

Freie wissenschaftliche Arbeit zur Erlangung des akademischen Grades  
Master of Science in Wirtschaftsinformatik

# **Einsatzmöglichkeiten des SAP NetWeaver CE 7.3 für modellgetriebene Entwicklungen an- hand eines konkreten Szenarios bei der Berli- ner Stadtreinigung**

## **Masterthesis**

im Fachbereich Wirtschaftswissenschaften II  
im Studiengang Wirtschaftsinformatik  
der Hochschule für Technik und Wirtschaft Berlin

vorgelegt von: Tobias Lucke  
Brünhildestraße 1a  
14542 Werder (Havel)  
Matrikel-Nr: S0531240

Erstbetreuer: Prof. Dr. Jörg Courant  
Zweitbetreuer: Prof. Dr. Andreas Böhne

Abgabetermin: 03.09.2012

## Abstract

*„Einsatzmöglichkeiten des SAP NetWeaver CE 7.3 für modellgetriebene Entwicklungen anhand eines konkreten Szenarios bei der Berliner Stadtreinigung“*

von Tobias Lucke

Die Softwareindustrie befindet sich in einem stetigen Wandel um komplexere Anwendungen in kürzerer Zeit in einer hohen Qualität zu entwickeln. In den Evolutionsstufen hin zur objektorientierten Entwicklung wurde stetig der Abstraktionsgrad der Programmiersprachen erhöht, so dass sich Anwendungsentwickler vermehrt mit der Programmierung von Business-Logik befassen können. Mit der modellgetriebenen Softwareentwicklung steht nun die nächste Generation der Programmierverfahren bereit.

Die SAP AG hat mit dem NetWeaver CE einen Applikationsserver mit entsprechender Entwicklungsumgebung auf den Markt gebracht, welcher mehrere modellgetriebene Technologien mit sich bringt. Inwiefern diese sich gegen etablierte Standards und Best-Practices im Bereich von Geschäftsanwendungen bewähren können, ist Aufgabe dieser Masterthesis. Hierfür werden das Composite Application Framework, das Business Process Management, der Visual Composer und Web Dynpro Java hinsichtlich ihres Funktionsumfangs, der Produktivität und den Einsatzmöglichkeiten innerhalb einer Composite Application beleuchtet.

Um sich nicht nur auf Untersuchungen im Labor zu beschränken, wird ein Szenario bei der Berliner Stadtreinigung mit den zuvor evaluierten Technologien erfolgreich realisiert. Währenddessen erfolgt eine Analyse wie die modellgetriebene Softwareentwicklung ihren hohen Anspruch an Dokumentation, Wartbarkeit, Entwicklungskosten und Qualität erfüllt. Weiterhin werden Veränderungen am Entwicklungsprozess beobachtet und ihre Bedeutung für zukünftige Entwicklungsarbeiten beschrieben.

## Abstract

*„Application possibilities of the SAP NetWeaver CE 7.3 for model-driven software development by reference to a specific scenario of the Berliner Stadtreinigung“*

by Tobias Lucke

The software industry is in a continuous change to develop more complex applications in less time and higher quality. The degree of abstraction of the programming language in the field of object-oriented development has been increased permanently, resulting in the application developer spending more time with the programming of business-logic. With the model-driven software development, the next generation of programming methods is now available.

The SAP AG, by developing the NetWeaver CE, released an application server with corresponding development infrastructure and several model-driven technologies. The assignment of this master thesis is to examine how they can prove themselves compared to established standards and best-practices in the domain of business applications. Therefore the composition application framework, the business process management, the visual composer and web dynpro java have been evaluated according to their range of functions, productivity and the application possibilities within composite applications.

In Order not to be confined to laboratory studies, a scenario of the Berliner Stadtreinigung has been successfully implemented with the previously evaluated technologies. Meanwhile an analysis takes place on how the model-driven software development fulfills its high demand for documentation, maintainability, development costs and quality. Moreover, changes to the development process are observed and their impact for future development work has been described in this master thesis.

## Inhaltsverzeichnis

Abstract .....	II
Abstract .....	III
Inhaltsverzeichnis .....	IV
Abbildungsverzeichnis .....	VI
Abkürzungsverzeichnis .....	VII
1 Einleitung .....	1
1.1 Motivation .....	1
1.2 Zielsetzung .....	2
1.3 Methodisches Vorgehen .....	3
2 Modellgetriebene Softwareentwicklung .....	5
2.1 Schwierigkeiten in der traditionellen Softwareentwicklung .....	5
2.2 Grundidee und Begriffsbestimmung .....	9
2.2.1 Ansatz .....	9
2.2.2 Modellierungssprache .....	10
2.2.3 Metamodell .....	12
2.3 Varianten der Modelltransformation .....	14
2.4 Auswirkungen auf identifizierte Schwächen .....	16
2.5 Beispiele für konkrete Ansätze der Umsetzung .....	18
2.5.1 Computer-aided Software Engineering (CASE) .....	18
2.5.2 Model Driven Architecture (MDA) .....	20
3 SAP NetWeaver 7.3 – Composition Environment .....	24
3.1 Grundlagen .....	24
3.1.1 Die Firma SAP .....	24
3.1.2 Serviceorientierung in der Anwendungsarchitektur .....	25
3.2 SAP NetWeaver Composition Environment .....	27
3.2.1 Einführung .....	27
3.2.2 Composite Applications .....	28
3.2.3 Komponenten des SAP NetWeaver CE .....	30
4 Werkzeuge und Technologien zur modellgetriebenen Entwicklung .....	34
4.1 NetWeaver Development Studio .....	34
4.2 Composite Application Framework .....	35
4.2.1 Einsatzgebiet und Konzept .....	35
4.2.2 Modellierungswerkzeug .....	37
4.2.3 Modelltransformation .....	40
4.2.4 Bewertung .....	41
4.3 Visual Composer .....	42
4.3.1 Einsatzgebiet und Konzept .....	42
4.3.2 Modellierungswerkzeug .....	44
4.3.3 Modelltransformation .....	47
4.3.4 Bewertung .....	48
4.4 Web Dynpro Java .....	49
4.4.1 Einsatzgebiet und Konzept .....	49

---

4.4.2 Modellierungswerkzeug.....	51
4.4.3 Modelltransformation.....	55
4.4.4 Bewertung.....	56
4.5 Business Process Management.....	57
4.5.1 Einsatzgebiet und Konzept.....	57
4.5.2 Modellierungswerkzeug.....	59
4.5.3 Modelltransformation.....	61
4.5.4 Bewertung.....	62
4.6 Zusammenfassung.....	63
5 Praktischer Einsatz bei der Berliner Stadtreinigung.....	65
5.1 Berliner Stadtreinigung.....	65
5.2 Betriebliches Eingliederungsmanagement.....	66
5.2.1 Bedeutung für das Unternehmen.....	66
5.2.2 Ist-Zustand.....	68
5.2.3 Soll-Zustand.....	69
5.3 Lösungsansatz.....	70
5.4 Realisierung.....	71
6 Auswirkungen auf den Entwicklungsprozess.....	76
6.1 Organisation der Entwicklungsarbeit.....	76
6.2 Qualität und Flexibilität.....	78
6.3 Kosten.....	81
7 Fazit.....	84
7.1 Zusammenfassung.....	84
7.2 Bewertung.....	86
7.3 Ausblick.....	88
Literaturverzeichnis.....	VIII
Internetverzeichnis.....	XI
Anhang.....	XIV
Anhang I: BPMN 2.0 Sprachelementabdeckung BPM.....	XIV
Anhang II: BPMN Prozess zum BEM.....	XV

---

## Abbildungsverzeichnis

Abbildung 2-1: Phasen des traditionellen Softwareentwicklungszyklus.....	7
Abbildung 2-2: Begriffsbeziehungen.....	10
Abbildung 2-3: UML Klassendiagramm für die Entität „Buch“ .....	13
Abbildung 2-4: Beschreibungshierarchie .....	13
Abbildung 2-5: Evolution der Softwareentwicklung.....	16
Abbildung 2-6: Ablauf der Abstraktionsebenen bei MDA.....	21
Abbildung 2-7: Metaebenen .....	22
Abbildung 3-1: Schichten einer Composite Application .....	29
Abbildung 3-2: SAP NetWeaver Composition Environment .....	31
Abbildung 4-1: Benutzeroberfläche des NetWeaver Developer Studios .....	35
Abbildung 4-2: Einordnung der Service Schicht in die Gesamtarchitektur .....	36
Abbildung 4-3: Modellierung eines Geschäftsobjekts im NWDS .....	38
Abbildung 4-4: Modelltransformation CAF.....	40
Abbildung 4-5: Eigenschaften von Web Dynpro und Visual Composer.....	43
Abbildung 4-6: Strategische Ausrichtung der Java-UI-Technologien .....	44
Abbildung 4-7: Architektur des Visual Composer .....	45
Abbildung 4-8: Design Board des Visual Composer.....	46
Abbildung 4-9: Struktur einer Web Dynpro Component .....	50
Abbildung 4-10: Component Modeler.....	52
Abbildung 4-11: Data Modeler.....	53
Abbildung 4-12: Navigation Modeler .....	54
Abbildung 4-13: View Designer .....	55
Abbildung 4-14: Universal Worklist für den Zugriff auf Prozessaufgaben .....	59
Abbildung 4-15: Process Composer.....	60
Abbildung 5-1: Zugriffsebenen auf Daten einer BEM-Maßnahme .....	68
Abbildung 5-2: Informationsumfang in Abhängigkeit der Nutzerrolle.....	69
Abbildung 5-3: BPM-Prozess zur Fallklärung in Phase 2 .....	72
Abbildung 5-4: Web Dynpro ABAP Oberfläche zur Anschreibenerstellung .....	74
Abbildung 6-1: BPM-Statusübersicht.....	79
Abbildung 6-2: Planungsübersicht der BSR Personaleinsatzplanung .....	80
Abbildung 7-1: Vergleich von Composite Applications mit BSP-Anwendungen	87

## Abkürzungsverzeichnis

ABAP	Advanced Business Application Programming
BEM	Betriebliches Eingliederungsmanagement
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Model and Notation
BSP	Business Server Pages
BW	Business Warehouse
CASE	Computer-aided software engineering
CE	Composition Environment
CHIP	Collaborative Human Interface Part
CRM	Customer Relationship Management
DC	Development Componenten
DSL	Domain specific language
EMF	Eclipse Modeling Framework
EPK	Ereignisgesteuerte Prozesskette
ERP	Enterprise Resource Planning
ESR	Enterprise Services Repository
GML	Generic Modeling Language
GML DOM	Generic Modeling Language Document Object Model
IDE	Integrated Development Environment
M-Team	Maßnahmenteam
MDA	Model Driven Architecture
MDS	Model Driven Software Development
MOF	Meta Object Facility
MVC	Model-View-Controller
NWDI	NetWeaver Development Infrastructure
NWDS	NetWeaver Developer Studio
OMG	Object Management Group
PI	Process Integration
SC	Software Component
SOA	Serviceorientierte Architektur
SOAP	Simple Object Access Protocol
UDDI	Universal Description, Discovery and Integration
UI	User Interface
UME	User Management Engine
UML	Unified Modeling Language
VC	Visual Composer
WSDL	Webservice Description Language
XML	Extensible Markup Language
XGL	Executable GUI Language
XSD	XML Schema Definition

# 1 Einleitung

## 1.1 Motivation

In den vergangenen zehn Jahren hat die Wichtigkeit der IT für die Geschäftsprozesse von Unternehmen stark zugenommen, denn nahezu alle Unternehmensaktivitäten sind von der IT abhängig und wären ohne sie nicht lauffähig. Darüber hinaus gewinnt eine flexible Architektur der Software an Bedeutung, um sich schnell an neue Anforderungen und Wettbewerbssituationen anzupassen. Auch eine steigende Komplexität der Software an sich und eine höhere Vernetzung von Softwaresystemen untereinander verlangen nach hohen Qualitätsstandards in der Softwareanalyse und -entwicklung.

In der Vergangenheit wurde bereits viel für die Vereinfachung von Softwareentwicklungen unternommen. So ist es insbesondere durch die stetige Erhöhung des Abstraktionsgrades von modernen Programmiersprachen heutzutage üblich, dass sich der Entwickler nicht um Sachverhalte wie Register, Speicherverwaltung und Kommunikationsprotokolle kümmern muss.<sup>1</sup> Doch selbst etablierte Standards, wie beispielsweise die objektorientierte Programmierung, sind oftmals nicht ausreichend, um gestellte Anforderungen in der gewünschten Komplexität, Flexibilität und Qualität mit vertretbarem Zeit- und Kostenaufwand umzusetzen. An diesem Punkt soll das Model-Driven-Software-Development (MDSD) ansetzen, d.h. es wird ein Softwaresystem durch ein Modell beschrieben, welches durch einen Rechner in eine ausführbare Anwendung transformiert werden kann. Dieser Schritt soll ein weiteres Mal den Abstraktionsgrad erhöhen und den Fokus auf die eigentliche technische Umsetzung verringern.

Auch die SAP AG, als marktführender Anbieter von Unternehmenslösungen<sup>2</sup>, hat in den letzten Jahren ihr Portfolio an modellgetriebenen Technologien und Entwicklungswerkzeugen stetig vorangetrieben und seit Mai 2011 im NetWeaver 7.3 vereint. In diesem Release ist unter anderem die zuvor eigenständige Komponente Composition Environment (CE) integriert.<sup>3</sup> Durch diesen Schritt haben SAP-Kunden nun geringe Hürden zu überwinden, um die in dieser Masterarbeit beschriebenen modellgetriebenen Entwicklungsansätze erproben und nutzen zu können, da die CE-Komponente bereits in der Installation des Java-Stacks des

---

<sup>1</sup> Vgl. Gitzel, Ralf (2006), S. 1.

<sup>2</sup> Vgl. SAP AG (2012), (siehe Internetverzeichnis).

<sup>3</sup> Vgl. SAP AG (2012b), (siehe Internetverzeichnis).

NetWeaver-Applikationsserver als zusätzlicher Useage Type installiert werden kann. Man kann daher sagen, dass die Software nun für den Großteil der SAP-Kunden in einer stabilen Version zur Verfügung steht und für den großflächigen Einsatz bereit ist.

## **1.2 Zielsetzung**

Die Arbeit hat den Anspruch, einen Überblick über die Möglichkeiten der modellgetriebenen Entwicklung auf Basis des NetWeaver CE 7.3 zu geben. Dabei soll zu Beginn die Notwendigkeit dieses Paradigmenwechsels in der Softwareentwicklung erläutert werden. Anschließend wird auf die wegweisenden Ansätze außerhalb des SAP-Universums in der theoretischen Betrachtung Augenmerk gelegt. Die hauptsächliche Arbeitsleistung wird in der Betrachtung der NetWeaver Composition Environment 7.3 liegen. Es sollen die involvierten Komponenten erklärt, der Verwendungszweck abgrenzt und das Zusammenspiel untereinander erläutert werden.

Anhand der strategischen Positionierung der Komponenten seitens SAP soll diese Arbeit dabei helfen, Technologieentscheidungen für konkrete Projekte zu treffen. Ein exemplarisches Vorgehen für die Umsetzung soll am Beispiel des Szenarios des „betrieblichen Eingliederungsmanagements“ gegeben werden. Als Ergebnis entsteht eine Anwendung, die den kollaborativen Geschäftsprozess der betrieblichen Wiedereingliederung nach häufiger oder lang andauernden Krankheiten elektronisch abbildet. Hierfür soll hinterfragt werden, ob das gesamte System ausschließlich per Modell beschrieben werden kann. Sofern manuelle Änderungen am generierten Quellcode in Form von Java bzw. ABAP Programmierung möglich sind, erfolgt eine Konsistenzprüfung des Modells, d.h. ist das Modell nach manuellen Änderung stets gültig und vollständig oder ist man ab einem bestimmten Punkt gezwungen, den Weg der Modellierung zu verlassen um anschließend klassisch zu programmieren. Diese Betrachtung wird individuell für alle Technologien aus dem Bereich der Integration, Prozessgestaltung und der Erstellung von Benutzeroberflächen durchgeführt.

Nach der praktischen Durchführung sollen die Änderungen im Entwicklungsprozess eingeschätzt werden und hinsichtlich der Dimensionen Kosten, Zeit, Qualität, Anforderungen an Entwickler und Organisation analysiert und beurteilt werden. Weiterhin sollen Vor- und Nachteile der modellgetriebenen Entwicklung gegenüber der traditionellen Vorgehensweise herausgestellt werden und Handlungsempfeh-

lungen für die Art der Entwicklung bei neuen Projekten anhand spezifischer Anforderungsgruppen gegeben werden.

### **1.3 Methodisches Vorgehen**

Der Titel der Masterthesis adressiert drei Kernbereiche. Dies sind die modellgetriebene Softwareentwicklung, das SAP Produkt NetWeaver CE in der Version 7.3 und ein konkretes Szenario bei der Berliner Stadtreinigung. In diesem Abschnitt soll erklärt werden, wie der Fokus dieser drei Themenkomplexe im Rahmen dieser Arbeit gelegt ist und welcher Zusammenhang zwischen den Aspekten besteht.

Ein wesentlicher Bestandteil der theoretischen Betrachtung stellen die Grundlagen zur modellgetriebenen Softwareentwicklung im zweiten Kapitel dar. Um die Notwendigkeit eines Paradigmenwechsels für den Softwareentwicklungsprozess aufzuzeigen, werden die Schwachstellen der traditionellen Softwareentwicklung beleuchtet. Nachdem der Leser hierfür sensibilisiert ist, werden Begriffe erläutert und Konzepte der modellgetriebenen Softwareentwicklung beschreiben, welche der Problemlösung dienen. Da es sich bei der modellgetriebenen Softwareentwicklung um keinen Standard handelt, werden zum Ende des Kapitels Lösungsansätze aus verschiedenen Epochen betrachtet. Dies soll Evolutionsstufen dieser Disziplin der Informatik aufzeigen, um später Lösungsansätze der SAP bewerten zu können.

Im dritten Kapitel wird der SAP NetWeaver CE 7.3 betrachtet. Um die strategische Positionierung des Produkts zu verdeutlichen, wird das Unternehmen SAP in aller Kürze vorgestellt und der Trend zu serviceorientierten Anwendungsarchitekturen beschrieben. Schließlich wird die Bedeutung und Einsatzszenarien der Composition Environment (CE) aufgezeigt sowie das Konzept hinter den so genannten „Composite Applications“ erklärt. Abgeschlossen wird dieses Kapitel durch eine kurze Betrachtung aller wesentlichen Komponenten des Applikationsservers.

Das nachfolgende Kapitel befasst sich mit den konkreten Werkzeugen und Technologien zur modellgetriebenen Entwicklung mit dem SAP NetWeaver CE. Hierfür werden ausgewählte Technologien der Service-, Prozess-, und UI-Schicht vorgestellt. Es erfolgt jeweils eine Betrachtung des Einsatzgebietes sowie des zugrundeliegenden Konzepts. Anschließend werden die Modellierungswerkzeuge hinsichtlich der Modellerstellung evaluiert und die Transformation des Modells zu einer ausführbaren Anwendung beleuchtet. Abschließend wird jede Technologie gesondert bewertet. Das Kapitel wird durch eine Zusammenfassung abgeschlossen, bei der unter anderem das Konzept der SAP mit den im Kapitel 2 vorgestellten Lösungsansätzen verglichen und eingeordnet wird.

Um das Zusammenspiel der verschiedenen Werkzeuge und Technologien zu verdeutlichen, werden sie im fünften Kapitel auf ihre Praxistauglichkeit im Projektalltag überprüft. Hierfür wird das Projekt des betrieblichen Eingliederungsmanagements bei der Berliner Stadtreinigung (BSR) beschrieben. Aus diesem Grund wird die BSR vorgestellt und die Bedeutung von Maßnahmen zur Gesundheitsförderung für das Unternehmen dargelegt. Nachdem der Ist- und Soll-Zustand beschrieben wurde, erfolgt die Beschreibung des Lösungsansatzes. Dieser beinhaltet Technologieentscheidungen sowie Besonderheiten des Projekts. Abschließend wird die Realisierungsphase reflektiert. Auf eine detaillierte Projektbeschreibung und Umsetzungsdokumentation wird aufgrund des Umfangs verzichtet. Neben einer Beschreibung der Projektorganisation werden Erfahrungen und Erkenntnisse präsentiert, welche allgemeingültig sind und somit einen Mehrwert bei zukünftigen Projekten darstellen.

Im sechsten Kapitel werden Auswirkungen auf den Softwareentwicklungsprozess betrachtet. Hierfür werden mehrere Aspekte untersucht. Begonnen wird mit der Analyse von organisatorischen Veränderungen aus Sicht der Personalanforderungen und des gesamten Softwarelebenszyklus. Weiterhin erfolgt eine Betrachtung der Qualität der auf diese Weise erstellten Software. Die Kostenanalyse rundet die Betrachtung durch einen Vergleich mit einem ähnlichen BSR-Projekt aus der Vergangenheit ab.

Im letzten Kapitel der Arbeit wird eine abschließende Betrachtung der Arbeitsergebnisse vorgenommen. Dazu werden die Erkenntnisse zusammengestellt und Schlussfolgerungen abgeleitet, die insbesondere Handlungsempfehlungen sowie einen Ausblick bezüglich des aktuellen und zukünftigen Stellenwerts von modellgetriebener Softwareentwicklung geben sollen.

## 2 Modellgetriebene Softwareentwicklung

### 2.1 Schwierigkeiten in der traditionellen Softwareentwicklung

In der Einleitung wurde beschrieben, dass die Softwareentwicklung einer stetiger Modernisierung und Vereinfachung unterworfen ist. Daher stellt sich die Frage, weshalb diese Änderungen notwendig sind und wo die Probleme beim traditionellen Vorgehen liegen. In den folgenden Abschnitten werden daher problematische Aspekte eingehender beschrieben.

#### *Entwicklungsgeschwindigkeit und Kosten*

Innovative IT-Systeme sind mit einem Wettbewerbsvorteil gleichzusetzen und daher ist für das Umsetzen einer guten Idee bzw. Anforderung die Zeit bis zum Produktivstart sehr wichtig. Wenn eine Software also früher im Einsatz sein kann, ist das für das Unternehmen gewinnbringend. Ähnliches gilt für das Aufholen eines Rückstands zu einem Mitbewerber und das schnelle Anpassen an neue Wettbewerbssituationen. Es gibt zwar bereits eine Vielzahl von Werkzeugen, die den Entwicklungsprozess beschleunigen, jedoch müssen noch viele Arbeiten manuell vorgenommen werden, weshalb die Aussage nahe liegt, dass in diesem Bereich noch Optimierungspotential zu finden ist.<sup>4</sup>

Auch Kosten sind ein Erfolgsfaktor für IT-Projekte, weswegen viele Projekte aufgrund zu hoher Kosten gar nicht begonnen werden, da Aufwand und Nutzen in keinem sinnvollen Verhältnis stehen.<sup>5</sup>

#### *Komplexitätsproblem*

Die IT-Branche steht unter dem Druck, Anforderungen umzusetzen, bei denen stetig der Umfang und die Qualitätserwartungen ansteigen. So ist es nicht verwunderlich, dass die Bewältigung höherer Komplexität der Software seit jeher eine der zentralen Herausforderungen in der Softwareentwicklung ist.<sup>6</sup> Neben einer ausgereiften Projektorganisation sind die technischen Rahmenbedingungen von entscheidender Bedeutung. In den letzten Jahrzehnten gab es bereits viele Fortschritte, so dass sich Entwickler vermehrt mit der Entwicklung von Geschäftslogik

---

<sup>4</sup> Vgl. Stahl, Thomas u.a. (2007), S. 3.

<sup>5</sup> Vgl. Ebenda, S. 11.

<sup>6</sup> Vgl. Suhl, Leena (2004), S. 1.

befassen konnten und nicht mit technische Aspekte wie Register, Speicherverwaltung und Kommunikationsprotokolle.<sup>7</sup>

Die Bewältigung dieser Herausforderung wurde in den letzten Jahrzehnten erfolgreich durch die Erhöhung des Abstraktionsgrades bei der Erstellung eines Softwaresystems begegnet. So ist beispielsweise eine deutliche Vereinfachung für Entwickler von der Assemblersprache zu modernen objektorientierten Programmiersprachen wie Java erreicht worden. Die Betrachtung dieses Aspektes wird im Unterkapitel 2.2 fortgesetzt und dort näher beschrieben.

Es ist bereits absehbar, dass sich der Trend der zunehmenden Komplexität weiter fortsetzen wird und die Softwareindustrie diesem Verlauf mit neuen technischen Fortschritten oder methodischen Ansätzen unterstützen muss.

#### *Spezifikations- und Dokumentationsprobleme*

Oftmals gehen Softwaremodellierungstools nicht über den Zweck der Dokumentation oder der Erzeugung von Codegerüsten aus statischen Modellen hinaus. Daraus resultiert, dass während der Implementierung die Modelle aus den Analyse- und Entwurfsphasen nicht mehr ausreichend gepflegt werden. Folglich reflektieren die Modelle nicht mehr das aktuelle Design des Codes, was zur Folge hat, dass die Dokumentation der Software häufig lückenhaft und nicht aktuell ist.<sup>8</sup>

Ausgehend von einem iterativen Entwicklungsprozess, wie er in der Abbildung 2-1 dargestellt ist, wird in den ersten drei Phasen viel Zeit in das Erstellen von Modellen und verbalen Beschreibungen zum IT-System investiert. Dies soll das gemeinsame Verständnis des Systems sicherstellen und es auf eine verständliche Art beschreiben. Weiterhin bildet die Spezifikation die Grundlage für die Implementierungsphase. Da es keine direkte Verbindung zwischen Spezifikation und Code gibt, müssen bei Änderungen in der Erstellungsphase als auch für die Wartungsphase stets Dokumentation und Quellcode angepasst werden.<sup>9</sup> Diese theoretische Vorgehensweise entspricht oftmals jedoch nicht der Realität, folglich werden Änderungen an Software nicht ausreichend dokumentiert. Das hat zur Konsequenz, dass die Dokumentation ihre Aussagekraft verliert, da sie nicht mehr das tatsächlich implementierte System widerspiegelt.<sup>10</sup>

---

<sup>7</sup> Vgl. Gitzel, Ralf (2006), S. 1.

<sup>8</sup> Vgl. Universität Ulm (2009), (siehe Internetverzeichnis).

<sup>9</sup> Vgl. Kleppe, Anneke / Warmer, Jos / Bast, Wim (2003), S. 2.

<sup>10</sup> Vgl. Rechenberg, Peter / Pomberger, Gustav (2006):, S. 820.

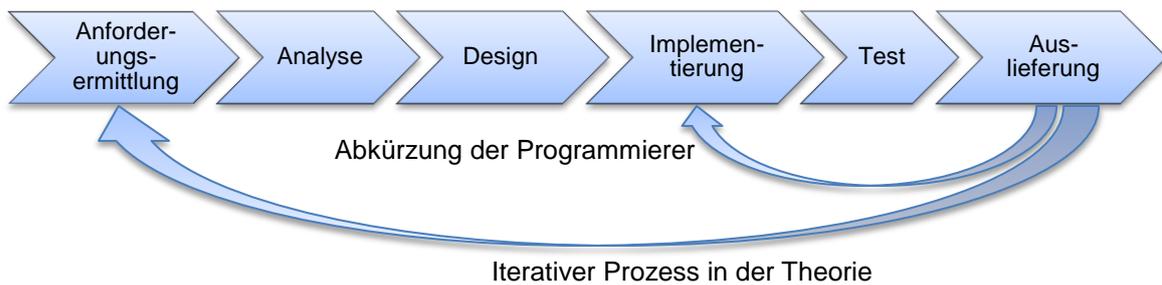


Abbildung 2-1: Phasen des traditionellen Softwareentwicklungszyklus

Quelle: Eigene Darstellung in Anlehnung an Kleppe, Anneke/ Warmer, Jos/ Bast, Wim (2003), S. 3.

Gründe für die in der Abbildung dargestellte Abkürzung sind meist in dem zusätzlichen Aufwand für die Anpassung der Modelle und Dokumentation zu suchen.<sup>11</sup> Weiterhin stehen Entwickler häufig unter Zeitdruck und erachten diesen Aspekt als weniger wichtig für eine funktionierende Software.<sup>12</sup>

### *Qualitätsprobleme*

Entwickler haben bei der Umsetzung von Anforderungen viel Freiraum in der Gestaltung des Quellcodes. Aus der Spezifikation gibt es keine Aussagen über den inneren Aufbau der Quellcodeartefakte, so dass Problemstellungen auf eine vielfältige Art und Weise gelöst werden können. Ausgehend von der Annahme, dass es guten und schlechten Quellcode für die gleiche Problemstellung gibt, stellt sich die Frage, was die Einflussfaktoren für die Qualität der Entwicklungsarbeit sind.

Zum einen werden die Arbeitsergebnisse durch Zeitdruck negativ beeinflusst. Eine grundlegend funktionierende Komponente, die alle funktionalen Anforderungen erfüllt, ist zwar mit relativ wenig Aufwand herzustellen, jedoch sind dann Mängel an nicht-funktionalen Anforderungen wie Fehlertoleranz, Benutzbarkeit und Wartbarkeit zu erwarten. Weiterhin ist die Qualität abhängig von der Motivation und den Kenntnissen des Entwicklers.<sup>13</sup>

### *Anpassungs- und Wartungsprobleme*

Insbesondere betriebswirtschaftliche Anwendungen unterliegen im Laufe ihres Lebenszyklus vielen Änderungen. Die beiden vorherigen Abschnitte bilden dabei bereits die Ursache für das Problem der Anpassungs- und Wartungsprobleme, denn Voraussetzung für problemlose Anpassungen in bestehender Software ist

<sup>11</sup> Vgl. Stahl, Thomas u.a. (2007), S. 3.

<sup>12</sup> Vgl. Kleppe, Anneke / Warmer, Jos / Bast, Wim (2003), S. 5.

<sup>13</sup> Vgl. Stahl, Thomas u.a. (2007), S. 15.

zum einen eine gute Dokumentation und zum anderen ein Softwaredesign, das flexible Änderungen zulässt.

Es ist essentiell, dass sich Entwickler schnell in ein Softwaresystem einarbeiten können, um die gewünschten Änderungen durchführen zu können. Aufgrund der vielen Freiheitsgrade in der traditionellen Softwareentwicklung und verschiedenen Denkweisen von Entwicklern, kann dies jedoch ein zeitaufwändiges und somit auch teures Unterfangen werden. Weiterhin besteht die Möglichkeit, dass durch Arbeiten am System Folgefehler auftreten, denn auch bei kleinen Änderungen am Quellcode können sich große Auswirkungen herausstellen, welche durch umfangreiche Tests überprüft werden müssen. Dieser Testaufwand muss daher auch bei der Schätzung einer Änderung mit berücksichtigt werden.<sup>14</sup>

### *Technologieprobleme*

Die Grundlage dieses Problems ist, dass eine Implementierung im klassischen Sinne stets technologieabhängig ist. Der Quellcode, welcher die einzige formale Beschreibung des Systems darstellt, funktioniert daher nur auf einer Plattform, für die sich im Vorfeld der Entwicklung entschieden worden ist. Wenn man nun den IT-Markt beobachtet, stellt man fest, dass jährlich mehrere neue Technologien erscheinen, welche entscheidende Vorteile mit sich bringen können. So können beispielsweise mit neuer Technologie seit längerem bestehende Problemstellungen gelöst werden oder der Support für ein älteres Produkt kann aufgrund des neueren eingestellt werden. Dies zeigt, dass Technologiewechsel für Unternehmen in bestimmten Situationen unabdingbar sind.<sup>15</sup>

Für ein bestehendes Softwaresystem kann dies bedeuten, dass im Rahmen der Technologieumstellung der Quellcode nicht mehr lauffähig ist und daher entweder angepasst oder vollständig neu implementiert werden muss. Selbst bei einem Update auf eine neuere Version des Produkts können diese Probleme auftreten. Demzufolge sind Technologiewechsel mit hohen Kosten verbunden und vorherige Investitionen verlieren ihren Wert.<sup>16</sup>

---

<sup>14</sup> Vgl. Kleppe, Anneke / Warmer, Jos / Bast, Wim (2003), S. 5.

<sup>15</sup> Vgl. Ebenda, S. 4.

<sup>16</sup> Vgl. Ebenda, S. 4.

## 2.2 Grundidee und Begriffsbestimmung

### 2.2.1 Ansatz

Aufgrund der zuvor genannten Schwächen der traditionellen Softwareentwicklung ergibt sich Handlungsbedarf. Als ein Lösungsansatz ist die modellgetriebene Softwareentwicklung identifiziert worden. Als Definition für modellgetriebene Softwareentwicklung (Model Driven Software Development, MDSD) dient jene von Thomas Stahl. Demnach „ist [sie] ein Oberbegriff für Techniken, die aus formalen Modellen automatisiert lauffähige Software erzeugen“.<sup>17</sup> Die Idee dahinter ist, dass zentrale Artefakte eines Softwaresystems als formales Modell vorliegen, die durch Interpreter oder Generatoren ohne manuellen Implementierungsaufwand in ein lauffähiges System überführt werden können. Derweil wird, abweichend von der traditionellen Entwicklung, nicht eine 3GL-Programmiersprache, wie es zum Beispiel bei Java der Fall ist, verwendet, sondern eine domänenspezifische Beschreibungssprache.<sup>18</sup>

Ein formales Modell muss so exakt und so ausdrucksstark wie möglich die durch die Software zu erbringende Funktionalität beschreiben. Das ist nur dann möglich, wenn die Elemente eines solchen Modells mit Semantik hinterlegt sind, welche eindeutig ein bestimmtes Verhalten zur Laufzeit bzw. eine bestimmte Software-Struktur definiert.<sup>19</sup> Modelle sind daher nicht nur eine grafische Darstellung des Programmcodes (z.B. ein Rechteck pro Klasse), sie spiegeln auch die innere Struktur von Softwaresystemen wieder, die auf dem niedrigeren Abstraktionsniveau von Programmiersprachen verloren geht. „Wie gut und konsequent ausgeprägt diese Struktur ist, beeinflusst in direkter Weise die Aspekte Entwicklungsgeschwindigkeit, Qualität, Performance, Wartbarkeit, Interoperabilität und Portabilität, also extrem wichtige ökonomische Schlüsselfaktoren.“<sup>20</sup> Entsprechend dieser Aussage ist der Erfolg des modellgetriebenen Ansatzes als nächste Evolutionsstufe generell nur schwer prognostizierbar, da zum einen die Güte der Modellierungssprache selbst ausschlaggebend ist und zusätzlich inwiefern sie für die technische bzw. fachliche Domäne anwendbar und geeignet ist. Am Beispiel der Herstellungsautomatisierung im Automobilbau wird dies veranschaulicht. Es wird davon ausgegangen, dass das Kundenbestellformular ein formales Modell für einen PKW ist, da dort alle relevanten Parameter beschrieben sind, und die Herstellung des

---

<sup>17</sup> Stahl, Thomas u.a. (2007), S. 11.

<sup>18</sup> Vgl. Vogel, Oliver u.a. (2009), S. 170.

<sup>19</sup> Vgl. Universität Ulm (2009), (siehe Internetverzeichnis).

<sup>20</sup> Stahl, Thomas u.a. (2007), S. 15.

Autos, wie heutzutage in der Großserienfertigung üblich, weitestgehend automatisiert ist. Die Domänenbindung wird deutlich, wenn man versucht mit der gegebenen Modellierungssprache (Bestellformular) und der Produktionsstraße des Automobilherstellers Fertighäuser zu produzieren.<sup>21</sup>

Die folgende Abbildung gibt einen Überblick in die Begriffswelt der modellgetriebenen Softwareentwicklung und stellt diese in Beziehung zueinander. In den folgenden Abschnitten werden diese Begriffe aufgegriffen und näher erläutert.

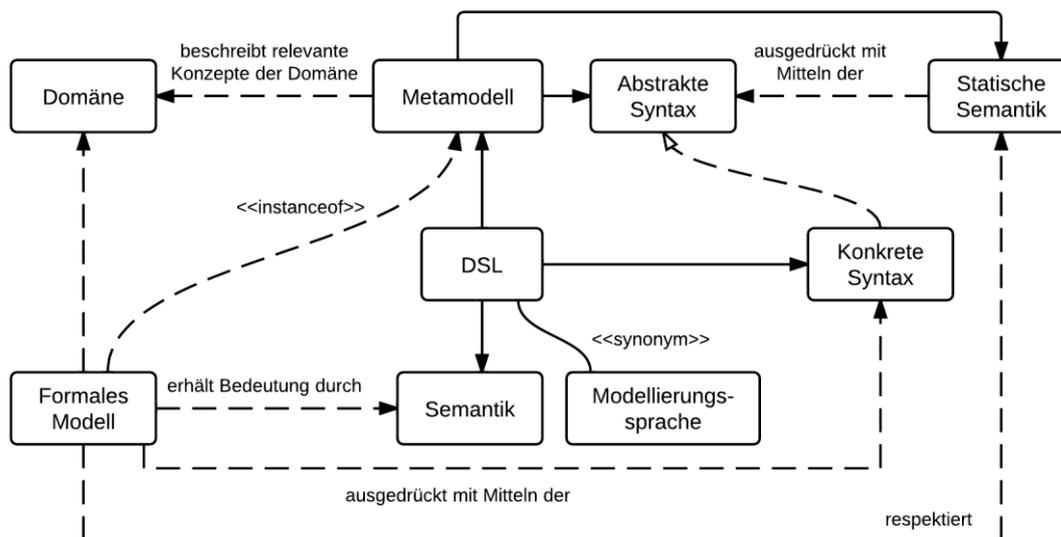


Abbildung 2-2: Begriffsbeziehungen

Quelle: Eigene Darstellung in Anlehnung an Stahl, Thomas u.a. (2007), S. 28.

