S. 164]

**Projektleiter** Der Projektleiter unterstützt die erstmalige Implementierung des Prozessmanagements. Insbesondere ist er verantwortlich für die

- Leitung des Prozessmanagementteams,
- Führung der Projektmitarbeiter und die Berichterstattung an die Unternehmensleitung.

Vgl. [Gad07, S. 6]

**Prozessberater** Prozessberater sind vor allem bei der Einführung eines Geschäftsprozessmanagements wichtig. Gerade für Unternehmen, die zum ersten Mal ein Prozessmanagementsystem einführen, sind sie von großer Bedeutung. Prozessberater begleiten die Einführung eines Prozessmanagementsystems bis die Implementierung abgeschlossen ist, die Methoden bei den Mitarbeitern gefestigt sind sowie ein kontinuierlicher Verbesserungsprozess etabliert ist.

In vielen Organisationen ist der Prozessberater ein Mitarbeiter des eigenen Unternehmens. Dies bietet einerseits die Vorteile, dass die entsprechende Person das Unternehmen und dessen Kultur bereits kennt, andererseits bleibt das gesammelte Wissen des Prozessberaters dem Unternehmen erhalten. Vgl. [SS07, S. 155]

**Modellierer** Der Modellierer beschreibt die Workflows genau und legt die technische Umsetzung fest.

**Prozessauditor** Der Prozessauditor ist nach der Etablierung eines Prozesses wichtig. Er deckt mögliche Schwachstellen bei der Prozessausführung auf und versucht, gemeinsam mit dem Prozessteam, diese zu bereinigen.

**Prozessteam** Eine wesentliche Rolle beim Prozessmanagement spielt das Prozessteam. Es wird vom Prozessverantwortlichen (Process Owner) geführt und setzt sich aus einer Gruppe von ungefähr vier bis acht Personen zusammen, die an dem Prozess teilnehmen. Es ist darauf zu achten, dass diese Personen einerseits fachlich sowie sozial geeignet sind und andererseits den Willen zur kontinuierlichen Veränderung besitzen. Vgl. [Wag06, S. 91]

#### 2.1.4 Gründe für Prozessmanagement

Die Anforderungen an ein Unternehmen steigen kontinuierlich an. Schnelle Kommunikationswege wie das Internet in Verbindung mit der Leichtigkeit, mit der es heutzutage möglich ist um die Welt zu reisen, spornen Unternehmen an, weltweit zu agieren und neue Märkte zu erforschen. Durch die zunehmende Markttransparenz ist die Anzahl der Anbieter von Waren und somit der Konkurrenzdruck enorm gestiegen. Nur diejenigen Unternehmen, welche sich schnell an die wechselnden Marktbedürfnisse, neue Technologien oder veränderte Kundenwünsche anpassen können, sind im Stande, sich von ihren Konkurrenten absetzen. Für Unternehmen ist es wichtig, den stetigen Wandel als permanente Herausforderung und kontinuierlichen Prozess zu betrachten. Dieser Wandel betrifft alle Bereiche eines Unternehmens. Vgl. [SS07, S. 1 f.].

Die Prozessorientierung hat für Unternehmen sowohl eine strategische (Identifikation von Erfolgspotenzialen, Schaffung neuer Kernkompetenzen) als auch eine operative (Steigerung von Effizienz und Qualität) Bedeutung. [Fin03, S. 23f.]

Bei einer von *Steinle* [STK00] durchgeführten Expertenbefragung wurden vor allem folgende Gründe für ein Prozessmanagement angegeben: Kostensenkungspotenzial, stärkere Markt- und Kundenorientierung, Schnittstellenprobleme in funktionalen Organisationen.

Ein wesentliches Ziel der prozessorientierten Unternehmensführung ist eine starke Prozessorientierung, die höhere Effizienz- und Effektivitätssteigerungen ermöglicht als die Orientierung an traditionellen Bezugsobjekten wie Leistungen, Produkt-Markt-Kombinationen, funktionale Organisationseinheiten oder Technologien. [BWW03, S. 4]

Zu den entscheidenden Faktoren, die dazu beitragen, um bei einer prozessorientierten Unternehmensführung erfolgreich zu sein, zählen laut [BWW03, S. 4]:

- **Kundenorientierung:** Ein wichtiges Ziel eines prozessorientierten Unternehmens ist die Zufriedenheit seiner Kunden. Kundenorientierung bedeutet, dass jeder Mitarbeiter zur Erfüllung der Kundenzufriedenheit beiträgt.
- **Qualitäts-, Zeit- und Kostenorientierung:** Mit Qualität ist hier die einem Kunden versprochene Produkt- sowie Servicequalität gemeint. Bei der Kostenorientierung werden für die Prozesse Kosten berechnet, um auf diese Art und Weise Kostentreiber zu identifizieren und gezielt an diesen Stellen anzusetzen, um die Kosten zu senken. Zeitorientierung bezeichnet in diesem Zusammenhang eine Liefer- bzw. Entwicklungszeit.
- **Flexibilität** kann in zwei Gebiete unterteilt werden: Quantitative Flexibilität (die Fähigkeit, unterschiedliche Mengen eines Produktes herzustellen) und Qualitative Flexibilität (die Fähigkeit, ein Produkt mit verschiedenen qualitativen Eigenschaften zu produzieren).
- **Mitarbeiterorientierung:** Die eigenen Mitarbeiter sind für Organisationen eine der wichtigsten Ressourcen, denn motivierte und gut ausgebildete Mitarbeiter sind zu wesentlich höheren Leistungen fähig als unmotivierte. Deswegen macht es für Organisationen oft Sinn, in Fortbildung sowie soziale Leistungen, die Mitarbeiter betreffend, zu investieren.

### **2.1.5 Prozessorientierte Sichtweise einer Organisation**

Es gibt viele verschiedene Arten, wie ein Unternehmen organisiert sein kann. Es wird zwischen der hierarchischen Struktur (*Organisationstruktur*) sowie der Art,

wie die Organisation innerhalb der Struktur funktioniert (*Organisationsform*), unterschieden. Es existiert eine Vielzahl verschiedener Organisationsformen und Organisationsstrukturen, weshalb im Folgenden nur die wichtigsten kurz vorgestellt werden.

### 2.1.5.1 Organisationsformen

#### 2.1.5.1.1 Ein- und Mehrliniensystem

Im Einliniensystem (vgl. Abbildung 2.5) herrscht eine klare Befehlshierarchie. Eine Person hat immer genau einen Vorgesetzten und ein Vorgesetzter hat immer mehrere ihm untergebene Personen. Dadurch sind die Befugnisse und Zuständigkeiten klar definiert. Dieses System geht aber mit einem gewissen Mangel an Flexibilität einher. Einerseits wird die Kommunikation in der gesamten Organisation erschwert, andererseits hat das Management in den höheren Ebenen oft einen erhöhten Koordinationsaufwand. Diese Mängel vermag das Mehrliniensystem (vgl. Abbildung 2.6) teilweise zu lösen. Hier regiert das Prinzip des „kürzesten Weges“. Jede Person hat hier mehrere Vorgesetzte, umgekehrt hat jede Person mehrere ihr untergebene Personen. Kommunikation zwischen Abteilungen kann wesentlich schneller erfolgen. Dadurch kann die Zeit, welche für die Durchführung von Anweisungen und die Erteilung derselben benötigt wird, verkürzt werden. Jedoch sollten die Zuständigkeiten und Weisungsbefugnisse im Vorfeld gut abgeklärt werden, um ein reibungsloses Funktionieren zu ermöglichen. [Sch00, S. 2 - 4]

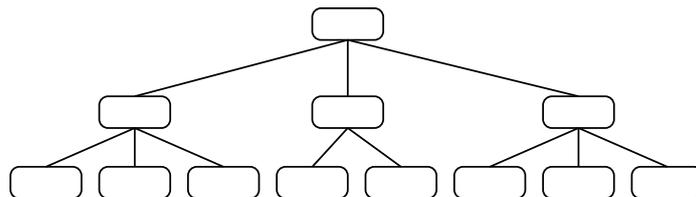


Abbildung 2.5: Schematische Darstellung eines Einliniensystems

#### 2.1.5.1.2 Stabliniensystem

Das Stabliniensystem (vgl. Abbildung 2.7) ist eine Spezialform des Ein- bzw. Mehrliniensystems. Manchen Entscheidungsträgern ist eine so genannte Stabsstelle zugeteilt. In der Abbildung ist diese als Doppelkreis dargestellt. Diese Stabsstelle hat die Aufgabe, den Entscheidungsträger zu beraten, Ideen zu entwickeln

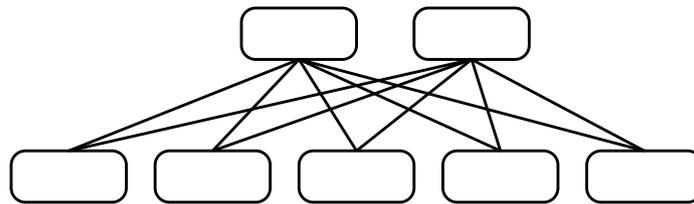


Abbildung 2.6: Schematische Darstellung eines Mehrliniensystems

und Strategien zu planen. Dadurch werden zwar einerseits höhere Instanzen entlastet, es birgt jedoch andererseits das Risiko, dass die Stäbe die Entscheidungsträger manipulieren könnten. [Pap02, S. 4], [Sch05, S. 3]

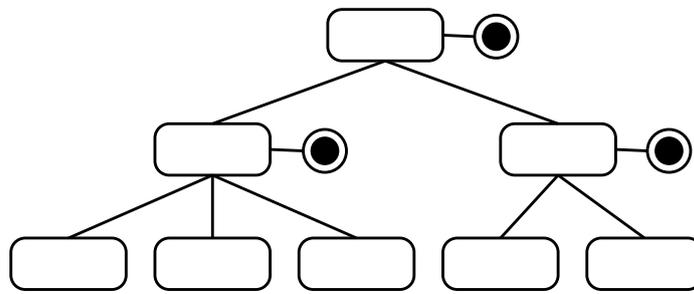


Abbildung 2.7: Schematische Darstellung eines Stabliniensystems

### 2.1.5.2 Organisationsstrukturen

Kaum eine der hier beschriebenen Organisationsstrukturen wird in der Praxis tatsächlich genauso verwendet, wie sie hier abgebildet ist. Die hier beschriebenen Strukturen sind viel mehr als ideale Modelle zu sehen. In der Realität existieren immer Mischformen dieser Modelle.

#### 2.1.5.2.1 Funktionale Organisation

Eine funktionale Organisation (auch Verrichtungsorganisation) ist in der zweiten Hierarchieebene in die funktionalen Bereiche gegliedert (z.B.: Einkauf, Fertigung). Darunter sind die Ebenen oft divisional (z.B.: die einzelnen Produkte) unterteilt. Das Management nimmt oft die Rolle eines Koordinators ein, welcher bei allen wichtigen Entscheidungen mit einbezogen wird. [Sch00, S. 8]

#### 2.1.5.2.2 Divisionale Organisation

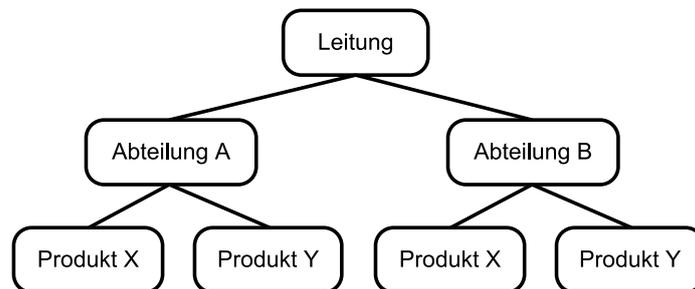


Abbildung 2.8: Schematische Darstellung einer funktionalen Organisation

Im Gegensatz zur funktionalen Organisation folgt die divisionale Organisation (auch Sparten-Organisation) einer Objektorientierung. Die zweite Hierarchieebene ist in Produkte, Regionen oder Kunden gegliedert. Daraus resultieren sehr selbstständige Unternehmenseinheiten. Auf der dritten Hierarchieebene sind diese Organisationen oftmals funktional gegliedert. Um zu verhindern, dass die einzelnen Sparten sich zu weit voneinander entfernen, werden oft bestimmte Stellen wie zum Beispiel die Rechts- oder Personalabteilung gemeinsam eingerichtet. [Töp07, S. 1220 f.], [Sch05, S. 9 f.]

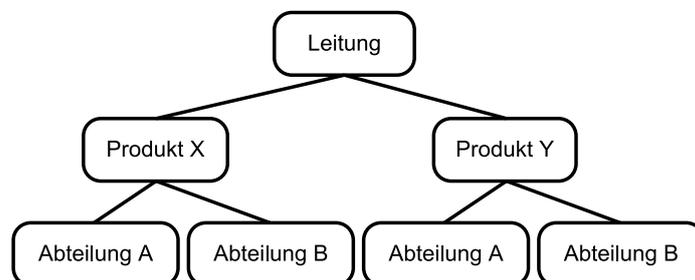


Abbildung 2.9: Schematische Darstellung einer divisionalen Organisation

### 2.1.5.2.3 Matrixorganisation

Die Matrixorganisation ist aus dem Problem entstanden, dass Unternehmen immer komplexer werden. Dadurch steigt der Koordinationsaufwand nach innen und es entsteht ein Repräsentationsproblem nach außen, da es für viele Kunden unklar ist, wer ihr jeweiliger Ansprechpartner ist. (Vgl.[Oel99, S. 75])

Die Matrixorganisation ist eine Mischung aus funktionaler und divisionaler Organisation, insofern dass beide Ausrichtungen parallel verfolgt werden. Im Gegensatz zur funktionalen und divisionalen Organisation werden in der zweiten Hierarchieebene beide Ansätze gleichzeitig verfolgt. [Töp07, S. 1228]

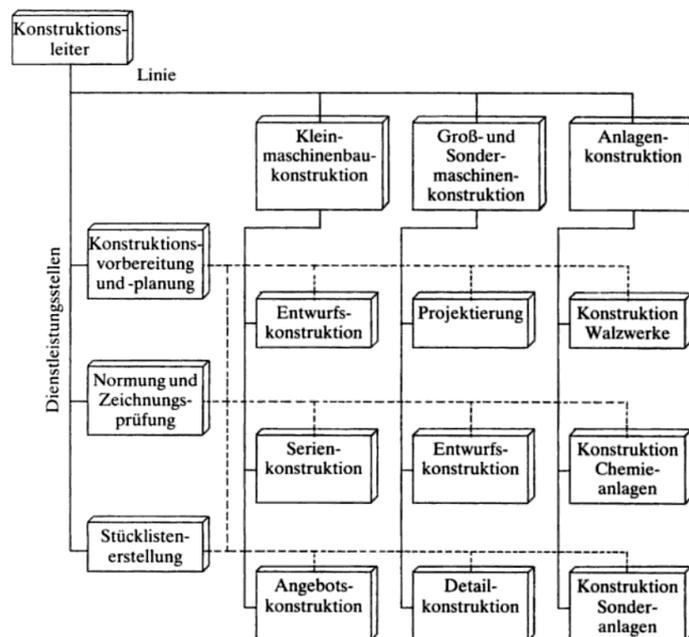


Abbildung 2.10: Schematische Darstellung einer Matrixorganisation [Büh04, S. 170]

#### 2.1.5.2.4 Prozessorientierte Organisation

Die prozessorientierte Organisation ähnelt in ihrem Aufbau sehr der Matrixorganisation, sie verlangt jedoch eine konsequente Ausrichtung auf die Geschäftsprozesse. Diese sollen optimal ausgeführt werden. Der Prozessverantwortliche übernimmt die Steuerung eines Prozesses und die funktionale Gliederung wird fast gänzlich aufgehoben. Dies führt in der Praxis bei einer Umstellung der Organisationsstruktur auf eine prozessorientierte Organisation oft zu erheblichen Widerständen. [Wag06, S. 11-12], [KV05, S. 221 - 222]

#### 2.1.6 Prozesslebenszyklus

Das „Leben“ eines Prozesses wird in der Literatur meistens als Zyklus, ähnlich dem in Abbildung 2.12, dargestellt. Zuerst wird ein „Prozess-Kandidat“ auf seine Prozesswürdigkeit untersucht. Die Prozesswürdigkeit ist gegeben, falls der Nutzen der Aufnahme des Prozesses in das Prozessmanagementsystem (PMS) die dabei entstehenden Kosten überwiegt. Wird ein Prozess als würdig erklärt, so gilt es, diesen zu analysieren und von den anderen Prozessen im PMS abzugrenzen. Weiters sollten die Auswirkungen dieses Prozesses auf die anderen, die zeitliche Abfolge der durchzuführenden Aktivitäten und die benötigten Organisationseinheiten und

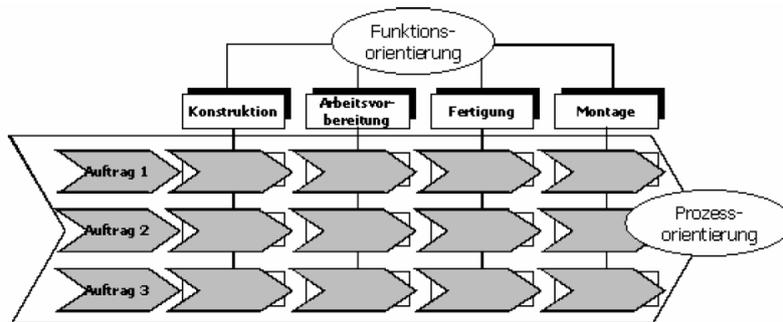


Abbildung 2.11: Schematische Darstellung einer prozessorientierten Organisation [GA04]

Rollen untersucht werden. Dies geschieht in der Phase „Aufnahme ins PM System“.

In der darauf folgenden Phase „Definition und Implementierung“ sollte eine abermalige Validierung der Prozesswürdigkeit erfolgen, da mittlerweile wesentlich mehr Fakten zu dem Prozess bekannt sind. Wird die Prozesswürdigkeit positiv beurteilt, so kann der Prozess im Detail definiert werden. Die Details, wie dies erfolgen könnte, sind Kapitel 2.1.8 zu entnehmen. Bei der Implementierung muss es sich nicht um eine rein technische Umsetzung handeln, auch organisatorische Lösungen sind denkbar [Bor02, S. 154].

In der Phase der „Durchführung und Optimierung“ wird der Prozess in den produktiven Betrieb übernommen. Da während der Prozessausführung spezielle Herausforderungen auftreten oder Sonderfälle eintreten können, gibt es in dieser Phase auch eine Optimierung. Dies bedeutet, dass die Ausführungslogik des laufenden Prozesses von dem Prozessverantwortlichen sowie den Prozessmitarbeitern flexibel an die geänderten Bedürfnisse angepasst werden kann.

Die während der Durchführung gesammelten Informationen über den Prozess werden in der nächsten Phase („Monitoring“) ausgewertet. Diese Daten bilden die Basis für Vergleiche zwischen verschiedenen Prozessinstanzen. Es können statistische Werte wie z.B. die Durchlaufzeit<sup>4</sup> oder die Prozesszeit<sup>5</sup> erhoben werden. Mögliche Verbesserungspotenziale können dadurch erkannt und in der nächsten Phase realisiert werden. Treten im Prozessablauf Fehler auf, so kann von dieser Phase aus in eine beliebige andere gewechselt werden, um die Fehler dort auszubessern. [Goe02, S. 13 - 15],[Wag06, S. 83 - 90]

<sup>4</sup>Summe des Zeitbedarfs aller Arbeitsschritte

<sup>5</sup>Zeit zwischen Prozessstart und Prozessende

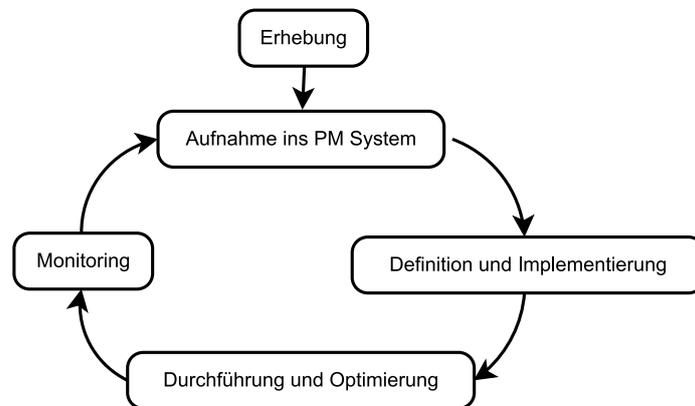


Abbildung 2.12: Prozesslebenszyklus

## 2.1.7 Visualisierung von Prozessen

Die Dokumentation zu Prozessen sollte zusätzlich zu dem Text eine grafische Repräsentation des Prozesses beinhalten. Diese ermöglicht dem Leser in kurzer Zeit, einen Überblick über den Prozess zu erlangen. Es muss hierbei die Zielgruppe der Dokumentation und der Zweck, der mit der Visualisierung erreicht werden soll, beachtet werden. Dies ist in den folgenden zwei Unterkapiteln näher beschrieben.

### 2.1.7.1 Anforderungen an die Visualisierung

Für die Visualisierung muss unterschieden werden, welche Ziele damit erreicht werden wollen. Soll die Visualisierung der Kommunikation, Dokumentation oder Information dienen, dann ist es wichtig, einen guten Mittelweg zwischen Detailgetreue und Übersichtlichkeit zu finden. Diese Art der Visualisierung wird oft für die Einschulung von Mitarbeitern (welche später diese Prozessschritte durchführen sollen) verwendet. Von besonderem Interesse sind die folgenden Punkte:

- Prozessschritte und deren Abfolge,
- Verantwortlichkeiten,
- Schnittstellen,
- Mögliche Verzweigungen und dazugehörige Entscheidungskriterien sowie
- Ein- und Ausgaben der Prozessschritte.

[Wag06, S. 108]

Handelt es sich um eine Visualisierung mit dem Ziel der Optimierung bzw. Analyse, so sollten möglichst viele Informationen in einem Diagramm abgebildet werden. Mögliche Optimierungsziele sind die sechs Dimensionen der Prozessqualität (siehe auch Kapitel 2.1.8.2):

- Prozesskunden
- Prozesswirtschaftlichkeit
- Prozessrisiko
- Prozessfähigkeit
- Prozessinformation
- Prozessorganisation

[Wag06, S. 109]

### **2.1.7.2 Visualisierungsmethoden**

Prozesse können, zur besseren Übersicht, in eine Prozesslandkarte eingetragen werden. Die Prozesslandkarte kann als höchstes Level einer Hierarchie von Prozessen gesehen werden. In ihr sind die für das Unternehmen wichtigsten Management-, Geschäfts-, Mess-, Analyse-, Verbesserungs- und unterstützenden Prozesse abgebildet. Im Unterschied zum Organigramm beinhaltet die Prozesslandkarte oft die Kunden einer Organisation.

Wie aus Abbildung 2.14 zu entnehmen ist, existieren sehr viele verschiedene Arten, Prozesse zu modellieren. Hier sollen lediglich vier, vom Autor als besonders praxistauglich empfundene Methoden, kurz vorgestellt werden. Bei allen diesen Modellierungsmethoden können mit einer Diagrammart verschiedene Sichten einer Organisation auf den aktuellen Geschäftsprozess dargestellt werden. Dies kann zur Beleuchtung des Prozesses von mehreren Seiten (um mögliche Verbesserungspotenziale besser aufzuzeigen) oder dem besseren Verständnis dienen.

#### **2.1.7.2.1 Pfeilformdarstellung**

Hierbei handelt es sich um eine stark vereinfachte Darstellung von Prozessen. Sie eignet sich, um komplexe Prozesse schnell überblicksweise zu erfassen. Diese Darstellungsart könnte innerhalb einer Prozesslandkarte oder als zweite Ebene der Visualisierung verwendet werden. Siehe Abbildung 2.15. [Wag06, S. 110]

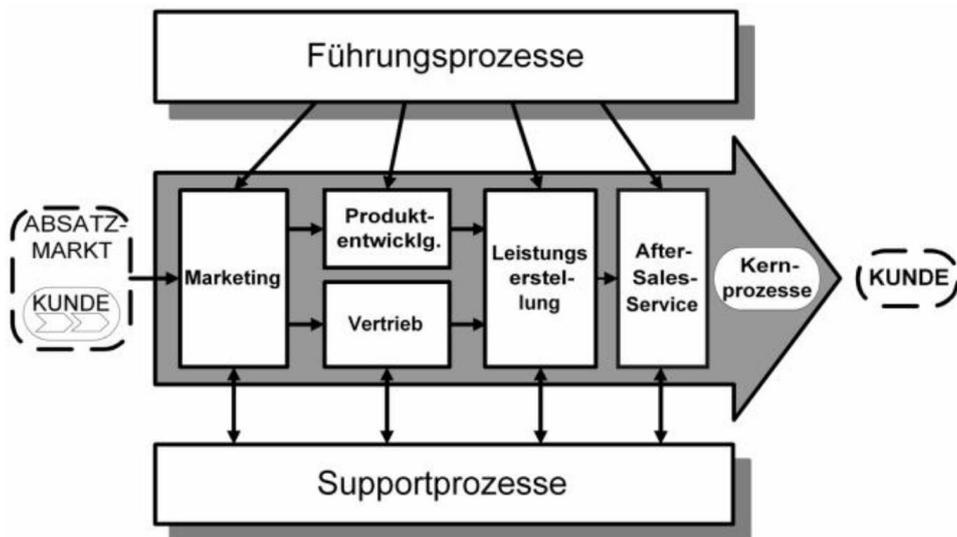


Abbildung 2.13: Prozesslandkarte [Len05, S. 33]

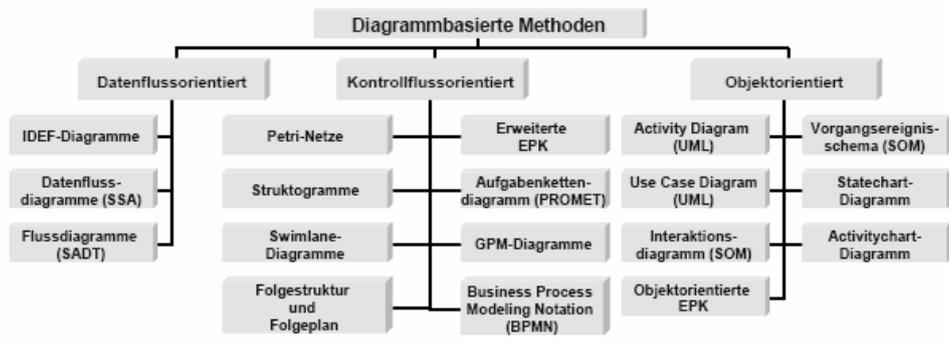


Abbildung 2.14: Übersicht über diagrammbasierte Methoden [Gad07, S. 81]

### 2.1.7.2.2 Prozessablaufdarstellung

In dieser Darstellungsart werden die einzelnen Prozessschritte auf der senkrechten Achse aufgetragen. Neben der Spalte, welche die Prozessschritte beinhaltet, lassen sich beliebig viele andere Spalten mit verschiedensten Inhalten darstellen, beispielsweise benötigte Ressourcen, Verantwortlichkeiten oder im aktuellen Schritt generierte Informationen. [Wag06, S. 110 f.]

### 2.1.7.2.3 Ereignisgesteuerte Prozesskette (EPK)

Diese Darstellungsart wurde von Scheer et al. im Rahmen des ARIS-Konzepts

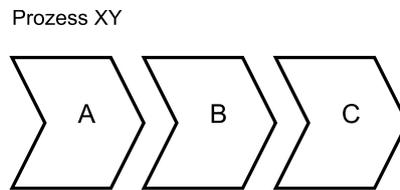


Abbildung 2.15: Pfeilformdarstellung eines Prozesses

<sup>6</sup> entwickelt und wird mittlerweile von SAP <sup>7</sup> sowie generell im Business Process Reengineering oft eingesetzt. EPKs beinhalten vier verschiedene Sprachelemente:

- Funktionen bezeichnen die in einem Geschäftsprozess zu leistenden Tätigkeiten. Funktionen können beliebig in Unter-Funktionen zerlegt und wieder zusammengefügt werden, sinnvoll ist dies allerdings nur bis zu einem gewissen Ausmaß.
- Ereignisse steuern den Ablauf in einer Organisation und müssen daher aus betriebswirtschaftlicher Sicht relevant sein.
- Organisationseinheiten dienen der Definition, in welchem Bereich eines Unternehmens eine Aufgabe durchgeführt werden muss.
- Die in Funktionen entstehenden oder benötigten Informationsobjekte werden ebenfalls in die EPK eingetragen.

[Sta06] Diese Sprachelemente werden durch verschiedene logische Operatoren miteinander verknüpft, um so einen Geschäftsprozess abzubilden.[Sta06, S. 59 - 69] Der interessierte Leser sei für eine detaillierte Beschreibung der ereignisgesteuerten Prozessketten auf [Sta06] verwiesen.

#### 2.1.7.2.4 UML Aktivitätsdiagramm

Die Unified Modeling Language (UML) wurde ursprünglich für die Modellierung von Software entwickelt, wird aber heutzutage auch in anderen Systemen eingesetzt. Die UML beinhaltet 13 verschiedene Diagrammart (sechs Struktogramme und sieben Verhaltensdiagramme), eines davon ist das Aktivitätsdiagramm. Es beschreibt einen Ablauf und beinhaltet verschiedene Knotenarten, welche durch Kontrollflüsse miteinander verbunden sind. Start und Ende eines Anwendungsfalles werden durch einen eigenen Start/Endknoten dargestellt, dazwischen kann sich eine beliebige Ablauflogik befinden. Es ist zu beachten, dass von UML Version 1 zu Version 2 im Aktivitätsdiagramm wesentliche Änderungen

<sup>6</sup>Architektur integrierter Informationssysteme

<sup>7</sup>SAP Aktiengesellschaft. <http://www.sap.com>

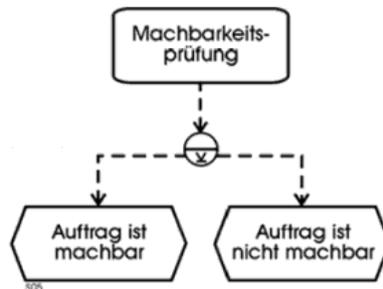


Abbildung 2.16: Ereignisgesteuerte Prozesskette aus [Sta06, S. 70]

stattfinden. Deswegen sollte beim Lesen eines UML Aktivitätsdiagrammes immer auf die Version geachtet werden. [Gad07, S. 108], [Oes06, S. 302 ff.]

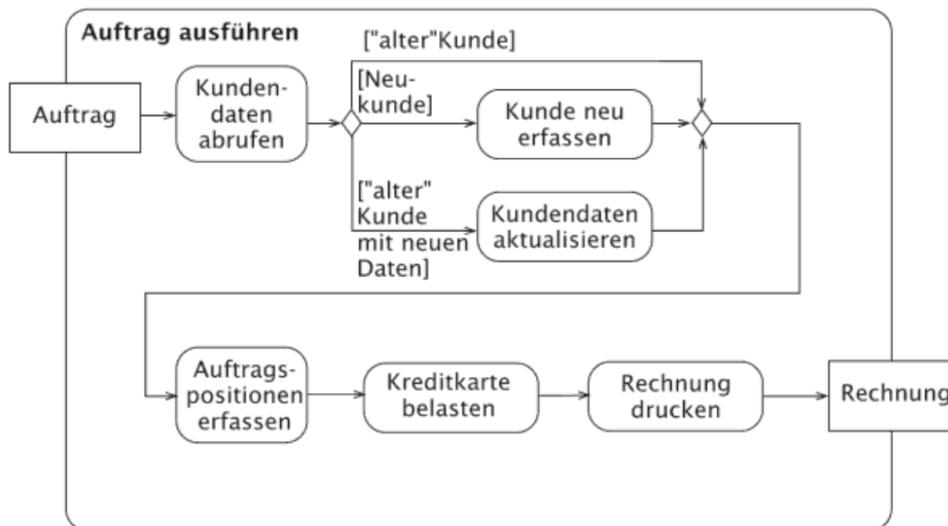


Abbildung 2.17: UML 2.1 Aktivitätsdiagramm aus [Bal08, S. 88]

### 2.1.8 Optimierung von Prozessen (Die Vier Schritte Methode)

Das Ziel der Vier Schritte Methode ist es, Prozesse zu optimieren. In einem ersten Schritt wird ein Prozess erfasst und identifiziert. Danach muss der aktuelle Zustand eines Prozesses erhoben und anschließend derart verändert werden, dass ein idealer Soll-Prozess entsteht. Im letzten Schritt werden die dafür notwendigen Verbesserungspotenziale erfasst und umgesetzt.

### 2.1.8.1 Prozessidentifikation

Laut [Hin08, S. 13] haben folgende Punkte einen Einfluss auf die Prozessidentifikation:

- Geschäftsmodell
- Geschäftsfelder
- Kundengruppen
- Kundenanforderungen
- Leistungsangebot
- Kernkompetenzen

Das Prozessteam hat die Aufgabe, die einzelnen Elemente eines Prozesses herauszufiltern, zu diskutieren und aufzuschreiben. Außerdem wird ein eindeutiger Prozessname festgelegt. Um eine einfache Abgrenzung zu anderen Prozessen zu ermöglichen, werden der erste und letzte Prozessschritt festgelegt sowie Input und Output definiert. Siehe Abbildung 2.18.

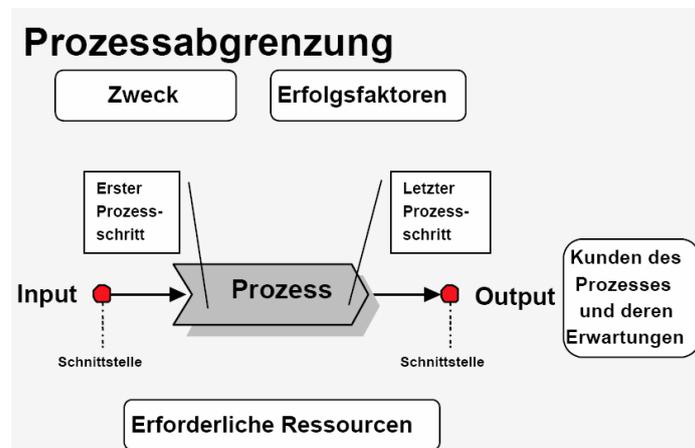


Abbildung 2.18: Prozessabgrenzung [Wag06, S. 99]

Bei Prozessen kann zwischen Kern- (Geschäftsprozessen) und Supportprozessen unterschieden werden. [Hin08, S. 20] definiert diese Begriffe folgendermaßen: Kernprozess: „Prozess, dessen Aktivitäten einen direkten Bezug zum Produkt eines Unternehmens aufweisen und damit einen Beitrag zur Wertschöpfung im Unternehmen leisten“

Supportprozess: „Prozess, dessen Aktivitäten aus Kundensicht nicht selbst wertschöpfend sind, die aber notwendig sind, um einen Kernprozess ausführen zu können“

Zu Supportprozessen zählen beispielsweise Managementprozesse. Diese dienen der strategischen Ausrichtung einer Organisation. Außerdem beinhalten Supportprozesse Mess-, Analyse- und Verbesserungsprozesse. Diese Prozesse werden für den kontinuierlichen Verbesserungsprozess benötigt. Vgl. [Wag06, S. 40-41]

Um Kernprozesse zu identifizieren, können verschiedene Kriterien herangezogen werden ([BM05, S. 128]):

- Direkter Bezug zu den Kernkompetenzen,
- Begründung eines nachhaltigen Wettbewerbsvorteils, Differenzierung von den Wettbewerbern,
- Strategisch entscheidende Bedeutung,
- Herstellung eines wahrnehmbaren Kundennutzens,
- Nicht-Imitierbarkeit durch Wettbewerber,
- Nicht-Substituierbarkeit durch andere Problemlösungen,
- Ausrichtung auf externe Leistungsempfänger, unmittelbarer Marktkontakt. Ausnahme: Ausrichtung auf andere Kernprozesse und Hinausreichen über traditionelle Organisationsgrenzen.

Ähnliche Kriterien sind in der Literatur für die Identifikation von Supportprozessen definiert ([BM05, S. 129]):

- Unterstützung der Kernprozesse, Sicherung ihrer Funktionsfähigkeit,
- Bereitstellung und Verwaltung der für die Kernprozesse erforderlichen Ressourcen,
- Ausrichtung auf interne Kunden,
- Durchführung unterstützender Aufgaben, die nicht zu einem von den Kunden wahrnehmbaren Zusatznutzen führen.

