

**Oliver Neumann**

Praktische Evaluierung der  
aspektorientierten Programmierung zur  
Stärkung der Wandlungsfähigkeit eines  
ERP-Systems

**Diplomarbeit**

# BEI GRIN MACHT SICH IHR WISSEN BEZAHLT



- Wir veröffentlichen Ihre Hausarbeit, Bachelor- und Masterarbeit
- Ihr eigenes eBook und Buch - weltweit in allen wichtigen Shops
- Verdienen Sie an jedem Verkauf

Jetzt bei [www.GRIN.com](http://www.GRIN.com) hochladen  
und kostenlos publizieren



## **Bibliografische Information der Deutschen Nationalbibliothek:**

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de/> abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlanges. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

## **Impressum:**

Copyright © 2006 GRIN Verlag  
ISBN: 9783638821889

## **Dieses Buch bei GRIN:**

<https://www.grin.com/document/80641>

**Oliver Neumann**

**Praktische Evaluierung der aspektorientierten Programmierung zur Stärkung der Wandlungsfähigkeit eines ERP-Systems**

## **GRIN - Your knowledge has value**

Der GRIN Verlag publiziert seit 1998 wissenschaftliche Arbeiten von Studenten, Hochschullehrern und anderen Akademikern als eBook und gedrucktes Buch. Die Verlagswebsite [www.grin.com](http://www.grin.com) ist die ideale Plattform zur Veröffentlichung von Hausarbeiten, Abschlussarbeiten, wissenschaftlichen Aufsätzen, Dissertationen und Fachbüchern.

### **Besuchen Sie uns im Internet:**

<http://www.grin.com/>

<http://www.facebook.com/grincom>

[http://www.twitter.com/grin\\_com](http://www.twitter.com/grin_com)

Diplomarbeit

Praktische Evaluierung der  
aspektorientierten Programmierung  
zur Stärkung der Wandlungsfähigkeit  
eines ERP-Systems

Oliver Neumann

Technische Universität Berlin  
Fakultät IV - Elektrotechnik und Informatik  
Institut für Softwaretechnik und Theoretische Informatik  
Fachgebiet Softwaretechnik

Universität Potsdam  
Lehrstuhl für Wirtschaftsinformatik und Electronic Government

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
1.1	Motivation . . . . .	7
1.2	Struktur . . . . .	8
<b>2</b>	<b>Enterprise Resource Planning</b>	<b>10</b>
2.1	Begriffsklärung und Aufgabendefinition . . . . .	10
2.2	Entstehungsgeschichte . . . . .	13
2.3	Enterprise Systeme . . . . .	16
2.4	Aufbau eines ERP-Systems . . . . .	17
2.5	Open-Source in unternehmensweiten Anwendungen . . . . .	19
<b>3</b>	<b>Wandlungsfähigkeit</b>	<b>23</b>
3.1	Begriffsklärung . . . . .	23
3.2	Wandlungsfähige ERP-Systeme . . . . .	24
3.2.1	Kriterien der Wandlungsfähigkeit . . . . .	24
3.2.2	Wandlungsfähige Architekturmodelle . . . . .	27
3.2.3	Ermittlung der Wandlungsfähigkeit . . . . .	28
3.2.4	Auswertung der Wandlungsfähigkeit . . . . .	31
3.2.5	Bedeutung der Wandlungsfähigkeit für ERP-Systeme . . . . .	32
<b>4</b>	<b>Compiere</b>	<b>34</b>
4.1	Funktionalitäten . . . . .	34
4.1.1	Quote-to-Cash . . . . .	35
4.1.2	Requisition-to-Pay . . . . .	36
4.1.3	Customer-Relations-Management (CRM) . . . . .	37
4.1.4	Partner-Relations-Management . . . . .	37
4.1.5	Supply-Chain-Management (SCM) . . . . .	38
4.1.6	Performanz Analyse . . . . .	39
4.1.7	Web-Store . . . . .	39
4.1.8	Workflows . . . . .	40
4.1.9	Zusammenfassung . . . . .	40
4.2	Aufbau und Struktur . . . . .	41
4.2.1	Technische Anforderungen . . . . .	41
4.2.2	Applikationsstruktur . . . . .	43