## 2.2.2 Modellierungssprache

Im Grunde ist jede formale Sprache, die von Generatoren verstanden werden kann, eine Modellierungssprache. Sowohl Textdokumente als auch Tabellen lassen sich syntaktisch und semantisch so aufbauen, dass sie einer bestimmten Struktur folgen und somit von Generatoren in Quellcode transformiert werden können. Eine Modellierungssprache ist also nicht unbedingt eine grafische Sprache.<sup>22</sup>

Es gibt viele Modellierungssprachen für spezifische Einsatzgebiete in der Informatik. Ein Beispiel für eine grafische Notation ist die Unified Modeling Language (UML), welche sich als Standard in der Softwareentwicklung etabliert hat. Die verschiedenen Diagrammtypen dienen der Spezifikation, Konstruktion, Visualisierung

<sup>21</sup> Vgl. Stahl, Thomas u.a. (2007), S. 3f.

<sup>22</sup> Vgl. Pietrek, Georg / Trompeter, Jens (2007), S. 49.

und Dokumentation von Modellen für Softwaresysteme. Hierbei deckt UML ein weites Spektrum ab und kann sowohl statische als auch dynamische Aspekte einer Software beschreiben.<sup>23</sup> Ein Beispiel einer weiteren grafischen Modellierungssprache, welche ein konkretes Einsatzgebiet fokussiert, ist Business Process Model and Notation (BPMN) aufzuführen. Sie zielt auf die Domäne der Geschäftsprozessmodellierung ab.

Eine domänenspezifische Sprache (DSL) ist eine Beschreibungssprache für ein spezifisches Anwendungsgebiet. Sie ist aufgrund des Fokus auf ein bestimmtes Fachgebiet hochgradig angepasst, d.h. die Sprache löst Aufgaben der Domäne sehr effizient, ist aber gerade aufgrund der Spezialisierung außerhalb der Domäne kaum nutzbar.<sup>24</sup> Dieser Sachverhalt wurde bereits im Abschnitt 2.2.1 durch das Beispiel der Fließbandfertigung im Automobilbereich beschrieben. Das Bestellformular stellt in diesem Fall die domänenspezifische Sprache dar. DSLs werden nicht nur im Kontext der modellgetriebenen Entwicklung verwendet, sondern haben bereits eine lange Tradition in der Informatik. Bekannte Beispiele sind reguläre Ausdrücke zur Beschreibung von Zeichenketten oder SQL als Datenbanksprache für relationale Datenbanken.<sup>25</sup>

Die domänenspezifische Sprache stellt das Bindeglied zwischen menschlichen Domänenexperten und Compiler, Generatoren bzw. Interpreter dar. Dies erklärt die Anforderung an die Sprache, dass sie von Mensch und Maschine verstanden werden muss. Der Domänenexperte muss den Sachverhalt effizient abbilden und der Computer muss die Beschreibung auswerten und in Quellcode transformieren können. Neben der Syntax einer DSL ist auch die Semantik von Bedeutung, dies wird am Beispiel eines Klassendiagramms sichtbar. Wird das Diagramm im Kontext der UI-Entwicklung betrachtet, sind andere Ergebnisse zu erwarten als bei der Betrachtung aus Sicht der Datenbankentwicklung.<sup>26</sup>

Domänenspezifische Sprachen können nach fachlichen und technischen DSL kategorisiert werden. In der Vergangenheit wurden mehrheitlich technische domänenspezifische Sprachen verwendet. Diese haben den Nutzen, dass sie die Programmierung an sich erleichtern. Im Zuge der modellgetriebenen Entwicklung hat sich jedoch auch das Potential für fachliche DSL ausgeweitet. Sie sind maßgeblich für die Formalisierung und Effizienzsteigerung der Modellierung verantwortlich und ermöglichen den fachlichen Ansprechpartnern die Modellierung, da sie in diesem

---

<sup>23</sup> Vgl. Pietrek, Georg / Trompeter, Jens (2007), S. 50.

<sup>24</sup> Vgl. Völter, Markus (2005): S. 1.

<sup>25</sup> Vgl. Pietrek, Georg / Trompeter, Jens (2007), S. 57.

<sup>26</sup> Vgl. Ebenda, S. 57.

Fall die Domänenexperten sind. Allein aus diesem Aspekt heraus lassen sich Effizienz- und Qualitätssteigerungen erklären, da auf diesem Weg keine manuelle Transformation von verbalen Beschreibungen eines Fachexperten in eine technische Umsetzung notwendig ist.<sup>27</sup> Für den Softwareentwicklungsprozess ergeben sich aus der Verwendung von fachlichen domänenspezifischen Sprachen folgende weitere Vorteile:

- Die Erhöhung des Abstraktionsniveaus und der Formalisierung ermöglichen eine schnelle Erfassung von Spezifikationen.
- Die Wartungsphase profitiert durch die schnelle Umsetzung von Änderungsanforderungen, da diese sich in einer DSL effizienter als in einer verbalen Sprache umsetzen lassen.
- Die Semantik einer DSL ist eindeutig, daher sind Interpretationsspielräume minimiert und das Risiko von Missverständnissen zwischen Fachbereich und IT wird signifikant gesenkt.
- Bei der Transformation des Modells in ausführbaren Quellcode ist aufgrund der begrenzten Sprachelemente einer DSL ein geringeres Fehlerrisiko zu erwarten. Dies steigert die Softwarequalität.<sup>28</sup>

### 2.2.3 Metamodell

Eine Domäne stellt ein begrenztes Interessen- oder Wissensgebiet dar, welches eine gewisse Struktur aufweist. Das Verständnis dieser Struktur ist für die modellgetriebene Softwareentwicklung wichtig und muss für die automatisierte Bearbeitung verstanden und formal beschrieben werden.<sup>29</sup> Die formalisierte Beschreibung einer Domäne stellt das Metamodell dar. Es definiert die abstrakte Syntax und statische Semantik der Modellierungssprache.<sup>30</sup>

Es kann beispielsweise das Konstrukt „Buch“ im Falle eines UML Klassendiagramms als Klasse abgebildet werden. Der Titel eines Buches ist entsprechend der Regeln der UML Notation ein Attribut "titel" der Klasse "Buch". Die folgende Abbildung visualisiert das Modell dieser Klasse.

---

<sup>27</sup> Vgl. Pietrek, Georg / Trompeter, Jens (2007), S. 64.

<sup>28</sup> Vgl. Ebenda, S. 58ff.

<sup>29</sup> Vgl. Rempp, Gerhard u.a. (2011), S. 18f.

<sup>30</sup> Vgl. Stahl, Thomas u.a. (2007), S. 29.

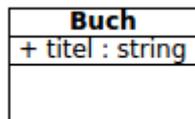


Abbildung 2-3: UML Klassendiagramm für die Entität „Buch“

Quelle: Eigene Darstellung.

Man erkennt, dass die Modellierungssprache bestimmten Regeln folgt, dies wiederum zeigt, dass ein Modell selbst durch ein Modell, das Metamodell, spezifiziert werden kann. Jedes in einem Modell verwendbare Element ist daher in dem Metamodell der Modellierungssprache definiert. Die Metamodellelemente und ihre Beziehung untereinander werden als abstrakte Syntax bezeichnet.

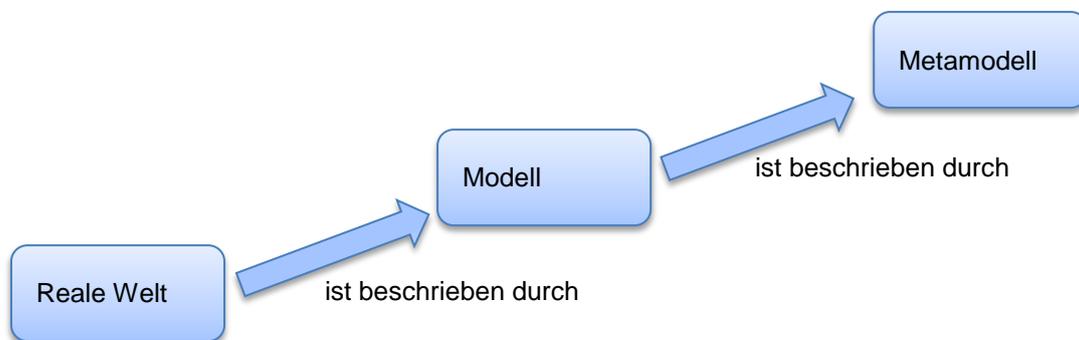


Abbildung 2-4: Beschreibungshierarchie

Quelle: Eigene Darstellung in Anlehnung an Pietrek, Georg / Trompeter, Jens (2007), 53.

Ausgehend von dieser Betrachtung ist anzudeuten, dass im Prinzip eine unendliche Anzahl von Metaebenen möglich ist. Das Konzept der Model Driven Architecture (MDA) schlägt beispielsweise vier Stufen vor, diese werden im Abschnitt 2.5.2 eingehender erläutert.

Formale Modelle werden zwar in der konkreten Syntax einer Modellierungssprache definiert, da sie aber eine Instanz des Metamodells darstellen, kann die syntaktische Korrektheit eines Modells anhand der abstrakten Syntax des Metamodells überprüft werden.<sup>31</sup> Die Trennung der konkreten von der abstrakten Syntax ist wichtig, denn die abstrakte Syntax wird zur automatisierten Verarbeitung der Modelle herangezogen. Die konkrete Syntax hingegen bildet die Schnittstelle zum Modellierer und ist Voraussetzung bei der Erstellung von Modellen. „Ihre Qualität entscheidet z.B. über die Lesbarkeit von Modellen.“<sup>32</sup> So beschreibt beispielswei-

<sup>31</sup> Vgl. Petrasch, Roland / Meimberg, Oliver (2006), S. 20.

<sup>32</sup> Stahl, Thomas u.a. (2007), S. 60.

se die abstrakte Syntax der Programmiersprache Java, dass es Klassen gibt, welche einen Namen haben und von einer anderen Klasse erben können. Die konkrete Syntax hingegen legt fest, dass eine Klasse mit dem Schlüsselwort „class“ definiert wird und die Basisklasse mit dem Schlüsselwort „extends“ angegeben wird.<sup>33</sup>

Weiterhin enthält ein Metamodell statische Semantik, das sind Bedingungen, welche das Modell erfüllen muss. So kann beispielsweise vorgeschrieben werden, dass eine Variable vor ihrer ersten Verwendung deklariert werden muss. Dies ist bei der Validierung von Modellen hilfreich, da sonst Fehler erst beim Kompilieren bzw. beim Test zur Laufzeit auffallen würden.<sup>34</sup>

Das Metamodell selbst muss ebenfalls von einer entsprechenden Sprache, der Metasprache, beschrieben werden. Zur Definition der Modellierungssprache kommen beispielsweise die Metasprachen EBNF (Erweiterte Backus-Naur-Form) oder XML (Extensible Markup Language) in Frage.<sup>35</sup>

Eine praxisnahe und verbreitete Technologie zur Umsetzung von Modellen und Metamodellen ist die Beschreibung durch XML. Eine Schemadefinition (XSD) dient hierbei als Metamodell, denn sie beschreibt die Struktur schemakonformer XML-Dokumente. Da eine XSD selbst ein XML-Dokument mit vorgeschriebenen Regeln ist, gibt es eine XSD, die den Aufbau einer XSD beschreibt (Metametamodell).<sup>36</sup>

### **2.3 Varianten der Modelltransformation**

Grundsätzlich gibt es zwei Varianten, wie die Überführung des Modells in eine lauffähige Anwendung ablaufen kann. Die Unterscheidung beruht auf dem Zeitpunkt der Modellnutzung. Zum einen kann aus dem Modell zum Compile-Zeitpunkt Quellcode erzeugt werden. Hierbei erzeugen Generatoren Quelltext in einer anderen Programmiersprache wie Java. Im Prinzip handelt es sich also um einen Übersetzer von der Modellierungssprache in eine ausführbare Programmiersprache.<sup>37</sup> Dieser Ansatz ist derzeit am weitesten verbreitet.<sup>38</sup> Besonders bei strukturellen Aspekten eines Systems wird dieses Vorgehen verwendet, da dies mehrere Vorteile mit sich bringt. Bei der Modelltransformation werden Templates verwendet, das sind Abbildungsregeln, wie Elemente aus dem Modell im Quellcode dar-

---

<sup>33</sup> Vgl. Stahl, Thomas u.a. (2007), S. 29.

<sup>34</sup> Vgl. Ebenda, S. 30.

<sup>35</sup> Vgl. Pietrek, Georg / Trompeter, Jens (2007), S. 54ff.

<sup>36</sup> Vgl. Stahl, Thomas u.a. (2007), S. 74.

<sup>37</sup> Vgl. Ebenda, S. 12.

<sup>38</sup> Vgl. Ebenda, S. 173.

gestellt werden.<sup>39</sup> Innerhalb dieses Templates kann einheitlich auf den finalen Quellcode Einfluss genommen werden, hierdurch erhält die Anwendung eine hohe Flexibilität. Würde man die Flexibilität durch eigene Programmierung im Quellcode verankern wollen, würde man nicht ohne Programmierparadigmen wie Polymorphie und Reflection auskommen. Dies würde jedoch die Performance des Systems negativ beeinflussen, da sie zur Laufzeit einer festen Codierung unterlegen sind. Die Codegenerierung ermöglicht daher eine hohe Leistungsfähigkeit ohne Verlust der Flexibilität. Weiterhin ist die Codegröße geringer, da man bereits zum Compile-Zeitpunkt weiß, welche Funktionen zur Laufzeit benötigt werden. Darüber hinaus lässt sich eine generierte Software im Vergleich zu komplexen generischen Frameworks besser analysieren und debuggen.<sup>40</sup>

Die Vollständigkeit der Codegenerierung bildet ein weiteres Unterscheidungsmerkmal. Einerseits ist es möglich, eine vollständige Codegenerierung durchzuführen, d.h. das gesamte System wird durch das Modell beschrieben und nach der Generierung sind keine manuellen Tätigkeiten am Quelltext notwendig. Dies stellt den Idealfall dar und ist mit hohem Aufwand verbunden, da jeder Aspekt der Software vom Modell berücksichtigt und durch ein Template abgedeckt sein muss. Ein weiterer Ansatz ist die Verwendung einer teilweisen Generierung mit Abschnitten von nicht-generiertem Code, welche durch den Generator nicht überschrieben werden. Dies stellt einen pragmatischen Ansatz für die Standardaufgaben dar, da diese effizient durch das Modell beschrieben werden können. Besonderheiten der Software werden hingegen weiterhin manuell codiert. Als letzte Möglichkeit kann der Generator nur einen Rahmen für die Software bereitstellen, welcher manuell ausgeprägt werden muss. Anschließend ist das Modell allerdings vom Quellcode getrennt, dies eliminiert viele Vorteile, die sich aus der modellgetriebenen Softwareentwicklung ergeben. Diese Methode kann bei der Erstellung von Prototypen sehr hilfreich sein.

Beim Interpreter hingegen wird zur Laufzeit das Modell eingelesen und inhaltsabhängig differenzierte Aktionen ausgeführt. Diese Methode eignet sich eher für die Steuerung des Verhaltens einer Anwendung. Dies kann z.B. das Auswerten einer Entscheidungstabelle zur Laufzeit sein. Interpreter haben den Vorteil, dass sie grundsätzlich leichter zu entwickeln und testen sind. Der größte Vorteil dieser Variante der Modellüberführung ist jedoch, dass Modelländerungen deutlich schneller

---

<sup>39</sup> Vgl. Pietrek, Georg / Trompeter, Jens (2007), S. 71.

<sup>40</sup> Vgl. Stahl, Thomas u.a. (2007), S. 141f.

im laufenden System übernommen werden. Gerade bei häufigen Änderungen des Modells ist hierdurch ein reibungsloser Betrieb sichergestellt.<sup>41</sup>

## 2.4 Auswirkungen auf identifizierte Schwächen

Der MDSD-Ansatz ist aus der Notwendigkeit heraus entstanden, bisherige Schwachstellen in der Softwareentwicklung zu verbessern. In Unterkapitel 2.1 „Schwierigkeiten in der traditionellen Softwareentwicklung“ wurden bereits problematische Aspekte beleuchtet. Nachdem der Gedanke hinter der modellgetriebenen Entwicklung erläutert wurde, werden nun die Auswirkungen für die identifizierten Problemfelder diskutiert.

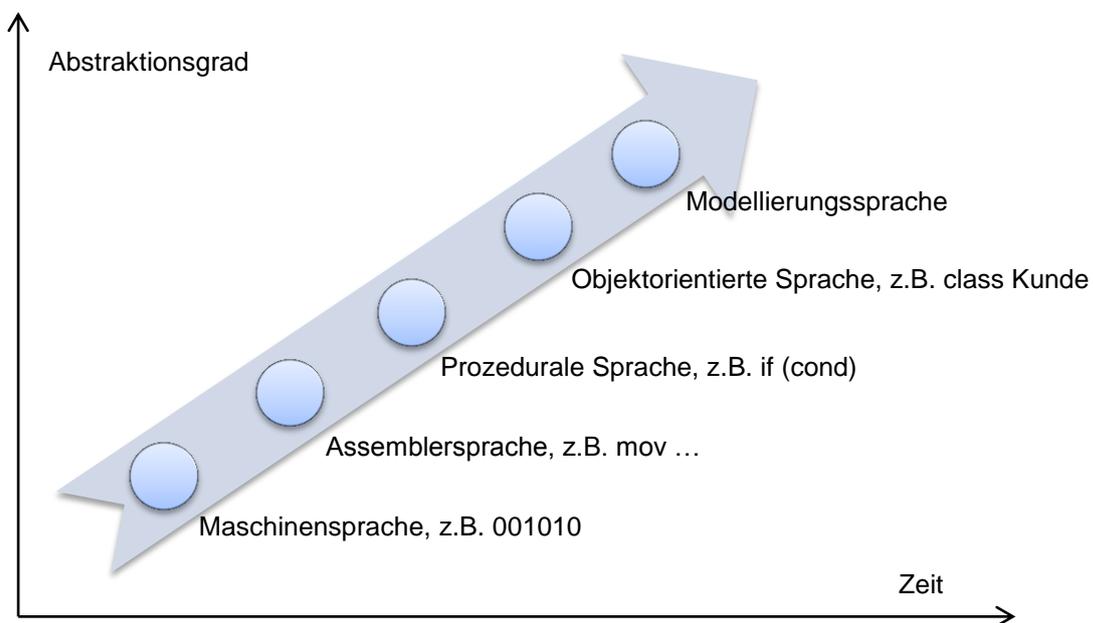


Abbildung 2-5: Evolution der Softwareentwicklung

Quelle: Eigene Darstellung in Anlehnung an Petrasch, Roland / Meimberg, Oliver (2006), S. 46.

Die Abbildung veranschaulicht den Trend der Steigerung des Abstraktionsgrades in der Softwareentwicklung. Genauso wie der Programmcode einen deutlich höheren Abstraktionsgrad als der Maschinencode besitzt, bieten auch Modelle im MDSD-Kontext eine abstraktere Sicht auf den Programmcode. So ist die modellgetriebene Entwicklung die nächste Stufe, welche mit den Fortschritten von der Assemblerprogrammierung zu den heutigen Hochsprachen verglichen werden kann.<sup>42</sup>

<sup>41</sup> Vgl. Stahl, Thomas u.a. (2007), S. 173f.

<sup>42</sup> Vgl. Ebenda, S. 4.

Aufgrund des höheren Abstraktionsniveaus von Modellen ist es möglich, technische Details auszublenden. Als Ergebnis erhält man eine konzeptionelle Sicht auf das Softwaresystem. Dieser Sachverhalt macht es für den Entwickler leichter, sich auf die Geschäftslogik der Anwendung zu konzentrieren und somit die Komplexität der Anwendung zu beherrschen.<sup>43</sup> Weiterhin hat das Ausblenden von technischen Aspekten eines Systems den Vorteil, dass keine systemspezifischen Implementierungen geschaffen werden. Die Transformation des Modells in ausführbaren Quellcode ist für verschiedene Plattformen möglich und begünstigt aus diesem Grund einen schnelleren und kostengünstigeren Technologiewechsel.<sup>44</sup>

Weiterhin sind die Probleme in der Spezifikation und Dokumentation durch den Ansatz als gelöst zu betrachten. Das Modell als ein zentrales Artefakt für die Dokumentation und Spezifikation bildet gleichzeitig auch die Basis des lauffähigen Systems. Dies bedeutet, dass Änderungen an der Software nur über das Modell möglich sind und damit der Entwickler keine Möglichkeit zur Abkürzung im Entwicklungsprozess hat (vgl. Abbildung 2-1: Phasen des traditionellen Softwareentwicklungszyklus).

Die modellgetriebene Softwareentwicklung hat darüber hinaus das Potential, die Entwicklungsgeschwindigkeit zu erhöhen und damit Kosten zu senken. Dies ist unter anderem mit der Automatisierung wiederkehrender Abläufe begründet. Es werden Routinetätigkeiten reduziert und der Entwickler kann sich auf die Umsetzung von fachlichen Anforderungen konzentrieren. Weiterhin besitzen die Phasen Design und Implementierung eine größere Schnittmenge, so dass die Implementierung besser auf den Arbeitsergebnissen der Design-Phase aufbauen kann. Insbesondere wenn ein langwieriger Wartungszeitraum für die Software prognostiziert wird, steigt das Potential, die Gesamtkosten für die Software zu senken. Bei der Erstellung eines überschaubaren Produktes ohne längere Wartungsphase kann hingegen der Aufwand über den der traditionellen Entwicklung steigen. Dieser Zusammenhang wird im Unterkapitel 2.5 „Beispiele für konkrete Ansätze der Umsetzung“ erläutert.<sup>45, 46</sup>

Die Qualität der aus einem Modell entstandenen Software hängt stark von der Umsetzung in lauffähigen Code ab. Somit lässt sich nicht pauschal sagen, dass die Softwarequalität steigt. Es besteht jedoch Potential, einen Qualitätsgewinn zu erreichen, da Systeme eine einheitliche, getestete und dokumentierte Architektur

---

<sup>43</sup> Vgl. Gitzel, Ralf (2006), S. 55.

<sup>44</sup> Vgl. Ebenda, S. 55.

<sup>45</sup> Vgl. Pietrek, Georg / Trompeter, Jens (2007), S. 16f.

<sup>46</sup> Vgl. Stahl, Thomas u.a. (2007), S. 15.

erhalten.<sup>47</sup> Sofern also bei der Erstellung des Generators ein hoher Qualitätsstandard angewendet worden ist, verbessert dies maßgeblich auch die Qualität des Endprodukts. Mindestens jedoch garantiert der MDSD-Ansatz eine gleich bleibende Qualität des generierten Quelltextes. Als kritisch kann jedoch betrachtet werden, dass auch Qualitätsausreißer nach oben, z.B. eine sehr ausgereifte Detaillösung in einer Anwendung durch die Starrheit des vorgegebenen Rahmens verhindert werden bzw. nur mit größerem Aufwand möglich sind, da der Generator zusätzlich angepasst werden muss. Dies ist jedoch für eine vereinzelt Detaillösung nur selten möglich.<sup>48</sup>

Die Anpassbarkeit und Wartung einer Anwendung wird erheblich vereinfacht. Zum einen, da die Konsistenz zwischen Modellen, der Anwendung und der Dokumentation gegeben ist und somit die Einarbeitungszeit verringert. Zum anderen, da bei der Anpassung von fachlichen Anforderungen nur das Modell geändert werden muss und die Implementierung vollständig ausgeblendet werden kann.<sup>49</sup> Selbst wenn Änderungen an der Software mehrheitlich am Modell vorgenommen werden, ist die Verständlichkeit des generierten Quellcodes trotzdem insbesondere bei Fehlersuchen von hoher Bedeutung. An dieser Stelle ergibt sich der Vorteil, dass Code, welcher aufgrund von Transformationsregeln generiert wurde, in sich sehr konsistent und damit einfach zu verstehen ist. Es zeigt sich beispielsweise eine Einheitlichkeit in Bezug auf Aufrufkonventionen, Bezeichnungen und Parameterübergabe.

## **2.5 Beispiele für konkrete Ansätze der Umsetzung**

### **2.5.1 Computer-aided Software Engineering (CASE)**

Der CASE Ansatz war in den 90er Jahren sehr beliebt und wurde als die Zukunft der Softwareentwicklung gehandelt. Bei diesem Ansatz ist man davon ausgegangen, dass man jegliche Software vollständig aus einem Modell erstellen kann. Die Modellierungssprache sollte dabei nicht domänenspezifisch, sondern für alle Einsatzszenarien geeignet sein.<sup>50</sup> Das Erschaffen einer solch allgemeingültigen Modellierungssprache, welche die Vielseitigkeit einer 3GL-Sprache wie z.B. Java ab-

---

<sup>47</sup> Vgl. Stahl, Thomas u.a. (2007), S. 16.

<sup>48</sup> Vgl. Pietrek, Georg / Trompeter, Jens (2007), S. 16.

<sup>49</sup> Vgl. Ebenda, S. 16.

<sup>50</sup> Vgl. Cooke, Daniel (1994), S. 1ff.

deckt, wurde durch proprietäre Ansätze verschiedener konkurrierender Hersteller forciert.<sup>51</sup>

Durch den fehlenden Fokus auf eine konkrete Domäne stellt CASE an sich den Anspruch, für alle Anwendungsgebiete passend zu sein. Dies ist jedoch in der Praxis vollkommen unrealistisch und bereitet erhebliche Probleme, da eine vorge-dachte Lösung nicht für alle Eventualitäten passt. Aus Erfahrungswerten weiß man, dass sich ca. 80% der Software sehr effizient erstellen lassen. Die fehlenden 20% bereiten aber große Schwierigkeiten, da CASE keine Einflussnahme auf den generierten Quellcode vorsieht.<sup>52</sup> Die manuelle Anpassung des generierten Quelltextes ist nur bedingt möglich, weil manuelle Änderungen nicht durch das Werkzeug geschützt werden, sondern stattdessen beim nächsten Generieren überschrieben werden. Dies hat ebenso zur Folge, dass man die grundsätzliche Architektur einer CASE-Anwendung nicht beeinflussen kann, da sie vom Hersteller des Werkzeugs fest vorgegeben ist. Als ein Vorteil der modellgetriebenen Software wurde genannt, dass man unabhängiger von Technologien ist. Viele CASE Tools unterstützen jedoch nur eine Zielplattform, so dass man sich bereits bei der Auswahl des Modellierungstools für die technische Implementierung entscheiden muss.<sup>53</sup> Weiterhin fehlten zu dieser Zeit geeignete Standards, was zur Folge hatte, dass viel Zeit- und Kostenaufwand zur Entwicklung der Infrastruktur investiert werden musste.<sup>54</sup>

In den Ausführungen der bisherigen Abschnitte wurde die modellgetriebene Softwareentwicklung stets als sehr flexibler Ansatz beschrieben, beispielsweise könne man selbst Einfluss auf Templates und die Architektur der Anwendung nehmen. Bei den CASE Tools ist man jedoch den proprietären Konzepten des Herstellers unterworfen.<sup>55</sup> Dies kann zwar in Einzelfällen hilfreich und zeitsparend sein, sofern alle vorgedachten Parameter des Herstellers mit dem tatsächlichen Verwendungszweck und der vorhandenen Systemlandschaft übereinstimmen, jedoch widerstrebt es dem Gedanken einer modernen modellgetriebenen Softwareentwicklung wie sie im Theorieteil dieser Arbeit vorgestellt wurde.

Aufgrund dieser gravierenden Nachteile stellte sich CASE als wenig vorteilhaft heraus. Daher sollte Computer-aided Software Engineering eher als Vorgänger des Model-Driven-Software-Development gesehen werden. Im Rahmen dieser

---

<sup>51</sup> Vgl. Gitzel, Ralf (2006), S. 54.

<sup>52</sup> Vgl. Stahl, Thomas u.a. (2007), S. 41f.

<sup>53</sup> Vgl. Pietrek, Georg / Trompeter, Jens (2007), S. 17f.

<sup>54</sup> Vgl. Gitzel, Ralf (2006), S. 54.

<sup>55</sup> Vgl. Pietrek, Georg / Trompeter, Jens (2007), S. 17f.

Masterarbeit soll der Ansatz trotzdem Beachtung finden, da dieser Weg für SAP zweckmäßig sein kann. Aufgrund der hohen Marktstärke und Verbreitung in Unternehmen kann SAP maßgeschneiderte Lösungen anbieten, welche sich gut in eigene Produkte integrieren und eine hohe Produktivität zulassen. Dies wird im Kapitel 4 „Werkzeuge und Technologien zur modellgetriebenen Entwicklung“ untersucht.

### 2.5.2 Model Driven Architecture (MDA)

Ein möglicher Weg zur Implementierung des MDSB ist der herstellerunabhängige Ansatz Model-Driven Architecture (MDA). Seitdem die Object Management Group (OMG) im Jahre 2000 eine Spezifikation für MDA veröffentlicht hat, wird dieser Ansatz unterschiedlich interpretiert und durchgeführt. Beim Ansatz des MDA wird der Programmcode durch automatisierte Transformationen aus Modellen erzeugt. Dafür setzt MDA mehrere Modelltransformationen und die Nutzung von UML als Modellierungssprache voraus.<sup>56</sup>

Im Gegensatz zu den CASE-Tools handelt es sich bei MDA-basierenden Lösungen wie openArchitectureWare, AndroMDA und Acceleo nicht um fertige Generatoren, sondern um Generatorframeworks. Der Aufbau des Generators ist Template-basiert, das heißt, es werden wartbare, übersichtliche und wiederverwendbare Definitionen der zu generierten Artefakte erstellt. MDA-Frameworks enthalten meist bereits fertige Templates für Standardaufgaben. Dies hat den Vorteil, dass der Nutzer Einfluss auf den generierten Quellcode nehmen und somit auf projektspezifische Bedürfnisse eingehen kann.<sup>57</sup>

MDA erfordert keine hundertprozentige Automatisierung. Da die Architektur eines Systems vollständig manuell erzeugt wird, ist MDA hochgradig flexibel und gewährt, im Gegensatz zum CASE-Ansatz, die vollständige Kontrolle während des Entwicklungsprozesses.<sup>58</sup>

Die OMG spezifiziert in ihrem MDA Guide drei verschiedene Abstraktionsebenen:

- **Computation Independent Model (CIM):** Das Modell ist eine Beschreibung der domänenspezifischen Zusammenhänge. Sie werden auf dieser Ebene völlig unabhängig von technischen Möglichkeiten erstellt. Die Umformung in die nächste Ebene wird manuell vorgenommen.

---

<sup>56</sup> Vgl. Pietrek, Georg / Trompeter, Jens (2007), S. 17f.

<sup>57</sup> Vgl. Ebenda, S. 17f.

<sup>58</sup> Vgl. Ebenda, S. 230f.

- **Platform Independent Model (PIM):** In dieser Ebene werden Anforderungen auf technischer Ebene beschrieben, jedoch erfolgt dies noch plattformunabhängig. Die Beschreibung erfolgt in der Regel durch UML-Modelle. Die Transformation in die nächste Ebene erfolgt automatisiert.
- **Platform Specific Model (PSM):** Das Modell wird um alle technischen Details erweitert, um auf einer konkreten Plattform, wie J2EE oder .NET, lauffähig zu sein. Hierbei kann es sich beispielsweise darum handeln, dass Datentypen einer bestimmten Programmiersprache verwendet werden. Nachdem das PSM definiert ist, kann hieraus der Quellcode für die Anwendung generiert werden.<sup>59</sup>

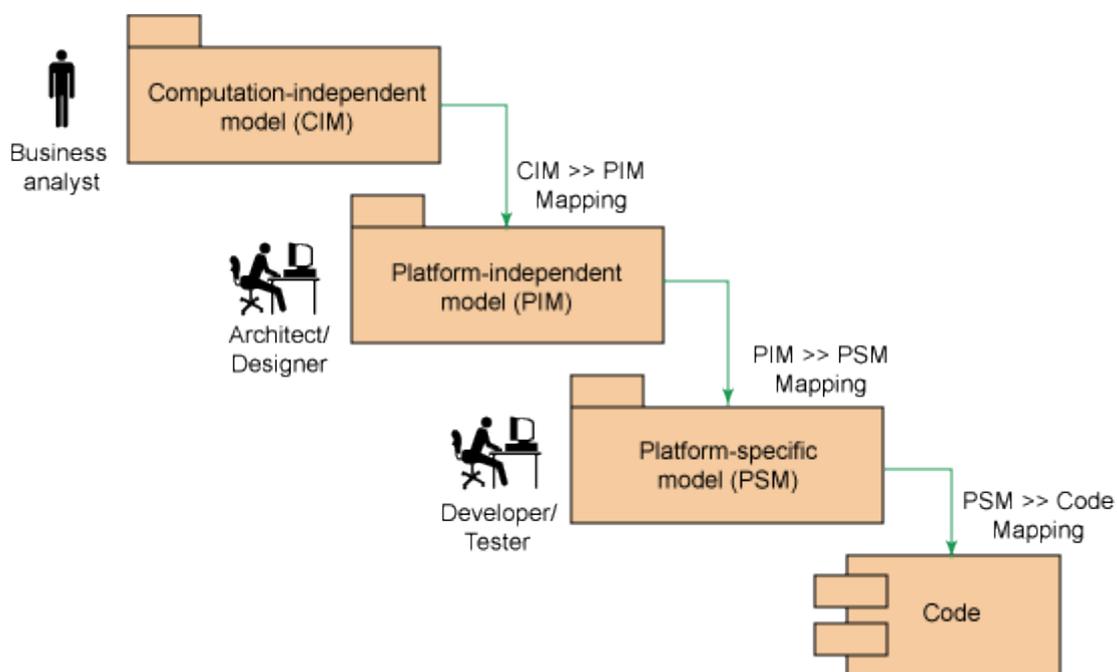


Abbildung 2-6: Ablauf der Abstraktionsebenen bei MDA

Quelle: Litoiu, Milena (2004), siehe Internetverzeichnis.

Das Konzept von Model Driven Architecture sieht zwei Meta-Ebenen zur Beschreibung der Modelle vor. Als Meta-Metamodellierungssprache definiert die OMG die Meta Object Facility (MOF). Diese beschreibt den formalen Aufbau des Metamodells. Die MOF kann daher als Infrastruktur für Sprachdefinitionen und Metamodellierung gesehen werden. Sie ist in der Metaebene M3 angesiedelt.<sup>60</sup> Die folgende Abbildung visualisiert die konzeptionelle Sicht auf die Metaebenen.

<sup>59</sup> Vgl. OMG (2003), S. 15 (siehe Datenträger).

<sup>60</sup> Vgl. Pietrek, Georg / Trompeter, Jens (2007), S. 54.

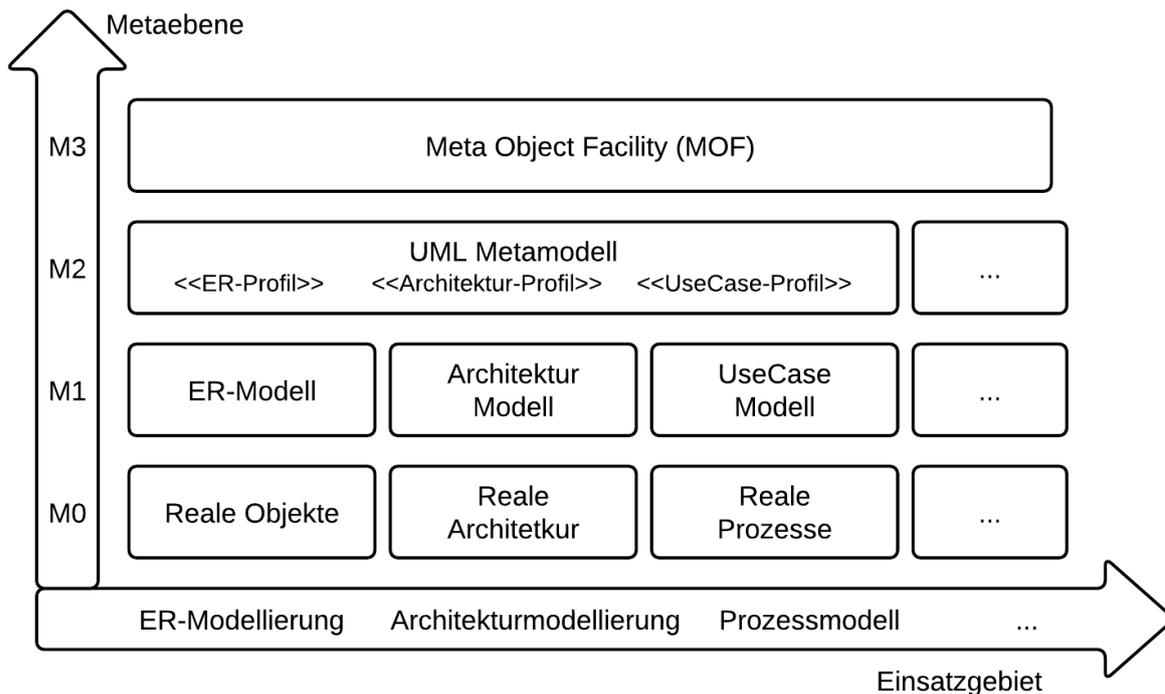


Abbildung 2-7: Metaebenen

Quelle: Eigene Darstellung in Anlehnung an Pietrek, Georg / Trompeter, Jens (2007), S. 55.