### **2.1.8.2 Ist-Analyse**

Wurde ein Prozess identifiziert, muss dieser in seinem aktuellen Zustand analysiert werden. Wichtige Zugänge zu einem Prozess sind die sechs Dimensionen der Prozessqualität. Bei der Analyse sollte für jeden Prozess abgewogen werden,

welche Dimension für den Kunden des Prozesses am wichtigsten ist, damit diese Dimension priorisiert analysiert werden kann. Dies soll nicht bedeuten, dass alle anderen Dimensionen vollständig außer Acht gelassen werden. Nach [Wag06, S. 131 - 154] gliedert sich die Prozessqualität in die folgenden Dimensionen:

**Kundensicht** Die Bedürfnisse der Kunden sind in Prozessen von hoher Wichtigkeit. Prozesse haben sowohl interne als auch externe Kunden. Beide stellen an das zur Verfügung gestellte Produkt Ansprüche, welche zufrieden gestellt werden müssen. Nur jene Organisationen, die die Erwartungen ihrer Kunden kennen und verstehen, schaffen es, eine hohe Kundenzufriedenheit zu erreichen. Diese Zufriedenheit kann durch verschiedene Methoden gemessen werden: Beispielsweise durch Umfragen direkt beim Kunden oder durch die Befragung von Mitarbeitern, welche oft Kundenkontakt haben.

**Wirtschaftlichkeitssicht** In der Wirtschaftlichkeitssicht kann beispielsweise eine Wertschöpfungsanalyse durchgeführt werden. Dabei werden die einzelnen Prozessschritte betrachtet und in drei Kategorien eingeteilt (siehe Abbildung 2.19): Erstens Arbeitsschritte (AS), die einen direkten Nutzen für den Kunden haben (*wertschöpfende Tätigkeiten*). Zweitens in Arbeitsschritte, welche zwar keine direkte Auswirkung auf einen Kunden haben, aber notwendig sind, um wertschöpfende AS durchführen zu können (*unterstützende Tätigkeiten*). Und drittens existiert eine Gruppe von AS, welche weder wertschöpfend noch unterstützend sind. Sie stellen eine *Verschwendung* von Ressourcen da und können normalerweise aus dem Prozess gestrichen werden. Ein

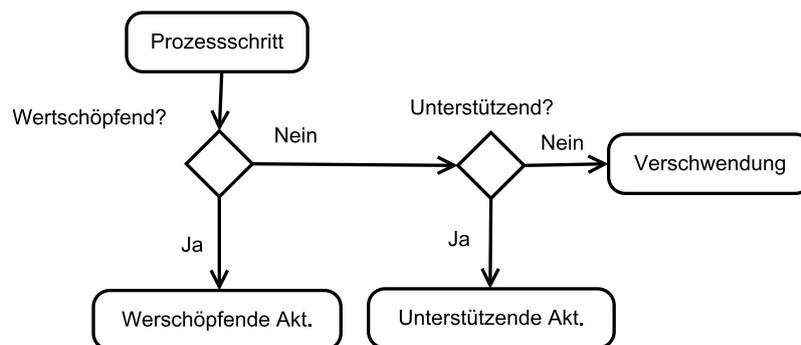


Abbildung 2.19: Vorgehen bei der Wertschöpfungsanalyse

weiterer Ansatzpunkt für eine Analyse sind die Prozesszeiten. Die Prozesszeit gibt an, wie lange an einem Prozess gearbeitet wurde. Dieser Wert ist für gewöhnlich größer als die Durchlaufzeit eines Prozesses. Der Grund dafür ist, dass bei der Durchlaufzeit parallel laufende Arbeitsschritte nicht mehrfach gezählt werden.

**Risikosicht** Durch ein Risikomanagement können frühzeitig mögliche Risiken erkannt und bewertet werden. Dadurch ist es möglich, Risiken gezielt entgegenzusteuern oder zumindest deren Kosten abzuschätzen.

**Fähigkeitssicht** Für die Analyse der Fähigkeiten eines Prozesses stehen Reifegradmodelle wie beispielsweise SPICE<sup>8</sup> zur Verfügung. SPICE definiert sechs Fähigkeitslevels, welche von „unvollständiger Prozess“ bis „optimierender Prozess“ reichen.

**Informationssicht** In der Informationssicht werden der Fluss von Informationen durch den Prozess bzw. die einzelnen Prozessschritte analysiert. Generell wird zwischen Informationen, welche beim Beginn einer Aktivität bereits vorliegen (Input) und jenen Informationen, welche nach dem Prozessschritt vorliegen (Output), unterschieden. Beim Input ist vor allem die Schnittstelle zwischen den Prozessschritten und somit die Art und Weise, wie die Information übergeben wird, von Interesse. Für den Output ist interessant, wie die Information während des Prozessschrittes generiert wird. Ein wichtiger Punkt hier ist die Qualifikation der Mitarbeiter. Denn, nur wenn diese ausreichend qualifiziert sind, kann die Information qualitativ hochwertig sein. Am Ende der Aktivität muss einerseits sichergestellt sein, dass die generierten Informationen auch jene sind, welche im nächsten Prozessschritt benötigt werden und andererseits müssen die Informationen eine Form besitzen, welche im nächsten Schritt direkt verwendet werden kann.

**Organisationssicht** In der Organisationssicht lassen sich drei Analysen durchführen. Erstens die Kompetenzanalyse. Hier wird untersucht, ob Personen, die eine Tätigkeit durchführen, auch dafür befähigt sind. Es wird für jeden Prozessschritt festgelegt, welche Qualifikation die durchführende Person besitzen muss. Fehlen benötigte Qualifikationen, müssen diese im Rahmen einer Personalentwicklung erworben werden. Zu den Qualifikationen zählen auch Entscheidungsbefugnisse.

Zweitens existiert die Schnittstellenanalyse, in welcher festgestellt wird, zwischen welchen Prozessen es überhaupt Schnittstellen gibt, welche Daten übergeben werden und auf welche Art und Weise sie übergeben werden. Es ist sinnvoll, eine Matrix (Name der Prozesse in Zeilen und Spalten) für alle wichtigen Prozesse der Organisation zu erstellen, um einerseits die vorhandenen Schnittstellen zu identifizieren und andererseits die Schnittstellen bewerten zu können.

Drittens kann eine Ablaufanalyse durchgeführt werden. Ziel dieser ist es,

---

<sup>8</sup>Software Process Improvement and Capability Determination oder ISO/IEC 15504

gleichzeitig die Prozesszeit zu reduzieren sowie die Effizienz des Prozesses zu steigern. Oft hat sich der Ist-Ablauf von Prozessen über längere Zeit und auf Grund verschiedener Rahmenbedingungen ergeben. Es gilt, den Prozessablauf optimal auf die aktuellen Rahmenbedingungen anzupassen. Hierfür stehen sieben Gestaltungsmaßnahmen für die einzelnen Prozessschritte zur Verfügung (siehe auch Abbildung 2.20):

- Weglassen,
- Zusammenlegen,
- Parallelisieren,
- Überlappen,
- Auslagern,
- Ergänzen,
- Ändern der Reihenfolge

von Prozessschritten.

Gestaltungsmaßnahmen	vorher	nachher
1. Weglassen	1 → 2 → 3 → 4	1 → 2 → <del>3</del> → 4
2. Zusammenlegen	1 → 2 → 3 → 4	1 → 2+3 → 4
3. Parallelisieren	1 → 2 → 3 → 4	1 → (2, 3) → 4
4. Überlappen	1 → 2 → 3 → 4	1 → (2, 3) → 4
5. Auslagern	1 → 2 → 3 → 4	1 → 2 → 3 → 4
6. Ergänzen	1 → 2 → 3	1 → 2 → 3 → 4
7. Ändern der Reihenfolge	1 → 2 → 3 → 4	1 → 3 → 2 → 4

Abbildung 2.20: Gestaltungsmaßnahmen der Ablaufanalyse [WP07, S. 154]

### 2.1.8.3 Soll-Konzeption

Nach der Ist-Analyse gilt es, aufgrund der durchgeführten Analysen, Verbesserungspotenziale zu identifizieren. Diese orientieren sich meist an den Prozesszielen. Bei Prozesszielen ist zu beachten, dass sie S.M.A.R.T. sein sollten ([Fro09]):

**Spezifisch** Detaillierte Auflistung was zu tun ist, Elimination jeglicher Unklarheiten.

**Messbar** Der Weg zum Ziel muss gemessen werden können. An jedem Punkt muss eine Differenz zum Ziel feststellbar sein.

**Angemessen** Mit den aktuell zur Verfügung stehenden Mitteln muss das Ziel erreichbar sein.

**Realistisch** Es muss realistisch sein, dass das Ziel erreicht werden kann.

**Terminiert** Jedes Ziel muss mit einem Zieldatum versehen werden, bis zu dem es erfüllt werden muss.

Prozessziele sollten gemeinsam vom Prozessteam festgelegt werden. Unter Einbeziehung der Personen, welche zukünftig den Prozess durchführen werden, und mit Hilfe der durchgeführten Analyse sowie der Visualisierungsmethoden (Kapitel 2.1.7.2) gilt es nun, den idealen Soll-Prozess zu definieren. Außerdem sollte das Prozessteam eine Prozessdokumentation erstellen. Diese beinhaltet eine textuelle Beschreibung des Prozesses sowie ein Ablaufdiagramm [Blo03, S. 27]. Je nach Organisation kann die Prozessbeschreibung unterschiedlich umfangreich sein. Wichtig ist vor allem, dass die Prozessbeschreibung für die Mitarbeiter leicht verständlich ist. Für das Prozesscontrolling ist es notwendig, Kennzahlen und somit Messgrößen für Prozesse festzulegen ([Erd00, S. 59]). Es gibt zwei Arten von Prozessmessgrößen: *Führungsmessgrößen* (werden aus übergeordneten Zielen abgeleitet, z.B.: Durchlaufzeit) und *Einflussmessgrößen* (werden aus der systemischen Unternehmensbetrachtung abgeleitet, z.B.: Qualifikation), wobei ersteren die größere Bedeutung zukommt ([Len05, S. 25]). Eine Messgröße ist nur dann sinnvoll, wenn die Mitarbeiter sie auch verstehen können. Es ist von Vorteil, für jeden Prozess mehrere aussagekräftige Messgrößen zu definieren, da diese Veränderungen anzeigen können und im weiteren Verlauf eine Aussage darüber ermöglichen, ob der Prozess besser oder schlechter als zuvor ausgeführt wird. Um die Erreichung der Prozessziele quantifizierbar zu machen, werden oft eine Messgröße/Kennzahl, ein Zielwert, die Messmethode, ein Verantwortlicher sowie die Frequenz der Messung festgelegt.

#### **2.1.8.4 Umsetzung der Verbesserungspotenziale**

Nachdem eine Freigabe des Soll-Konzepts eines Prozesses erteilt wurde, gilt es, die Umsetzung des Prozesses zu planen. In der Regel sollte das oberste Management einer Organisation die Soll-Prozesse und deren Implementierung freigeben. Sind bei einem Prozess große Änderungen durchzuführen, so empfiehlt es

sich, die Änderungen in mehreren Schritten umzusetzen ([WP07, S. 169]): Zuerst in einem *dry run*, wobei die einzelnen Prozessschritte mit dem Prozessteam durchgesprochen werden, um mögliche Schwachstellen zu identifizieren. Danach erfolgt der *wet run*, in welchem innerhalb eines definierten Zeitraums und eines bestimmten Bereichs der Organisation der Soll-Prozess umgesetzt und getestet wird. Jegliche Abweichungen vom Soll-Prozess sollten dokumentiert und im Anschluss mit dem Prozessteam besprochen werden. Danach erfolgt die *Installation* des Soll-Prozesses, bei der der Prozess tatsächlich in Betrieb genommen wird.

Außerdem schlägt [WP07, S. 169] vor, eine Prozessbegehung durchzuführen. Er stellt die These auf, dass jede Veränderung eines Prozesses eine spezielle Herausforderung für alle an dem Prozess beteiligten Personen darstellt. Als Grund dafür gibt er an, dass neu definierte Prozesse in der Theorie zwar sehr gut aussehen können, die Praxis der theoretischen Einschätzung aber widerspricht. Bei der Prozessbegehung wird der Prozess gemeinsam mit den Mitarbeitern in der realen Umgebung ausgeführt und es werden die Unterschiede zum alten Prozess genau erläutert.

Um sicherzustellen, dass die Aufgaben für die Umsetzung des Soll-Prozesses überhaupt erledigt werden, ist es von Vorteil, TODO-Listen zu erstellen, welche den Arbeitsauftrag, einen Termin für diesen und die Verantwortlichkeiten beinhalten. Mit Hilfe dieser Listen kann im Nachhinein überprüft werden, welche Aufgaben tatsächlich durchgeführt wurden.

Bei der Umsetzung der Verbesserungspotentiale durchlaufen die Mitarbeiter aufgrund der veränderten Rahmenbedingungen und Bedürfnisse sieben Phasen (Abbildung 2.21). Diese sollten dem Chief Process Officer und dem Projektleiter bekannt sein, um Spannungen im Vorfeld zu vermeiden und die Zeit bis zur Integration zu verkürzen. In der ersten Phase (Schock) werden die eigenen Kompetenzen mit den geänderten Bedürfnissen verglichen und es wird festgestellt, dass diese sich nicht decken. Die subjektive Wahrnehmung der eigenen Kompetenz sinkt. Danach, in der Phase der Ablehnung, steigt die wahrgenommene eigene Kompetenz wieder, da es scheint, als müssten die eigenen Handlungsweisen nicht verändert werden, um die veränderten Bedürfnisse abzudecken. Während der Phase der rationalen Einsicht wird erkannt, dass es doch notwendig und unumgänglich ist, die eigenen Handlungsweisen zu verändern. Darauf folgt die Phase der emotionalen Akzeptanz. Hier wird das „Tal der Tränen“, der Punkt, an jenem die subjektiv wahrgenommene eigene Kompetenz am niedrigsten ist, erreicht. Es ist hier besonders wichtig, die Mitarbeiter zu motivieren. Daraus können unentdeckte Potenziale erschlossen werden. Schlägt dies jedoch fehl, so wird der Veränderungsprozess verlangsamt oder im schlimmsten Fall gestoppt. Ist diese Phase überwunden, so folgt die Phase des Lernens. Hier werden die neuen Verhaltensweisen getestet

und geübt. Durch die dadurch entstehenden Erfolge und Misserfolge schwankt die wahrgenommene Kompetenz stetig. Scheitert der Mitarbeiter in dieser Phase, so kann dies einen Sprung zurück in die zweite Phase zur Folge haben. In der sechsten Phase (Erkenntnis) werden neues Wissen und neue Fähigkeiten auf die Aufgaben angewendet, wodurch diese beherrschbar werden. Die Flexibilität im eigenen Verhalten steigt und dadurch auch die wahrgenommene eigene Kompetenz. Diese steigt sogar über das Niveau vor der Veränderung hinaus. Darauf folgt eine völlige Integration und Verselbstständlichung der Verhaltensmuster in der siebten Phase (Integration). [Mar08, S. 8 - 9], [KM06, S. 10 - 12]

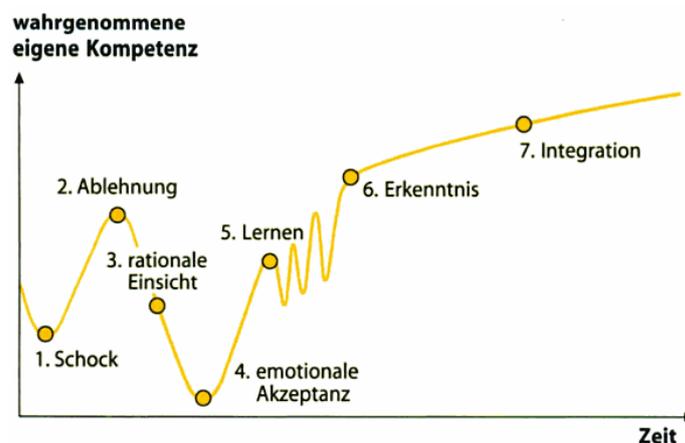


Abbildung 2.21: Phasenmodell von Veränderungsprozessen [KM06, S. 11]

## 2.2 IT Infrastructure Library

ITIL wurde Mitte der 80er Jahre in England entwickelt und seitdem stetig verbessert. Aktuell ist Version 3 (ITIL 3). Sie dient der Planung, Steuerung und Überwachung von IT-Leistungen. Mittlerweile hat es sich als Defacto-Standard für IT Dienstleister etabliert [ZHB05, S. 19]. In ITIL wird die IT lediglich als Unterstützer der Kernprozesse und ohne Selbstzweck gesehen [Gol06, S. 10]. Im Unterschied zu den ISO Normen (z.B. ISO 9000) ist ITIL keine Norm, sondern eher ein Best-Practice Leitfadens. Das bedeutet, ITIL gibt vor, „Was“ zu tun ist, aber macht keine Aussagen darüber, „Wie“ etwas zu geschehen hat [Olb08, S. 1]. In ITIL spielt die Kundenorientierung eine wichtige Rolle, sie wird als IT Service Management verstanden. Wenn sich eine Organisation entschließt, ITIL einzusetzen, so bedeutet dies nicht, dass die gesamte Organisation neu gestaltet werden muss. Meistens sind schon viele Teile bereits umgesetzt und es muss nur noch optimiert und ergänzt werden. Unter Service (Dienst) versteht ITIL „ein oder mehrere

IT-Systeme, welche einen Geschäftsprozess ermöglichen“ ([Bre07, S. 16]).

Neben ITIL existieren für IT Service Management auch noch andere Referenzmodelle, welche sich vor allem durch ihre unterschiedlichen Schwerpunkte unterscheiden, beispielsweise Prince II <sup>9</sup> ([Köh06]) oder CobIT <sup>10</sup> ([BB04]).

ITIL umfasst fünf Kapitel, welche als eigenständige Bücher publiziert werden:

- Servicestrategie (Service Strategy - SS)
- Serviceentwurf (Service Design - SD)
- Serviceüberführung (Service Transition - ST)
- Servicebetrieb (Service Operation - SO)
- Kontinuierliche Serviceverbesserung (Continual Service Improvement - CSI)

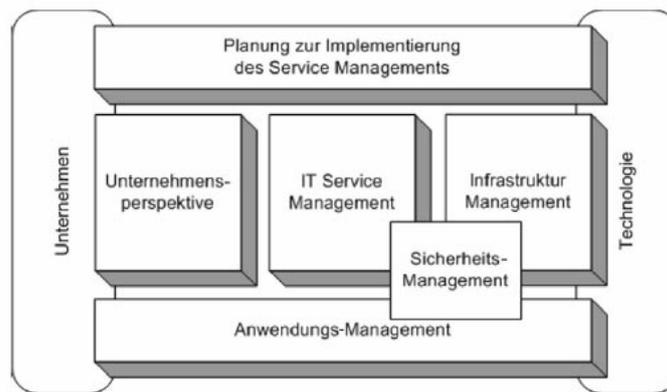


Abbildung 2.22: Die ITIL Rahmenstruktur [Olb08]

### 2.2.1 IT Service Management

ITIL definiert das IT Service Management folgendermaßen: „*Service Management ist eine Reihe von spezialisierten, organisationalen Fähigkeiten zur Stiftung von Wert für Kunden in Form von Services*“ [VB08, S. 17]. Wichtig ist hier vor allem die Orientierung an den Geschäftsprozessen der Kunden, denn ohne Kenntnis dieser Prozesse ist es nur schwer möglich, angepasste IT Services zu entwickeln und in einer bestimmten Qualität zu betreiben. Ein weiterer ausschlaggebender Punkt ist die Orientierung an den Bedürfnissen der Kunden. Vor allem wenn ein

<sup>9</sup>Prince: Projects in Controlled Environments

<sup>10</sup>CobIT: Control Objectives for Information and Related Technology

Service-Desk betrieben wird, ist es notwendig, dass benutzerorientiert gearbeitet wird. Für die Akzeptanz dieses Services ist die individuelle Wahrnehmung der Kunden entscheidend. IT Service Management versucht, die Qualität als auch die Quantität von IT Services zu planen, zu überwachen und zu steuern. Hierfür existieren verschiedene Kriterien, wonach IT Service Management gestaltet werden muss:

- Durch Benutzerfreundlichkeit muss neben den hochwertigen Services die Benutzerakzeptanz hoch gehalten werden.
- IT Services müssen geschäftsorientiert sein, sie müssen also die Geschäftsprozesse der Kunden bestmöglich unterstützen.
- Um den Fortbestand der eigenen Organisation zu sichern, müssen IT Services auch wirtschaftlich sein. Idealerweise werden die Ziele mit möglichst wenig Aufwand erreicht.
- Die IT Services müssen sich an zuvor definierten Zielen ausrichten, deren Grad der Erreichung kontinuierlich gemessen wird.

Vgl [Bei08, S. 1-3]

### **2.2.2 Servicestrategie**

„Es werden hier Empfehlungen gegeben, um Service Management Richtlinien (Policies), Strukturen, Verfahren und Prozesse für den gesamten ITIL Service Lebenszyklus zu erstellen und diese auf bestehende und künftige Service Management Prozesse anzuwenden“ [Olb08, S. 146]. Das Ausfühungen des Buch richten sich dabei eher an die Geschäftsführung, welche sich überlegen sollte, warum die Services auf eine bestimmte Art und Weise gestaltet werden sollten. Risiken sollen erkannt und berücksichtigt werden. Zusätzlich zu dieser Überlegung wird in ITIL 3 berücksichtigt, wer ein Service erbringt. Es erfolgt hier eine Einteilung in drei Arten von Providern: Der *Internal Service Provider* ist auf die eigene Organisation fokussiert und hat direkten Einfluss auf das Hauptgeschäft, wohingegen das *Shared Services Unit* an verschiedene Geschäftsbereiche der Organisation liefert. Der dritte Provider ist der *External Service Provider*, welcher auf Grund von Verträgen seine Services liefert. Mit einer guten Servicestrategie sollte es möglich sein, dass neue Geschäftsfelder erkannt werden und somit die Marktposition ausgebaut und gefestigt wird. Vgl. [Bei08, S. 23 f.] und [Olb08, S. 146].

### **2.2.3 Serviceentwurf**

Das Buch „Serviceentwurf“ beschäftigt sich einerseits mit der Entwicklung neuer IT Services als auch großen Veränderungen bereits bestehender Services. In die

Serviceentwicklung wird das Wissen über Geschäftsprozesse des Kunden integriert. Vgl. [Uni08, S. 40] Die zu entwickelnden Services sind für die Produktivumgebung des Kunden bestimmt, wo mit deren Hilfe Geschäftsziele erfüllt werden sollen. Ziel des Serviceentwurfs ist es, sowohl bestehende als auch neue Anforderungen abzudecken. Dazu sollen diese unter anderem Architektur, Dokumentation und Richtlinien enthalten. Die Qualität soll nicht länger dem Zufall überlassen werden. Durch gezielte Planung sollen die Qualität gesteigert und gleichzeitig die Kosten gesenkt werden. Zum Serviceentwurf gehören u.a. die Prozesse

- Capacity Management, welches sich damit beschäftigt, dass genügend Kapazitäten, die Services betreffend, vorhanden sind.
- Service Catalogue Management: Hierbei handelt es sich um einen Katalog von Leistungen, welche dem Kunden angeboten werden können.

Vgl. [Ebe08, S. 185 ff.]

#### **2.2.4 Serviceüberführung**

Weiterführend zum Serviceentwurf wird in diesem Buch die Übergabe der Services an den Kunden vorbereitet. Serviceüberführung zielt darauf ab, das Risiko für Fehler bei der Einführung eines neuen oder stark veränderten Services zu minimieren und gleichzeitig Möglichkeiten zu bieten, wie Services im Alltag optimiert werden können. Einerseits müssen Services stabil funktionieren, um damit Kundenanforderungen zu erfüllen und die Kundenzufriedenheit hoch zu halten, andererseits muss es möglich sein, notwendige Veränderungen (Weiterentwicklungen) in die Services einzubauen. Teilprozesse der Serviceüberführung sind

- Change Management: Dieser Prozess sorgt dafür, dass die Veränderung (Change) auf eine standardisierte Art und Weise geschieht.
- Release and Deployment hat die Aufgabe, für die im Serviceentwurf entwickelten Services Releases zu konzipieren, diese zu testen und sicher zu stellen, dass die dem Kunden zugesicherte Qualität gehalten werden kann [Uni08, S. 44].
- Knowledge Management ist ein Prozess, der dafür sorgt, dass das in einer Organisation vorhandene Wissen jenen Personen zur Verfügung steht, welche es gerade benötigen.

Vgl [Bei08, S. 89 f.] und [Olb08, S. 151]

### 2.2.5 Servicebetrieb

In diesem Buch sind Kennzahlen für operative Steuerungszwecke (Überwachung von Prozessen und Schnittstellen) berücksichtigt. [GKS08, S. 130]

Der Servicebetrieb befasst sich mit dem Management der vereinbarten Services, um deren Nutzen tatsächlich zu realisieren.

- **Event Management:** Das Event Management beschäftigt sich damit, den Status von Services zu überwachen und basierend auf diesem Status Informationsmeldungen auszusenden. Bei einem Fehler-Status können automatisch Gegenmaßnahmen eingeleitet oder zumindest eine zuständige Person verständigt werden.
- **Incident Management:** Hier geht es vorwiegend darum, gestörte Services möglichst schnell wieder in Gang zu setzen, um den Geschäftsbetrieb so schnell wie möglich wieder aufnehmen zu können bzw. eine Beeinträchtigung des Geschäftsbetriebs zu verhindern.
- **Das Prozess Access Management** regelt, welche Benutzer und Gruppen auf welche Services Zugriff haben. Der Prozess schließt außerdem eine Autorisierung<sup>11</sup> der Benutzer/Gruppen ein.

Vgl. [Uni08, S. 46], [Bei08, S. 127 f.], [Olb08, S. 154]

### 2.2.6 Kontinuierliche Serviceverbesserung

Mit Hilfe der kontinuierlichen Serviceverbesserung sollen Effizienz und Effektivität der Prozesse sowie der Service Erbringung kontinuierlich gemessen und dargestellt werden. Daraus gilt es, Verbesserungspotenziale abzuleiten. In der Praxis wird die kontinuierliche Serviceverbesserung oft erst nach der Entstehung bedeutender Probleme etabliert und gerät danach rasch wieder in Vergessenheit.

- **Der Prozess Service Reporting** übernimmt das Management der Berichte und der zugrunde liegenden Daten. Die Daten werden aufbereitet und in übersichtlicher Art und Weise dargestellt. Es wird in diesem Prozess entschieden, welche Personen welche Daten zu welchen Zeiträumen abfragen können.
- **Der Verbesserungsprozess:** Er besteht aus sieben Schritten, welche auf die taktischen, operativen und strategischen Ziele in der Organisation ausgerichtet werden müssen.

Vgl. [Olb08, S. 156 - 157].

---

<sup>11</sup>Das Überprüfen und Zuweisen von Rechten

## **Kapitel 3**

# **Beschreibung der Organisation des Fallbeispiels**

Im Verlauf des Kapitels wird die betrachtete Organisation etwas näher beschrieben. Dies soll dazu dienen, dass der Leser einen besseren Einblick in die Probleme und auch möglichen Potenziale des Unternehmens gewinnen kann.

### **3.1 Szenario allgemein**

Die betrachtete Organisation ist im Bereich Medizintechnik tätig. Von der Organisation werden sowohl medizintechnische Geräte als auch Software für die Verwendung in Krankenhäusern vertrieben. Dies geschieht von mehreren Standorten in Österreich aus. Aktuell sind in der Organisation 84 Mitarbeiter beschäftigt. Die Organisation erzielte im Jahr 2008 einen Umsatz von ca. 22 Mio. €.

Im Folgenden, sowie in der gesamten Masterarbeit, wird lediglich der IT Bereich der Organisation betrachtet. Die Organisationsform basiert auf dem Einliniensystem (2.1.5.1) mit funktionaler Struktur (2.1.5.2). Die IT Abteilung ist in zwei Gruppen aufgeteilt: Projekt- und Technikteam. Jedes Team wird von einem Bereichsleiter geführt. Für jedes Softwareprodukt existiert jeweils ein Produktverantwortlicher aus dem Projekt- sowie dem Technikteam. Im Rahmen der vorliegenden Arbeit bestand die Möglichkeit, im Technikteam mitzuarbeiten und direkt vor Ort wichtige Erfahrungen für die Erstellung dieser Arbeit zu sammeln.

### **3.2 Produkte**

Die vom IT Bereich der Organisation ausgeschriebenen Produkte sind ausschließlich Softwareprodukte von Drittanbietern, welche von der Organisation für einen Kunden konfiguriert und installiert werden. Zusätzlich zu diesen Produkten haben

die bestehenden Kunden die Möglichkeit, Erweiterungen zu der Drittanbietersoftware zu erwerben.

### **3.3 Struktur**

Der IT Bereich der betrachteten Organisation besteht aus drei Teams, welche zwar eigenständig arbeiten, aber von einer guten Zusammenarbeit untereinander abhängig sind.

#### **3.3.1 Projektteam**

Das Projektteam beschäftigt sich hauptsächlich mit den Bedürfnissen der Kunden. Es ist der erste Ansprechpartner, wenn Kunden Erweiterungen zu Produkten bestellen möchten oder falls ein Fehler im Softwaresystem auftritt. Während einer Ausschreibung werden vom Projektteam Demostellungen bei den möglicherweise zukünftigen Klienten organisiert, um das eigene System zu präsentieren. Nach Auftragszuschlag nehmen die Mitglieder des Projektteams als Erste Kontakt mit dem neuen Kunden auf. Danach folgen Bestandsaufnahmen beim Auftraggeber sowie die Erstellung eines Projektplans. Das gesamte Projektmanagement, welches die Abstimmung mit den anderen Teams inkludiert, wird vom Projektteam übernommen. Während der Umsetzung übernimmt das Projektteam die Konfiguration der vom Kunden bestellten Leistungen. Diese erfolgt oft direkt beim Kunden. Nach der Konfiguration und der Installation, welche auch vom Technikteam durchgeführt wird, folgen die Schulungen für alle Nutzer des neuen Systems sowie ein Inbetriebnahmesupport in den ersten Wochen, in denen das neue Softwareprodukt vom Kunden eingesetzt wird. Weiters ist es die Aufgabe des Projektteams, beim Kunden dafür zu sorgen, dass alle notwendigen Rahmenbedingungen für eine Integration des erworbenen Systems in die Systeme des Kunden erfüllt werden. Dazu zählt zum Beispiel die Definition von Standards.

#### **3.3.2 Technikteam**

Die Aufgaben des Technikteams sind breit gestreut. Sie reichen vom Instandhalten der organisationsinternen IT Infrastruktur (siehe auch Supportprozesse in Kapitel 2.1.8.1) bis zum Second-Level Support für den Kunden. Dieses Team trägt Sorge, dass die Hard- und Software der gesamten Organisation einwandfrei funktioniert. Dazu zählen jegliche Notebooks und Arbeitsplatzrechner der eigenen Kollegen sowie auch die Server. Besonders wichtig sind jene Server, welche für eine Verbindung der Organisation zu ihren Kunden benötigt werden. Diese müssen rund um die Uhr voll funktionsfähig gehalten werden, da tagsüber das Technik- sowie

das Projektteam Zugriff auf die Server und Clients beim Kunden benötigen und nachts vollautomatisch Backups von Kundenkonfigurationen erstellt werden. Weitere Aufgabengebiete des Technikteams sind das Event- und Incident Management, welche bereits in Kapitel 2.2.5 erklärt wurden. Auch Kapitel 3.4 beschäftigt sich mit diesem Thema. Zudem übernimmt das Technikteam den First-Level Support für einige der eigenen Produkte. Der Kunde hat die Möglichkeit, rund um die Uhr Hilfe anzufordern. Muss bei einem Auftraggeber ein Softwareupdate durchgeführt werden, so wird vom Technikteam ein Wartungsfenster beim Kunden vereinbart sowie das Update durchgeführt. Des Weiteren ist das Technikteam für Systemschnittstellen zwischen den eigenen und den Softwareprodukten beim Kunden zuständig.

**Softwareentwicklungsteam** Dieses Team beschäftigt sich einerseits mit der Entwicklung von Software für die Kunden, andererseits werden auch Softwareprodukte erstellt, welche für interne Zwecke gedacht sind. Software für den Kunden basiert meistens auf der Drittanbietersoftware, welche zum Verkauf ausgeschrieben wird und erweitert diese um neue Funktionen (zum Beispiel Schnittstellen oder Weboberflächen für Produkte, die nicht webbasierend sind). Die Software, welche für die eigene Organisation entwickelt wird, soll den Ablauf der Tätigkeiten in der Organisation unterstützen. Siehe 3.4.

### 3.4 Eingesetzte Software

Organisationsintern werden zu Organisationszwecken momentan zwei selbst entwickelte Softwareprodukte eingesetzt. Das erste Softwareprodukt dient der Projekt- und Tätigkeitsdokumentation (Zeitdokumentationssystem - ZDS). In dieser PHP-basierten Software werden alle laufenden Projekte der Organisation angelegt. Jedes Projekt kann beliebig viele Subprojekte besitzen, welche wiederum beliebig viele Tätigkeitsarten zugeteilt haben können. Unter Tätigkeitsarten sind Vorgänge wie Besprechungen, Softwareentwicklung, Schulungen etc. zu verstehen. Weiters kann bei jeder Tätigkeit der Durchführungsort (organisationsintern oder beim Kunden) angegeben werden. Diese Software ermöglicht genaue Auswertungen, wie viele Ressourcen in welche Projekte eingeflossen sind. Diese Daten dienen immer der Planung von neuen Projekten. Weiters unterstützt diese Software ein einfaches Call Tracking, welches für die Kundenkontakte wichtig ist.

Die zweite Software ist ein intranetbasiertes Programm für das Event Management (Kap. 2.2.5). In einer Liste kann der Status von verschiedensten Diensten, welche extern auf Kundenservern laufen, überwacht werden. Im Fehlerfall werden die für das jeweilige Produkt zuständigen Mitarbeiter informiert.

## **Kapitel 4**

# **Optimierungsbedürfnisse der Organisation - Problemstellung**

Aufbauend auf der Problemstellung aus Kapitel 1.1 wird hier detailliert auf die Optimierungsbedürfnisse der Organisation eingegangen.

### **4.1 Einschulung neuer Mitarbeiter**

Werden weitere Arbeitskräfte benötigt, so wird diese Ressource beim jeweiligen Team/Abteilungsleiter angefordert. Ist ein neuer Mitarbeiter aufgenommen, wird dieser von seinen Kollegen im jeweiligen Team eingeschult. Dieses Vorgehen hat sowohl Vor- als auch Nachteile. Positiv daran ist, dass der neue Mitarbeiter von der langjährigen Arbeitserfahrung des schulenden Kollegen profitieren kann. Negativ ist aber, dass der neue Mitarbeiter zwangsweise die Arbeitsmethodik des einschulenden Kollegen übernimmt. Außerdem ist es oft so, dass die Personen, die den neuen Mitarbeiter einschulen sollten, unter starkem Zeitdruck stehen und somit die Einschulung schnell erledigen, um mit ihrer Arbeit fortfahren zu können.

In der Organisation existieren für verschiedene Prozesse und die darin eingebetteten Arbeitsschritte sehr wenig Dokumentation/Richtlinien, wie diese zu erfolgen haben. Lediglich der Projektplan kann als grobes Ablaufkonzept herangezogen werden. Somit haben sich im Laufe der Zeit bei den Mitarbeitern jeweils andere Vorgangsweisen bei der Durchführung eines Prozesses ergeben. Aufgrund fehlender Dokumentation geht bei Auflösung eines Vertragsverhältnisses (Pensionsantritt, Ausscheiden) wertvolles Wissen der Organisation verloren. Der Verlust bereits errungenen Wissens und gesammelter Erfahrung wäre geringer, wenn die erfahrenen Mitarbeiter ihr Prozesswissen dokumentierten, so dass vor allem neue Mitarbeiter von den Aufzeichnungen profitieren könnten.