4.2.3	Datenbankstruktur . . . . .	47
4.2.4	Datenbankzugriff . . . . .	50
4.2.5	Grafische Abbildung der Datenbank . . . . .	56
<b>5</b>	<b>Aspektororientierte Softwareentwicklung</b>	<b>63</b>
5.1	Begriffsklärung und allgemeine Übersicht . . . . .	64
5.1.1	Problematik . . . . .	64
5.1.2	Elemente der Aspektororientierung . . . . .	66
5.1.3	Verwebung . . . . .	68
5.1.4	Die Vor- und Nachteile der Aspektororientierung . . . . .	72
5.1.5	AOP-Implementierungen für Java . . . . .	74
5.2	JBoss AOP . . . . .	77
5.2.1	Das Pointcut-Modell . . . . .	77
5.2.2	Verwebung und Deployment von Aspektklassen . . . . .	82
5.3	Object Teams . . . . .	84
5.3.1	Rollenbasierte Programmierung . . . . .	84
5.3.2	Kollaborationen . . . . .	85
5.3.3	Teams . . . . .	86
5.3.4	Rollen . . . . .	89
5.3.5	Modellierung aspektorientierter Anwendungen . . . . .	93
5.3.6	Translations-Polymorphismus . . . . .	95
<b>6</b>	<b>Compiere Monitor</b>	<b>97</b>
6.1	Aspektororientierung in Compiere . . . . .	97
6.2	Zielsetzung . . . . .	99
6.3	Funktionalitäten . . . . .	100
6.4	Die Compiere-Aspektgeneratoren . . . . .	106
6.4.1	Aspektklassen . . . . .	106
6.4.2	Aspektkonfiguratoren . . . . .	123
6.5	Verwendete Techniken . . . . .	127
6.6	Datenbankmodell . . . . .	131
6.7	Implementierung . . . . .	134
6.7.1	Allgemeine Konzepte . . . . .	134
6.7.2	Design Patterns . . . . .	144
6.8	Zusammenfassung . . . . .	152
<b>7</b>	<b>Ergebnis und Zusammenfassung</b>	<b>155</b>
7.1	Evaluierung der Wandlungsfähigkeit von Compiere . . . . .	155
7.1.1	Ermittlung der Einflüsse des Compiere Monitor . . . . .	156
7.1.2	Ergebnisportfolio . . . . .	159
7.2	Zusammenfassung und Ausblick . . . . .	161

# Abbildungsverzeichnis

2.1	Horizontale und vertikale Integration betrieblicher Anwendungssysteme (nach [Gro04]) . . . . .	12
2.2	Aufgabenverteilung innerhalb betrieblicher Informationssysteme . . . . .	13
2.3	Evolutionsschritte des ERP . . . . .	15
2.4	ERP-Prozesse in Enterprise Software . . . . .	16
2.5	Aufbau eines ERP-Systems (nach [Gro04]) . . . . .	18
3.1	Architekturmodell für wandlungsfähige ERP-Systeme (nach [GLA06]) . . . . .	27
3.2	Hierarchie zur Ermittlung der Wandlungsfähigkeit eines ERP-Systems . . . . .	30
3.3	Ergebnisportfolio für ERP-Systeme bezüglich Wandlungsfähigkeit (nach [ALG06]) . . . . .	32
4.1	Basisfunktionalitäten von Compiere (Nach [Com]) . . . . .	35
4.2	Zentralisierungen in einem Compiere-Systemverbund (Nach [Com]) . . . . .	39
4.3	Zugriffsebenen der Compiere-Architektur . . . . .	44
4.4	Architektur der Compiere-Prozessor-Implementierung . . . . .	47
4.5	Klassendiagramm der beim Datenbankzugriff beteiligten Objekte . . . . .	51
4.6	Screenshot der Compiere-Client-Anwendung mit Unterteilung der Elementtypen . . . . .	58
4.7	Datenbankstruktur zur Darstellung der dynamischen Window-Elemente . . . . .	59
4.8	Integration der dynamischen Window-Elemente in die grafische Benutzeroberfläche von Compiere . . . . .	61
5.1	Scattering und Tangling innerhalb eines moduldurchschneidenden concern . . . . .	65
5.2	Aspect-Weaving zur Kompilierzeit . . . . .	70
5.3	Aspect-Weaving zur Ladezeit am Beispiel von JMangler (nach [KCA04]) . . . . .	71

