



UNIVERSITÄT
KOBLENZ · LANDAU

Institut für Wirtschafts-
und Verwaltungsinformatik



FB 4

Informatik

Künftige Anforderungen an ERP-Systeme: Deutsche Anbieter im Fokus

Norbert Frick

Nr. 11/2008

**Arbeitsberichte aus dem
Fachbereich Informatik**

Die Arbeitsberichte aus dem Fachbereich Informatik dienen der Darstellung vorläufiger Ergebnisse, die in der Regel noch für spätere Veröffentlichungen überarbeitet werden. Die Autoren sind deshalb für kritische Hinweise dankbar. Alle Rechte vorbehalten, insbesondere die der Übersetzung, des Nachdruckes, des Vortrags, der Entnahme von Abbildungen und Tabellen – auch bei nur auszugsweiser Verwertung.

The “Arbeitsberichte aus dem Fachbereich Informatik“ comprise preliminary results which will usually be revised for subsequent publication. Critical comments are appreciated by the authors. All rights reserved. No part of this report may be reproduced by any means or translated.

Arbeitsberichte des Fachbereichs Informatik

ISSN (Print): 1864-0346

ISSN (Online): 1864-0850

Herausgeber / Edited by:

Der Dekan:
Prof. Dr. Zöbel

Die Professoren des Fachbereichs:

Prof. Dr. Bátori, Prof. Dr. Beckert, Prof. Dr. Burkhardt, Prof. Dr. Diller, Prof. Dr. Ebert, Prof. Dr. Furbach, Prof. Dr. Grimm, Prof. Dr. Hampe, Prof. Dr. Harbusch, Jun.-Prof. Dr. Hass, Prof. Dr. Krause, Prof. Dr. Lämmel, Prof. Dr. Lautenbach, Prof. Dr. Müller, Prof. Dr. Oppermann, Prof. Dr. Paulus, Prof. Dr. Priese, Prof. Dr. Rosendahl, Prof. Dr. Schubert, Prof. Dr. Staab, Prof. Dr. Steigner, Prof. Dr. Troitzsch, Prof. Dr. von Kortzfleisch, Prof. Dr. Walsh, Prof. Dr. Wimmer, Prof. Dr. Zöbel

Kontaktdaten der Verfasser

Norbert Frick
Institut für Wirtschafts- und Verwaltungsinformatik
Fachbereich Informatik
Universität Koblenz-Landau
Universitätsstraße 1
D-56070 Koblenz
EMail: norbert.frick@uni-koblenz.de

Vorwort und Danksagung

ERP-Systeme stellen einen wichtigen Kompetenzbereich der Forschungsgruppe Betriebliche Anwendungssysteme (FG BAS) am Institut für Wirtschafts- und Verwaltungsinformatik (IWVI) der Universität Koblenz-Landau dar. In enger Zusammenarbeit mit Partnern aus Wirtschaft und Wissenschaft vermittelt die Forschungsgruppe sowohl Grundlagenforschung als auch praxisbezogenes Wissen an Studierende und Unternehmen.

Der vorliegende Arbeitsbericht „Künftige Anforderungen an ERP-Systeme: Deutsche Anbieter im Fokus“ beleuchtet auf vier Ebenen die *Architektur*, die verwendete *Technologie*, die praktischen *Einsatzmöglichkeiten* und die allgemeine *Marktsituation* aktueller ERP-Systeme auf dem deutschsprachigen Markt. Ein besonderer Dank geht in diesem Zusammenhang an Peter Herzog (zum damaligen Zeitpunkt Marketingleiter der Bison AG), der die Idee einer solchen Untersuchung hatte und die FG BAS mit deren Durchführung betraute. Die Studie wurde vom Sommer 2007 bis zum Frühling 2008 von Norbert Frick im Rahmen seiner Diplomarbeit im Fach Informatik durchgeführt. Die Auswertung der Ergebnisse wurde inhaltlich und wissenschaftlich betreut von Prof. Dr. Petra Schubert. Die Arbeit entstand aufgrund von Literaturrecherchen, schriftlichen Umfragen und Telefoninterviews mit ERP-Anbietern.

Die Forschungsgruppe von Prof. Schubert ist auf die Durchführung solcher Studien spezialisiert. In den letzten Jahren entstanden mehr als zehn empirische Untersuchungen zum Einsatz und zur Anwendung von Informationstechnologie im deutschsprachigen Raum (z.B. Netzreport-Reihe, ERP-, Procurement- und Billing-Studien). Die Wirtschaftspartner profitieren bei diesen Initiativen von der sachorientierten und unabhängigen Sicht der FG BAS, vom Know-how und den Kapazitäten bei der Durchführung der Untersuchung, bei der Systematisierung und Aufbereitung von Inhalten und bei der Vermittlung von Wissen über firmenunabhängige Kanäle. Die Kooperationspartner in den beteiligten Unternehmen tragen mit ihrem persönlichen Engagement zum Gelingen und zur Relevanz der Studie bei.

An dieser Stelle geht ein Dank an Norbert Frick, der so freundlich war, die Ergebnisse seiner Diplomarbeit in einer umfassenden Form als Arbeitsbericht des Fachbereichs Informatik zu veröffentlichen. Ebenfalls Dank gilt den 32 Teilnehmern der Studie, die durch ihre Antworten diese Publikation überhaupt ermöglicht haben, indem sie ihr Wissen und ihre Informationen zur Verfügung gestellt haben.

Koblenz, im August 2008

Petra Schubert

Inhaltsverzeichnis

| | |
|--|------------|
| Vorwort und Danksagung | i |
| Inhaltsverzeichnis | iii |
| Abbildungs- und Tabellenverzeichnis..... | vii |
| 1 Einleitung | 1 |
| 1.1 Historische Entwicklung..... | 1 |
| 1.2 Motivation | 2 |
| 1.3 Aufbau der Arbeit | 3 |
| 2 Forschungsmethodik | 5 |
| 2.1 Von der Problemstellung zum Modell | 6 |
| 2.2 Vom theoretischen Modell zur Erhebungsmethode..... | 8 |
| 2.3 Qualitative Forschung..... | 8 |
| 2.4 Das Experteninterview..... | 9 |
| 2.5 Die Stichprobe..... | 9 |
| 2.6 Der Fragebogen | 9 |
| 2.7 Gliederung des Fragebogens | 10 |
| 2.8 Die Auswertung | 17 |
| 2.9 Prognose und kritische Bewertung | 18 |
| 2.10 Befragte Unternehmen | 18 |
| 3 Theoretische Grundlagen | 23 |
| 3.1 Grundbegriffe | 23 |
| 3.1.1 Betriebliche Anwendungssysteme..... | 23 |
| 3.1.2 Enterprise Resource Planning - ERP | 25 |
| 3.1.3 Business Software..... | 26 |
| 3.1.4 Standardsoftware vs. Individualsoftware | 27 |
| 3.1.5 Releasefähigkeit..... | 27 |
| 3.1.6 Customizing..... | 27 |
| 3.1.7 Legacy-System..... | 27 |
| 3.2 Architektur | 28 |
| 3.2.1 Planungsebene..... | 28 |
| 3.2.2 Systemarchitektur..... | 33 |
| 3.2.3 Modulararchitektur | 46 |
| 3.2.4 Integration..... | 49 |
| 3.3 Technologie..... | 51 |
| 3.3.1 Technologie - intern..... | 51 |

| | | |
|----------|--|------------|
| 3.3.2 | Technologie – extern | 53 |
| 3.4 | Betreibermodelle..... | 60 |
| 3.4.1 | In-House | 60 |
| 3.4.2 | Application Service Providing - ASP..... | 61 |
| 3.4.3 | Software as a Service - SaaS..... | 61 |
| 3.5 | Markt | 62 |
| 3.5.1 | KMU | 62 |
| 3.5.2 | Branchen | 63 |
| 3.5.3 | Anbieter | 64 |
| 4 | Auswertung..... | 65 |
| 4.1 | Architektur | 65 |
| 4.1.1 | Planungsebene..... | 65 |
| 4.1.2 | Systemarchitektur..... | 72 |
| 4.1.3 | Modularchitektur | 79 |
| 4.1.4 | Integration..... | 84 |
| 4.2 | Technologie | 88 |
| 4.2.1 | Systemintern..... | 88 |
| 4.2.2 | Systemextern..... | 95 |
| 4.3 | Betreibermodell..... | 98 |
| 4.4 | Markt | 102 |
| 4.4.1 | Anwender | 102 |
| 4.4.2 | Anbieter | 107 |
| 5 | Prognose - Ein Blick in die Zukunft..... | 115 |
| 5.1 | Ein mögliches Szenario | 115 |
| 5.2 | Architektur | 116 |
| 5.2.1 | Planungsebene..... | 116 |
| 5.2.2 | Systemarchitektur..... | 118 |
| 5.2.3 | Modularchitektur | 120 |
| 5.2.4 | Integration..... | 121 |
| 5.3 | Technologie | 122 |
| 5.3.1 | Systemintern..... | 122 |
| 5.3.2 | Systemextern..... | 123 |
| 5.4 | Betreibermodell..... | 125 |
| 5.5 | Markt | 126 |
| 5.5.1 | Anwender | 126 |
| 5.5.2 | Anbieter | 127 |
| 6 | Kritische Zusammenfassung und Ausblick..... | 129 |

| | | |
|-------|---|----------|
| 6.1 | Ausblick: Architektur | 129 |
| 6.1.1 | Ausblick: Planungsebene | 129 |
| 6.1.2 | Ausblick: Systemarchitektur | 129 |
| 6.1.3 | Ausblick: Funktionsarchitektur..... | 130 |
| 6.1.4 | Ausblick: Integration | 130 |
| 6.2 | Ausblick: Technologie..... | 131 |
| 6.2.1 | Ausblick: Technologie - Systemintern..... | 131 |
| 6.2.2 | Ausblick: Technologie - Systemextern..... | 131 |
| 6.3 | Ausblick: Betreibermodell | 132 |
| 6.4 | Ausblick: Markt | 132 |
| 6.4.1 | Ausblick: Anwender | 132 |
| 6.4.2 | Ausblick: Anbieter..... | 133 |
| 6.5 | Fazit | 133 |
| | Literaturverzeichnis..... | 1 |
| | Anhang | 8 |

Abbildungs- und Tabellenverzeichnis

| | | |
|----------------|--|----|
| Abbildung 2.1 | Von der Fragestellung zum Modell, Quelle: [Mayer, 2004, S. 29]. | 5 |
| Abbildung 2.2 | Vier grundlegende Aspekte eines ERP-Systems, beispielhaft skizziert. | 7 |
| Abbildung 2.3 | Mögliche architekturelle Ausprägungen. | 11 |
| Abbildung 2.4 | Eingesetzte Module. | 12 |
| Abbildung 2.5 | Beispiel: Kernfunktionalitäten in Ihrem ERP | 13 |
| Abbildung 2.6 | Beispiel eines Betreibermodells Ihres ERP | 15 |
| Abbildung 2.7 | Übersicht der interviewten Unternehmen, Teil I. | 19 |
| Abbildung 2.8 | Übersicht der interviewten Unternehmen, Teil II. | 20 |
| Abbildung 2.9 | Übersicht der interviewten Unternehmen, Teil III. | 21 |
| Abbildung 2.10 | Übersicht der interviewten Unternehmen, Teil IV | 22 |
| Abbildung 3.1 | Kommunikationskanäle | 23 |
| Abbildung 3.2 | Klassifikation nach Verwendungszweck | 24 |
| Abbildung 3.3 | Begriffssystematik zu Business Software, Quelle: [Wölfle & Schubert, 2006]. | 26 |
| Abbildung 3.4 | Bidirektionale Abhängigkeiten, Quelle: Wölfle & Schubert, 2006 | 29 |
| Abbildung 3.5 | Prozess-Qualitätsstufen, Quelle: Wölfle, 2006, S. 10 | 30 |
| Abbildung 3.6 | Monolithische Architektur, Quelle: Overhage et al., 2007 | 34 |
| Abbildung 3.7 | Client-Server Architektur, Quelle: Overhage et al., 2007 | 35 |
| Abbildung 3.8 | Verschiedene Client-Ausprägungen, Quelle: Overhage et al., 2007 | 36 |
| Abbildung 3.9 | Workflow Spezialisierung, Quelle: Overhage et al., 2007 | 37 |
| Abbildung 3.10 | Vermittler Spezialisierung, Quelle: Overhage et al., 2007 | 38 |
| Abbildung 3.11 | Middleware, Quelle: Overhage et al., 2007 | 39 |
| Abbildung 3.12 | Beispielhafter Request-Aufruf eines Clients an einen Server unter CORBA | 40 |
| Abbildung 3.13 | Schichten-Sichtweise auf J2EE | 41 |
| Abbildung 3.14 | Formen der Integration | 49 |
| Abbildung 3.15 | Ein Klassifikationsschema von elektronischen Standards nach einer Studie von Berlecon Research, vgl. [Quantz und Wichmann, 2003] | 54 |
| Abbildung 3.16 | EDI-Standards im Rahmen der drei Ebenen | 55 |
| Abbildung 3.17 | Web Services im Überblick | 58 |
| Abbildung 3.18 | Klassifikation von kleinen und mittelgroßen Betrieben der Europäischen Union, Stand 2003 | 63 |
| Abbildung 4.1 | Graphische Umsetzung von Prozessen | 66 |
| Abbildung 4.2 | Automatisierte Umsetzung der Prozesse in Code | 68 |

| | | |
|----------------|---|-----|
| Abbildung 4.3 | Verwendung der Prozesse beim Kunden | 69 |
| Abbildung 4.4 | Branchenstandards im Einsatz | 70 |
| Abbildung 4.5 | Anpassungsmöglichkeiten | 71 |
| Abbildung 4.6 | Umgesetzte Konzepte | 73 |
| Abbildung 4.7 | Verwendete Architekturen | 74 |
| Abbildung 4.8 | Verwendete Clients | 75 |
| Abbildung 4.9 | Verbreitung der Plattformunabhängigkeit..... | 76 |
| Abbildung 4.10 | Verwendete Middleware | 77 |
| Abbildung 4.11 | Browsertechnologien im Einsatz..... | 78 |
| Abbildung 4.12 | Besonders im sekundären Bereich Fremdanbieter benötigt | 80 |
| Abbildung 4.13 | Einsatz von CRM und BI in den Systemen | 81 |
| Abbildung 4.14 | Eigenentwicklungen im CRM/BI Bereich..... | 82 |
| Abbildung 4.15 | Releasefähigkeit größtenteils vorhanden..... | 83 |
| Abbildung 4.16 | Anbindungsmöglichkeiten an Fremdsysteme auf Applikations- bzw. Datenbankebene | 85 |
| Abbildung 4.17 | Sehr verschiedene Angaben zu Integrationsmöglichkeiten..... | 86 |
| Abbildung 4.18 | Nur etwa 10 Prozent der Systemlandschaften sind modular aufgebaut | 87 |
| Abbildung 4.19 | Möglichkeiten für Betriebssysteme | 89 |
| Abbildung 4.20 | Relationale Datenbanken als Grundlage von ERP-Systemen..... | 90 |
| Abbildung 4.21 | Java bei mehr als 50 Prozent im Einsatz | 91 |
| Abbildung 4.22 | Sechzehn unterschiedliche Angaben zur Programmiersprache..... | 92 |
| Abbildung 4.23 | Eclipse und IntelliJ als Java-Entwicklungsumgebung weit verbreitet | 93 |
| Abbildung 4.24 | Zwei Drittel sehen OS als wichtiges Entwicklungswerkzeug an | 94 |
| Abbildung 4.25 | Web Services als weitflächig genutztes Werkzeug | 96 |
| Abbildung 4.26 | Klassische Browser öfter im Einsatz als Citrix | 97 |
| Abbildung 4.27 | ASP und SaaS holen auf In-House Lösungen auf | 99 |
| Abbildung 4.28 | Kombination aus Direkt- und Partnervertrieb dominant..... | 100 |
| Abbildung 4.29 | Direkte Wartung überwiegt Kombinationslösung bzw. reinen Partnervertrieb..... | 101 |
| Abbildung 4.30 | Abdeckung aller Unternehmensgrößen und -klassen gegeben..... | 103 |
| Abbildung 4.31 | Technologischer Vorsprung als Unterscheidungsmerkmal Nummer 1 | 104 |
| Abbildung 4.32 | Kunden haben bei vielen Anbietern Einfluss auf die Systemgestaltung..... | 106 |
| Abbildung 4.33 | Das produzierende Gewerbe wird bei den Befragten am meisten unterstützt | 108 |
| Abbildung 4.34 | OS wird nicht als ernsthafter Konkurrent angesehen..... | 110 |

| | | |
|----------------|--|-----|
| Abbildung 4.35 | Wenige mit aggressivem Expansionskurs, eher stetiges Wachstum im Vordergrund | 112 |
| Abbildung 4.36 | Konsolidierung des Marktes und neue Technologie-Impulse sehen die Befragten voraus | 113 |