## **4.2 Verteilte Organisation**

Die Organisation ist auf mehrere Standorte innerhalb Österreichs verteilt. Daraus ergibt sich, dass die Kommunikation unter Kollegen schwieriger ist. Viele Gespräche können nicht persönlich durchgeführt werden, sondern müssen per Email oder Telefon erfolgen. Dies verringert nicht nur die sozialen Kontakte zwischen den Mitarbeitern, sondern erhöht beispielsweise auch den Zeitaufwand zur Abklärung, ob ein bestimmter Prozessschritt bereits abgeschlossen ist oder nicht.

## **4.3 Projektstatus**

Zurzeit ist es für das Management der Organisation nicht einfach, den Status für die vielen verschiedenen laufenden Projekte sowie Prozesse zu überblicken. Es gibt zwar für alle Projekte einen Projektplan, dieser ist aber oft unvollständig bzw. nicht auf dem aktuellen Stand. Aus diesen Gründen ist die einzige Möglichkeit, um den Projektstatus abzufragen, direkt den Projektverantwortlichen anzusprechen. Für viele Projekte ist das ein beträchtlicher Aufwand. Teilweise kann aus der Software, welche für die Tätigkeitsdokumentation verwendet wird, sofern dort richtig dokumentiert wird, ausgelesen werden, wie viel Zeit bereits in ein Projekt geflossen ist. Daraus lässt sich aber nicht schnell abschätzen, welchen Status ein Projekt hat bzw. wie viele Tage ein Projekt noch benötigen wird. Oft ist es für das Management einfach nur interessant zu wissen, um wie viele Tage ein Projekt verspätet ist oder wie die Performance bei einem Prozess ist. Diese Information kann nicht auf schnelle Art und Weise generiert werden.

## **4.4 Unklare Richtlinien**

Richtlinien werden in der Organisation nicht zentral abgelegt, oftmals per Email verschickt oder mündlich weitergegeben. Das hat zur Folge, dass die Mitarbeiter häufig nicht wissen, welche Richtlinie für einen Vorgang nun aktuell ist bzw. wo diese genau abgelegt ist.

## **4.5 Nicht einheitliche Dokumentenablage**

Für jedes Projekt gibt es beim Kunden Ansprechpartner. Die Ansprechpartner der im Fallbeispiel beschriebenen Organisation sind technische Verantwortliche, Oberschwester auf Stationen, Institutsvorstände, KeyUser<sup>12</sup> etc. Die Kontaktda-

---

<sup>12</sup>Anwender, welche speziell geschult werden, um Teile der Konfiguration der Produkte selbst durchführen zu können.

ten werden meistens von jedem Mitarbeiter in einem eigenen Adressbuch oder in verschiedensten Listen verspeichert. Zusätzlich werden für jeden Kunden Daten abgespeichert, wie die Kundenkonfigurationen aufgebaut sind. Auch diese Daten werden in eigenen Listen abgelegt. Der Speicherort dieser Listen ist oft nur dem jeweiligen Team bekannt und für die einzelnen Projekte/Softwareprodukte nicht einheitlich festgelegt. Klarerweise ist es für die Mitarbeiter deswegen nicht einfach, für ein Projekt oder einen Kunden alle Ansprechpartner bzw. Dokumentation ausfindig zu machen.

#### **4.6 Ansprechpartner für die Kunden**

Oft sind dem Kunden mehrere Stellen innerhalb der Organisation bekannt, jedoch kennt er nur selten, den zuständigen Ansprechpartner für seine Anfrage. Mögliche Ansprechpartner wären der Projektleiter oder Personen aus dem Technikteam. Dadurch kommen Anfragen oft nur über Umwege zu dem zuständigen Betreuer. Um dem Kunden einerseits die Sicherheit zu geben, dass seine Anfrage zur zuständigen Person weitergeleitet wird und andererseits sicherzustellen, dass Anfragen standardisiert bearbeitet werden, sollte es für jeden Kunden eine fixe Ansprechperson bei der Organisation geben. Zurzeit ist dies nicht der Fall.

#### **4.7 Rechtliche Verbindlichkeiten**

Arbeitsabläufe sind mitunter sehr umfangreich. Sie beinhalten oft Tätigkeiten, welche für die durchführende Person keinen Nutzen darstellen, allerdings durchgeführt werden müssen. Ein Beispiel hierfür ist die rechtliche Verbindlichkeit zur Dokumentation der Zeiten, während ein Mitarbeiter mittels Fernwartungssoftware mit einem Kunden verbunden hat. Diese Dokumentation ermöglicht es, im Fehlerfall (beim Kunden) festzustellen, ob zum Fehlerzeitpunkt Wartungsarbeiten am Kundensystem durchgeführt wurden und ob diese Wartungsarbeiten mit dem Fehler in Zusammenhang stehen. Zurzeit existiert in der Organisation keine Möglichkeit, die Mitarbeiter zum richtigen Zeitpunkt an die Durchführung der notwendigen Arbeitsschritte zu erinnern. Der Einsatz eines WMS und die Definition entsprechender Prozesse könnten die Anwender dabei unterstützen.

#### **4.8 Standardisiertes Arbeiten**

Abhängig vom Grad der Erfahrung, der Häufigkeit der Durchführung eines Arbeitsablaufes und der Art, wie der Mitarbeiter eingeschult wurde, werden Arbeitsabläufe im Vergleich zu anderen Mitarbeitern oft sehr unterschiedlich durch-

geführt. Dadurch entstehen oft Unterschiede in der Qualität der Arbeitsausführung, was vom Management nicht erwünscht ist. Daher sollte die Grobstruktur der Arbeitsabläufe standardisiert durchgeführt werden. Dies stellt eine Aufgabenstellung der vorliegenden Arbeit dar, siehe Kapitel 5.1 und 6.

## **Kapitel 5**

# **Methodik des Erfassens der Prozesse und des Entwurfs des WMS**

In diesem Kapitel wird sowohl das Erfassen der Prozesse als auch der Entwurf des Workflow-Management-Systems beschrieben. Für beide Teilgebiete wird erläutert, wie die Methodik auf den Kontext der Organisation angewendet werden kann.

### **5.1 Erfassen der Prozesse/Workflows**

Zum Zwecke der Erfassung der Prozesse und Workflows wurde anhand der Vier-Schritte-Methode (Kapitel 2.1.8) vorgegangen. Mit Hilfe einer iterativen Vorgehensweise wurden die einzelnen Prozesse von dem Projektleiter, dem Prozessteam (siehe Kapitel 2.1.3) sowie den Bereichsleitern identifiziert und anschließend detailliert definiert. Dieser Vorgang erfolgte in mehreren Prozessteam Meetings (PTM, Ablauf in Anlehnung an [Wag06]).

#### **5.1.1 Prozessteam Meeting 1**

Zu Beginn erfolgte eine kurze Vorstellung des gesamten Projektes durch den Projektleiter, woraufhin die eingesetzte Prozessmanagement Methodik erklärt wurde, um alle Teilnehmer der Besprechung auf annähernd denselben Wissensstand (bezogen auf das Prozessmanagement) zu bringen. Im nächsten Schritt wurde vom Projektleiter der Sinn der Prozesslandkarte erläutert. Darauf Bezug nehmend wurde von den Bereichsleitern, gemeinsam mit dem Projektleiter, eine erste Prozesslandkarte der eigenen Organisation erstellt. Daraus wurden die fünf wichtigsten Prozesse extrahiert mit dem Ziel, diese anschließend zu identifizieren. Dazu zählte

die genaue Abgrenzung der Prozesse. Die Prozesse wurden in einer eigens eingerichteten Datenbank erfasst, welche später den Zugriff auf dieses Wissen aus dem Workflow-Management-System ermöglicht.

Am Ende des Meetings wurden Aufgaben für die nächste Sitzung verteilt. Diese Aufgaben bestanden im Wesentlichen aus der Vorbereitung für eine grobe Definition der Prozesse.

### **5.1.2 Prozessteam Meeting 2**

Am Beginn des zweiten Prozessteam Meetings (PTM 2), welches eine Woche nach dem ersten Prozessteam Meeting stattfand, wurden die fünf ausgewählten Prozesse grob von den Bereichsleitern definiert. Diese Definition umfasste vor allem Prozessschritte, die von Rechts wegen verbindlich durchzuführen sind. Außerdem wurde festgelegt, welche Personen nach bestimmten Prozessschritten zu informieren sind. Des Weiteren wurden Prozessziele und daraus abgeleitete Messgrößen bestimmt. Anschließend an dieses Meeting wurde PTM 3 durchgeführt.

### **5.1.3 Prozessteam Meeting 3**

Am dritten Prozessteam Meeting (PTM 3) nahmen das Prozessteam sowie der Projektleiter teil. Die Ziele dieses Meetings waren, einerseits den aktuellen IST-Zustand der Prozesse zu erfassen, andererseits basierend auf den Vorgaben aus dem zweiten Prozessteam Meeting einen idealen SOLL-Prozess zu modellieren. Nach erfolgter Modellierung des IST-Zustandes wurden daraus gemeinsam, im Konsens des Prozessteams, Verbesserungspotenziale abgeleitet. Aus diesen Verbesserungspotenzialen sowie den Vorgaben aus PTM 2 wurden möglichst ideale SOLL-Prozesse geschaffen. Außerdem wurden die zuvor definierten Messgrößen und Prozessziele diskutiert und weiter ausgefeilt. Die Prozesse wurden in einer speziellen Software (jPDL Designer siehe 8.2.3.1.1) grafisch modelliert. Zusätzlich wurden die einzelnen Arbeitsaufgaben der Prozesse grob definiert (Verantwortlichkeiten, kurze Beschreibung der Aufgabe). Diese Informationen wurden wiederum in einer Datenbank abgelegt. Dieses Meeting wurde mehrmals, bis ein endgültiges Konzept des SOLL-Zustandes gefunden wurde durchgeführt. Zwischen den Meetings wurde dem Prozessteam mehrere Tage Zeit gegeben, um sich etwaige Verbesserungen am SOLL-Zustand zu überlegen, bzw. Daten zu sammeln.

### **5.1.4 Prozessteam Meeting 4**

Im letzten Prozessteam Meeting (PTM 4) wurde das SOLL-Konzept vom Prozessteam den Bereichsleitern präsentiert. Die Prozesse wurden nochmals durchgesprochen und gegebenenfalls auf die Wünsche der Bereichsleiter angepasst. Dar-

aufhin erfolgte die Freigabe der Prozesse zur Implementierung durch die Bereichsleiter.

Prozessteam	Projektleiter	Bereichsleiter
PTM 1	<div data-bbox="687 521 1417 577">Vorstellung Projekt</div> <div data-bbox="687 589 1417 645">Vorstellung Prozesslandkarte</div> <div data-bbox="687 656 1417 712">Erstellung Prozesslandkarte</div> <div data-bbox="687 723 1417 779">Identifikation &amp; Abgrenzung d. Prozesse</div> <div data-bbox="687 790 1417 846">Erfassen d. Prozesse in Datenbank</div>	
PTM 2	<div data-bbox="687 869 1417 925">Definition Prozesszuständigkeiten</div> <div data-bbox="687 936 1417 992">Definition Prozessziele</div> <div data-bbox="687 1003 1417 1059">Definition Messgrößen</div>	
PTM 3	<div data-bbox="292 1097 1018 1153">Erfassung IST-Zustand</div> <div data-bbox="292 1164 1018 1220">Identifikation Verbesserungspotentiale</div> <div data-bbox="292 1232 1018 1288">Modellierung SOLL-Zustand</div> <div data-bbox="292 1299 1018 1355">Grobdefinition d. Arbeitsschritte</div>	
PTM 4	<div data-bbox="292 1379 1417 1435">Präsentation SOLL-Konzept</div> <div data-bbox="292 1447 1417 1503">Diskussion</div>	<div data-bbox="1075 1514 1417 1570">Freigabe</div>

Abbildung 5.1: Prozessteam Meetings (PTM)

## 5.2 Entwurf des Workflow-Management-Systems

Der Entwurf des WMS gliedert sich in zwei Phasen. Erstens das Erstellen eines funktionalen Prototyps und zweitens die Vorbereitung der Implementierung.

### 5.2.1 Prototyp

Für das Erstellen des Prototyps wurden ein MySQL Server, Eclipse sowie die Frameworks JBoss jBPM und Seam auf einen Testserver kopiert. Basierend auf der Anleitung von [Küh] wurde Eclipse derart konfiguriert, dass es möglich war, einen JBoss Application Server (AS) in Eclipse zu starten. Auf diesem AS wurden jBPM und Seam installiert (publiziert). Mit Hilfe der in jBPM eingebauten Webkonsole war es möglich, die Funktionalität von jBPM auf einfache Art und Weise zu testen. Ausgehend von diesen viel versprechenden Tests wurde ein einfaches Seam Projekt zur Steuerung von jBPM erstellt. Aufbauend auf diesen Untersuchungen wurde die Kombination aus jBPM und Seam als für die weitere Implementierung gut geeignet befunden. Aufbauend auf dem Prototypen und basierend auf den Anforderungen wurde eine Evaluierung zwischen den zwei Softwarewerkzeugen Track+ (siehe 7.2.1) und jBPM durchgeführt.

### 5.2.2 Methodik der Implementierung

Vorbereitend zur Implementierung wurde ein eigener Server aufgesetzt, welcher einzig der Implementierung und dem Betrieb des WMS dienen sollte. Auf diesem Server wurden die benötigten Server (JBoss Application Server, MySQL Server, Apache Webserver), die Entwicklungsumgebung (Eclipse) sowie die Frameworks (jBPM, Seam, Richfaces) installiert. Da eine Integration des WMS in das bestehende Zeiterfassungssystem eine unbedingt notwendige Anforderung darstellt, wurde auch das Zeiterfassungssystem auf diesem Server installiert. Das Zeiterfassungssystem basiert auf PHP 4 und verwendet einen MySQL Server für die Persistenz. Weiters wurde ein Backup der gesamten Datenbank des Zeiterfassungssystems auf diesen Server kopiert, um die Tests so realistisch wie möglich zu gestalten. Um den Verlauf der Implementierung des WMS zu dokumentieren und um die Integration in das Zeiterfassungssystem zu erleichtern, wurde die Software Subversion<sup>13</sup> zur Versionskontrolle eingesetzt.

---

<sup>13</sup><http://subversion.tigris.org/>

## **Kapitel 6**

# **Erfassen der Workflows**

Im folgenden Kapitel werden die erhobenen Workflows erläutert. Zentraler Bestandteil der Workflows ist der ideale SOLL-Zustand, welcher in diesem Kapitel detailliert beschrieben wird. Wie 5.1.3 zu entnehmen ist, wurden die Workflows direkt im jPDL Designer modelliert. Die hier dargestellten Abbildungen dienen lediglich der Visualisierung der Prozesse. Als Visualisierungsmethode wurde eine Variante der Prozessablaufdarstellung gewählt (siehe Kapitel 2.1.7.2.2). Sie soll dem Leser auf einen Blick die einzelnen Arbeitsschritte und Zuständigkeiten vermitteln.

### **6.1 Workflow „Kundenwunsch“**

Viele der von der Organisation verkauften Softwareprodukte sind modular aufgebaut. Das bedeutet, es lassen sich nach der Installation weitere Funktionen hinzufügen. Beispielsweise können für eine Software Datenbankprozeduren erstellt werden, welche es ermöglichen, für die Nutzer wichtige Daten zu errechnen. Eine derartige Prozedur könnte, basierend auf dem von einem Benutzer eingegebenen aktuellen Gewicht eines Patienten, die Differenz zum letzten Gewicht errechnen und diese anzeigen. Daraus ließe sich beispielsweise ablesen, ob die künstliche Ernährung eines Patienten ausreichend ist.

#### **6.1.1 Grund der Aufnahme in das WMS**

Der Workflow zur Erstellung einer Produkterweiterung wurde in das WMS aufgenommen, da es für die Organisation notwendig ist, den Ablauf zu standardisieren, um immer eine möglichst hohe Qualität zu liefern. Mangels zu Kundenwünschen zugehöriger Aufträge kam es bei der Verrechnung oft zu Problemen, da keine Verbindung zwischen Auftrag und Kundenwunsch hergestellt werden konnte. Weiters

wurde eines der Softwareprodukte kürzlich zum Medizinprodukt ([Öst]) erklärt, wodurch bei der Implementierung von Erweiterungen bestimmte rechtliche Verbindlichkeiten einzuhalten sind. Die tatsächliche Durchführung der rechtlich verbindlichen Tätigkeiten soll vom WMS unterstützt werden.

### **6.1.2 IST-Zustand**

Im aktuellen Zustand wird ein Erweiterungswunsch von Kunden direkt an den betreuenden Mitarbeiter des Projektteams (meistens den Projektleiter) kommuniziert. Der Projektleiter veranlasst danach die Implementierung bei einem Kollegen des Technik-Teams. Dieser führt die entsprechenden Arbeiten durch und installiert das neue Feature direkt beim Kunden.

### **6.1.3 Verbesserungspotenziale**

Um sicherzustellen, dass die implementierten Erweiterungen die Wünsche der Kunden vollständig abdecken, sollte vom Kunden ein Anforderungsformular ausgefüllt werden, in welchem die Wünsche und Anforderungen genau spezifiziert sind. Zusätzlich muss vom Kunden ein schriftlicher Auftrag an die Organisation erteilt werden. Dieser garantiert die Verrechenbarkeit der durchgeführten Leistungen und erleichtert es der Organisation zu verfolgen, welche Erweiterungen bereits implementiert wurden. Von der implementierenden Person sollte eine genaue Dokumentation der Erweiterung verfasst werden. Diese kann einerseits für den Kunden als Handbuch dienen, andererseits besteht von Gesetzes wegen die Notwendigkeit dazu. Nach erfolgter Implementierung und Produktivsetzung der Erweiterung beim Kunden sollte die Abnahme der Erweiterung erfolgen. Im Zuge der Abnahme sollte der Kunde schriftlich erklären, dass die Erweiterung alle gestellten Wünsche abdeckt und korrekt funktioniert. Zuletzt sollte von dem Projektteam die Verrechnung der durchgeführten Leistungen, basierend auf dem Auftrag, veranlasst werden. Es sollte für den Kunden von Beginn an genau einen Ansprechpartner bei der Organisation geben (Projektleiter). Dieser ist für sämtliche Kundenkontakte während des gesamten Prozesses verantwortlich. Dies soll verhindern, dass der Kunde mit mehreren verschiedenen Personen Kontakt aufnimmt und niemals tatsächlich weiß wer für ihn zuständig ist.

### **6.1.4 SOLL-Zustand**

In diesem Abschnitt sind die einzelnen Aufgabenblöcke des SOLL-Zustandes des Prozesses Kundenwunsch erklärt. Die Aufgabenblöcke sind dem Diagramm 6.1 entnommen, der dazugehörige Quellcode befindet sich in Listing A.2.

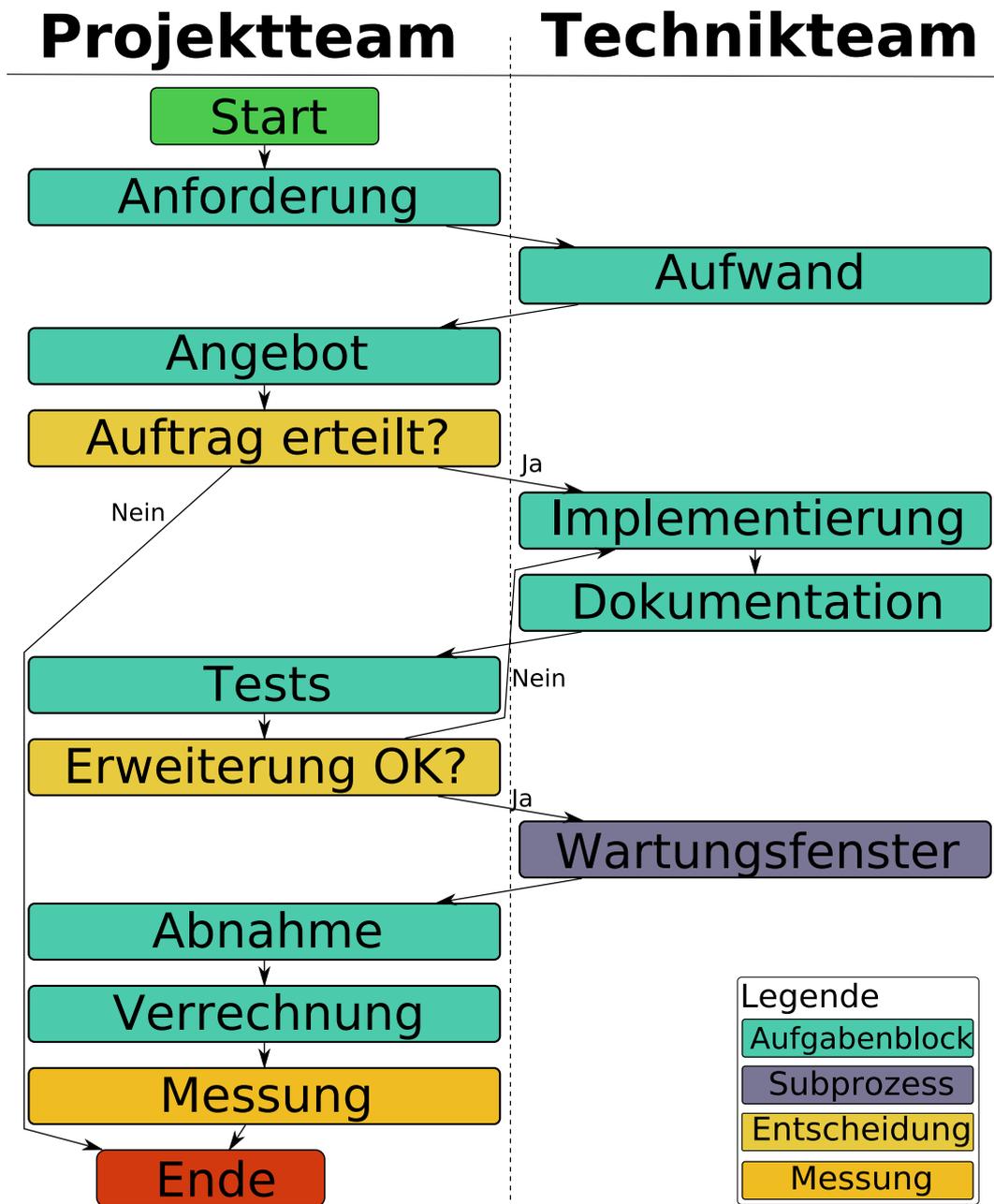


Abbildung 6.1: SOLL-Zustand des Prozesses „Kundenwunsch“

**Anforderung** Die Anforderung des Kunden wird vom verantwortlichen Projektleiter entgegengenommen und geprüft. Diese Prüfung beinhaltet die Validierung, ob der Benutzer berechtigt ist, die Anforderung zu stellen. Nach der Validierung wird der Inhalt der Anforderung geprüft: einerseits auf die technische Realisierbarkeit, andererseits auf die Vollständigkeit der Informationen. Gemeinsam mit dem Kunden wird anschließend ein detailliertes Anforderungsdokument erstellt, welches alle für die Implementierung wichtigen Punkte enthält.

**Aufwand** Nach der Anforderung erfolgt eine Aufwandsschätzung durch einen Mitarbeiter des Technikteams.

**Angebot** Der geschätzte Aufwand wird dem Projektleiter mitgeteilt. Dieser erstellt aus der Schätzung und dem Anforderungsdokument ein Angebot und sendet dieses an den Kunden.

**Auftrag erteilt?** Der Kunde hat dann die Möglichkeit, einen Auftrag zu erteilen und somit die Entwicklung zu veranlassen oder keinen Auftrag zu erteilen. In letzterem Fall endet der Prozess an dieser Stelle.

**Implementierung** Wurde der Auftrag erteilt, so beginnt ein Mitarbeiter des Technikteams möglichst rasch mit der Implementierung. Parallel dazu verlaufen erste Tests der neuen Erweiterung.

**Dokumentation** Ist die Implementierung abgeschlossen, so muss vom Technikteam die Dokumentation zu der Erweiterung erstellt werden.

**Tests** Danach erfolgen finale Tests durch den Projektleiter. Dieser hat sicherzustellen, dass die Anforderungen des Kunden mit der implementierten Erweiterung vollständig abgedeckt sind und dass diese einwandfrei funktioniert.

**Erweiterung OK** Ist dies nicht der Fall, so wird wiederum die Phase „Implementierung“ durchgeführt.

**Wartungsfenster** Andernfalls erfolgt der Start eines Wartungsfensters<sup>14</sup> beim Kunden. Dies ist ein Subprozess, welcher alle notwendigen Tätigkeiten zur Produktivsetzung der Erweiterung beinhaltet.

**Abnahme** Nach erfolgter Produktivsetzung hat vom Kunden eine Abnahme zu erfolgen.

---

<sup>14</sup>Ein Wartungsfenster ist ein zeitlicher Abschnitt (einige Stunden), in welchem das Softwareprodukt dem Kunden teilweise nur beschränkt zur Verfügung steht. Produktives Arbeiten des Kunden ist in dieser Zeitspanne meistens nicht möglich. Wartungsfenster müssen für gewöhnlich im Vorfeld beim Kunden beantragt werden.

**Verrechnung** Schließlich wird vom Projektleiter die Verrechnung veranlasst.

#### 6.1.4.1 Zusätzliche Aufwände vom IST- zum SOLL-Zustand

Es müssen für die im Prozess eingebetteten Dokumente (Angebot, Auftrag, Dokumentation) Vorlagen erstellt werden. Diese Dokumente müssen für jede Prozessinstanz ausgefüllt werden.

Zusätzlich zu den bereits vorhandenen Tests (vom Technikteam) muss auch der zuständige Projektleiter Tests durchführen. Dafür kann u.U. eine Einschulung für den Projektleiter erforderlich sein. Da für das Einspielen der Erweiterung beim Kunden ein Wartungsfenster notwendig ist, entsteht zusätzlicher organisatorischer Aufwand. Die anschließende Abnahme sollte vor Ort beim Kunden vom Projektleiter durchgeführt werden.

#### 6.1.4.2 Verbesserungen vom IST- zum SOLL-Zustand

Da für jeden Kundenwunsch ein Auftrag unterfertigt werden muss, kann mit Sicherheit eine Verrechnung der erbrachten Leistungen durchgeführt werden. Weiters kann durch das zweimalige Testen der Erweiterung eine höhere Qualität der Leistung garantiert werden als zuvor (ohne zweiten Test). Zusätzlich garantiert der Prozessschritt „Abnahme“, dass der Kunde mit den implementierten Leistungen zufrieden ist und dass die Leistungen erwartungsgemäß funktionieren.

#### 6.1.5 Messgrößen

	Messgröße 1	Messgröße 2
Name	Anzahl durchgeführter Prozessschritte	Erteilung des Auftrags
Verantwortung	Projektleiter	Projektleiter
Prozessziel	Vollständige Implementierung	Auftragserteilung
Zielwert	100 %	Ja
Messmethode	Manuell	Manuell
Messfrequenz	Bei Prozessende	Bei Prozessende

## 6.2 Workflow „Produktaktualisierung“

Dieser Workflow beschreibt die Durchführung eines großen Produktupdates für eines der verkauften Softwareprodukte. Die Software basiert auf einer Client-Server Architektur und wird bei jedem Kunden von vielen Benutzern gleichzeitig verwendet. Es ist für die Kunden von hoher Bedeutung, dass die Software jederzeit voll funktionsfähig ist, da sie Teil seiner Hauptgeschäftsprozesse ist. Daher muss

das Produktupdate möglichst rasch und ohne Fehler vorstattgehen. Für den Fall, dass die Aktualisierung fehlschlägt, muss ein Konzept vorhanden sein, welches beschreibt, wie der zuletzt funktionierende Zustand rasch wieder hergestellt werden kann. Dieser Prozess existierte bislang in der Organisation nicht, da es noch keine Produktupdates für dieses Produkt gab. Deswegen werden lediglich die Punkte „SOLL Zustand“ und „Messgrößen“ behandelt.

### **6.2.1 Grund der Aufnahme in das WMS**

Die Notwendigkeit der Aufrechterhaltung der hohen Verfügbarkeit der Software war der Hauptgrund für die Aufnahme in das WMS. Da während der Laufzeit dieses Prozesses hauptsächlich auf den IT-Systemen des Kunden gearbeitet werden muss, müssen bestimmte Rahmenbedingungen dafür beachtet werden. Das WMS soll die Anwender dabei unterstützen.

### **6.2.2 SOLL-Zustand**

In diesem Abschnitt sind die einzelnen Aufgabenblöcke des SOLL-Zustandes des Prozesses Produktaktualisierung erklärt. Die Aufgabenblöcke sind dem Diagramm 6.1 entnommen, der dazugehörige Quellcode befindet sich in Listing A.2.

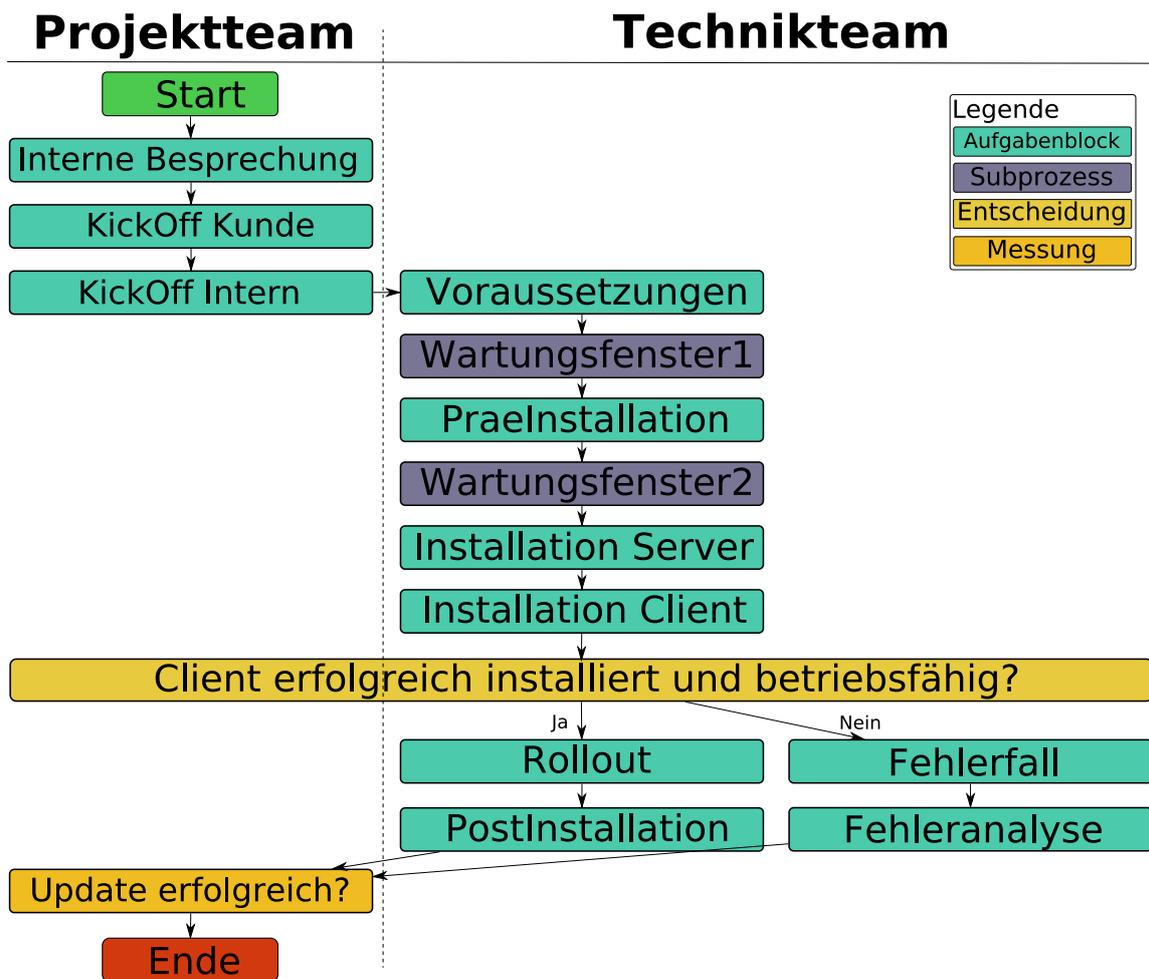


Abbildung 6.2: SOLL-Zustand des Prozesses „Produktaktualisierung“

**Interne Besprechung** Zum Prozessstart erfolgt eine Besprechung, welche der Abklärung des Machbarkeitszeitraumes der Softwareaktualisierung dient. Basierend auf den in der Organisation freien Ressourcen wird ein Termin für die Aktualisierung geplant.

**KickOffKunde** Im Rahmen eines Kick-off-Meetings beim Kunden werden der Termin für die Aktualisierung endgültig festgelegt, ein Projektplan erstellt sowie notwendige Anforderungen abgeklärt.

**KickOffIntern** Im nächsten Schritt erfolgt eine interne Kick-off Besprechung. Dabei sollten die benötigten Ressourcen festgelegt sowie die technischen Anforderungen für den Prozess (z.B.: notwendige Berechtigungen für die Kundenserver) geklärt werden. Der im vorhergehenden Schritt mit dem Kunden definierte Projektplan wird um technische Details erweitert und die Zuständigkeiten für die einzelnen Prozessschritte werden fixiert.

**Voraussetzungen** In einer Checkliste wurden im Vorfeld die Voraussetzungen für eine erfolgreiche Produktaktualisierung definiert (z.B.: vorhandene Softwarelizenzen, Installationspakete stehen bereit, . . .). Diese werden im aktuellen Prozessschritt überprüft. Weiters wird von dem Kundensystem ein Backup erstellt, welches im Fehlerfall rasch wieder eingespielt werden kann, um den Betrieb des Systems aufrechtzuerhalten.

**Wartungsfenster1** Um die Installation beim Kunden vorzubereiten, muss ein Wartungsfenster gestartet werden. Siehe auch Kapitel 6.1.

**PraeInstallation** Im Zuge der Präinstallation wird u.a. das Betriebssystem des Servers aktualisiert sowie für die neue Version zusätzlich benötigte Software installiert. Dieser Prozessschritt findet typischerweise einige Tage vor der eigentlichen Softwareaktualisierung statt.

**Wartungsfenster2** Für die tatsächliche Softwareaktualisierung muss ein eigenes Wartungsfenster beantragt werden.

**InstallationServer** Zusätzlich zu der im Prozessschritt „Voraussetzungen“ erstellten Sicherungskopie werden unmittelbar vor der Aktualisierung nochmals alle Datenbanken gesichert. Daraufhin erfolgt die Installation der neuen Serversoftware. Anschließend an diese wird am Server die neue Clientsoftware eingespielt. Mit dieser Clientsoftware erfolgen erste Funktionstests. Entstehen bei den Tests keinerlei Probleme, so kann die Installation aller anderen Clients veranlasst werden.

**InstallationClient** Im Zuge dieses Prozessschrittes wird vom Kunden mit Hilfe eines speziellen Installationspakets ein Client aktualisiert. Dieser wird weiters auf seine volle Funktionalität getestet.

**Entscheidung** Basierend auf den Resultaten der vorangegangenen Tests muss an diesem Punkt des Prozessablaufes eine Entscheidung getroffen werden, ob alle Clients des Kunden aktualisiert werden können oder ob die Aktualisierung abgebrochen werden muss, da Fehler aufgetreten sind.

**Pfad Rollout** Dieser Pfad wird ausgeführt, sofern alle vorhergehenden Tests positiv durchgeführt wurden.

**Rollout** Hier wird die Aktualisierung aller beim Kunden vorhandenen Clients angeordnet. Gleichzeitig wird die Konfiguration der Software vom Projektteam angepasst.

**PostInstallation** Im Zuge der Postinstallation wird der Server bereinigt (Entfernen von Installationspaketen und Backups). Weiters wird das alte (für die vorherige Softwareversion verwendete) Sicherungskonzept <sup>15</sup> angepasst und aktiviert.

**Pfad Fehlerfall** Dieser Pfad wird ausgeführt, sofern in den Tests Fehler festgestellt wurden, welche unmittelbar nicht behebbar waren.

**Fehlerfall** Im Fehlerfall werden die in den Abschnitten „Voraussetzungen“ sowie „InstallationServer“ erstellten Sicherungskopien wieder hergestellt. Außerdem wird der Kunde über die fehlgeschlagene Aktualisierung informiert.

**Fehleranalyse** Um den/die aufgetretenen Fehler zukünftig möglichst zu vermeiden bzw. das Risiko des Eintretens zu verringern, werden diese analysiert.