In der darunter liegenden Ebene (M2) werden konkrete Metamodelle beschrieben. Dies können beispielsweise existierende UML-Metamodelle, angepasste UML-Profile oder auch selbst erstellte EMF-ecore-Modelle sein, in denen domänenspezifische Sprachen beschrieben werden. In dieser Ebene werden die abstrakte Syntax und statische Semantik der Modelle in Metaebene M1 definiert, welche wiederum Bestandteile der realen Welt (M0) beschreiben.<sup>61, 62</sup>

Zusammenfassend kann festgehalten werden, dass es sich bei MDA um ein Konzept handelt, welches vor allem aufgrund der breiten industriellen Unterstützung und des Einsatzes von Standards bedeutsam für die zukünftige Entwicklung und die breite Akzeptanz des MDSD-Ansatzes ist.<sup>63</sup> Einige der verwendeten Standards sind jedoch noch nicht weit verbreitet oder stehen nicht in einer finalen Version zur Verfügung. Selbst die anerkannte Standardmodellierungssprache UML, die bei der MDA eine zentrale Rolle spielt, wird für den Einsatz in einigen Domänen als unpassend eingestuft.<sup>64</sup>

<sup>61</sup> Vgl. Pietrek, Georg / Trompeter, Jens (2007), S. 54.

<sup>62</sup> Vgl. OMG (2003), S. 51ff (siehe Datenträger).

<sup>63</sup> Vgl. Raistrick, Chris u.a. (2004), S. 8.

<sup>64</sup> Vgl. Gardner, Tracy / Yusuf, Larry (2006), (siehe Internetverzeichnis).

Weiterhin ist der MDA-Ansatz bisher vornehmlich theoretischer und abstrakter Natur. Es bestehen viele Spielräume für eigene Interpretationen und Implementierungen.<sup>65</sup> Ein wichtiger Bestandteil von MDA ist die mehrfache Transformation von Modellen, die sich auf unterschiedlichen Abstraktionsniveaus befinden, um letztlich zum Programmcode zu gelangen. Vor allem aber die Notwendigkeit der Transformation von einem Platform Independent Model (PIM) in ein Platform Specific Model (PSM) wird jedoch von der Literatur in Frage gestellt und stattdessen eine direkte Codegenerierung aus einem PIM als sinnvoller erachtet.<sup>66, 67</sup>

Es bleibt also zunächst abzuwarten, wie sich MDA und die Werkzeuge, die an dem Erfolg bzw. Misserfolg dieses Ansatzes eine große Rolle spielen dürften, weiter entwickeln werden.

---

<sup>65</sup> Vgl. Gitzel, Ralf (2006), S. 66.

<sup>66</sup> Vgl. Ebenda, S. 59.

<sup>67</sup> Vgl. Stahl, Thomas u.a. (2007), S. 36.

## 3 SAP NetWeaver 7.3 – Composition Environment

### 3.1 Grundlagen

#### 3.1.1 Die Firma SAP

Die SAP AG ist marktführender Anbieter von Unternehmenslösungen und weltweit der viertgrößte unabhängige Softwarehersteller.<sup>68</sup> Gegründet wurde das Unternehmen von fünf ehemaligen Mitarbeitern von IBM<sup>69</sup> im Jahre 1972. Die gemeinsame Vision war die Entwicklung einer Standard-Anwendungssoftware für die Echtzeitverarbeitung. Die Abkürzung SAP stand damals für „Systemanalyse und Programmentwicklung“, später wurde die Abkürzung mit „Systeme, Anwendungen und Produkte in der Datenverarbeitung“ allgemeiner interpretiert. Schlussendlich wurde im Jahr 2005 auf eine Abkürzung verzichtet. Seitdem ist SAP AG die offizielle Bezeichnung der Firma.<sup>70</sup>

Schwerpunkt der SAP AG ist seit jeher die Entwicklung von Software für große und mittelständische Unternehmen. Die Basis der SAP-Unternehmenslösung bildet das SAP ERP, welches bis 2003 die Bezeichnung SAP R/3 trug. Es handelt sich hierbei um ein Enterprise-Resource-Planning (ERP)-System, welches dabei hilft vorhandene Ressourcen (z.B. Kapital, Betriebsmittel oder Personal) eines Unternehmens möglichst effizient im betrieblichen Ablauf einzusetzen. Zum Produktportfolio gehören darüber hinaus noch viele branchenspezifische Lösungen sowie Produkte für Geschäftsanalysen und Anwendungsintegration.

Im Jahr 2010 beschäftigte das Unternehmen 55.000 Mitarbeiter und betreute über 183.000 Kunden in 130 Ländern. Von den ca. 8,2 Milliarden € Umsatzerlösen investierte SAP etwa 14% (1,7 Milliarden €) in Forschung und Entwicklung.<sup>71</sup> Dies verdeutlicht, dass SAP stets bestrebt ist, Innovationen zu fördern und das eigene Produktportfolio zu optimieren, um im Konkurrenzkampf gegen andere Anbieter wie beispielsweise Oracle zu bestehen.

---

<sup>68</sup> Vgl. SAP AG (2012), (siehe Internetverzeichnis).

<sup>69</sup> Claus Wellenreuther, Hans-Werner Hector, Klaus Tschira, Dietmar Hopp und Hasso Plattner.

<sup>70</sup> Vgl. SAP AG (2012d), (siehe Internetverzeichnis).

<sup>71</sup> Vgl. SAP AG (2012e), (siehe Internetverzeichnis).

Aktuell ist die SAP AG bestrebt, ihre Anwendungsarchitekturen zu flexibilisieren, sich im Geschäftsfeld des Cloud-Computings zu positionieren und die Verbreitung der In-Memory-Datenbank HANA voranzutreiben.<sup>72</sup>

### 3.1.2 Serviceorientierung in der Anwendungsarchitektur

Aufgrund der hohen Dynamik in der Geschäftswelt sind Unternehmen dazu angehalten, ihre Geschäftsprozesse flexibel anzupassen und neue Ideen zeitnah zu etablieren. Mit einer starren IT-Infrastruktur ist die Anpassung der Softwaresysteme für die optimale Geschäftsprozessunterstützung nur schwer möglich und von daher mit hohen Kosten verbunden. Der Begriff SOA (Serviceorientierte Architektur) ist in der IT-Branche weit verbreitet und wird von Experten als möglicher Lösungsansatz für den Wunsch nach hoher Flexibilität in der Anwendungsarchitektur erachtet.

Der Ansatz der SOA ist abstrakt, d.h. unabhängig von einer konkreten Implementierung. Daher handelt es sich bei diesem Architekturkonzept um keinen festgeschriebenen Standard. Die Grundlage einer SOA sind Services. In der Literatur und bei Produktanbietern sind viele Definitionen des Begriffs auffindbar. Sie alle vereinigen, dass Services wiederverwendbare, standardisierte und erweiterbare Softwarekomponenten sind, welche durch eine standardisierte Infrastruktur verwaltet werden und aus einem Service Repository abrufbar sind, um sie sinnvoll in Geschäftsprozessen zu kombinieren.<sup>73</sup>

Services sind nach dem Blackbox-Prinzip aufgebaut, d.h. sie kapseln bestimmte Funktionen und Daten. Der Aufruf von außen ist dabei nur über definierte und standardisierte Schnittstellen möglich. Dieses Verfahren ermöglicht, dass die Implementierung der einzelnen Services unterschiedlich (z.B. in verschiedenen Programmiersprachen) sein kann sowie dass die Services auch in verschiedenen Systemen oder sogar verschiedenen Unternehmen angesiedelt sein können.<sup>74</sup> Ein Geschäftsprozess wird durch eine Aneinanderreihung von Services abgebildet. Die Zusammenführung von mehreren Services zu einem Prozess wird auch als Orchestrierung bezeichnet. Dies ist möglich, weil die Services lose gekoppelt sind und die Kommunikation über standardisierte Schnittstellen ablaufen.<sup>75</sup>

---

<sup>72</sup> Vgl. Süddeutsche.de (2012), (siehe Internetverzeichnis).

<sup>73</sup> Vgl. Baumgartl, Axel / Mebus, Frank / Seemann, Volker (2010), S. 52ff.

<sup>74</sup> Vgl. Holley, Kerrie / Arsanjani, Ali (2011), Frage 12.

<sup>75</sup> Vgl. Salein, Marcel u.a. (2010), S. 35.

Die SAP AG hat sich diesem Konzept angenommen und eine eigene Implementierung des SOA-Gedanken, die SAP-Enterprise-Service, geschaffen. Das Unternehmen setzt bei der Umsetzung von SOA auf Webservice-Standards. Dabei handelt es sich um eine plattformunabhängige Schnittstellentechnologie, die auf vielen etablierten Technologien aufbaut:

- **Webservice Description Language (WSDL)** ist eine XML-basierte Sprache zur Schnittstellenbeschreibung eines Webservices.
- **Simple Object Access Protocol (SOAP)** ist eine XML-basiertes Protokoll zum Austausch der Daten zwischen einem Webservice-Anbieter und einem Webservice-Nutzer.
- **Universal Description, Discovery and Integration (UDDI)** beinhaltet einen Verzeichnisdienst zum Auffinden vorhandener Webservices, dies ist mit einer Art „Gelbe Seiten“ zu vergleichen.<sup>76</sup> SAP nennt diese Komponente Services Registry.

Durch die Einführung des SAP NetWeaver Applikationsservers im Jahr 2003, welcher die technologische Basis für die SAP Business Suite darstellt, wurden die Grundlagen einer Serviceorientierung in SAP Systemen gelegt. Ziel war der Wechsel von einer starren Systemlandschaft hin zu einer flexiblen, serviceorientierten Architektur. Die SAP Business Suite, welche das ERP, CRM und weitere Softwarepakete beinhaltet, fungiert als ein zentrales Backend-System dieser Architektur, da es viele Enterprise Services zur Verfügung stellt. Diese können durch andere Softwarekomponenten verwendet werden. Die Business Suite dient daher, wie alle anderen Systeme, welche Webservices bereitstellen, als Service Provider.<sup>77</sup> Zur Servicemodellierung und -implementierung kommt das Enterprise Services Repository zum Einsatz.<sup>78</sup>

Auf diesem Konzept baut der SAP NetWeaver CE auf, da dort entwickelte Softwarekomponenten dem SOA-Prinzip folgen. Anders als die zuvor genannten Backend-Systeme, treten Anwendungen auf Basis des NetWeaver CE vermehrt als Service Consumer auf, d.h. sie verwenden von anderen Systemen bereitgestellte Funktionalitäten. Applikationen, welche mehrheitlich aus Services bestehen, werden auch Composite Applications genannt.<sup>79</sup>

---

<sup>76</sup> Vgl. Richardson, Leonard / Ruby, Sam (2007), S. 355.

<sup>77</sup> Vgl. Baumgartl, Axel / Mebus, Frank / Seemann, Volker (2010), S. 67ff.

<sup>78</sup> Vgl. Ebenda, S. 81.

<sup>79</sup> Vgl. Ebenda, S. 54.

## 3.2 SAP NetWeaver Composition Environment

### 3.2.1 Einführung

Die Composition Environment (CE) ist seit dem Release 7.1 im Jahr 2007 Bestandteil des SAP NetWeaver Java-Stacks. Sie war zunächst ein eigenständiger Usage Type und war nicht in Kombination mit anderen Usage Types für den Produktivbetrieb freigegeben. Im Jahr 2011 wurde schließlich die CE-Komponente in den SAP NetWeaver 7.3 so integriert, dass ein paralleler Betrieb mit anderen Usage Types möglich ist. Der SAP NetWeaver ist die Integrations- und Technologieplattform von SAP und stellt die Basis für alle Anwendungen der Business Suite dar. Der SAP NetWeaver ist also als die Basis eines „gewöhnlichen“ SAP-System zu verstehen.<sup>80</sup> Der SAP NetWeaver CE ist hingegen eine Plattform für das Design, die Entwicklung und die Ausführung von Composite Applications, welche auf Java basieren und den SOA-Prinzipien von SAP entsprechen.<sup>81</sup>

Ziel des NetWeaver CE ist die Entwicklung von leichtgewichtigen, flexiblen und kollaborativen Anwendungen, welche den Unternehmen dabei helfen, ihre Geschäftsprozesse flexibel und effektiv zu unterstützen. Hierfür stellt der NetWeaver CE Werkzeuge bereit, welche zum Großteil auf modellgetriebene Entwicklungsansätze beruhen, um den hohen Anforderungen der zu erstellenden Software gerecht zu werden.<sup>82</sup> SAP verspricht in diesem Zusammenhang Vorteile in den bereits im Unterkapitel 2.1 „Schwierigkeiten in der traditionellen Softwareentwicklung“ benannten Aspekten.<sup>83, 84</sup> Auch der Fokus auf die Verwendung von offenen Standards besitzt für den NetWeaver CE einen hohen Stellenwert, so spielen u.a. XML und Webservices eine wichtige Rolle, Geschäftsprozesse werden in BPMN 2.0 modelliert, der Applikationsserver basiert auf Java und die Softwareentwicklung erfolgt in einem Derivat der Open Source IDE Eclipse.<sup>85</sup>

Im folgenden Abschnitt wird zunächst der Aufbau einer Composite Application näher betrachtet. Anschließend werden auf die Komponenten des NetWeaver CE eingegangen und erklärt, welchen Nutzen sie bei der Umsetzung der verschiedenen Schichten einer zusammengesetzten Anwendung verfolgen.

---

<sup>80</sup> Vgl. Salein, Marcel u.a. (2010), S. 35f.

<sup>81</sup> Vgl. Baumgartl, Axel / Mebus, Frank / Seemann, Volker (2010), S. 80.

<sup>82</sup> Vgl. Ebenda, S. 80.

<sup>83</sup> Vgl. Salein, Marcel u.a. (2010), S. 34f.

<sup>84</sup> Vgl. Rauscher, Jan / Stiehl, Volker (2007), S. 41f.

<sup>85</sup> Vgl. Ebenda, S. 38f.

### 3.2.2 Composite Applications

Wie bereits erwähnt, verfolgt der NetWeaver CE das Ziel, Composite Applications zu erstellen. Hierunter sind Anwendungen zu verstehen, welche möglichst effizient auf vorhandene Funktionalitäten aus verschiedenen Backend-Systemen zurückgreifen, um diese in neuen Geschäftsprozessen wiederzuverwenden. Somit ist die Übersetzung „zusammengesetzte Anwendungen“ recht passend. An Stellen, bei denen keine wiederverwendbaren Services verfügbar sind, kann unterdessen eigene Geschäftslogik entwickelt werden. Zusätzlich wird das Erstellen von einheitlichen Benutzeroberflächen unterstützt. Weiterhin erhalten Composite Applications eine eigene Architekturschicht für Geschäftsprozesse, damit keine Prozessabläufe im Quelltext codiert werden müssen. Das Ziel dahinter ist eine erhöhte Flexibilität der Geschäftsprozesse, so sollen beispielsweise mit einem grafischen Modellierungstool ohne großen Aufwand Prozesse angepasst oder Prozessschritte ausgetauscht werden, sofern diese alle fachlichen Abhängigkeiten erfüllen. Die Interaktion des Mitarbeiters im Geschäftsprozess wird bewusst berücksichtigt, so dass man von kollaborativen Geschäftsprozessen spricht. Der Vorteil einer solchen Anwendung liegt darin, dass die Anwender besser in den Geschäftsprozess integriert sind. Dies zeigt sich beispielsweise in der Verringerung von manuellen Tätigkeiten, wie dem Übertragen von Arbeitsergebnissen aus Hilfsmitteln wie Tabellenkalkulations-, Textverarbeitungsprogrammen oder E-Mail in das ERP-System. Neben der Verkürzung von Prozessdurchlaufzeiten wird davon ausgegangen, dass Daten aufgrund der direkten systemseitigen Erfassung in einer strukturierten Form vorliegen und somit die Datenqualität insgesamt positiv beeinflusst wird.<sup>86</sup>

Die Architektur einer Composite Application sieht nach SAP die Schichten „Prozessschicht“, „UI-Schicht“ und „Geschäftsobjekt-, Serviceschicht“ vor (Vgl. Abbildung 3-1: Schichten einer Composite Application). Demnach erfolgt eine Trennung nach Prozessablauf, Oberfläche und Datenzugriff. SAP sieht keine Backend-Schicht im klassischen Sinne vor, da der SOA-Gedanke davon ausgeht, dass bereits viele Funktionen in der IT-Landschaft geeignet zur Verfügung stehen. Um diese zu nutzen, werden die vorhandenen Services in einer einheitlichen Form als „entfernte Services“ gekapselt und in Form eines Geschäftsobjekts zusammengefasst. Die Definition von lokalen Services sowie Geschäftsobjektmodellen ermöglicht zusätzlich die Speicherung von Daten, welche in keinem Quellsystem vorhanden oder thematisch nicht passend sind. Diese Geschäftsobjekt-basierte Ser-

---

<sup>86</sup> Vgl. Rauscher, Jan / Stiehl, Volker (2007), S. 23ff.

viceschicht hat den Vorteil, dass die UI-Schicht eine einheitliche, Webservice-basierte Schnittstelle auf lokale und entfernte Daten vorweisen kann. Es ist daher für die Oberfläche und den Prozessablauf unwesentlich, ob zur Laufzeit mit lokalen oder entfernten Daten gearbeitet wird. Eine Änderung des Datenzugriffs kann somit durchgeführt werden, ohne dass andere Schichten der Anwendung davon betroffen sind. Dies könnte zum Beispiel der Wechsel des Geschäftsobjekts „Kunde“ aus dem ERP in ein eigenständiges CRM-System sein.<sup>87</sup>

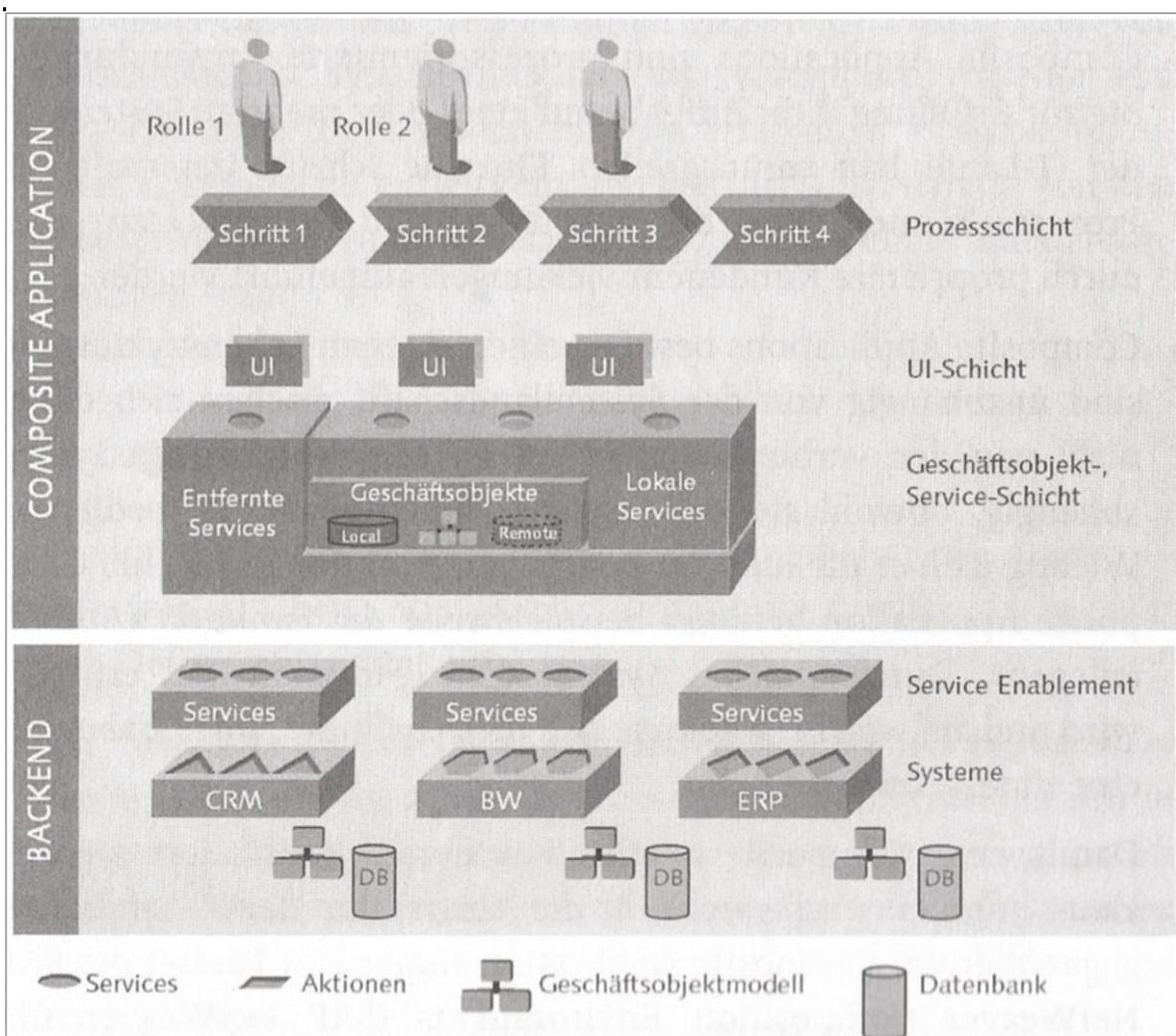


Abbildung 3-1: Schichten einer Composite Application

Quelle: Rauscher, Jan / Stiehl, Volker (2007), S. 28.

Die UI-Schicht sieht für jeden Prozessschritt mindestens eine Maske vor. Es handelt sich bei den Oberflächen in der Regel um einfache Ein- und Ausgabemasken sowie Tabellen, welche klare Eingabe- und Ausgabeparameter besitzen. So gibt

<sup>87</sup> Vgl. Rauscher, Jan / Stiehl, Volker (2007), S. 28ff.

es beispielsweise für einen fiktiven Vorgang „Bestätige Abwesenheit“ ein Formular, das beim Aufruf den Name des Mitarbeiters sowie den gewünschten Abwesenheitszeitraum erwartet. Als Ergebnis gibt die Oberflächenkomponente die Entscheidung des Vorgesetzten sowie einen Kommentar zurück.<sup>88</sup>

Für die Gestaltung des Geschäftsprozesses bedeutet dies, dass die Reihenfolge der Aktivitäten beliebig variieren kann, sofern der Datenfluss des Prozessschrittes bedient wird und keine fachlichen Abhängigkeiten zu anderen Schritten bestehen. Eine weitere wichtige Aufgabe dieser Schicht ist das Zuordnen von Rollen zu den einzelnen Prozessschritten, um zur Laufzeit bestimmen zu können, welcher Anwender ein Arbeitspaket bearbeiten soll.<sup>89</sup>

Der NetWeaver CE besitzt mehrere modellbasierte Methoden und Werkzeuge für die Erstellung und Wartung der unterschiedlichen Schichten einer Composite Application. Diese werden im folgenden Abschnitt erläutert.

### **3.2.3 Komponenten des SAP NetWeaver CE**

Die Abbildung 3-2: SAP NetWeaver Composition Environment fasst wesentliche Bestandteile des SAP NetWeaver CE zusammen. Die Basis des gesamten Systems ist ein Java EE 5-kompatibler und zertifizierter Applikationsserver. SAP war als einer der ersten Hersteller mit einem Java EE5-kompatiblen Server am Markt und unterstreicht damit das Bestreben zu offenen Standards. Der Applikationsserver unterstützt gängige Standards wie EJB 3.0, JSF 1.2, JPS 2.1, JMS, 1.1, JPA, JAX-WS 2.0 und SOAP 1.2.<sup>90</sup>

---

<sup>88</sup> Vgl. Rauscher, Jan / Stiehl, Volker (2007), S. 32.

<sup>89</sup> Vgl. Ebenda, S. 32ff.

<sup>90</sup> Vgl. Ebenda, S. 38.

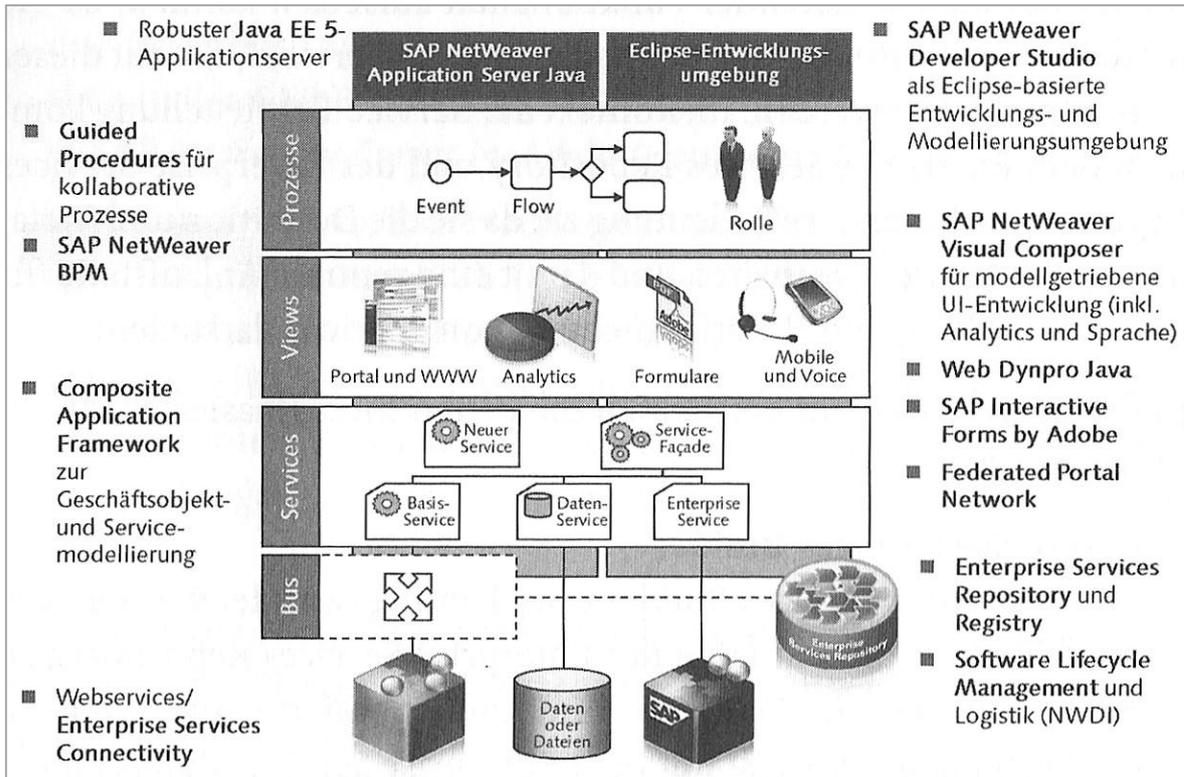


Abbildung 3-2: SAP NetWeaver Composition Environment

Quelle: Salein, Marcel u.a. (2010), S. 49.

Ein weiterer wichtiger Bestandteil des SAP NetWeaver CE ist die Entwicklungs- und Modellierungsumgebung SAP NetWeaver Developer Studio (NWDS). Das NWDS basiert in der Version 7.3 auf dem Eclipse 3.5-Framework (Galileo) und liefert eine Vielzahl proprietärer Plug-Ins, welche bei der Entwicklung von SAP spezifischer Software verwendet werden. Das SAP NetWeaver Developer Studio ist wiederum eng mit der SAP NetWeaver Development Infrastructure (NWDI) verknüpft, die für das Management des Software-Lebenszyklus verantwortlich ist. Verglichen mit der klassischen SAP Entwicklung auf Basis der Programmiersprache ABAP, übernimmt die NWDI die Aufgaben der Versionsverwaltung und des Transports.<sup>91, 92</sup>

Im vorherigen Abschnitt wurde herausgestellt, dass eine Composite Application aus mehreren Schichten besteht. Die Composition Environment stellt für jeden Teil dieses Schichtenmodells verschiedene Technologielösungen bereit. In der Serviceschicht ist das Composite Application Framework angesiedelt, welche bei der SOA-konformen Umsetzung von Geschäftsobjektmodellen verwendet wird. Hiermit ist die Erstellung von Geschäftslogik, die Nutzung vorhandener Dienste und

<sup>91</sup> Vgl. Rauscher, Jan / Stiehl, Volker (2007), S. 38.

<sup>92</sup> Vgl. Salein, Marcel u.a. (2010), S. 127.

die Erstellung neuer Services gemeint, welche in den weiteren Schichten einer Composition Environment zum Einsatz kommen.<sup>93</sup>

In der Schicht der Benutzeroberflächen (auch UI-Schicht bzw. View-Schicht genannt) ist das Technologieportfolio des SAP Netweaver CE breiter aufgestellt. In Abhängigkeit der Anwendungsfälle können der Visual Composer, Web Dynpro Java, Web Dynpro ABAP, Interactive Forms by Adobe oder Federated Portal Network die Lösung der Aufgabenstellung sein. Diese Komponenten sind wie folgt charakterisiert:

- **Visual Composer:** Dieses Werkzeug ermöglicht die Erstellung von weniger komplexen Weboberflächen durch eine intuitive, grafische Modellierungsoberfläche. Da es sich hierbei um ein modellgetriebenes Entwicklungswerkzeug handelt, wird der Visual Composer im Abschnitt 4.3 eingehend betrachtet.
- **Web Dynpro Java:** Diese Technologie ist die erste Wahl für komplexe Browserbasierte Oberflächen. Hier kann auf eine Vielzahl vorhandener Funktionen zurückgegriffen werden, welche insbesondere für das Geschäftsumfeld konzipiert sind. Als Beispiele können Wertehilfen, Internationalisierung, Scrollen in Masendaten und Zugriff für behinderte Menschen aufgeführt werden. Diese Technologie beinhaltet auch grafische Modellierungswerkzeuge, so dass Web Dynpro Java im Abschnitt 4.4 gesondert betrachtet wird.<sup>94</sup>
- **Web Dynpro ABAP:** Ab der Version 7.3 können auch Web Dynpro Anwendungen, welche in einem ABAP-System entwickelt sind, in Geschäftsprozessen verwendet werden. Die Integration der Anwendung erfolgt über so genannte WDA CHIPs (Collaborative Human Interface Part).
- **Interactive Forms by Adobe:** Bei den beiden zuvor genannten UI-Technologien ist nur eine Online-Übertragung der Benutzerinteraktion an die wartende Anwendung auf dem Applikationsserver möglich. Da dies jedoch nicht immer durchführbar ist, bieten interaktive Formulare basierend auf Adobe-Technologie eine Alternative. So kann sich ein Service-Techniker beispielsweise zu Beginn seines Arbeitstages seinen Arbeitsvorrat lokal auf sein Notebook laden. Dies erfolgt in Form von PDF-Dokumenten, welche der Techniker später direkt beim Kunden am Notebook ausfüllen und lokal speichern kann. Der Datentransfer zum Backend erfolgt später durch das Hochladen der PDF-Dateien.

---

<sup>93</sup> Vgl. Rauscher, Jan / Stiehl, Volker (2007), S. 38.

<sup>94</sup> Vgl. Ebenda, S. 39ff.

Diese Technologie stößt immer dort auf großes Interesse, wo bereits existierende Papierformulare ersetzt werden sollen.<sup>95</sup>

- **Federated Portal Network:** Der SAP Netweaver CE beinhaltet eine Portalkomponente, die die Erstellung und Einbettung von iViews ermöglicht. Dies sind Portalanwendungen, die in einem begrenzten Anzeigebereich eines Portals eingebettet sind. Da jedoch das Netweaver CE-Portal nicht das führende Portalsystem des Unternehmens sein muss, ermöglicht das Federated Portal Network ein Einbinden des Contents in SAP Portale, welche auf dem SAP NetWeaver ab Version 7.0 (2004s) basieren. Diese Technologie hat für die Bereitstellung von Composite Applications keine große Bedeutung und beinhaltet keine Ansätze für die modellgetriebene Entwicklung und wird aus diesem Grund nicht näher betrachtet.<sup>96</sup>

In der Prozessschicht wird die Verbindung zwischen Oberflächen und Service-Aufrufen hergestellt. Hierfür sind zwei Technologien entscheidend. Zum einen ist die Prozessmodellierung mit Guided Procedures möglich. Eine Guided Procedure überwacht den Prozessablauf, informiert Prozessteilnehmer und gibt Administratoren die Möglichkeit Prozesse zu überwachen und gegebenenfalls auf sie Einfluss zu nehmen.<sup>97</sup> Die Guided Procedures waren bis zum Release 7.2 des NetWeaver CE das bevorzugte Werkzeug zur Prozessmodellierung. Diese Technologie wird jedoch nicht weiterentwickelt und stattdessen strategisch auf Business Process Management (BPM) gesetzt.<sup>98</sup> Unterstrichen wird diese Aussage durch die Tatsache, dass Guided Procedures keine Bestandteile der SAP-Schulung zum NetWeaver CE 7.3 sind.<sup>99</sup> Der nun präferierte Ansatz mittels BPM setzt auf eine Prozessmodellierung nach BPMN. Die Technologie und das hierfür verwendete grafische Modellierungswerkzeug werden im Abschnitt 4.5 eingehend betrachtet.

---

<sup>95</sup> Vgl. Salein, Marcel u.a. (2010), S. 51.

<sup>96</sup> Vgl. Rauscher, Jan / Stiehl, Volker (2007), S. 39ff.

<sup>97</sup> Vgl. Ebenda, S. 41.

<sup>98</sup> Vgl. Baumgartl, Axel / Mebus, Frank / Seemann, Volker (2010), S. 201.

<sup>99</sup> Vgl. SAP AG (2012c): OKP SAP NetWeaver Composition Environment 7.3 (siehe Internetverzeichnis).

## 4 Werkzeuge und Technologien zur modellgetriebenen Entwicklung

### 4.1 NetWeaver Development Studio

Das NetWeaver Developer Studio (NWDS) ist die Entwicklungs- und Modellierungsumgebung von SAP für alle Java-basierten Softwareprojekte. Neben den üblichen Funktionen der integrierten Entwicklungsumgebung Eclipse erweitert SAP den Funktionsumfang für die Entwicklung von SAP-Technologien durch proprietäre Plug-Ins.<sup>100</sup> Somit unterstützt die Entwicklungsumgebung eine Vielzahl unterschiedlicher Entwicklungsprojekte. Neben gewöhnlichen Java-Projekten sollen insbesondere Projekte in SAP-Technologien mit dem Werkzeug entwickelt werden. Dies beinhaltet beispielsweise das Erstellen aller Komponenten einer Composite Application.<sup>101</sup>

Das Developer Studio koordiniert zusammen mit der NetWeaver Development Infrastructure den Entwicklungsprozess durch die Zuordnung der Entwicklungsobjekte zu Produkten, Software Components und Development Components. Ein Produkt besteht aus einem oder mehreren Software Components (SC), welche wiederum aus einem oder mehreren Development Components (DC) besteht. Eine DC wäre beispielsweise eine WebDynpro-Anwendung. Alle thematisch zusammengehörigen Development Components werden in einer Software Component zusammengefasst. Diese Zuordnung wird für die Anbindung an das System Landscape Directory (SLD), die Entwicklungskoordination für teamorientierte Entwicklung und dem Transport von Software benötigt.<sup>102</sup>

Die Benutzeroberfläche des Developer Studios besteht grundlegend aus Perspektiven, Views und Editoren. Alle visuellen Komponenten, welche zu einem Zeitpunkt sichtbar sind, werden als Window bezeichnet. Es gibt bereits vorgegebene Windows, welche Komponenten für die Entwicklung bestimmter Aspekte zusammenfassen. Diese Anordnungen werden Perspektiven genannt. Exemplarisch zeigt die folgende Abbildung ein Window mit einem Ausschnitt aus der Prozessentwicklungs-Perspektive.<sup>103</sup>

---

<sup>100</sup> Vgl. Kessler, Karl u.a. (2007), S. 55f.