## Abbildungsverzeichnis

---

5.4	Deployment und Instrumentalisierung von Klassen innerhalb des JBoss-Applikation-Server (nach [DJS04]) . . . .	83
5.5	Dimension der Kollaborationen innerhalb von Klassenhierarchien . . . . .	86
5.6	Modellierung von ObjectTeams-Anwendungen mit UFA (nach [Her02a]) . . . . .	94
5.7	Beispiel einer Adaption von Basisklassen mit ObjectTeams	95
6.1	Modell der Aspekte des Compierre-Aspektgenerators. (Modelliert nach UML 2.1 mit Anlehnungen an UFA aus Abschnitt 5.3.5) . . . . .	119
6.2	Vereinfachtes Entity-Relationship-Modell für Tabellen des Aspektmanagement zur Speicherung der Team- und Rollenklassen . . . . .	133
6.3	Klassendiagramm der Komponenten der Tabellenabbildung	137
6.4	Klassendiagramm der Objekte des Rule-Monitor . . . . .	141
6.5	Komponenten der graphischen Darstellung von Tabellendaten . . . . .	142
6.6	Anwendung des Model-View-Controller-Pattern im Compierre Monitor . . . . .	146
6.7	Kommunikationsstruktur bei Datenänderung eines CM-DataType durch ein CMPanelBoolean . . . . .	149
6.8	Implementierung des Command-Pattern im Compierre Monitor . . . . .	150
6.9	Observer-Pattern innerhalb des Compierre Monitor . . . .	151
7.1	Ergebnisportfolio der Wandlungsfähigkeit von Compierre .	161

# Listings

4.1	Zugriff auf Tabelleninhalte durch Objekte der Persistenz- und Businessschicht am Beispiel der Tabelle <i>M_Product</i>	53
5.1	Beispiel eines JBoss-AOP XML-Deskriptor zur Definition von Pointcuts und Interceptors	77
5.2	Beispiel einer JBoss-AOP Interceptor-Klasse mit Advices	78
5.3	Teamklasse mit Rollendefinitionen	87
5.4	Möglichkeiten der Teamaktivierung einer Teamklassen-Instanz	88
5.5	Beispiel einer Teamklasse mit Vererbung und <i>callin</i> - bzw. <i>callout</i> -Bindungen	91
5.6	Compiler-Typisierung durch lowering-Translation in einer Rollenklasse	96
6.1	Ausschnitt aus der Teamklasse <i>CompiereTeam</i>	108
6.2	Ausschnitt einer generierten <i>CompiereFormMonitor</i> -Teamklasse	112
6.3	Ausschnitt aus der Teamklasse <i>CompiereWorkflowMonitor</i>	115
6.4	Ausschnitt aus der Teamklasse <i>CompiereWorkflowMonitor</i>	117
6.5	Beispiel einer Definition von Filternklassen (Pseudo-Java)	138

# 1 Einleitung

## 1.1 Motivation

Das Aufkommen der globalen Marktwirtschaft in den 60'er Jahren des letzten Jahrhunderts eröffnete den Unternehmen völlig neue Märkte und Möglichkeiten des Wachstums. Diese, noch immer anhaltende, Entwicklung führte zwangsläufig zu einer stärkeren Konzentration an Konkurrenzunternehmen, wobei diese den Druck auf ein Unternehmen erhöhen. Innovationen, Produktionskosten, Kontakt zum Kunden oder die geographische Nähe zu den Absatzmärkten sind heute wichtige Faktoren im ökonomischen Wettstreit der Unternehmen.