1 Einleitung

1.1 Historische Entwicklung

Seit Mitte der 50er Jahre des vergangenen Jahrhunderts spricht man von der so genannten Software Industrie [Campbell-Kelly, 2004, S. 4]. Begonnen hatte sie mit Firmen, die individuelle Programme schrieben, entweder für große Unternehmen oder damals noch überwiegend Regierungen (z.B. Sage, U.S. Luftabwehrprogramm). Diese Programme wurden für einen bestimmten Zweck geschrieben und waren nur in diesem speziellen Umfeld nutzbar. Die Firmen, die solche Software produzierten, besaßen langfristige Verträge und wurden als so genannte „Software Contractors“ bezeichnet [Campbell-Kelly, 2004, S. 4]. Der Preis betrug meist mehr als eine Million Dollar je Ausführung.

Anfang der 60er Jahre verbesserte sich die Rechenleistung und neue Systeme konnten preisgünstiger erworben werden. Dadurch war die Möglichkeit einer öfters einsetzbaren Software gegeben. Neue Firmen entstanden, die nicht mehr für einzelne Vertragspartner, sondern für eine Gruppe von Unternehmen (z.B. Finanzsektor) Programme erstellten. Dies war die Geburtsstunde der heute bekannten Unternehmenssoftware [Campbell-Kelly, 2004, S. 6]. ADR (Applied Data Research) und Informatics waren Vorreiter, bekanntere Namen wie SAP (1972) oder Oracle (1977) folgten.

Die dritte Gruppe von Softwareherstellern entstand Ende der 70er Jahre, als der PC Einzug in die Wohnungen von Privatpersonen hielt. Die Masse an PCs verlangte auch eine massenhafte Produktion von Software (Betriebssystem, Textverarbeitung, Spiele, etc.). Software Arts (1979), Lotus (1982) oder Microsoft (1975) kann man beispielhaft als Unternehmen nennen, die der Nachfrage nachkamen [Campbell-Kelly, 2004, S. 7]. Diese Einteilung ist jedoch nur als Richtlinie zu verstehen. Die Grenze zwischen den einzelnen Herstellern kann nicht klar gezogen werden, denn z.B. Microsoft, ursprünglich als Produzent von Massensoftwarepaketen am Markt bekannt, hat erst vor vergleichsweise kurzer Zeit begonnen, Unternehmen mit Business Lösungen anzusprechen. Tatsächlich wurden früher und werden auch heute noch von den Anbietern Fremdprodukte eingekauft oder mit in ihre Installation einbezogen.

Zu Beginn der 80er Jahre gab es entweder branchenübergreifende Lösungen für Unternehmen (z.B. Rechnungswesen, Personalmanagement, etc.) oder branchenspezifische Lösungen (z.B. Chemie, Versicherungen, herstellende Gewerbe etc.). Besonders die branchenspezifischen Lösungen waren nur auf diese eine Nische ausgerichtet, in der sie angewendet wurden. Eine weitere Charakteristik der damaligen Software war die funktional ausgerichtete Sichtweise der Hersteller auf ein Unternehmen [Allweyer, 2005, S. 13]. Allweyer beschreibt es folgendermaßen:

„Die erstellte Software bediente nur die jeweiligen Abteilungen, in die ein Unternehmen aufgrund arbeitsrelevanter Effizienzgründe eingeteilt war“ [Allweyer, 2005, S. 13].

Die Folge war eine heterogene Anwendungslandschaft, in der Daten redundant in mehreren Abteilungen vorgehalten wurden und eine direkte Zusammenarbeit der Softwarepakete untereinander kaum möglich war.

Die nächste Stufe der Entwicklung wurde in breitem Umfang von SAP eingeläutet, welche mit der auf Client-Server ausgerichteten ERP-Lösung R/3 den weltweiten Markt eroberte [Campbell-Kelly, 2004, S. 195]. Diese Software war customizable, d.h. für jedes denkbare Unternehmen anpassbar, und integrierte überdies die verschiedenen Funktionen aller

Systeme. So konnte man mit einer Komplettlösung, so lautete zumindest das Versprechen [Rettig, 2007, S. 22], auf diese unüberschaubare Vielzahl unterschiedlicher Produkte verschiedener Anbieter verzichten und dadurch mögliche Konflikte zwischen den Anwendungen vermeiden.

Mit dieser neuen Softwaregeneration wandelte sich der Blick auf ein Unternehmen und seine Wertschöpfung von der funktionsorientierten Sicht hin zur prozessorientierten Sicht [Allweyer, 2005, S. 20]. Das bedeutete, dass z.B. empfangene Aufträge über die Abteilungsgrenzen hinweg betrachtet und betreut wurden. Eine einheitliche Datenbasis vermied überflüssige Redundanz und erlaubte einen unternehmensweit einheitlichen Blick auf die relevanten Daten. Es wurde ein Grad der Interoperabilität erreicht, der mit den alten individuell angefertigten Lösungen nur unter großem Aufwand zu erreichen war [Hitt et al., 2002, S. 72].

Auf ein Unternehmen allein bezogen war ein solches ERP-System theoretisch sehr gut geeignet, um die alten „Legacy-Systeme“, (Softwaresysteme, die vor der Einführung neuer Applikationen bereits existierten, vgl. Kapitel 3.1.7) zu ersetzen. In der Praxis existierten jedoch aus unterschiedlichen Gründen, z.B. des Investitionsschutzes oder einer Unternehmensfusion, viele verschiedene Anwendungssysteme in einem Unternehmen, evtl. sogar auf mehrere Standorte verteilt. Die in sich geschlossenen ERP-Systeme konnten in dieser heterogenen Landschaft ihre Aufgabe nicht erfüllen und wurden ebenfalls ein Teil der Legacy-Systeme [Rettig, 2007, S. 23].

Mit der Neuorientierung der Unternehmen weg von der klassischen produktzentrierten Sicht hin zur kundenorientierten Sicht [Bach & Österle, 2000] und der technologischen Weiterentwicklung des Internets eröffneten sich weitere Kommunikationskanäle, über die Anwendungssysteme miteinander in Verbindung treten konnten. 1999 wurden etwa 22 Prozent der Investitionen der Unternehmen für die IT aufgebracht, etwa 70-80 Prozent fielen dabei allein für die Aufrechterhaltung des Betriebes an, wobei etwa 75 Prozent der versuchten ERP-Einführungen als fehlerhaft bezeichnet werden konnten [Rettig, 2007, S. 21 & 23].

In der gegenwärtigen Situation scheinen neue Ansätze wie z.B. Service-Orientierung (vgl. Kapitel 3.2.2.7) oder SaaS (vgl. Kapitel 3.4.3) einen Ausweg aus der komplizierten Lage zu versprechen. Mit welchen Mitteln und ob dies überhaupt gelingen kann und wird, soll in den folgenden Kapiteln näher beleuchtet werden.

1.2 Motivation

Service orientierte Architektur, Web Services, XML oder Modularität sind Begriffe, die seit einigen Jahren in der Welt der Business Software durchaus als Schlagwörter galten und teilweise noch immer gelten. Die Unternehmenssoftware von heute sollte am besten serviceorientiert und modular aufgebaut sein, via Web Services kommunizieren und der überwältigenden Vielzahl von unterschiedlichen Dokumentenstandards mit XML ein Ende setzen. So könnte eine pauschale Anforderung durchaus lauten. Die Frage, die sich dabei stellt, ist jedoch viel weitreichender. Ist eine Unternehmenslösung zukunftsfähig, wenn sie diese Anforderung nicht erfüllt?

Die Frage ist sehr schwer zu beantworten, denn trotz der markant klingenden Stichwörter sind viele Systeme, die sich im aktiven Einsatz befinden, älter als 6 Jahre [Schmitt, 2007] und haben kaum oder gar keine der angestrebten Prinzipien umgesetzt. Trotzdem funktionieren sie und erbringen dem Unternehmen die benötigten Dienstleistungen. Gerade bei kleinen und mittelständischen Unternehmen ist die Einführung eines neuen Systems bzw. die Migration

des alten auf eine neue Plattform mit hohen Kosten und natürlich den entsprechenden Risiken verbunden.

Müsste man sich also für ein neues System entscheiden, sollte man genau wissen, was das System leisten kann und wie es die Leistung erbringen soll. Hier werden schnell weitere Stichwörter in die Diskussion mit eingebracht: Application Service Provider (ASP) oder Software as a Service (SaaS). Warum Software mit der Hardware kaufen, wenn man die Software nicht einfach mieten kann? Ist dies zukunftsfähig? Und kann diese Mietlösung überhaupt die Leistung erbringen, die man von alten Systemen her bereits kennt?

Zuletzt spielt natürlich auch die Zukunftsfähigkeit der Anbieter solcher Komplettlösungen eine Rolle. In letzter Zeit wurden von den großen Konzernen wie SAP, Oracle oder Microsoft Aufkäufe getätigt, die das Portfolio der Softwareanbieter erweitern sollen [Alexander, 2007]. Die Sicherheit in der Zusammenarbeit mit dem Softwarepartner kann somit durch eventuelle Übernahmen von Dritten gefährdet sein. Wird ein IT-Partner womöglich morgen keine Updates mehr für die genutzte Software zur Verfügung stellen?

Dies sind Fragen, die Kunden der Softwarehersteller von Business Software beschäftigen und somit auch die Softwarehersteller selbst. Die eigene Weiterentwicklung in der Software ist sowohl vom inneren Willen her, Neues zu schaffen, als auch vom äußeren Druck her, Neues umzusetzen, getrieben. Die Konkurrenzfähigkeit zu seinen Mitbewerbern beizubehalten und auszubauen, ist das Bestreben jedes Anbieters. Innovative Systemarchitekturen, neue Technologien, ausgefeilte Business Modelle oder die Durchsetzungsfähigkeit am Markt - man versucht sich in dem jeweiligen Gebiet, von seinem Konkurrenten abzugrenzen. Dieser Wettbewerb führt dazu, dass Systeme nicht in ihrer alten Form bestehen bleiben, sondern zwingt förmlich zur Neu- bzw. Umgestaltung.

Neue Trends und Entwicklungen in den vier oben genannten Bereichen aufzuzeigen, ist das Bestreben dieser Arbeit. Die zugrunde liegende Architektur bildet die konzeptionelle Basis, die mit Hilfe technischer Werkzeuge umgesetzt wird. Die Inbetriebnahme für den Kunden und die harte Konkurrenz am Markt bilden die wirtschaftliche Komponente. In allen vier Bereichen sehen sich die Hersteller mit immer wieder neuen oder neuartigen Herausforderungen konfrontiert, die sie im Rahmen ihrer Strategie bewerten und gegebenenfalls selbst umsetzen müssen. Anwendungen besitzen inzwischen einen durchschnittlichen Lebenszyklus von etwa vier Jahren [Masak, 2005], der von Datenbanken und Middleware ist noch geringer. Die Notwendigkeit einer flexiblen aber auch beständigen Lösung ist gegeben. Ob sich diese beiden, auf den ersten Blick inkompatiblen Ansätze miteinander verbinden lassen, wird die Zukunft zeigen.

SAP hat erst vor kurzem eine neue Lösung vorgestellt: "Business by Design" [Jung, 2007], eine On-Demand Lösung, die eine Vor-Ort Installation unnötig macht. Diese Lösung bedeutet eine neue architekturelle, technologische und wirtschaftliche Ausrichtung hin zum Markt für kleine und mittelständische Unternehmen. Ob sich andere Anbieter dieser Entwicklung anschließen wollen oder die Initiative zum Scheitern verurteilt ist, soll unter anderem auf den folgenden Seiten näher beleuchtet werden.

1.3 Aufbau der Arbeit

Als Einführung in die Thematik "Business Software" und im Speziellen ERP-Systeme wurde bereits die Historie von Unternehmenssoftware erläutert und auf die aktuell in der Entwicklung diskutierten Themen als Ausgangspunkt der in dieser Arbeit geleisteten Forschungsarbeit eingegangen (Kapitel 1.1 und 1.2).

Kapitel 2 beschreibt die Forschungsmethodik zur Erstellung der Arbeit. Diese umfasst aufgrund des empirischen Charakters neben der theoretischen Grundlagenermittlung eine Beschreibung der qualitativen Forschungsmethode, die Entwicklung des für die Interviews verwendeten Fragebogens, eine Beschreibung der Durchführung der Interviews so wie eine detaillierte Beschreibung der Interviewpartner.

Das darauf folgende Kapitel 3 legt die theoretischen Grundlagen dar, auf deren Basis die Interviews geführt und ausgewertet wurden. Dabei orientiert sich die Kapitelstruktur an den in Kapitel 2.1 vorgestellten vier Bereichen Architektur, Technologie, Betreibermodell und Markt, die sich während der Grundlagenforschung als die vier zentralen Aspekte eines ERP-Systems herausgestellt haben. Die wichtigen Forschungsfragen können hiermit beantwortet werden.

Die Auswertung der Interviewergebnisse, die sich in Kapitel 4 wiederfindet, orientiert sich ebenfalls an diesen vier Aspekten. Dort werden die Angaben der Befragten unter Berücksichtigung der in Kapitel 3 vorgestellten Grundlagen zu jedem Aspekt detailliert bewertet.