**Messung** Im letzten Prozessschritt erfolgt die Messung der Prozessleistung. Weitere Informationen sind dem Kapitel 6.2.3 zu entnehmen.

---

<sup>15</sup>Das Sicherungskonzept beinhaltet Methoden, welche in regelmäßigen Intervallen Sicherheitskopien der Software bzw. der Datenbanken erstellen.

### 6.2.3 Messgrößen

	Messgröße 1
Name	Update erfolgreich
Verantwortung	Projektleiter
Prozessziel	Erfolgreiche Durchführung der Produktaktualisierung
Zielwert	Ja
Messmethode	Manuell
Messfrequenz	Bei Prozessende

## 6.3 Workflow „Neuer Mitarbeiter“

### 6.3.1 Grund der Aufnahme in das WMS

Der Prozess ‚Neuer Mitarbeiter‘ wurde deshalb in das WMS aufgenommen, um bei Neueinstellung eines Mitarbeiters sicherzustellen, dass er ab seinem ersten Arbeitstag produktiv arbeiten bzw. seine Ausbildung rechtzeitig geplant und organisiert werden kann. Es entstand zeitweise das Problem, dass am ersten Arbeitstag des Mitarbeiters noch nicht alle IT-Berechtigungen gesetzt waren oder Hardware teilweise nicht vorhanden war. Eine weitere Problematik ergab sich, dass die Mitarbeiter die organisationsinternen Regeln/Bestimmungen entweder nicht kannten oder erst (zu) spät damit vertraut gemacht wurden. Dem soll mit diesem Schritt entgegengewirkt werden.

### 6.3.2 IST-Zustand

Um einen neuen Mitarbeiter einzustellen, finden mehrere Bewerbungsgespräche statt. Nach dem letzten Gespräch werden die Rahmenbedingungen des Arbeitsvertrages an das Personalbüro übermittelt, welches dann den Vertrag aufsetzt. Nach erfolgter Unterschrift des Vertrages wird vom zuständigen Bereichsleiter eine mündliche Anforderung, betreffend die notwendige Hard- und Software an den Systemadministrator gestellt. Am ersten Arbeitstag wird dem Mitarbeiter ein Arbeitsplatz zugewiesen sowie die angeforderte Hard- und Software bereitgestellt. Anschließend erfolgen bereits Schulungen für die zu betreuenden Softwaresysteme. Parallel dazu beginnt eine zweistufige Evaluierung der Leistungen des neuen Mitarbeiters. Sie wird von den Bereichsleitern durchgeführt.

### **6.3.3 Verbesserungspotenziale**

Da Richtlinien in der Organisation nicht immer einheitlich abgelegt oder sogar nur mündlich weitergegeben werden (siehe Kapitel 4), sollten diese einem neuen Mitarbeiter bereits zu Beginn des Dienstverhältnisses verständlich erklärt werden. Bei Beachtung der vorgegebenen Regeln/Richtlinien - die sehr wichtig und informativ für den neuen Mitarbeiter sind - können eventuell auftretende Spannungen zwischen den Kollegen vermieden werden. Die mündliche Übermittlung von Anforderungen für Hard- und Software an den Systemadministrator birgt die Gefahr, dass die Bereitstellung wichtiger Arbeitsmittel übersehen wird. Deswegen sollte die Erfüllung der Anforderungen vor Arbeitsbeginn des Mitarbeiters überprüft werden.

### **6.3.4 SOLL-Zustand**

In diesem Abschnitt sind die einzelnen Aufgabenblöcke des SOLL-Zustandes des Prozesses „Neuer Mitarbeiter“ erklärt. Die Aufgabenblöcke sind dem Diagramm 6.3 entnommen, der dazugehörige Quellcode befindet sich in Listing A.4.

**Personalbüro    Bereichsleiter    Technikteam**

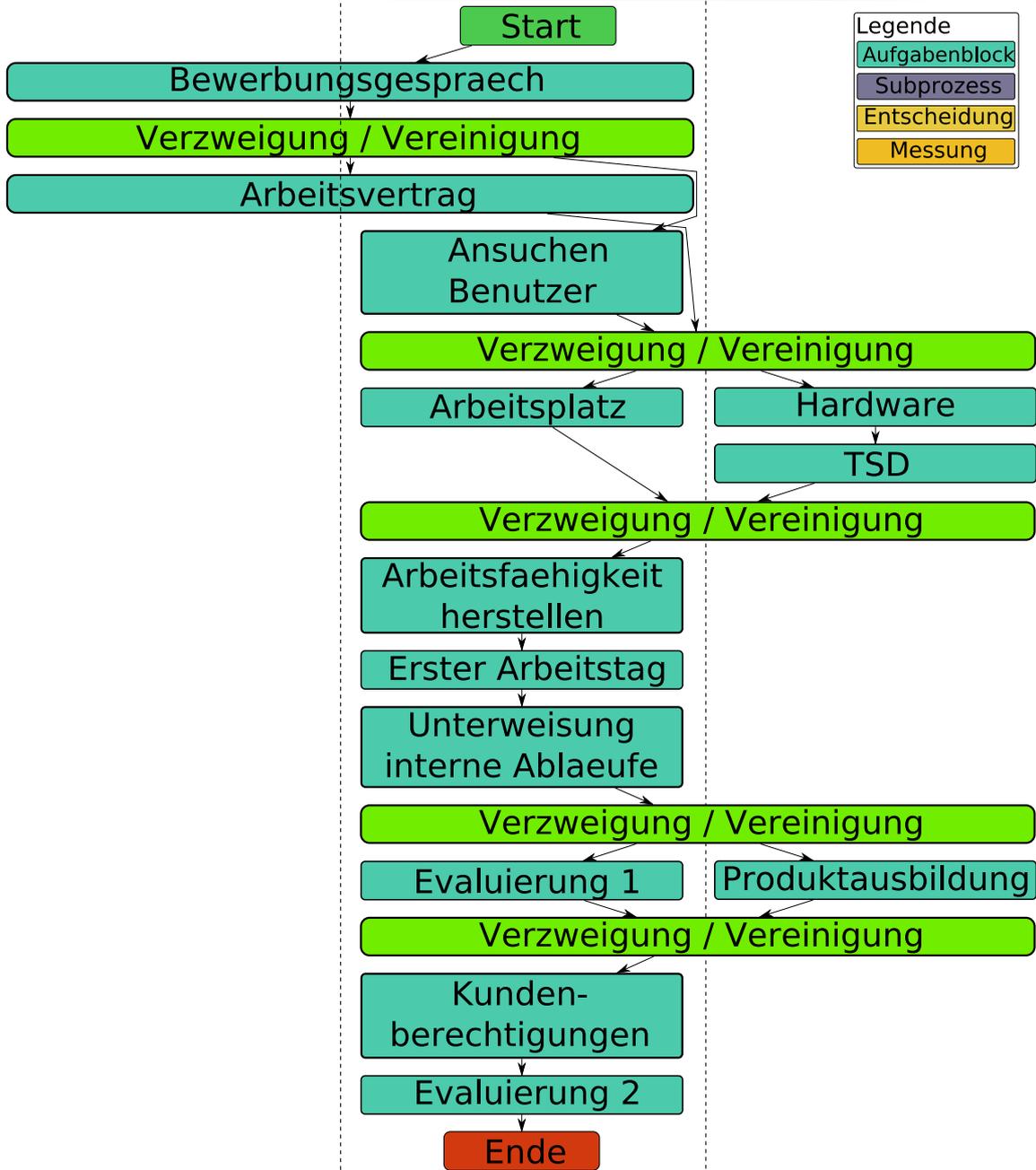


Abbildung 6.3: SOLL-Zustand des Prozesses „Neuer Mitarbeiter“

**Bewerbungsgespräch** Nach dem Bewerbungsgespräch (sobald feststeht, dass eine Person in die Organisation aufgenommen wird) sollen im Zuge dieser Prozessaufgabe der Typ des Arbeitsvertrags sowie zusätzlich ausgehandelte Bedingungen an das Personalbüro weitergeleitet werden, so dass dort der Arbeitsvertrag aufgesetzt werden kann.

**Arbeitsvertrag** In diesem Schritt hat der Bereichsleiter zu verifizieren, dass der Arbeitsvertrag vom neuen Mitarbeiter und dem Personalchef unterschrieben wurde. Dies ist eine Bedingung für die weitere Durchführung des Prozesses.

**AnsuchenBenutzer** Um sicherzustellen, dass am ersten Arbeitstag des neuen Mitarbeiters sämtliche Arbeitsmittel vorhanden und einsatzbereit sind, muss in diesem Schritt ein Anforderungsformular hierfür ausgefüllt und an den Systemadministrator weitergereicht werden. Dieses Formular beinhaltet benötigte IT-Berechtigungen, spezielle Hard- und Software, etc.

**Arbeitsplatz** Vor dem ersten Arbeitstag des neuen Mitarbeiters muss geklärt sein, welcher Arbeitsplatz diesem zugeteilt wird. Dies wird in diesem Prozessschritt festgelegt.

**Hardware** Basierend auf der Anforderung aus der Prozessaufgabe „AnsuchenBenutzer“ wird nun vom Systemadministrator der neue Computer eingerichtet.

**TSD** Sobald die Hardware funktionsfähig ist, soll die Schulungs- und Testumgebung (TSD: Training-/Schulungs-/Demosystem) auf dem Computer eingerichtet werden.

**ArbeitsfaehigkeitHerstellen** An diesem Punkt ist zu überprüfen, ob alle Anforderungen, welche während der Aufgabe „AnsuchenBenutzer“ gestellt wurden, erfüllt sind.

**ErsterArbeitstag** Diese Prozessaufgabe soll am ersten Arbeitstag des Mitarbeiters gestartet werden. Der Mitarbeiter soll u.a. durch die Firma geführt und den neuen Kollegen vorgestellt werden. Weiters sind im Rahmen dieser Prozessaufgabe wichtige Schulungen (Zutrittskontrolle, Zeiterfassung, ...) durchzuführen.

**UnterweisungInterneAblaeufe** Viele Organisationen besitzen ein Regelwerk, welches die Angestellten zu befolgen haben. Diese Regeln sollen dem Mitarbeiter erklärt werden.

**Evaluierung1** Diese Aufgabe soll ca. zwei Wochen nach Aufnahme des Mitarbeiters durchgeführt werden. Anhand der Qualität der vom Mitarbeiter be-

reits durchgeführten Aufgaben und seiner sozialen Kompetenzen soll über die Zukunft des Mitarbeiters bei der Organisation entschieden werden.

**Produktausbildung** Während der Produktausbildung wird dem neuen Mitarbeiter die Möglichkeit gegeben, sich mit den im Einsatz befindlichen Softwaresystemen vertraut zu machen. Es erfolgt einerseits eine Anwenderschulung, welche dem Mitarbeiter einen Überblick über das Produkt verschaffen soll, andererseits eine detaillierte technische Schulung, welche es dem Mitarbeiter letztendlich ermöglichen soll, Probleme von Kunden zu bewältigen.

**Kundenberechtigungen** Nach Abschluss der Ausbildung werden für den Mitarbeiter Benutzerkonten auf den Kundenservern bestellt (beim Kunden) und eingerichtet.

**Evaluierung<sup>2</sup>** Am Ende des Prozesses erfolgt eine zweite, letzte Evaluierung im Zuge derer von den Bereichsleitern entschieden wird, ob der Mitarbeiter über die Probezeit hinaus bei der Organisation beschäftigt werden soll.

#### **6.3.4.1 Zusätzliche Aufwände vom IST- zum SOLL-Zustand**

Um die Vollständigkeit der Anforderungen für Hard- und Software zu gewährleisten, muss ein Formular entwickelt werden, welches alle wichtigen Anforderungen beinhaltet. Dieses muss vom Bereichsleiter ausgefüllt und an den Systemadministrator gesendet werden. Zusätzlich dazu muss die tatsächliche Erfüllung der Anforderungen vor Dienstbeginn des neuen Mitarbeiters überprüft werden. Um dem neuen Mitarbeiter die organisationsinternen Richtlinien erklären zu können, müssen diese zuvor in der Organisation dokumentiert und einheitlich abgelegt werden.

#### **6.3.4.2 Verbesserungen vom IST- zum SOLL-Zustand**

Durch die Dokumentation und einheitliche Ablage der Richtlinien entsteht eine Art Nachschlagewerk, welches von allen Angestellten genutzt werden kann. Somit wird die Zeit für das Auffinden einer Richtlinie wesentlich verkürzt. Außerdem wird durch den Prozessschritt „Unterweisung Interne Abläufe“ sichergestellt, dass jeder Mitarbeiter die Regeln kennt und somit in der Lage ist diese anzuwenden. Durch die Verwendung eines Anforderungsbogens für Hard- und Software und die zusätzlich Überprüfung der Erfüllung der Anforderungen kann sichergestellt werden, dass für den neuen Mitarbeiter am ersten Arbeitstag bereits alle Arbeitsmaterialien vorhanden sind und er somit bereits vollständig einsatzbereit ist.

### 6.3.5 Messgrößen

	Messgröße 1	Messgröße 2
Name	Einsatzfähigkeit hergestellt	Aufnahme Mitarbeiter
Verantwortung	Bereichsleiter	Bereichsleiter
Prozessziel	Benötigte Hard- und Software vorhanden und eingerichtet	Aufnahme, Grundausbildung und Integration eines neuen Mitarbeiter
Zielwert	Ja	Ja
Messmethode	Manuell	Manuell
Messfrequenz	Einmalig beim Arbeitsantritt des neuen Mitarbeiters	Zweimalige Evaluierung

## 6.4 Workflow „Anwendersupport“

Entsteht bei einem Kunden ein Problem mit einem Softwareprodukt, so hat dieser die Möglichkeit, die Organisation zu kontaktieren und Unterstützung anzufordern. Die Lösung des Problems ist im Prozess Anwendersupport abgebildet. Der Prozess wird bei Kontaktaufnahme mit dem Kunden gestartet.

### 6.4.1 Grund der Aufnahme in das WMS

Häufig kommt es vor, dass Kunden bei Problemen den ihnen bekannten Projektleiter kontaktierten. War dieser allerdings gerade in einer Besprechung und konnte nicht umgehend zurückrufen, reagierten die Kunden oft verärgert. Diese Problematik soll durch eine einzelne Ansprechperson, welche auf die Lösung der Probleme des jeweiligen Kunden spezialisiert ist, gelöst werden. Weiters besteht die Notwendigkeit, für den Kunden einen Nachweis der für ihn durchgeführten Tätigkeiten zu erbringen. Dies gestaltet sich insofern schwierig, als der Kunde seine Anliegen nicht immer bei derselben Person bekannt gibt. Dadurch sind die Mitarbeiter dadurch gezwungen, eigene Listen für Anfragen der Kunden zu führen. Durch die Aufnahme in das WMS könnten diese Listen abgeschafft werden und das Auslesen der Anfragen je Kunde wäre einerseits rasch, andererseits automationsgestützt möglich.

### 6.4.2 IST-Zustand

Im IST-Zustand ruft der Kunde eine ihm bekannte Person aus der Organisation an. Diese nimmt die Anfrage/das Problem auf und leitet es an den zuständigen Projektleiter weiter. Der Projektleiter analysiert das Problem und verteilt dement-

sprechend Aufgaben. Nach der Erledigung dieser Aufgaben wird der Kunde über den Status informiert.

### **6.4.3 Verbesserungspotenziale**

Es ist problematisch, dass kein zuständiger Ansprechpartner für den Kunden zugeteilt ist. Der Kunde ruft meistens den Projektleiter oder das Sekretariat an, die beide nicht zuständig sind und das Gespräch weiter verbinden müssen. Dies kann einerseits die Kundenzufriedenheit drastisch senken, andererseits vor allem den Projektleitern kostbare Zeit stehlen. Deswegen sollte es für den jeweiligen Kunden einen klaren Ansprechpartner in der Organisation geben. Es kommt auch vor, dass Kunden wissen wollen, welche Anforderungen von ihnen bereits gestellt wurden. Um dies beantworten zu können, müssen sämtliche Anforderungen in einer Liste dokumentiert werden. Idealerweise sollte diese Liste automatisch ausgelesen werden können.

### **6.4.4 SOLL-Zustand**

In diesem Abschnitt sind die einzelnen Aufgabenblöcke des SOLL-Zustandes des Prozesses Anwendersupport erklärt. Die Aufgabenblöcke sind dem Diagramm 6.4 entnommen, der dazugehörige Quellcode befindet sich in Listing A.5.

## Projektteam      Technikteam

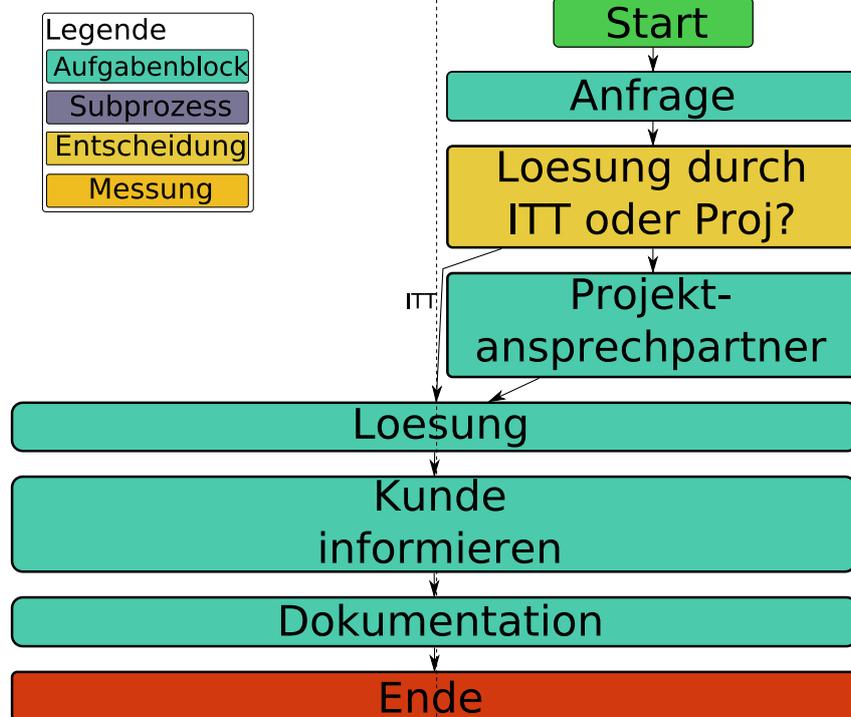


Abbildung 6.4: SOLL-Zustand des Prozesses „Anwendersupport“

**Anfrage** Ist eine Anfrage in der Organisation eingelangt (telefonisch oder elektronisch), wird von einem Mitarbeiter des Technikteams das Problem zunächst analysiert. Dazu zählt die Überprüfung, ob die Anfrage bearbeitet werden kann (z.B.: je nach Vorhandensein eines Wartungsvertrages). Weiters werden alle für eine Lösung notwendigen Informationen erhoben.

**Loesung durch ITT oder Projekt** In diesem Schritt gilt es, die Zuständigkeit für die Anfrage abzuklären. Falls das Projektteam zuständig ist, wird der Aufgabenblock „Projektansprechpartner“ gestartet, ansonsten erfolgt die Lösung des Problems „Loesung“ durch das Technikteam (ITT: IT-Technik).

**Projektansprechpartner** Ist das Projektteam zuständig (Konfigurationsproblem, Wunsch für die Implementierung einer Erweiterung), so wird der zuständige Projektleiter informiert und mit der Problembehebung beauftragt.

**Loesung** Im Zuge dieses Aufgabenblocks wird die Anfrage des Kunden bearbeitet.

**Kunde informieren** Nach Abschluss der Arbeiten wird der Kunde über den Status in Kenntnis gesetzt.

**Dokumentation** Sollten Änderungen an den Softwaresystemen des Kunden durchgeführt worden sein, müssen diese entsprechend dokumentiert werden.

#### **6.4.4.1 Zusätzliche Aufwände vom IST- zum SOLL-Zustand**

Es muss organisationsintern ein Ansprechpartner für den Kunden festgelegt werden. Die zuständige Person muss für alle Softwareprodukte geschult sein, um tatsächlich alle Anwender unterstützen zu können. Zusätzlich muss diese Person bei allen Kunden als alleiniger Ansprechpartner der Organisation vorgestellt werden.

#### **6.4.4.2 Verbesserungen vom IST- zum SOLL-Zustand**

Die Definition eines einzelnen Kontaktpunktes zu den Kunden hat mehrere Vorteile: Erstens ist für den Kunden klar, wen er kontaktieren muss, um rasch Unterstützung zu erhalten. Zweitens weiß diese zentrale Ansprechperson, ob ein Problem bereits mehrfach gemeldet wurde. So können andere Mitarbeiter entlastet werden. Drittens bietet das Starten eines Prozesses je Anfrage die Möglichkeit, automatisch auszulesen, wie viele/welche Anfragen ein Kunde bereits gestellt hat. Außerdem kann mit der Registrierung der Anfrage eventuell auch eine Verrechnung von Leistungen einhergehen.

#### 6.4.5 Messgrößen

	Messgröße 1
Name	Rasche Problemlösung
Verantwortung	Ansprechperson
Prozessziel	Lösen des Problems innerhalb der vertraglich festgelegten Zeitspanne
Zielwert	Ja
Messmethode	Manuell
Messfrequenz	Bei Prozessende

## **Kapitel 7**

# **Anforderungen an das Workflow-Management-System**

Die Anforderungen (allgemeine, nichtfunktionale) an das Workflow-Management-System werden in diesem Kapitel beschrieben. Zusätzlich erfolgt eine Evaluierung zwischen zwei Softwaresystemen um zu entscheiden, welches der beiden Systeme als Basis des zu implementierenden WMS dienen soll.

### **7.1 Anforderungen**

#### **7.1.1 Allgemeine Anforderungen**

Um Arbeitsabläufe (Workflows) in einem automatischen System verfolgen zu können, ist es in erste Linie notwendig, die Prozesse, welche die Workflows umfassen, zu definieren. Für die Prozessmanager ist es wichtig, dass sie Prozesse auf einfache, komfortable Art und Weise erfassen und visualisieren können. Für diesen Zweck muss eine gut zu bedienende Software zur Verfügung stehen. Neben der reinen Modellierung des Ablaufs der Prozesse ist es wichtig, auch Zuständigkeiten festzulegen. Nach der Modellierung der Prozesse muss es in dem zu implementierenden System möglich sein, Prozessinstanzen zu starten und die darin enthaltenen Aufgabenblöcke bestimmten Anwendern zuzuteilen.

Um die Anwender bei ihrer Arbeit mit dem WMS bestmöglich zu unterstützen, ist es von Vorteil, wenn diese stets auf Ressourcen, die aktuelle Aufgabe betreffend, Zugriff haben. Beispiele für derartige Ressourcen sind Dokumentenvorlagen, Checklisten oder Webseiten. Idealerweise soll beim Starten einer Aufgabe im WMS dem Benutzer eine Liste mit diesen verknüpften Ressourcen angezeigt werden. Die Ressourcen sollen schon während der Prozessdefinition vom Prozessverantwortlichen sowie dem Prozessmanager den einzelnen Aufgaben zuge-

wiesen werden können. Die Ressourcen sollen auf diese Art und Weise nicht bei dem Start jeder Prozessinstanz neu zugeteilt werden müssen, sondern sofort zur Verfügung stehen. Während des Betriebs des WMS muss es möglich sein, die Dokumentation zu den Aufgaben zu bearbeiten. Diese Dokumentation umfasst:

**Beschreibung** Eine kurze textuelle Beschreibung der Aufgabe. Hier sollten Punkte aufgelistet werden, die für das Erfüllen der Aufgabe besonders wichtig sind.

**Input** Der Input beschreibt Dokumente, welche an die aktuelle Aufgabe übergeben werden.

**Output** Der Output beschreibt sämtliche Dokumente oder Arbeitsergebnisse, die an den (die) nächste(n) Aufgabe(n) weitergereicht werden.

**Ziel** Beschreibt das Ziel der Aufgabe. Mögliche Ziele könnten die Maximierung der Kundenzufriedenheit oder der Abschluss einer Programmieraufgabe sein.

**Abgrenzung** Dieses Feld beschreibt, in welchen Punkten sich die aktuelle Aufgabe von anderen z.B. nachfolgenden Aufgaben unterscheidet.

**Durchführungsverantwortlicher** Der Durchführungsverantwortliche ist jene Person, welche die Aufgabe erledigen sollte.

**Entscheider** Fallen bei der Aufgabe zu treffende Entscheidungen an, so ist diese Person dafür verantwortlich.

**Mitarbeiter** Unter Mitarbeiter werden alle Personen aufgelistet, welche diese Aufgabe bearbeiten können.

**Informationsrecht** Alle Personen im Feld Informationsrecht sollten über den Status der Aufgabe informiert werden.

Für jede Prozessinstanz muss es möglich sein, ein definiertes Prozessenddatum vorzugeben. Basierend auf dem Enddatum soll es möglich sein, für einzelne Aufgabenblöcke <sup>16</sup> einerseits ein eigenes Enddatum, andererseits eine erwartete Zeitaufwand für die Arbeitsaufgaben anzugeben. Da Prozesse für gewöhnlich mehrere Aufgabenblöcke umfassen, ist es für die Anwender praktisch, wenn es vordefinierte Vorlagen für die Erstellung einer neuen Prozessinstanz gibt. Diese Vorlagen umfassen die Zuständigkeiten und Fälligkeiten für die einzelnen Aufgabenblöcke. Nach dem Start der Prozessinstanz müssen von dem WMS die Zuständigkeiten für die Aufgabenblöcke ausgelesen und die einzelnen Aufgaben damit

---

<sup>16</sup>Ein Aufgabenblock ist eine Liste von Aufgaben, welche parallel von derselben Person durchzuführen sind.

versehen werden. Jeder Benutzer soll eine eigene Aufgabenliste haben. Da in dieser Liste erwartungsgemäß viele Aufgaben aufscheinen werden, muss dem Anwender eine Möglichkeit gegeben werden, diese Liste u.a. nach der Priorität der Aufgaben zu sortieren. Das WMS soll es erlauben, den Benutzer zu warnen, falls eine Aufgabe schon gestartet hätte werden müssen. Auch dem Management soll es möglich sein, auf einen Blick benutzerübergreifend verspätete Aufgabenerledigungen zu identifizieren, um darauf entsprechend reagieren zu können.

Da Projektleiter bzw. Prozessverantwortliche einen Überblick über den momentanen Status der Projekte/Prozesse benötigen, muss es in dem WMS möglich sein, auf Prozessebene anzuzeigen, ob Prozesse verspätet sind bzw. welche Arbeitsaufgaben bereits abgeschlossen sind. Diese Informationen sollen in einem GANTT Diagramm<sup>17</sup> dargestellt werden.

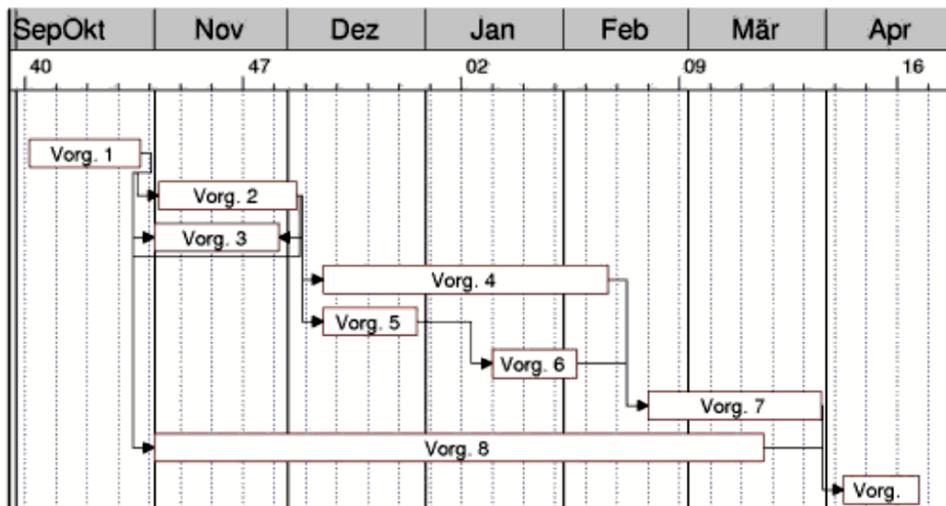


Abbildung 7.1: Gantt-Diagramm [Ang05, S. 72]

Damit im Nachhinein nachvollziehbar ist, welcher Anwender des WMS für welche Aufgabe wie viel Zeit benötigt hat (und somit auch wie viel Zeit für einen Prozess benötigt wurde), besteht die Anforderung, eine Schnittstelle zu einem Zeitdokumentationssystem zu schaffen. Wurde eine Aufgabe gestartet, dann soll es möglich sein, Zeitmarken konkret zu dieser Aufgabe zu dokumentieren. Die Dokumentation muss die Vorgangsart (beispielsweise „Besprechung“ oder „Programmierung“) sowie den Vorgangsort („Organisationsintern“ oder „Extern“ [bei einem

<sup>17</sup>Eine Diagrammart aus dem Projektmanagement. Es wird je durchzuführender Aufgabe ein Balken, welcher den nötigen Zeitbedarf sowie die zeitliche Abfolge der Aufgabe repräsentiert, dargestellt. Aus der Summe der Balken lässt sich gut erkennen, wann welche Aktivitäten beginnen und enden.

Kunden]) beinhalten. Außerdem müssen die chronologisch zuletzt gesetzten Zeitmarken zu der aktuellen Aufgabe ersichtlich sein. Ferner muss das WMS aus dem ZDS bedienbar sein um zu verhindern, dass der Anwender ständig zwischen mehreren Programmen wechseln muss.

Um Anwender über Veränderungen der Aufgaben am Laufenden zu halten, soll es eine Funktionalität geben, welche für jeden Benutzer die Änderungen seit dem letzten Login anzeigt. Diese Daten sollen u.a. das Zuteilen neuer Aufgaben, das Ändern der Zuständigkeit für Aufgaben oder etwa das Erstellen einer neuen Prozessinstanz beinhalten.

Die für die Bereichsleiter bzw. Angestellten wichtigsten Anforderungen an das neue System wurden gemeinsam mit diesen wie folgt festgelegt:

**Niedrige Anschaffungskosten** Die Anschaffungskosten sollen, wenn möglich, niedrig gehalten werden.

**Automatische Zuteilung von Zuständigkeiten** Zuständigkeiten für Aufgaben sollen vom System automatisch zugewiesen werden.

**Vorlagen f. Zuständigkeiten** Beim Starten eines Prozesses sollen die Zuständigkeiten für die einzelnen Aufgaben automatisch, basierend auf einer Vorlage, angezeigt werden.

**Definiertes Prozessendedatum** Für die Zeitplanung soll ein Prozessendedatum angegeben werden können.

**Zeitliche Abhängigkeiten** Arbeitsaufgaben sollen zeitlich voneinander abhängig sein.

**Projektfortschritt** Es soll dem Management immer möglich sein, den Stand der Prozesse abzufragen.

**Projektübersicht** In einer Übersicht sollen alle Prozesse grafisch dargestellt werden, um deren zeitlichen Verlauf beurteilen zu können (pünktlich, verspätet).

**Reporting Planüberschreitungen** Bei Überschreitungen der geplanten Dauer bzw. der Fälligkeit von Aufgaben sollen vom System automatisch Warnungen generiert werden.

**Zusammenfassung gesamte Prozessdokumentation** Sämtliche Dokumentation zu einem Prozess soll unkompliziert auslesbar sein.

**Timeline** Es soll den Nutzern ermöglicht werden, Änderungen seit ihrem letzten Login anzuzeigen (Beispielsweise ob ihnen neue Aufgaben zugeteilt wurden).

**Anzeige Status paralleler Aufgaben** Nutzer sollen den Status paralleler Arbeitsaufgaben (welche von Kollegen durchgeführt werden) jederzeit einsehen können.

**Anlegen von Ressourcen** Zu jeder Aufgabe sollen Ressourcen (Checklisten, etc.) hinterlegt werden können.

**Beschreibung für Aufgaben** Zu jeder Aufgabe soll Dokumentation hinterlegt werden können.

**Prozessmessungen** Um die Leitung der Prozesse zu messen, sollen Prozessmessungen ermöglicht werden.

**Anpassung Design** Das Design des neuen Systems soll an das bereits bestehende Design des ZDS angepasst werden können.

**Zeitdoku** Es soll möglich sein, zu den einzelnen Aufgaben etwaige Kommentare zu dokumentieren.

**Support** Bei auftretenden Problemen soll mittels Kundensupport rasch eine geeignete Lösung gefunden werden können.

**Regelm. Produktupdates** Um etwaige Programmfehler auszubessern, soll das Produkt stets auf dem letzten Stand gehalten werden.

## 7.1.2 Nichtfunktionale Anforderungen

Die Unterkapitel der nichtfunktionalen Anforderungen wurden der Norm ISO/IEC 9126 [JKC04, S. 3] entnommen:

### 7.1.2.1 Funktionalität

**Angemessenheit:** Mit dem WMS soll es lediglich möglich sein, den Ablauf von Prozessen zu steuern, sowie Prozesse und dazugehörige Dokumentation zu verwalten.

**Richtigkeit:** Es ist sicherzustellen, dass beispielsweise beim Starten eines Prozesses tatsächlich der vom Nutzer ausgewählte Prozess gestartet wird. Ebenso muss bei der Durchführung einer Aktion (für eine Aufgabe) die Aktion für genau die ausgewählte Aufgabe durchgeführt werden.

**Interoperabilität:** Eine Interaktion zwischen dem WMS und dem ZDS muss reibungslos möglich sein. Es müssen vom WMS die Softwaresysteme verwendet werden.

**Sicherheit:** Es soll nur berechtigten Benutzern gestattet sein, auf die Funktionen des WMS zuzugreifen. Als berechtigt gelten alle aktiven Nutzer des ZDS. Die Benutzerdaten sind der Datenbank des ZDS zu entnehmen. Weiters soll ein Berechtigungskonzept basierend auf Benutzergruppen erstellt werden, welches es ermöglicht, verschiedene Bereiche des WMS nur für bestimmte Anwender zugänglich zu machen. Diese Benutzergruppen sollen in der Oberfläche des WMS während der Laufzeit bearbeitbar sein. Es ist unbedingt erforderlich, dass mehrere Benutzer gleichzeitig mit dem System arbeiten. Es muss daher Sorge getragen werden, dass die (getrennten) Sitzungen der Anwender sich nicht überschneiden.

#### 7.1.2.2 Zuverlässigkeit

**Reife:** Eventuell auftretende Fehlerzustände sollen vom WMS abgefangen und verarbeitet werden. Ein einzeln auftretender Fehler darf den Betrieb des WMS nicht beeinträchtigen.

**Wiederherstellbarkeit:** Die Daten der aktiven Prozessinstanzen sollen stets in einer Datenbank gesichert werden, um bei einem allenfalls auftretenden Systemabsturz nicht Gefahr zu laufen, Daten zu verlieren. Im Falle eines Ausfalls des WMS darf dies nicht den reibungslosen Betrieb des ZDS beeinträchtigen.

#### 7.1.2.3 Benutzbarkeit

**Verständlichkeit:** Das System soll zum größten Teil selbsterklärend sein. Das Interface muss logisch aufgebaut sein, sodass einem Anwender klar sein muss, welche Folgen seine Aktionen haben.

**Erlernbarkeit:** Für den versierten EDV-Anwender muss das WMS von Anfang an bedienbar sein. Es dürfen keine Schulungen für die richtige Bedienung erforderlich sein.

**Bedienbarkeit:** Der Aufwand für die Bedienung soll möglichst gering gehalten werden. Das WMS soll dem Anwender hauptsächlich Arbeit abnehmen anstatt zusätzliche zu schaffen.

**Attraktivität:** Die Integration vom WMS in das ZDS soll nahtlos erfolgen. Dies bedeutet, die Anwender sollen nicht bemerken, dass sie mit mehreren verschiedenen Softwareprodukten arbeiten.

#### 7.1.2.4 Effizienz

**Zeitverhalten:** Der Zugriff auf WMS Funktionen aus dem ZDS muss rasch erfolgen, es dürfen für den Anwender keine Wartezeiten von mehr als zwei Sekunden entstehen (ausgenommen komplexe Operationen wie das Erstellen eines GANTT-Diagramms aus einer großen Menge Daten).