Seit den Anfängen der Globalisation werden Methodiken zur Planung oder Optimierung der Prozessketten verwendet, um in diesem Wettstreit konkurrenzfähig zu bleiben. Diese Entwicklung schreitet bis zum heutigen Tage voran. Die Einführung einer computergestützten Planung und Kommunikation führte zu einer weiteren Effizienzsteigerung und stellte damit einen Wettbewerbsvorteil dar, der die Entwicklung der sogenannten Unternehmenssoftware förderte. Der Begriff Unternehmenssoftware bezeichnet dabei eine Software, welche typischerweise verteilt lauffähig ist und eine oder mehrere betriebliche Bereiche abdeckt. Eine Gruppierung solcher Unternehmenssoftware bildet der Bereich zur Planung von Ressourcen, sei diese nun in menschlicher, materieller oder auch finanzieller Form. Allgemein bezeichnet man diese als *Enterprise Resource Planning-Software*.

Eine Software, welche in die betrieblichen Prozesse eingebunden ist, erfordert einen möglichst hohen Grad an *Wandlungsfähigkeit*, um sich schnell und effizient den dynamischen Prozessen anpassen zu können. Eine Erhöhung der Wandlungsfähigkeit von ERP-Systemen ist die Bestrebung zahlreicher Forschungsinstitute und ERP-System-Anbieter.

Dem gegenüber steht in der Softwareentwicklung der Wunsch nach mehr Flexibilität und Modularität. Eine Entwicklung die dieses Ziel verfolgt, ist die der aspektorientierten Programmierung (*AOSD - Aspect Oriented Software Development*). Diese Arbeit stellt den Versuch dar, die beiden genannten Elemente vorteilhaft zu verknüpfen. Dazu wird die Wandlungsfähigkeit eines Open-Source-ERP-Systems untersucht, um

anschließend die positiven Einflüsse bei der Verwendung von Aspektorientierung im Sinne der Wandlungsfähigkeit aufzuzeigen. Als Grundlage dient dabei das Java-basierende Open-Source-ERP-System *Compiere*. Unter Verwendung des aspektorientierten Programmiermodells *Object-Teams* wird dieses im weiteren Verlauf dieser Arbeit mit Aspekten angereichert werden. Die Erweiterung zielt hierbei auf eine Stärkung einzelner Kriterien der Wandlungsfähigkeit ab, um dadurch eine Steigerung der Gesamtwandlungsfähigkeit zu erreichen.

## 1.2 Struktur

Die vorliegende Arbeit ist unterteilt in die Kapitel:

- *Kapitel 2 - Enterprise Resource Planning*  
Dieses Kapitel gibt einen Überblick über den Begriff des Enterprise Resource Planning und erläutert dessen Aufbau und Bedeutung innerhalb eines Unternehmens.
- *Kapitel 3 - Wandlungsfähigkeit*  
Der Begriff der Wandlungsfähigkeit im Kontext von ERP-Systemen wird in diesem Kapitel näher erläutert.
- *Kapitel 4 - Compiere*  
Das Kapitel 4 gibt einen Einblick in den Aufbau und der Funktionsweise des ERP-Systems Compiere. Ebenso werden hier die verwendeten Technologien von Compiere, wie dem Applikations-Server JBoss, kurz erläutert.
- *Kapitel 5 - Aspektorientierte Softwareentwicklung*  
Hier werden die Konzepte der Aspektorientierung allgemein und speziell die für diese Arbeit relevanten, aspektorientierten Programmiermodelle vorgestellt und näher erläutert.
- *Kapitel 6 - Compiere Monitor*  
Kapitel 6 behandelt die Implementierung des Compiere Monitor. Dieser nutzt verschiedene Techniken der vorhergehenden Kapitel zur Einbettung unterschiedlicher Funktionalitäten in Compiere über die Verwendung von Aspekten.
- *Kapitel 7 - Ergebnis und Zusammenfassung*

Abschließend gibt Kapitel 7 eine Übersicht der erzielten Steigerung der Gesamtwandlungsfähigkeit des ERP-Systems Compiere und zeigt die dabei einwirkenden Vorteile der Verwendung von Aspektorientierung.