Die so gewonnenen Erkenntnisse dienen im Kapitel 5 als Grundlage zur Erstellung einer Prognose, die zukünftige Entwicklungen in den jeweiligen vier Bereichen Architektur, Technologie, Betreibermodell und Markt beschreibt. Dazu werden parallel aktuelle Entwicklungen vorgestellt, die den Blick in die Zukunft thematisch abrunden sollen. Die meisten dieser Informationen stammen aus kürzlich erschienenen Zeitschriftenartikel bzw. Journals, deren Aktualität zu diesem Zeitpunkt sehr hoch, aber deren Wirkung auf zukünftige Entwicklungen noch schwer einzuschätzen ist.

Das letzte Kapitel 6 setzt sich zum Abschluss kritisch mit denen in dieser Arbeit gewonnen Erkenntnissen auseinander. In diesem Zusammenhang werden Forschungsmethodik, die Befragung der Unternehmen aber auch die Resultate noch einmal auf potentielle Verbesserungsmöglichkeiten für zukünftige Forschungen in diesem Themengebiet untersucht.

Anmerkungen:

- Zwischendurch finden sich immer wieder Zitate, welche bei den Befragungen als äußerst aussagekräftig bzw. auch polarisierend charakterisiert wurden und als zusätzliche Untermauerung der Auswertung dienen sollen. Die Zitate werden im Rahmen der Arbeit anonym behandelt. Jedes Zitat ist mit einem Buchstaben/Nummern-Code versehen, um den Referenten der Arbeit die Nachvollziehbarkeit der wissenschaftlichen Aussage zu ermöglichen. Die Urheber der Aussagen werden jedoch zum Zwecke des Datenschutzes nicht veröffentlicht.
- Die befragten Unternehmen werden zur besseren Übersicht und Lesbarkeit während der folgenden Kapitel nur mit einem Eigennamen erwähnt. Beispiel: „SHD Holding GmbH“ erscheint nur als „SHD“ (vgl. Kapitel A.3).
- Da einige Angaben der befragten Unternehmen spezifische Aspekte ihres Systems betrafen, werden die entsprechenden Beschreibungen am Ende des jeweiligen Protokolls aufgeführt, anstatt sie im normalen Text näher zu erläutern.

2 Forschungsmethodik

Im folgenden Kapitel wird zunächst die zugrunde liegende Forschungsmethodik dieser Arbeit vorgestellt und die genaue Vorgehensweise zur Erstellung erläutert. Des Weiteren wird die Entwicklung des Fragebogens näher beleuchtet und zuletzt eine kurze Übersicht der Interviewpartner hinsichtlich ihrer Größe und Branchenzugehörigkeit mit zusätzlichen unternehmensspezifischen Informationen zu einer besseren Orientierung gegeben.

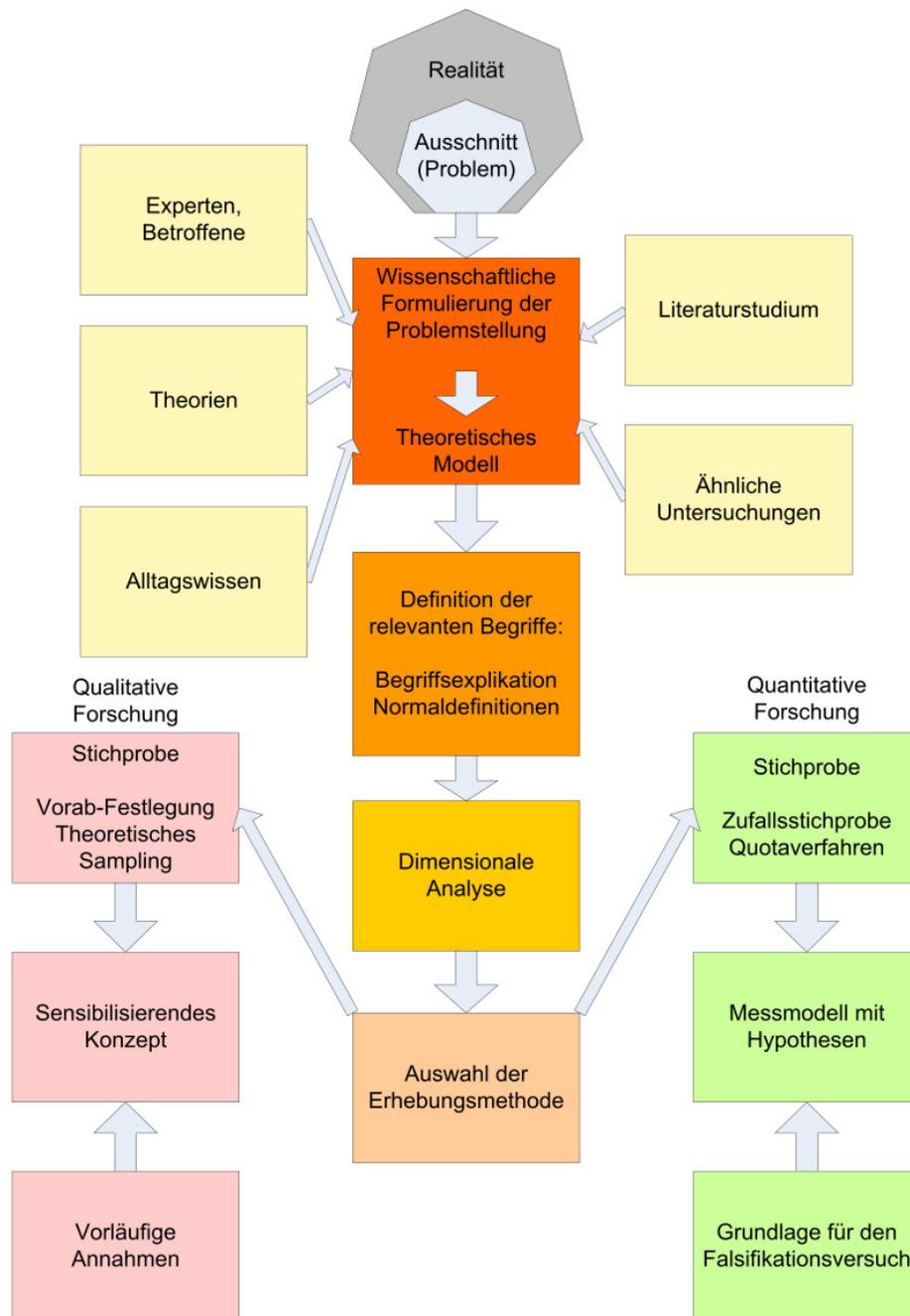


Abbildung 2.1

Von der Fragestellung zum Modell, Quelle: [Mayer, 2004, S. 29].

Das Vorgehen bei der Ermittlung der theoretischen Grundlagen und der anschließenden Durchführung der Befragungen orientierte sich nach der von Mayer [Mayer, 2004, S. 29] vorgestellten wissenschaftlichen Forschungsmethodik (vgl. Abbildung 2.1).

Dabei wird die Problemstellung im Rahmen des zu betrachtenden Realitätsausschnitts gemäß der zur Verfügung stehenden Informationsquellen (Expertenwissen, Theoretische Abhandlungen, Alltagswissen, Literaturquellen und ähnlichen Untersuchungen) wissenschaftlich formuliert und daraus ein theoretisches Modell abgeleitet. Sind die notwendigen Definitionen zum Verständnis der Problemstellung erfasst und der Umfang der Untersuchung festgelegt, stehen zwei Erhebungsmethoden für die weitere Forschung zur Verfügung: die qualitative und quantitative Forschung.

Die quantitative Forschung ermittelt falsifizierbares Wissen, dass basierend auf der theoretischen Vorarbeit als wahr oder falsch bewertet werden kann. Die qualitative Forschung ermittelt basierend auf einer Vorab-Annahme und in der Erhebung gesammelten Daten neues Wissen.

In den folgenden Kapiteln wird die genaue Vorgehensweise gemäß der gewählten Forschungsmethodik erläutert.

2.1 Von der Problemstellung zum Modell

Mitinitiator dieser Arbeit war die Bison AG, ein bekannter ERP-Anbieter aus der Schweiz, dessen allgemeines Interesse in der zukünftigen Entwicklung von Business Software lag. Das allgemeine Ziel bestand darin zu ermitteln, wie sich betriebliche Anwendungssysteme, im Besonderen ERP-Systeme, in den nächsten Jahren weiter entwickeln werden. Der Begriff „Business Software“ beschrieb in diesem Zusammenhang eine zu allgemeine Menge an Systemen, weshalb der Begriff ERP-System gewählt wurde. Zudem war das Interesse nicht globaler, sondern regionaler bzw. nationaler Natur. Deswegen wurde die Reichweite der Untersuchung auf Anbieter in Deutschland eingegrenzt.

Ausgehend von diesen Prämissen wurde zu Beginn die zu behandelnde Problemstellung definiert, die den Untersuchungsgegenstand genauer eingrenzte: „Künftige Anforderungen an ERP-Systeme: Deutsche ERP-Anbieter im Fokus“.

Der nächste Schritt bestand darin, ein Konzept für die Untersuchung aus der Problemstellung heraus zu entwickeln. Durch die so genannte Exploration (der zielgerichteten Suche nach der Erkenntnis eines Objekts) [Mayer, 2004, S. 30] wurde mit Hilfe von Expertengesprächen (unter anderem mit dem Marketingleiter des Initiators), ähnlichen Untersuchungen (einer in 2007 durchgeführten Studie der FHNW Schweiz und der Bison AG mit dem Thema: „Welchen Anforderungen müssen Business Software Systeme in den kommenden 3-5 Jahren Stand halten?“) und einer gründlichen Recherche der theoretischen Grundlagen in bekannten Literaturquellen, Fachzeitschriften sowie internet-basierten Quellen (aufgrund der teilweise sehr jungen Entwicklungen auf dem Gebiet der Business Software) ein theoretisches Modell für die Untersuchung entwickelt.

Das Modell wird bei dieser Vorgehensweise aus den zuvor ermittelten Theorien (Systeme logisch widerspruchsfreier Aussagen über den Untersuchungsgegenstand) abgeleitet. Es versucht, die Wirklichkeit zu beschreiben, zu verstehen, vorherzusagen und zu beeinflussen [Mayer, 2004, S. 29].

Es zeigte sich, dass sich die zu betrachtenden Eigenschaften eines ERP-Systems in Systemaspekte einordnen ließen, die sowohl konzeptionelle und technische Gesichtspunkte beinhalteten, als auch mögliche Einsatzszenarien und marktrelevante Entwicklungen

begutachteten. Dieses inhaltliche Rahmenwerk bildete somit die im Fokus stehenden Eigenschaften eines ERP-Systems vollständig ab. Während der Exploration wurden vier wichtige Aspekte eines solchen Systems identifiziert und in einem ersten Modell skizziert (vgl. Abbildung 2.2):

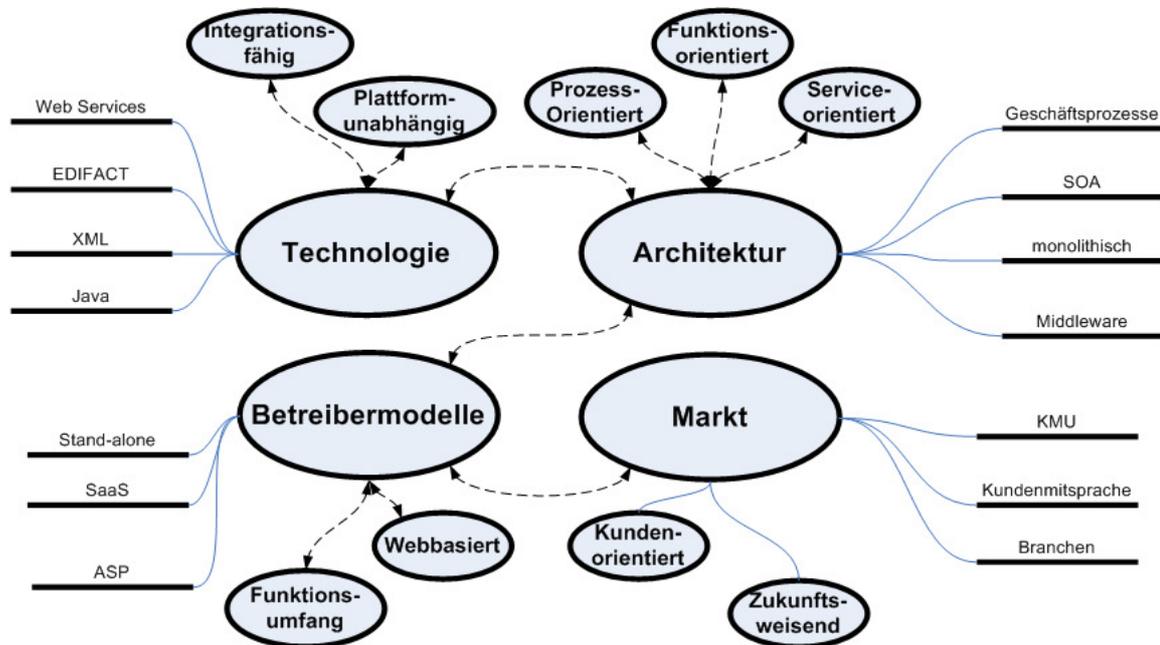


Abbildung 2.2 Vier grundlegende Aspekte eines ERP-Systems, beispielhaft skizziert.

Die Aufteilung in die vier Bereiche Architektur, Technologie, Betreibermodell und Markt erfolgte aufgrund der Notwendigkeit, alle internen und externen Einflussfaktoren, die auf eine komplette Unternehmenslösung einwirken können, abzudecken:

- **Architektur:** Die Architektur bildet das konzeptionelle und strukturelle Rahmenwerk eines ERP-Systems. Insgesamt lassen sich vier Unterbereiche identifizieren:
- Die Planungsebene ermöglicht unter anderem die Modellierung neuer Geschäftsabläufe, die Systemarchitektur bildet das System logisch auf darunter liegende Komponenten ab, die Modularchitektur beschreibt den Funktionsumfang und die Integration erlaubt die geschäftsübergreifende Anbindung.
- **Technologie:** Technologie umfasst Hardware- und Softwareeigenschaften, die sowohl für das System im internen als auch im externen Bereich als Bereiter der Infrastruktur fungieren.
- **Betreibermodell:** Hat man ein tragfähiges Systemkonzept und die für die Umsetzung notwendigen Werkzeuge, so kann das System für den Kunden verfügbar gemacht werden.
- **Markt:** Befindet sich das System einmal im Einsatz, so ist es sowohl der Nachfrage der Kunden als auch dem Konkurrenzdruck des Marktes ausgesetzt.