<sup>101</sup> Vgl. Salein, Marcel u.a. (2010), S. 72f.

<sup>102</sup> Vgl. Kessler, Karl u.a. (2007), S. 476ff.

<sup>103</sup> Vgl. Ebenda, S. 56f.

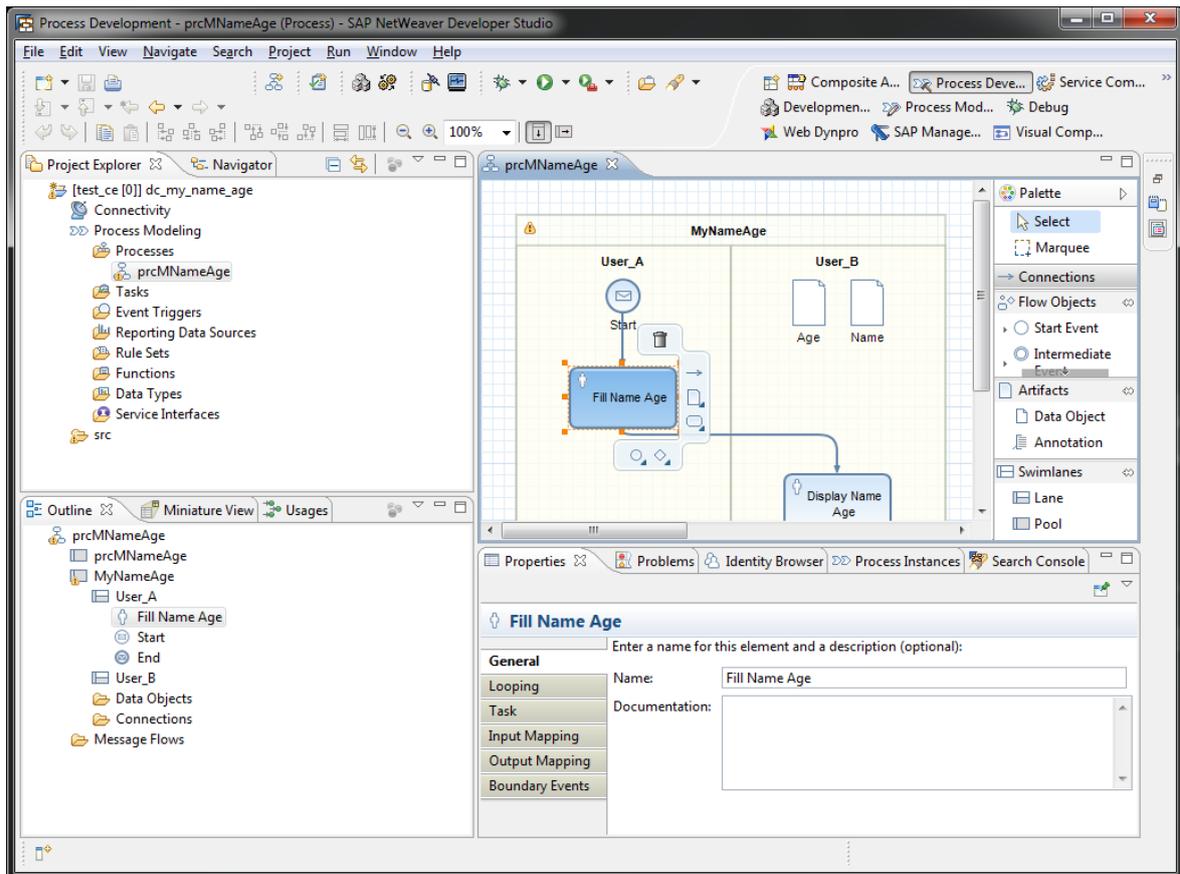


Abbildung 4-1: Benutzeroberfläche des NetWeaver Developer Studios

Quelle: Eigene Darstellung

## 4.2 Composite Application Framework

### 4.2.1 Einsatzgebiet und Konzept

Für die effiziente Erstellung von Composite Applications stellt SAP das Composite Application Framework bereit (CAF). Die SOA-basierte Entwicklungsumgebung erlaubt die Erstellung von Anwendungen ohne sich mit der low-level API eines technischen Frameworks befassen zu müssen.<sup>104</sup> Der Entwickler muss lediglich die Anwendungskomponenten modellieren, diese werden als Metadattendateien abgespeichert und anschließend durch das Composite Application Framework in Java-Source-Code transformiert.<sup>105</sup>

Das Framework ermöglicht insbesondere die Erstellung von Services mit Hilfe von CAF Core sowie die Service-Orchestrierung mit CAF Guided Procedures.<sup>106</sup> Letztes hat jedoch in der Version 7.3 des NetWeaver CE an Bedeutung verloren, da

<sup>104</sup> Vgl. SAP Help – NetWeaver CE 7.3 (2012), (siehe Internetverzeichnis).

<sup>105</sup> Vgl. Rauscher, Jan / Stiehl, Volker (2007), S. 45.

<sup>106</sup> Vgl. SAP Help – NetWeaver CE 7.0EHP2 (2010), (siehe Internetverzeichnis).

BPM der durch SAP präferierte Ansatz für Prozessmodellierung ist (vgl. 3.2.3 Komponenten des SAP NetWeaver CE). Daher wird in diesem Unterkapitel der Fokus auf die Bereitstellung und Konsumierung von Services mit Hilfe von CAF Core gelegt.

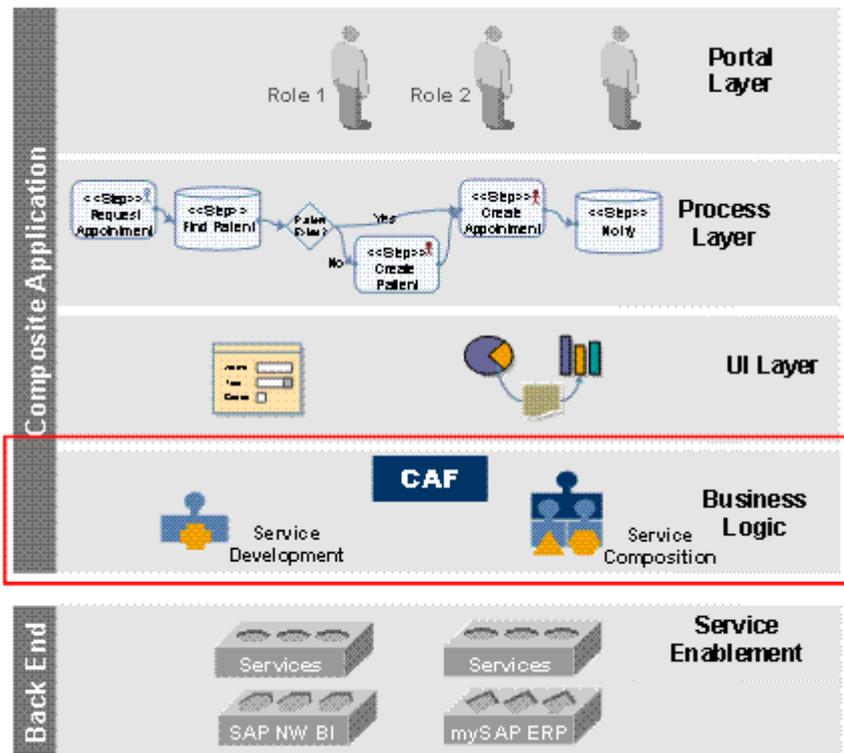


Abbildung 4-2: Einordnung der Service Schicht in die Gesamtarchitektur  
Quelle: SAP Help – NetWeaver CE 7.3 (2012), (siehe Internetverzeichnis).

Die Serviceschicht stellt die Basis einer jeden Composite Application dar, da sie die Businesslogik beinhaltet und eine einheitliche Verbindung zum Datenmodell bereitstellt. Realisiert wird diese Abstraktionsebene über verschiedene Arten von Services. Besonders relevant für Composite Applications sind die so genannten Geschäftsobjekte (Business Objects). Die Daten werden hierbei entweder lokal in des NetWeaver CE gespeichert oder durch entfernte Systeme bereitgestellt. Neben Attributen, welche das Datenmodell repräsentieren, sieht das Konzept der Business Objects auch Operationen vor, mit denen von außen auf das Objekt zugegriffen wird. Der Zugriff selbst erfolgt über eine Webservice-Schnittstelle.<sup>107</sup>

Die Erstellung einer Anwendung auf Basis des Composite Application Frameworks erfolgt vollständig innerhalb des SAP NetWeaver Developer Studios. Dort sind

<sup>107</sup> Vgl. SAP Help – NetWeaver CE 7.3 (2012), (siehe Internetverzeichnis).

Modellierungswerkzeuge für die Erzeugung von Business Objects, Application Services und für den Import von externen Services verfügbar. Die folgende Tabelle stellt die drei Servicearten anhand möglicher Eigenschaften gegenüber.

<b>Business Object (Entity Service)</b>	<b>Application Service</b>	<b>External Service</b>
Struktur und Assoziationen	Abhängigkeiten	Operationen
Operationen	Operationen	Konfiguration
Persistenz	Implementierung	
Datenquellen		
Berechtigungen		
Implementierung		

Tabelle 4-1: Eigenschaften von CAF-Objekten

Quelle: Eigene Darstellung in Anlehnung an SAP Help - NetWeaver CE 7.0EHP2 (2010b), (siehe Internetverzeichnis).

Je nach Verwendungszweck gilt es, die richtige Serviceart auszuwählen. Externe Services kommen beispielsweise nur dann in Frage, wenn die benötigte Funktionalität bereits durch ein anderes System in Form eines Services oder RFC-fähigen Funktionsbausteins bereitgestellt wird. Diese können direkt importiert werden und für eine Composite Application genutzt werden. Application Services eignen sich für Anwendungslogik, bei der Zugriff auf Fremdsysteme und persistente Datenhaltung keine Rolle spielt. Der größte Funktionsumfang ist bei den Business Objects anzutreffen. Hier können zusätzlich Berechtigungen festgelegt und die Datenhaltung konfiguriert werden.

Die CAF Umgebung stellt neben den Designwerkzeugen auch verschiedene Laufzeitwerkzeuge bereit. Die erstellten Services werden auf den Java-Applikationsserver deployed. Sie nutzen die Ablaufumgebung für Enterprise-Java-Beans (EJB-Container) sowie die integrierten Persistenz- und Sicherheitsmechanismen.<sup>108</sup>

#### 4.2.2 Modellierungswerkzeug

Die Modellierung eines Business Objects erfolgt im NetWeaver Developer Studio in der Perspektive „Composite Application“. In diesem Abschnitt wird sich auf die Erstellung eines Geschäftsobjekts konzentriert, da diese Servicevariante die komplexeste der verschiedenen Servicearten darstellt und auch für Composite Appli-

<sup>108</sup> Vgl. SAP Help – NetWeaver CE 7.3 (2012), (siehe Internetverzeichnis).

cations die größte Bedeutung aufweist. Das Entwicklungswerkzeug stellt mehrere Reiter für die Modellierung der verschiedenen Aspekte eines Services bereit, welche im Folgenden hinsichtlich ihres Funktionsumfangs betrachtet werden.

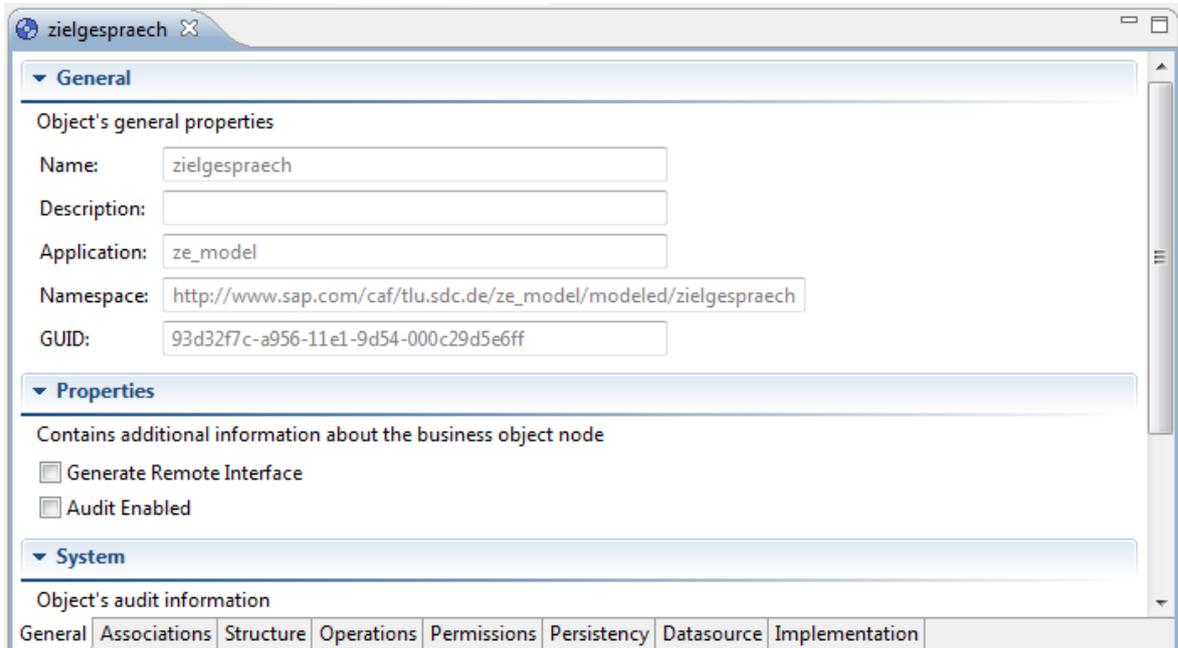


Abbildung 4-3: Modellierung eines Geschäftsobjekts im NWDS

Quelle: Eigene Darstellung.

### *Struktur und Assoziationen*

Attribute eines Geschäftsobjekts können entweder in Form von primitiven oder komplexen Datentypen vorliegen. Hier gilt jedoch die Einschränkung, dass nur 0...1 und 1...1 Beziehungen durch das Modellierungswerkzeug zugelassen werden. Um 0...n und 1...n Beziehungen zu realisieren, ist die Verwendung von Assoziationen notwendig. In diesem Fall wird für das mehrfach auftretende Element ein eigenes Geschäftsobjekt angelegt. Dieses kann dem übergeordneten Geschäftsobjekt anschließend in der gewünschten Kardinalität zugeordnet werden.

### *Operationen*

Die Erstellung der grundsätzlichen Zugriffsoperationen erfolgt durch das Modellierungswerkzeug. Die CRUD-Methoden (create/createMultiple, read, update/updateMultiple und delete/deleteMultiple) werden anhand der Attribute des Business Objects automatisch generiert und können durch Entwickler nicht verändert werden. Dies gilt auch für die beiden Suchmethoden „findAll“ und „findByMultipleParameters“. Das Anlegen neuer Suchoperationen ist jedoch uneingeschränkt möglich. Hierfür ist das Auswählen der Attribute, welche für die gewünschte Selektion

relevant sind, ausreichend. Die Auswahl von Attributen assoziierter Geschäftsobjekte wird ebenfalls unterstützt, was auch komplexe Suchabfragen ermöglicht.<sup>109</sup> Bei Anwendungsservices kann die Schnittstelle einer Operation modelliert und anschließend individuell in Form von Java-Coding implementiert werden.

### *Persistenz*

Für die Speicherung des Geschäftsobjekts in einer Datenbank sind keine Entwicklungsarbeiten innerhalb der Datenbank notwendig, da alle Informationen für die lokale Persistenz durch das formale Modell des Business Objects beschrieben werden. Die modellierten Business Objects werden zur Laufzeit durch Java Enterprise Beans repräsentiert. Das bedeutet, dass der generierte Quellcode entsprechen objektorientierten Programmierparadigmen erzeugt wird. Da das Geschäftsobjekt zur Laufzeit ein Java-Objekt ist und die Speicherung in relationalen Datenbanktabellen erfolgt, kommt bei der Persistenz die Java Persistence API (JPA) zum Einsatz.

Auf dem Persistenz-Reiter kann der Entwickler die Abbildung der modellierten Attribute auf Datenbanktabellen einsehen. So ist beispielsweise ersichtlich, dass alle Einzelattribute in einer Datenbanktabelle gehalten werden, inkludierte Strukturen werden hingegen in einer separaten Datenbanktabelle gespeichert. Weiterhin werden Tabellen zur Abbildung der Beziehungen zwischen Geschäftsobjekten (Assoziationen) an dieser Stelle aufgeführt.

### *Datenquellen*

Auf diesem Reiter findet man alle Steuerungsparameter für entfernte Datenquellen. Dieser Reiter kommt daher immer dann zum Einsatz, wenn man für die Datenhaltung des Geschäftsobjekts nicht die lokale Persistenz des NetWeaver CE verwendet. Es kann an dieser Stelle ein Mapping zwischen der Struktur des Fremdsystems und der des Geschäftsobjekts definiert werden.

### *Berechtigungen*

Die Verwendung von Geschäftsobjekten kann durch Berechtigungen gesteuert werden. Als Standardeinstellung sind die Berechtigungsprüfungen aktiviert. Die Berechtigungszuordnung erfolgt im NetWeaver Administrator mithilfe des Composition Application Framework Authorization Tools. An dieser Stelle kann hinterlegt werden, welche User, Gruppen bzw. Rollen Zugriff auf das Geschäftsobjekt haben

---

<sup>109</sup> Vgl. SAP Help – NetWeaver CE 7.3 (2012), (siehe Internetverzeichnis).

sollen. Weiterhin kann der Zugriff von Attributwerten der Instanz abhängig gemacht werden, wenn z.B. ein Sachbearbeiter nur eine ihm zugeteilte Kundengruppe bearbeiten darf.

### Implementierung

Auf diesem Reiter werden die für diesen Service generierten Artefakte aufgeführt. Es können Änderungen bzw. Erweiterungen in der EJB Implementierungsklasse (Namenskonvention: \*BeanImpl.java) vorgenommen werden. Diese werden nicht beim erneuten Generieren überschrieben, da das Ergebnis der Modelltransformation in eine abstrakte Klasse einfließt, von der die Implementierungsklasse erbt.

### 4.2.3 Modelltransformation

SAP verwendet für die Erstellung des lauffähigen Quellcodes Generatoren und greift im Hintergrund auf Templates zurück. Es besteht jedoch keine explizite Möglichkeit für SAP Kunden, Änderungen an diesen Templates und Generatoren vorzunehmen. Somit ist zwar seitens SAP mit einem flexiblen MDS-Ansatz gearbeitet worden, jedoch wurde diese Flexibilität nicht an die Kunden weitergegeben. Somit kann das Werkzeug nur in dem von SAP angedachten Rahmen verwendet werden.<sup>110</sup>

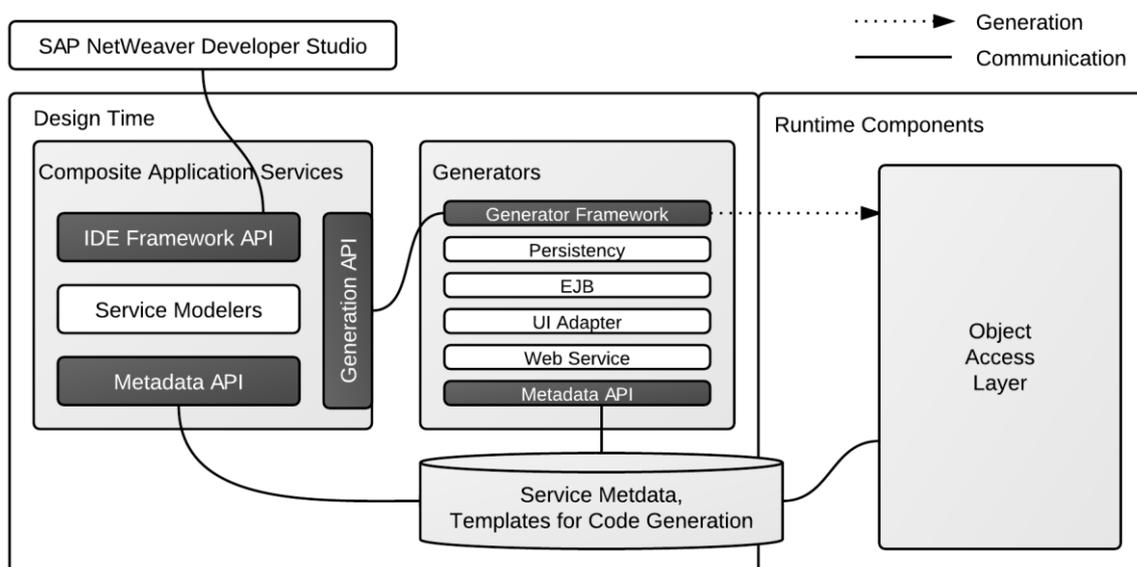


Abbildung 4-4: Modelltransformation CAF

Quelle: Eigene Darstellung in Anlehnung an SAP Help - NetWeaver CE 7.0EHP2 (2010b), (siehe Internetverzeichnis).

<sup>110</sup> Vgl. SAP Help - NetWeaver CE 7.0EHP2 (2010b), (siehe Internetverzeichnis).

Als Ergebnis der Modelltransformation werden mehrere Java-Klassen erzeugt. Darunter sind Interface-Klassen, welche interne und externe Schnittstellen definieren. Diese werden durch eine abstrakte Klasse implementiert.

Manuelle Änderungen am generierten Quellcode sind trotz alledem möglich und werden nicht durch erneutes Generieren überschrieben. Durch die Verwendung einer abstrakten Klasse für die generierten Bestandteile und einer Implementierungsklasse für die manuellen Änderungen ist es möglich, den Funktionsumfang des Geschäftsobjekts zu erweitern bzw. einzelne Funktionen durch eine Redefinition der Methode zu beeinflussen. Auf diese Möglichkeit sollte jedoch weitestgehend verzichtet werden, da das Modell nichts von diesen Änderungen bemerkt und somit nicht mehr den tatsächlichen Gegebenheiten der Anwendung entspricht. Die Implementierung von Quellcode ist nur bei Operationen von Anwendungsservices explizit gefordert. Diese Änderungen werden vom Modell erwartet und korrekt behandelt, so dass keine Inkonsistenzen zwischen Modell und Implementierung entstehen.

Bei der Verwendung von externen Services reicht eine WSDL-Schnittstellenbeschreibung als Metabeschreibung für das Generieren von Proxyobjekten. Abschließend ist nur die Konfiguration der Destination des Services im Administrationsbereich notwendig. Somit ist auch an dieser Stelle keine manuelle Entwicklungsarbeit notwendig.

#### **4.2.4 Bewertung**

Es können durch das Composite Application Framework Geschäftsobjekte inklusive Persistenz und Zugriffsmethoden ohne eine Zeile Programmierung erstellt und grundsätzlich via Webservice darauf zugegriffen werden. Somit ist die Service-schicht flexibel und eignet sich nicht nur für die Verwendung innerhalb einer Composite Application nach dem Designentwurf von SAP.

Ein Anwendungsservice kann genutzt werden um den Zugriff auf Geschäftsobjekte noch einmal zu kapseln. So ist es beispielsweise über die generierte Webservice-Schnittstelle des Geschäftsobjekts nicht möglich, Assoziationen zu anderen Geschäftsobjekten zu verwalten. Dieser Funktionsumfang kann durch einen Anwendungsservice bereitgestellt werden, allerdings sind dafür Java-Programmierkenntnisse notwendig. Durch Anwendungsservices lassen sich auch Extraktionsoperationen für SAP BW entwickeln, um Daten aus den Geschäftsobjekten für Auswertungen im Business Warehouse nutzen zu können.

Das Konsumieren von Services anderer Systeme oder RFC-fähiger Bausteine erfolgt ohne Programmieraufwand. Fremde Services, die auf diese Weise gekapselt sind, können in Composite Applications identisch wie lokale Services verwendet werden. Somit kann in den übergeordneten Schichten stets gegen eine einheitliche Schnittstelle entwickelt werden.

Das Modellierungswerkzeug innerhalb des NetWeaver Developer Studios lässt sich nach kurzer Einarbeitungszeit gut bedienen. Änderungen werden schnell übernommen und somit ist das Modell stets konsistent. Viele Fehlerkonstellationen werden bereits zur Designzeit abgefangen. Negativ ist jedoch aufgefallen, dass bestimmte Änderungen am Datenmodell nicht ohne weiteres nachträglich möglich sind. Dies sind beispielsweise das Verkürzen von Tabellenfeldern oder das Löschen von Spalten.<sup>111</sup> Dies ist zwar sinnvoll, um keinen Datenverlust zu verursachen, jedoch sind solche Änderungen in manchen Fällen gewollt. Gerade zu Beginn von Entwicklungsarbeiten unterliegt das Datenmodell häufigen Änderungen. Dieser Fehlerkonstellation wird erst während des Deployments bemerkt. Ein erfolgreiches Deployment der Development Component ist erst wieder möglich, nachdem die nicht durchführbaren Datenbankänderungen manuell zurückgesetzt wurden. Durch das vollständige Entfernen der Development Component vom Applikationsserver (Undeployment) können die Datenbankänderungen umgesetzt werden, jedoch ist ein vollständiger Datenverlust hinzunehmen.

Im Vergleich mit klassischen SAP-Werkzeugen zur Realisierung eines vergleichbaren Funktionsumfangs ist der Zeitaufwand mit dem Composite Application Framework geringer und die technische Umsetzung stets einheitlich und von konstanter Qualität.

### **4.3 Visual Composer**

#### **4.3.1 Einsatzgebiet und Konzept**

Der Einstieg in die Betrachtung des Visual Composers soll durch einen Vergleich mit Web Dynpro Java begonnen werden. Beide Technologien zielen auf die gleiche technische Domäne, der UI-Erstellung für Geschäftsanwendungen, ab. Es soll hierbei abgegrenzt werden, wo die Einsatzszenarien und individuellen Stärken beider Technologien liegen.

---

<sup>111</sup> Vgl. SAP Hinweis 1383461.

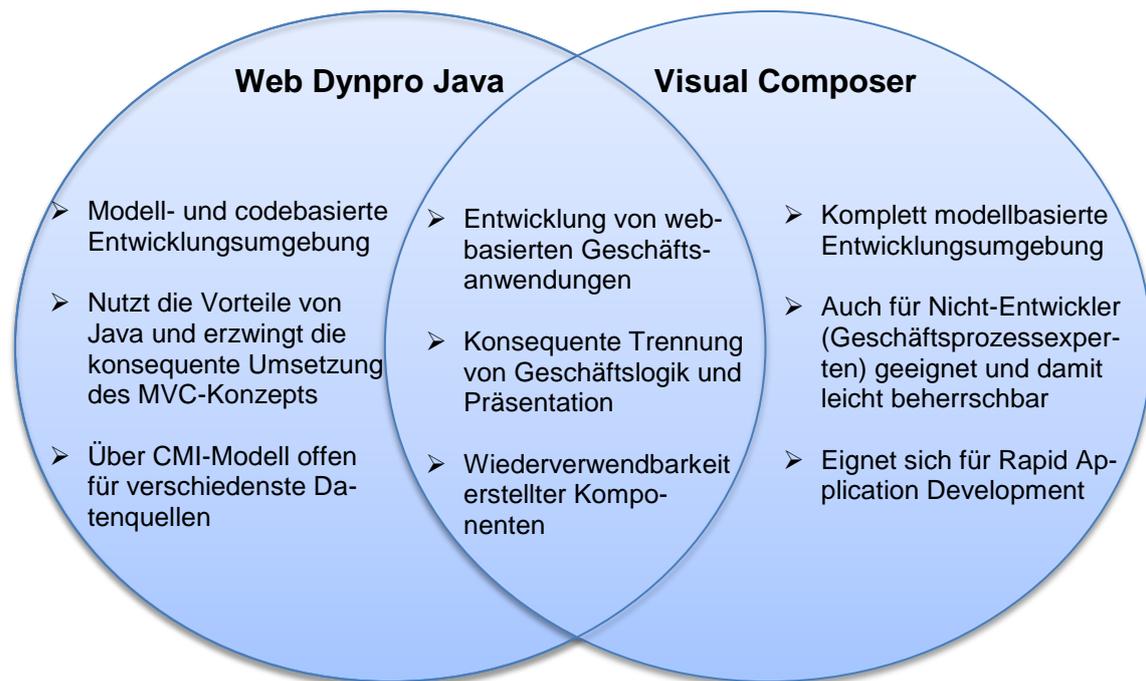


Abbildung 4-5: Eigenschaften von Web Dynpro und Visual Composer

Quelle: Eigene Darstellung in Anlehnung an Salein, Marcel u.a. (2010), S. 54.

Neben der Gemeinsamkeit, dass beide Technologien die Erstellung von Webanwendungen unterstützen, wird besonderen Wert auf eine wiederverwendbare Komponentenstruktur gelegt. Die Gegenüberstellung zeigt, dass der Visual Composer seine Stärken in der Einfachheit der Bedienung hat, da zum Erstellen einer Anwendung keine Programmierkenntnisse vorhanden sein müssen. Dieser Vorteil wird jedoch durch einen vergleichsweise kleinen Funktionsumfang erkauft, so dass bereits vor dem Realisierungsbeginn analysiert werden muss, ob die Anforderungen überhaupt durch den Visual Composer abgebildet werden können.<sup>112</sup>

Der Visual Composer wurde in den vergangenen beiden Releases des NetWeaver CE als führende Oberflächentechnologie für Composite Applications vermarktet.<sup>113</sup> In der aktuellen SAP-Roadmap für UI-Technologien, die auch für den NetWeaver CE 7.3 maßgebend ist, wird der Visual Composer jedoch nicht mehr als strategische UI-Technologie erwähnt.<sup>114</sup> Dies bedeutet konkret, dass der Visual Composer zukünftig zwar unterstützt, jedoch nicht weiterentwickelt wird. Somit ist keine oder nur geringfügige Erweiterung des Funktionsumfangs an kommende Gegebenheiten zu erwarten. Fehlerkorrekturen werden jedoch bis mindestens 2020 geliefert.<sup>115</sup>

<sup>112</sup> Vgl. Baumgartl, Axel / Mebus, Frank / Seemann, Volker (2010), S. 83.

<sup>113</sup> Vgl. Salein, Marcel u.a. (2010), S. 69f.

<sup>114</sup> Vgl. SAP AG (2012), Folie 14.

<sup>115</sup> Vgl. SAP AG (2011b), (siehe Internetverzeichnis).

Seitens SAP wird empfohlen stattdessen Web Dynpro Java zu verwenden. Die folgende Grafik vergleicht hierfür aktuelle und zukünftige Java-basierte UI-Technologien.

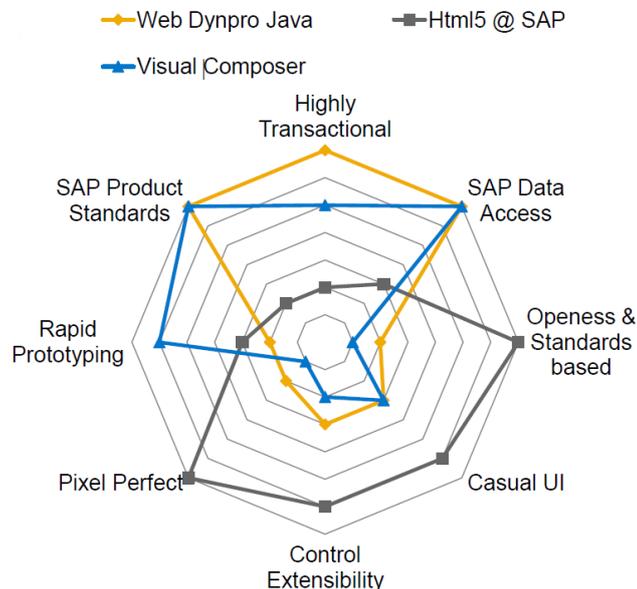


Abbildung 4-6: Strategische Ausrichtung der Java-UI-Technologien

Quelle: SAP AG (2012), Folie 64.

Entsprechend dieser Einschätzung der SAP ist der Visual Composer nur beim Rapid Prototyping die erste Wahl. Trotz alledem soll die Technologie beschrieben werden, weil es sich hierbei um einen gänzlich modellgetriebenen Ansatz handelt und die Verfügbarkeit für SAP-Kunden für die nächsten Jahre gesichert ist. Im direkten Vergleich zu Web Dynpro Java zeigt die Abbildung 4-6, dass der Funktionsumfang in den einzelnen Kategorien identisch oder nur geringfügig unterlegen ist. Es ist also denkbar, Prototypen und Oberflächen mit geringen Anforderungen in Visual Composer zu entwickeln.

#### 4.3.2 Modellierungswerkzeug

Die Entwicklungsumgebung des Visual Composers kann prinzipiell über einen Webbrowser oder über das SAP NetWeaver Developer Studio aufgerufen werden. Der Funktionsumfang unterscheidet sich zwischen den beiden Clients nicht. Im Rahmen dieser Masterarbeit wird auf die integrierte Variante des NWDS fokussiert, da das Developer Studio für andere Komponenten einer Composite Application unabkömmlich ist und somit vollständig innerhalb eines Entwicklungswerkzeuges gearbeitet werden kann.

Ein wesentlicher Anspruch des Werkzeuges ist die einfache Bedienung und die intuitive Modellierung der Ablauflogik sowie der Benutzeroberfläche mithilfe grafischer Notationen. Dabei soll die Visualisierung der Ablauflogik möglichst einfach dargestellt werden, die Benutzeroberflächen sollen hingegen möglichst präzise die fertige Software repräsentieren.<sup>116</sup>

Die Komponenten des Visual Composers lassen sich in server- und clientseitige Bestandteile sowie in Design- und Laufzeitkomponenten unterteilen. Die nachfolgende Abbildung gibt einen Überblick über die Architektur des Visual Composers. Anschließend werden wichtige Komponenten benannt und erklärt.

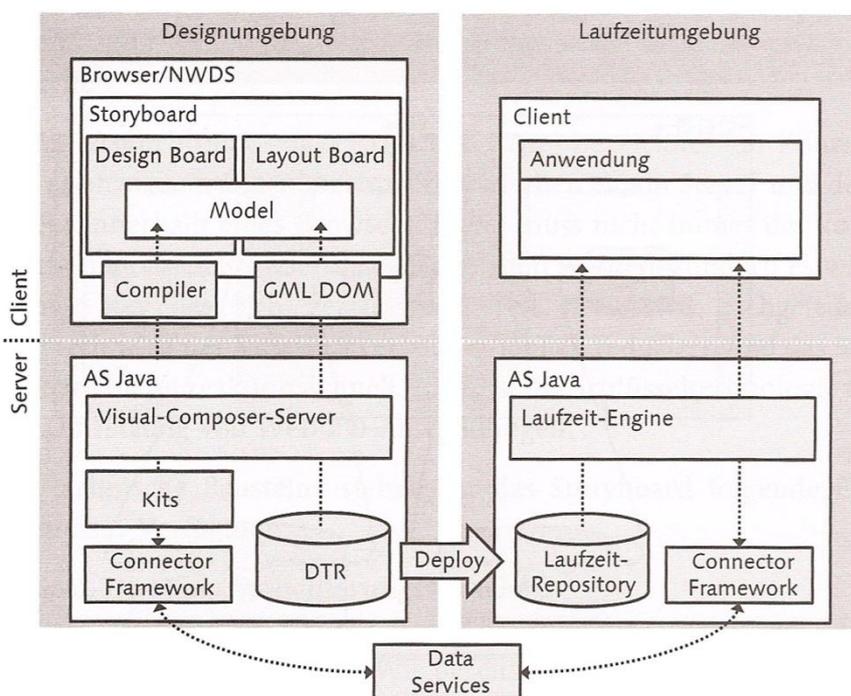


Abbildung 4-7: Architektur des Visual Composer

Quelle: Salein, Marcel u.a. (2010), S. 113.

Für die Erstellung eines Visual Composer Modells ist insbesondere das Storyboard relevant. Es umfasst alle Werkzeuge, welche für die Modellierung notwendig sind. Das Storyboard stellt hierfür zwei Sichten bereit, zum einem das Design Board für die Ablauflogik und zum anderen das Layout Board für die Darstellung der Oberflächen. Das Storyboard bedient sich einer vollständig grafischen Notation für die Beschreibung der Elemente.<sup>117</sup>

<sup>116</sup> Vgl. Salein, Marcel u.a. (2010), S. 87.

<sup>117</sup> Vgl. Ebenda, S. 112ff.

Weiterhin sind während der Erstellungsphase mehrere Serverkomponenten involviert. Diese werden für die zentrale Ablage der Modelle und der Übertragung in die entsprechenden Laufzeitumgebungen benötigt. Die Laufzeitumgebungen können sich je nach gewählter Zielplattform unterscheiden. Auf diesen Aspekt wird in einem späteren Abschnitt gesondert eingegangen.