# 2 Enterprise Resource Planning

## 2.1 Begriffsklärung und Aufgabendefinition

Die Anforderungen heutiger Unternehmen an Bereiche wie Logistik, Buchhaltung oder Verkauf sind vielschichtig, wobei die Kernanforderungen bei allen in der Planung von Ressourcen, sei es in personeller, materieller oder finanzieller Form, liegt. Eine möglichst effiziente Verteilung der vorhandenen Ressourcen auf die Geschäftsprozesse wird somit zu einer wichtigen Aufgabe innerhalb eines Unternehmens, dem *Enterprise Resource Planning* (ERP). Für den Begriff ERP existieren in der einschlägigen Literatur verschiedenste Definitionen. Aus diesem Grund soll der Begriff für die vorliegende Arbeit explizit festgelegt werden. Dazu wird im Folgenden die Klärung der Aufgaben und Funktionen vorgenommen.

Die Aufgaben und Eigenschaften des ERP können zusammenfassend beschrieben werden als [WK01]:

- eine Sammlung unternehmensweiter Programme zum Management von Ressourcen im Zyklus von Angebot und Nachfrage
- die Fähigkeit den Kunden und Anbieter in einer kompletten Lieferkette miteinander zu verbinden
- die Verwendung bewährter Unternehmensprozesse zur Entscheidungsfindung zu unterstützen
- die Eigenschaft einen hohen Grad an Inter-funktioneller Integration zwischen den Bereichen Verkauf, Marketing, Herstellung, Betrieb, Logistik, Einkauf, Finanzen, Produktentwicklung und menschlicher Ressourcen zu bieten
- der Unterstützung des unternehmerischen Betriebs, speziell den Bereichen Kundendienst und Produktivität, auf einem hohen Niveau bei gleichzeitig geringeren Kosten und nötigen Lagerkapazitäten

- der Bereitstellung der Basis eines effektiven e-Commerce

In [Gro04] wird des Weiteren als Wesensmerkmal von ERP-Systemen die Integration verschiedener Funktionen, Aufgaben und Daten in ein gemeinsames Informationssystem genannt, wobei der wesentliche Vorteil in der Automatisierung von Abläufen und in einer Standardisierung von Prozessen liegt. Der Begriff Integration umfasst in diesem Zusammenhang zwei Dimensionen (Abb. 2.1). Zum einen gibt es die horizontale Integration. Sie beschreibt die abteilungs- bzw. funktionsübergreifenden Prozesse, wie z.B. einen gemeinsamen Prozess aus den Bereichen Vertrieb und Lagerhaltung. Senkrecht dazu entsteht die zweite Dimension, die der vertikalen Integration. Sie umfasst alle Ebenen der Aufgaben, welche auf einen Bereich angewandt werden können, so z.B. die Ebenen der Administration oder der Planung des gemeinsamen Wirkbereichs Vertrieb.

Der Umfang einer vollzogenen Integration wird von einem Integrationsgrad beschrieben. Dieser gibt an, wie viele betriebliche Funktionen in dem ERP-System vereinigt wurden. Typischerweise ist der Integrationsgrad abhängig von den betrachteten Bereichen, d.h. Bereiche wie die Logistik oder Produktion bilden Bereiche besonders hoher integrativer Möglichkeiten. Für diese Bereiche, in [Gro04] Integrationsinseln genannt, entstehen in Unternehmen oft eigenständige ERP-Systeme. Nur wenn alle diese Integrationsinseln in einem gemeinsamen ERP-System zusammengefasst sind, spricht man von einem *voll integrierten ERP-System*.