2.2 Vom theoretischen Modell zur Erhebungsmethode

Die nächste Aufgabe bestand darin, die relevanten Begriffe genau zu erklären und zu definieren, die zu der Beschreibung des Realitätsausschnitts und des theoretischen Modells notwendig waren [Mayer, 2004, S. 32]. Dabei galt es sowohl den Begriff „ERP“ korrekt einzuordnen als auch die damit zusammenhängenden Aspekte und deren einzelne Eigenschaften so einzuführen, dass eine Nachvollziehbarkeit der Forschung ermöglicht wird und das Verständnis zwischen Forscher und Befragten bzw. Leser über den Begriff einheitlich ist [Mayer, 2004, S. 32]. Dazu wurden Realdefinitionen, Begriffsexplikationen und Nominaldefinitionen verwendet [Mayer, 2004, S. 9]:

- Realdefinition: Es gilt, das Wesen einer Sache zu erfassen. Realdefinitionen sind entweder wahr oder falsch.
- Begriffsexplikation: Es gilt, den allgemein üblichen Gebrauch eines Ausdrucks zu ermitteln und seine verschiedenen Ausprägungen miteinander zu vergleichen.
- Nominaldefinition: Es gilt, eine Übereinkunft über die Verwendung eines Begriffes darzustellen, der für einen bestimmten Zweck innerhalb eines eingegrenzten Zusammenhangs gelten soll.

Diese Definitionen wurden anschließend noch durch eine dimensionale Analyse vertieft, d.h. Begriffe wie z.B. Systemarchitektur wurden weiter detailliert in monolithisch, Client-Server und service-orientiert. Dadurch ist es möglich zu überprüfen, ob alle Aspekte des betrachteten Begriffs adäquat berücksichtigt wurden [Mayer, 2004, S. 33].

Als Erhebungsmethoden kamen drei mögliche Vorgangsweisen in Frage:

- 1. Die Beobachtung
- 2. Die Befragung
- 3. Die Inhaltsanalyse

Da sich der Betrachtungsspielraum auf deutsche ERP-Anbieter beschränkte und deren Expertenwissen die einzige Zugangsmöglichkeit zu den benötigten Informationen darstellte, wurde die Befragung, insbesondere die persönliche Befragung im Rahmen eines Gesprächs, als Erhebungsmethode festgelegt.

2.3 Qualitative Forschung

Dabei orientierte sich die weitere Vorgehensweise an der so genannten qualitativen Forschung. Im Gegensatz zur quantitativen Forschung, bei der Hypothesen der Falsifikation ausgesetzt und so Theorien überprüft werden, ist laut Mayer die Aufgabe der qualitativen Forschung, „Theorien aus empirischen Untersuchungen heraus zu entwickeln“ [Mayer, 2004, S. 23]. Aufgrund der Tatsache, dass solche Untersuchungen im Hinblick auf die zukünftige Entwicklung von ERP-Systemen bislang sehr selten durchgeführt wurden, und aufgrund der unüberschaubaren Abhängigkeiten der vier Aspekte untereinander, war es sehr schwierig, aus dem theoretischen Modell heraus begründete und falsifizierbare Hypothesen für eine quantitative Auswertung zu gewinnen. Stattdessen wurden interpretative Verfahren zur Auswertung qualitativer Daten verwendet.

Diese wurde im Rahmen der empirischen Inhaltsanalyse nach Kromrey durchgeführt, „mit der man aus jeder Art von Bedeutungsträgern durch semantische und objektive Identifizierung

ihrer Elemente Schlüsse ziehen kann, die über das einzelne analysierte Dokument hinaus verallgemeinerbar sein sollen.“ [Kromrey, 2002, S. 311]

2.4 Das Experteninterview

Die erhobenen Daten wurden aus Experteninterviews mit Verantwortlichen bzw. kompetenten Mitarbeitern der jeweiligen ERP-Systemhäuser gewonnen. Mayer beschreibt den Experten folgendermaßen:

„Der Befragte ist hier weniger als Person (wie z.B. bei biographischen Interviews), sondern in seiner Funktion als Experte für bestimmte Handlungsfelder interessant. [...] Auch wird der Befragte nicht als Einzelfall, sondern als Repräsentant einer Gruppe in die Untersuchung einbezogen.“ [Mayer, 2004, S. 37]

Er verfügt auf einem begrenzten Gebiet über ein klares und abrufbares Wissen. Seine Ansichten gründen sich auf sichere Behauptungen und seine Urteile sind keine bloße Raterei oder unverbindliche Annahmen [Mayer, 2004, S. 40].

Alle Interviewpartner waren somit dieser Definition nach Experten, da sie entweder in ihrer Funktion als Geschäftsführer des Unternehmens, IT-Verantwortlicher oder im Vertrieb Beschäftigter ein fundiertes Wissen über das jeweilige ERP-System inne hatten.

2.5 Die Stichprobe

Nach der Modellbildung, den Definitionen und der Auswahl der Erhebungsmethode galt es, eine aussagekräftige Menge an Unternehmen zu definieren, welche in die Befragung miteinbezogen werden sollten. Verständlicher Weise konnten nicht alle Anbieter von ERP-Systemen, die in Deutschland aktiv sind, an der Befragung teilnehmen. Deshalb musste eine Stichprobe ermittelt werden, die eine solide Basis für die anschließende Auswertung bilden konnte. In der qualitativen Forschung gibt es die so genannte Vorab-Festlegung, bei der die Stichprobe absichtsvoll bzw. begründet gebildet wird. [Mayer, 2004, S. 38].

Diese Methode wurde mit der Zufallsauswahl aus der quantitativen Forschung kombiniert. So wurde zum einen darauf Wert gelegt, dass bestimmte Klassen von Anbietern (Open Source, proprietär), bestimmte Größen (z.B. Nische, branchenübergreifend) und aktuell diskutierte Unternehmen (z.B. SAP, Oracle) vertreten waren, um für die Forschung relevante und leitende Aussagen zu erhalten. Zum anderen sollte die Stichprobe auch eine statistische Repräsentativität besitzen. Insgesamt wurden 130 Anbieter von ERP-Systemen in Deutschland gebeten, sich an der Befragung zu beteiligen. Die 32 Unternehmen, die teilgenommen haben (vgl. Kapitel 2.10 und 4.4.1), verteilen sich gut auf die unterschiedlichen Unternehmensgrößen und Branchen, allerdings ist das im Hinblick auf die Repräsentativität keine optimale Zahl. Dafür entspricht die Vorab-Festlegung den Erwartungen, da alle inhaltlich interessanten Bereiche vertreten sind.

2.6 Der Fragebogen

Anlehndend an Mayer [Mayer, 2004, S. 42] wurde der Fragebogen für die Experteninterviews mit einem Leitfaden versehen, der sich aus dem ersten Modell und den darin enthaltenen vier Aspekten formte. Sie bilden gleichermaßen das theoretische und konzeptionelle Fundament für die Betrachtung des Realitätsausschnittes, so dass sie die Grundlage für das so genannte sensibilisierende Konzept darstellen, das Mayer als Grundlage für den Leitfaden in einem Experteninterview verlangt.

2.7 Gliederung des Fragebogens

Da von Beginn an der Kern der Arbeit nicht in der theoretischen Beschreibung, sondern in einer empirischen Wissensermittlung lag, wurden nach den theoretischen Bestandsaufnahmen zu jedem der vier Themenbereiche Fragen erstellt, die in darauf folgenden Interviews beantwortet werden sollten (Zur Beschreibung der Interviews und deren Ablauf vgl. Kapitel 2.6 und 2.7). Bezogen auf die jeweiligen vier Aspekte Architektur, Technologie, Betreibermodell und Markt richtete sich der Inhalt der Fragen zum Teil nach aktuellen Entwicklungen und zum Teil nach in der Literatur vorgeschlagenen idealen Szenarien.

Insbesondere sollten zum einen verschiedene Architekturansätze von Business Software vorgestellt und von den Anbietern bewertet, sowie bekannte Architekturen mit den Betrachteten verglichen werden. Zum anderen sollten verschiedene Ausprägungen der Anwendungssysteme dank eines heterogenen Abbildes der Business Software Anbieter gefunden, mögliche Zukunftsaussichten der Befragten gewonnen und in der Arbeit verwertet werden. Im Folgenden werden die einzelnen Fragen und die dahinter liegende Motivation kurz angeführt:

Architektur

Die Architektur beschreibt die konzeptionelle Ausrichtung des Systems. Dabei wirken

- Planungsebene als Geschäftsprozess-Planungswerkzeug
- Systemarchitektur als zugrunde liegendes Konzept
- Modulararchitektur als Funktionsübersicht
- und Integration als Bindeglied zu anderen Systemen

als essentielle Werkzeuge zur Umsetzung der Gesamtarchitektur. Sie beschreiben nicht nur das Grundgerüst der Software, sondern charakterisieren auch deren Interaktionsfähigkeit mit außen stehenden Systemen.

Planungsebene

Zuerst soll die Rolle von Geschäftsprozessen, insbesondere Referenzprozessen, in betrieblichen Anwendungssystemen näher beleuchtet werden, da die Sichtweise auf ein solches System (funktionsorientiert bzw. prozessorientiert, vgl. Kapitel 3.2.1.1) eine entscheidende Rolle für seinen konzeptionellen und architekturellen Aufbau darstellt. Dabei stellt eine visuelle Geschäftsprozessdarstellung einen ersten Schritt hin zu einem prozessorientierten System dar (Frage 1). Davon ausgehend ist die Vision einer automatisierten Codeerzeugung, wie sie das Business Process Management vorsieht, vgl. Kapitel 3.2.1.5, eine Weiterentwicklung (Frage 2). Im Hinblick auf zukünftiges Collaborative Business und allgemeiner Standardisierungsbemühungen ist es interessant zu beobachten, ob bereits postulierte Prozessstandards, vgl. Kapitel 3.2.1.7, eine Rolle in der Prozessdarstellung und -organisation der Anbieter spielen (Frage 4). Aus Anwendersicht wird zuletzt untersucht, ob eine Nutzung dieser Prozesse beim Kunden zur Laufzeit möglich ist (Frage 3) bzw. Änderungen durch den Kunden selbst durchgeführt werden können und mit welchem Aufwand dies geschieht (Frage 5).

- Existieren Referenzprozesse in graphischer Form in Ihrem System?
- Wenn ja, werden sie zur Codegenerierung genutzt?
- Wenn ja, werden diese Referenzprozesse beim Kunden genutzt?
- Werden entsprechende Branchenstandards verwendet? (z.B. Score, GS1, ...)?
- Können Geschäftsprozesse auch vom Benutzer angepasst werden?

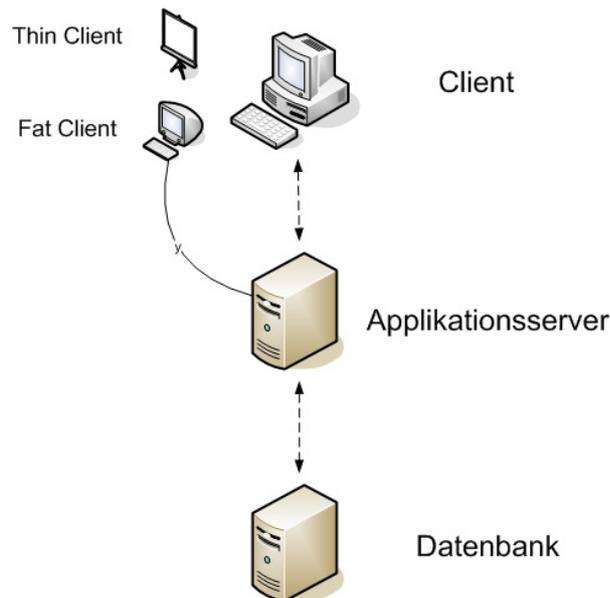


Abbildung 2.3 Mögliche architekturelle Ausprägungen

Systemarchitektur

In diesem Abschnitt sollen vor allem Konzeption und Systemarchitektur untersucht werden. Das zum Untersuchungszeitpunkt bereits länger bekannte, aber nicht immer von allen Beteiligten scharf definierte Schlagwort zum Konzept ist die service-orientierte Architektur (Frage 1, Teil 1), vgl. Kapitel 3.2.2.7. Parallel dazu existiert die Systemarchitektur (Client-Server), vgl. Kapitel 3.2.2, die sich logisch bzw. auch physisch auf mehrere Einheiten verteilen kann (Frage 1, Teil 2, vgl. Abbildung 2.3). Weitere Aspekte einer Systemarchitektur sind ihre Plattform-unabhängigkeit, vgl. Kapitel 3.2.2.8, (Frage 2) und die Middleware, auf die sie aufsetzt, vgl. Kapitel 3.2.2.5, (Frage 3). Zuletzt werden die Eigenschaften eines browserbasierten Systems untersucht (Frage 4).

- 1. Folgt die Architektur einem generellen Konzept? (z.B. SOA) und wie gestaltet sich Ihre Client-Server Architektur? (vgl. Abbildung 2.3)
- 2. Ist die Software plattformunabhängig?
- 3. Welche Middleware wird verwendet?

- 4. Wie sieht im Falle von browserbasierten Systemen die eingesetzte Technologie aus? (z.B. Java, Ajax, ...)

Modularchitektur

Dieser Abschnitt beschäftigt sich mit der Modularchitektur eines ERP-Systems. Zuerst soll ermittelt werden, welche wertschöpfenden Aktivitäten nach Porter unterstützt werden [Porter, 2000, S. 66], vgl. Kapitel 3.2.3.1, (Frage 1, vgl. Abbildung 2.4) und ob es evtl. Aktivitäten gibt, die nicht durch den Anbieter abgedeckt werden können, vgl. Abbildung 2.5, (Frage 2). Customer Relationship Management, CRM, und Business Intelligence, BI, werden als eigenständige Module gesondert betrachtet gesehen, vgl. Kapitel 3.2.3.2 und 3.2.3.3. Deren Integration in ein Anwendungssystem (Frage 3) und deren herstellerbezogener Ursprung (Frage 4) runden die Betrachtung der Modullandschaft ab. Im Hinblick auf die Anwender solcher Systeme wird in der letzten Frage die Möglichkeit von eigenen Customizings, vgl. Kapitel 3.1.6, und deren Releasefähigkeit eingegangen.