**Verbrauchsverhalten:** Auf der Serverseite muss das zu entwickelnde System auf einem Computer mit zwei Gigahertz single core CPU, drei Gigabyte Arbeitsspeicher und 20 Gigabyte Festplattenspeicher lauffähig sein. Das WMS muss auf den aktuellen Windows-Server Betriebssystemen arbeiten. Für die Datenbank steht ein MySQL 5 Server bereit, welcher aber auf einem anderen Server installiert ist. Das bedeutet, das WMS muss auf die benötigten Datenbanken über das Intranet der Organisation zugreifen können. Für die Clients soll keine spezielle Software zur Bedienung des WMS notwendig sein. Die Verwendung des WMS muss mittels eines aktuellen Webbrowsers (Opera 9.6, Firefox 3, Internet Explorer 7, oder ähnliche) fehlerfrei möglich sein.

#### 7.1.2.5 Änderbarkeit

**Analysierbarkeit:** Der Quellcode soll ausreichend dokumentiert und Fehlermeldungen aussagekräftig sein. Dies soll sicherstellen, dass ein Fehler im Programm rasch gefunden werden kann.

#### 7.1.2.6 Übertragbarkeit

**Anpassbarkeit:** Sämtliche Konfigurationen müssen in möglichst wenigen XML Dateien erfolgen, um eine einfache Portierbarkeit des Systems zu gewährleisten.

**Installierbarkeit:** Die Installation muss unkompliziert vonstattengehen. Idealerweise sollen nur Dateien kopiert und Konfigurationsdateien angepasst werden müssen.

**Koexistenz:** Das WMS soll problemlos parallel mit anderen Softwareprodukten arbeiten können.

#### 7.1.2.7 Benutzerschnittstelle

Die Benutzeroberfläche des WMS soll nahtlos in die Benutzeroberfläche des ZDS integriert werden. Für den Nutzer soll es den Anschein haben, als ob er lediglich ein Softwareprodukt verwenden würde. Zu diesem Zweck ist das Design des

WMS auf das des ZDS abzustimmen. Das WMS soll weiters intuitiv verwendbar sein. Dies bedeutet, dass für einen versierten EDV-Anwender keine speziellen Schulungen notwendig sein dürfen.

### 7.1.2.7.1 Entwürfe

Die Aufgabenliste (Abbildung 7.2) soll dem Anwender einen raschen Überblick über alle anstehenden Aufgaben verschaffen. Zu den Aufgaben sollen zumindest die Namen der Aufgabe, des Prozesses und der Prozessdefinition angezeigt werden. Die Aufgabenliste soll nach Fälligkeit der Aufgaben sortierbar sein. Zusätzlich zur Fälligkeit soll eine Aufgabe eine optionale Priorität besitzen. Diese soll bei gleicher Fälligkeit der Aufgaben dem Anwender einen Hinweis geben, welche zuerst durchzuführen ist. Auf einfache Art und Weise soll es dem Anwender möglich sein, bereits gespeicherte Dokumentation zu einzelnen Prozessaufgaben anzuzeigen. Zudem muss der Anwender per Knopfdruck Dokumentation zu einer Prozessaufgabe verspeichern können. Für das Starten/Beenden von Prozessaufgaben sollen Buttons vorhanden sein, welche die durchzuführende Aktion an das WMS übermitteln. Weitere Informationen zu einer Prozessaufgabe sollen per Klick auf den Namen der Aufgabe erhältlich sein.

PRIORITÄT	FÄLLIGKEIT	KUNDE	AUFGABE	AKTIONEN
2	28.05.2009	INTERV	PROZESS_A: INSTANZ_1: <del>ABC</del>	START STOPP EDIT
4	27.04.2009	KUNDE1	PROZESS_B: INSTANZ_2: ABC	—

• 20.04.2009 15:18

DOKU 1 ... TEXT ...

... TEXT

• 22.04.2009 07:12

DOKU 2 ... TEXT

NEUE DOKU

Abbildung 7.2: Entwurf der Aufgabenliste

Die Aufgabendetails (Abbildung 7.3) sollen in einem eigenen Fenster angezeigt werden. Es gilt, hier möglichst viele für den Anwender wichtige Informationen darzustellen. Zu der aktuellen Prozessaufgabe müssen zumindest Name, Beschreibung und Ressourcen einsehbar sein. Die Ressourcen sollen als Liste dargestellt werden. Parallele Aufgaben anderer Nutzer sollen in einer eigenen Liste angezeigt werden. Diese Liste soll dem Anwender das Nachfragen (Zeitaufwand)

bei Kollegen betreffend den Status von Aufgaben abnehmen.

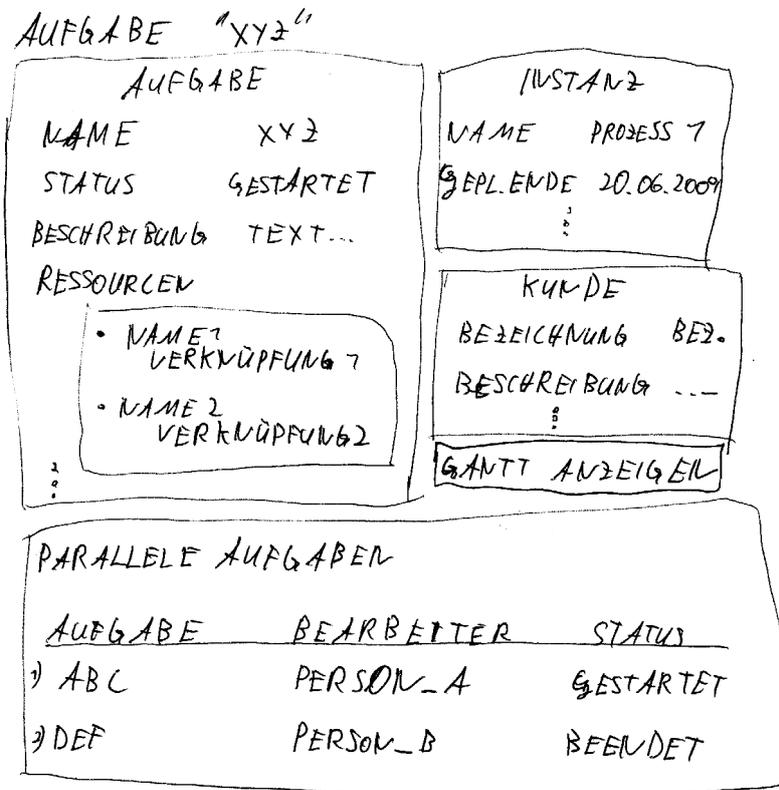


Abbildung 7.3: Entwurf der Aufgabendetails

Die Evaluierung der Benutzerschnittstelle ist Kapitel 9 zu entnehmen.

## 7.2 Build-or-Buy Analyse

Im Zuge der Implementierung war es notwendig, eine Entscheidung zu treffen, in welchem Framework Prozesse abgebildet werden sollen. Dieser Abschnitt beschreibt die Analyse, welche als Entscheidungshilfe für die Auswahl eines geeigneten Frameworks für die Prozesse diente. Für die Analyse wurden die Anforderungen an die Frameworks aufgelistet und gewichtet. Danach wurde zu jeder Anforderung der Grad der Erfüllung durch beide Frameworks in eine Tabelle eingetragen. Dieser Grad (Erf.) reicht von -1 (= Totale Nichterfüllung) bis +1 (Totale Erfüllung). Daraus konnte der gewichtete Grad der Erfüllung berechnet werden ( $Gew. = Gewichtung * Grad\ der\ Erfüllung$ ). Diese Daten wurden in eine Tabelle eingetragen und ermöglichen die Berechnung des Prozentsatzes der Erfüllung der

Anforderungen. Siehe Tabelle 7.1.

Für die Analyse standen zwei Softwareprodukte zur Auswahl: Track+ oder eine Eigenentwicklung basierend auf dem Framework JBoss jBPM (siehe 8.2.3.1). Die Software Track+ wurde von der Organisation bereits seit längerem testweise eingesetzt und leistete für die Vorgangsverwaltung gute Dienste. Daraus entstand die Idee, diese Software auch für das Workflow-Management einzusetzen. Das Framework jBPM wurde von der Organisation nominiert, da es einerseits auf Java basiert und das WMS auf möglichst vielen Betriebssystemen einsetzbar sein sollte und andererseits weil jBPM, wie auch einige der eigenen Softwareprodukte, auf einem JBoss Application Server betrieben werden können.

### **7.2.1 Track+**

Die Projektmanagement bzw. Vorgangsverwaltungs Software Track+ <sup>18</sup> wird von der gleichnamigen Firma Trackplus, einigen kommerziellen Entwicklern sowie Entwicklern aus der Open Source Community entwickelt. Nach der Installation der Software auf einem Tomcat Webserver oder JBOSS Application Server ist ein Zugriff darauf über das Inter- oder Intranet möglich. Nach der Installation müssen Benutzer angelegt werden. Diese können entweder direkt in Track+ erstellt werden oder alternativ besteht auch die Möglichkeit, einen LDAP <sup>19</sup> Server für die Benutzerauthentifizierung zu verwenden. Da in Track+ alle Aufgaben Projekten zugeordnet werden müssen, muss - nachdem Benutzer angelegt wurden - zwingend mindestens ein Projekt erstellt werden, um produktiv arbeiten zu können. Außerdem müssen diesem Projekt Benutzer zugewiesen werden ([Tra09b]). In Track+ können Vorgänge/Aufgaben angelegt und beliebigen Benutzern oder Gruppen zugeteilt werden. Der Benutzer kann dann zu dieser Aufgabe seine Tätigkeiten dokumentieren, Fälligkeiten abspeichern, zu informierende Personen angeben etc. Siehe Abbildung 7.4. Die meisten Teile des Benutzerinterfaces können über ein Templatesystem an eigene Designvorgaben angepasst werden. Tabellen können mit Hilfe von AJAX je Benutzer umgestaltet werden. Track+ unterstützt Emailbenachrichtigungen, falls eine Aufgabe bearbeitet oder eine Fälligkeit erreicht wurde. Weiters unterstützt Track+ ein Interface, in welchem Berichte über die im System gespeicherten Aufgaben generiert werden können.

#### **7.2.1.1 Vor- und Nachteile von Track+**

Der große Vorteil von Track+ liegt darin, dass Aufgaben sehr einfach definierbar sind (und viele Detailinformationen zu den Aufgaben angegeben werden können)

---

<sup>18</sup><http://www.trackplus.de/>

<sup>19</sup>LDAP: Lightweight Directory Access Protocol. Ermöglicht den Zugriff auf einen Verzeichnisdienst.

und Track+ ein fertiges Produkt ist, welches nur installiert werden muss. Auf Kostenseite gilt es, ein einmaliges Entgelt zu bezahlen, in welchem auch ein Maintenancevertrag (Produktunterstützung und Updates) für ein Jahr inkludiert ist ([Tra09a]).

Es in Track+ nicht möglich, Vorlagen für Prozesse zu hinterlegen und diese dann zu starten. Außerdem gestaltet sich die Integration in die in der Organisation verwendeten Softwaresysteme schwierig, da erst entsprechende Schnittstellen erstellt werden müssen. Nach dem Erstellen derartiger Schnittstellen ist fraglich, was damit bei Softwareupdates/Bugfixes von Track+ geschieht. Vermutlich müssen sie nach jedem Update erneut getestet und validiert werden.

## **7.2.2 Eigenentwicklung**

Eine Eigenentwicklung erfordert den Entwurf einer Architektur für die Datenbank, sowie die neu zu schreibenden Programmteile. Es muss ein System geschaffen werden, welches einerseits alle jBPM Funktionen unterstützt und andererseits erweiterbar ist, um alle Anforderungen vollständig abzudecken. Dafür werden zumindest ein Datenbankserver sowie ein Server, auf welchem der JBoss Application Server installiert werden muss, benötigt.

### **7.2.2.1 Vor- und Nachteile einer Eigenentwicklung**

Der Hauptvorteil liegt in der Möglichkeit, verschiedene Prozesse zu definieren. Diese Prozesse können gestartet werden und jeglicher Output, der von ihnen generiert wird, kann in Drittsysteme übergeleitet werden. Allgemein können die Anforderungen vollständig abgedeckt werden. Weiters ist es auf einfache Art und Weise möglich, die Eigenentwicklung optisch an die anderen in der Organisation eingesetzten Softwaresysteme anzupassen, so dass es für die Mitarbeiter den Anschein hat, als würden sie mit nur einer Software (mit erweitertem Funktionsumfang) arbeiten. Dies kann dazu beitragen, die Akzeptanz der Mitarbeiter zu erhöhen. Siehe auch [VR05].

Klarerweise entstehen bei einer Eigenentwicklung mehr Kosten als beim Kauf einer Lizenz. Zusätzlich muss das Softwareprodukt von der Organisation selbst gewartet werden.

## **7.2.3 Entscheidung**

Der folgenden Tabelle (7.1) ist zu entnehmen, welche der wichtigsten Anforderungen (siehe 7.1.1) Track+ bzw. jBPM standardmäßig abdecken.

Entscheidungskriterium	Gew.	Track+		jBPM + SEAM	
		Erf.	Gew.	Erf.	Gew.
Niedrige Anschaffungskosten	10 %	0	0 %	1	10 %
Automatische Zuteilung von Zuständigkeiten	10 %	0	0 %	1	10 %
Vorlagen f. Zuständigkeiten	5 %	0	0 %	0.5	2.5 %
Definiertes Prozessendedatum	5 %	0	0 %	0	0 %
Zeitliche Abhängigkeiten	10 %	0	0 %	1	10 %
Projektfortschritt	5 %	1	5 %	0.5	2.5 %
Projektübersicht	5 %	1	5 %	0.5	2.5 %
Reporting Planüberschreitungen	3 %	0	0 %	0	0 %
Zusammenfassung gesamte Prozessdokumentation	5 %	0	0 %	0	0 %
Timeline	2 %	0.5	1 %	0.5	1 %
Anzeige Status paralleler Aufgaben	5 %	0	0 %	0.5	2.5 %
Anlegen von Ressourcen	5 %	0	0 %	0	0 %
Beschreibung für Aufgaben	5 %	0.5	2.5 %	0.5	2.5 %
Prozessmessungen	3 %	0	0 %	0	0 %
Anpassung Design	5 %	1	5 %	1	5 %
Zeitdoku	5 %	0	0 %	0	0 %
Auslastung der Mitarbeiter	5 %	1	5 %	0.5	2.5 %
Support	5 %	0.5	2.5 %	0.5	2.5 %
Regelm. Produktupdates	2 %	0.5	1 %	1	1 %
<b>Total</b>	<b>100 %</b>		<b>27 %</b>		<b>55.5 %</b>

Tabelle 7.1: Erfüllung der Anforderungen durch Track+ bzw. jBPM und SEAM.  
Gew.: Gewichtung in Prozent; Erf.: Grad der Erfüllung (-1 bis 1)

Wie aus Tabelle 7.1 zu erkennen ist, erfüllt Track+ vier Anforderungen vollständig und ebenfalls vier teilweise, wohingegen jBPM fünf Anforderungen vollständig und acht teilweise erfüllt. Da es mit Hilfe von Track+ nicht möglich ist, Prozesse/Workflows<sup>20</sup> abzubilden, stellt diese Anforderung ein K.-o.-Kriterium für die Software Track+ dar. Zwar ist es möglich, in Track+ zu jedem Vorgang beliebig viele verschachtelbare Subvorgänge anzugeben, diese müssen aber für jedes Projekt neu angelegt werden. Außerdem ist keine Unterscheidung zwischen parallelen und seriellen Vorgängen möglich. Deswegen wurde entschieden, eine eigene Software zu entwickeln, welche die Anforderungen (Kapitel 7.1) vollständig

<sup>20</sup>Der Begriff Workflow wird in Track+ beispielsweise für Zustandsänderungen von Aufgaben („Geöffnet“, „In Bearbeitung“, „Abgeschlossen“, ...) verwendet. Ein Workflow gibt dort an, welchen Zustand eine Aufgabe (ausgehend von dem Aktuellen) annehmen kann. Diese „Workflows“ können in Track+ beliebig definiert werden.

abdeckt. Dies führt dazu, dass in der Organisation für die Verwaltung von Prozessen/Workflows zwei Softwaresysteme eingesetzt werden müssen (jBPM in Verbindung mit SEAM sowie der eigenen Zeitdokumentationssoftware).

John Guest

Cockpit Neuer Vorgang Abfragen Abmelden Hilfe

Vorgang anschauen

Bearbeiten Kopieren Verschieben Zustand ändern Kind hinzu Übergeordnet Privatisieren Zurück

**Vorgang Nr. 1100 : implementiert : issue1**

<b>Eintrag Nr. :</b> 1100	<b>Projekt :</b> TryItOut	<b>Vorgangstyp :</b> Änderungsanforderung
<b>Erzeugt :</b> 08.10.08 10:00	<b>Subsystem :</b> Application	<b>Priorität :</b> sofort
<b>Verfasser :</b> Guest, John	<b>Klasse :</b> Software	<b>Auswirkung :</b> blockierend
<b>Manager :</b> Administrator, Track+ System !	<b>Release festgestellt :</b> 2.2	<b>Beginn am :</b> 08.10.2008
<b>Bearbeiter :</b> Guest, John	<b>Ziel-Release :</b> 1.0	<b>Fällig am :</b> 09.10.2008
	<b>Version :</b> 123546	<b>Zustand :</b> implementiert

**Titel :** issue1

**Beschreibung :** description of issue1

**Übergeordnet :**

Abhängige Vorgänge	Nr.	Zustand	Titel
	1101	analysiert	issue2
	1461	geöffnet	Kinder und Neffen

**Historie** | Kommentar (0) | Anlagen (0) | Aufwände/Kosten | Beobachter | Versionsverwaltung | Verknüpfungen

Datum	Geändert von	Änderungsart	Neuer Wert	Alte
27.02.2009 17:07	Guest, John	Zustand	implementiert	ans
05.11.2008 11:57	Guest, John	Zustand	analysiert	zug
08.10.2008 10:15	Guest, John	Zustand	zugewiesen	geC
08.10.2008 10:13	Guest, John	Zustand	geöffnet	ges
08.10.2008 10:12	Guest, John	Zustand	geschlossen	geC
08.10.2008 10:00	Guest, John	Zustand	geöffnet	
27.02.2009 17:06	Guest, John	Allgemein	Comment was added. Done	
31.12.2008 14:54	Guest, John	Allgemein	Release noticed was changed from "2.0" to "2.2". Severity was changed from "major" to "blocking". Build was added. T...	
12.11.2008 20:21	Guest, John	Allgemein	Comment was added. At the tone it will be two o'clock in central europe. Live from the United States the news is next...	
05.11.2008 11:57	Guest, John	Allgemein	Bearbeiter wurde von "Management " nach "Guest, John " geändert. Kommentar wurde hinzugefügt. TEST	
08.10.2008 10:15	Guest, John	Allgemein	Parent No. was removed. The old value was "1080".	
08.10.2008 10:04	Guest, John	Allgemein	Parent No. was added. The new value is "1080".	

Abbildung 7.4: Details eines Vorgangs in Track+

## Kapitel 8

# Entwicklung und Entwurf des Workflow-Management-Systems

Dieses Kapitel behandelt das implementierte Workflow-Management-System (WMS). Zu Beginn wird die Architektur des WMS dargestellt, später werden interessante Aspekte der Implementierung detailliert beschrieben. Abschließend werden die eingesetzten Technologien erläutert.

### 8.1 Architektur

Eine Übersicht über die Architektur des WMS ist in Abbildung 8.1 <sup>21</sup> dargestellt. Aus der Abbildung ist zu erkennen, dass das WMS aus mehreren Komponenten besteht: Die Enterprise JavaBeans EJB3 beinhalten die Geschäftslogik der am JBoss AS (8.2.3.4) publizierten Anwendungen. jBPM (8.2.3.1) dient der Prozessausführung, Hibernate (8.2.3.1.2) wird für die Persistenz der Daten verwendet, Seam (8.2.3.2), JSF (8.2.3.3) und JSP werden für die Darstellung von Webseiten im Browser benötigt. Das WMS greift auf zwei Datenbanken (WMS und ZDS, s.u.) zu. Beide können sich auf unterschiedlichen Datenbankservern befinden. Die Clients kommunizieren hauptsächlich mit dem Apache Webserver, auf welchem das ZDS installiert ist.

---

<sup>21</sup>Die Logos wurden aus folgenden Quellen entnommen: GNOME Icons <http://ftp.gnome.org/pub/gnome/sources/gnome-icon-theme/> (Lizenz: <http://www.gnu.org/licenses/gpl-2.0.html>), Clker <http://www.clker.com/clipart-mainframe-server.html>, WMS Architektur angelehnt an [http://docs.jboss.com/seam/2.1.1.GA/reference/en-US/html\\_single/](http://docs.jboss.com/seam/2.1.1.GA/reference/en-US/html_single/)

### **8.1.1 UML Diagramme**

Das Paketdiagramm sieht denkbar einfach aus und besteht aus lediglich zwei Paketen (Abbildung 8.2): `Model` und `Session`. Das Paket `Model` enthält die Datenmodelle. `Session` beinhaltet alle Interfaces sowie die dazugehörigen Action-Methoden. Das Klassendiagramm ist wesentlich komplexer als das Paketdiagramm des WMS (Abbildung 8.3). Für jede Javaklasse existiert ein dazugehöriges Interface. Dies ist ein Erfordernis für die Verwendung von JBoss SEAM bzw. JSF. Wird kein Interface für die Beans zur Verfügung gestellt, so wird von JSF zur Laufzeit eine Exception geworfen („Method not found“). Die Diagramme wurden mit Hilfe des Eclipse Plugins EclipseUML JEE 2008 ([Omo]) erstellt.





## 8.1.2 MySQL Datenbanken

Im Zuge der Implementierung war der Zugriff auf zwei Datenbanken notwendig. Einerseits wurde für das WMS eine neue Datenbank entwickelt und erstellt, andererseits musste eine Schnittstelle zu der Datenbank des ZDS angelegt werden.

### 8.1.2.1 WMS

Die Datenbank des WMS beinhaltet einerseits die Tabellen, welche vom WMS verwendet werden, andererseits jene Tabellen, welche jBPM zur Ausführung von Prozessen benötigt. Im Folgenden werden die Tabellen des WMS erklärt. Die Beschreibung betreffend die jBPM Tabellen ist dem jBPM Handbuch zu entnehmen.

**Activitylog** Diese Tabelle dient dem Verspeichern aufgetretener Ereignisse. Unter Ereignissen sind das Ändern von Zuständigkeiten, das Starten eines Prozesses, fehlerhafte bzw. erfolgreiche Loginversuche, vorzeitig beendete Prozesse oder allgemeine Fehler zu verstehen.

**Processdescription** Hier befindet sich eine Beschreibung jeder Prozessdefinition aus den jBPM Tabellen. Diese Beschreibung soll den Anwendern bei der Durchführung des Prozesses helfen, indem sie wichtige Informationen - wie etwa das Ziel des Prozesses oder die Erwartungen des Kunden - zur Verfügung stellt.

**Massnahme** Um die Implementierung eines Prozesses sicherzustellen, sollten Maßnahmen und zuständige Personen definiert werden (vgl. Kapitel 2.1.8.4). Diese Informationen werden in dieser Tabelle abgespeichert.

**Messgroesse** Um die Messung der Prozessleistung zu ermöglichen, müssen im Vorfeld Messgrößen definiert werden. Diese Messgrößen sind in dieser Tabelle abgelegt und mit dem entsprechenden Prozess verknüpft.

**Taskdescription** Ähnlich der Beschreibung einer Prozessdefinition (s.o.) existieren auch Beschreibungen für einzelne Prozessaufgaben. Diese können Informationen enthalten, welche Personen nach Abschluss der Aufgabe informiert werden sollten oder welcher Output bei der Aufgabe erzeugt werden sollte.

**Linkressource** Zu den Aufgabenbeschreibungen (Tabelle Taskdescription) können weiters beliebige Ressourcen hinterlegt werden. Diese Ressourcen werden in der Form von Verknüpfungen (auf das Dateisystem oder das Intra/Internet) abgelegt.

**Processinstancedescription** Wird im WMS ein Prozess gestartet, so wird automatisch ein Eintrag in dieser Tabelle angelegt. Er beinhaltet Informationen über die Prozessinstanz. Dazu zählen die für den Prozess verantwortliche Person, das geplante Enddatum des Prozesses sowie der Name der Prozessinstanz.

**Messung** Ist in einer Prozessinstanz ein Knoten für eine durchzuführende Messung erreicht, so kann der Anwender in dieser Tabelle die Ergebnisse des Messvorgangs abspeichern. Es können beliebig viele Messungen durchgeführt werden. Erst wenn eine Messung als „final“ gekennzeichnet wird, ist es für die aktuelle Prozessinstanz nicht mehr möglich, Messungen für diese Messgröße zu verspeichern.

**Nodeactormapping** Diese Tabelle dient der Definition von Zuständigkeiten. Beim Starten eines Prozesses werden hier alle Aufgabenblöcke mitsamt den dafür zuständigen Personen angelegt. Während der Laufzeit wird dann die Zuständigkeit für eine Aufgabe hier ausgelesen.

**Namvorlage** Um zu verhindern, dass Anwender beim Starten von Prozessen die Zuständigkeiten immer neu eingeben müssen, existiert die Möglichkeit, eine Vorlage für die Zuständigkeiten der Prozessdefinitionen in dieser Tabelle abzuspeichern.

**Usergroup** Da das Berechtigungskonzept des WMS auf Benutzerrollen basiert, können diese hier definiert werden.

**Usergroupmapping** Die Tabelle Usergroupmapping enthält die Verbindung zwischen Benutzerrolle und Nutzern.

### 8.1.2.2 Zeitdokumentation

Da das WMS auf gewisse Tabellen des *Zeitdokumentationssystems (ZDS)* zugreift, sind diese hier kurz beschrieben.

**Aktion** Dies ist die umfangreichste Tabelle dieser Datenbank. Hier werden alle Tätigkeiten der Anwender verspeichert. Diese Tabelle beinhaltet zwei Spalten, welche es ermöglichen, Aktionen zu einem bestimmten Prozess bzw. zu einer Aufgabe abzuspeichern.

**User** Diese Tabelle enthält die Daten der Anwender (Benutzername, Passwort, Name, ...).

**Kunden** Hier sind alle Kunden der Organisation abgelegt.

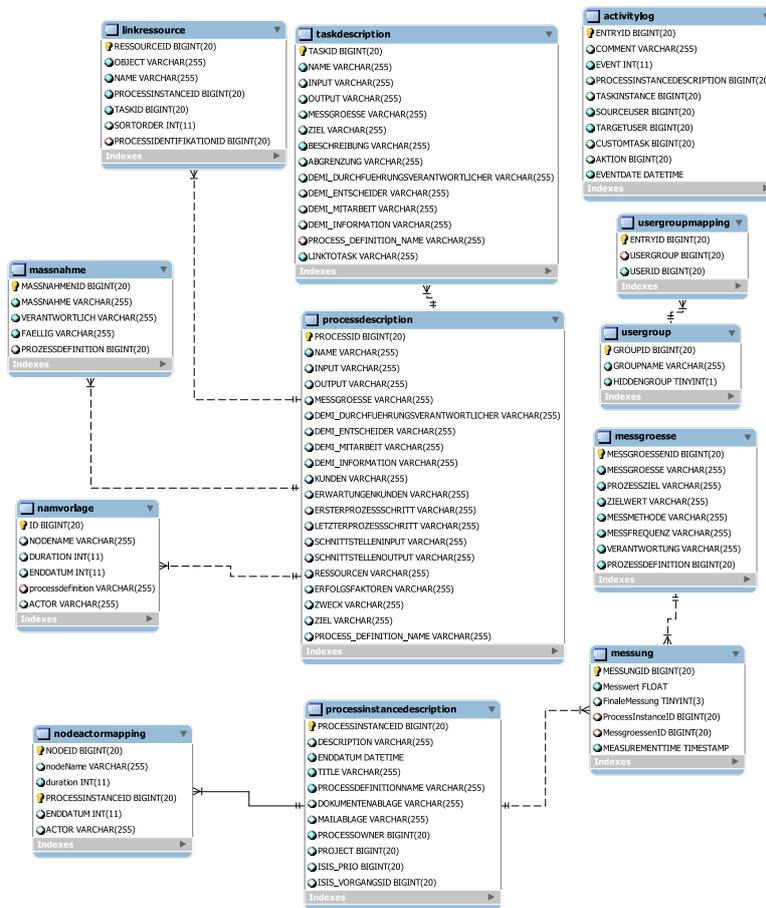


Abbildung 8.4: Diagramm der Datenbank des WMS

**Projekt** Alle bereits abgeschlossenen sowie aktiven Projekte sind in dieser Tabelle gespeichert. Zu jedem Projekt existieren mehrere mögliche Vorgänge.

**Vorgang** Vorgänge, welche bei den Aktionen angegeben werden müssen, sind hier abgelegt. Vorgänge beinhalten Tätigkeiten wie Organisation, Schulung, Projektmanagement, Programmierarbeiten etc.

### 8.1.3 Abläufe im WMS

Die Arbeitsabläufe im WMS gliedern sich in mehrere Punkte. Zuerst muss eine Prozessdefinition erstellt werden, welche anschließend von einem Benutzer gestartet werden kann. Die daraus entstehende Prozessinstanz generiert Aufgaben für verschiedene Anwender, welche in der jeweiligen Aufgabenliste aufscheinen. Sie

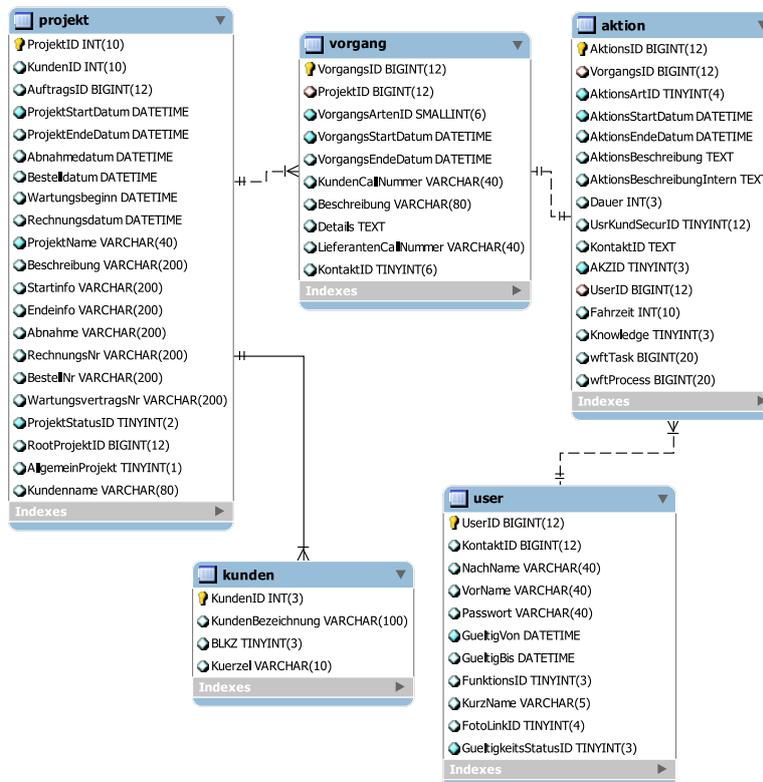


Abbildung 8.5: Diagramm der Datenbank des Zeitdokumentationssystems

können von den Anwendern Schritt für Schritt abgearbeitet werden. Am Ende eines Prozesses ist meistens eine Messung erforderlich, welche von dem Prozessverantwortlichen durchzuführen ist. Parallel zu diesen Punkten hat das Management der Organisation stets die Möglichkeit, sich anzeigen zu lassen, welche Prozesse im Moment aktiv sind und welche Anwender welche Tätigkeiten durchführen. Außerdem können Prozess- und Aufgabendefinitionen, welche zur Laufzeit angezeigt werden können, verwaltet werden.

## 8.2 Implementierung

### 8.2.1 Berechtigungskonzept

Die Benutzerrechte werden im WMS aufgrund der Zugehörigkeit zu bestimmten Rollen (s.u.) vergeben. Anwender können Mitglied einer oder mehrerer Rollen sein:

**Admin** Administratoren haben Zugriff auf alle Bereiche des WMS, insbesondere

der Verwaltung der Benutzerrollen.

**Management** Das Management hat die Berechtigung, den Status der aktiven Prozesse/Aufgaben einzusehen sowie Prozess- und Aufgabendefinitionen zu erstellen.

**Designer** Designer dürfen Prozess- und Aufgabendefinitionen erstellen und bearbeiten.

**StdUser** Die Standarduser haben keinen Zugriff auf die Verwaltungsfunktionen des WMS. Sie können lediglich Prozesse starten, Aufgaben verwalten und die Details zu den ihnen zugewiesenen Aufgaben einsehen.

Die Rollen werden im WMS auf der Seite `userGroups.seam` verwaltet. Dort können den Rollen Admin, Management und Designer beliebige Anwender zugewiesen werden.

## 8.2.2 Implementierung und Integration des WMS in das ZDS

Hier wird die technische Umsetzung der Abläufe näher beschrieben. Aufgrund der Anforderungen muss das entwickelte WMS aus der Zeitdokumentationssoftware (ZDS) steuerbar sein. Deswegen wurden viele Funktionen des WMS direkt in die Oberfläche des ZDS integriert. Der Aufruf von WMS-Funktionen erfolgt durch den Aufruf spezieller Links (mit verschiedenen Parametern) über Javascript-Funktionen. Das ZDS besteht aus einer Hauptseite (`ze.php`), welche einerseits das Interface zur Dokumentation von Zeitmarken und andererseits die Prozessaufgaben beinhaltet. Für die Integration des WMS wurden zwei zusätzliche PHP Dateien (`WFT_Integration.php` und `process_functions.php`) erstellt sowie die Hauptseite erweitert: `WFT_Integration.php` enthält eine Zusammenfassung aller Links und Parameter, welche Funktionen des WMS aufrufen. Die Datei `process_functions.php` beinhaltet Hilfsfunktionen, welche beispielsweise die Prozessaufgabenliste für die Anwender erstellt.

```
1 <?php
   $wft_url="http://localhost:8080/WorkflowManagement/";
3   $wft_task_end_trans=$wft_url."external_task_actions.seam?AufgabenID=%d&
   Aktion=end&Weg=%s";
   $wft_task_end=$wft_url."external_task_actions.seam?AufgabenID=%d&Aktion=
   end";
5   $wft_task_start=$wft_url."external_task_actions.seam?AufgabenID=%d&
   Aktion=start";
   $wft_login=$wft_url."externalLogon.seam?usr=%s&pwd=%s";
7   $wft_measurement=$wft_url."measurement.seam?processInstanceID=%d&TokenID
   =%d";
   $wft_process_start=$wft_url."newProcess.seam?ProjektID=%d&VorgangsID=%d&
   Prio=%d";
```

```

9   $wft_task_details=$wft_url."task_details.seam?AufgabenID=%d&ProzessID=%d
   ";
   $wft_task_reassign=$wft_url."external_task_actions.seam?AufgabenID=%d&
   NewOwner=%s&Aktion=reassign";
11  $wft_process_edit=$wft_url."processResponsibilities.seam?Prozess=%d&
   Aktion=edit";
   $wft_management=$wft_url."management.seam";
13  ?>

```

Listing 8.1: Liste aller Links und Parameter zur Steuerung von WMS-Funktionen

Um den Wartungsaufwand zu minimieren, wurde für das WMS keine eigene Benutzerdatenbank angelegt, es wird stattdessen die Benutzerdatenbank der ZDS verwendet.

```

1  function pageReload2(fenster){
   if(fenster.closed == true) window.location.reload();
3  else window.setTimeout(function(){pageReload2(fenster);},500);
   }

```

Listing 8.2: Neu laden des Hauptfensters des ZDS

### 8.2.2.1 Login

Der erstmalige Login eines Nutzers erfolgt über den Aufruf eines speziellen Links in einem iFrame des ZDS. Da das WMS auf die Benutzerdaten des ZDS zugreift, ist dieser Vorgang problemlos möglich. Dieses iFrame wird mittels einer Javascript-Funktion alle 300000 ms (fünf Minuten) neu geladen um sicherzustellen, dass der Benutzer im WMS angemeldet bleibt.

```

function refreshiFrame(){
2  if( typeof( window[ 'refresh' ] ) != "undefined") clearTimeout( refresh );
   document.getElementById( 'wmsiframe' ).src=document.getElementById( '
   wmsiframe' ).src;
4  refresh=window.setTimeout( 'refreshiFrame()',300000);
   }

```

Listing 8.3: Aufrechterhalten der Anmeldung am WMS

```

<iframe id="wmsiframe" src="<?php echo sprintf( $wft_login ,
   getUserIDForString(),getUserIDForString()); ?>" frameborder="0"
   width="0px" height="0px" style="visibility:hidden;"></iframe>

```

Listing 8.4: iFrame für Schnittstelle zu WMS

Der Aufruf des Login-Links führt dazu, dass basierend auf einer Regel, definiert in der Datei pages.xml, die Methode `LoginAction.externalLogon()` (Listing 8.5) aufgerufen wird. Diese Methode überprüft, ob der Anwender bereits am System angemeldet ist und führt im Falle, dass keine Anmeldung vorliegt, eine Berechtigungsprüfung durch (`isAllowedToLogin()`). Innerhalb dieser Methode werden Benutzername und Passwort überprüft sowie Benutzerrollen zugewiesen.

```

1  public String externalLogon(){
2      if (Identity.instance().isLoggedIn()) {
3          log.debug("Refreshed login for "+user.getKurzName());
4          return "empty";
5      }
6      if (isAllowedToLogin(new User(ext_pwd, ext_user))){
7          log.debug("User logged on from external: "+ext_user);
8          return "empty";
9      }
10     else {
11         return "externalLogon";
12     }
13 }

```

Listing 8.5: Externer Login

Während des Loginvorganges wird aus den Datenbanktabellen `Usergroup` und `Usergroupmapping` die Zugehörigkeit des Benutzers zu Gruppen ausgelesen. Gemäß dem Berechtigungskonzept (siehe Kapitel 8.2.1) werden dem Nutzer bestimmte Rollen zugewiesen.