### *Design Board*

Das Design Board ist für die Modellierung der Ablauflogik verantwortlich. Die Basis hierfür stellen Services dar. Diese können innerhalb des Werkzeuges aus verschiedenen Quellen gesucht und anschließend per Drag & Drop in das Design Board gezogen werden. Es ist eine vielfältige Auswahl von Service Providern möglich. Neben den Services aus der Services Registry gibt es beispielsweise die Möglichkeit, direkt auf SAP Backend-Systeme, BI-Datenquellen oder JDBC-Prozeduren zuzugreifen. Anhand der Schnittstellenbeschreibung der Services lassen sich Oberflächenelemente für Ein- und Ausgabefelder definieren. Der Datenfluss wird durch Verbindung der Oberflächen- und Servicekomponenten mithilfe von Konnektoren hergestellt. Anschließend kann die Datenübergabe zwischen den Elementen flexibel konfiguriert werden. Eine grafische Anordnung verschiedener Formulare und Tabellen ist durch so genannte Containerelemente möglich. Die nachfolgende Abbildung zeigt diese beiden wichtigen Gruppen von Elementen, wobei UI-basierte Elemente grün und Service-Elemente blau dargestellt werden.<sup>118</sup>

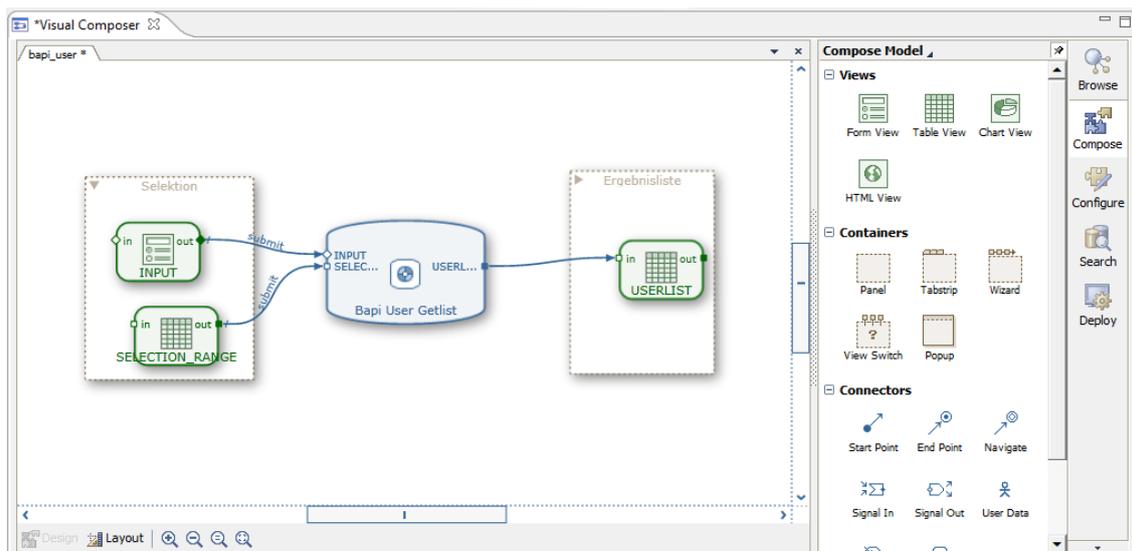


Abbildung 4-8: Design Board des Visual Composer

Quelle: Eigene Darstellung.

<sup>118</sup> Vgl. Salein, Marcel u.a. (2010), S. 115.

### *Layout Board*

Wie bereits erwähnt, werden Oberflächen aus vordefinierten und wiederverwendbaren UI-Komponenten zusammengesetzt, die sich aus den Schnittstellen verwendeter Services ergeben.<sup>119</sup> Im Design Board kann bereits der Datenfluss zwischen den Elementen konfiguriert werden, was sich auf dargestellte Oberflächenelemente im Layout Board auswirkt. Die Layout-Ansicht dient daher zur Nachbearbeitung und Optimierung dieser konfigurierten Bildschirmmasken.

Die Darstellung im Layout Board sieht der fertigen Anwendungsoberfläche sehr ähnlich. In diesem Arbeitsschritt geht es prinzipiell um die Anordnung und Darstellung der UI-Komponenten. Änderungen an den Attributen der jeweiligen Oberflächenelemente werden sofort in die Vorschau übernommen.

### **4.3.3 Modelltransformation**

Die Beschreibung der Visual Composer Modelle erfolgt in GML (Generic Modeling Language). Hierbei handelt es sich um eine auf XML basierende proprietäre SAP Modellierungssprache. Die Validierung des Modells erfolgt anhand des GML DOM, welches das Metamodell der Modellierungssprache darstellt. Das Modell wird fortwährend vom Modellierungswerkzeug auf seine Gültigkeit überprüft, so dass keine inkonsistenten Modelle entwickelt werden können. Die Speicherung des Modells wird zentral im DTR (Design Time Repository) vorgenommen.<sup>120</sup>

Im Zuge einer ersten Transformation wird das Modell durch den Compiler nach XGL (Executable GUI Language) umgewandelt. „Die Executable GUI Language ist eine generische, ausführbare Spezifikation grafischer Benutzerschnittstellen (GUI) für Geschäftsanwendungen.“<sup>121</sup> Hierbei handelt es sich um eine deklarative Spezifikationssprache für mehrere UI-Laufzeitumgebungen. Dies bedeutet, dass das XGL-Modell geräte- und programmiersprachenunabhängig ist.<sup>122</sup>

Erst durch weitere Schritte wird eine lauffähige Anwendung erzeugt. Derzeit werden als Laufzeitumgebungen für das XGL-Modell Web Dynpro Java und Flex unterstützt. Das Vorgehen hin zu einer lauffähigen Anwendung ist für beide Technologien differenziert. Für die Adobe Flex Umgebung, die ein Adobe Flash Plug-In beim Endanwender erfordert, ist eine erneute Transformation durch den Compiler notwendig. Hierbei wird aus dem XGL-Modell Quellcode in den Adobe-eigenen

---

<sup>119</sup> Vgl. Rauscher, Jan / Stiehl, Volker (2007), S. 71.

<sup>120</sup> Vgl. Kessler, Karl u.a. (2007), S. 364.

<sup>121</sup> Salein, Marcel u.a. (2010), S. 90.

<sup>122</sup> Vgl. Salein, Marcel u.a. (2010), S. 90.

Standards MXML und Actionscript erzeugt.<sup>123</sup> Bei Web Dynpro Java wird hingegen auf eine erneute Transformation des XGL-Modells verzichtet und stattdessen das Modell direkt an das Web Dynpro Framework übergeben und zur Laufzeit interpretiert.<sup>124</sup>

Der Vollständigkeit halber soll erwähnt werden, dass auch VoiceXML als Laufzeitumgebung unterstützt wird. Hiermit lassen sich Anwendungen erstellen, die über Spracheingabe gesteuert werden können. Es wird jedoch zusätzlich der SAP Voice Server benötigt, und diese Laufzeitumgebung findet nur vereinzelt innerhalb einer Composite Application Anwendung.<sup>125</sup>

#### 4.3.4 Bewertung

Mit dem Visual Composer ist der SAP AG ein sehr einfach und intuitiv zu bedienendes Modellierungswerkzeug gelungen, sofern stets die benötigten Services in einer geeigneten Art und Weise zur Verfügung stehen. Die Erfahrung hat gezeigt, dass die Nutzung der mit dem CAF erzeugten Services am einfachsten unter Zuhilfenahme des generierten Webservices aus dem Service Repository gelingt. Hierfür sind jedoch noch Konfigurationsarbeiten außerhalb des NetWeaver Developer Studios notwendig, wie beispielsweise das Einrichten einer Destination zum benötigten Zielsystem.

Das Modellierungskonzept des Visual Composer lässt wesentliche Elemente aus dem Konzept von MDA wiedererkennen. So ist das führende Entwicklungsartefakt ein dokumentiertes Modell. Weiterhin entsteht die lauffähige Software durch mehrere Modelltransformationen, auch wenn die Zuordnung zu den von MDA definierten Modellen nicht eindeutig ausfällt. Das Modell im GML-Format ist mit dem PIM (Platform-Independent-Model) gleichzusetzen. Durch die Transformation in das XGL-Format bleibt das Modell weiterhin plattformunabhängig, da weiterhin alle Laufzeitumgebungen einheitlich bedient werden können. Entsprechend der Auffassung der MDA ist dies also kein PSM (Platform-Specific-Model).

Aus dem XGL-Modell können sowohl Compiler als auch Interpreter bedient werden. Dies zeigt, dass der von SAP gewählte Ansatz tatsächlich flexibel und wiederverwendbar ist. Genauere Aussagen diesbezüglich können jedoch nicht getroffen werden, da kein Einblick in die Modelltransformationen und Templates geworfen werden kann. Somit sind auch keine kundenspezifischen Änderungen am

---

<sup>123</sup> Vgl. Salein, Marcel u.a. (2010), S. 90.

<sup>124</sup> Vgl. Ebenda, S. 121f.

<sup>125</sup> Vgl. Ebenda, S. 124.

Framework möglich, was der Idee hinter MDA widerspricht. Als weiteres Differenzierungsmerkmal zur Model Driven Architecture ist die Modellierungssprache zu erwähnen, da an dieser Stelle kein UML zum Einsatz kommt. Dies ist jedoch eher von Vorteil, da die domänenspezifische Sprache sehr gut an die Erfordernisse der GUI-Erstellung für Geschäftsanwendungen angepasst ist.

Bei der Erstellung einer Anwendung müssen alle Anforderungen durch die grafische Notation des Visual Composers unterstützt sein, da kein Einfluss auf die erstellte Software genommen werden kann. So wird beispielsweise für Web Dynpro keine eigene Anwendung geniert, welche nachbearbeitet werden könnte. Auf diese Weise ist das Modell stets konsistent und repräsentiert die tatsächliche Software. Durch den fehlenden Einfluss auf die Modelltransformation und der Unveränderbarkeit des Metamodells können schon kleine Anforderungen, welche nicht umgesetzt werden können, zu einem K.O.-Kriterium werden. Einzig durch das Einbinden von Web Dynpro Komponenten in die Visual Composer Anwendung kann aus dem engen Rahmen ausgebrochen werden.<sup>126</sup> Dieses Vorgehen ist jedoch nicht zu empfehlen, da Technologien miteinander vermischt werden und das Visual Composer Modell anschließend nicht die vollständige Software beschreibt.

## **4.4 Web Dynpro Java**

### **4.4.1 Einsatzgebiet und Konzept**

Einige Aspekte dieser UI-Technologie wurden bereits im vorherigen Unterkapitel thematisiert. Im Vergleich zu einer im Visual Composer erstellten Anwendung wurde dabei herausgestellt, dass Web Dynpro der flexiblere Ansatz hinsichtlich des Funktionsumfangs ist. Des Weiteren stellt diese Technologie die bevorzugte Alternative seitens SAP für Anwendungen im BPM Kontext dar.<sup>127</sup>

Als SAP im Jahr 2001 mit der Entwicklung von Web Dynpro begonnen hat, wurden folgende Ziele mit hoher Priorität verfolgt:

- Verwendung eines MVC-Programmiermodells (Model-View-Controller) zur Entwicklung von transaktionalen und webbasierten Geschäftsanwendungen
- Reduzierung von manuellem Entwicklungsaufwand durch automatische Generierung umfassender Codebestandteile

---

<sup>126</sup> Vgl. SAP AG (2008), Folie 13.

<sup>127</sup> Vgl. SAP AG (2012), Folie 14.

- Entwicklung einer den Erfordernissen angepassten modellgetriebenen Entwicklungsumgebung zur Erstellung von Geschäftsanwendungen
- Client-unabhängiges Metamodell
- Einheitliche Anbindung unterschiedlicher Backend-Technologien
- Einführung eines Komponentenmodells für lose Kopplung und hohe Wiederverwendbarkeit von Anwendungsbestandteilen
- Nutzung gewohnter Annehmlichkeiten aus der klassischen SAP-Oberflächenentwicklung in Dynpro wie Wertehilfen, Mehrsprachigkeit und Typenvalidierung<sup>128</sup>

Die aus diesen Anforderungen heraus entstandene Architektur setzt das MVC Entwurfsmuster konsequent um und begünstigt somit eine hohe Wiederverwendbarkeit und eine klare Trennung von Oberflächengestaltung und Anwendungslogik. Die nachfolgende Abbildung gibt hierzu einen Überblick.

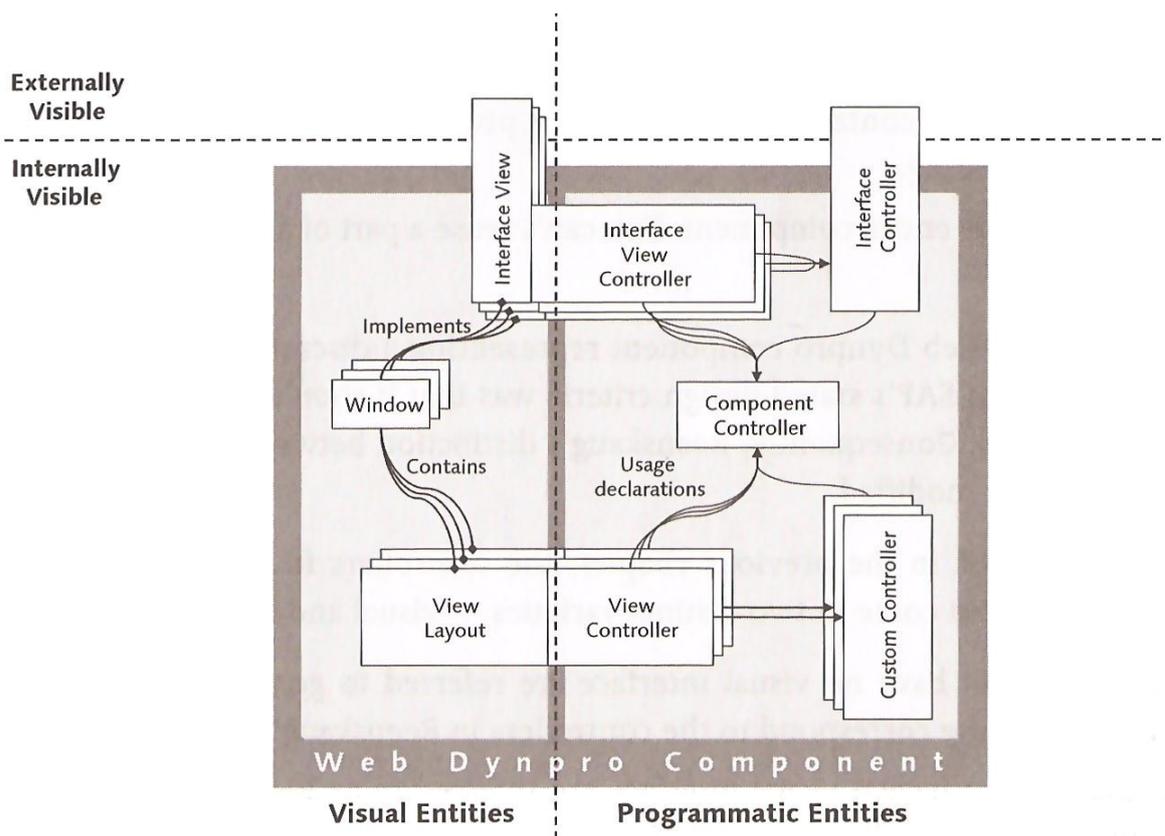


Abbildung 4-9: Struktur einer Web Dynpro Component

Quelle: Whealy, Chris (2007), S. 60.

<sup>128</sup> Vgl. Kessler, Karl u.a. (2007), S. 255ff.

Eine Web Dynpro Komponente besteht demnach aus mehreren sichtbaren Elementen (Windows und Views) sowie programmierten Bestandteilen (verschiedene Arten von Controllern). Ein Window bettet eine oder mehrere Views ein und steuert auf diese Weise, welche Bestandteile der Oberfläche zu einem Zeitpunkt sichtbar sind. Weiterhin ist das Window für die Definition von Navigation-Pfaden verantwortlich. Eine View ist eine konkrete Oberfläche, welche Ein- und Ausgabefelder sowie deren Layout definiert. Die Interaktion mit der Benutzeroberfläche durch Reagieren auf Ereignisse des Phasenmodells erfolgt durch den zugehörigen View Controller. Dieser wiederum steht in Verbindung mit dem Component Controller. Dieser ist in jeder Web Dynpro Komponente genau einmal vorhanden und die zentrale Controller-Instanz. Er ist unter anderem für die übergreifende Speicherung von Daten zuständig und innerhalb aller anderen Controller-Klassen referenziert. Sofern eine separate Modellklasse verwendet wird, steuert der Component Controller die Kommunikation zwischen Modell und View.<sup>129</sup> Custom Controller sind optionale Elemente und können zur besseren Strukturierung der Anwendung verwendet werden.<sup>130</sup>

Interface Views und Interface Controller stellen die nach außen hin sichtbare Schnittstelle dar und werden für die Wiederverwendung innerhalb anderer Web Dynpro Komponenten benötigt. Hierdurch wird die Komplexität der Anwendung für den Aufrufer ausgeblendet. Folglich wird es ermöglicht, eine Web Dynpro Komponente wie eine Black-Box zu nutzen. Dieses Verfahren wird beispielsweise für die Verwendung des SAP List Viewers (ALV) genutzt, welcher vielfältige Anzeige- und Auswertungsfunktionen für Tabellen bereitstellt.<sup>131</sup>

#### **4.4.2 Modellierungswerkzeug**

Die Entwicklung von Web Dynpro Anwendungen erfolgt anhand eines abstrakten Modells, was den Entwickler von grundlegenden und wiederkehrenden Aufgaben befreit sowie weite Teile der Entwicklung auf die einfache Definition von Eigenschaften diverser Entitäten und deren Beziehungen zueinander verringert. Diese deklarative Entwicklung erfolgt unter Verwendung der Web Dynpro Werkzeuge, welche im SAP NetWeaver Developer Studio als eigene Perspektive bereitgestellt werden.

Die grundsätzliche Architektur der Anwendung ist vorgegeben und wird durch die Entwicklungsumgebung zu weiten Teilen durch eine grafische Modellierung unter-

---

<sup>129</sup> Vgl. Kessler, Karl u.a. (2007), S. 261ff.

<sup>130</sup> Vgl. Ebenda, S. 276.

<sup>131</sup> Vgl. Whealy, Chris (2007), S. 62f.

stützt. Der durch den Anwendungsentwickler manuell zu erstellende Quellcode beschränkt sich auf die eigentliche Anwendungslogik. Hierzu zählen die Behandlung von Benutzeraktionen, das Auslösen von Navigationsübergängen und der Aufruf der Geschäftslogik.<sup>132</sup>

Aus dem Blickwinkel der modellgetriebenen Entwicklung und der Verwendung innerhalb von Composite Applications sind vier Web Dynpro Werkzeuge von besonderem Interesse. Mit diesen Werkzeugen alleine ist zwar keine lauffähige Anwendung erstellbar, jedoch sind sie an der hohen Produktivität des Frameworks maßgeblich beteiligt.

### Component Modeler

Diese Sicht erlaubt die Verwaltung von Web Dynpro Komponenten, hierzu zählen insbesondere das Anlegen von Komponenten und das Herstellen von Verbindungen zu Web Dynpro Anwendungen, welche den Einstiegspunkt in die Komponente definieren. So kann beispielsweise der Zugang für zwei verschiedene Anwendergruppen über zwei Anwendungen realisiert werden, welche auf die identische Komponente verweisen, jedoch differenzierte Startparameter an die Komponente übergeben.<sup>133</sup> Die nachfolgende Abbildung visualisiert diesen Sachverhalt für die gemeinsame Nutzung einer Komponente für die Benutzergruppen Mitarbeiter und Manager.

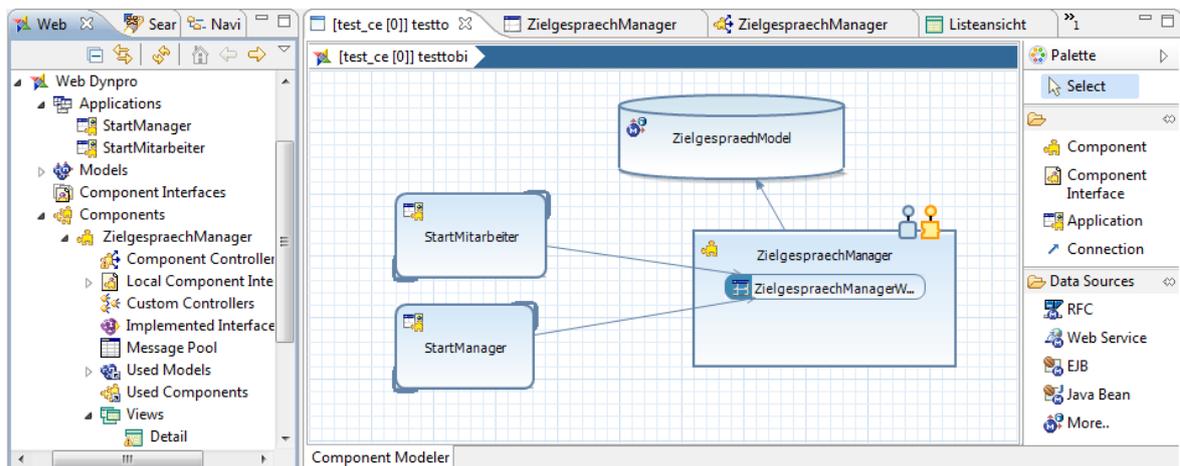


Abbildung 4-10: Component Modeler

Quelle: Eigene Darstellung

Weiterhin bietet der Component Modeler die Möglichkeit, verschiedene Datenquellen zu nutzen. Im Kontext einer Composite Application und einer konsequenten

<sup>132</sup> Vgl. Kessler, Karl u.a. (2007), S. 275.

<sup>133</sup> Vgl. Ebenda, S. 256f.

Umsetzung des SOA-Gedanken ist die Entscheidung für eine Webservice-Schnittstelle sinnvoll. In der Abbildung 4-10 wurde beispielsweise ein Datenmodell basierend auf dem Composite Application Framework (Vgl. 4.2) aus der Service Registry importiert. Das Web Dynpro Generationsframework erzeugt anschließend ein Adaptive Webservice Model anhand der Informationen aus der WSDL des Services.<sup>134</sup>

### *Data Modeler*

Dieses Werkzeug unterstützt beim Anlegen und Zusammenfügen der Elementarten View, Window und verschiedener Controller sowie dem Einbetten von anderen Komponenten innerhalb einer Web Dynpro Komponente. Weiterhin kann auch an dieser Stelle die Verbindung zu verschiedenen Datenquellen hergestellt werden. Ein weiterer wichtiger Aspekt dieses Modellierungswerkzeuges ist die Konfiguration des Datenflusses, das so genannte Context-Mapping, zwischen den verschiedenen Elementen.

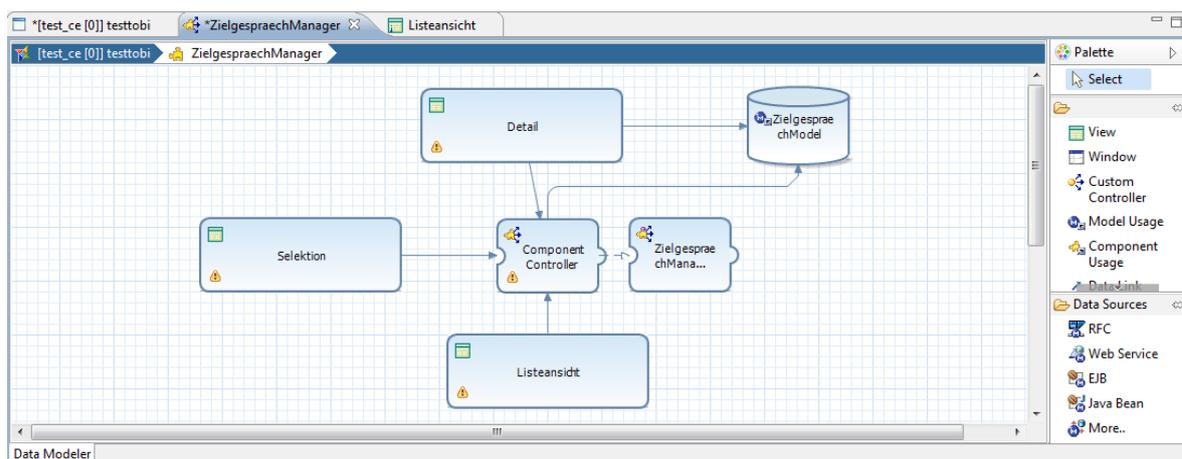


Abbildung 4-11: Data Modeler

Quelle: Eigene Darstellung

### *Navigation Modeler*

Wie bereits erwähnt wurde, ist ein Window maßgeblich am Layout der Web Dynpro Anwendung beteiligt. Durch die Komposition verschiedener Views in einem Viewset können mehrere Sichten in einem Browserfenster gleichzeitig angezeigt werden. Die Navigation wird durch Verbinden von sogenannten Inbound- und Outbound-Plugs der Views hergestellt. Diese können in den View-Eigenschaften

<sup>134</sup> Vgl. Kessler, Karl u.a. (2007), S. 278ff.

definiert werden. In der nachfolgenden Abbildung ist exemplarisch eine dreistufige Oberfläche aus Selektions-, List- und Detailview abgebildet.

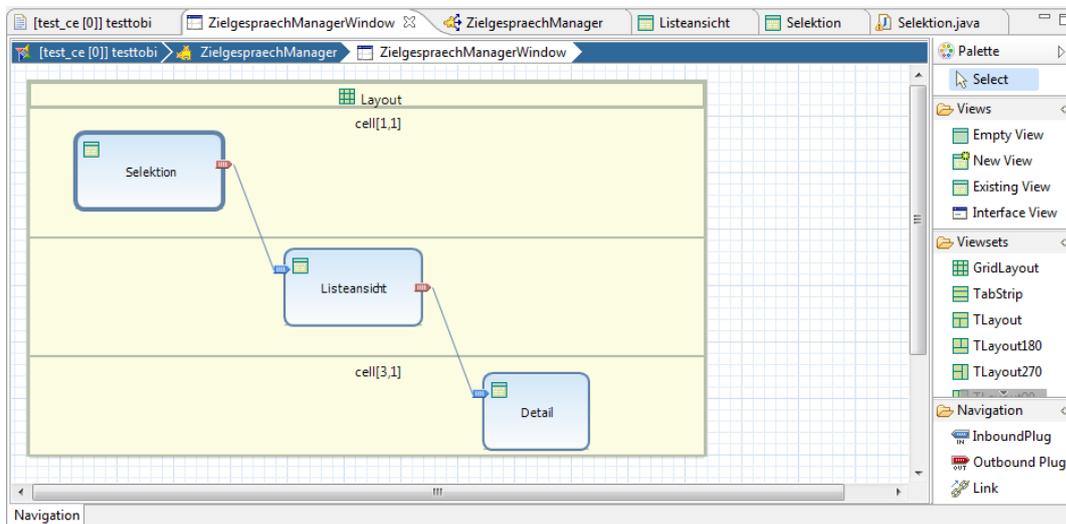


Abbildung 4-12: Navigation Modeler

Quelle: Eigene Darstellung

### *View Designer*

Die Erstellung der Oberflächen erfolgt durch einen WYSIWYG-Editor (What you see is what you get). Das bedeutet, die erstellten und konfigurierten UI-Elemente werden dem Aussehen in der fertigen Applikation möglichst ähnlich in der Vorschau dargestellt. Verwendbare Oberflächenelemente können aus einem umfangreichen Katalog ausgewählt werden und mit Daten des Controllers verbunden werden. Dieses Verfahren wird Data-Binding genannt. Weiterhin können UI-Elemente Ereignisse auslösen, die im Controller behandelt werden können. Aus dem View Designer können Methoden erstellt werden, die die ausgelösten Ereignisse behandeln.<sup>135</sup>

<sup>135</sup> Vgl. Kessler, Karl u.a. (2007), S. 277f.

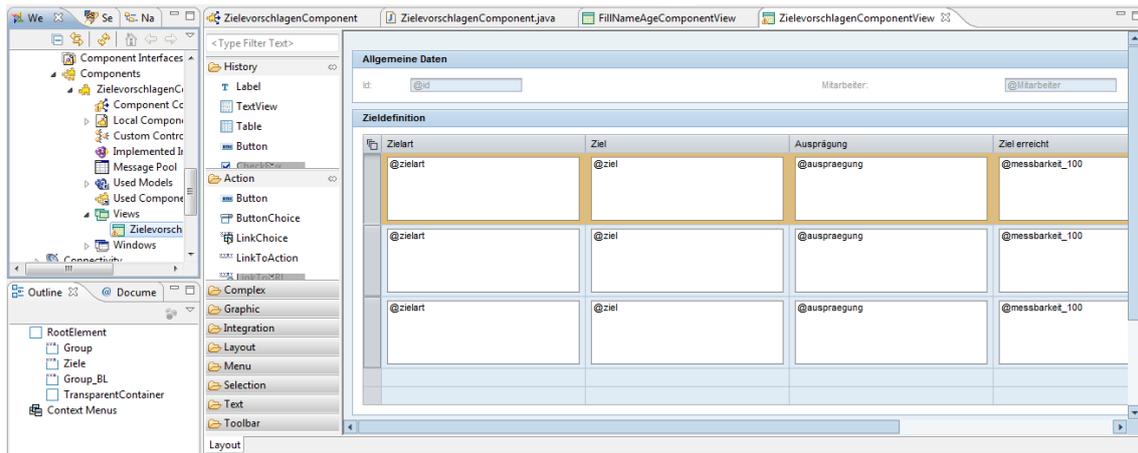


Abbildung 4-13: View Designer

Quelle: Eigene Darstellung

#### 4.4.3 Modelltransformation

Entsprechend der Zielsetzung von SAP sollte das Metamodell hinter Web Dynpro auf keine spezielle Clienttechnologie fokussieren, sodass die UI anhand der Web Dynpro Komponente für mehrere Arten von Endgeräten erzeugt werden kann. Da die Oberflächenentwicklung mithilfe des View Designers vollständig deklarativ mithilfe eines Modellierungswerkzeuges erfolgt, kann das Modell in verschiedene Implementierungsvarianten transformiert werden. Die Erzeugung des notwendigen Quellcodes der UI, wie beispielsweise HTML, JavaScript oder Cascading Style Sheets, erfolgt indes erst während der Laufzeit. Der Applikationsserver erstellt diese Artefakte und überträgt sie an den Client.<sup>136, 137</sup>

Serverseitig ist diese Flexibilität nicht gegeben, da durch Generatoren des Web Dynpro Frameworks bereits zum Designzeitpunkt Artefakte in der Programmiersprache Java angelegt werden, welche in der Regel durch den Entwickler mit eigenen Java-Anweisungen angereichert werden. So wird beispielsweise beim Anlegen einer View im Data Modeler automatisch das Grundgerüst des zugehörigen View Controllers als Java Klasse generiert. Die manuell eingefügten Codestrecken erfolgen innerhalb geschützter Bereiche der Java-Klasse. Diese sind durch Annotationen kenntlich gemacht. Ein erneutes Generieren der Java-Klasse ist beispielsweise notwendig, wenn mit Hilfe des Modellierungswerkzeuges eine neue Controller-Methode angelegt wird. Bei den Backend-Komponenten handelt es sich daher prinzipiell um klassische Softwareentwicklung mit der Hilfestellung, dass ein Rahmenwerk beim Anlegen des Artefakts vorgegeben wird.

<sup>136</sup> Vgl. SDN WDJJava FAQs (2008), (siehe Internetverzeichnis).

<sup>137</sup> Vgl. Whealy, Chris (2007), S. 39f.

Eine weitere Stelle, an denen Generatoren zur Designzeit verwendet werden, ist der Import von Serviceschnittstellenbeschreibungen. Hierbei werden Proxy-Klassen angelegt, welche zur Laufzeit die Kommunikation mit den Services realisieren.

Da das Modell der UI erst zur Laufzeit interpretiert wird und eine Programmierschnittstelle zwischen Controller und dem Web Dynpro Metamodell vorhanden ist, hat dieser Technologieansatz die Eigenart, dass modellierte Anwendungsteile durch dynamische Programmierung zur Laufzeit durch Manipulation des Modells verändert werden können. So können beispielsweise während der Anwendungsausführung neue Formularfelder zur Benutzeroberfläche hinzugefügt werden.<sup>138</sup> Dieses Verfahren erhöht zwar die Flexibilität des Frameworks, sollte jedoch nur in Ausnahmefällen eingesetzt werden, da Inkonsistenzen zwischen Modell und der tatsächlichen UI resultieren.

#### **4.4.4 Bewertung**

Aufgrund des hohen Funktionsumfangs von Web Dynpro Java wurde die Betrachtung auf Aspekte der modellgetriebenen Entwicklung und Funktionen, welche für Composite Applications von Bedeutung sind, beschränkt. Es hat sich gezeigt, dass Web Dynpro die technische Alternative für Benutzeroberflächen mit hohen Anforderungen an die Interaktion mit dem Anwender sowie Flexibilität bei der Gestaltung ist. Aufgrund der domänenspezifischen Modellierungswerkzeuge wird dabei eine hohe Produktivität erreicht und erstellte Oberflächen erhalten stets ein einheitliches Erscheinungsbild. Der Funktionsumfang des Frameworks deckt weite Teile der Anforderungen gewöhnlicher Geschäftsanwendungen ab.

Vom Standpunkt der modellgetriebenen Entwicklung ist Web Dynpro kein bahnbrechender Ansatz, da manuelles Coding zu einem wesentlichen Anteil notwendig ist. Die konsequente Umsetzung im Bereich der clientspezifischen Artefakte ist jedoch eine wesentliche Stärke des Frameworks. Die Funktionsfähigkeit dieses Ansatzes erkennt man daran, dass auch Oberflächen aus dem Visual Composer durch das Web Dynpro Framework gerendert werden können. Allerdings ist derzeit der Webbrowser die einzig unterstützte Clientplattform. Aufgrund vieler Verbesserungen des Frameworks in den letzten Jahren wurden der Funktionsumfang sowie die Leistungsfähigkeit bereits erheblich verbessert.

---

<sup>138</sup> Vgl. Ebenda, S. 331f.

Der durch Web Dynpro generierte UI-Quellcode für den Webbrowser ist recht komplex und somit nur schwer verständlich. So erfolgt beispielsweise keine Trennung des Layouts (HTML) und clientseitiger Logik (JavaScript), stattdessen sind HTML- und JavaScript-Codefragmente vermischt. Eine Einflussnahme ist an dieser Stelle nicht möglich, da SAP keinen Zugriff auf Templates gewährt, jedoch erhöht diese Tatsache die Inkompatibilität zwischen verschiedenen Webbrowsern. Somit sollte Web Dynpro nicht für nach außen geöffnete Geschäftsanwendungen verwendet werden, da externen Anwendern nur bedingt die Verwendung des Microsoft Internet Explorer vorgeschrieben werden kann.

Die Serverkomponenten einer Web Dynpro Anwendung können nur durch einen Entwickler programmiert werden. Erschwerend kommt hinzu, dass der Java-Quellcode aufgrund der Interaktion mit dem Web Dynpro Frameworks recht schnell komplex und unübersichtlich wird. Hilfsmittel, wie der aus Web Dynpro ABAP bekannte Code-Wizard, gibt es für das Java Pendant derzeit nicht.

## **4.5 Business Process Management**

### **4.5.1 Einsatzgebiet und Konzept**

Das Business Process Management unterstützt den gesamten Lebenszyklus eines IT-gestützten Prozesses und erleichtert die Entwicklung lauffähiger Prozessmodelle. Das Vorgehen hin zu einem vollständigen systemtechnisch abgebildeten Geschäftsprozess unterteilt SAP in einen Zyklus aus vier Phasen.<sup>139</sup> Zu Beginn wird in der Analysephase der Geschäftsprozess analysiert und fachlich in einer Notation wie beispielsweise BPMN oder EPK dokumentiert. Die Arbeitsergebnisse bilden den Ausgangspunkt der Prozessentwicklungsphase. In diesem Schritt wird der Geschäftsprozess innerhalb eines technischen Systems implementiert. Die anschließende Deployment-Phase bezeichnet die Produktivsetzung des Geschäftsprozesses innerhalb der bestehenden IT-Landschaft. Abschließend erfolgt ein Monitoring des Geschäftsprozesses mittels Kennzahlen.<sup>140</sup>

Für die Anwendung des SAP BPM sind grundsätzlich zwei Prozessarten zu unterscheiden. Zum einen gibt es Application Core Processes, welche bereits durch Funktionen der SAP Business Suite abgedeckt werden. Diese werden in der Regel einmalig durch Customizing eingerichtet und unterliegen anschließend kaum Veränderungen. Als zweite Prozessart werden Composite Processes genannt.

---

<sup>139</sup> Vgl. Snabe, Jim Hagemann u.a. (2009), S. 27.

<sup>140</sup> Vgl. Baumgartl, Axel / Mebus, Frank / Seemann, Volker (2010), S. 202.