- 1. Welche Module deckt Ihre Software ab? (vgl. Abbildung 2.4)
- 2. Welche Module sind evtl. von außen zu beziehen? (vgl. Abbildung 2.5)
- 3. Sind Aspekte wie CRM oder BI integriert?
- 4. Benutzen Sie eigene Lösungen für CRM/BI oder werden diese zugekauft? (z.B. Cognos, ...)
- 5. Gibt es ein Releasekonzept, das es dem Kunden ermöglicht, Individualentwicklungen durchzuführen?

| | | | Stand-alone Module | Integration mit anderen Modulen |
|-----------------------|---|--|--------------------------|---------------------------------|
| Sekundäre Aktivitäten |  | Finanz- und Rechnungswesen | <input type="checkbox"/> | <input type="checkbox"/> |
| |  | Personalwesen | <input type="checkbox"/> | <input type="checkbox"/> |
| |  | Geschäftsführung/ Management | <input type="checkbox"/> | <input type="checkbox"/> |
| |  | Interne Services | <input type="checkbox"/> | <input type="checkbox"/> |
| Primäre Aktivitäten |  | Kundenservice | <input type="checkbox"/> | <input type="checkbox"/> |
| |  | Beschaffung/Einkauf | <input type="checkbox"/> | <input type="checkbox"/> |
| |  | Produktion/ betriebliche Leistungserstellung | <input type="checkbox"/> | <input type="checkbox"/> |
| |  | Marketing und Vertrieb | <input type="checkbox"/> | <input type="checkbox"/> |
| |  | Materialwirtschaft/ Warenwirtschaft | <input type="checkbox"/> | <input type="checkbox"/> |
| |  | Produktentwicklung | <input type="checkbox"/> | <input type="checkbox"/> |

Abbildung 2.4

Eingesetzte Module

Integration

Der letzte Abschnitt zum Thema Architektur bildet die Integration sowohl innerhalb als auch außerhalb des Systems. Business to Business Integration, B2B, vgl. Kapitel 3.2.4.4, ist auf Daten- und Applikationsebene ebenso wichtig (Frage 1) wie die internen Abhängigkeitsverhältnisse (Frage 2). Ein voll ausgebautes Berechtigungskonzept (Frage 3) kann im Hinblick auf Integration sehr hilfreich sein.

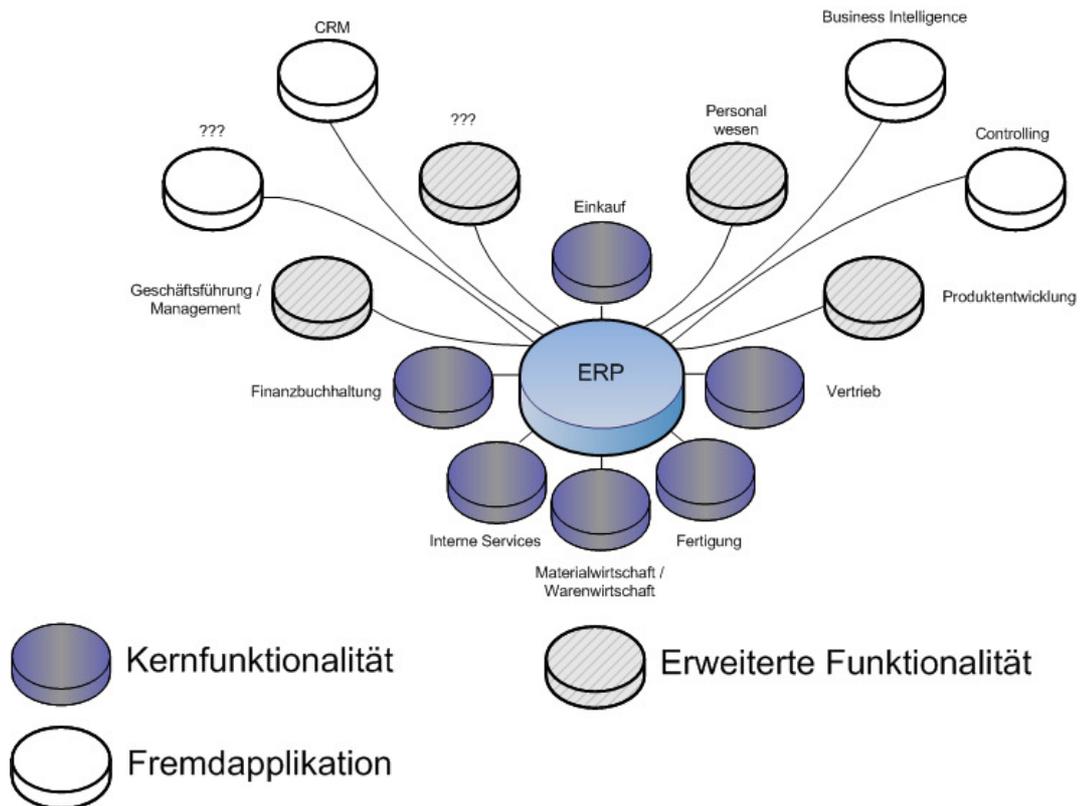


Abbildung 2.5 Beispiel: Kernfunktionalitäten in Ihrem ERP

- 1. Ist eine Anbindung an ein Fremdsystem bzw. in eine bestehende IT-Landschaft möglich, wenn ja, wie (Datenebene/ Applikationsebene)?
- 2. Gibt es Module, die sehr stark voneinander abhängen? (vgl. Abbildung 2.4)
- 3. Wird auf ein Berechtigungskonzept (sowohl intern als auch extern) Wert gelegt und wenn ja, wie? (z.B. Rollen)

Technologie

Die Technologie fungiert als Werkzeugbasis, mit deren Hilfe das System operiert, intern wie auch extern. Hier soll ein Eindruck über die technologische Innovationsfähigkeit gewonnen werden, wie auch über die Fähigkeit, sich in bereits existierende Systemlandschaften einfügen zu können.

Systemintern

Die Basis und ein Indikator für die technische Eigenschaften der betrachteten Systeme sind zum einen die verwendeten Betriebssysteme und Datenbanken, vgl. Kapitel 3.3.1.1 und 3.3.1.2 (Frage 2), zum anderen die Programmiersprache, in der sie entwickelt worden sind (Frage 2), und die damit verbundene Entwicklungsumgebung, vgl. Kapitel 3.3.1.3 und 3.3.1.4, (Frage 3). Dabei verwendete Open Source Programme, z.B. in der Entwicklung, und deren allgemeine Popularität werden ebenfalls ermittelt (Frage 4).

- 1. Welche Betriebssysteme und Datenbanken setzt Ihr System voraus?
- 2. Welche Programmiersprache wurde benutzt und warum? (C/C++, Java, ...)
- 3. Welche Entwicklungsumgebung benutzen Sie?
- 4. Was halten Sie von Open Source als Werkzeug in Ihrem System?

Systemextern

Dokumentenstandards im Rahmen von Electronic Data Interchange, EDI - vgl. Kapitel 3.3.2.1, wurden und werden immer noch als sehr wichtig angesehen. Ihre Unterstützung (Frage 1) und der gleichzeitige Einsatz von Web Services, vgl. Kapitel 3.3.2.4, als mögliches Integrationswerkzeug (Frage 2) werden ebenso untersucht wie die in den immer öfter genutzten Web-Anwendungen gebrauchten Übertragungsmethoden (Frage 3).

- 1. Welche Dokumentenstandards unterstützt Ihr System im Hinblick auf externe Kommunikation mit anderen Systemen, (z.B. EDIFACT, ...) und nutzen Sie in diesem Rahmen Konverter?
- 2. Werden Web Services eingesetzt und wenn ja, wie?
- 3. Setzen Sie in Bezug auf Web-Anwendungen eher auf die bereits existierenden Ressourcen (native Web) oder auf proprietäre Lösungen für einen verbesserten Durchsatz? (Bsp. Citrix)

Betreibermodell

Das Betreibermodell gibt Auskunft über die möglichen Einsatzvarianten beim Kunden und lässt gleichzeitig einen Schluss über die Vertriebsstruktur des Systems zu. Die von den Anbietern technisch möglichen (Frage 1) und tatsächlich beim Kunden eingesetzten Betreibermodelle, vgl. Kapitel 3.4, (Frage 5) bilden den Kern dieses Frageblocks (vgl. 2.6). Im Hinblick auf SaaS wird außerdem die Mandantenfähigkeit untersucht. Zuletzt ist eine Untersuchung der Vertriebs- und Wartungskanäle im Rahmen des beim Kunden verwendeten Betreibermodells interessant als Hinweis auf die Verlagerung der vertrieblichen und Support-Kompetenzen (Frage 3 und 4).

- 1. Für welche Betreibermodelle ist Ihre Software konzipiert worden? (ASP, SaaS, stand-alone, ...)
- 2. Ist Ihre Software mandantenfähig?
- 3. Wer vertreibt die Software?

- 4. Wer wartet die Software? (Support)
- 5. In welcher Art und Weise werden die Funktionsbereiche Ihres Systems dem Kunden verfügbar gemacht? (vgl. Abbildung 2.6)

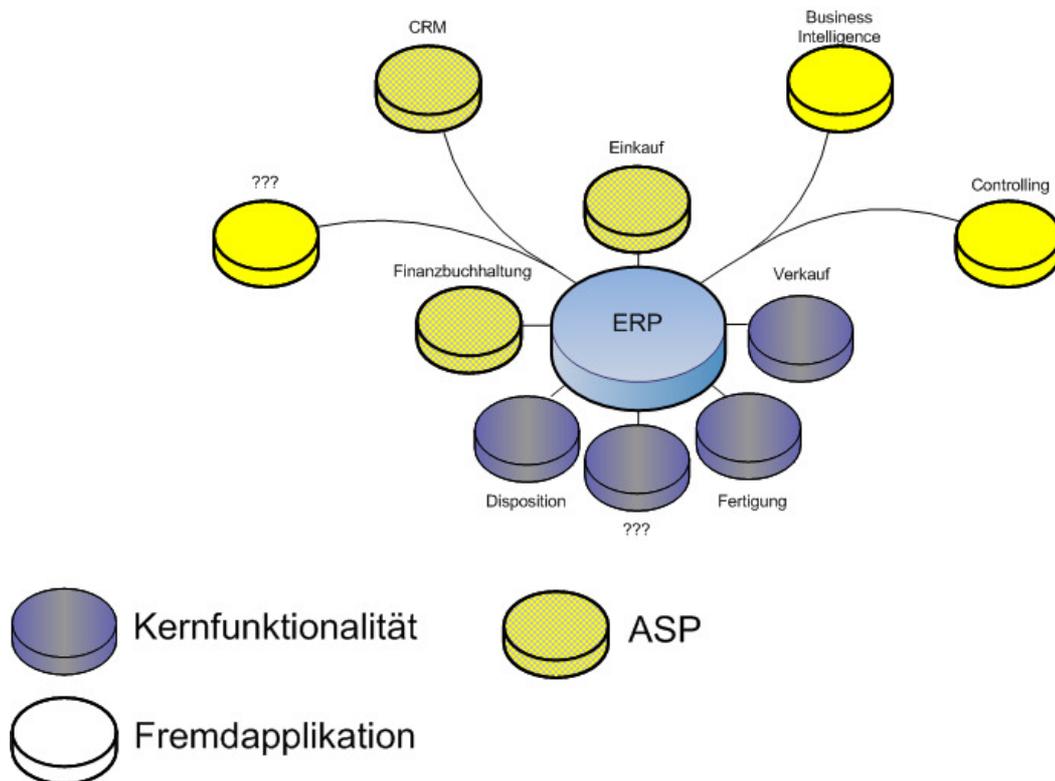


Abbildung 2.6 Beispiel eines Betreibermodells Ihres ERP

Markt

Der Markt ist das Feld, auf dem sich die Software im Wettbewerb um den Kunden gegen Ihre Konkurrenz durchsetzen muss. In diesem Abschnitt sollen die von den Anbietern erwarteten Entwicklungen dargelegt werden, Eigenschaften, mit denen sie sich von Ihren Konkurrenten abgrenzen, und deren Kundennähe.

Anwender

Es soll nicht nur das System selbst betrachtet werden, sondern auch Eindrücke der Anwenderseite gewonnen werden. Als Klassifikationsmerkmale dienen zum einen die Kundenanzahl (Frage 1) der Systeme und zum anderen die mögliche Userzahl (Frage 2). Auf Unterscheidungskriterien zur Konkurrenz für potentielle Kunden zielte Frage 3 ab. Zuletzt waren der Einfluss der Kunden und deren Kommunikationswege als Zugangsmöglichkeit interessant (Frage 4 und 5).

- 1. Wie viele Kunden setzen Ihre Software ein?
- 2. Wie viele User können Ihr System nutzen?

- 3. Was grenzt Sie zu Ihrer Konkurrenz ab?
- 4. Wie groß ist der Einfluss der Kunden auf das fertige Produkt?
- 5. Wie organisieren sie die Kommunikation mit dem Kunden? (z.B. Usergroups, ...)

Anbieter

Auf Anbieterseite findet zuerst eine weitere Klassifikation der Systeme nach Branche statt, vgl. Kapitel 3.5.2, (Frage 1). Anschließend wird die eigene Einschätzung der Anbieter im Vergleich zu ihrer Konkurrenz abgefragt (Frage 2). Die dritte Frage soll eine Einschätzung der Open Source Konkurrenz bzw. der proprietären Anbieter aus Open Source-Sicht ermöglichen, vgl. Kapitel 3.5.3. Frage 4 zielt auf die eigene Einschätzung im Hinblick auf eine zukünftige Marktentwicklung, während die letzte Frage auf mögliche Zukunftsszenarien des Marktes allgemein abzielt.

- 1. Für welche Branchen ist Ihr Produkt besonders geeignet? (Handel, Produktion, Dienstleistung)
- 2. Wie sehen Sie sich selbst am Markt der Software-Anbieter positioniert?
- 3. Sehen Sie Open Source / proprietäre Software-Anbieter als ernsthafte Konkurrenz und wenn nicht, warum?
- 4. Wie bzw. wo im Markt sehen Sie sich in zehn Jahren?
- 5. Wie sehen Sie die Zukunft des Marktes?

Da zu Beginn der Arbeit noch eine zusätzliche Anwenderbefragung angedacht war, existierte im Bereich Markt/Anbieter noch eine zusätzliche Frage: "Gibt es eine Liste von Referenzkunden?"

Damit sollten mögliche Interviewpartner auf Anwenderseite ermittelt werden. Allerdings entwickelte die FHNW und die Bison AG eine Studie parallel zu dieser Arbeit, die Anforderungen an ERP-Systeme aus Anwendersicht begutachtet [Felley, 2007]. Daher wurde diese Frage aus dem Fragebogen gestrichen.

Durchführung der Befragung

Noch vor den eigentlichen Befragungen wurden drei Pretests durchgeführt, die problematische, zu komplexe oder unverständliche Formulierungen aufzeigen [Mayer, 2004, S. 44] und gleichermaßen als Test für die Länge der Interviews dienen sollten. An diesen telefonischen Tests nahmen SoftM, Abacus und Bison teil. Dadurch konnte der Fragebogen verbessert und eine Interviewzeit von 30 Minuten ermittelt werden.

In der nächsten Phase wurde eine Stichprobe von 130 Unternehmen, die eine ERP-Lösung anbieten, zu einem Interview eingeladen. Es wurde darauf geachtet, dass Unternehmen jeder Größe und Branchenfokussierung in der Stichprobe vertreten waren. 32 Unternehmen partizipierten an der Befragung, was eine Rücklaufquote von 24,6 Prozent ergibt. Festzuhalten bleibt, dass die hier gewonnenen Erkenntnisse kein vollständiges Bild für die Gesamtheit aller ERP-Anbieter im deutschsprachigen Raum darstellen. Allerdings sind Unternehmen jeder Größe und mit Lösungen für eine breite Palette von Branchen interviewt

worden, so dass zumindest eine inhaltlich qualitativ korrekte Einschätzung getroffen werden kann.