### 8.2.2.2 Prozess starten

Mittels eines Buttons auf der Hauptseite des ZDS ist es jedem Benutzer möglich, eine der hinterlegten Prozessdefinitionen zu starten. Ein Klick auf diesen Button öffnet per JavaScript ein neues Browserfenster, dessen Inhalt direkt vom WMS erzeugt wird. Dem Browserfenster werden die Parameter „ProjektID“, „VorgangsID“ sowie „Prio“ übergeben, welche für das spätere Verspeichern von Aktionen zu Prozessaufgaben notwendig sind.

```

1  function wftNewProcess(){
2      projekt=window.document.erfass.projekt.value;
3      vorgang=window.document.erfass.vorgangsart.value;
4      prio=window.document.erfass.priority.value;
5      proc=window.open("<?php echo $wft_url;?>newProcess.seam?ProjektID="+
6          projekt+"&VorgangsID="+vorgang+"&Prio="+prio, 'newProcess', '
7          scrollbars=yes, status=no, width=880, height=400, resize=yes');
8      proc.focus();
9      pageReload2(proc);
10 }

```

Listing 8.6: Prozess starten

Im Browserfenster werden innerhalb einer `DataTable` alle verfügbaren Prozessdefinitionen angezeigt.

```

<h:dataTable value="#{processDefinitionList}" var="ta" styleClass="
bodyline">

```

Listing 8.7: Anzeige aller Prozessdefinitionen

Diese Liste wird in der Methode `ProcessSystemAction.getDefinitions()` erstellt, welche wiederum mit Hilfe der jBPM Methode `jbpmContext .getGraphSession().findLatestProcessDefinitions()` eine Liste der aktuellsten Prozessdefinitionen ausliest. Der Benutzer hat nun die Möglichkeit, mittels des in der Webseite eingebauten Command-Buttons „Starten“ eine der aufgelisteten Prozessdefinitionen auszuwählen. Entsprechend der Auswahl wird der Benutzer zu der Seite `processDetails.seam` weitergeleitet, auf welcher der Titel der Prozessinstanz, ein Prozessverantwortlicher sowie das geplante Enddatum des Prozesses angegeben werden müssen. Sind diese Eingaben korrekt erfolgt, werden dem Benutzer auf der Seite `processResponsibilities.seam` alle Aufgabenblöcke der gewählten Prozessdefinition angezeigt. Die genannte Seite enthält auch Aufgabenblöcke, welche sich in einem in die Prozessdefinition eingebundenen Subprozess befinden. Die Liste wird mit Hilfe der rekursiven Methode `ProcessDetailsAction.getSubprocessNodes(ProcessDefinition p)` erzeugt, welche für jeden Knoten der übergebenen Prozessdefinition ausliest, ob es sich um einen Subprozess handelt und im Falle dessen, sich selbst mit der gerade gefundenen Prozessdefinition aufruft.

Zu jedem gefundenen Aufgabeblock kann auf der Seite `processResponsibilities.seam` eine verantwortliche Person, die geplante Dauer sowie das Datum, bis zu welchem der Aufgabeblock abgeschlossen sein muss, angegeben werden (Objekt `NodeActorMapping`). Diese Daten werden für die Erstellung der Aufgabenlisten benötigt. Auf dieser Seite existiert ebenfalls eine `select-BooleanCheckbox` namens „Als Vorlage speichern“. Wird diese aktiviert, so werden die soeben eingegebenen Daten als Vorlage gespeichert und beim nächsten Start der gleichen Prozessdefinition automatisch geladen.

Ein Klick auf „Prozess starten“ ruft die Methode `ProcessResponsibilitiesAction.newProcess()` auf. Diese verspeichert die Beschreibung der Prozessinstanz (`ProcessInstanceDescription`), die `NodeActorMappings` und sie startet mittels der jBPM-Funktionen einen neuen Prozess:

```
1 org.jbpm.graph.exe.ProcessInstance p = jbpmContext.getGraphSession().
  findLatestProcessDefinition(selectedDefinitionName).
  createProcessInstance();
```

Listing 8.8: Starten einer Prozessinstanz

### 8.2.2.3 Aufgabenliste

Im Zuge der Integration des WMS in das ZDS wurde dort eine Liste für Prozessaufgaben integriert. Sie beinhaltet eine Priorität, einen Fälligkeitszeitpunkt, den Namen als auch Aktionen zu den Prozessaufgaben.

**Priorität** Diese wird beim Starten eines Prozesses automatisch an das WMS übergeben, siehe Kapitel 8.2.2.2.

**Fälligkeitszeitpunkt** Wird automatisch aus dem geplanten Enddatum des Prozesses ( $eP$ ) sowie der geplanten Dauer des Aufgabenblocks ( $d$ ) und dem Ende des Aufgabenblocks ( $eA$ ) errechnet: Fälligkeitszeitpunkt =  $eP - (d + eA)$ .

**Name** Ist eine Kombination aus dem Namen der Prozessdefinition, dem Prozesstitel und dem Aufgabennamen.

**Aktionen** Aktionen beinhalten das Starten bzw. Stoppen, Ändern der Zuständigkeit der Aufgabe sowie das Bearbeiten der Prozessinstanz.

Die Prozessaufgaben werden mit Hilfe der Funktion `getAufgaben()` ausgelesen (Listing 8.9):

```
1 function getAufgaben() {
2     include_once("resources/inc/secure.php");
3     include_once("resources/inc/mysql.php");
4     include_once("resources/inc/time.php");
5     global $mylnk, $mylnk_process;
6     $tasks=array();
7     $sids=array();
8
9     $query=sprintf("select id_, procinst_ from jbpm_taskinstance where
10         actorid_ = '%s' and end_ is null",mysql_real_escape_string(
11             getUserIDForString()));
12     $result=mysql_query($query, $mylnk_process);
13     $rows=mysql_num_rows($result);
14
15     for ($i=0;$i<$rows;$i++){
16
17         $row=mysql_fetch_array($result);
18         $query_processTasks=sprintf("SELECT ti.ID_ as AufgabenID, ti.Name_ as
19             AufgabenName, ti.Actorid_ as Bearbeiter, ti.Start_ as StartDatum,
20             ti.End_ as EndeDatum, pi.processinstanceid as Prozess,
21             pi.enddatum as ProzessEnde, pi.Prio as Prioritaet, pi.Vorgangsid as
22             VorgangsID, pi.project as ProjektID, na.duration as
23             GeplAufgabenDauer, na.enddatum as GeplAufgabenEnde, pi.title as
24             ProzessTitel,
25             pi.processdefinitionname as ProzessDefinition, pi.processowner as
26             ProzessEigner
27         FROM ticket.jbpm_taskinstance ti, ticket.processinstancedescription
28             pi, ticket.nodeactormapping na, ticket.jbpm_task ta, ticket.
29             jbpm-node nod
30         where na.nodename=nod.name_ and ta.id_=ti.task_ and ta.tasknode_=nod
31             .id_ and pi.processinstanceid=na.processinstanceid and pi.
32             processinstanceid='%d' and ti.id_='%d'",
33             mysql_real_escape_string(getRootProcessID($row[1])),
34             mysql_real_escape_string($row[0]));
35
36         $result_processTasks = mysql_query($query_processTasks, $mylnk_process)
37             ;
38     }
39 }
```

```

25     $n_rows_processTasks = mysql_num_rows($result_processTasks);
    $taskrow=mysql_fetch_array($result_processTasks);
    array_push($tasks,$taskrow);
27 }
29 usort($tasks,"cmp_wft"); //Sortierung der Aufgaben nach Fälligkeit
    return $tasks;
31 }

```

Listing 8.9: Auslesen der Prozessaufgaben für einen Anwender

### 8.2.2.3.1 Starten/Stoppen von Aufgaben

Wurde eine Prozessaufgabe noch nicht gestartet, so kann diese mittels des Buttons „Aufgabe starten“ gestartet werden. Eine JavaScript Funktion ruft einen Link auf das WMS auf, welcher die AufgabenID sowie die Art der Aktion (in diesem Fall „start“) enthält. Im WMS wird daraufhin die Methode `ExternalTaskAction.externalTaskActions()` aufgerufen. Diese Methode liest aus der jBPM Datenbank die zur AufgabenID gehörige Aufgabeninstanz (`org.jbpm.taskmgmt.exe.TaskInstance`) mittels `TaskInstance ti = jbpContext.getTaskInstance( AufgabenID );` aus. Diese Aufgabeninstanz wird dann gestartet (`ti.start()`).

Ist eine Prozessaufgabe bereits gestartet, so kann diese jederzeit beendet werden. Enthält der Aufgabenblock mehrere mögliche Übergänge zu folgenden Prozessaufgaben, so kann davon einer ausgewählt werden. Analog zum Starten einer Prozessaufgabe wird wiederum mittels JavaScript das WMS aufgerufen und die Aufgabe dort beendet (`ti.end()`).

### 8.2.2.3.2 Editieren der Prozessinstanz

Über den Link „Edit“ kann von berechtigten Personen (Prozessverantwortliche, Administratoren, Management) eine Prozessinstanz verändert werden. Es können alle Daten, welche beim Starten des Prozesses eingegeben wurden, bearbeitet werden.

### 8.2.2.3.3 Aufgabendetails

Da zu jedem Prozess als auch zu jeder Aufgabe Dokumentation hinterlegt werden kann, können diese Informationen auf der Hauptseite des ZDS jederzeit eingesehen werden. Ein Klick auf den Namen der Prozessaufgabe öffnet ein Fenster des WMS (`task_details.seam`). Im WMS wird gleichzeitig die Methode `ExternalTaskAction.initDetails()` aufgerufen. Diese Methode initialisiert basierend auf den beim Fensteraufruf übergebenen Parametern die Objekte für das Fenster „Aufgabendetails“. Nach dem Ausführen der Methode wird der Inhalt des Fensters geladen. Dieser besteht aus Informationen zu der Aufgabe (Titel,

Status, Beschreibung, verknüpfte Ressourcen, ...), der Prozessinstanz (Titel, Prozessdefinition, Prozessverantwortlicher, Beschreibung, geplantes Enddatum), der Prozessdefinition, dem Kunden (Name und Kurzbezeichnung) sowie dem Projekt (Name und Beschreibung). Zusätzlich werden parallele Aufgaben aufgelistet: In dieser Liste werden alle gleichzeitig durchzuführenden Aufgaben (auch von anderen Aufgabenblöcken) dargestellt. Name, Status (Zugewiesen, In Bearbeitung, Beendet), durchführender Anwender als auch die zugehörigen Zeitstempel (Beginn und Ende) werden angezeigt. Diese Liste wird von der Factory Methode `ExternalTaskActions.getParallelTasks()` erstellt:

```

1  public void getParallelTasks () {
    parallelTasks=em.createQuery("select ti from org.jbpm.taskmgmt.exe.
        TaskInstance ti , ProcessInstanceDescription pi , NodeActorMapping
        na , org.jbpm.taskmgmt.def.Task ta , org.jbpm.graph.def.Node no
3  WHERE " +
        "na.nodeName = no.name and ti.task = ta and ta.taskNode = no and
        pi.processInstanceID = na.processInstanceID and pi.
        processInstanceID = :id and ti.actorId <> :actor order by ti.
        end").setParameter("actor", login.getUser().getKurzName()).
        setParameter("id", processInstanceDescription.
        getProcessInstanceID()).getResultList();
    }

```

Listing 8.10: Anzeige paralleler Prozessaufgaben

Außerdem werden Informationen zu den Messgrößen (Auffistung aller zur Prozessdefinition vorhanden Messgrößen) und zur Prozessinstanz bereits durchgeführte Messungen angezeigt.

Ferner besteht die Möglichkeit, ein GANTT-Diagramm anzuzeigen. Dieses GANTT-Diagramm beinhaltet zwei Gruppen von Balken:

**Geplanter Prozessablauf** Basierend auf den Daten der NodeActorMappings (gepl. Dauer und Ende der Aufgabenblöcke) des Prozesses wird der geplante Ablauf des Prozesses visualisiert. Darunter wird der tatsächliche Prozessablauf dargestellt.

**Tatsächlicher Prozessablauf** Der tatsächliche Prozessablauf wird aus den Zeitmarken der Aufgabeninstanzen ausgelesen.

Aus diesem Diagramm lässt sich gut ablesen, welche Aufgabenblöcke vor/nach dem geplanten Start/Ende begonnen bzw. beendet wurden.

Zur Erstellung des GANTT-Diagramms wird auf die Bibliothek `JFreeChart` ([Lim]) zugegriffen. Der Methode `ChartTest.createImage( ProcessInstanceDescription pid)` wird die aktuelle Prozessinstanz übergeben. In der Methode werden die NodeActorMappings ausgelesen und zwei Gruppen von Balken erstellt

( `TaskSeries` ). Zu der einen Gruppe werden die geplanten Aufgaben hinzugefügt, zu der anderen die tatsächlichen Aufgabeninstanzen. Beide Gruppen werden in einer `TaskSeriesCollection` vereint und an `JFreeChart` übergeben:

```
2   final JFreeChart chart = ChartFactory.createGanttChart(  
3       title , // chart title  
4       yAchse , // domain axis label  
5       xAchse , // range axis label  
6       dataset , // data  
7       true , // include legend  
8       true , // tooltips  
9       false // urls  
10  );  
11  byte [] image = ChartUtilities.encodeAsPNG(chart.createBufferedImage(  
12      width , height));
```

Listing 8.11: Erstellen eines GANTT-Diagrammes

Anschließend kann das GANTT-Diagramm als PNG Bild ausgegeben werden:

```
<s:graphicImage value="#{chartTest.image}" alt="GANTT-Diagramm"/>
```

#### 8.2.2.4 Messung durchführen

Um das Messen der Prozessleistung zu unterstützen, kann in eine Prozessdefinition ein Messknoten eingebaut werden. Wird im Prozessablauf dieser Knoten erreicht, so scheint eine neue Aufgabe „Messung“ in der Aufgabenliste des Prozessverantwortlichen auf. Mit einem Klick auf den Button „Messung durchführen“ wird ein Fenster des WMS gestartet ( `measurement.seam` ), welches einerseits die zur Prozessdefinition vorhandenen Messgrößen, andererseits die bereits durchgeführten Messungen anzeigt. Neben jeder Messgröße existiert ein Button „Messen“. Wird dieser angeklickt, öffnet sich ein von `RichFaces` erzeugtes Panel. Darin kann der Messwert eingetragen werden. Weiters kann die Messung hier als „final“ gekennzeichnet werden. Dies führt dazu, dass keine weiteren Messungen zu der Messgröße eingetragen werden können.

#### 8.2.2.5 Verwaltung

In der Verwaltung können Prozess- und Aufgabendefinitionen angelegt und bearbeitet sowie der Status der laufenden Prozesse eingesehen werden. Diese Informationen werden in den Aufgabendetails (siehe Kapitel 8.2.2.3.3) angezeigt. Ein Klick auf den Button „Verwaltung“ öffnet die Seite `management.seam` des WMS.

##### 8.2.2.5.1 Prozessdefinition anlegen

Über die Seite `newProcessDescription` und der dazugehörigen Javaklasse `CreateProcessDescriptionAction.java` können Prozessdefinitionen verändert werden. Dazu gehört das Anlegen von Messgrößen.

### 8.2.2.5.2 Aufgabendefinition anlegen

Analog zu Kapitel 8.2.2.5.1 können mittels der Webseite `newTaskDescription` und der dazugehörigen Javaklasse `CreateTaskDescriptionAction.java` Aufgabendefinitionen verändert werden. Zu jeder Aufgabe muss eine Verknüpfung zur dazugehörigen Prozessdefinition und dem darin enthaltenen Aufgabenblock angelegt werden. Dazu wird mittels GPD (siehe Kapitel 8.2.3.1.1) in der Beschreibung des Aufgabenblocks ein bestimmter Schlüssel (beispielsweise eine Zahl) gefolgt von einem Bindestrich angegeben. Dieser Schlüssel muss ebenfalls in der Aufgabenbeschreibung angegeben werden. Zur Laufzeit werden dann dieser Schlüssel samt der Prozessdefinition ausgelesen und dementsprechend die richtige Aufgabenbeschreibung zu einer Aufgabe angezeigt.

Zu jeder Aufgabe können beliebig viele Ressourcen angelegt werden. Ressourcen bestehen aus einem Namen und einer URL <sup>22</sup>. Sie dienen dazu, dem Anwender zusätzliche Informationen (auf einer Webseite, in beliebigen Dokumenten) zur Laufzeit zur Verfügung zu stellen.

### 8.2.2.5.3 Laufende Prozesse

Besonders für das Management ist es interessant zu wissen, welche Prozesse zu einem gewissen Zeitpunkt aktiv sind. Die Seite `runningProcesses.seam` und die dazugehörige Klasse `ProcessDetailsAction.java` stellen alle laufenden Prozesse dar. Zu jedem Prozess werden die Prozessdefinition, der Prozesstitel, der Prozessverantwortliche, die aktiven Aufgabenblöcke, der Status und weitere Informationen angezeigt. Der Prozessstatus wird aus den aktiven Aufgabenblöcken errechnet. Als Grundlage dienen die `NodeActorMappings`, welche das geplante Enddatum für die Aufgabenblöcke beinhaltet. Die Methode `ProcessDetailsAction.getTimeStatusForTaskNode( ProcessInstanceDescription pi)` errechnet den Status für die einzelnen Prozessinstanzen:

```
1 public String getTimeStatusForTaskNode( ProcessInstanceDescription pi){
2     List<Node> nodes=activeNodes( pi );
3     Map<Long, NodeActorMapping> mapNam=new HashMap<Long, NodeActorMapping >()
4         ;
5     List<NodeActorMapping> nam;
6     NodeActorMapping currentNam;
7     long days_late=0;
8     Calendar now= new GregorianCalendar();
9     Calendar end = new GregorianCalendar();
10    if ( pi . getEnddatum () == null ) return "Unbekannt";
```

<sup>22</sup>Uniform Resource Locator. Eine Verknüpfung zu Dateien in einem Dateisystem bzw. Webseiten

```

11     end.setTime(pi.getEnddatum());
    nam=em.createQuery("Select n from NodeActorMapping n where n.
        processInstanceID=:pid").setParameter("pid", pi.
        getProcessInstanceID()).getResultList();
13     for(int i=0;i<nam.size();i++){
        mapNam.put(nam.get(i).getNodeID(), nam.get(i));
    }
15     for(int i=0;i<nodes.size();i++){
        end.setTime(pi.getEnddatum());
17         if(!mapNam.containsKey(nodes.get(i).getId())) log.warn("Node with ID
            "+nodes.get(i).getId()+" for PID "+pi.getProcessInstanceID()+"
            not found in NodeActorMapping");
        else{
19             currentNam=(NodeActorMapping)mapNam.get(nodes.get(i).getId());
            end.add(Calendar.DAY_OF_MONTH, -currentNam.getEnddatum()-
                currentNam.getDuration());
21             long o = end.getTimeInMillis()-now.getTimeInMillis();
            if(o<days_late) days_late=o/1000/60/60/24; // days
23         }
    }
25     if(days_late >0) return days_late+" Tage vorraus";
27     if(days_late <0) return Math.abs(days_late)+" Tage zurueck";
    if(days_late ==0) return "Puenktlich";
29     else return "";
}

```

Listing 8.12: Status der Prozessinstanzen

Wie auch in Kapitel 8.2.2.3.3 lässt sich zu jeder Prozessinstanz ein GANTT-Diagramm generieren. Zusätzlich kann auch ein GANTT-Diagramm über alle aktiven Prozesse gezeichnet werden. Dieses beinhaltet sowohl den geplanten Prozessstart/das geplante Prozessende als auch den tatsächlichen Prozessstart/das tatsächliche Prozessende.

**8.2.2.5.4 Laufende Aufgaben** Ähnlich zu den aktiven Prozessinstanzen können auch die im Moment in Bearbeitung befindlichen Prozessaufgaben angezeigt werden. Es ist hier möglich zu sehen, welcher Person welche Aufgaben zugewiesen sind. Darüber hinaus kann die Zuständigkeit geändert werden.

### 8.2.3 Eingesetzte Technologien

Dieses Unterkapitel soll dem Leser einen Überblick über die wichtigsten eingesetzten Technologien bieten.

### 8.2.3.1 JBoss jBPM 3.2.3

Das Framework jBPM wird von JBoss <sup>23</sup> entwickelt und bietet die Möglichkeit, beliebige Arbeitsabläufe in der Java EE Plattform abzubilden. jBPM unterstützt dabei verschiedene „Prozessausführungssprachen“, in denen Prozessdefinitionen verfasst werden können und die alle XML basierend sind:

**jPDL** jPDL steht für „Java Process Definition Language“. jPDL ist einfach zu modellieren und bietet Java-Integration mit der Möglichkeit, Aufgaben zu erstellen und zu verwalten. jPDL basiert auf einem gerichteten Graphen, welcher neben Aufgaben und Steuerungselementen (Joins/Forks, Subprozesse) auch Aktionen beinhaltet, welche es ermöglichen, im Hintergrund beliebigen Java-Code auszuführen. Diese Sprache wird in dem im Rahmen dieser Masterarbeit entwickelten Workflow-Management-System eingesetzt. Siehe Kapitel 7.1

**BPEL** BPEL steht für „Business Process Execution Language“ und ist eine Prozessausführungssprache für Web-Service Schnittstellen. BPEL wurde von OASIS <sup>24</sup> standardisiert.

**SEAM Pageflow** Diese Sprache ähnelt jPDL, wird aber im Gegensatz zu jPDL nicht für beliebige Prozesse verwendet, sondern für die Navigation auf JBoss SEAM basierten Webseiten. Bei SEAM Pageflows kann, basierend auf dem aktuellen Status der Webanwendung, definiert werden, welche Webseite als nächste angezeigt werden soll. Mit Hilfe eines grafischen Editors kann ein Entwickler auf einen Blick den Weg eines Anwenders durch die Webapplikation erkennen.

Für das Generieren von Prozessdefinitionen beinhaltet das Framework jBPM das Modul jPDL Designer (siehe 8.2.3.1.1). Dies ist ein grafischer, auf der Entwicklungsumgebung Eclipse basierender Editor, in dem der Benutzer zwischen der grafischen Darstellung der Prozesse sowie der zugrunde liegenden XML Datei umschalten kann. jBPM kann als Stand-Alone Lösung oder integriert in einen JBoss Applikationsserver verwendet werden. Eine Übersicht über die einzelnen Module zeigt Abbildung 8.6.

**8.2.3.1.1 jPDL und der grafische Editor (jPDL Designer, GPD)** Je Prozessdefinition werden von dem GPD drei Dateien erstellt: Erstens die Prozessdefinition an sich („processdefinition.xml“), zweitens eine Datei, welche Metadaten über die

---

<sup>23</sup><http://www.jboss.com>

<sup>24</sup>Advancing open standards for the information society. <http://www.oasis-open.org/>

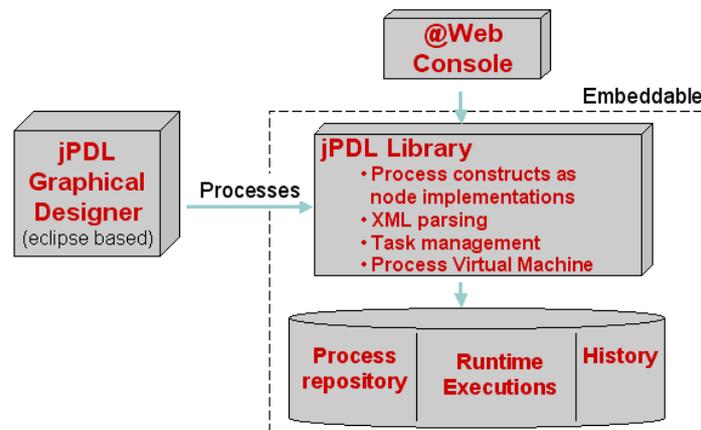


Abbildung 8.6: Module des Frameworks JBoss jBPM [JBo08a]

Prozessdefinition enthält („gpd.xml“). Zweitere gibt an, an welcher Stelle des grafischen Interfaces des GPD die einzelnen Knoten und Kanten angezeigt werden sollen. Die dritte Datei („processimage.jpg“) beinhaltet eine grafische Repräsentation der Prozessdefinition.

jPDL basiert auf einem Graphen, welcher aus Knoten und Übergängen besteht. Ein Beispiel für eine einfache Prozessdefinition ist in 8.7 abgebildet. Der Quellcode zu diesem Beispiel ist dem Listing 8.2.3.1.1 zu entnehmen. Der grafische Editor bietet mehrere Möglichkeiten (unterteilt in Tabs), eine Prozessdefinition zu bearbeiten:

**Diagramm** Mittels Drag & Drop können aus einer Werkzeugleiste alle Sprachkonstrukte der jPDL in ein Arbeitsblatt gezogen und dort miteinander verbunden werden. Wird ein Element im Arbeitsblatt markiert, können detaillierte Eigenschaften des Objekts in der Eclipse Ansicht „Eigenschaften“ bearbeitet werden. Dort ist es beispielsweise möglich, den Namen des Elements anzupassen.

**Design** In dieser Ansicht wird die XML Datei als Baum angezeigt und kann bearbeitet werden.

**Quellcode** Hier wird der reine XML Quellcode angezeigt. Diese Ansicht ist vor allem dann nützlich, wenn zu vielen Elementen bestimmte Attribute hinzugefügt werden müssen, beispielsweise ein Assignment-Handler.

Eine Prozessdefinition muss mindestens einen Start sowie einen oder mehrere Stopp-Knoten enthalten. Je nach Typ des Knotens wird bei der Ausführung von dem Framework entschieden, was passieren soll. jBPM jPDL enthält einige vor-

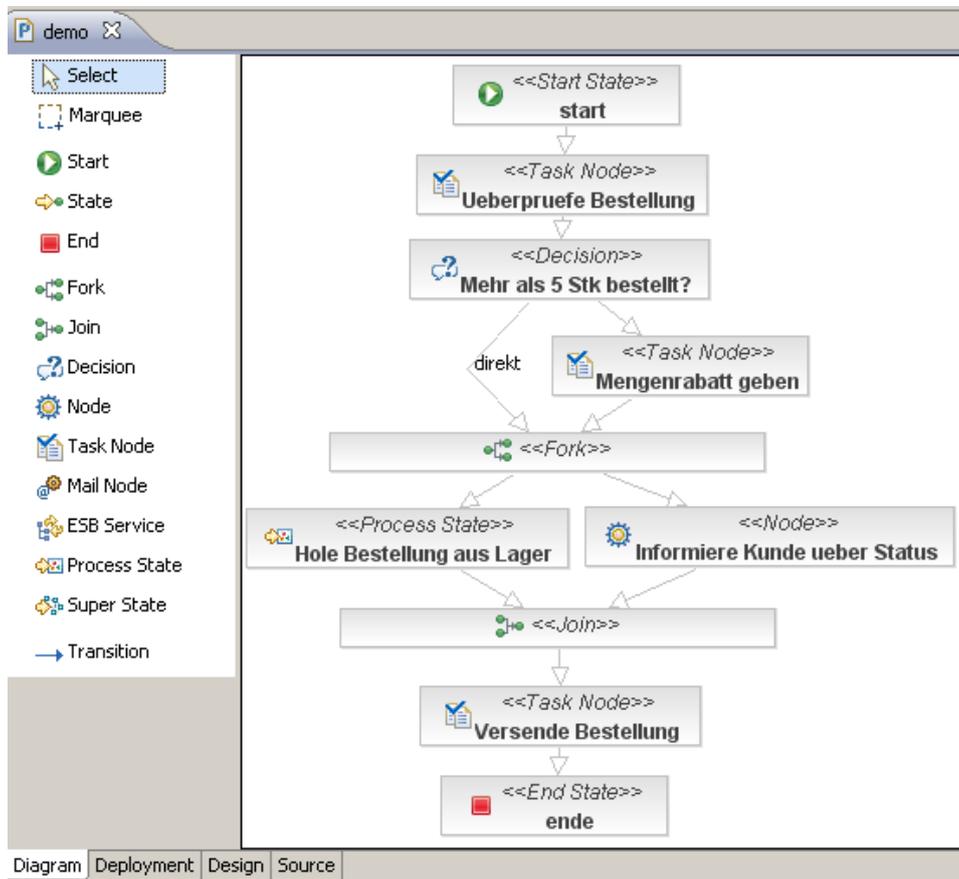


Abbildung 8.7: Grafischer Designer für jPDL (integriert in Eclipse)

definierte Knoten. Es steht dem Entwickler jedoch frei, eigene Knotentypen zu implementieren. Die wichtigsten Knoten in jPDL sind:

**Start** Beim Starten eines Prozesses wird in diesem Knoten ein Zeiger (Token) auf den aktuellen Prozessstatus erzeugt. Sind alle Tätigkeiten in einem Knoten abgeschlossen, so wandert der Token weiter zum nächsten Knoten.

**Fork** Teilt den eingehenden Token in mehrere, parallel auszuführende Token auf, welche auf unterschiedliche Knoten zeigen.

**Join** Führt mehrere Token wieder zu einem zusammen.

**Decision** Ein Decision (Entscheidungs-)Knoten hat mehrere ausgehende Übergänge. Er bietet die Möglichkeit, während der Laufzeit eine Entscheidung zu treffen, welcher Ausführungspfad gewählt werden soll. Dies kann entweder

mittels einfacher Ausdrücke in der Prozessdefinition oder per Implementierung einer Decision-Handler Methode in Java-Code passieren.

**TaskNode** Mit TaskNodes ist es möglich, Aufgaben für Benutzer zu hinterlegen. Jedes TaskNode kann beliebig viele Aufgaben beinhalten, welche alle parallel ausgeführt werden. Sind alle Aufgaben abgeschlossen, verlässt der Token diesen Knoten.

**Process States** dienen dem Aufruf eines Subprozesses. Zusätzlich zum reinen Aufruf können dem Subprozess Prozessvariablen übergeben werden.

**State** Hierbei handelt es sich um einen Warte-Zustand. Erreicht die Prozessausführung diesen Knoten, dann wird mit der Ausführung auf ein externes Signal gewartet.

**End** Sobald der Token in diesem Knoten eintrifft, wird die Ausführung des Prozesses beendet.

```
2 <?xml version="1.0" encoding="UTF-8"?>
4 <process-definition xmlns="" name="demo">
6   <start-state name="start">
8     <transition to="Ueberpruefe Bestellung"></transition>
10   </start-state>
12   <fork name="fork1">
14     <transition to="Hole Bestellung aus Lager"></transition>
16     <transition to="Informiere Kunde ueber Status"></transition>
18   </fork>
20   <task-node name="Ueberpruefe Bestellung">
22     <transition to="Mehr als 5 Stk bestellt?"></transition>
24   </task-node>
26   <node name="Informiere Kunde ueber Status">
28     <transition to="join1"></transition>
30   </node>
32   <process-state name="Hole Bestellung aus Lager">
34     <sub-process name=""></sub-process>
     <transition to="join1"></transition>
   </process-state>
   <join name="join1">
     <transition to="Versende Bestellung"></transition>
   </join>
   <task-node name="Versende Bestellung">
     <transition to="ende"></transition>
   </task-node>
```

```

36 <decision name="Mehr als 5 Stk bestellt?" expression="">
    <transition to="Mengenrabatt geben">
        <condition expression="#{anzahl ge 5: 'true' ? 'false'}"></condition
        >
38 </transition>
    <transition to="fork1" name="direkt">
40 <condition expression="#{anzahl lt 5: 'true' ? 'false'}"></condition
        >
        </transition>
42 </decision>

44 <task-node name="Mengenrabatt geben">
    <transition to="fork1"></transition>
46 </task-node>

48 <end-state name="ende"></end-state>

50 </process-definition>

```

Listing 8.13: jPDL Prozess Demo

### Wichtige Sprachelemente

Einige Sprachelemente von jPDL sind besonders erwähnenswert, da sie für die Ausführung eines Prozesses sehr wesentlich sind. Im Laufe eines Prozesses ist es oft notwendig, Entscheidungen über den weiteren Prozessablauf zu treffen. Dafür gibt es mehrere Möglichkeiten. Welche verwendet werden soll, ist davon abhängig, wer die Entscheidung zu treffen hat. Vgl. [JBo08a, Kapitel 10.3.4].

Wird die Entscheidung von einer externen Applikation getroffen, so empfiehlt es sich, von einem State-Knoten mehrere Übergänge zu den möglichen Folgezuständen in der Prozessdefinition einzutragen. Die externe Anwendung muss dann dem aktuellen Token mit *Token.signal(String transitionName)* signalisieren, welcher Übergang gewählt werden soll.

Wird die Entscheidung über den Exekutionspfad von dem Prozess getroffen, so existieren drei Möglichkeiten, wie die Entscheidung gefällt werden kann: Erstens kann in dem Decision-Knoten ein Ausdruck (beispielsweise in Expression Language [Mic06]) angegeben werden, welcher den Namen eines Überganges als Rückgabewert besitzt. Zweitens kann zu jedem Übergang eine Bedingung, welche einen booleschen Wert zurückliefern muss, hinterlegt werden. Zur Laufzeit werden von dem Decision-Knoten alle ausgehenden Übergänge auf solche Bedingungen untersucht, die Bedingungen ausgewertet und der erste Übergang genommen, bei dem die Bedingung logisch wahr ist. Die dritte Möglichkeit, eine Entscheidung über den zukünftigen Exekutionspfad zu treffen, ist, in dem Decision-Knoten einen Decision-Handler anzugeben. Dieser Handler ist eine Javaklasse, die das Interface *DecisionHandler* implementiert. Zur Laufzeit wird dann die Methode *decide(ExecutionContext executionContext)* der Javaklasse aufgerufen. Innerhalb

dieser Methode kann ein beliebiger Code verwendet werden, um ein Ergebnis zu berechnen. Die Methode muss den Namen des zu verwendenden Übergangs zurückliefern.

Sollte der Rückgabewert der hier erwähnten Methoden keinem Namen eines ausgehenden Übergangs entsprechen, so wird automatisch der erste Übergang, der in der Datei „processdefinition.xml“ eingetragen ist, verwendet.

Es ist von großem Vorteil, wenn in Prozessen klare Verantwortlichkeiten definiert sind. Sind Verantwortlichkeiten nicht definiert, kann es vorkommen, dass einzelne Prozessschritte nicht oder zu spät durchgeführt werden. jBPM unterstützt das Konzept der Verantwortlichkeiten. Zu jeder Aufgabe kann spezifiziert werden, welcher Anwender des Systems diese durchzuführen hat. jBPM bietet hierfür fünf verschiedene Wege, wie dies geschehen kann <sup>25</sup>:

**Handler** Ähnlich dem Decision Handler existiert ein Interface namens *Assignment-Handler*, in welchem beliebiger Java-Code ausgeführt werden kann, um zu errechnen, welcher Anwender für die aktuelle Aufgabe zuständig sein soll.