Diese werden nicht durch SAP ausgeliefert, da sie in der Regel unternehmensspezifisch sind oder Fremdsysteme mit einbeziehen. Es wird davon ausgegangen, dass diese Prozessart häufigen Änderungen unterliegt. Das Einsatzgebiet des SAP BPM zielt, wie es der Name bereits vermuten lässt, auf Composite Processes ab.<sup>141</sup>

Beim Geschäftsprozessmanagement ist es wichtig, dass die Fach- und IT-Welt eine gemeinsame Sprache sprechen und sich durch Modelle verständigen können, die von beiden Seiten verstanden werden. Die Schwierigkeit hierbei liegt in den unterschiedlichen Sichtweisen beider Rollen. Während für die fachliche Seite der grundsätzliche Ablauf wichtig ist, benötigt die technische Seite wesentlich mehr Details. Die Anreicherung des fachlichen Modells mit technischen Informationen und zusätzlichen Prozessschritten, die der tatsächlichen Architektur Rechnung tragen, stört die Übersichtlichkeit und somit die Verständlichkeit des Prozessmodells. In den letzten Jahren zeichnet sich ab, dass für dieses Notationsproblem ein gemeinsamer Nenner gefunden worden ist. Die Business Process Modeling Notation (BPMN) der OMG vereint die notwendige grobe Granularität in der Übersicht, ermöglicht es aber trotzdem ausreichend technische Informationen in das Modell zu verankern.<sup>142</sup>

Innerhalb der Referenzarchitektur einer Composite Application ist die Steuerung des Prozessablaufes zentrale Aufgabe des BPM. Hierbei liegt der Fokus auf einer User-zentrierten Prozessverarbeitung, das bedeutet BPM findet dort Einsatz, wo eine zentrale Anforderung die Interaktion mit verschiedenen Anwendern ist. Diese kollaborativen Geschäftsprozesse haben andere Erfordernisse als beispielsweise Geschäftsprozesse zwischen verschiedenen Anwendungssystemen ohne Nutzerinteraktion. Dieser Fakt stellt ein wichtiges Abgrenzungskriterium zur SAP PI (Process Integration) dar. Diese fokussiert auf Anwendungsintegration und stellt dafür eine Vielzahl von Adaptertechnologien bereit, welche beim BPM derzeit nicht anzutreffen sind. Zukünftig werden jedoch PI und BPM näher zusammengeführt, da Adapterfunktionen auch vermehrt in BPM-Szenarien benötigt werden.<sup>143</sup>

Bei der Interaktion mit Anwendern spielt zur Laufzeit das SAP Portal eine wesentliche Rolle, da Prozessaufgaben (Tasks) beim User durch einen Eintrag in der Universal Worklist (UWL) dargestellt werden. Von dort aus wird in die aufgabenspezifische Benutzeroberfläche verzweigt.

---

<sup>141</sup> Vgl. Snabe, Jim Hagemann u.a. (2009), S. 292ff.

<sup>142</sup> Vgl. Stiehl, Volker (2009), (siehe Internetverzeichnis).

<sup>143</sup> Vgl. SAP AG (2012b), Folie 15.

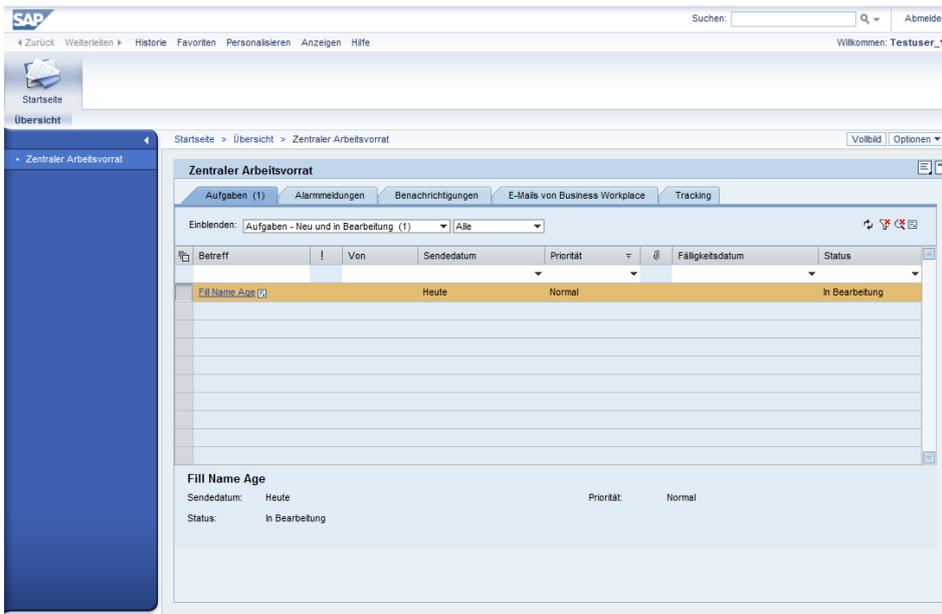


Abbildung 4-14: Universal Worklist für den Zugriff auf Prozessaufgaben

Quelle: Eigene Darstellung

#### 4.5.2 Modellierungswerkzeug

Für die Verwaltung des Prozesslebenszyklus spielt das NetWeaver Developer Studio eine zentrale Rolle. Sowohl Analysten als auch Entwickler steht eine eigene Perspektive zur Verfügung. Während in der Prozessmodellierungsperspektive nur ein begrenzter Funktionsumfang zur Verfügung steht, ist in der Ansicht für den Prozessentwickler der volle Funktionsumfang inklusive Zugang zu technischen Details vorhanden. Dies ermöglicht es dem Fachanwender, ein Grundgerüst des angestrebten Prozesses zu modellieren, welcher anschließend ohne einer weiteren manuellen Transformationen als Basis der Realisierung dient.

Im vorherigen Abschnitt wurde bereits erwähnt, dass als domänenspezifische Modellierungssprache BPMN zum Einsatz kommt.<sup>144</sup> Die Notation wird laut SAP in der Version 2.0 verwendet.<sup>145</sup> Die Realität zeigt jedoch, dass BPM im Vergleich zur BPMN 2.0 Spezifikation nur wenig Sprachelemente zur Verfügung stellt und somit den Standard nicht vollständig unterstützt. Im Anhang I befindet sich eine Übersicht der durch BPM unterstützten BPMN 2.0 Sprachelemente. Die nachfolgende Abbildung zeigt das grafische Modellierungswerkzeug. Hierbei ist die typische Anordnung von Aktivitäten, Events und Gateways innerhalb von Pools und Lanes erkennbar. Auf eine umfassende Einführung in BPMN wird an dieser Stelle verzichtet.

<sup>144</sup> Vgl. Snabe, Jim Hagemann u.a. (2009), S. 306.

<sup>145</sup> Vgl. SAP AG (2012b), Folie 9.

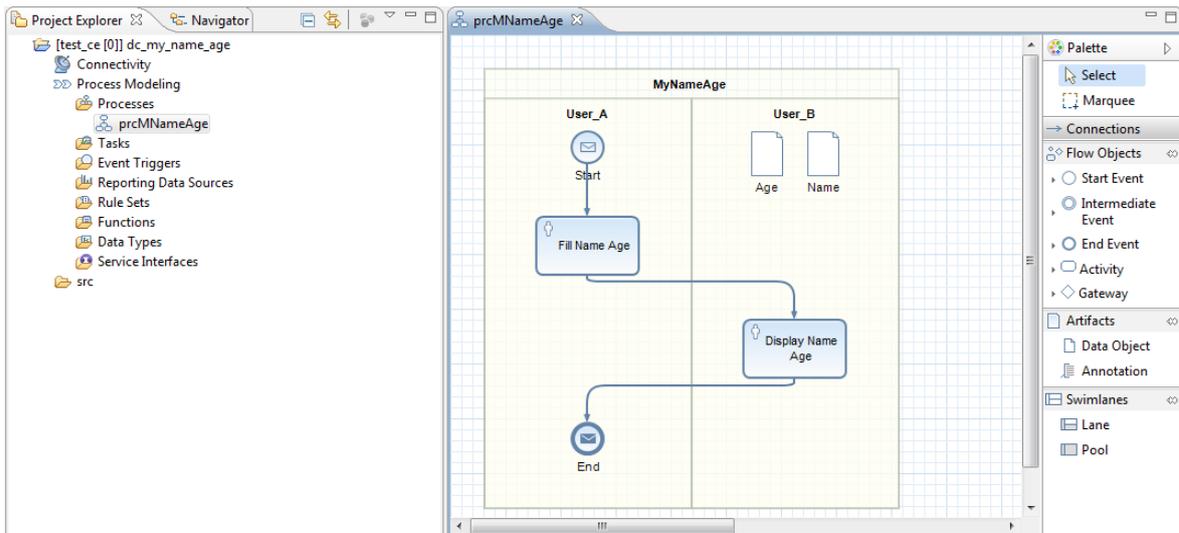


Abbildung 4-15: Process Composer

Quelle: Eigene Darstellung

Das Modellierungswerkzeug bietet eine Importschnittstelle für BPMN 2.0 Modelle an. Der Versuch, ein in Aris Express 2.3 erstelltes Prozessmodell zu importieren, war erfolglos. Diese Funktion ist dem Anschein nach nur für den Austausch innerhalb verschiedener NWDS Installationen nützlich. Anderenfalls wäre es möglich gewesen, dass die fachlichen Prozessexperten auf die Nutzung des recht komplexen NWDS verzichten könnten.

Ein BPM Prozess besitzt einen Kontext, dieser dient dazu, Daten zwischen den einzelnen Prozessschritten zu speichern. Diese Daten können für jede Aktivität entsprechend der Erfordernisse der Schnittstelle zugewiesen werden. Eine Aktivität kann entweder automatisch oder manuell durchlaufen werden. Für automatische Aktivitäten wird ein Serviceaufruf mit entsprechendem Mapping hinterlegt. Die Services können aus dem Enterprise Service Repository, der Service Registry oder per WSDL-Import bekannt gemacht werden. Bei manuellen Aufgaben, welche auch Tasks genannt werden, ist eine der Aufgabe korrespondierende Benutzeroberfläche erforderlich. Diese können entweder mit den bereits beschriebenen Werkzeugen Visual Composer oder Web Dynpro Java erstellt und angebunden werden. Zwar ist Web Dynpro Java die von SAP präferierte UI-Technologie im BPM Kontext, jedoch kann anhand des definierten Datenflusses automatisch sowohl eine Web Dynpro Komponente als auch ein Visual Composer Model generiert werden. Änderungen an den generierten Anwendungen sind möglich, da an dieser Stelle keine Modellbindung vorhanden ist. Dies bedeutet auch, dass Änderungen an der Schnittstelle stets manuell an der UI-Komponente nachvollzogen werden müssen. Eine dritte UI-Alternative stellt die Integration von Web Dynpro ABAP CHiPs dar. Diese Variante ist besonders für Unternehmen mit starkem

ABAP Know-how eine Alternative, allerdings ist diese Komponente nicht Bestandteil des NetWeaver CE und muss daher auf einem separaten SAP-System mit ABAP Applikationsserver betrieben werden.

Die Bearbeitung der Tasks erfolgt zur Laufzeit durch Anwender, diese müssen bereits beim Design spezifiziert werden. Dies erfolgt im NWDS entweder durch eine direkte Zuordnung des Tasks zu einem Nutzer oder durch Zuordnung zu einer Nutzergruppe oder Rolle. Die zweite Variante ist aufgrund der Flexibilität zu bevorzugen. Somit ist es möglich, Aufgaben mehreren potentiellen Bearbeitern zukommen zu lassen.

Der Prozess selbst wird wie ein Service behandelt, so erfolgt beispielsweise der Start einer Prozessinstanz durch den Webserviceaufruf des asynchronen Startinterfaces.

#### **4.5.3 Modelltransformation**

Die dritte Phase des SAP Business Process Management, die Deployment-Phase, beinhaltet die Übertragung des Prozessmodells zum Process Server. Dieser ist integraler Bestandteil der SAP NetWeaver CE Laufzeitumgebung. Der Process Server ist über der reinen Ablaufsteuerung modellierter Prozesse hinaus für Monitoring und Konfiguration zuständig. Dies erfolgt über den SAP NetWeaver Administrator und deckt zudem die vierte Phase des SAP BPM ab.<sup>146</sup>

Der Process Server führt die Prozessmodelle in der BPMN-Notation ohne Übersetzung aus. Das Modell wird also keiner Transformation unterzogen, und demnach wird die Prozessdefinition zur Laufzeit gelesen und interpretiert.<sup>147</sup>

Im Gegensatz zu einem anderen in der Literatur diskutierten Ansatz, der Transformation des BPMN Modells in eine ausführbare BPM Notation, wie beispielsweise BPEL (Business Process Execution Language), bietet das Vorgehen der SAP mehrere Vorteile. Zum einen ist keine Transformation des Modells in eine andere Notation notwendig, welche nach eindeutigen Abbildungsregeln und einer vollständigen Funktionsunterstützung verlangt.<sup>148</sup> Zwar sind in der BPMN 2.0 Spezifikation Abbildungsregeln für die Transformation von BPMN nach WS-BPEL normiert, jedoch weisen die Autoren darauf hin, dass aufgrund des Aufbaus beider Notationen kein vollständiges Mapping erfolgen kann.<sup>149</sup> Der Grund hierfür ist im

---

<sup>146</sup> Vgl. Snabe, Jim Hagemann u.a. (2009), S. 301f.

<sup>147</sup> Vgl. SAP AG (2011), (siehe Internetverzeichnis).

<sup>148</sup> Vgl. Stiehl, Volker (2009), (siehe Internetverzeichnis).

<sup>149</sup> Vgl. OMG (2011), S 475.

flussorientierten Ansatz bei BPMN im Vergleich zu einem blockorientierten Ansatz bei WS-BPEL erklärt. Ein weiterer Vorteil aus der direkten Verwendung des BPMN Modells ist im Debugging zu finden, so kann ein Breakpoint auf eine Aktivität innerhalb des bekannten grafischen Prozessmodells gesetzt werden, um den Wert von Context-Variablen zur Laufzeit einzusehen. Innerhalb einer BPEL-Transformation müsste der Breakpoint in einem wesentlich schwieriger lesbaren Quellcode gesetzt werden.<sup>150</sup>

Die Engine des SAP Process Servers adaptiert bestmöglich die Idee hinter BPMN. Die Abarbeitungsreihenfolge wird daher durch den Sequenzfluss vorgegeben. Hierfür merkt sich die Laufzeitumgebung mit Hilfe von Tokens, wie sie beispielsweise aus der Informatik aus Petri-Netzen bekannt sind, an welcher Stelle im Modell sich die konkrete Instanz befindet und der Token wird bildlich nach Beendigung eines Prozessschrittes zum Folgeschritt geschoben.<sup>151</sup>

#### **4.5.4 Bewertung**

Die Verwaltung des SAP BPM Lebenszyklus mithilfe des NetWeaver CE hinterlässt einen guten Eindruck. Das grafische Modellierungswerkzeug für die Analyse- und Entwicklungsphase lässt sich einfach und intuitiv bedienen, die Erstellung von Benutzeroberflächen in den gängigen browserbasierten UI-Technologien des NetWeaver CE wird unterstützt, und die konsequente Umsetzung der SOA-Gedanken ist erkennbar. Weiterhin ist die Integration von Web Dynpro ABAP als UI-Komponente innerhalb eines Prozesses ein Lösungsansatz für Unternehmen, die ihr Entwickler-Know-how im Bereich der Programmiersprache ABAP weiterhin nutzen möchten. Auf diese Weise sind für die Prozesserstellung keine Java-Programmierkenntnisse notwendig.

Weiterhin bietet die Laufzeitumgebung einen modernen Ansatz zur Interpretation von BPMN Modellen, da die Engine ohne Kompromisse auf den BPMN Standard ausgerichtet ist und keine zusätzliche Transformation durchgeführt wird. Zwar ist bislang die Anzahl unterstützter Sprachelemente im Vergleich zur BPMN 2.0 Spezifikation vergleichsweise gering, jedoch lässt sich mit den vorhandenen Mitteln bereits jetzt eine Vielzahl von Prozessvarianten abbilden. Da BPM ein strategisches Produkt von SAP ist, wird in der Zukunft mit neuen Sprachelementen zu rechnen sein.

---

<sup>150</sup> Vgl. Stiehl, Volker (2009), (siehe Internetverzeichnis).

<sup>151</sup> Vgl. Ebenda, (siehe Internetverzeichnis).

Sobald der BPMN Standard eine steigende Normierung und Verbreitung innerhalb der Wirtschaft erfährt, steigt die Austauschbarkeit von Prozessmodellen und somit auch der Nutzen aus dem Konzept des SAP Process Server. Denn dieser kann bei Vorhandensein eines einheitlichen Standards sowohl eigene als auch fremde Prozessdefinitionen zur Ausführung bringen.<sup>152</sup>

Die Funktionen des NetWeaver Administrators bieten Methoden zur Überwachung von laufenden Prozessen. Dies ist insbesondere dafür wichtig, dass keine Prozessinstanzen aufgrund von Fehlersituationen oder ausbleibender Bearbeitung verweisen. An dieser Stelle können darüber hinaus Erfahrungen für den nächsten Analysezyklus im Rahmen einer kontinuierlichen Verbesserung des Prozesses gesammelt werden.

#### *Business Rules Management*

In dieser Stelle sei auf das Business Rules Management verwiesen, welches das Definieren und Anwenden von Geschäftsregeln realisiert. Es ist möglich, Entscheidungsbäume und -tabellen zu modellieren, welche später zur Laufzeit durch den BPM Prozess berücksichtigt werden. Dieses Verfahren kann auch in anderen Technologien des NetWeaver CE verwendet werden und bietet eine Alternative zum klassischen Customizing über eigens für diese Aufgabe erstellte Datenbanktabellen.

## **4.6 Zusammenfassung**

In der theoretischen Betrachtung von MDSD-Ansätzen wurde aufgeführt, dass die Erstellung von Templates und Generatoren viel Aufwand bedeuten kann und sich deswegen diese Art der Softwareentwicklung erst bei größeren Projekten oder bei einer häufigen Wiederverwendung lohnend auswirkt. Dies gilt für die Konzepte und Technologien des SAP NetWeaver CE kaum, da bereits für die vorgegebene Laufzeitumgebung ausgereifte domänenspezifische Modellierungssprachen und Templates geliefert werden, welche jedoch nicht änderbar sind. Die Betrachtung der einzelnen Technologien hat gezeigt, dass SAP an mehreren Stellen moderne MDSD Architekturen aufweisen kann, diese jedoch durch fehlenden Zugriff auf Templates sowie durch die Vorgabe von Laufzeitumgebungen auf den Nutzwert eines CASE-Ansatzes zurückstuft.

---

<sup>152</sup> Vgl. Stiehl, Volker (2009), (siehe Internetverzeichnis).

Dies mindert die Flexibilität des Modellierungsansatzes, erleichtert jedoch die Nutzung der jeweiligen Technologien, da die Werkzeuge bereits für den Einsatz in der Composition Environment sinnvoll konfiguriert sind. Daher gilt die Einschränkung, dass MDSD für einmalige und kleine Softwarelösungen nicht lohnend ist, an dieser Stelle nur begrenzt. Dies impliziert jedoch eine gewisse Abhängigkeit zum Produkthanbieter, inwiefern das Produkt weiterentwickelt wird und der Fortbestand des Anbieters gesichert ist. Aufgrund der Unternehmensgröße der SAP AG wird dieses Risiko begrenzt. Weiterhin betreiben ein Großteil potentieller NetWeaver CE Nutzer bereits ein SAP ERP, weswegen sie sich bereits in einer starken Abhängigkeitssituation zu SAP befinden und dieses zusätzliche Produkt durch den erhöhten Nutzen innerhalb einer homogenen Systemlandschaft begründet werden kann.

Weiterhin fügen sich die Technologien des NetWeaver CE zu einem gut funktionierenden Gesamtpaket zusammen. Jede Technologie hat ihre spezifische Funktion innerhalb der Referenzarchitektur einer Composite Application. Die Auswahl zwischen den Alternativen bei der Oberflächengestaltung kann anhand einfacher Kriterien zusammengefasst werden. Der Visual Composer bietet sich an, sofern die Oberflächen lediglich geringen Anforderungen genügen müssen, wie sie beispielsweise bei Formularen und Tabellen bestehen, und kaum Programmiererfahrung vorhanden sind. Web Dynpro bietet einen größeren Funktionsumfang und mehr Flexibilität, verlangt jedoch nach fundierten Entwicklerkenntnissen. Sofern das benötigte Java oder alternativ auch ABAP Know-how verfügbar ist, wird die Verwendung von Web Dynpro in der jeweiligen Variante als vorteilhafter bewertet.

Unter Verwendung des Composite Application Frameworks (Service-Schicht), des Business Prozess Managements (Prozess-Schicht) und des Visual Composers (UI-Schicht) lassen sich vollständige Geschäftsprozesse umsetzen ohne dafür eine Zeile Quellcode zu implementieren. Die Bedingung hierfür ist jedoch, dass benötigte Services bereits zur Verfügung stehen und keine hohen Anforderungen an Plausibilitätsprüfungen und Fehlerhandling umgesetzt werden sollen.

In der Betrachtung anhand einschlägiger Literatur und der Umsetzung einfacher Erprobungsszenarien hinterlässt der NetWeaver CE daher einen durchdachten und ausgereiften Eindruck. Inwiefern sich komplexe Anforderungen von Softwareprojekten in der Praxis durch die Composition Environment umsetzen lassen, wird im folgenden Kapitel untersucht.

## 5 Praktischer Einsatz bei der Berliner Stadtreinigung

### 5.1 Berliner Stadtreinigung

Die Berliner Stadtreinigung (BSR) ist der größte kommunale Entsorger Europas. Sie hat die Aufgabe, Abfälle sicher, effizient und umweltverträglich zu entsorgen und ein sauberes Straßenbild zu schaffen. Die hohe Leistung der rund 5.300 Beschäftigten ist auch im bundesweiten Vergleich bekannt. So hat Berlin unter allen vergleichbaren Großstädten die niedrigsten Gebühren. Als modernes Dienstleistungsunternehmen im Eigentum des Landes Berlin verfügt die BSR über das Wissen, die qualifizierten Beschäftigten sowie die unternehmerische Flexibilität, um die abfallwirtschaftlichen Ziele wie auch die Anforderungen an die Reinigung der Hauptstadt zuverlässig und nachhaltig zu gewährleisten.<sup>153</sup>

Für Unternehmen der öffentlichen Hand haben Themen der sozialen Verantwortung und des Engagements einen hohen Stellenwert. Neben einer gesellschaftlichen Verantwortung der gesamten Berliner Bevölkerung gegenüber, stehen auch die Belange der eigenen Mitarbeiter im Mittelpunkt. Hierfür gibt es mehrere Maßnahmen, welche für die Mitarbeitermotivation, den Arbeitsschutz und die Gesundheitsvorsorge eine wichtige Rolle spielen.<sup>154</sup> Durch die Einführung eines betrieblichen Eingliederungsmanagements im Jahr 2010 wurde die Lage für bereits erkrankte Mitarbeiter verbessert und eine Rückkehr in die Arbeitswelt erleichtert.

Weiterhin setzt die Berliner Stadtreinigung zur Unterstützung ihrer operativen Tätigkeiten seit 1986 auf Software der SAP AG und nutzt seitdem eine Vielzahl SAP Systemen. Derzeit befinden sich unter anderem die Produkte ERP, IS-U Waste and Recycling, HCM, CRM, PI und BI neben diversen Eigenentwicklungen im Einsatz. Aufgrund dieser Tatsache hat sich ein vielfältiges Wissen bezüglich des traditionellen SAP Softwarelebenszyklus und der Programmierung in ABAP entwickelt. Dieser Wissensschatz soll bestmöglich bei zukünftigen Erweiterungen und Anpassungen der IT-Landschaft genutzt werden. Da die führende Entwicklungssprache des NetWeaver CE jedoch Java ist, stellt dies einen Paradigmenwechsel und somit eine besondere Herausforderung für die BSR dar.

---

<sup>153</sup> Vgl. BSR (2012), (siehe Internetverzeichnis).

<sup>154</sup> Vgl. BSR (2010b), (siehe Internetverzeichnis).

## 5.2 Betriebliches Eingliederungsmanagement

### 5.2.1 Bedeutung für das Unternehmen

Mit der „Dienstvereinbarung zur Durchführung eines betrieblichen Eingliederungsmanagements (BEM) bei den Berliner Stadtreinigungsbetrieben (BSR) gemäß § 84 Abs. 2 Sozialgesetzbuch IX (SGB IX)“ vom Oktober 2010 hat sich die BSR entschlossen, ein betriebliches Eingliederungsmanagement (BEM) für alle Beschäftigten zu implementieren.<sup>155</sup>

Ein betriebliches Eingliederungsmanagement beinhaltet Maßnahmen der Prävention, Gesundheitsförderung und Rehabilitation nach häufiger oder lange andauernder Krankheit. Die Berliner Stadtreinigung beabsichtigt durch diesen Schritt, ihre Fürsorgepflicht des Arbeitgebers sicherzustellen und Beschäftigte gesund zu halten. Dies beinhaltet Maßnahmen zur Überwindung von Arbeitsunfähigkeit oder Vorbeugung vor erneuter Arbeitsunfähigkeit, Erhalt und Förderung der Arbeitsfähigkeit und Gesundheit sowie Vermeidung von Behinderungen einschließlich chronischer Krankheiten. Die BSR verfolgt somit das Ziel, Arbeitsunfähigkeit vorzubeugen, um Arbeitsplätze langfristig zu erhalten und Kosten für Ausfallzeiten zu senken. Als konkrete Maßnahmen zur Prävention, Gesundheitsförderung und Rehabilitation kommen insbesondere folgende Möglichkeiten in Betracht:

- Arbeitsplatz-, Arbeitsablauf- und Arbeitsplatzgefährdungsanalysen,
- ärztliche Untersuchungen der Betroffenen,
- Prüfung der behindertengerechten Gestaltung des Arbeitsplatzes, der Arbeitsgeräte und des Arbeitsumfeldes, der Arbeitsorganisation und Arbeitszeit,
- Prüfung alternativer Einsatzmöglichkeiten,
- Maßnahmen zur Gesundheitsförderung sowie
- Maßnahmen der betriebsnahen Rehabilitation einschließlich der stufenweisen Wiedereingliederung während bestehender Arbeitsunfähigkeit.<sup>156</sup>

Die Teilnahme des Mitarbeiters am BEM erfolgt auf freiwilliger Basis und unterliegt strengen Datenschutzbestimmungen für alle am Prozess beteiligten Personen.

---

<sup>155</sup> Vgl. BSR (2010), Dienstvereinbarung zur Durchführung eines BEM, S. 1, (siehe Datenträger).

<sup>156</sup> Vgl. Ebenda, S. 1ff, (siehe Datenträger).

Für die Umsetzung des BEM wurde eine eigenständige Organisationseinheit „BM“ in direkter Verantwortung des Personalvorstands eingerichtet sowie die Umsetzung der Dienstvereinbarung durch eine Standardisierung des Prozesses beschlossen. Dies soll eine BEM-Maßnahme grob in folgende vier Phasen gliedern:

1. Die erste Phase dient der Identifizierung betroffener Beschäftigte gemäß den Regelungen der Dienstvereinbarung.
2. Die zweite Phase beinhaltet die Informationsbeschaffung zum Beschäftigten, die Angebotsunterbreitung und das Einholung der Zustimmungen oder Ablehnung des Beschäftigten zur Durchführung des BEM.
3. Nach schriftlicher Zustimmung erfolgt die Betreuung des Beschäftigten durch Maßnahmen der Prävention, Gesundheitsförderung und Rehabilitation.
4. Die letzte Phase sorgt für die Durchführung der Nachsorge und den Abschluss des BEM Verfahren.

An der Durchführung von BEM-Maßnahmen sowie der strategischen Ausrichtung des BEM sind mehrere Personengruppen beteiligt:

- Das **BEM-Team** plant und steuert die strategische Ausrichtung des BEM,
- das **Eingliederungsteam** plant und steuert das operative Geschäft im BEM,
- das **Maßnahmenteam (M-Team)** plant und steuert die Maßnahmen im Einzelfall und
- der **BEM-Maßnahmenverantwortliche** ist verantwortlich für den Einzelfall (Betreuung, Nachsorge etc.).<sup>157</sup>

Die folgende Abbildung veranschaulicht die Zugriffsebenen auf Daten zur BEM-Maßnahme. Je weiter man sich vom Kern entfernt, desto weniger Informationen erhält die Personengruppe zu den BEM-Maßnahmen. Das BEM-Team auf der äußersten Sicht erhält beispielsweise nur den Zugang zu anonymisierten Auswertungen.

---

<sup>157</sup> Vgl. BSR (2010), Dienstvereinbarung zur Durchführung eines BEM, S. 1ff, (siehe Datenträger).

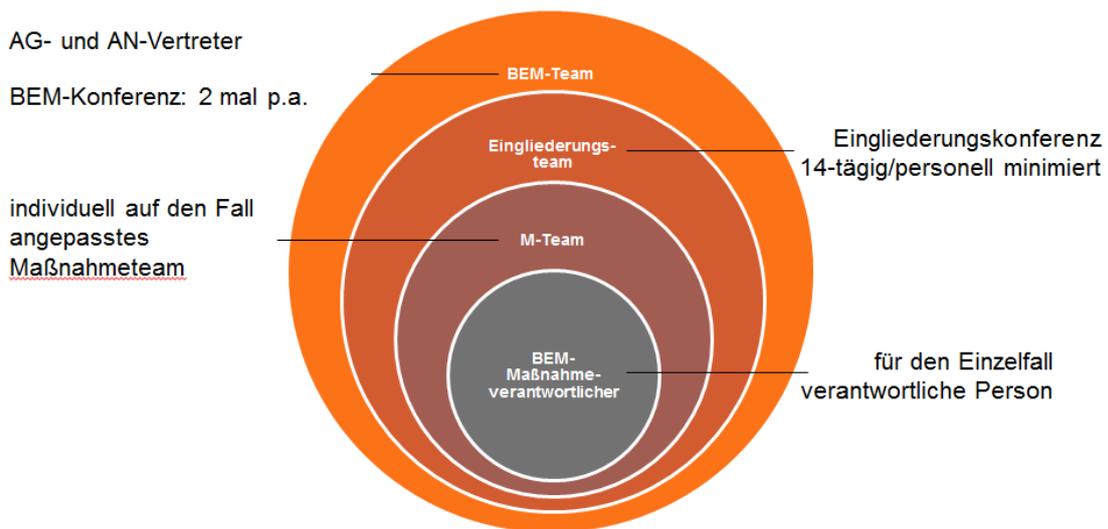


Abbildung 5-1: Zugriffsebenen auf Daten einer BEM-Maßnahme

Quelle: BSR (2012), S. 2.

### 5.2.2 Ist-Zustand

Das betriebliche Eingliederungsmanagement hat bereits seine Arbeit aufgenommen, wird jedoch derzeit noch nicht durch die IT unterstützt. Zurzeit existieren kein standardisierter, dokumentierter und abgestimmter Prozess und die Erfassung der erforderlichen Daten erfolgt unter Zuhilfenahme von Microsoft Excel. Auswertungen sind daher bereits möglich, jedoch ist dies mit manueller Datenpflege verbunden. Somit ist die Aufgabenerfüllung von einem hohen Kommunikationsaufwand begleitet, der durch die fehlende Prozessentwicklung und IT-Unterstützung in der Bearbeitung und Auswertung begründet ist.

Die manuelle Identifizierung der BEM-Fälle, das sind alle Beschäftigten, die mehr als 42 Kalendertage in den vergangenen 365 Tagen arbeitsunfähig waren, bezieht sich derzeit ausschließlich auf die Langzeiterkrankten. Die Selektion der Fälle durch Mehrfacherkrankung, das sind jene, bei denen sich Einzelerkrankungen auf 42 Tage aufsummieren, findet derzeit noch nicht statt. Aus diesem Grund kann es vorkommen, dass berechnete Personen übersehen werden und diese Leistung des Unternehmens nicht angeboten bekommen. Sofern jedoch der Mitarbeiter Kenntnis vom BEM hat und die Kriterien erfüllt, kann er sich selbst in Abstimmung mit seiner Führungskraft für das BEM anmelden.

Die Dokumentation und Auswertung aller im BEM anfallenden Daten wird derzeit manuell in Excel realisiert. Ein automatisierter Zugriff auf benötigte Personaldaten aus SAP HR mit einer zusätzlichen Pflegemöglichkeit aller im BEM anfallenden Daten besteht nicht.

### 5.2.3 Soll-Zustand

Im standardisierten Prozess, welcher durch ein IT-System unterstützt wird, sollen mehrere Aspekte berücksichtigt werden. Zum einen soll die Fürsorgepflicht des Arbeitgebers sichergestellt sein. Dies ist beispielsweise dadurch gekennzeichnet, dass keine Beschäftigten durch das Raster der Fallidentifikation fallen sollen. Weiterhin muss die Einhaltung der datenschutzrechtlichen Richtlinien stets gesichert sein. Die Prozessdurchführung soll einen hohen Automatisierungsgrad erreichen, die verschiedenen Prozessrollen berücksichtigen und ohne Medienbrüche stattfinden, um die Prozessqualität positiv zu beeinflussen. Der hierfür entworfene BPMN-Prozess ist im Anhang II und auf dem beigelegten Datenträger einsehbar. Als Ergebnis sollen sich die Mitarbeiter des BEM stärker auf ihre Kernaufgaben konzentrieren können und somit ihre Produktivität weiter erhöhen.

Ein besonderer Fokus liegt auf dem Datenschutz und dem Berechtigungskonzept, um die verschiedenen Zugriffsebenen sicherzustellen. Es soll beispielsweise eine strikte Trennung von den Personaldaten im SAP HR und BEM-Daten berücksichtigt werden und nach dem Ablauf einer Frist von drei Jahren sollen die Daten zum BEM-Vorgang anonymisiert werden, indem der Personenbezug entfernt wird.

Das entworfene Rollenkonzept sieht vor, dass der Informationsumfang, der zu BEM-Maßnahmen eingesehen werden kann, stark von der Rolle der Person abhängt. Basierend auf den Zugriffsebenen aus Abbildung 5-1: Zugriffsebenen auf Daten einer BEM-Maßnahme ergeben sich die folgenden Datenzugriffsbeschränkungen.

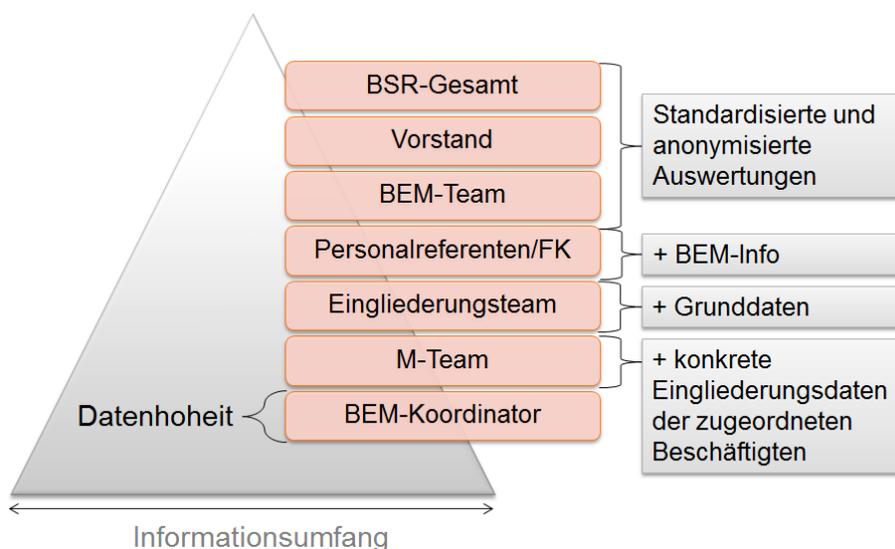


Abbildung 5-2: Informationsumfang in Abhängigkeit der Nutzerrolle

Quelle: BSR (2012), S. 3.

Die für die Personalreferenten relevanten Informationen werden in einem kunden-eigenen Infotypen im HR-System abgespeichert. Somit erhalten die Personalreferenten keinen Zugriff auf das BEM-System sondern lediglich einen Zugriff auf die relevanten Informationen innerhalb des HR-Systems. Innerhalb des BEM-Systems stehen die für die Durchführung des BEM erforderlichen Personaldaten zum konkreten BEM-Fall zur Verfügung.

### **5.3 Lösungsansatz**

Das Projekt des betrieblichen Eingliederungsmanagements wurde bei der BSR als Referenzszenario zur Evaluation der Möglichkeiten des NetWeaver CE ausgewählt. Der Prozess ist daher prädestiniert, weil es sich um einen kollaborativen Geschäftsprozess handelt. Dieser ist dadurch gekennzeichnet, dass mehrere Personengruppen beteiligt sind, die zum Teil nur kleine, in sich geschlossene Aufgaben übernehmen. Weiterhin soll die Speicherung der Daten zum BEM außerhalb des SAP HCM erfolgen, um eine systemseitige Trennung von Personalstammdaten und den sensiblen personenbezogenen Daten des BEMs zu gewährleisten.