Die Interviews wurden zum größten Teil telefonisch im Rahmen von etwa 30 Minuten geführt, ein Interview wurde mit der Software Skype aufgezeichnet. Für jede Befragung wurde eine handschriftliche Transkription angefertigt. Die ersten drei Interviews wurden im Zeitrahmen von Juni bis Juli geführt, um den Fragebogen hinsichtlich seiner Qualität zu testen. Anschließend wurden die Fragen weiter optimiert und ergänzt.

Ab Oktober bis in den Dezember hinein erfolgten dann weitere 17 Interviews. Ein Interview fand persönlich vor Ort beim Hersteller statt, zwei wurden schriftlich eingereicht. Jedem Teilnehmer wurden Reviews zur nachträglichen Begutachtung via E-Mail geschickt und die Antworten bzw. Verbesserungen nachgetragen.

Nach der ersten Auswertung wurden noch einmal leichte Änderungen im Fragebogen vorgenommen. Die Abbildung 2.5 wurde noch einmal verbessert und eine Frage aus dem Bereich Architektur/Integration zum Thema „Standards“ mit der Frage aus dem Bereich Technologie/Systemextern zum Thema „Dokumentenstandards“ verschmolzen. Hinzu kam im Bereich Markt/Anwender die Frage „Wie viele User können Ihr System nutzen?“, um eine weitere Klassifikationsmöglichkeit der Unternehmen zu erhalten. Anschließend wurden weitere 12 Interviews geführt und nach dem oben beschriebenen Vorgehen ausgewertet, davon wurden 8 Fragebögen schriftlich eingereicht.

2.8 Die Auswertung

Laut Mayer ist das Ziel der Auswertung bei Experteninterviews, das Überindividuell-Gemeinsame herauszuarbeiten [Mayer, 2004, S. 46]. Mit dieser Prämisse als Grundlage wurden die schriftlichen Transkriptionen nach einem Verfahren von Mühlfeld u.a. ausgewertet:

Demnach ist es nicht wichtig, „ein einzelnes Interview so exakt wie möglich zu interpretieren.“ „Die Konstruktion von Mustern aus Gemeinsamkeiten, Unterschieden, tendenziellen Analogien erfordert weitere theoretische und textgebundene Arbeitsschritte, um zu einer theoretisch wie empirisch abgesicherten Darstellung und Interpretation der Ergebnisse zu gelangen.“ [Mühlfeld, 1981, S. 334]

Die Vorgehensweise ist wie folgt:

- 1. Antworten markieren: Alle Textstellen, die ersichtliche Antworten auf entsprechende Fragen darstellen, werden markiert.
- 2. In Kategorienschema einordnen: Beim zweiten Durchlesen wird der Text in ein zuvor erstelltes Kategorienschema eingeordnet.
- 3. Innere Logik herstellen: Einzelinformationen innerhalb eines Interviews werden logisch miteinander verknüpft, wobei gleiche aber auch sich widersprechende Informationen berücksichtigt werden müssen.
- 4. Text zur inneren Logik erstellen: Anschließend wird diese innere Logik niedergeschrieben.
- 5. Text mit Interviewausschnitten: Nun wird die Auswertung mit Interviewausschnitten angereichert erstellt.
- 6. Bericht: Zuletzt wird aus dem Auswertungstext eine Präsentation entwickelt.

2.9 Prognose und kritische Bewertung

Ausgehend von den aus den Interviews gewonnenen Erkenntnissen wurde so unter Berücksichtigung von aktuellen und während der Arbeit neu auftauchenden Entwicklungen eine Prognose erstellt, die zukünftige Eigenschaften von ERP-Systemen beschreiben soll (vgl. Kapitel 5). Anschließend erfolgte eine kritische Bewertung der Inhalte und der Vorgehensweise bzw. der weiteren Forschungsmöglichkeiten in diesem Themengebiet (vgl. Kapitel 6).

2.10 Befragte Unternehmen

Auf den folgenden Seiten werden die einzelnen Unternehmen hinsichtlich bestimmter Eigenschaften tabellarisch aufgelistet. Kurzbeschreibungen der Anbieter und Informationen zur

Befragung finden sich im Anhang (vgl. Kapitel A.4). Das Ziel der Einteilung ist es, dem Leser eine Einordnung der Unternehmen hinsichtlich

- ihrer Branchenausrichtung
- der Kundenanzahl
- der üblichen Userzahlen für ihr System
- der Anzahl der Mitarbeiter und
- der regionalen Verbreitung

zu geben. So kann man sich einen groben Überblick über die betrachteten Anbieter machen und deren Unterschiede untereinander auf einen Blick erfassen. Es ist wichtig anzumerken, dass die meisten Softwarehäuser, die an der Befragung teilgenommen haben, Standardsoftware-Produkte vertreiben. Das heißt, eine bereits vorgefertigte Software wird für den Kunden installiert und seinen Wünschen entsprechend angepasst. Es gibt aber auch Anbieter, die im Gegensatz zu einem bereits vorgefertigten Anwendungssystem eine individuelle Betreuung bzw. sogar Programmierung vornehmen. Unter den befragten Unternehmen gibt es drei Anbieter, die dieser Kategorie entsprechen: Ramco, Seat-1 und Günther-BS.

| ERP- Anbieter | Produkt | Branchenausrichtung | Kundenanzahl | Übliche Userzahlen | Anzahl der Mitarbeiter | Region | Webseite |
|--------------------------------|-------------------|--|-------------------------------------|--------------------|------------------------|---|--|
| Abacus Research AG | Abacus Enterprise | branchenübergreifend | 30.000 | bis 500 | etwa 150 Mitarbeiter | Schweiz, Deutschland, Österreich | http://www.abacus.ch |
| Alea GmbH | Amc ² | Versandhandel | 19 | bis 2500 | über 40 Mitarbeiter | Deutschland, Schweiz, Österreich | http://www.alea.de |
| ams.hinrichs+müller GmbH | ams.erp | Einzel-, Auftrags- und Variantenfertigung | 450 | bis 400 | etwa 60 Mitarbeiter | Deutschland, Schweiz, Österreich | http://www.ams-erp.com |
| B.I.M.-Consulting mbH | PDV | Unikatfertiger im Maschinen-, Apparate- und Anlagenbau | 32 | ab 10 bis über 500 | etwa 30 Mitarbeiter | Deutschland, Schweiz, Österreich | www.bim-consulting.de |
| e.bootis AG | e.bootis-ERP/II | Handel und Diskrete Fertigung | 500 (alte Lösung), 70 (neue Lösung) | ab 30 bis über 500 | etwa 120 Mitarbeiter | Deutschland, Afrika | http://www.ebootis.de |
| ERP-21 GmbH | Use-21 | Dienstleister und Produktion | 7 | ab 10 bis 50 | etwa 10 Mitarbeiter | Deutschland | http://www.erp-21.de |
| ERP4all Business Software GmbH | Michas.ERP/PPS | Handel, Produktion und Dienstleistung | 300 | ab 5 bis 25 | etwa 35 Mitarbeiter | Deutschland, Österreich, Tschechische Republik, Thailand, China | http://www.erp4all.com |
| Godyo AG | Godyo P/4 | Maschinenbau / Anlagenbau, Auftrags- und Kleinserienfertiger, diskrete Fertigung | 50 | ab 5 bis 200 | etwa 50 Mitarbeiter | Deutschland | http://www.godyo.com |

Abbildung 2.7

Übersicht der interviewten Unternehmen, Teil I

| ERP-Anbieter | Produkt | Branchenausrichtung | Kundenanzahl | Übliche Lisenzahlen | Anzahl der Mitarbeiter | Region | Webseite |
|----------------------------------|------------------------|--|---|------------------------------|---------------------------|--|---|
| Günther Datenverarbeitung GmbH | Praktikus Professional | Handel (Groß-/Versandhandel), Produktion (Einzel/Kleinserienfertiger), Dienstleistung (u.a. Werbebranche, Eventmanagement) | 400 | bis 250 | etwa 15 Mitarbeiter | Deutschland | http://www.guenther-dv.de |
| GUS Group AG & Co KG | GUS-OS | Pharma, Chemie, Food, Logistik, Kosmetik | 100 | ab 5 bis 1.000 | etwa 100 Mitarbeiter | Deutschland, Schweiz | http://www.gus-group.com |
| Hilmer Software GmbH | FERIX | Maschinenbau / Anlagenbau | 70 | ab 5 bis über 10 (bis 99) | etwa 7 Mitarbeiter | Deutschland, Schweiz, Österreich | http://www.hilmer-software.de |
| IAS GmbH | CANIAS ERP | Handel und Produktion, aber sehr produktionslastig. Projekte in Lebensmittelindustrie und Kunststoff. | 150 | bis 300 | etwa 80 Mitarbeiter | Deutschland, Türkei, Vereinigte Arabische Emirate, China | http://www.canias.de |
| IFS DEUTSCHLAND GMBH & CO. KG | IFS Applications | Produktion und Dienstleistung, zudem projektorientierte Unternehmen | 2.200 | ab 50 bis über 500 | etwa 140 Mitarbeiter | Weltweit | http://www.ifs-world.com |
| Informing AG | IN:ERP | Prozessindustrie, im speziellen Nahrungsmittel. Aber auch Großhandel, Produktion. Wenig Dienstleistung. | 275 (alte Lösung), 86 (neue Lösung) | ab 10 bis 500 | etwa 70 Mitarbeiter | Schweiz, Deutschland, Österreich | http://www.informing.ch |
| Intraprend | cierp3 | Handel, Dienstleistung, e-Commerce, Produktion (Industrie) | 350 | ab 1 bis über 500 | etwa 12 Mitarbeiter | Weltweit | http://www.intraprend.com |
| JENTECH Datensysteme AG | Zephyr Avenue | Produktionsbetriebe, speziell Lohnfertiger | 15 | ab 1 bis 50 | etwa 50 Mitarbeiter | Deutschland | http://www.jentech.de |

Abbildung 2.8

Übersicht der interviewten Unternehmen, Teil II

| ERP-Anbieter | Produkt | Branchenausrichtung | Kundenanzahl | Übliche Userzahlen | Anzahl der Mitarbeiter | Region | Webseite |
|----------------------------------|----------------------------------|--|---|-----------------------------|-------------------------|--|---|
| Lawson Software Deutschland GmbH | M3/S3 | In Europa: Produktions-Logistik Unternehmen (Mode, Food&Beverage, serviceorientierte, Retail-Wholesale, Anlagenmanagement) | 4.000 | ab 20 bis mehrere 1.000 | etwa 3.600 Mitarbeiter | Weltweit | http://www.lawson.com |
| Microsoft Deutschland GmbH | MS Dynamics Navision | branchenübergreifend | 16.500 in Deutschland, über 60.000 weltweit | ab 10 bis 200 | etwa 79.000 | Weltweit | http://www.microsoft.com/germany/dynamics |
| myfactory Software GmbH | myfactory.ERP | Handel, Produktion und Dienstleistung | 500 | ab 10 bis über 20 (bis 100) | etwa 25 Mitarbeiter | Deutschland, Österreich, Schweiz, Benelux, Italien, Griechenland | http://www.myfactory.com |
| nGroup (Compra) | eVolution | branchenübergreifend | 1.000 | ab 1 bis 300 | etwa 35 Mitarbeiter | Deutschland, Österreich, Schweiz, Frankreich | http://www.ngroup.info |
| Oracle Deutschland GmbH | E-Business Suite | Telco, Retail, Banking, Logistic Service Provider, Process Manufacturing, Utilities | 30.000 | ab 1 bis unbegrenzt | etwa 68.000 Mitarbeiter | Weltweit | http://www.oracle.com/global/de/index.html |
| PSI AG | PSI/penta | Produktion (Automobile und Anlagenbau) | 500 | ab 30 bis unbegrenzt | etwa 1.000 Mitarbeiter | Europa und China | http://www.psi.de |
| Ramco | Ramco ERP | branchenübergreifend | 1.000 | ab 1 bis unbegrenzt | etwa 1.800 Mitarbeiter | Weltweit | http://www.ramco.ch |
| Sage Baurer | baurer (trade/industry/wincarat) | Maschinen/Anlagenbau, Automobile, Kunststoff, techn. Handel. | 1.200 | ab 150 bis über 1.000 | etwa 230 Mitarbeiter | Deutschland, Schweiz, Österreich | http://www.sage.de/baeurer/ |

Abbildung 2.9

Übersicht der interviewten Unternehmen, Teil III

| ERP-Anbieter | Produkt | Branchenausrichtung | Kundenanzahl | Übliche Userzahlen | Anzahl der Mitarbeiter | Region | Webseite |
|--------------------------------|--------------------|--|----------------------------------|----------------------|-------------------------|---|---|
| SAP Deutschland AG & Co. KG | SAP Business Suite | Etwas 20 Branchen (Umfang selbst definiert) bzw. Spezialbranchenlösungen durch Partner (Microverticals). | 44.000 | ab 1 bis unbegrenzt | etwa 44.000 Mitarbeiter | Weltweit | http://www.sap.com |
| Seat-1 Software GmbH | IntarS | Produktion (Anlagenbau) / Handel | 30 | ab 1 bis über 50 | etwa 12 | Deutschland, Österreich | http://www.seat-1.com |
| SHD Holding GmbH | Sangross/Ecoro | Einzelhandel (Möbelhandel) und technischer Großhandel | Einzelhandel 350, Großhandel 140 | ab 1 bis 6.000 | etwa 450 | Deutschland, Österreich, Schweiz, Benelux | http://www.shd.de |
| Sofem Software und Beratung AG | Semiramis | Großhandel, Dienstleistung, Fertigung | 200 | keine Angabe | etwa 470 | Deutschland, Österreich, Schweiz, Frankreich, Polen, Tschechien | http://www.sofem.com |
| SoftM & Bison Schweiz AG | Greenax | Handel | 15 | ab 20 bis über 1.000 | etwa 340 | Deutschland, Schweiz | http://www.bison-group.com |
| SynERP Y GmbH | AvERP | Generell Produktion und Dienstleistung | 200 | keine Angabe | etwa 20 Mitarbeiter | Europaweit | http://www.synerpy.de/ |
| Topix Informationssysteme AG | Topix:5 | Handel und Dienstleistung | 2450 | ab 1 bis unbegrenzt | etwa 30 Mitarbeiter | Deutschland, Österreich | http://www.topix.de |
| Wilka Systems | MS Dynamics Axapta | branchenübergreifend | 9000 | ab 1 bis unbegrenzt | etwa 20 Mitarbeiter | Weltweit | http://www.wilka-systems.de |

Abbildung 2.10

Übersicht der interviewten Unternehmen, Teil IV

3 Theoretische Grundlagen

In den folgenden Abschnitten werden grundlegende Begriffe erläutert, die für die Betrachtung des Themas relevant sind. Dabei werden zuerst allgemeine Definitionen eingeführt. Anschließend werden zu den vier Teilaspekten Architektur, Technologie, Betreibermodell und Markt Begriffe erklärt, die wesentlich zum Verständnis der Arbeit beitragen werden und die Grundlage für die Befragung bilden.