Die Einbettung von ERP innerhalb der von Unternehmen eingesetzten betrieblichen Informationssysteme zeigt Abbildung 2.2. Im Zentrum steht dabei die ERP-Komponente, wobei das *Supply Chain Management* (SCM) und das *Customer Relationship Management* (CRM) die Verbindung in die Bereiche ausserhalb des Unternehmens darstellen. Sie stellen die Verknüpfung vom innerbetrieblichen ERP-System zum ausserbetrieblichen Lieferanten bzw. Kunden dar. Diese beiden Begrifflichkeiten werden im Folgenden nun kurz beschrieben werden.

## Supply Chain Management

„Nothing Entirely New...Just a Significant Evolution“ [Hug03]

Der Begriff Supply Chain Management beschreibt eine Lieferkette, Versorgungskette oder auch eine unternehmensübergreifende Wertschöpfungskette. Diese Kette erstreckt sich dabei typischerweise vom Lieferanten bis hinüber zum Kunden und kann auch Querverbindungen zu verschiedenen Organisationen beinhalten. Sie dient der unternehmensübergreifenden Koordination der Material- und Informationsflüsse über

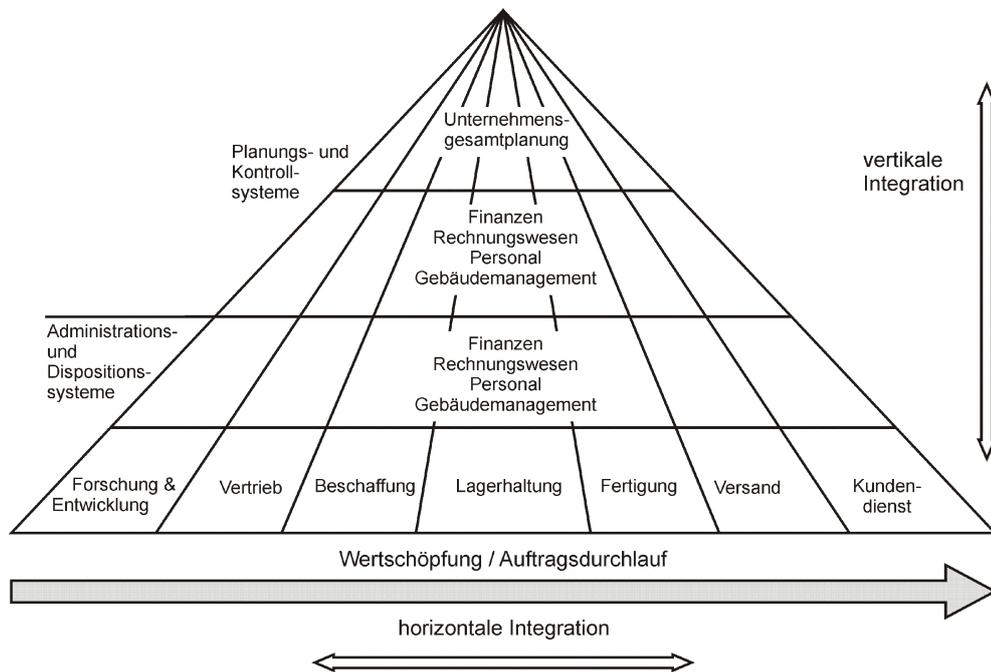


Abbildung 2.1: Horizontale und vertikale Integration betrieblicher Anwendungssysteme (nach [Gro04])

den gesamten Wertschöpfungsprozess hinweg, d.h. von der Rohstoffgewinnung über die einzelnen Veredelungsstufen bis hin zum Kunden. Ziel ist hierbei, den Gesamtprozess sowohl zeit- als auch kostenoptimal zu gestalten. [BD04][SRJ99]