**Actor** Bei der Methode „Actor“ wird in der Prozessdefinition der Name (beispielsweise der Benutzername) statisch definiert.

**Pooled Actors** Ist ähnlich dem Actor mit dem Unterschied, dass hier mehrere Verantwortliche durch Kommas getrennt angegeben werden können.

**Swimmlane** Swimmlanes werden zu mehreren Aufgaben gleichzeitig angegeben. Sie repräsentieren Prozessrollen. Sobald eine Person zu einer Swimmlane zugewiesen wird, werden alle folgenden Aufgaben ebenfalls dieser Person zugeordnet.

**Expression** Dies ist ein Ausdruck, welcher die Kennung einer Person zurückliefern muss.

#### 8.2.3.1.2 Persistenz

Um den Status der aktiven/laufenden Prozesse sowie die Prozessdefinitionen zu speichern, verwendet jBPM Hibernate <sup>26</sup>. Hibernate ist ein Framework, das es erlaubt, Object-Relational Mapping (ORM) durchzuführen. Das bedeutet, es werden einfache Javaklassen (POJOs <sup>27</sup>) auf einer beliebigen relationalen Datenbank abgebildet. ORM bietet u.a. die Vorteile, dass sich der Entwickler nicht mit den datenbankspezifischen SQL Syntaxen beschäftigen muss, sondern unabhängig von

---

<sup>25</sup>Zu finden in der grafischen Darstellung des GPD: Eigenschaften des TaskNodes - Task - Tab Assignment

<sup>26</sup>[www.hibernate.org/](http://www.hibernate.org/)

<sup>27</sup>Plain Old Java Objects

der darunterliegenden Datenbank nur einen bestimmten Syntax, genannt HQL <sup>28</sup>, beherrschen muss. Außerdem lassen sich auf einfache Art und Weise (Annotationen) Beziehungen zwischen Objekten definieren. Da Prozesse für gewöhnlich sehr langlebig sind und eine Prozessinstanz mitunter mehrere Wochen lang ausgeführt wird, ist es notwendig, den jeweils aktuellen Status stets in die Datenbank zu speichern, um Datenverlust vorzubeugen. jBPM erledigt dies automatisch.

### **8.2.3.1.3 Aufgabenverwaltung**

Bei der Ausführung von Prozessen entstehen zwei Typen von Aufgaben: Erstens Aufgaben, die von einer Software durchgeführt werden müssen (z.B.: dem automatischen Verschicken von Emails), zweitens Aufgaben, welche von einem Menschen erledigt werden müssen. Zweitere werden von jBPM in der Form von TaskNodes zur Verfügung gestellt. jBPM erlaubt es, Aufgabenlisten für die einzelnen Anwender des Systems zu generieren. Aufgaben (Tasks) können entweder direkt in die Prozessdefinition geschrieben werden oder sie werden innerhalb von TaskNodes spezifiziert. Jede Aufgabe muss innerhalb der Prozessdefinition einen eindeutigen Namen haben. Zur Ausführungszeit werden aus den Aufgaben Aufgabeninstanzen (TaskInstances) erstellt. Nach dem Erstellen der Instanzen kann jede Instanz optional gestartet sowie beendet werden. Außerdem kann der Aufgabeninstanz eine für die Ausführung verantwortliche Person (Actor) zugewiesen werden. Anhand dieses Actors ist es möglich, persönliche Aufgabenlisten für die Benutzer zu erstellen. Es existiert außerdem die Möglichkeit, Aufgabenlisten für eine Gruppe von Anwendern zu erstellen. Bevor eine Person an einer Aufgabe arbeiten kann, muss sie die Aufgabe aus der Gruppen-Aufgabenliste erst in die persönliche Aufgabenliste verschieben.

### **8.2.3.2 JBoss SEAM 2.1.1.GA**

SEAM wird ebenfalls von JBoss entwickelt und ist ein Framework, welches die Interaktion zwischen der Präsentations-, Geschäftslogik- und Persistenzschicht vereinfacht. Die Präsentationsschicht wird zurzeit mittels JavaServer Faces (JSF) realisiert. Es ist jedoch möglich, auch andere Frameworks für die Anzeige zu verwenden, beispielsweise Google Web Toolkit <sup>29</sup> (GWT). Mit Hilfe von Seam ist es nicht mehr notwendig, die drei Schichten manuell über „BackingBeans“ <sup>30</sup> miteinander zu verbinden. Der Programmierer kann sich somit auf die Entwicklung der Webanwendung (Präsentation und Geschäftslogik) konzentrieren.

---

<sup>28</sup>Hibernate Query Language

<sup>29</sup><http://docs.jboss.org/seam/latest-2/reference/en-US/html/gwt.html>

<sup>30</sup>Ein BackingBean ist eine Javaklasse, die bestimmten Spezifikationen folgt und dazu dient, Daten für ein Userinterface bereitzustellen.

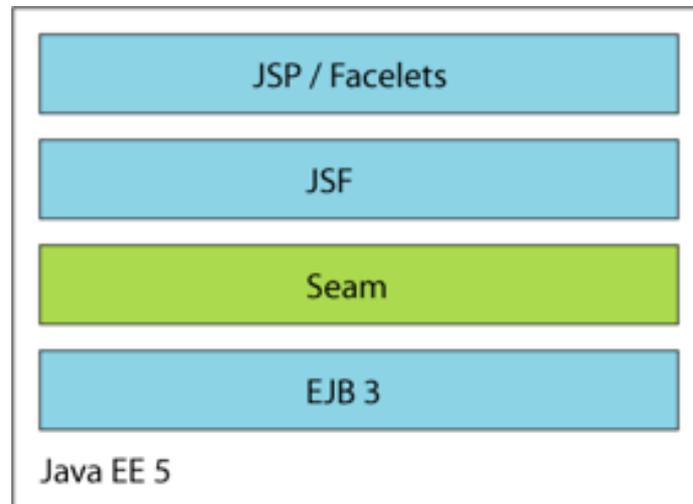


Abbildung 8.8: Übersicht über Java EE 5.0 [JBo08b]

Seam hat einige Konzepte eingeführt, welche dem Programmierer die Arbeit mit der Präsentationsschicht wesentlich erleichtern. So können mit Hilfe des Konzepts der Konversationen (auch Workspaces genannt) Daten über mehrere Seitenaufrufe hinweg zur Verfügung gestellt werden oder mit Hilfe der Bijektion auf einfache Art und Weise Daten zwischen Präsentationsschicht und Geschäftslogikschicht ausgetauscht werden. Mittels von Kontexten kann in Seam ausgedrückt werden, welchen Gültigkeitsbereich eine Seam Komponente hat. Seam definiert hierfür sieben verschiedene Kontexte: Stateless, Session, Application, Event, Page, Conversation und Business Process. Wie bereits in Kapitel 8.2.3.1 angesprochen, kann die gesamte Navigation durch Seam Webseiten in einem externen Pageflow hinterlegt werden. Dies hat den Vorteil, dass die Navigation durch die Webanwendung auf einen Blick erkennbar ist und das falls der Pageflow geändert werden muss, dies nur in einer Datei zu geschehen hat.

#### 8.2.3.2.1 Annotationen

Mit Annotationen wurden in Java 5.0 Metadaten zu Javaklassen hinzugefügt. Früher wurden Metadaten ausschließlich in XML Dateien gespeichert und bei großen Projekten entstanden sehr viele XML Dateien. Das Problem dabei war, dass bei der Laufzeit nicht überprüft werden konnte, ob der Inhalt der XML Dateien nach wie vor korrekt ist. Annotationen lösen dieses Problem, da sie direkt in den Sourcecode geschrieben werden.[NN07, S. 14 f.] Im Folgenden sind die wichtigsten von Seam verwendeten Annotationen kurz vorgestellt:

**@Name** Gibt einen Seam Komponentennamen für die aktuelle Klasse an.

**@Scope** Ermöglicht es, für die Klasse ein Kontext anzugeben.

**@In** Mit @In wird beim Laden einer Klasse eine Kontextvariable in ein Objekt injiziert.

**@Out** Funktioniert genau umgekehrt zu @In, indem ein Attribut in eine Kontextvariable geladen wird.

**@Factory** Factory Methoden werden verwendet, um Daten zu initialisieren. Wird zur Laufzeit auf ein Attribut, welches *null* ist zugegriffen und existiert eine Factory Methode mit demselben Namen, so wird diese Methode aufgerufen.

**@Logger** Bietet Log4J<sup>31</sup> Loggingfunktionalitäten in Seam Klassen.

**@DataModel** Ist ein Attribut (vom Typ `java.util.List`) mit dieser Annotation versehen, so ist es möglich, darauf aus einer JSF Seite zuzugreifen. Es kann automatisch eine Tabelle, welche als Inhalt das annotierte Attribut hat, erzeugt werden.

**@DataModelSelection** Wird auf einer JBoss Seam Webseite eine Zeile einer Tabelle ausgewählt (mittels eines command Buttons), so wird der Inhalt der aktuellen Zeile (welcher ein POJO ist) in das mit @DataModelSelection annotierte Attribut gespeichert.

**@Begin** Mit dieser Annotation wird eine Konversation gestartet. Konversationen lassen sich verschachteln und können über mehrere Seitenaufrufe hinweg bestehen.

**@End** Diese Annotation beendet jede laufende Konversation.

**@Restrict** Die Restrict Annotation ist Teil der Seam Security und dient der Validierung, ob der aktuelle Benutzer das Recht besitzt, eine mit @Restrict annotierte Methode oder Klasse aufzurufen.

#### 8.2.3.2.2 Sicherheit

Seam bietet mehrere Arten an, wie Benutzer authentifiziert bzw. Benutzerrechte gesetzt und überwacht werden können. Dazu gehören u.a.: Authentifizierung, Identitätsmanagement sowie das Rechtemanagement.

---

<sup>31</sup><http://logging.apache.org/log4j/1.2/index.html>

### **Authentifizierung**

Die Authentifizierung beschreibt in diesem Fall die durchzuführenden Aktionen, um festzustellen, ob die von einem Benutzer (durch Benutzernamen und Passwort) vorgegebene Identität korrekt ist (vgl. [Rec06, S. 253]). Für die Authentifizierung wird die von Seam bereitgestellte *Authenticator* Komponente benötigt. Sobald diese in der Datei *components.xml* definiert wurde, ist es dem Entwickler möglich, eine eigene Methode, welche die Authentifizierung übernimmt, zu programmieren. Diese sollte sich auf das Wesentliche beschränken (zu überprüfen, ob der Benutzer berechtigt ist, sich anzumelden oder nicht), da es vorkommen kann, dass diese Methode von Seam oft aufgerufen wird. Wenn nach dem erfolgreichen Login bestimmte Aktionen ausgeführt werden sollen, so kann dies mit Hilfe eines Observers geschehen. Vgl. dazu [JBo08b, Kap. 15.3.2.2]

### **Identitätsmanagement**

Das Identitätsmanagement dient der Verwaltung der Benutzer (und der dazugehörigen Rechte), welche auf eine Seam Applikation zugreifen. Das Identitätsmanagement ist unabhängig vom Identitätsspeicher, welcher meist eine externe Anwendung ist (zum Beispiel ein LDAP<sup>32</sup> Server). Zugriff auf das Identitätsmanagement erlangt der Programmierer mittels der Seam Komponente *identityManager*.

### **Rechtmanagement**

In Seam existieren mehrere Möglichkeiten, dem Benutzer den Zugriff auf Seiten/Klassen/Methoden zu erlauben oder zu verweigern. Benutzer können in dem System bestimmte Rollen haben. Beispiele für Rollen könnten „Projektverantwortlicher“ oder „Projektmitarbeiter“ sein. Rollen können Benutzern zugeordnet werden und somit ist es möglich, Benutzergruppen zu definieren. Vgl. [JBo08b, Kap. 15.6.1.1].

Zusätzlich zu Rollen gibt es in Seam das Konzept der Berechtigungen (Permissions). Eine Berechtigung besteht aus drei Elementen:

**Ziel** Das Ziel gibt das Objekt an, auf welches zugegriffen wird.

**Aktion** Unter Aktion wird die Operation verstanden, die auf dem Objekt ausgeführt wird.

**Empfänger** Der Empfänger ist die die Aktion durchführende Person.

---

<sup>32</sup>Lightweight Directory Access Protocol: Wird in s.g. Verzeichnisdiensten eingesetzt. Über eine Suchanfrage vom Client an den Server können beispielsweise Personendaten oder Konfigurationen ausgelesen werden.

Dies kann an einem einfachen Beispiel erläutert werden: Angenommen, der Anwender einer Software möchte in einer Weboberfläche einen Bug in der Software beschreiben. Dann wäre der Anwender der Empfänger, der Bug (bzw. ein Objekt Bug in der Datenbank) das Ziel und „erstellen“ wäre die Aktion.

Die Konzepte Rollen und Berechtigungen können in Seam beliebig kombiniert werden.

Soll der Zugriff auf eine Seam Komponente beschränkt werden, so muss die Annotation `@Restrict` verwendet werden. Mit `@Restrict` können Klassen sowie auch Methoden annotiert werden. Sind Methode und Klasse annotiert, so hat die Annotation der Methode Vorrang. Schlägt eine Berechtigungsüberprüfung fehl, so wird eine Exception geworfen. Neben der Möglichkeit zu überprüfen, ob der Zugriff auf eine Methode erlaubt ist, bietet Seam zusätzlich die Möglichkeit, Berechtigungsüberprüfungen innerhalb von Methoden durchzuführen (Inline restriction).

Neben den Berechtigungsüberprüfungen in dem Seam Komponenten ist es ebenfalls wichtig, in der Benutzeroberfläche Berechtigungsüberprüfungen durchzuführen. Mit Hilfe des `rendered` Attributs in den JSP Seiten können Bereiche der Benutzeroberfläche für Benutzer, die ungenügende Rechte haben, um diese zu verwenden, ausgeblendet werden. Außerdem ist es möglich, den Zugriff auf ganze Seiten für Benutzer zu sperren. Dies kann über die Datei `pages.xml` geschehen. Dort kann, ähnlich zu der Vorgehensweise in den Seam Komponenten, eine `Restrict` Anweisung definiert werden.

Falls die hier vorgestellten Möglichkeiten des Rechtemanagements nicht ausreichen, kann zusätzlich das JBoss Framework Drools (JBoss Rules)<sup>33</sup> eingebunden werden, welches es erlaubt, komplexe Regeln für das Rechtemanagement zu hinterlegen.

### 8.2.3.3 JavaServer Faces

JavaServer Faces (JSF) ist ein Java-basiertes Framework, welches erlaubt, grafische Benutzerschnittstellen für Webanwendungen zu erstellen. JSF besteht aus mehreren Bibliotheken, welche vor der Verwendung in eine Webseite eingebunden werden müssen, beispielsweise HTML-Komponenten. Die Navigation wird mittels Regeln (`from-view-id` und `to-view-id`) einer Datei definiert. Diese Regeln können `from-outcome` Elemente beinhalten, welche von Javaklassen aufgerufen werden können. Eine zu einer JSF View gehörende Java Methode muss also einen `String` zurückliefern, um auf eine andere Seite zu navigieren. [SBH06, S. 3, 11, 14]

Um AJAX-Funktionalitäten<sup>34</sup> in JSF Seiten zu integrieren, kann die Bibliothek

---

<sup>33</sup><http://www.jboss.org/drools/>

<sup>34</sup>Asynchronous JavaScript and XML

Richfaces (hier in Version 3.3.0 GA) eingebunden werden.

#### **8.2.3.4 JBoss Application Server 4.2.2.GA**

Die Frameworks SEAM, jBPM und RichFaces basieren auf dem jBOSS Application Server (AS). Deswegen wird dieser für die Implementierung und den Betrieb des WMS benötigt. Während der Implementierung wurde der JBoss Applikationsserver mittels eines Eclipse Plugins in Eclipse eingebunden, sodass dieser in einem Container gestartet werden konnte. Daraus ergaben sich für die Entwicklung einige Vorteile, da u.a. der Re-Deploy<sup>35</sup> von Änderungen sehr schnell vonstatten ging.

#### **8.2.3.5 MySQL 5.0.67 Datenbank**

Als Datenbank-Backend wird der MySQL Server eingesetzt, da die Datenbank des ZDS auch auf einem MySQL Server läuft. Der MySQL Datenbankserver ist (samt Quellcode) frei erhältlich, leicht zu konfigurieren und schnell aufzusetzen. Er wird von Sun Microsystems entwickelt ([Mic]). MySQL wird häufig in Kombination mit dem Apache Webserver und PHP eingesetzt (wie auch im ZDS). Die Firma Sun Microsystems bietet zusätzlich einige Softwareprodukte, welche den MySQL Server ergänzen, an. Dazu zählen der MySQL Administrator und die MySQL Workbench. Der MySQL Administrator ermöglicht den Zugriff auf sowie die Verwaltung (Benutzermanagement, Tabellenverwaltung, ...) von beliebigen MySQL Servern. Die MySQL Workbench dient dem Erstellen von Tabellenschemata. In einem grafischen Editor können auf einfache Art und Weise Tabellen samt Verknüpfungen (Fremdschlüssel) erstellt und anschließend als SQL Skript exportiert werden. Die MySQL Workbench ermöglicht weiters das Erstellen von ER Diagrammen aus SQL Dateien.

#### **8.2.3.6 Eclipse 3.4.0**

Eclipse ist eine integrierte Entwicklungsumgebung, welche es mittels Plugins ermöglicht, Programme in verschiedensten Programmiersprachen zu entwickeln. Im Rahmen dieser Masterarbeit wurden beispielsweise die Plugins PHP Development Tools (PDT) 2.0 und JBoss Tools verwendet. Ersteres ermöglicht die Entwicklung von PHP basierten Webseiten, zweiteres integriert JBOSS Funktionen in Eclipse.

#### **8.2.3.7 JFreeChart 1.0.12**

JFreeChart ist eine Java Bibliothek, die das Erstellen von vielen verschiedenen Diagrammtypen (Balken-, Torten-, GANTT-Diagramme, Histogramme, ...) in

---

<sup>35</sup>Das Publizieren von Änderungen im Quellcode auf den Applikationsserver

Java ermöglicht. JFreeChart wird unter der GNU Lesser General Public Licence <sup>36</sup> veröffentlicht. [Lim]

#### 8.2.4 Aufgetretene Probleme

Während der Entwicklung des WMS traten einige Probleme auf, welche es zu überwinden galt. Da auf dem Computer, auf welchem die Entwicklung stattfand, alle Server (JBoss, MySQL), die Frameworks (Seam, jBPM, Richfaces) sowie Eclipse gleichzeitig laufen mussten, ergab sich ein großer RAM Verbrauch. Vor allem der grafische Editor (Kapitel 8.2.3.1.1) hatte einen hohen Arbeitsspeicherverbrauch. Dieses Problem konnte durch die Aufstockung des Arbeitsspeichers auf vier Gigabyte gelöst werden.

Für das Testen des WMS war es oft notwendig, die Datenbank zu leeren, um ein mit einem „sauberen“ System arbeiten zu können. Unter „sauber“ ist in diesem Zusammenhang zu verstehen, dass keine sich möglicherweise beeinflussenden Daten in der Datenbank existieren. Für diese Zwecke musste ein die Abhängigkeiten der jBPM sowie der eigenen Tabellen berücksichtigendes SQL Skript entwickelt werden, welches alle Daten (Prozess-/Aufgabeninstanzen, Logs, Prozessbeschreibungen, etc.) aus der Datenbank löscht. Das SQL Skript hierfür ist dem Anhang (A.1) zu entnehmen.

Um die Datenbank einer Fremdsoftware in das WMS zu integrieren, war es notwendig, einige Hibernate Mappings zu der „fremden“ Datenbank zu erstellen. Dabei ist zu beachten, dass der Hibernate Parameter „hbm2ddl.auto“ auf keinen Fall gesetzt wird, da ansonsten die Möglichkeit besteht, dass Teile der Fremddatenbank bei einem Neustart des WMS gelöscht werden.

Am Beginn der Entwicklung wurde für die Verbindung zur Datenbank eine Hibernate TX-Datenquelle <sup>37</sup> verwendet. Dies funktionierte beim Testen problemlos jedoch ging die Verbindung zur Datenbank verloren, wenn das WMS längere Zeit (mehrere Stunden oder Tage) nicht verwendet wurde. Dies konnte durch den Einsatz einer XA-Datenquelle gelöst werden.

Wie in Kapitel 8.2.3.1.1 beschrieben, ermöglicht es jBPM in Prozesse beliebige Subprozesse einzubinden. Ist ein Prozess (A) gestartet und wird die Prozessdefinition des eingebundenen Subprozesses (B) verändert, so stellt sich die Frage, welche Version des Subprozesses ausgeführt wird. jBPM startet in diesem Fall die zur Startzeit des Prozesses A letzte Version von Prozess B. Dieses Vorgehen ist unter

---

<sup>36</sup><http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html>

<sup>37</sup>Mittels einer Datenquelle werden in Hibernate Datenbanken eingebunden.

Umständen nicht erwünscht. Wenn immer die aktuellste Version eines Subprozesses B gestartet werden soll, dann muss in der Prozessdefinition von A die Option `binding=late` angegeben werden. Dies hat aber zur Konsequenz, dass das Auslesen von Subprozessknoten vor dem Start des Subprozesses schwieriger wird, da in dem State-Knoten keine Verknüpfung zu einer Prozessdefinition vorhanden ist. Diese muss gesondert ausgelesen werden.

## Kapitel 9

# Ergebnisse der Evaluierung

Die Evaluierung dieser Arbeit erfolgte mittels Online-Fragebogen. Dieser wurde mit Hilfe der Software LimeSurvey<sup>38</sup> erstellt. LimeSurvey ist eine PHP und MySQL basierte Software, welche es ermöglicht, beliebige Fragebögen mit insgesamt 20 Fragentypen zu definieren. Die Software besteht aus einem Front- und Backend. Im Frontend können in einem Online-Formular die Fragebögen ausgefüllt werden, im Backend erfolgt die Administration und Auswertung der Fragebögen. Die Durchführung der Befragung erfolgte anonym.

Der Fragebogen wurde von 83% (zehn von zwölf) der Mitarbeiter, welche mit dem WMS arbeiten, vollständig ausgefüllt. Nicht vollständig ausgefüllte Fragebögen wurden verworfen. Die Befragung wurde zwei Wochen nach Einführung des WMS durchgeführt.

### 9.1 Situation vor Einführung des WMS

Es sind 70% der Befragten der Meinung, dass die Prozesse der Organisation vor Einführung des WMS von den Mitarbeitern unterschiedlich durchgeführt wurden. Nur 10% meinen, dass die Prozesse zum größten Teil ident durchgeführt wurden. Gleichzeitig ergibt die Umfrage, dass alle Befragten die vorhandenen Prozesse für verbesserungswürdig halten (Frage 3). Die Befragten sind auch der Meinung, dass es sinnvoll bzw. sehr sinnvoll ist, einen groben Ablaufplan der Prozesse vorzulegen sowie Dokumentation direkt bei den Arbeitsaufgaben zu hinterlegen.

---

<sup>38</sup><http://www.limesurvey.org/>

## **9.2 Situation nach Einführung des WMS**

Betreffend die Analyse der Prozesse (Frage 7) ergibt die Umfrage, dass 50% der Befragten die Prozessteam Meetings für produktiv halten, 30% wollten dazu keine Aussage machen oder waren an diesen Besprechungen nicht beteiligt. Weiters meinen 80%, dass sich der Ablauf der Prozesse nach der Einführung des WMS verbessert hat (20% sind unentschlossen). Auch die Aufgabendetails finden bei den Anwendern große Zustimmung. 90% sind der Ansicht, dass diese Funktionalität ihre Arbeit erleichtert, der zusätzliche Arbeitsaufwand, welcher durch die Verwendung der WMS Funktionalitäten entsteht, aber eher gering ist. Die Umsetzung der im ZDS angezeigten (Prozess-) Aufgabenlisten bewerten alle Anwender als gut bis sehr gut. Um den Status paralleler Aktivitäten zu erfragen, werden Email, Telefon, parallele Aufgaben in den Aufgabendetails sowie persönlicher Kontakt fast gleich häufig wie zuvor eingesetzt (siehe Fragen 1 und 14 ). Das WMS nimmt hier also eine unterstützende Rolle ein.

Die Einführung des WMS stellt laut Umfrage keine Einschränkung in der täglichen Arbeit der Mitarbeiter dar. 90% der Befragten geben an, dass sie das WMS nicht (70%) bzw. gar nicht (20%) bei ihrer Arbeit einschränkt. 10% sind unentschlossen. Weiters vertreten 90% die Meinung, dass die Einführung des WMS ihre tägliche Arbeit vereinfacht hat. Nur 10% meinen, dass die Einführung des WMS keinen Einfluss auf ihre Arbeit hat.

## **9.3 Evaluierung der Benutzerschnittstelle**

70% empfinden die Integration des WMS in das ZDS als optisch ansprechend, ebenso viele Personen meinen, die Prozessfunktionalitäten seien intuitiv bedienbar. Die Antworten zu den Fragen 17 - 25 fielen durchwegs sehr positiv aus, lediglich bei Frage 23 („Die Anordnung von Informationen zeigt die relative Wichtigkeit und Reihenfolge an“) scheint noch Optimierungspotenzial zu bestehen.

## **Kapitel 10**

# **Zusammenfassung und Ausblick**

Dieses Kapitel fasst einerseits die Ergebnisse der Arbeit zusammen, andererseits wird ein Ausblick auf mögliche darauf aufbauende Arbeiten gegeben.

Die Ergebnisse dieser Arbeit gliedern sich in zwei Bereiche: Einerseits Ziele, welche durch Prozessmanagement, andererseits Ziele, welche durch die Implementierung eines Workflow-Management-Systems erreicht wurden.

### **10.1 Umsetzung des Prozessmanagements**

Die Ergebnisse der Umsetzung des Prozessmanagements in der betrachteten Organisation stellen hauptsächlich die erhobenen Workflows aus Kapitel 6 dar. Diese wurden zuerst grob vom Management definiert und anschließend in einem iterativen Prozess vom Prozessteam genauer festgelegt (siehe Kapitel 5). Es war dabei wichtig, einen guten Mittelweg zwischen der Detailgetreue der Prozesse und dem Erhalt eines Handlungsspielraumes der Mitarbeiter zu finden. Zusätzlich zur Definition der Prozesse wurden auf diesem Weg Optimierungspotenziale in den aktuellen Prozessen erkannt und umgesetzt. Diese beinhalten Veränderungen des bestehenden Prozessablaufs sowie die Ergänzung des Prozessablaufes um neue Aufgaben. Für die Prozessoptimierung wurde die Vier Schritte Methode (Kapitel 2.1.8) verwendet. Mithilfe dieser war es auf einfache Art und Weise möglich, Probleme im Prozessablauf zu identifizieren und daraus Optimierungspotenziale abzuleiten. Das Erfassen der Prozesse fand direkt in dem vom jBPM zur Verfügung gestellten Editor (8.2.3.1.1) statt. Dies bot den Vorteil, dass die Prozesse nicht nochmals in eine vom WMS ausführbare Sprache übersetzt werden mussten. Für die Visualisierung der Prozesse (zum Zweck der Präsentation und Einschulung neuer Mitarbeiter) wurde eine andere Art der Darstellung gewählt: Die Prozessablaufdarstellung. Siehe Abbildungen in Kapitel 6.

Im Zuge des Erfassens der Workflows konnten Zuständigkeiten klar definiert, Grob Abläufe für Prozesse vorgegeben und die Erstellung einer ausführlichen Dokumentation zu den Prozessen veranlasst werden.

## **10.2 Umsetzung WMS**

Im Kapitel 8 wurde die Implementierung des Workflow-Management-Systems beschrieben. Das WMS hat zum Ziel, einen groben Ablauf der Prozesse vorzugeben (den Anwendern dennoch Freiheiten bei der Ausführung zuzugestehen) und die Anwender bei ihrer Arbeit bestmöglich zu unterstützen, ohne sie zu behindern. Anhand eines Fallbeispiels wurde das WMS in ein bestehendes Zeitdokumentationssystem integriert. Um die Akzeptanz der Benutzer möglichst hoch zu halten, wurde das Design des WMS an das ZDS angepasst. Dies stellte eine wichtige Anforderung dar, welche auch laut Kapitel 9 erfüllt wurde. Die Einschulung neuer Mitarbeiter wurde erleichtert, da dafür einerseits ein Prozess definiert wurde und andererseits durch die Prozesse ein Grob Ablauf samt Dokumentation vorgegeben wird. Dies erhöht die Qualität der von den Angestellten produzierten Leistungen. Mit Hilfe der „Liste paralleler Aufgaben“ in den Aufgabedetails (technische Details: 8.2.2.3.3, Entwurf: 7.3) wird den Anwendern das mühsame Abfragen betreffend den Status von Aufgaben, welche von Kollegen bearbeitet werden, abgenommen. Der Kommunikationsaufwand zwischen den Angestellten wird somit verringert, wodurch diese nun mehr Zeit für ihre eigentlichen Aufgaben haben. Die Aufgabedetails beinhalten weiters die Möglichkeit, beliebige Dokumente (Richtlinien, Checklisten, ...) zu Aufgaben zu hinterlegen. Dadurch entfällt für den Anwender die Zeit, die er andernfalls für das Suchen der Dokumente benötigt hätte.

Dem Management bietet das WMS viele Vorteile. Es kann auf einen Blick der Status vieler Projekte eingesehen werden. Es lassen sich Listen mit den Aufgaben der Mitarbeiter generieren, wodurch die Entscheidung, wem eine neue Aufgabe zugeteilt werden kann, erleichtert wird. Das Management kann sich weiters sicher sein, dass rechtlich verbindliche Tätigkeiten verlässlich durchgeführt (oder zumindest vom Anwender ganz bewusst übersprungen) werden, sobald sie in der Prozessdefinition vorhanden sind.

## **10.3 Ausblick**

Aufbauend auf die Implementierung des webbasierten Prozessmanagement-Systems können aus den davon generierten Daten (Zeitmarken für Prozesse und Aufgaben, Messgrößen, Dokumentation zu Aufgaben) Berichte erstellt werden, welche für die Planung von zukünftigen Prozessen Verwendung finden können. Diese Daten

ermöglichen auch die Verbesserung der Prozessdefinitionen. Um Kennzahlen für das Management schnell erfassbar zu machen, besteht die Möglichkeit, auf einer eigenen Intranetseite die Performance der Prozesse grafisch und numerisch über die Zeit darzustellen. Dadurch ließen sich frühzeitig Probleme erkennen und entsprechende Gegenmaßnahmen einleiten. Durch das Vorhandensein von Zuständigkeiten für Aufgaben können zukünftige freie/belegte Personalressourcen einfach ausgelesen werden. Dies vereinfacht die Planung von zukünftigen Projekten.

# Kapitel 11

## Begriffsdefinitionen

**AJAX** Asynchronous JavaScript and XML

**Aktion** Eine im ZDS eingegebene Beschreibung zu einer Arbeitsaufgabe

**AS** Application Server

**BPM** Business Process Management

**EPK** Ereignisgesteuerte Prozesskette

**GPD** Graphical Process Designer

**ITIL** IT Infrastructure Library

**ITT** IT-Technik

**jBPM** Java Business Process Management

**jPDL** Java Process Definition Language

**JSF** JavaServer Faces

**JSP** JavaServer Pages

**PM** Prozessmanagement

**PTM X** Prozessteam Meeting X (1-4).

**TaskNode** Ein Aufgabenblock des WMS, welcher mehrere Aufgaben beinhalten kann.

**Task** Eine Aufgabe in einem Aufgabenblock.

**TSD** Trainings-/Schulungs-/Demosystem

**UML** Unified Modeling Language

**WMS** Workflow-Management-System

**ZDS** Zeitdokumentationssystem

# Anhang A

## Quellcode

### A.1 Leeren der Datenbank

```
CREATE DEFINER='root'@'localhost' PROCEDURE 'emptyDB3'()  
2 BEGIN  
  set foreign_key_checks = 0;  
4 delete from activitylog;  
  delete from processinstancedescription;  
6 delete from nodeactormapping;  
  delete from customtask;  
8 delete from usergroup;  
  delete from usergroupmapping;  
10 delete from messung;  
  delete from JBPM_ACTION;  
12 delete from JBPM_BYTEARRAY;  
  delete from JBPM_BYTEBLOCK;  
14 delete from JBPM_COMMENT;  
  delete from JBPM_DECISIONCONDITIONS;  
16 delete from JBPM_DELEGATION;  
  delete from JBPM_EVENT;  
18 delete from JBPM_EXCEPTIONHANDLER;  
  delete from JBPM_JOB;  
20 delete from JBPM_LOG;  
  delete from JBPM_MODULEDEFINITION;  
22 delete from JBPM_MODULEINSTANCE;  
  delete from JBPM_NODE;  
24 delete from JBPM_POOLEDACTOR;  
  delete from JBPM_PROCESSDEFINITION;  
26 delete from JBPM_PROCESSINSTANCE;  
  delete from JBPM_RUNTIMEACTION;  
28 delete from JBPM_SWIMLANE;  
  delete from JBPM_SWIMLANEINSTANCE;  
30 delete from JBPM_TASK;  
  delete from JBPM_TASKACTORPOOL;  
32 delete from JBPM_TASKCONTROLLER;  
  delete from JBPM_TASKINSTANCE;  
34 delete from JBPM_TOKEN;  
  delete from JBPM_TOKENVARIABLEMAP;  
36 delete from JBPM_TRANSITION;
```

```

38 delete from JBPM.VARIABLEACCESS;
delete from JBPM.VARIABLEINSTANCE;
set foreign_key_checks = 1;
40 END

```

Listing A.1: Leeren der Datenbank

## A.2 SOLL-Zustand des Prozesses „Kundenwunsch“

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <process-definition xmlns="" name="Kundenwunsch.SOLL">
4
5
6   <start-state name="start-state1">
7     <transition to="Anforderung"></transition>
8   </start-state>
9
10
11   <task-node name="Aufwand">
12     <task name="Aufwandschaetzung"><description>
13       30-
14     </description>
15     <assignment class="at.org.session.UserAssignmentActor"/> </task>
16     <transition to="Angebot"></transition>
17   </task-node>
18
19   <task-node name="Dokumentation">
20     <task name="Dokumentation"><assignment class="at.org.session.
21       UserAssignmentActor"/> </task>
22     <task name="Field Validation"><assignment class="at.org.session.
23       UserAssignmentActor"/></task>
24     <transition to="Finale Tests"></transition>
25   </task-node>
26
27   <task-node name="Finale Tests">
28     <task name="Tests durch Projektleiter"><description>
29       50-
30     </description>
31     <assignment class="at.org.session.UserAssignmentActor"/> </task>
32     <transition to="Erweiterung OK?"></transition>
33   </task-node>
34
35   <task-node name="Anforderung">
36     <task name="Anforderung entgegennehmen und pruefen"><description>
37       10-
38     </description>
39     <assignment class="at.org.session.UserAssignmentActor"/> </task>
40     <task name="Anforderungsdokument erstellen">
41       <description>
42         20-
43       </description>
44       <assignment class="at.org.session.UserAssignmentActor"/>
45     </task>

```

```

44     <transition to="Aufwand"></transition>
    </task-node>
46
47 <task-node name="Implementierung">
48     <task name="Implementierung des Auftrags"><description>
49         40-
50     </description>
51     <assignment class="at.org.session.UserAssignmentActor"/> </task>
52 <task name="Tests"><assignment class="at.org.session.
53     UserAssignmentActor"/> </task>
54 <transition to="Dokumentation"></transition>
    </task-node>
56 <process-state name="Wartungsfenster">
57     <sub-process name="WartungKAV"></sub-process>
58     <transition to="Abnahme"></transition>
    </process-state>
60
61 <task-node name="Abnahme">
62     <task name="Abnahme des Features vom Kunden">
63         <description>
64             60-
65         </description>
66         <assignment class="at.org.session.UserAssignmentActor"/>
67     </task>
68     <transition to="Verechnung"></transition>
    </task-node>
70
71 <task-node name="Verechnung">
72     <task name="Veranlassung der Verrechnung">
73         <description>
74             70-
75         </description>
76         <assignment class="at.org.session.UserAssignmentActor"/>
77     </task>
78     <transition to="Messung"></transition>
    </task-node>
80
81 <task-node name="Angebot">
82     <task name="Angebot erstellen und an Kunden weiterleiten"><assignment
83         class="at.org.session.UserAssignmentActor"/></task>
84     <transition to="Auftrag erteilt?"></transition>
    </task-node>
86 <task-node name="Erweiterung OK?">
87     <transition to="Implementierung" name="Erweiterung muss ueberarbeitet
88         werden"></transition>
89     <transition to="Wartungsfenster" name="Erweiterung funktional und
90         bereit zum ausspielen"></transition>
    </task-node>
91 <state name="Messung">
92     <transition to="end-state1"></transition>
    </state>
94 <task-node name="Auftrag erteilt?">

```