3.1 Grundbegriffe

Allgemeine Begriffe wie Betriebliche Anwendungssysteme, ERP oder Business Software, die in allen vier Bereichen Architektur, Technologie, Betreibermodell und Markt thematisch immer wieder auftauchen und als Grundwissen vorausgesetzt sind, werden in diesem Abschnitt erläutert.

3.1.1 Betriebliche Anwendungssysteme

Die im Fokus der Arbeit stehenden Systeme sind im weiteren Sinne Betriebliche Anwendungssysteme. Die folgenden Unterkapitel beleuchten den allgemeinen Begriff und erläutern den prinzipiellen Aufbau solcher Systeme.

3.1.1.1 Definitionen

Ein Betriebliches Anwendungssystem ist der automatisierte Teil eines Betrieblichen Informationssystems [Ferstl & Sinz, 2006, S. 2]. Das Informationssystem an sich dient zur Speicherung, Wiedergewinnung und Verknüpfung von Informationen, die Leistungsprozesse und Austauschbeziehungen sowohl innerhalb des Betriebs als auch mit seiner Umwelt unterstützen sollen. Sie bestehen aus der technischen (Hardware, Software) und der menschlichen Komponente, die als sozio-technisches Informationssystem miteinander interagieren.

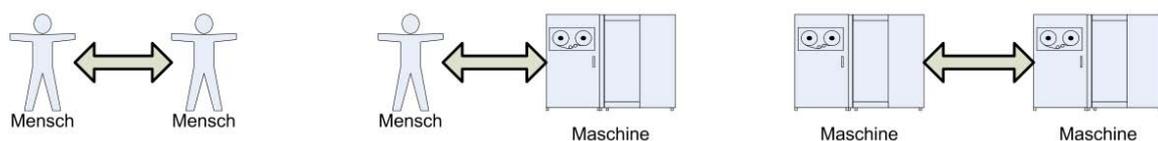


Abbildung 3.1 Kommunikationskanäle

Mensch und/oder Maschine erzeugen/nutzen Informationen, die sie über verschiedene Kommunikationskanäle austauschen (vgl. Abbildung 3.1). Nach Stahlknecht und Hasenkamp bildet ein betriebliches Anwendungssystem die Gesamtheit aller Programme und der zugehörigen Daten für ein konkretes betriebliches Anwendungsgebiet [Stahlknecht & Hasenkamp, 2005, S. 326]. Im weiteren Sinne werden dabei Hardware, Systemsoftware, Kommunikationskanäle und eventuell auch Menschen in die Definition miteinbezogen.

Amberg [Amberg, 1999, S. 11f] sieht den Nutzen folgendermaßen:

„Der Einsatz von Anwendungssystemen ist für alle (teil)automatisierbaren betrieblichen Aufgaben sinnvoll, bei denen der wirtschaftliche Nutzen bei der automatisierten Durchführung den Gestaltungs- und Entwicklungsaufwand rechtfertigt.“

3.1.1.2 Klassifikationen

Eine Klassifikationsmöglichkeit von solchen Anwendungssystemen liefern Stahlknecht und Hasenkamp mit einer Einteilung nach dem jeweiligen Verwendungszweck eines Systems [Stahlknecht & Hasenkamp, 2005, S. 327] (vgl. Abbildung 3.2):

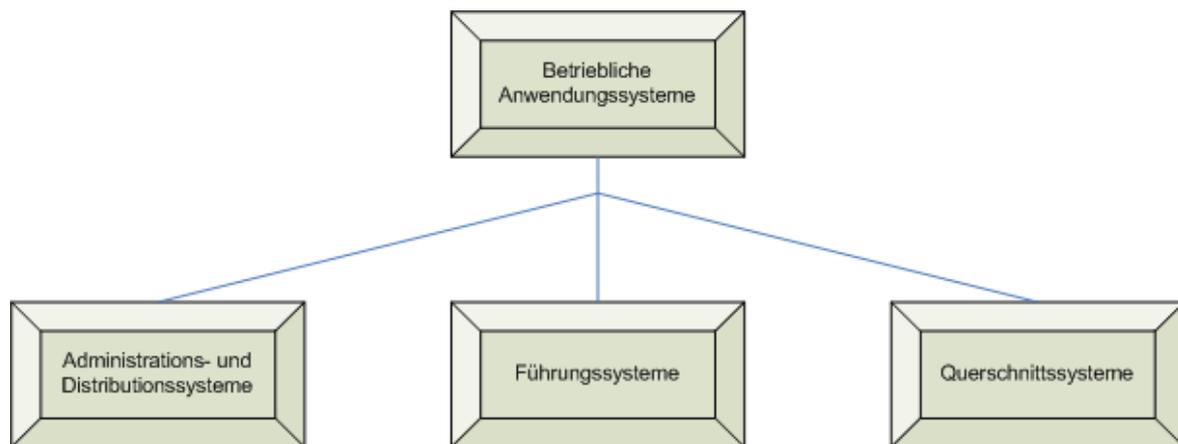


Abbildung 3.2 Klassifikation nach Verwendungszweck

Administrationssysteme bilden den Einsatz von Elementarfaktoren (Potenzial- und Verbrauchsfaktoren) im Leistungsprozess ab, wobei sie ihn zusätzlich dokumentieren und bewerten [Gluchowski et al., 2008, S. 6]. Sie unterstützen die Verwaltung von Bestandsdaten wie z.B. Kontendaten oder Lagerdaten, ihre Funktionen nutzen nur die jeweiligen Fachabteilungen.

Distributionssysteme unterstützen das operative Management [Gluchowski et al., 2008, S. 7]. Sie ermöglichen anhand von vorliegenden Bestellungen oder Wareneingängen ein gewisses Maß an Planungsmöglichkeit innerhalb einer Fachabteilung.

Führungssysteme hingegen unterstützen die Unternehmensleitung hinsichtlich der strategischen Planung und Optimierungsmöglichkeiten mit Daten, die sie sowohl intern durch Administrations- und Distributionssysteme erhalten als auch extern, z.B. durch Marktforschungsergebnisse. Querschnittssysteme schließlich unterstützen über alle betrieblichen Ebenen hinweg die operativen Tätigkeiten (Bsp.: Bürosysteme).

3.1.1.3 Komponentenbasierte Anwendungssysteme

Anwendungssysteme aus wiederverwendbaren Softwarebausteinen zu erstellen, ist seit vielen Jahren ein angestrebtes Ziel [Zaha & Kelch, 2004, S. 151]. Komponenten kapseln die verschiedenen Funktionalitäten eines Anwendungssystems und stellen sie über Schnittstellen nach außen zur Verfügung. Turowski definiert eine Komponente folgendermaßen [Turowski, 2003]:

Eine Komponente besteht aus verschiedenartigen (Software-)Artefakten. Sie ist wiederverwendbar, abgeschlossen und vermarktbar, stellt Dienste über wohldefinierte Schnittstellen zur Verfügung, verbirgt ihre Realisierung und kann in Kombination mit anderen Komponenten eingesetzt werden, die zur Zeit der Entwicklung nicht unbedingt vorhersehbar ist.

Die Vorteile einer solchen komponentenorientierten Sicht sind zum einen eine Verringerung des Aufwands, ein Anwendungssystem zusammenzustellen, verbunden mit gleichzeitiger Qualitätssteigerung, zum anderen werden Wartbarkeit und Erweiterbarkeit deutlich erhöht. Bekannte Komponentenmodelle sind in diesem Zusammenhang z.B. CORBA, EJB oder Web Services (vgl. Kapitel 3.2.2.6, 3.2.2.6, 3.3.2.4) [Acker et al., 2004, S. 9], für die mächtige Entwicklungs- und Ausführungsumgebungen existieren.

3.1.2 Enterprise Resource Planning - ERP

Enterprise Resource Planning Systeme (ERP) sind eine Teilmenge von betrieblichen Anwendungssystemen. Hervorgegangen aus den MRP (Material Requirements Planning) und MRP II Systemen, die eine reine Ressourcenverwaltung unterstützen sollten, bieten ERP Systeme ein größeres Potenzial, das weit über das seiner Vorgänger hinaus geht. Eine Definition liefern Mertens et al. [Mertens, 2001, S. 183]:

„Enterprise Resource Planning System (ERP-System) bezeichnet ein aus mehreren Komponenten bestehendes integriertes Anwendungssystem, das alle wesentlichen betrieblichen Funktionsbereiche abbildet (Beschaffung, Materialwirtschaft, Produktionsplanung- und steuerung, Finanzen, Rechnungswesen, Vertrieb, Controlling, etc.)“.

Nach Stahlknecht und Hasenkamp vereint ein ERP-System die aus der Klassifikation von Anwendungssystemen (vgl. Kapitel 3.1.1.2) genannten Administrations- und Distributionsfunktionalitäten, Funktionalitäten aus dem Führungsbereich und auch Aufgaben von Querschnittssystemen [Stahlknecht & Hasenkamp, 2005, S. 327]. Es hat also den Anspruch einer kompletten Unternehmenslösung.

Gronau beschreibt die generelle Gestalt eines ERP-Systems als einen mehrstufigen Aufbau [Gronau, 2004, S. 9f]. Zugrunde liegt ein Datenbankmanagementsystem, das Schnittstellen sowohl zu der darüber liegenden Applikationsschicht, als auch zu anderen DBMS vorhält. In der Applikationsschicht ist der Applikationskern untergebracht, eine Programmierumgebung für Ergänzungen oder Neuerungen im Kern und eine durch die Middleware organisierte RPC-Schnittstelle (Remote Procedure Call) bzw. die Möglichkeit zur Integration von Fremdapplikationen. Die Adaptionsschicht enthält zum einen die Customizing-Funktionen, d.h. die Anpassung des Systems an kundenspezifische Wünsche und ein Workflow-Management-System, das unter anderem Weiterleitungs- und Benachrichtigungsmechanismen umsetzt. Die oberste Schicht bildet die Client-Ebene, die dem Nutzer eine Benutzeroberfläche zur Verfügung stellt. Der wesentliche Vorteil der ERP-Systeme wird in ihrer Fähigkeit zur Standardisierung von Prozessen und der Automatisierung von betrieblichen Abläufen gesehen [Gronau, 2004, S. 4]. So kann die Produktivität erhöht, die Koordination erleichtert, Führungskräfte entlastet und die Stabilität des organisatorischen Systems gestärkt werden. Hitt et al. stellen in ihrer Studie einen echten messbaren Mehrwert für Unternehmen fest, der mit der Einführung eines ERP-Systems einher geht [Hitt et al., 2002, S. 93].

Allerdings gibt es auch mögliche Nachteile, wie z.B. die verminderte Anpassungsfähigkeit an veränderte Bedingungen. Mitarbeitern werden Potenziale zur selbstständigen Entscheidung und zum Handeln genommen [Gronau, 2004, S. 5]. Unternehmen sind unter Umständen gezwungen, ihre Prozesse an das ERP-System anzupassen [Vilpola et al., 2007, S. 1].

Gerade kleine und mittelgroße Unternehmen können es sich aufgrund der begrenzten Mittel nicht leisten, eine ERP-Lösung ihren Bedürfnissen entsprechend individuell anpassen zu lassen. Dieser Konflikt zwischen Standardisierung und Flexibilität ist ein wichtiger Aspekt, der auch in der Befragung ausführlich diskutiert wurde.

3.1.3 Business Software

Business Software wird als Überbegriff für alle Arten betriebswirtschaftlicher Software gesehen. Abbildung 3.3 zeigt die verschiedenen Systeme im Kontext zu seinen Beteiligten, den Managementkonzepten und der E-Business Sichtweise [Wölfle & Schubert, 2006].

Business Software besteht also nicht nur aus unternehmensintern angewandten Systemen wie ERP, Content Management Systemen oder PPS (Production Planning Systemen), sondern beinhaltet auch die elektronische Unterstützung von externen Beziehungen sowohl zur Kunden- wie auch zur Zuliefererseite.

Im Kern steht das Unternehmen, dessen Funktionen über die einzelnen Abteilungen hinweg miteinander integriert sind. Zudem werden die relevanten Geschäftsprozesse, die auch andere Partner miteinbeziehen, durch eine hinreichende Integration mit externen Systemen abgebildet.

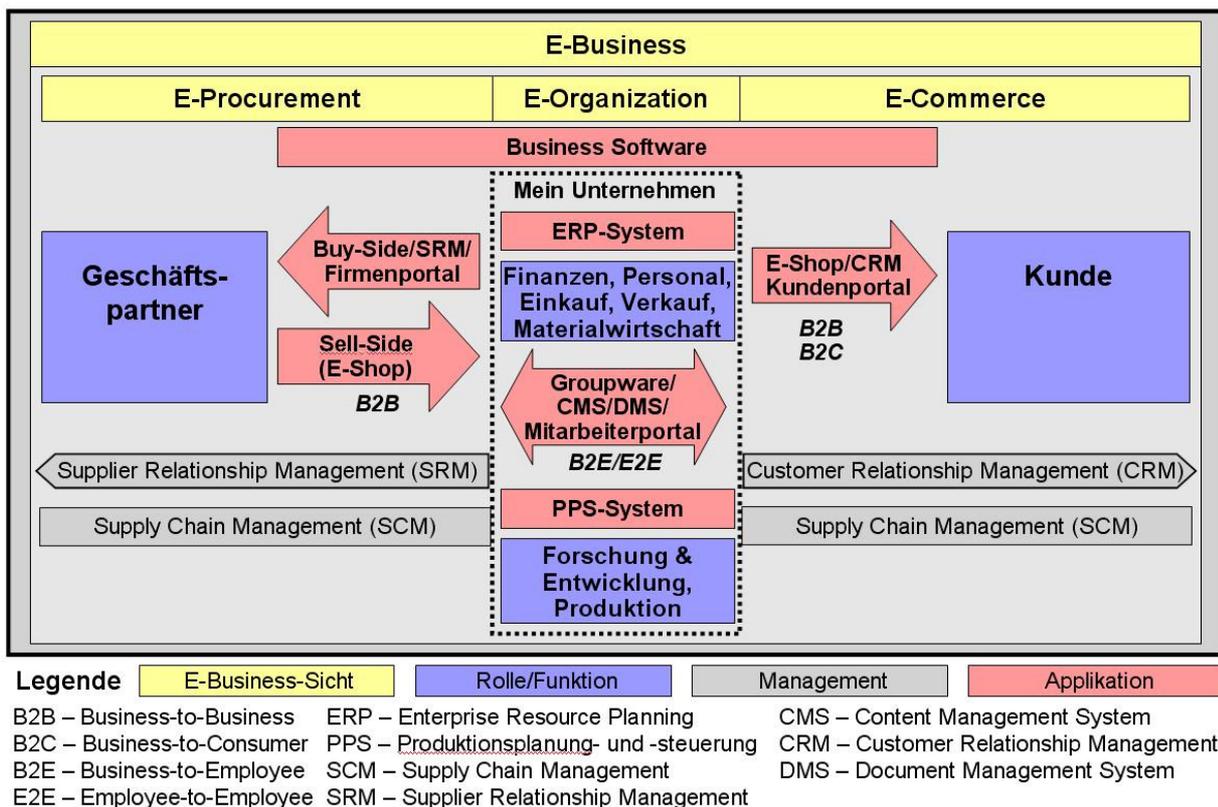


Abbildung 3.3 Begriffssystematik zu Business Software, Quelle: [Wölfle & Schubert, 2006]

3.1.4 Standardsoftware vs. Individualsoftware

ERP-Systeme wie auch viele andere betriebliche Anwendungssysteme gelten als so genannte Standardsoftware. Zu Beginn der Softwareepoche war es üblich, von Kunde zu Kunde eine individuelle Lösung zu erstellen, die in Betrieb genommen wurde (vgl. Kapitel 1.1). Dies änderte sich, als Softwarehersteller wie SAP und andere Anbieter eine Lösung für mehrere, meist ähnliche Probleme anboten. Die Standardsoftware entstand sowohl für private Anwender als auch für große Unternehmen. Gluchowski et al. differenzieren zwischen den beiden Begriffen folgendermaßen [Gluchowski et al., 2008, S. 11]:

„Standardsoftware bezeichnet Programme, die auf Allgemeingültigkeit und mehrfache Nutzung hin ausgelegt sind. Individualsoftware bezeichnet Programme, deren Erstellung für einen bestimmten Anwendungsfall erfolgt und die spezifische Eigenschaft des konkreten Falls adressieren.“

Wie sich in den Interviews später gezeigt hat (vgl. Kapitel 4), ist es nicht immer Standardsoftware, die als betriebliches Anwendungssystem zum Einsatz kommt.