Somit ist das Composite Application Framework, welches die Service-Schicht repräsentiert, für die Persistenz zuständig. Hierfür werden mehrere Business Objekte für am Prozess beteiligte Entitäten angelegt. Weiterhin werden an dieser Stelle auch externe Services angebunden. Diese werden beispielsweise zur Nutzung des Archiv-Systems zur Ablage von erstellten oder gescannten Dokumenten zum BEM-Fall benötigt. Weiterhin werden Services implementiert, welche die benötigten Informationen zur Person aus dem SAP HCM bereitstellen. Ziel soll es sein, dass auf lokale und externe Services in der Prozess- und UI-Schicht auf einheitliche Weise zugegriffen wird. Die Business Objekte und externen Services werden unter einem gemeinsamen Applikationsservice zusammengefasst. Somit sollen alle Funktionalitäten des BEM innerhalb einer Servicedefinition, welche in der Service Registry bekannt gemacht wird, für Fremdsysteme nutzbar sein.

In der Prozessschicht wird auf SAP BPM gesetzt, da es sich beim BEM um einen User-zentrierten Geschäftsprozess handelt, bei dem die Anwendungsintegration ebenfalls stark im Fokus steht. Konzeptionell wird darauf geachtet, dass der Prozess sowie die Datenhaltung innerhalb des Prozesses schlank bleiben. Wann immer möglich, sollen nur Schlüsselfelder im Prozesskontext gespeichert werden.

Aufgrund des bereits erwähnten umfangreichen ABAP Know-hows bei der Berliner Stadtreinigung wird die UI-Schicht nicht innerhalb des NetWeaver CE entwickelt. Die Oberflächen werden stattdessen mit Web Dynpro ABAP innerhalb des SAP

ERPs entwickelt. Grundsätzlich werden dabei zwei Arten von Oberflächen für dieses Projekt unterschieden. Zum einem sind das Oberflächen, die durch den BPM Prozess aufgerufen werden und zur Laufzeit im zentralen Arbeitsvorrat des Anwenders innerhalb des SAP Portals erscheinen. Diese Oberflächen können demnach nur aufgerufen werden, wenn zu diesem Zeitpunkt eine konkrete Aufgabe für den Anwender vorliegt. Diese Web Dynpro Komponenten werden mit Hilfe von Web Dynpro CHIPs für die Verwendung innerhalb eines BPM Prozesses zur Verfügung gestellt.<sup>158</sup> Zum anderen gibt es UI-Komponenten, welche jederzeit durch den Anwender aufgerufen werden können, um Prozesse zu starten oder fortzuführen, welche sich in einem Wartestatus befinden, da sie auf Events von außerhalb warten (Intermediate Message Event). Dies ist beispielsweise innerhalb des BEM-Prozesses der Fall, wenn auf die postalische Antwort des Beschäftigten über die Teilnahme am BEM-Verfahren gewartet wird. Diese zweite Art von UI-Komponenten werden im SAP Portal als einfache Web Dynpro ABAP-Anwendungen eingebunden, welche den entsprechenden Portalrollen zugeordnet werden. Somit sind alle Benutzermasken des BEM über das Portal erreichbar.

#### **5.4 Realisierung**

Um die Bedürfnisse der Anwender bei der Umsetzung des Projektes möglichst gut zu berücksichtigen, wurde festgelegt, dass der Fachbereich möglichst früh mit dem System arbeiten soll. Daher wurde das Projekt für die Realisierung in vier Phasen gegliedert, welche jeweils nach Fertigstellung dem Kunden übergeben werden. In der ersten Phase wird die Fallidentifikation im SAP HCM automatisiert, diese Anforderung wird als SAP Business Workflow umgesetzt und durch eine Ereigniskopplung beim Anlegen einer Abwesenheit gestartet. Zum Ende dieser Phase wird der BEM-Koordinator mit einer E-Mail über neue Mitarbeiter informiert, welche die Kriterien eines BEM-Falls erfüllen. Zu diesem Zeitpunkt kommt der NetWeaver CE noch nicht zum Einsatz. In den drauffolgenden Ausbaustufen wird stattdessen ein BPM-Prozess im NetWeaver CE gestartet.

Die Phase zwei bildet den Prozess der Fallklärung ab. Dies umfasst das Erstellen von Anschreiben zur Angebotsunterbreitung, das Erfassen der Antwort sowie die Durchführung eines telefonischen Vorabgesprächs mit dem Beschäftigten. Die dritte Phase ergänzt die Maßnahmenvorbereitung, -durchführung und -nachbereitung. Die vierte Phase komplettiert das Szenario mit zusätzlichen Statistik- und Auswertungsfunktionen im SAP BW.

---

<sup>158</sup> SAP Help - NetWeaver 7.3EHP1 (2012), (siehe Internetverzeichnis).

Für den Inhalt dieser Masterthesis sind insbesondere die Phase zwei und drei entscheidend, da dort die Entwicklung innerhalb des NetWeaver CE erfolgt. In der vierten Phase ist die Composition Environment nur für das Bereitstellen von BW-Extraktoren verantwortlich. Für die beiden entscheidenden Realisierungsphasen soll jedoch im Rahmen dieser Arbeit keine detaillierte Dokumentation der Entwicklungs- bzw. Modellierungsarbeit erfolgen, vielmehr sollen das grundsätzliche Vorgehen beschrieben, interessante Erkenntnisse aufgegriffen und die Gangbarkeit des gewählten Architekturansatzes beschrieben werden.

Um sicherzustellen, dass die Bedürfnisse des Fachbereiches umgesetzt werden, erfolgte die Entwicklung in enger Abstimmung mit Vertretern des betrieblichen Eingliederungsmanagements. Bei den Besprechungen innerhalb der Projektgruppe wurde von den grafischen Notationen der verwendeten Technologien profitiert. Insbesondere die visuelle Darstellung des Prozessablaufes, wie beispielsweise in Abbildung 5-3 dargestellt, stellte eine gemeinsame Kommunikationsbasis sicher und besprochene Prozessänderungen konnten wann immer möglich unmittelbar innerhalb des Modells dokumentiert werden. Auch bei der Anpassung der Benutzeroberfläche ist das direkte Einholen von Feedback der späteren Anwender möglich und vermeidet somit unnötige Abstimmungsschleifen.

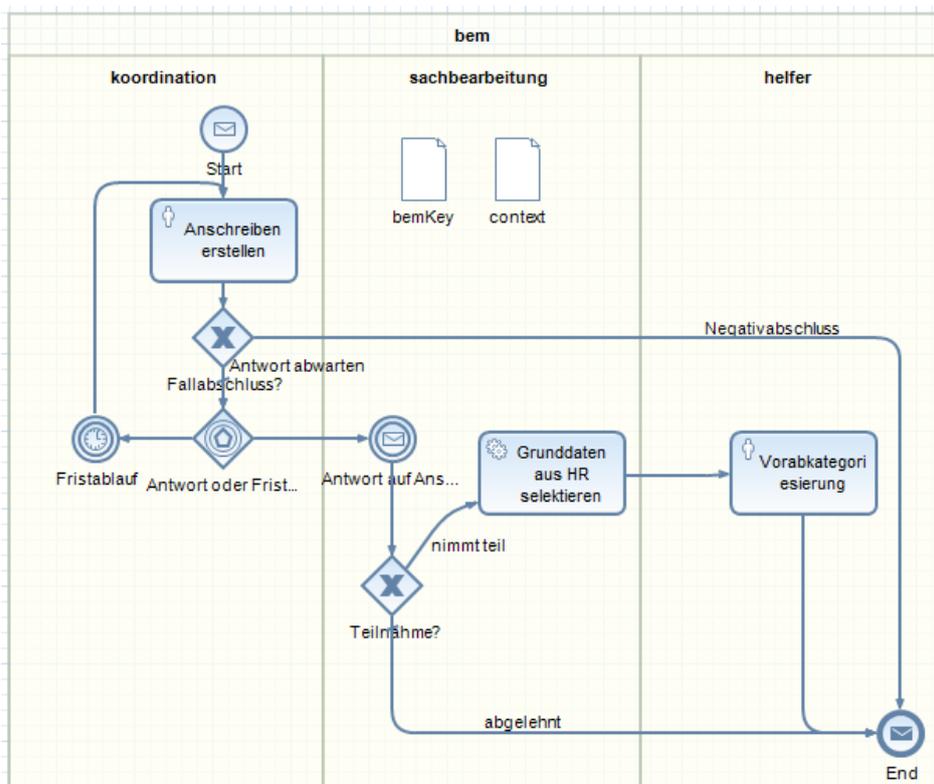


Abbildung 5-3: BPM-Prozess zur Fallklärung in Phase 2

Quelle: Eigene Darstellung.

Bei der Modellierung von Services wurde grundsätzlich nach dem Outside-In Ansatz vorgegangen. Dieses Vorgehen besagt, dass eine Schnittstellenbeschreibung vor der eigentlichen Implementierung der Funktionalität erstellt wird. Dies hat den Vorteil einer erhöhten konzeptionellen Sicht auf das System. Weiterhin können bereits definierte Services vorerst mit Platzhaltern versehen werden, damit sie ihr späteres Verhalten simulieren. Diese sogenannten „Mock-Services“ ermöglichen ein frühzeitiges Testen des gesamten Prozesses und der einzelnen Komponenten untereinander.<sup>159</sup>

Die Kapselung der Business Objekte, wie sie im Abschnitt 5.3 (Lösungsansatz) beschrieben wurden, stellt sich als hilfreich heraus, da zum einen nur tatsächlich benötigte Service-Methoden bereitgestellt werden und Serviceaufrufe den Gegebenheiten angepasst werden können, um sie effizienter einzusetzen. So benötigt man beispielsweise für die Aktion „Anlegen und Daten speichern“ mit den generierten CRUD-Methoden der Business Objekte stets zwei Serviceaufrufe. In einem ersten Schritt wird die create-Methode aufgerufen, welche jedoch nur Pflichtfelder als Übergabeparameter zulässt. In einem zweiten Aufruf können durch die update-Methode auch optionale Datenfelder übergeben werden. Daher wurde innerhalb des Applikationsservices eine create-Methode implementiert, welche die beiden Funktionen des Business Objekts vereint. Der Verwender des Services muss somit nur eine Methode aufrufen.

Während der Implementierung traten zahlreiche technische Komplikationen auf, welche hauptsächlich auf die erste Verwendung der Technologien zurückzuführen sind. Diverse Konfigurationen mussten vorgenommen und Berechtigungen zugewiesen werden. Diese hat die SAP-Basisbetreuung in einer CE-Entwicklerrolle zusammengefasst, sodass Folgeprojekte deutlich weniger Komplikationen erfahren werden. In der klassischen SAP-Entwicklung erfolgt die Programmierung innerhalb der ABAP Workbench, diese ist integraler Bestandteil des ABAP-Applikationsservers und daher stets versionsgleich zum Applikationsserver vorhanden. Beim NetWeaver CE ist die Entwicklungsumgebung hingegen vom Applikationsserver getrennt. Die Erfahrung zeigt, dass auf identische Versionsstände bis auf Ebene der Support Packages zwischen Applikationsserver und dem NetWeaver Developer Studio achten ist, anderenfalls kann es zu Inkompatibilitäten kommen. Bei der Umsetzung des betrieblichen Eingliederungsmanagements wurde dies bei der Zuordnung von Bearbeitern eines Prozessschrittes bemerkt. Der Aufruf der User Management Engine (UME) war aufgrund von verschiedenen

---

<sup>159</sup> Baresi, Luciano / Di Nitto, Elisabetta (2007), S. 154f.

Versionsständen nicht möglich. Ein Aktualisieren der Entwicklungsumgebung erzeugte Folgefehler aufgrund verschiedener Versionsstände zwischen bestehenden Projekten und neuen Projektbibliotheken. Diese konnten jedoch durch Synchronisieren der Abhängigkeiten innerhalb der Software-Komponente behoben werden, jedoch wirft dies die Frage auf, ob die Wartung eines Projektes über mehrere Versionsstände hinweg stets problemlos möglich ist.

**Anschreiben erstellen: PLMB Referentin (80109983)** Aktionen  
Ansicht

---

**Aufgabendaten**

Fällig am: <Kein Fälligkeitsdatum>    Status: Reserviert    Anhänge: 0    Prozess: [BEM](#)  
 Verantwortlicher: Lucke, Tobias    Priorität: Mittel    Notizen:

---

**Aufgabenanwendung**

**Falldaten**

Erstanschreiben erstellen    Zweitanschreiben erstellen    Zwischenspeichern    Negativabschluss    Erledigt

**Daten zur Person**

Anrede: Frau  
 Vorname: PLMB  
 Nachname: Referentin  
 Personalmr.: 80109983

**Organisatorische Zuordnung**

Vertretungsbereich: Hauptverwaltung  
 Geschäftseinheit: Personal  
 Organisationseinheit: OE: FK, VM, VMW, VR11/12  
 Taetigkeit: Personalreferent(in)

**Informationen zum Fall**

Fallart: Langzeiterkrankung  
 Meldedatum: 24.08.2012  
 Fall-Schlüssel: bb369020-edb8-11e1-a3a2-000025b907ce  
 Status: Erstanschreiben erstellt  
 Anzahl BEM-Verfahren: 2  
 Zeiträume: 22.08.2004 - 31.12.2004; 22.08.2007 - 10.10.2008;

**Bearbeitungshinweise**

aus HRE (Entwicklungssystem)

**Adressdaten**

Straße/Hausnr.: Waldweg 2  
 PLZ / Ort: 11111 Budapest

---

**Anschreiben**

Art des Anschreibens	Erstellungsdatum	Antwortdatum	Angelegt von	Zeitstempel Anlage	Zuletzt bearbeitet von	Zeitstempel letzte Änderung	
Erstanschreiben	24.08.2012		Guest	2012-08-24T09:00:41.912+02:00	Guest	2012-08-24T09:00:41.916+02:00	Anzeigen

Abbildung 5-4: Web Dynpro ABAP Oberfläche zur Anschreibenerstellung

Quelle: Eigene Darstellung.

Die Verwendung von Web Dynpro ABAP Komponenten für die Benutzeroberfläche hat bis auf wenige Einschränkungen gut funktioniert. Das Anzeigen der CHIPs bedarf zusätzlicher Berechtigungen, d.h. sowohl Entwickler- als auch Anwenderrollen müssen angepasst werden. Weiterhin ist eine Verbindung zwischen Portal und Quellsystem der Web Dynpro Komponente notwendig. Dies ist jedoch ein einmaliger Konfigurationsaufwand, der nicht stark ins Gewicht fällt. Weiterhin müssen Anwender einen namensgleichen User im ABAP-System besitzen und das Single-Sign-On sollte funktionieren, damit keine erneute Anmeldung beim Aufruf der Benutzeroberfläche notwendig ist. Diese Voraussetzungen sind bei der BSR erfüllt. Lediglich die Anforderung, dass Benutzeroberflächen mit dem Floorplan Manager für Web Dynpro ABAP umgesetzt werden sollen, konnte nicht reali-

siert werden. Hierfür war ein zu niedriger Versionsstand des ABAP-Systems ausschlaggebend, da der SAP NetWeaver 7.0 EhP 2 im Einsatz ist und 7.0 EhP 3 notwendig ist.<sup>160</sup>

Zusammenfassend ist zu sagen, dass das Zusammenspiel zwischen den Technologieschichten gut funktioniert. Da jedoch nur Software eines Herstellers innerhalb eines Applikationsservers und einer Entwicklungsumgebung verwendet wurde, erscheint das Zusammenspiel zwischen den Komponenten an manchen Stellen komplizierter als es sein müsste. Es wäre beispielsweise denkbar, dass Business Objekte als Variablen im Prozess instanziiert und beim Aufruf in der UI-Schicht als Referenz übergeben werden. Dies widerspricht jedoch dem Servicegedanken einer losen Kopplung der Komponenten, würde allerdings die Entwicklungszeiten weiter verkürzen und den Entwicklungsprozess an sich vereinfachen. Andererseits ist durch die standardisierten Webservice-Schnittstellen ein problemloser Austausch von Komponenten möglich, was bereits an der Historie verschiedener Technologien des NetWeaver CE sichtbar ist. So wurden beispielsweise Guided Procedures zugunsten von BPM ersetzt, trotzdem mussten keine Änderungen an Business Objekten vorgenommen werden, um sie weiterhin verwenden zu können. Der Webservice-Standard wird aller Wahrscheinlichkeit in den nächsten Jahren weiterhin ein zukunftssicherer Standard bleiben, sodass der strategische Kurs der SAP nachvollziehbar ist.

---

<sup>160</sup> SAP Community Network (2011), S. 258ff.

## 6 Auswirkungen auf den Entwicklungsprozess

### 6.1 Organisation der Entwicklungsarbeit

Der Alltag in der traditionellen Softwareentwicklung ist oftmals dadurch gekennzeichnet, dass Verständnisbarrieren zwischen der IT und den Fachbereichen herrschen. Während der Analysephase wird daher viel Aufwand in die exakte Erfassung von Anforderungen investiert und somit versucht, ein gemeinsames Verständnis für die Problemstellung zu entwickeln. Neben der textuellen Beschreibung werden in dieser Phase auch Diagramme und Prototypen von Oberflächen erstellt, welche als Diskussionsgrundlagen dienen. Mit der eigentlichen Implementierung wird schließlich erst begonnen, nachdem die Analysephase abgeschlossen ist. Für die Entwickler bedeutet dies, dass Modelle mit den Mitteln der Programmiersprache umgesetzt werden müssen. Während dieses Schrittes kann es potentiell jedoch erneut zu Verständnisfehlern kommen, da häufig keine standardisierten Notationen bzw. die Elemente der Notation nicht regelkonform eingesetzt werden.

Während der Umsetzung des Praxisbeispiels hat sich gezeigt, dass die Modellierungswerkzeuge des SAP NetWeaver CE ausreichend flexibel und bedienbar sind, um sie während Besprechungen als Dokumentationswerkzeug zu verwenden. Dies hat entscheidende Vorteile für das gemeinsame Verständnis. Einerseits ist man durch den technischen Rahmen und der Modellvalidierung stets dazu angehalten exakt zu modellieren und andererseits fallen Besprechungsprotokolle weniger umfangreich aus, da viele Entscheidungen direkt umgesetzt werden können. Hierbei hat die Praxis gezeigt, dass eine vollständige Modellierung innerhalb der Analysegespräche nicht das Ziel sein sollte, vielmehr dienen sie der grundsätzlichen Erfassung und sollte während der Nachbereitung um technische Parameter ergänzt werden. Weiterhin hat sich verdeutlicht, dass insbesondere die BPMN-Notation des SAP BPM für den Fachbereich verständlich ist und die daraus entstandenen Ergänzungen hilfreich für die IT sind. Auch das gemeinsame Gestalten von Benutzeroberflächen verlief problemlos, da durch die exakte Vorschau und gute Bedienbarkeit des Web Dynpro Modellierungswerkzeuges Anforderungen schnell umsetzen werden können. Das gemeinsame Modellieren des Datenmodells mithilfe des Composite Application Framework, erwies sich als weniger praktikabel, stattdessen waren die Oberflächenentwürfe sowie die derzeitig verwendete Excel-Tabelle (vgl. 5.2.2 Ist-Zustand) als Input für diese Aufgabe hilfreicher. In der Realisierung des betrieblichen Eingliederungsmanagements war zu beobach-

ten, dass sich die Kommunikation stark am Modell orientierte und es daher wenig Missverständnisse zwischen der IT und dem Fachbereich gab.

Es kann daher festgehalten werden, dass während der Analysephase ein enges Zusammenarbeiten zwischen IT und Fachabteilungen gefördert wird. Die Arbeitsergebnisse der Analyse bilden die Grundlage der Implementierung, diese können direkt verwendet werden und müssen kein weiteres Mal manuell transformiert werden. Hinsichtlich der Anforderungen an den Entwickler ergeben sich Veränderungen, er sollte neben guten Kommunikationsfähigkeiten und einer schnellen Auffassungsgabe auch die Modellierungsnotation verinnerlicht haben. Hinsichtlich dieser zusätzlichen Anforderungen stellt sich die Frage, inwiefern sich der klassische Entwickler zum Berater wandeln muss, um diesen Anforderungen gerecht zu werden. Aus der Erfahrung heraus ist diese Frage nicht eindeutig zu beantworten, da bereits in der Vergangenheit der Trend zu beobachten war, dass Entwickler maßgeblich an der Erstellung von Konzepten beteiligt sind und somit auch die zuvor genannten Fähigkeiten beherrschen müssen. Andererseits ist in der modellgetriebenen Softwareentwicklung der Anteil an diesen beratenden Tätigkeiten höher als in der klassischen Softwareentwicklung. So dass schon von einer Entwicklung vom Techniker zum Berater auszugehen ist und somit eine höhere Konzentration auf fachliche Aspekte beobachtet werden kann. Trotzdem werden weiterhin Entwickler benötigt, die fundierte Programmierkenntnisse besitzen, da viele Artefakte eines Softwaresystems manuell codiert werden müssen. Als Beispiele sind die Implementierung von Services in Backend-Systemen, die Erstellung von anspruchsvollen Benutzeroberflächen sowie das Entwickeln von komplexen Routinen zu nennen. Technologiekenntnisse erleichtern zudem das Verständnis des Modellierungswerkzeugs, da technische Begriffe oftmals auch in der Modellierungssprache verwendet werden. Ein Beispiel hierfür ist die Verwendung des Begriffs „colspan“ in Web Dynpro für das Verbinden von Spalten innerhalb eines Tabellenlayouts, welches typisch für die Entwicklung von HTML Oberflächen ist.

Die Fehlersuche innerhalb von Softwaresystemen geschieht in der modellgetriebene Entwicklung anders als bei traditioneller entwickelter Software. In der klassischen Softwareentwicklung wird bei Fehlermeldungen oftmals ausschließlich der Quellcode debuggt und anschließend direkt angepasst, ohne den Sachverhalt gegen die ursprünglichen Anforderungen zu prüfen. Da in der modellgetriebenen Entwicklung das Modell jedoch stets aktuell ist, bildet es auch den Einstiegspunkt einer Fehlersuche. Dieser Einstieg gewährt dem Entwickler einen höheren Abstraktionsgrad und hilft somit passende Lösungsalternativen abzuwägen. Die Fehlersuche wird zusätzlich durch den hohen Standardisierungsgrad begünstigt, da

Fehlerkonstellationen zur Laufzeit häufig automatisch im globalen Systemlog protokolliert werden.

Neben den Änderungen am Entwicklungsprozess, die der modellgetriebenen Entwicklung zuzuordnen sind, ist ein weiterer Faktor für den NetWeaver CE entscheidend. Da SAP auf eine Java-basierte Entwicklungsumgebung setzt, hat dies Einfluss auf das gesamte Application Lifecycle Management. Verglichen mit der ABAP Entwicklungsumgebung und ihrem integrierten Transportmanagement fallen mehrere Defizite auf. Wesentlich ist zum einen, dass die Java-Programmierung auf dem lokalen PC des Entwicklers erfolgt. Demnach liegt das gesamte Entwicklungsprojekt einzig auf einem Client-Rechner bis eine Sicherung erfolgt bzw. das Software-Repository aktualisiert wird. Das Speichern des Projekts auf einem Netzlaufwerk wird vom NetWeaver Developer Studio nicht unterstützt. Auch der Begriff des Deployment eröffnet ein neues Vorgehen für ABAP-Entwickler, da dort keine Übertragung des in Bytecode übersetzten Quelltextes an den Applikationsserver notwendig ist. Dieser Sachverhalt erschwert auch das Debuggen von Anwendungen, da zu kontrollierende Softwarekomponenten zuvor in das NWDS geladen, ein Breakpoint gesetzt und anschließend auf dem im Debug-Modus gestarteten Applikationsserver verbunden werden muss. Gerade in Produktivsystemen ist daher die Analyse von Fehlern erheblich schwieriger, als man es aus ABAP-Systemen gewohnt ist.

Im Rahmen des Projektes „Betriebliches Eingliederungsmanagement“ war die NetWeaver Development Infrastructure (NWDI) noch nicht einsatzbereit, sodass die lokal entwickelten Entwicklungskomponenten manuell auf den entsprechenden Applikationsserver übertragen wurden. Dies soll jedoch in den nächsten Monaten geändert werden, damit ein vollwertiges Change-Management realisiert werden kann. Sobald die NWDI im Einsatz ist, wird die Quellcodeverwaltung durch diese zentrale Komponente übernommen, sodass keine manuellen Backups von Entwicklungsprojekten durchgeführt werden müssen und die Produktivsetzung von Softwarekomponenten in einem koordinierten Prozess mit Anschluss an das Ticketsystem erfolgt.

## **6.2 Qualität und Flexibilität**

Die Erstellung von Anwendungen mithilfe von Modellierungswerkzeugen gibt ein Portfolio von Elementen vor, aus denen eine Softwarekomponente zusammengesetzt werden kann. Daher ist die Erkenntnis nicht verwunderlich, dass eine geringere Qualitätsstreuung bei der erstellten Software zu beobachten ist. Die standar-

disierten Lösungsansätze für spezifische Problemstellungen sind im NetWeaver CE für die Belange einer Geschäftsanwendung nützlich, da verfügbare Komponenten stets einer gewissen Qualität genügen und darüber hinaus zusätzlichen Funktionsumfang bereitstellen. Dies ist von der grundsätzlichen Architektur bis hin zu einzelnen Elementen wie einer Wertehilfe für ein Datumsfeld zu beobachten. Grundsätzlich profitiert man bei der Entwicklung von Geschäftsprozessen mit dem NetWeaver CE besonders von der Ausführungsumgebung und vom integrierten Monitoring. Dem Prozessadministrator ist jederzeit ersichtlich, welche Prozessinstanzen laufen, in welchem Status sich diese befinden und es können Laufzeitdaten zum Prozess betrachtet werden. Diese Funktionen stehen im SAP NetWeaver Administrator bereit.

The screenshot displays the 'BPM-Statusübersicht' (BPM Status Overview) page in the SAP NetWeaver Administrator. The page is divided into several sections:

- Mein Arbeitsbereich** (My Work Area): Includes 'Systemübersicht', 'Ressourcenüberwachung', 'Prozessüberwachung', and 'BPM-Statusübersicht'.
- Übersicht** (Overview):
  - BPM Systemstatus**: Shows 'Verfügbarkeit Prozessserver' (Availability Process Server) with a green status and 'Aktualisiert am 07/27/2012 8:32:31 AM MESZ'. Below it, 'Verfügbarkeit Java-Anwendungen' (Availability Java Applications) is shown with a green status and a link to 'Zur Java-Start-Stop-Anwendung'. Further down, 'Engine-Test' is shown with a diamond icon and a link to 'Zur BPM-Konfigurationsvorlage'.
  - Prozesse** (Processes): A table showing the number of process instances and their status:
 

Anzahl der Prozesse	
Prozessinstanzen insgesamt	( 10 )
Fehler	( 1 )
Fehlgeschlagen	( 0 )
Angehalten	( 0 )
Abgeschlossen	( 6 )
Abgebrochen	( 0 )
In Bearbeitung	( 4 )
  - Aufgaben** (Tasks): A table showing the number of task instances and their status:
 

Anzahl der Aufgaben	
Aufgabeninstanzen insgesamt	( 18 )
Fehler	( 0 )
Angehalten	( 1 )
Abgeschlossen	( 13 )
Abgebrochen	( 1 )
Bereit	( 4 )
In Bearbeitung	( 0 )

Abbildung 6-1: BPM-Statusübersicht

Quelle: Eigene Darstellung

Des Weiteren ist die Integration in das SAP Portal aus Sicht der Einheitlichkeit von Softwarelösungen positiv zu bewerten, da die Universal Worklist (UWL) als zentraler Arbeitsvorrat für den Anwender prozessübergreifend zu erledigende Aufgaben verwaltet und somit zu einem ähnlich wichtigen Arbeitsmittel wird, wie es das E-Mail Postfach ist. Weitere Beispiele nützlicher Funktionen, welche nicht durch den Entwickler explizit entwickelt werden müssen, sind die Möglichkeit, Notizen und Dokumente als Bearbeitungshinweise zum Prozess abzulegen, E-Mail Benach-

richtigungen zu versenden sobald neue Arbeitsaufgaben eintreffen oder das Definieren von Vertretungsregeln.

In der Oberflächengestaltung ist zu erkennen, dass die Modellierungswerkzeuge bestrebt sind, ein einheitliches Look & Feel durchzusetzen. Diese haben stets ein SAP Design, daher eignen sich Composite Applications prinzipiell eher für unternehmensinterne Anwendungsfälle, jedoch sind UI-Elemente über verschiedene SAP Technologien hinweg einheitlich. So ist es selbst für erfahrene Anwender nicht ersichtlich, ob eine Oberfläche mit dem Visual Composer, Web Dynpro Java oder Web Dynpro ABAP erstellt wurde.

Besonders kreative und anforderungsspezifische Lösungen sind jedoch aufgrund des höheren Standardisierungsgrades kaum möglich. Umfangreiche Planungsübersichten, wie sie beispielsweise im Projekt „Personaleinsatzplanung“ bei der Berliner Stadtreinigung gefordert waren, lassen sich nicht mit den in dieser Arbeit vorgestellten Oberflächentechnologien umsetzen.

Vertretungsart: 02 - Schichtvertr. 11 - Winterdienst													Januar 2011																											
Gruppen	KT	OL	BK	FT	W	FF	FPS	SPS	NPS	GU	VU	JZK	52. KW							01. KW							02. KW							03. KW						
													Sa 01	So 02	Mo 03	Di 04	Mi 05	Do 06	Fr 07	Sa 08	So 09	Mo 10	Di 11	Mi 12	Do 13	Fr 14	Sa 15	So 16	Mo 17	Di 18	Mi 19	Do 20	Fr 21	Sa 22	So 23	Mo 24	Di 25			
01_GRUPPE	KT	OL	BK	FT	W	FF	FPS	SPS	NPS	GU	VU	JZK	2	4	1	1	1	1	1	2	3	2	2	2	2	2	0	1	2	1	2	2	1	6	0	1	1			
02_GRUPPE	KT	OL	BK	FT	W	FF	FPS	SPS	NPS	GU	VU	JZK	4	3	2	2	2	2	1	2	5	2	2	1	2	1	1	1	1	1	1	1	1	0	0	1	1			
Kranke, Frank	144	-	-	1	9	9	9	0	0	28	30	-	-	FW	F	F	F	F	F	-	-	F	F	F	F	F	PSA	-	F	F	F	F	F	-	-	F	F			
Kranke, Michael	167	-	-	2	7	7	9	0	0	29	30	-	-	-	S	S	S	S	S	-	FW	F	F	FF	F	F	PSA	-	S	S	S	S	S	-	-	F	F			
London, Norbert	200	-	-	1	8	8	8	0	0	31	30	-	-	FW	F	F	F	F	FF	FPS	-	S	S	S	S	U	-	U	U	U	U	U	PSU	-	-	U	U			
Messner, Ralf	63	-	-	2	8	8	9	0	0	15	30	-	-	-	F	F	F	F	F	-	FW	F	F	F	F	F	PSA	-	F	F	F	F	F	-	-	F	F			
Messner, Jens	-	-	-	2	8	8	8	0	0	30	30	-	-	FW	F	F	F	F	F	-	-	F	F	F	F	F	PSA	-	F	F	F	F	F	-	-	F	F			
Messner, Udo	97	-	-	1	9	9	7	0	0	26	30	-	-	F	-	F	F	F	F	FPS	-	F	F	F	F	S	-	FW	F	F	F	F	F	PSA	-	-	S	S		
Messner, Frank	-	-	-	1	9	9	6	0	0	32	30	-	-	F	-	F	F	F	F	-	FW	F	F	F	F	F	PSA	-	F	F	F	F	-	-	F	F				
Messner, Frank	-	-	-	2	8	8	7	0	0	32	30	-	-	F	-	F	F	F	F	-	FW	F	F	F	F	F	PSA	-	F	F	F	F	-	-	F	F				
Messner, Udo	-	-	-	2	8	8	7	0	0	38	35	-	-	F	-	F	F	F	F	-	FW	F	F	F	F	F	PSA	-	F	F	F	F	-	-	F	F				
03_GRUPPE	KT	OL	BK	FT	W	FF	FPS	SPS	NPS	GU	VU	JZK	2	3	-2	-1	-1	-1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2	0	-1			
04_GRUPPE	KT	OL	BK	FT	W	FF	FPS	SPS	NPS	GU	VU	JZK	1	0	1	1	1	1	1	1	0	2	2	2	2	2	8	0	2	2	2	2	2	1	4	2	2			
05_GRUPPE	KT	OL	BK	FT	W	FF	FPS	SPS	NPS	GU	VU	JZK	1	0	1	2	2	2	2	8	0	2	2	2	2	2	0	4	2	2	2	2	2	4	0	2	2			

Abbildung 6-2: Planungsübersicht der BSR Personaleinsatzplanung

Quelle: Eigene Darstellung

Im Rahmen dieses Projektes wurden Prototypen in Web Dynpro angelegt, welche schnell zeigten, dass ein ähnliches Planungsraster als Tabelle bereits bei ca. 100 Mitarbeitern und 365 Tagen (=Spalten) Antwortzeiten im Bereich von einer Minute verursacht, die allein dem generischen Web Dynpro Ansatz zuzuordnen sind. Die Umsetzung mit Hilfe von SAP Business Server Pages (BSP) erforderte zwar eine vollständig manuelle Implementierung, es konnten jedoch alle Anforderungen in der geforderten Qualität mit sehr guten Antwortzeiten umgesetzt werden.

Die Verfügbarkeit einer Composite Application ist kritisch zu betrachten, da mehrere Systeme beteiligt sind. Angenommen man hätte das betriebliche Eingliederungsmanagement im ERP entwickelt, da die Anforderung besteht, dass BEM-

Daten und Personaldaten in getrennten Systemen gehalten werden sollen, wären zwei Systeme involviert. Bei einer Verfügbarkeit eines Einzelsystems von 99% ergibt sich die Gesamtverfügbarkeit aus dem Produkt der Einzelzuverlässigkeiten, demnach also  $99\% * 99\% = 98,01\%$ . Da bei der tatsächlichen Umsetzung drei Systeme (NetWeaver CE, HCM, ERP für Web Dynpro ABAP CHIPs) beteiligt sind, ergibt sich eine Verfügbarkeit von  $0,99\%^3 = 97,03\%$ . Wäre das SAP Portal ein dediziertes System, hätte man bereits eine Verfügbarkeit von 96,06%, was doppelt so häufige Ausfälle, verglichen mit der Zwei-System-Variante, bedeuten würde. Demnach ist also die Anzahl der involvierten Systeme für dieses Qualitätskriterium entscheidend.<sup>161</sup>

### 6.3 Kosten

Eine kaufmännische Betrachtung des Projektes zum betrieblichen Eingliederungsmanagement im Vergleich mit vergangenen Softwareprojekten sollte aus mehreren Gründen sehr behutsam durchgeführt werden. Zum einen trägt das in dieser Arbeit durchgeführte Projekt den Charakter eines Lernprojekts, da stets mehrere Lösungsvarianten evaluiert wurden, bevor sich für einen konkreten Lösungsansatz entschieden wurde. Das Vergleichen mit den langjährigen Best-Practices traditioneller Softwareentwicklung verursacht zusätzlichen Aufwand. Weiterhin mussten mehrfach Einstellungen am NetWeaver CE durchgeführt werden, wovon zukünftige Composite Applications profitieren.

Trotzdem soll ein Vergleich mit einem ähnlichen Projekt bei der BSR hergestellt werden. Das Projekt zur „Online-Anmeldung Veranstaltungsmanagement“ wird für diesen Vergleich herangezogen, da es grundsätzlich ähnliche Merkmale aufweist, welche miteinander verglichen werden können. In beiden Anwendungsszenarien handelt es sich um einen kollaborativen Geschäftsprozess mit Weboberflächen für die User-Interaktion. Die Realisierung des Prozesses zum Veranstaltungsmanagement beinhaltet das Buchen von Fort- und Weiterbildungen über das Unternehmensintranet und wurde technologisch mit SAP Business Server Pages (BSP) und einem SAP Business Workflow realisiert. Hierfür sollen die Entwicklungsaufwände entsprechend der Architekturebenen gegenübergestellt werden.

#### *Service-Schicht*

Die Arbeit für die Erstellung der Service-Schicht ist bei beiden Varianten vergleichbar, da im Entwicklungswerkzeug lediglich Feldbezeichnungen und Datenty-

---

<sup>161</sup> Vgl. Wagner, Karl W. / Brunner, Franz J. (2008), S. 148.

pen beschrieben werden müssen. Als Ergebnis der traditionellen Entwicklung erhält man Datenbanktabellen, die via SQL angesprochen werden können, beim Composition Application Framework erhält man einen Webservice für Lese- und Schreiboperationen auf das Datenmodell. Typischerweise wird das reine Datenmodell meist mit einer Service-Klasse angesprochen, dies kann gleichermaßen durch eine ABAP-Klasse oder einen Applikationsservice des Composition Application Framework erfolgen. Da die Implementierung in beiden Varianten manuell erfolgt, ist der Aufwand identisch. Zu beachten ist jedoch, dass das CAF-Datenmodell über Systemgrenzen hinweg genutzt werden kann und somit einen entscheidenden Mehrwert gegenüber dem klassischen Vorgehen entsteht. Allerdings hat die Erfahrung gezeigt, dass der Aufruf der Services mit mehr Aufwand als der direkte Datenbankzugriff verbunden ist. Dies ist zwar nicht verwunderlich, soll jedoch das Bewusstsein schaffen, dass eine Kapselung der Komponenten zusätzlichen Aufwand verursacht und es sich erst bei Wiederverwendung oder häufigen Änderungen der Software positiv bemerkbar macht.