## Customer Relationship Management

Die Ausrichtung auf den Kunden wird zunehmend zu einem kritischen Erfolgsfaktor innerhalb eines Unternehmens. Durch eine geringe Loyalität des Kunden gegenüber den Unternehmen sind diese aufgefordert, die Wünsche des Kunden zu erkennen und in ihrem Handeln zu berücksichtigen. Das Customer Relationship Management, im Weiteren auch als CRM bezeichnet, soll bei dieser Aufgabe helfen. Es umfasst die gesamte Interaktion eines Unternehmens mit bestehenden und zukünftigen Kunden während des gesamten Prozesses der Kaufentscheidung und des Besitzzeitraums. Das CRM beinhaltet damit die Planung, Durchführung, Kontrolle und Anpassung aller Unternehmensaktivitäten, wie Marketing oder Service, mit dem Ziel die Kundenbeziehungen zu optimieren. Um die Isolierung der einzelnen Unternehmensbereiche zu lockern, wird durch CRM eine globale Kundendatenbank etabliert, welche jedem

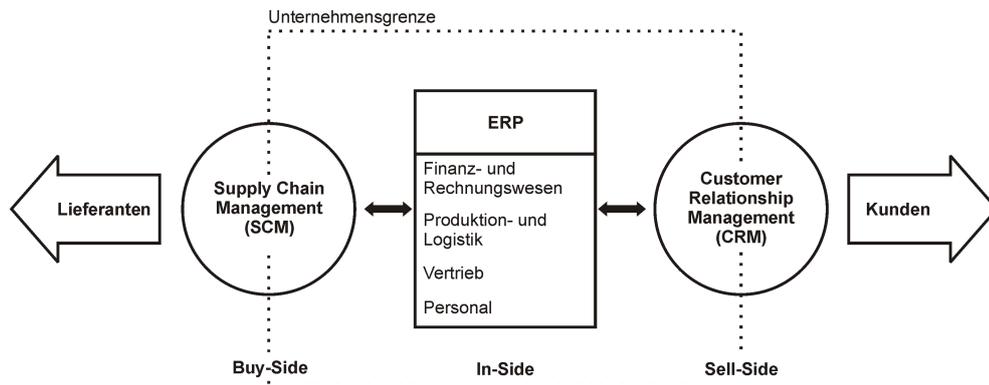


Abbildung 2.2: Aufgabenverteilung innerhalb betrieblicher Informationssysteme

Bereich Zugriff auf dieselben Kundendaten ermöglicht. Dies verringert die Informationsverluste bei separaten Kundendaten für jeden Bereich und erhöht damit die Effizienz des Kundenkontaktes. Typische Funktionalität eines CRM ist daneben unter anderem das Bereitstellen eines Helpdesk, Kundendienstes oder die Möglichkeit einer Kundenwertanalyse. [Gro04][HS00]

Letztendlich kann aber auch CRM allein keine Kundenfreundlichkeit gewährleisten, sondern dient nur zur Unterstützung derselbigen. Dieser Umstand wird in [Bre03] wie folgt dargelegt:

„...spielt die Technik im Gesamtkontext des Customer Relationship Managements keineswegs die dominierende Rolle. Denn eines ist klar: Beziehungen werden nicht von Computern aufgebaut und gepflegt, sondern einzig und allein von Menschen... Allerdings hängt der Erfolg neben Mitarbeitern und Technik auch von der organisatorischen Gesamtausrichtung eines Unternehmens ab. Kundenfeindlich strukturierte Geschäftsprozesse werden allein durch guten Willen und moderne Technik nicht zu kundenfreundlichen.“

## 2.2 Entstehungsgeschichte

Das Bedürfnis nach besseren Methoden zur Material- und Komponentenbestellung führte um 1960 zur Entwicklung des *Material Requirement Planning (MRP)*. Primär ging es dabei um die Erfüllung der grundlegenden Fragen bei der Herstellung eines Produkts [WK01]:

- Was möchte ich herstellen?