```

    <task name="Auftrag erteilt?"><assignment class="at.org.session.
96     UserAssignmentActor"/></task>
    <transition to="Messung" name="Auftrag nicht erteilt"></transition>
    <transition to="Implementierung" name="Auftrag erteilt"></transition>
98 </task-node>
100 <end-state name="end-state1"></end-state>
</process-definition>

```

Listing A.2: Prozess „Kundenwunsch“

### A.3 SOLL-Zustand des Prozesses „Produktaktualisierung“

```

1 <?xml version="1.0" encoding="UTF-8"?>
3 <process-definition xmlns="" name="Produktaktualisierung">
5
7   <start-state name="start-state1">
9     <transition to="Interne Besprechung"></transition>
11   </start-state>
13   <task-node name="KickOffKunde">
15     <task name="Kickoff Meeting bei Kunde">
17       <description>
19         20-
21       </description>
23     <assignment class="at.org.session.UserAssignmentActor"/></task>
25     <transition to="KickoffIntern"></transition>
27   </task-node>
29   <process-state name="Wartungsfenster2">
31     <sub-process name="Wartung" binding="late"></sub-process>
33     <transition to="InstallationServer"></transition>
35   </process-state>
37   <task-node name="Interne Besprechung">
39     <task name="Interne Besprechung">
41       <description>
43         10-
45       </description>
47     <assignment class="at.org.session.UserAssignmentActor"/></task>
49     <transition to="KickOffKunde"></transition>
51   </task-node>
53   <task-node name="KickoffIntern">
55     <task name="Internes Kickoff">
57       <description>
59         30-
61       </description>
63     <assignment class="at.org.session.UserAssignmentActor"/></task>
65     <task name="Projektplan erstellen">
67       <description>
69         40-

```

```

43     </description>
44     <assignment class="at.org.session.UserAssignmentActor"/></task>
45     <task name="Aufgabenverteilung">
46         <description>
47             50-
48         </description>
49         <assignment class="at.org.session.UserAssignmentActor"/></task>
50         <transition to="Voraussetzungen"></transition>
51     </task-node>
52
53     <task-node name="PraelInstallation">
54         <task name="Install vorbereitung .net 3.5 SP1">
55             <description>
56                 80-
57             </description>
58             <assignment class="at.org.session.UserAssignmentActor"/></task>
59             <task name="SQL2005 + aktuellstes SP">
60                 <description>
61                     90-
62                 </description>
63                 <assignment class="at.org.session.UserAssignmentActor"/></task>
64                 <task name="Pruefen d. DB KOnsistenz">
65                     <description>
66                         100-
67                     </description>
68                     <assignment class="at.org.session.UserAssignmentActor"/></task>
69                     <task name="Windows Update durchf">
70                         <description>
71                             200-
72                         </description>
73                         <assignment class="at.org.session.UserAssignmentActor"/></task>
74                         <task name="Clientpaket bereit">
75                             <description>
76                                 110-
77                             </description>
78                             <assignment class="at.org.session.UserAssignmentActor"/></task>
79                             <task name="Lizenzfile anforderbar">
80                                 <description>
81                                     120-
82                                 </description>
83                                 <assignment class="at.org.session.UserAssignmentActor"/></task>
84                                 <transition to="Wartungsfenster2"></transition>
85                             </task-node>
86
87                         <task-node name="InstallationServer">
88                             <task name="Serverinstall">
89                                 <description>
90                                     210-
91                                 </description>
92                                 <assignment class="at.org.session.UserAssignmentActor"/></task>
93                                 <task name="Clientinstall am Server">
94                                     <description>
95                                         220-
96                                     </description>
97                                     <assignment class="at.org.session.UserAssignmentActor"/></task>
98                                     <task name="Lizenz anfordern">

```

```

99     <description>
100         130-
101     </description>
102 <assignment class="at.org.session.UserAssignmentActor"/></task>
103 <task name="Testen">
104     <description>
105         140-
106     </description>
107 <assignment class="at.org.session.UserAssignmentActor"/></task>
108 <task name="Erste Aufgabe: Dumps aller DBs">
109     <description>
110         230-
111     </description>
112 <assignment class="at.org.session.UserAssignmentActor"/></task>
113 <transition to="InstallationClient"></transition>
114 </task-node>
115 <task-node name="PostInstallation">
116     <task name="Bereinigen der Server">
117         <description>
118             180-
119         </description>
120     <assignment class="at.org.session.UserAssignmentActor"/></task>
121 <task name="Deinstallation SQL 2000"><assignment class="at.org.session
122     .UserAssignmentActor"/></task>
123 <task name="Sicherungskonzept anpassen und aktivieren">
124     <description>
125         190-
126     </description>
127 <assignment class="at.org.session.UserAssignmentActor"/></task>
128 <transition to="join1"></transition>
129 </task-node>
130 <task-node name="Fehlerfall">
131     <task name="Snapshot 3.9 wiederherstellen">
132         <description>
133             260-
134         </description>
135     <assignment class="at.org.session.UserAssignmentActor"/></task>
136 <task name="Letzte Dumps einspielen">
137     <description>
138         270-
139     </description>
140 <assignment class="at.org.session.UserAssignmentActor"/></task>
141 <task name="Kunde ueber Status informieren">
142     <description>
143         150-
144     </description>
145 <assignment class="at.org.session.UserAssignmentActor"/></task>
146 <transition to="Fehleranalyse"></transition>
147 </task-node>
148 <task-node name="InstallationClient">
149     <task name="1 Testclient von Kunde installieren lassen"><assignment
150         class="at.org.session.UserAssignmentActor"/></task>
151 <task name="Testen des Clients">

```

```

153     <description>
        250-
    </description>
155 <assignment class="at.org.session.UserAssignmentActor"/></task>
    <transition to="Client erfolgreich installiert und betriebsfaehig"></
        transition>
157 </task-node>

159 <task-node name="Voraussetzungen">
    <task name="Voraussetzungen f. Installation erfuellt">
161     <description>
        60-
163     </description>
    <assignment class="at.org.session.UserAssignmentActor"/></task>
165 <task name="Snapshot erstellen lassen">
    <description>
167     70-
    </description>
169 <assignment class="at.org.session.UserAssignmentActor"/></task>
    <transition to="Wartungsfenster1"></transition>
171 </task-node>

173 <process-state name="Wartungsfenster1">
    <sub-process name="Wartung" binding="late"></sub-process>
175 <transition to="PaeInstallation"></transition>
</process-state>
177

<task-node name="Rollout">
179 <task name="Clientrollout anordnen">
    <description>
181     160-
    </description>
183 <assignment class="at.org.session.UserAssignmentActor"/></task>
    <task name="Konfiguration anpassen">
185 <description>
        170-
187 </description>
    <assignment class="at.org.session.UserAssignmentActor"/></task>
189 <transition to="PostInstallation"></transition>
</task-node>
191

<task-node name="Fehleranalyse">
193 <task name="Fehler analysieren"><assignment class="at.org.session.
    UserAssignmentActor"/></task>
    <transition to="join1"></transition>
195 </task-node>

197 <state name="M: Update erfolgreich?">
    <transition to="end-statel"></transition>
199 </state>

201 <join name="join1">
    <transition to="M: Update erfolgreich?"></transition>
203 </join>

205 <task-node name="Client erfolgreich installiert und betriebsfaehig">

```

```

207 <task name="Rollout oder Fehler?"><assignment class="at.org.session.
    UserAssignmentActor"/></task>
    <transition to="Rollout" name="Rollout"></transition>
209 <transition to="Fehlerfall" name="Fehlerfall"></transition>
    </task-node>
211 <end-state name="end-state1"></end-state>
    </process-definition>

```

Listing A.3: Prozess „Produktaktualisierung“

## A.4 SOLL-Zustand des Prozesses „Neuer Mitarbeiter“

```

<?xml version="1.0" encoding="UTF-8"?>
2
<process-definition xmlns="" name="NeuerMitarbeiter">
4
    <start-state name="start-state1">
6        <transition to="Bewerbungsgespraech"></transition>
    </start-state>
8
    <task-node name="Arbeitsvertrag">
10        <task name="Arbeitsvertrag unterschrieben"><description>
12            10-
        </description>
        <assignment class="at.org.session.UserAssignmentActor"/> </task>
14        <transition to="join3"></transition>
    </task-node>
16
    <task-node name="Arbeitsplatz">
18        <task name="Arbeitsplatz vorhanden"><description>
20            20-
        </description>
        <assignment class="at.org.session.UserAssignmentActor"/> </task>
22        <transition to="join2"></transition>
    </task-node>
24
    <task-node name="Hardware">
26        <task name="Hardware vorhanden und einsatzbereit"><description>
28            30-
        </description>
        <assignment class="at.org.session.UserAssignmentActor"/> </task>
30        <transition to="TSD"></transition>
    </task-node>
32
    <task-node name="Arbeitsfaehigkeit herstellen">
34        <task name="Systemadmin CCheckliste erledigt"><description>
36            70-
        </description>
        <assignment class="at.org.session.UserAssignmentActor"/> </task>
38        <task name="BL Checkliste erledigt">
        <description>
40            60-
        </description>

```

```

42     <assignment class="at.org.session.UserAssignmentActor"/>
43   </task>
44   <transition to="Erster Arbeitstag"></transition>
45 </task-node>
46
47 <task-node name="Erster Arbeitstag">
48   <task name="Firmenfuehrung durchfuehren"><description>
49     90-
50   </description>
51   <assignment class="at.org.session.UserAssignmentActor"/> </task>
52   <task name="Schulung Zutrittskontrolle"><description>
53     100-
54   </description>
55   <assignment class="at.org.session.UserAssignmentActor"/> </task>
56   <transition to="Unterweisung Interne Ablaeufe"></transition>
57 </task-node>
58
59 <task-node name="Unterweisung Interne Ablaeufe">
60   <task name="ISIS"><description>
61     110-
62   </description>
63   <assignment class="at.org.session.UserAssignmentActor"/> </task>
64   <task name="Interne Richtlinien"><description>
65     120-
66   </description>
67   <assignment class="at.org.session.UserAssignmentActor"/> </task>
68   <transition to="fork1"></transition>
69 </task-node>
70
71 <task-node name="Produktausbildung">
72   <task name="Produkt auf VM installieren und damit "spielen"><
73     assignment class="at.org.session.UserAssignmentActor"/> </task>
74   <task name="Anwenderschulung"><assignment class="at.org.session.
75     UserAssignmentActor"/></task>
76   <task name="Technische Schulung"><assignment class="at.org.session.
77     UserAssignmentActor"/></task>
78   <task name="Assistenz zu SPOC"><assignment class="at.org.session.
79     UserAssignmentActor"/></task>
80   <task name="EVA"><assignment class="at.org.session.UserAssignmentActor
81     "/></task>
82   <transition to="join1"></transition>
83 </task-node>
84
85 <task-node name="Kundenberechtigungen">
86   <task name="RSA Key + Berechtigungen fuer Kundenserver bestellen"><
87     assignment class="at.org.session.UserAssignmentActor"/> </task>
88   <transition to="Evaluierung2"></transition>
89 </task-node>
90
91 <task-node name="Evaluierung1">
92   <task name="Zwischenevaluierung"><assignment class="at.org.session.
93     UserAssignmentActor"/></task>
94   <transition to="join1"></transition>
95 </task-node>
96
97 <task-node name="Evaluierung2">

```

```

92     <task name="Finale Evaluierung"><assignment class="at.org.session.
        UserAssignmentActor"/></task>
94     <transition to="end-state1"></transition>
96     </task-node>
98     <fork name="fork1">
100     <transition to="Evaluierung1"></transition>
102     <transition to="Produktausbildung"></transition>
104     </fork>
106     <join name="join1">
108     <transition to="Kundenberechtigungen"></transition>
110     </join>
112     <join name="join2">
114     <transition to="Arbeitsfaehigkeit herstellen"></transition>
116     </join>
118     <fork name="fork2">
120     <transition to="Arbeitsplatz"></transition>
122     <transition to="Hardware"></transition>
124     </fork>
126     <task-node name="Ansuchen Benutzer">
128     <task name="Formular Ansuchen fuer einen neuen EDV Benutzer ausfuellen
        ">
130     <description>
132     50-
134     </description>
136     <assignment class="at.org.session.UserAssignmentActor"/>
138     </task>
140     <transition to="join3"></transition>
142     </task-node>
144     <task-node name="TSD">
146     <task name="TSD einrichten">
148     <description>
150     80-
152     </description>
154     <assignment class="at.org.session.UserAssignmentActor"/>
156     </task>
158     <transition to="join2"></transition>
160     </task-node>
162     <fork name="fork3">
164     <transition to="Ansuchen Benutzer"></transition>
166     <transition to="Arbeitsvertrag"></transition>
168     </fork>
170     <join name="join3">
172     <transition to="fork2"></transition>
174     </join>
176     <task-node name="Bewerbungsgespraech">
178     <task name="Ansuchen um Arbeitsvertrag an DLU">
180     <description>

```

```

146     Typ des Arbeitsvertrags + zusaetzlich ausgemachte Dinge an DLU
    </description>
    <assignment class="at.org.session.UserAssignmentActor"/>
148 </task>
    <transition to="fork3"></transition>
150 </task-node>
152 <end-state name="end-state1"></end-state>
154 </process-definition>

```

Listing A.4: Prozess „Neuer Mitarbeiter“

## A.5 SOLL-Zustand des Prozesses „Anwendersupport“

```

<?xml version="1.0" encoding="UTF-8"?>
2
<process-definition xmlns="" name="CO.Anwendersupport">
4
    <start-state name="start-state1">
6        <transition to="Anfrage"></transition>
    </start-state>
8
    <task-node name="Anfrage">
10        <task name="Problemanalyse"><assignment class="at.org.session.
            UserAssignmentActor"/></task>
        <transition to="Loesung durch ITT oder Proj"></transition>
12    </task-node>
14
    <task-node name="Loesung">
        <task name="Problembearbeitung und Loesung"><assignment class="at.org.
            session.UserAssignmentActor"/></task>
16        <transition to="Kunde informieren"></transition>
    </task-node>
18
    <task-node name="Projektansprechpartner">
20        <task name="Weiterleitung an Projektansprechpartner"><assignment class
            ="at.org.session.UserAssignmentActor"/></task>
        <transition to="Loesung"></transition>
22    </task-node>
24
    <task-node name="Kunde informieren">
        <task name="Kunde informieren"><assignment class="at.org.session.
            UserAssignmentActor"/></task>
26        <transition to="Dokumentation"></transition>
    </task-node>
28
    <task-node name="Dokumentation">
30        <task name="Anpassung Regelwerk / Projektdoku"><assignment class="at.
            org.session.UserAssignmentActor"/></task>
        <transition to="end-state1"></transition>
32    </task-node>
34
    <task-node name="Loesung durch ITT oder Proj">

```

```
36 <task name="Loesung durch ITT oder Projektteam"><assignment class="at.  
org.session.UserAssignmentActor"/></task>  
38 <transition to="Projektansprechpartner" name="Projektteam"></  
transition>  
<transition to="Loesung" name="ITT"></transition>  
40 </task-node>  
42 <end-state name="end-state1"></end-state>  
</process-definition>
```

Listing A.5: Prozess „Anwendersupport“

## Anhang B

# Fragebögen

### B.1 Ergebnisse des Fragebogens „Allgemeine Daten“

<b>Frage 26:</b> Bitte geben Sie Ihr Alter in Jahren an.	
16-20	0 %
21-25	20 %
26-30	40 %
31-35	20 %
36-40	20 %
41-45	0 %
<b>Frage 27:</b> Bitte geben Sie Ihr Geschlecht an.	
Weiblich	20 %
Männlich	70 %
Keine Angabe	10 %
<b>Frage 28:</b> Bitte geben Sie Ihre Berufserfahrung in Jahren an.	
0-5	40 %
6-10	20 %
11-15	30 %
16-20	10 %
21-25	0 %
mehr	0 %

Tabelle B.1: Ergebnisse des Fragebogens „Allgemeine Daten“

## **B.2 Ergebnisse des Fragebogens „Situation vor Einführung des WMS“**

<b>Frage 1:</b> Wie haben Sie den Status von parallelen Aktivitäten, welche von Kollegen durchgeführt wurden, erfragt (Mehrfachantworten möglich)?	
Email	9
Telefon	7
Persönlich	7
Anders	1
<b>Frage 2:</b> Die Prozesse des Unternehmens wurden früher (vor Einführung des WMS) von jedem Mitarbeiter ident (Ablauf der Arbeitsschritte, zeitliche Abfolge der Arbeitsschritte) durchgeführt.	
Trifft sicher zu	0 %
Trifft zu	10 %
Unentschlossen	20 %
Trifft nicht zu	60 %
Trifft sicher nicht zu	10 %
<b>Frage 3:</b> Der Ablauf der Prozesse vor Einführung des WMS war verbesserungswürdig.	
Trifft sicher zu	80 %
Trifft zu	20 %
Unentschlossen	0 %
Trifft nicht zu	0 %
Trifft sicher nicht zu	0 %
<b>Frage 4:</b> Trifft es zu, dass die Anzeige des Status von parallelen Aufgaben hilfreich ist?	
Trifft sicher zu	0 %
Trifft zu	10 %
Unentschlossen	10 %
Trifft nicht zu	60 %
Trifft sicher nicht zu	0 %
<b>Frage 5:</b> Wie bewerten Sie die Idee, einen groben Ablaufplan für Prozesse vorzugeben?	
Sehr sinnvoll	30 %
Sinnvoll	70 %
Unentschlossen	0 %
Nicht sinnvoll	0 %

Sinnlos	0 %
<b>Frage 6:</b> Sind Sie der Meinung, dass es sinnvoll ist, für einzelne Arbeitsaufgaben eine Dokumentation bereitzustellen, welche etwaig auftretende Fragen beantwortet?	
Sehr sinnvoll	70 %
Sinnvoll	30 %
Unentschlossen	0 %
Nicht sinnvoll	0 %
Sinnlos	0 %

Tabelle B.2: Ergebnisse des Fragebogens „Situation vor Einführung des WMS“

### B.3 Ergebnisse des Fragebogens „Situation nach Einführung des WMS“

<b>Frage 7:</b> Sofern Sie an der Analyse eines oder mehrerer Prozesse beteiligt waren: Trifft es zu, dass die Besprechungen für die Prozessanalyse produktiv waren?	
Trifft sicher zu	40 %
Trifft zu	10 %
Unentschlossen	20 %
Trifft nicht zu	0 %
Trifft sicher nicht zu	0 %
Keine Antwort	30%
<b>Frage 8:</b> Sind Sie der Meinung, dass zu Arbeitsaufgaben alle benötigten Informationen hinterlegt werden können?	
Trifft sicher zu	0 %
Trifft zu	40 %
Unentschlossen	30 %
Trifft nicht zu	10 %
Trifft sicher nicht zu	0 %
Keine Antwort	20%
<b>Frage 9:</b> Trifft es zu, dass sich der Ablauf der Prozesse nach Einführung des WMS verbessert hat?	

Trifft sicher zu	10 %
Trifft zu	70 %
Unentschlossen	20 %
Trifft nicht zu	0 %
Trifft sicher nicht zu	0 %
<b>Frage 10:</b>	
Trifft es zu, dass die Aufgabendetails Ihre tägliche Arbeit erleichtern?	
Trifft sicher zu	10 %
Trifft zu	80 %
Unentschlossen	10 %
Trifft nicht zu	0 %
Trifft sicher nicht zu	0 %
<b>Frage 11:</b>	
Trifft es zu, dass die Anzeige des Status von parallelen Aufgaben hilfreich ist?	
Trifft sicher zu	30 %
Trifft zu	50 %
Unentschlossen	20 %
Trifft nicht zu	0 %
Trifft sicher nicht zu	0 %
<b>Frage 12:</b>	
Trifft es zu, dass der Zeitaufwand, der durch die Verwendung des WMS entsteht, gering ist?	
Trifft sicher zu	40 %
Trifft zu	20 %
Unentschlossen	40 %
Trifft nicht zu	0 %
Trifft sicher nicht zu	0 %
<b>Frage 13:</b>	
Wie bewerten Sie die Umsetzung der Vorgabe eines groben Ablaufplans durch die Aufgabenlisten?	
Sehr gut	40 %
Gut	60 %
Mittelmäßig	0 %
Schlecht	0 %
Sehr schlecht	0 %
<b>Frage 14:</b>	

Wie erfragen Sie den Status von parallelen Aktivitäten, welche von Kollegen durchgeführt wurden, jetzt (Mehrfachantworten möglich)?	
Email	5
Telefon	5
Persönlich	6
Parallele Aufgaben in den Aufgabendetails	5
Anders	1
<b>Frage 15:</b> Fühlen Sie sich durch die Einführung des WMS bei Ihrer Arbeit eingeschränkt?	
Ja sehr	0 %
Ja	0 %
Unentschlossen	10 %
Nein	70 %
Nein garnicht	20 %
<b>Frage 16:</b> Die Einführung des WMS hat meine Arbeit...	
Vereinfacht	90 %
Nicht verändert	10 %
Erschwert	0 %

Tabelle B.3: Ergebnisse des Fragebogens „Situation nach Einführung des WMS“

## B.4 Ergebnisse des Fragebogens „Evaluierung der Benutzerschnittstelle“

<b>Frage 17:</b> Die Umsetzung des WMS ist optisch ansprechend	
Trifft sicher zu	20 %
Trifft zu	50 %
Unentschlossen	30 %
Trifft nicht zu	0 %
Trifft sicher nicht zu	0 %
<b>Frage 18:</b> Das WMS ist intuitiv zu bedienen	
Trifft sicher zu	20 %
Trifft zu	50 %
Unentschlossen	30 %

Trifft nicht zu	0 %
Trifft sicher nicht zu	0 %
<b>Frage 19:</b> Links sind immer als solche zu erkennen	
Trifft sicher zu	30 %
Trifft zu	60 %
Unentschlossen	10 %
Trifft nicht zu	0 %
Trifft sicher nicht zu	0 %
<b>Frage 20:</b> Die Schrift ist groß genug, um erkennbar und lesbar zu sein	
Trifft sicher zu	80 %
Trifft zu	20 %
Unentschlossen	0 %
Trifft nicht zu	0 %
Trifft sicher nicht zu	0 %
<b>Frage 21:</b> Die Schrift kontrastiert ausreichend mit dem Hintergrund. Sie ist gut erkennbar und lesbar	
Trifft sicher zu	60 %
Trifft zu	40 %
Unentschlossen	0 %
Trifft nicht zu	0 %
Trifft sicher nicht zu	0 %
<b>Frage 22:</b> Die Grafiken sind groß genug	
Trifft sicher zu	50 %
Trifft zu	50 %
Unentschlossen	0 %
Trifft nicht zu	0 %
Trifft sicher nicht zu	0 %
<b>Frage 23:</b> Die Anordnung von Informationen zeigt die relative Wichtigkeit und Reihenfolge an	
Trifft sicher zu	40 %
Trifft zu	30 %

Unentschlossen	30 %
Trifft nicht zu	0 %
Trifft sicher nicht zu	0 %
<b>Frage 24:</b>	
Die Texte sind verständlich (z.B. Wortwahl, Satzbau, Fachbegriffe)	
Trifft sicher zu	30 %
Trifft zu	60 %
Unentschlossen	10 %
Trifft nicht zu	0 %
Trifft sicher nicht zu	0 %
<b>Frage 25:</b>	
Abstürze oder Fehlermeldungen sind selten aufgetreten	
Trifft sicher zu	20 %
Trifft zu	70 %
Unentschlossen	10 %
Trifft nicht zu	0 %
Trifft sicher nicht zu	0 %

Tabelle B.4: Ergebnisse des Fragebogens „Evaluierung der Benutzerschnittstelle“

# Literaturverzeichnis

- [All05] ALLWEYER, T.: *Geschäftsprozessmanagement: Strategie, Entwurf, Implementierung, Controlling*. W3L-Verl, 2005
- [Ang05] ANGERMEIER, G.: *Projektmanagement-Lexikon*. Projekt Magazin, 2005
- [Bal08] BALZERT, H.: *UML 2 in 5 Tagen: Der schnelle Einstieg in die Objektorientierung*. W3l GmbH, 2008
- [BB04] BRAND, K. ; BOONEN, H.: *IT Governance: A Pocket Guide Based on COBIT*. Van Haren Publishing, 2004
- [Bei08] BEIMS, M.: *IT Service Management in der Praxis mit ITIL 3: Zielfindung, Methoden, Realisierung*. Hanser Fachbuchverlag, 2008
- [Blo03] BLONSKI, H.: *Prozessmanagement in Pflegeorganisationen: Grundlagen-Erfahrungen-Perspektiven*. Schlütersche, 2003
- [BM05] BECKER, Jörg ; MEISE, Volker: Strategie und Ordnungsrahmen. In: BECKER, Jörg (Hrsg.) ; KUGELER, Martin (Hrsg.) ; ROSEMAN, Michael (Hrsg.): *Prozessmanagement. Ein Leitfaden zur prozessorientierten Organisationsgestaltung* Bd. Fünfte, überarbeitete und erweiterte Auflage. Springer Berlin Heidelberg, 2005, S. 105 – 154
- [Bor02] BORN, U.: Innovatives und flexibles Prozessmanagement mit dem Phios Process Repository. In: *Retail Banking im Informationszeitalter: Integrierte Gestaltung Der Geschäfts-, Prozess-Und Applikations-ebene* (2002), S. 149
- [Bre07] BRENNER, M.: *Werkzeugunterstützung für ITIL-orientiertes Dienstmanagement: Ein modellbasierter Ansatz*. BoD–Books on Demand, 2007
- [Büh04] BÜHNER, R.: *Betriebswirtschaftliche Organisationslehre*. Oldenbourg Wissenschaftsverlag, 2004

- [BWW03] BÖNING, D.W.I.B. ; WILHELMSHAVEN, S. ; WIRTSCHAFTSINGENIEURWESEN, F.: Prozessorientierte Unternehmensführung. In: *FH Oldenburg/Ostfriesland/Wilhelmshaven* (2003)
- [Ebe08] EBEL, N.: *ITIL V3 Basiszertifizierung: Grundlagenwissen und Zertifizierungsvorbereitung für die ITIL Foundation-prüfung*. Pearson Education Deutschland, 2008
- [Erd00] ERDMANN, J.: *Integriertes Prozessmanagement.: Ein multidimensionaler Ansatz für das Management von Prozessen in Unternehmen*. BoD-Books on Demand, 2000
- [Fin03] FINK, C.A.: *Prozessorientierte Unternehmensplanung: Analyse, Konzeption und Praxisbeispiele*. Deutscher Universitätsverlag, 2003
- [Fro09] FROST, Greg: *The S.M.A.R.T. Way to Set Goals*. Webseite, 2009. – [http://www.chargedaudio.com/resources/Smart\\_Way\\_Set\\_Goals.html](http://www.chargedaudio.com/resources/Smart_Way_Set_Goals.html); Stand 15.03.2009
- [GA04] GAITANIDES, M. ; ACKERMANN, I.: Die Geschäftsprozessperspektive als Schlüssel zu betriebswirtschaftlichem Denken und Handeln. In: *BWP@: Lernfelder gestalten-miteinander Lernen-Innovationen vernetzen*. 1 (2004), S. 4–28
- [Gad07] GADATSCH, A.: *Grundkurs Geschäftsprozess-management: Methoden und Werkzeuge für die IT-praxis: Eine Einführung für Studenten und Praktiker*. Vieweg Friedr.+ Sohn Ver, 2007
- [GKS08] *Kapitel Der operative IT-Strategie-Ansatz*. In: GRIMM, Robert ; KEUPER, Frank ; SCHOMANN, Marc: *Strategisches IT-Management*. Gabler, 2008, S. 115 – 141
- [Goe02] GOESMANN, T.: *Ein Ansatz zur Unterstützung wissensintensiver Prozesse durch Workflow-management-systeme*. Berlin, 2002
- [Gol06] GOLTSCHKE, W.: *COBIT kompakt und verständlich der Standard zur IT-Governance; so gewinnen Sie Kontrolle über Ihre IT; so steuern Sie Ihre IT und erreichen Ihr Ziele*. Vieweg, 2006
- [Hin08] HINKELMANN, Knut: Prozessidentifikation. In: *Vorlesungsfolien* (2008)
- [Hol95] HOLLINGSWORTH, David: The Workflow Reference Model / Workflow Management Coalition. 1995. – Forschungsbericht. – Document Number TC00-1003

- [JBo08a] JBoss, Team: *jBPM jPDL User Guide*. Webseite, 2008. – <http://docs.jboss.com/jbpm/v3.2/userguide/>; Stand 03.03.2009
- [JBo08b] JBoss, Team: Seam - Contextual Components. In: [http://docs.jboss.com/seam/2.1.1.GA/reference/en-US/html\\_single/](http://docs.jboss.com/seam/2.1.1.GA/reference/en-US/html_single/) (2008)
- [JKC04] JUNG, H.W. ; KIM, S.G. ; CHUNG, C.S.: Measuring software product quality: a survey of ISO/IEC 9126. In: *IEEE software* (2004), S. 88–92
- [Küh] KÜHLE, Markus: *Tutorial: Mit JBoss Seam und JEE5 unter Eclipse starten*. Webseite: <http://javathreads.de/2008/09/tutorial-mit-jboss-seam-und-jee5-unter-eclipse-starten/#FazitWieGehtEsWeiter>, . – Stand 27.05.2009
- [Köh06] KÖHLER, P.T.: *PRINCE 2: Das Projektmanagement-Framework*. Springer, 2006
- [KM06] KOSTKA, C. ; MÖNCH, A.: *Change Management: 7 Methoden für die Gestaltung von Veränderungsprozessen*. Hanser, 2006
- [Kra08] KRAMBERG, Volker: *Zielorientierte Geschäftsprozesse mit WS-BPEL*, Universität Stuttgart, Diplomarbeit, 2008
- [KV05] KUGELER, Martin ; VIETING, Michael: Gestaltung einer prozessorientiert(er)en Aufbauorganisation. In: BECKER, Jörg (Hrsg.) ; KUGELER, Martin (Hrsg.) ; ROSEMANN, Michael (Hrsg.): *Prozessmanagement. Ein Leitfaden zur prozessorientierten Organisationsgestaltung* Bd. Fünfte, überarbeitete und erweiterte Auflage. Springer Berlin Heidelberg, 2005, S. 221 – 267
- [Len05] LENZ, G.: *Integriertes Prozessmanagement: Erfolgreiche Einführung und Optimierung*. BoD–Books on Demand, 2005
- [Lim] LIMITED, Object R.: *JFreeChart Homepage*. Webseite. <http://www.jfree.org/jfreechart>. – Stand 22.05.2009
- [Mar08] MARTIN, C.: *Wandel auf der Ebene der Organisation: Darstellung und Bewertung der Ansätze von Lewin, Bullock& Batten, Kotter, Beckhard& Harris, Senge, Stacey& Shaw*. GRIN Verlag OHG, 2008
- [Mic] MICROSYSTEMS, Sun: *MySQL Homepage*. Webseite. <http://de.sun.com/products/software/mysql>. – Stand 25.05.2009

- [Mic06] MICROSYSTEMS, Sun: *Expression Language*. Webseite. <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/JSPIntro7.html>.  
Version: 2006. – Stand 30.03.2009
- [NN07] NUSAIRAT, J. ; NUSAIRAT, J.F.: *Beginning JBoss Seam: From Novice to Professional*. Apress, 2007
- [Oel99] OELSNITZ, D. von d.: *Marktorientierte Organisationsgestaltung*. Kohlhammer, 1999
- [Oes06] OESTEREICH, B.: *Analyse und Design mit UML 2.1*. Oldenbourg Wissenschaftsverlag, 2006
- [Olb08] OLBRICH, A.: *ITIL kompakt und verständlich: Effizientes IT Service Management-den Standard für IT-prozesse kennenlernen, verstehen und erfolgreich in der Praxis umsetzen*. Vieweg+ Teubner Verlag, 2008
- [Omo] OMONDO: *EclipseUML 2008 Plugin*. Webseite. <http://www.omondo.com>. – Stand 25.05.2009
- [Pap02] PAPASTAMATELOU, Julie: Kommunikation in Organisationen. In: *Hauptseminar: Kommunikation in Unternehmen* (2002)
- [Rec06] RECHENBERG, P.: *Informatik-Handbuch*. Hanser, 2006
- [SBH06] SCHALK, C. ; BURNS, E. ; HOLMES, J.: *JavaServer faces: the complete reference*. McGraw-Hill Osborne Media, 2006
- [Sch00] SCHLACHTER, Marion: Grundmodelle der Aufbauorganisation im Unternehmen-ein kritischer Vergleich. In: *Fachhochschule Frankfurt am Main, Fachbereich Wirtschaft, Hausarbeit* (2000)
- [Sch05] SCHWAB, Simon: Grundformen von Organisationsstrukturen: Darstellung, kritische Analyse und praktische Anwendung der funktionalen und divisionalen Organisationsstruktur sowie der Matrixorganisation. In: *Betriebswirtschaftliches Hauptseminar* (2005)
- [SS07] SCHMELZER, H.J. ; SESSELMANN, W.: *Geschäftsprozessmanagement in der Praxis: Kunden zufrieden stellen-Produktivität steigern-Wert erhöhen*. Hanser Fachbuchverlag, 2007
- [Öst] ÖSTERREICH, Staat: *Bundesgesetz betreffend Medizinprodukte (Medizinproduktegesetz - MPG)*. Webseite. <http://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10011003>. – Stand 01.05.2009

- [Sta06] STAUD, J.: *Geschäftsprozessanalyse: ereignisgesteuerte Prozessketten und objektorientierte Geschäftsprozessmodellierung für betriebswirtschaftliche Standardsoftware*. Springer, 2006
- [STK00] STEINLE, C. ; THIEM, H. ; KIRCHHOFF, F.: Entwicklungsstand eines prozessorientierten Controlling. In: *controller magazin* 24 (2000), Nr. 2, S. 122–127
- [Töp07] TÖPFER, A.: *Betriebswirtschaftslehre: Anwendungs-und prozessorientierte Grundlagen*. Springer, 2007
- [Tra09a] TRACKPLUS: *Track+ Projektmanagement-Software*. Webseite, 2009. – <http://www.trackplus.de>; Stand 30.03.2009
- [Tra09b] TRACKPLUS (Hrsg.): *Track+ Schnelleinstieg (deutsch)*. Trackplus, 2009. – Stand 30.03.2009
- [Uni08] UNIFUSSER, W.: *Prozess der ITIL Einführung am Beispiel eines IT-Dienstleisters*, Universität Wien, Diplomarbeit, 2008
- [VB08] VAN BON, J.: *Foundations of IT Service Management basierend auf ITIL V3*. Van Haren Publishing, 2008
- [VR05] VANDERFEESTEN, Irene ; REIJERS, Hajo A.: A Human-Oriented Tuning of Workflow Management Systems. In: *Business Process Management 3rd International Conference, BPM 2005*, Wil M.P. van der Aalst, Boualem Benatallah, Fabio Casati, Francisco Curbera, 2005, S. 80 – 95
- [Wag06] WAGNER, Karl W.: *PQM - Prozessorientiertes Qualitätsmanagement*. Hanser, 3. Auflage, 2006
- [Wal01] WALLMÜLLER, Ernest: *Software-Qualitätsmanagement in der Praxis*. Hanser Verlag, 2001
- [WP07] WAGNER, Karl W. ; PATZAK, Gerold: *Performance Excellence: der Praxisleitfaden zum effektiven Prozessmanagement*. Hanser, 2007
- [ZHB05] ZARNEKOW, R. ; HOCHSTEIN, A. ; BRENNER, W.: *Serviceorientiertes IT-Management: ITIL-best-practices und-fallstudien*. Springer, 2005