#### *Prozess-Schicht*

In der Prozessschicht wurde entsprechend der Projekt-Zeiterfassungen am meisten Kosten eingespart. Beim Vergleich der Zeiten stehen zwei Tage für das SAP BPM zehn Tagen für den Business Workflow gegenüber. Begründet wird dies insbesondere dadurch, dass man beim BPM direkt die Operationen der Service-Schicht verwenden kann, ohne diese noch einmal durch klassische SAP Business Objekte kapseln zu müssen und dass SAP BPM die Nutzung von Weboberflächen vorsieht. Für das Projekt zum Buchen von Veranstaltung wurde viel Aufwand in die Integration der BSP-Oberflächen gesteckt. Weiterhin ist die Programmierung innerhalb des SAP Business Workflow verglichen mit heutigen Möglichkeiten sehr mühsam.

#### *UI-Schicht*

Auch hier stellt sich durch die modellgetriebene Entwicklung gegenüber der manuellen Implementierung ein Kostenersparnis ein. Entsprechend der Beobachtungen aus den beiden Projekten ist man mit Web Dynpro ABAP doppelt so schnell am Ziel, wie es bei der Verwendung von BSP mit HTMLB als Oberflächenbibliothek der Fall wäre. Sofern also alle Anforderungen durch das Modellierungswerkzeug unterstützt werden, sollte die modellgetriebene UI-Entwicklung zugunsten der manuellen Implementierung von Oberflächen bevorzugt werden.

Zusammenfassend kann gesagt werden, dass die modellgetriebene Softwareentwicklung mit dem NetWeaver CE 7.3 eine höhere Produktivität aufweist, als sie bei bisherigen Softwareentwicklungen im SAP Umfeld zu beobachten war. Prämisse dafür ist jedoch, dass benötigte Funktionalitäten durch die Modellierungswerkzeuge unterstützt werden.

Weiterhin muss für eine Kostenbetrachtung berücksichtigt werden, ob ausreichend geeignete Entwickler verfügbar sind. Sofern die benötigten Kenntnisse nicht von internen Mitarbeitern abgedeckt werden, müssen Unternehmen zusätzliche Kosten für Fortbildungen berücksichtigen oder das Know-how von externen Beratungs- und Entwicklungsunternehmen einkaufen.

Das zentrale Monitoring ist in der Betriebsphase kostensenkend. Nachdem sich Modul- und Anwendungsbetreuer in die neue Plattform eingearbeitet haben, werden sich Synergieeffekte bei jedem neuen, mit dem NetWeaver CE umgesetzten Szenario einstellen, da das Monitoring von Geschäftsprozessen zentral und einheitlich erfolgt.

## 7 Fazit

### 7.1 Zusammenfassung

In der theoretischen Betrachtung der modellgetriebenen Softwareentwicklung (MDSD) wurde herausgestellt, dass eine Software vollständig durch formale Modelle beschrieben und anhand von Transformationstemplates in ausführbare Software umgewandelt oder zur Laufzeit interpretiert wird. Die MDSD sieht sich als flexibler Ansatz, der aus den Schwachstellen bisheriger Konzepte gelernt hat. Während frühere Modellierungsansätze an proprietären Werkzeugen, starren Quellcodetransformationen und Plattformvorgaben scheiterten, bieten moderne Ansätze Möglichkeiten zur Einflussnahme in allen Phasen der Modellierung und Modelltransformation.

Die Modellierungswerkzeuge des SAP NetWeaver CE hinsichtlich der Verwendung von modernen Modellierungsansätzen einzuordnen, fällt nicht eindeutig aus. Zwar weisen die SAP AG Technologien eine zeitgemäße Architektur auf, welche den Anforderungen von MDSD genügen, aus Kundensicht kann man jedoch zu einer konträren Sichtweise kommen. Grund hierfür ist, dass keine Einflussnahme auf Modellierungssprachen und Transformationstemplates vorgesehen ist. Die Kunden können nur Funktionen nutzen, welche durch SAP vorgesehen sind. Die SAP AG stellt also mit modernen Mitteln ein für den Kunden fertig konfiguriertes Werkzeug zur Verfügung, welches aus Sicht der SAP für die Verwendung im gewöhnlichen Geschäftsumfeld geeignet ist und in diesem Bereich eine außerordentlich hohe Produktivität erlaubt. Die Betrachtung der einzelnen Technologien aus den verschiedenen Architekturebenen einer Composite Application hat gezeigt, dass dies der SAP AG gut gelungen ist. Die Modellierungswerkzeuge sind durchweg gut bedienbar und zeigten während der Erprobung und im Projekteinsatz keine signifikanten Schwächen beim Erstellen einer durchschnittlichen Geschäftsanwendung. Alle Modellierungswerkzeuge sind im NetWeaver Developer Studio integriert, so dass auch eine durchgängig einheitliche Arbeitsweise möglich ist.

Weiterhin ist es durch die Modellierungswerkzeuge nicht vorgesehen, dass man den Weg der modellgetriebenen Entwicklung verlässt. Das heißt, es ist nur mit viel technischem Verständnis möglich, den aus einem Modell generierten Quellcode als Arbeitsgrundlage für manuelle Erweiterungen oder Änderungen zu verwenden. Änderungen sollten stets nur über das Modell oder durch manuelle Implementie-

rung von Quellcode innerhalb von geschützten Bereichen vorgenommen werden. Anderenfalls würden die manuellen Anpassungen beim nächsten Generieren der Anwendung verloren gehen. Da das NetWeaver Development Studio jedoch vor jedem Deployment-Vorgang das Generieren der Anwendung voraussetzt, werden Änderungen bereits überschrieben, bevor sie zum Applikationsserver übertragen werden können. Bei Oberflächenkomponenten, welche die Web Dynpro Laufzeitumgebung verwenden, sind manuelle Änderungen aufgrund des Interpreter-Ansatzes sogar unmöglich. Daher sollten Restriktionen der Technologien im Vordergrund beleuchtet werden, um sicherzustellen, dass alle Anforderungen erfüllt werden können. Gegebenenfalls müssen Anforderungen umformuliert oder es muss auf flexiblere Technologiealternativen ausgewichen werden.

Im SAP NetWeaver CE 7.3 werden viele offene Standards verwendet, die die Integrationen von Softwarekomponenten abseits von SAP Backend-Systemen erleichtern. Der SOA-Gedanke wurde in diesem Produkt konsequent umgesetzt, sodass selbst Komponenten auf demselben Applikationsserver mehrheitlich via Webservices kommunizieren. Die Serviceschicht ist in der Lage gesamte Datenmodelle und Dienste von Dritten einheitlich als Service bereitzustellen. Selbst BPM Geschäftsprozesse werden über SOAP-Nachrichten von außen angesprochen, um beispielsweise Prozesse zu starten oder fortzuführen. Schlussendlich wird auch in der UI-Schicht die Nutzung vorhandener Services durch Mechanismen zum Generieren der Proxy-Klassen erheblich erleichtert. Hervorzuheben ist in diesem Zusammenhang auch die konsequente Anbindung der Service Registry, die als zentraler Punkt zum Publizieren und Auffinden von Services dient.

Weiterhin wird zur Prozessmodellierung eine etablierte Modellierungsnotation verwendet, welche für fachlich und technisch orientierte Projektmitarbeiter gleichermaßen verständlich und für deren Sichtweise angemessen ist. Die vollständige BPMN 2.0 Spezifikation ist zwar nicht im NetWeaver CE 7.3 verfügbar, aber da es sich beim BPM um ein strategisches Produkt der SAP handelt, ist anzunehmen, dass sich der Funktionsumfang in der Zukunft erweitern wird. Mit den derzeitigen Mitteln ist jedoch bereits die Umsetzung vieler Prozessvarianten möglich.

Bezugnehmend auf die Fragestellungen aus der Zielsetzung dieser Arbeit ist zu klären, ob die Erstellung von Geschäftsanwendung ohne manuelle Programmierung möglich ist. Hierzu kann festgehalten werden, dass dies grundsätzlich möglich ist. Voraussetzungen sind die Verwendung des Visual Composers für die Erstellung von Benutzeroberflächen und das Vorhandensein von benötigten Backend-Services. In der Service-Schicht hat es sich zwar in der Realisierung des

betrieblichen Eingliederungsmanagements als vorteilhaft herausgestellt, Zugriffsmethoden der Business Objekte durch einen manuell implementierten Anwendungsservice zu kapseln, jedoch ist dieses Vorgehen nicht zwingend notwendig.

Trotzdem sind Programmierkenntnisse für die Erstellung von Composite Applications hilfreich, bei gewissen Technologieentscheidungen wie beispielsweise Web Dynpro sogar unabkömmlich. Der Anteil des Programmcodes, welcher sich mit technischen Aspekten befasst, ist jedoch im Vergleich zu jenem mit eher fachlichem Bezug wesentlich geringer. Somit kann behauptet werden, dass sich das Aufgabengebiet des Entwicklers wesentlich verändert. In der Umsetzung des Projekts zum betrieblichen Eingliederungsmanagement hat sich diesbezüglich herausgestellt, dass man in Zusammenarbeit mit dem Fachbereich Modelle diskutiert und gegebenenfalls direkt anpasst. Daher kann beobachtet werden, dass zur klassischen Entwicklerrolle vermehrt analytische und beratende Tätigkeiten hinzukommen.

Die Betrachtung von Qualitäts- und Kostenaspekten ergab, dass man verglichen mit klassischer Softwareentwicklung tatsächlich eine Kostenersparnis erreichen kann, welche sich nicht auf Qualitätsaspekte niederschlägt. Die Qualität an sich erreicht ein konstant hohes Niveau, da viele Standardfunktionalitäten ohne großen Aufwand genutzt werden können und die Begrenztheit verwendbarer Komponenten eine einheitliche Systemlandschaft fördert. Der hohe Standardisierungsgrad wirkt sich auch in der Betriebsphase kostensenkend aus. Besonders große Vorteile ergeben sich für Szenarien zur Abbildung von kollaborativen Geschäftsprozessen, da das SAP Business Process Management eine einfach und flexible Prozessmodellierung und Ablaufsteuerung gewährleistet sowie die Möglichkeiten des NetWeaver CE bestmöglich ausschöpft.

## **7.2 Bewertung**

Es hat sich während der Betrachtung herausgestellt, dass die Modellierungswerkzeuge des NetWeaver CE nicht der Sichtweise der Literatur hinsichtlich der Anforderungen an eine moderne modellgetriebene Softwareentwicklung entsprechen. Stattdessen liegen charakteristische Ähnlichkeiten zum gescheiterten CASE-Ansatz vor, da Kunden sich auf eine Plattform festlegen und keinen Einfluss auf das finale Produkt haben. Dennoch hat der NetWeaver CE einen sehr guten Eindruck hinterlassen und hinsichtlich Kosten und Qualitätsaspekten überzeugt. Gründe hierfür resultieren aus dem scheinbar starren Ansatz der SAP. Der Aufbau projektspezifischer Modelltransformationen ist ein zeitaufwändiges und komplexes

Unterfangen, so dass es nur für große oder langlebige Projekte lohnend ist. Für den NetWeaver CE ist dies nicht der Fall, eine hohe Produktivität stellt sich schon bei kleinen und einmaligen Projekten ein. Weiterhin ist ein Vergleich mit CASE nicht passend, da mehrere domänenspezifische Modellierungssprachen verwendet werden, welche für ihr jeweiliges Einsatzgebiet passend sind. Zudem ist eine gewisse Flexibilität durch die Verwendung von offenen Standards gewährleistet. Die Nutzung von einzelnen Komponenten von Fremdanbietern erfolgt unproblematisch via Webservices.

Die nachfolgende Abbildung stellt einen Vergleich von Composite Applications mit traditioneller Softwareentwicklung am Beispiel der SAP Business Server Pages, welche eine vollständig manuelle Implementierung von Webanwendungen erfordern, her.

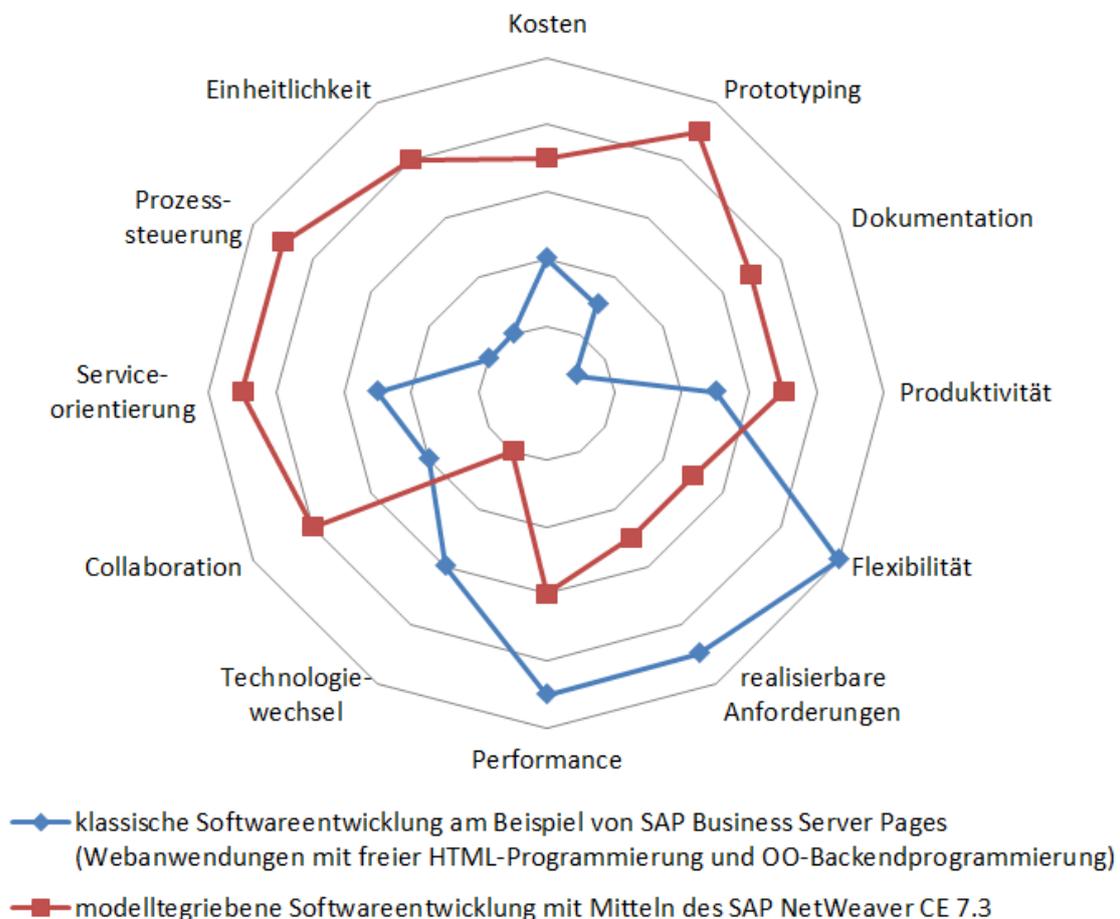


Abbildung 7-1: Vergleich von Composite Applications mit BSP-Anwendungen

Quelle: Eigene Darstellung.

Die modellgetriebene Softwareentwicklung soll Schwachstellen der klassischen Softwareentwicklung verbessern, um für zukünftige Anforderungen gewappnet zu

sein. Dies gelingt dem NetWeaver CE, da der Abstraktionsgrad bei der Softwareentwicklung ein weiteres Mal erhöht wurde. Anwendungsentwickler können sich nun stärker auf fachliche Aspekte konzentrieren und somit komplexere Anwendungen als zuvor realisieren, ohne dass Aufwand und Nutzen auseinanderklaffen. Spezifikationen können gemeinsam mit dem Fachbereich aufgenommen und in formalen Modellen erfasst werden, welche weiterentwickelt werden und schlussendlich eine gültige Dokumentation darstellen. Somit verbessert die Softwareentwicklung mit Hilfe des NetWeaver CE 7.3 alle Aspekte, die als Schwachstellen der klassischen Softwareentwicklung identifiziert wurden, bis auf eine Ausnahme. Lediglich die Problematik des Technologiewechsels kann durch die starre Modelltransformation nicht gelöst werden. Hierfür muss SAP das Vertrauen entgegengebracht werden, dass sie auch zukünftig erfolgreich am Markt agieren und der NetWeaver CE weiterentwickelt und mit Updates versorgt wird. Darüber hinaus hat ein Großteil potentieller NetWeaver CE Kunden bereits durch die Verwendung der SAP Business Suite eine hohe Abhängigkeit zur SAP AG, so dass diese zusätzliche Herstellerbindung kaum ins Gewicht fällt.

Weiterhin hat sich gezeigt, dass es nur begrenzt möglich ist, vollständig modellgetrieben zu entwickeln. Das Erreichen eines Automatisierungsgrades von ca. 80% entsprechend der Erfahrungswerte von Stahl, Thomas u.a. ist auch für das Projekt „Betriebliches Eingliederungsmanagement“ ein passender Richtwert.<sup>162</sup> Es sollten daher an sinnvollen Stellen Möglichkeiten zur manuellen Eingabe von Quellcode geschaffen sein, um trotz des hohen Standardisierungsgrades erweiterten Funktionsumfang zu gewährleisten. Beispiele hierfür sind die Applikation-Services des Composite Application Framework und die Möglichkeiten zur server-seitigen Programmierung bei Web Dynpro. Aufgrund dieser Mechanismen konnten im praktischen Einsatz alle Anforderungen ohne Kompromisse erfüllt werden.

### **7.3 Ausblick**

Zum Abschluss dieser Arbeit ist die zweite Phase des betrieblichen Eingliederungsmanagements fertiggestellt worden. Neben dem Umsetzen der fachlichen Anforderungen ging es darum, die finale Architektur aufzubauen und eine Referenzimplementierung für wiederkehrende Problemstellungen zu finden. Da dies gelungen ist, kann ohne Sorge auf die nachfolgend anstehende dritte Projektphase geblickt werden. Ziel ist es, bis zum Jahresende das Projekt abzuschließen und

---

<sup>162</sup> Vgl. Stahl, Thomas u.a. (2007), S. 41f.

dem Fachbereich eine umfassende IT-Unterstützung für die Durchführung seines Geschäftsprozesses zu gewährleisten.

In der anschließenden Betriebsphase werden tiefergehende Erkenntnisse zu Betriebskosten und den Aufwand für Wartungsarbeiten gesammelt, welche für spätere Wirtschaftlichkeitsanalysen herangezogen werden können. Die ersten Ideen zur Prozessoptimierung wurden bereits vom Fachbereich entwickelt. So ist es beispielsweise denkbar, dass zukünftig eingehende Antwortformulare nicht mehr manuell erfasst, sondern anhand von Barcodes auf den gescannten Dokumenten automatisiert verarbeitet werden. Ob diese Idee einer Kosten-Nutzen-Analyse standhält, wird zu gegebener Zeit überprüft, aber aus Sicht des BPM Prozess lässt sich diese Änderung leicht umsetzen.

Auch seitens der SAP wird die Zukunft Änderungen mit sich bringen. In der SAP Roadmap zur Process Orchestration werden Verbesserungen im zentralen Monitoring angekündigt. Zukünftig soll es dem Administrator unter anderem möglich sein, den Prozess-Kontext einer fehlerhaften Prozessinstanz zu manipulieren und anschließend fortzuführen. Neben der Weiterentwicklung von BPM können bestehende Composite Applications von der neuen Oberflächentechnologie SAP UI5 profitieren. Diese auf HTML5 basierende Technologie ermöglicht eine bessere Nutzung mobiler Endgeräte.<sup>163</sup>

---

<sup>163</sup> Vgl. SAP AG (2012b), Folie 19ff.

## Literaturverzeichnis

**Baresi, Luciano / Di Nitto, Elisabetta (2007):**

Test and Analysis of Web Services,  
Berlin, 2007.

**Baumgartl, Axel / Mebus, Frank / Seemann, Volker (2010):**

Das SOA-Praxisbuch für SAP,  
Bonn, 2010.

**BSR (2010):**

Dienstvereinbarung zur Durchführung eines BEM,  
Dokument auf Datenträger vorhanden, 2010.

**BSR (2012):**

Standardisierung des BEM-Prozesses mittels eines DV-Verfahrens,  
Dokument auf Datenträger vorhanden, 2012.

**Cooke, Daniel (1994):**

The Impact of CASE Technology on Software Processes,  
New York, 1994.

**Gitzel, Ralf (2006):**

Model-driven software development using a metamodel-based extension mechanism for UML,  
Frankfurt am Main, 2006.

**Holley, Kerrie / Arsanjani, Ali (2011):**

100 SOA Questions – Asked and Answered,  
Boston, 2011.

**Kessler, Karl u.a. (2007):**

Java-Programmierung mit SAP NetWeaver,  
Bonn, 2007.

**Kleppe, Anneke / Warmer, Jos / Bast, Wim (2003):**

MDA explained: the model driven architecture : practice and promise, 3. Auflage,  
Boston, 2003.

**OMG (2003):**

MDA Guide Version 1.0.1,  
Dokument auf Datenträger vorhanden, 2003.

**OMG (2011):**

Business Process Model and Notation (BPMN) – Version 2.0  
Dokument auf Datenträger vorhanden, 2011.

**Raistrick, Chris u.a. (2004):**

Model driven architecture with executable UML,  
Cambridge, 2004.

**Petrasch, Roland / Meimberg, Oliver (2006):**

Model Driven Architecture.  
Heidelberg, 2006.

**Pietrek, Georg / Trompeter, Jens (2007):**

Modellgetriebene Softwareentwicklung: MDA und MDSD in der Praxis,  
Paderborn, 2007.

**Rauscher, Jan / Stiehl, Volker (2007):**

SAP NetWeaver Composition Environment,  
Bonn, 2007.

**Rechenberg, Peter / Pomberger, Gustav (2006):**

Informatik-Handbuch,  
München, 2006.

**Rempp, Gerhard u.a. (2011):**

Model Driven SOA,  
Heidelberg, 2011.

**Richardson, Leonard / Ruby, Sam (2007):**

Web Services mit REST,  
Köln, 2007.

**Salein, Marcel u.a. (2010):**

Praxishandbuch SAP NetWeaver Visual Composer,  
Bonn, 2010.

**SAP AG (2008):**

Introduction to Visual Composer for Composition Environment,  
Präsentation auf Datenträger vorhanden, 2008.

**SAP AG (2012):**

SAP Product Roadmap – SAP User Interfaces,  
Präsentation auf Datenträger, 2012.

**SAP AG (2012b):**

SAP NetWeaver Process Orchestration,  
Präsentation auf Datenträger, 2012.

**SAP Community Network (2011):**

Floorplan Manager for Web Dynpro ABAP - Developer's Guide,  
Dokument auf Datenträger vorhanden, 2011.

**Snabe, Jim Hagemann u.a. (2009):**

Business Process Management – The SAP Roadmap,  
Bonn, 2009.

**Stahl, Thomas u.a. (2007):**

Modellgetriebene Softwareentwicklung,  
Heidelberg, 2007.

**Suhl, Leena (2004):**

Komplexitätsreduktion in der Softwareentwicklung: Ein systemtheoretischer An-  
satz, Paderborn, 2004.

**Vogel, Oliver u.a. (2009):**

Software-Architektur: Grundlagen - Konzepte – Praxis, 2. Auflage,  
Heidelberg, 2009.

**Völter, Markus (2005):**

Modellgetriebene Softwareentwicklung,  
Artikel veröffentlicht in DatenbankSpektrum,  
Dokument auf Datenträger vorhanden, 2005.

**Wagner, Karl W. / Brunner, Franz J. (2008):**

Qualitätsmanagement – Leitfaden für Studium und Praxis,  
München, 2008.

**Whealy, Chris (2007):**

Inside Web Dynpro for Java,  
Boston, 2007.

## Internetverzeichnis

### **BSR (2010):**

Der Vorstand der BSR,  
abgerufen am 09.07.2012,  
<http://www.bsr.de/9419.html>.

### **BSR (2010b):**

Soziale Verantwortung und Engagement,  
abgerufen am 09.07.2012,  
<http://www.bsr.de/9451.html>.

### **Gardner, Tracy / Yusuf, Larry (2006):**

Explore model-driven development (MDD) and related approaches: A closer look at model-driven development and other industry initiatives,  
abgerufen am 24.05.2012,  
<http://www.ibm.com/developerworks/library/ar-mdd3/> .

### **Litoiu, Milena (2004):**

Reduce complexity with model-driven development,  
abgerufen am 07.05.2012,  
<https://www.ibm.com/developerworks/ibm/library/i-modev1/> .

### **SAP AG (2011):**

Komponenten und Werkzeuge – SAP NetWeaver Business Process Management,  
abgerufen am 04.07.2012,  
<http://www.sap.com/germany/plattform/netweaver/components/sapnetweaverbpm/index.epx> .

### **SAP AG (2011b):**

Product Availability Matrix,  
abgerufen am 14.08.2012,  
<https://websmp202.sap-ag.de/pam> -> SAP NetWeaver 7.3.

### **SAP AG (2012):**

Von Walldorf an die Wall Street. Eine Erfolgsgeschichte,  
abgerufen am 20.04.2012,  
<http://www.sap.com/germany/about/index.epx> .

**SAP AG (2012b):**

SAP NetWeaver Composition Environment,  
abgerufen am 20.04.2012,  
<http://help.sap.com/nwce> .

**SAP AG (2012c):**

OKP SAP NetWeaver Composition Environment 7.3,  
abgerufen am 26.04.2012,  
<https://training.sap.com/de/de/course/oce73-okp-sap-netweaver-composition-environment-73-e-learning-095-g-en/> .

**SAP AG (2012d):**

Über die SAP AG,  
abgerufen am 24.05.2012,  
<http://www.sap.com/corporate-de/our-company/history.epx> .

**SAP AG (2012e):**

Unser Unternehmen – SAP auf einen Blick,  
abgerufen am 24.05.2012,  
<http://www.sap.com/corporate-de/our-company/inbrief/index.epx> .

**SAP Help - NetWeaver 7.3EHP1 (2012):**

Web-Dynpro-CHIPs erstellen,  
abgerufen am 12.07.2012,  
[http://help.sap.com/saphelp\\_nw73ehp1/helpdata/de/b1/914b8c3fb648babc568967d430a9f8/content.htm](http://help.sap.com/saphelp_nw73ehp1/helpdata/de/b1/914b8c3fb648babc568967d430a9f8/content.htm)

**SAP Help - NetWeaver CE 7.0EHP2 (2010):**

Creating Composite Applications,  
abgerufen am 31.05.2012,  
[http://help.sap.com/saphelp\\_nw70ehp2/helpdata/en/ad/019742ad14c46ae1000000a155106/content.htm](http://help.sap.com/saphelp_nw70ehp2/helpdata/en/ad/019742ad14c46ae1000000a155106/content.htm).

**SAP Help - NetWeaver CE 7.0EHP2 (2010b):**

Working with Composite Application Services,  
abgerufen am 31.05.2012,  
[http://help.sap.com/saphelp\\_nw70ehp2/helpdata/en/1c/8eb184e297414c8701c44e1aee42f9/content.htm](http://help.sap.com/saphelp_nw70ehp2/helpdata/en/1c/8eb184e297414c8701c44e1aee42f9/content.htm).

**SAP Help – NetWeaver CE 7.3 (2012):**

Composing Services – Concepts,

abgerufen am 31.05.2012,

[http://help.sap.com/saphelp\\_nw73/helpdata/de/44/5d64c1d9966d11e1000000a1553f6/content.htm](http://help.sap.com/saphelp_nw73/helpdata/de/44/5d64c1d9966d11e1000000a1553f6/content.htm).

**SDN WJava FAQs (2008):**

Meta Model Concept,

abgerufen am 26.06.2012,

<http://wiki.sdn.sap.com/wiki/display/WJava/Meta+Model+concept>.

**Stiehl, Volker (2009):**

BPMN - Vermittler zwischen den Welten,

abgerufen am 02.07.2012,

<http://it-republik.de/jaxenter/artikel/BPMN---Vermittler-zwischen-den-Welten-Teil-1-2531.html>.

**Süddeutsche.de (2012):**

Vorstoß gegen Konkurrent Oracle - SAP zieht es in die Datenwolke,

abgerufen am 22.05.2012,

<http://www.sueddeutsche.de/wirtschaft/vorstoss-gegen-konkurrent-oracle-sap-zieht-es-in-die-datenwolke-1.1364372>.

**Universität Ulm (2009):**

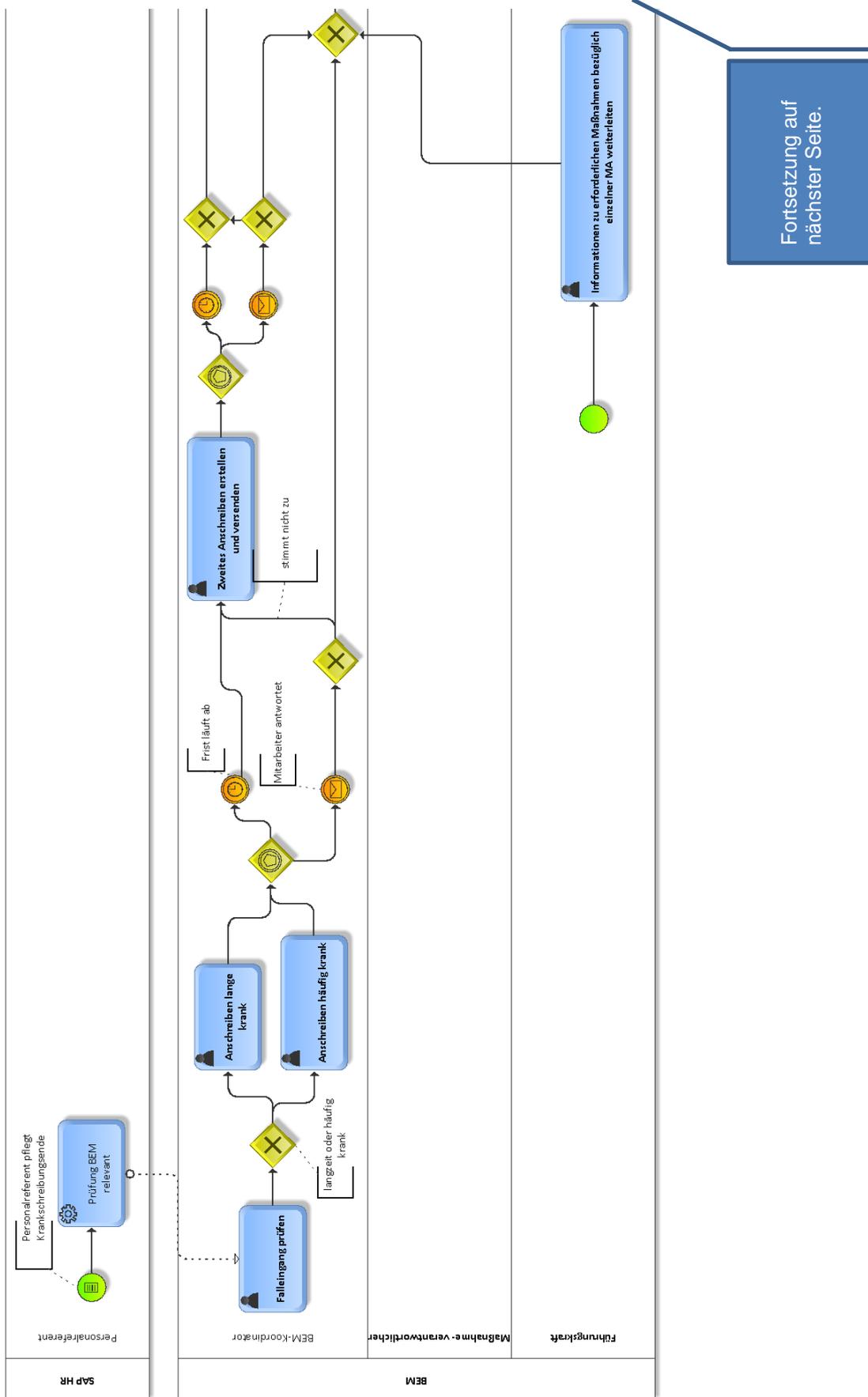
Model Driven Development,

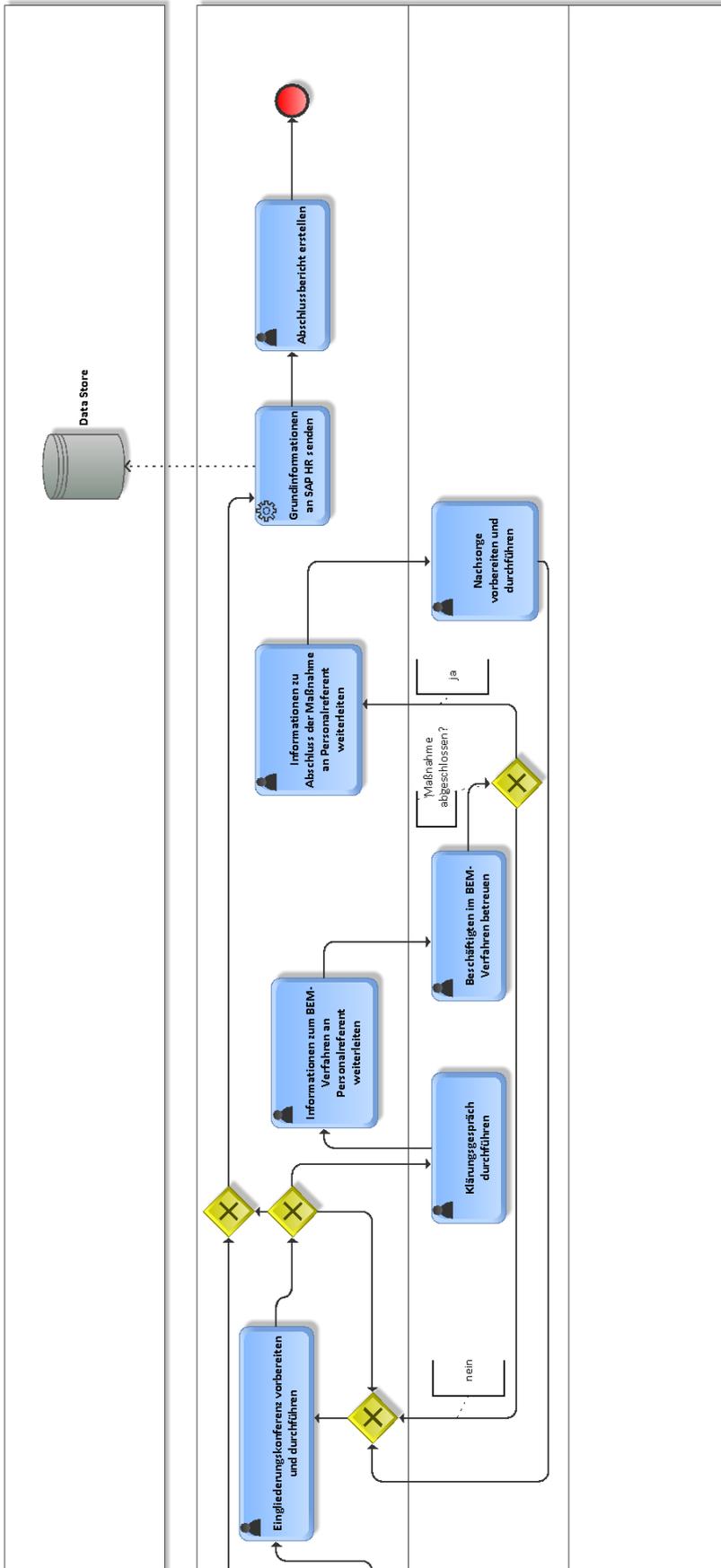
abgerufen am 21.04.2011,

<http://www.uni-ulm.de/in/pm/forschung/themen/mdd.html>.



Anhang II: BPMN Prozess zum BEM





## **Eidesstattliche Versicherung**

Ich versichere hiermit, dass ich die vorliegende Masterthesis selbstständig und ohne fremde Hilfe angefertigt und keine andere als die angegebene Literatur benutzt habe. Alle von anderen Autoren wörtlich übernommenen Stellen wie auch die sich an die Gedankengänge anderer Autoren eng anlehnenden Ausführungen meiner Arbeit sind besonders gekennzeichnet. Diese Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Berlin, den 30. August 2012

Tobias Lucke