3.1.5 Releasefähigkeit

Die neue Version einer Software bezeichnet man in der Regel als so genannten Release. Neue Funktionalitäten, eine Änderung des Datenbankmodells oder neue Sicherheitsapplikationen können mit einem Releasewechsel ein bestehendes System ändern und aufwerten. Betrachtet man ein betriebliches Anwendungssystem wie z.B. ein ERP-System, so ist ein solch neuer Release relativ schwierig umzusetzen, ohne bestehende Applikationen oder Daten unbeabsichtigt zu verändern bzw. sogar zu schädigen. Bleiben Daten und bereits vorgenommene Customizings (vgl. Kapitel 3.1.6) trotz eines Releasewechsels erhalten, so bezeichnet man das System als releasefähig.

3.1.6 Customizing

Standardsoftwareprodukte entsprechen nicht immer genau den Anforderungen eines Kunden. Manchmal fehlen Funktionalitäten, oder bereits vorhandene Applikationen müssen um weitere, kundenspezifische Wünsche erweitert werden. Eine solche Anpassung an betriebsspezifische Gegebenheiten und Anforderungen nennt man Customizing [Gluchowski et al., 2008, S. 12]. Ist die Releasefähigkeit (vgl. Kapitel 3.1.5) aufgrund von Anpassungen innerhalb des Systems nicht mehr gegeben, so nennt man diese Anpassungen nicht mehr Customizing, sondern Individualentwicklungen.

3.1.7 Legacy-System

In sehr vielen Unternehmen gibt es nicht nur ein Anwendungssystem, sondern es existieren mehrere nebeneinander. Soll ein neues System aus bestimmten Gründen eingeführt werden, so bleiben in der Regel die „alten“ Systeme weiterhin bestehen. Legacy-Software bzw. Legacy-Systeme sind somit jene Softwaresysteme, die entsprechend einer „alten“ Enterprise Architektur installiert wurde [Masak, 2005, S. 95]. Liebhart formuliert es im Kontext der Unternehmung [Liebhart, 2007, S. 182]:

„Ein Legacy-System ist ein sozio-technisches System, das Legacy-Software enthält“.

Es spiegelt also auch die Art und Weise wider, wie ein Unternehmen zu einem bestimmten Zeitpunkt arbeitet. Die Software kann also veraltet sein, während der Kontext (z.B. Geschäftsprozesse, Organisation des Unternehmens) durchaus aktuell ist.

3.2 Architektur

In der Architektur werden nun bestimmte Begriffe eingeführt, die im späteren Verlauf und in der Befragung der Unternehmen selbst eine wichtige Rolle gespielt haben.

3.2.1 Planungsebene

Die Planungsebene beschäftigt sich sowohl mit den Entwurfsmöglichkeiten, die ein ERP-System seinem Nutzer bietet, als auch deren praktische Umsetzung und Modifikationsmöglichkeiten. Im Fokus der Betrachtung stehen dabei die Geschäftsprozesse, welche als planerische Grundlage eines betrieblichen Anwendungssystems eine sehr wichtige Rolle spielen.

3.2.1.1 Geschäftsprozesse

Traditionell sind Unternehmen funktionsorientiert gegliedert [Allweyer, 2005, S. 13]. Das heißt, es gibt Abteilungen, in denen gleichartige Tätigkeitsfelder zusammengefasst sind wie z.B. in der Vertriebsabteilung, in der Verkauf und Auslieferung untergebracht sind. Weitere Abteilungen können Produktion, Rechnungswesen, Einkauf usw. sein. So ist eine effiziente Ressourcennutzung durch Spezialisierungen in den Abteilungen möglich. Entsprechend wurden für jede Abteilung eigene, spezialisierte Informationssysteme geschaffen. Geschäftsprozesse jedoch erstrecken sich meist über mehrere Abteilungen hinweg. Die Folge ist, dass in funktionsorientierten Informationssystemen in jeder Abteilung für den Prozess dieselben Daten vorgehalten werden müssen. Integrierte Informationssysteme schaffen seit den neunziger Jahren schließlich Abhilfe, indem sie abteilungsübergreifende Software zur Verfügung stellen, die auf einer einheitlichen Datenbasis arbeitet [Allweyer, 2005, S. 19]. ERP-Systeme erfüllen diese Anforderungen.

Geschäftsprozesse stehen immer mehr im Fokus aktueller ERP-Systeme. Es findet ein Wandel statt, der sich weg von der funktionsorientierten Sicht hin zur prozessorientierten Sicht auf Unternehmen bewegt [Allweyer, 2005, S. 20]. Für kleine Unternehmen, die eine überschaubare Anzahl von Vorgängen verwalten müssen, ist es nicht unbedingt notwendig, Geschäftsprozesse explizit auszuschreiben. Ist eine derart einfache Verwaltung jedoch nicht mehr möglich, müssen genau beschriebene Vorgänge implementiert werden. Deren automatisierte Umsetzung in lauffähigen Code stellt eine große und komplizierte Aufgabe dar, die einige Anbieter schon teilweise angegangen sind, wie man im Laufe der Arbeit sehen kann.

Definition

Zu Beginn der unternehmerischen Tätigkeit steht die allgemeine Unternehmensstrategie. Dort sind die Eckpfeiler definiert, nach denen sich die gesamte Tätigkeit richtet. Sie dient als Grundlage für das Geschäftsmodell, das die strategischen Zielvorgaben planhaft umzusetzen versucht. Es beschreibt die Struktur des Unternehmens und die damit verbundenen internen Zusammenhänge [Wölflé & Schubert, 2006, S. 6f].

Die Geschäftsprozesse formulieren die operative Umsetzung des Geschäftsmodells (vgl. Abbildung 3.4), das heißt für verschiedene Geschäftsvorfälle, die im Modell denkbar sind, wird eine bewusst gestaltete Abfolge von Aktivitäten festgelegt, die den Geschäftsvorfall abarbeiten soll. Laut Stahlknecht und Hasenkamp kann man einen Geschäftsprozess wie folgt definieren [Stahlknecht & Hasenkamp, 2005, S. 2]:

„Ein Geschäftsprozess (Business Process) ist eine zielgerichtete, zeitlich-logische Folge oder Vorgangskette von Tätigkeiten (Andere Bezeichnungen: Aktivitäten, Geschäftsvorgänge), dessen Ziele aus der Unternehmensstrategie abgeleitet sind.“

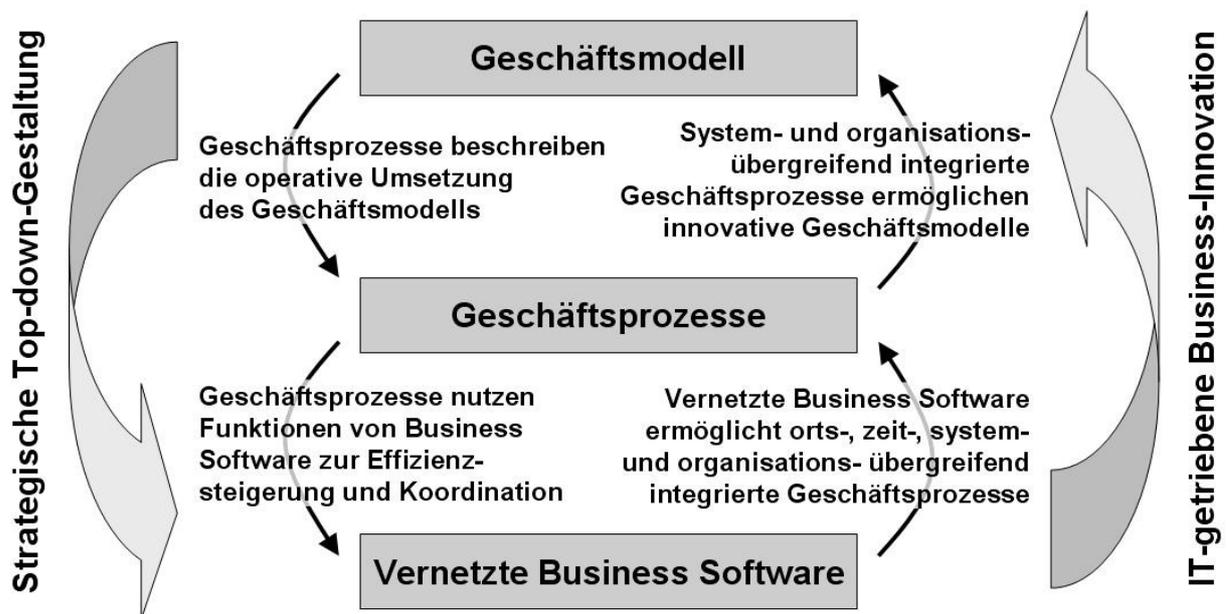


Abbildung 3.4 Bidirektionale Abhängigkeiten, Quelle: Wölfle & Schubert, 2006

Allweyer liefert eine genauere Definition [Allweyer, 2005, S. 51f]:

Ein Geschäftsprozess umfasst eine „[...] zeitlich-logische Abfolge von Aktivitäten zur Erfüllung einer betrieblichen Aufgabe, wobei eine Leistung in Form von Material- und/oder Informationstransformation erbracht wird.“ Diese Definition beinhaltet sowohl interne als auch unternehmensübergreifende Prozesse. Ein Geschäftsprozess hat somit mehrere Eigenschaften [Stahlknecht & Hasenkamp, 2005, S. 3]:

- Der Geschäftsprozess bildet die Basis für die Wertschöpfung
- Jeder Prozess hat einen Auslöser und ein Ergebnis
- Aktivitäten können parallel, wiederholt oder alternativ ausgeführt werden
- An jedem Geschäftsprozess sind meist mehrere organisatorische Einheiten des Unternehmens beteiligt
- Geschäftsprozesse beschreiben Routineaufgaben (wiederholbar)

Somit ist der Geschäftsprozess ideal geeignet, um einen strukturierten Blick auf ein Unternehmen zu gewinnen und planerische Tätigkeiten zu unterstützen. Dank der fortschreitenden Entwicklung, sowohl in der Technik als auch in der Softwareentwicklung, sind durch Vernetzung und Integration neue Möglichkeiten entstanden, Abläufe effizienter zu

gestalten bzw. durch andere zu ersetzen. So gibt die technische Ebene, die eigentlich nur als Hilfsmittel gedacht war, Impulse zur Umgestaltung der sie nutzenden Prozesse, die wiederum Einfluss auf das Modell und letzten Endes auf die allgemeine Strategie haben können (vgl. Abbildung 3.4).

Klassifikation

Eine Klassifikation liefert das St. Galler Modell (vgl. Abbildung 3.5), das Geschäftsprozesse in unterstützende, primäre und Management Prozesse kategorisiert [vgl. Rüegg-Stürm, 2002, S. 69]. Die unterstützenden Prozesse gleichen sich in den meisten Unternehmen bis zu einem gewissen Grad, da sie die Abwicklung von organisatorischen Tätigkeiten beschreiben (z.B. Finanzen, Personal etc.), die nach gleichen Prinzipien verfolgt werden. Sie sollen die Grundlage für die erfolgreiche Durchführung der Primärprozesse bilden.

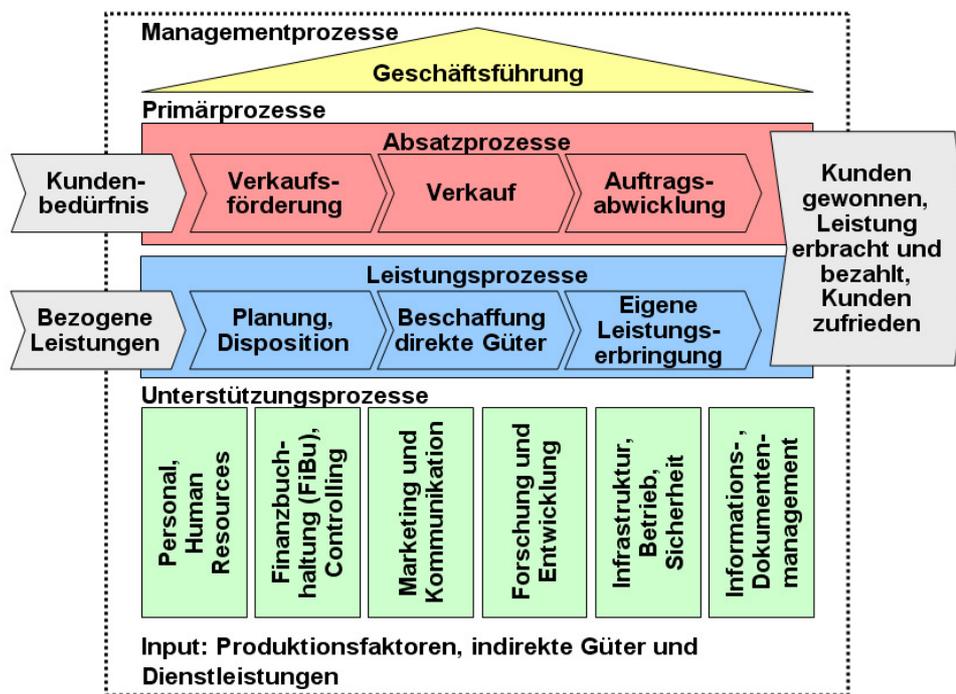


Abbildung 3.5 Prozess-Qualitätsstufen, Quelle: Wölflé, 2006, S. 10

Primäre Prozesse hingegen unterscheiden sich im Gegensatz zu den unterstützenden sehr stark von Unternehmen zu Unternehmen. Sowohl Absatz- als auch Leistungsprozesse beschreiben eventuell einzigartige Tätigkeiten, die so nirgendwo anders zu finden sind. Sie stellen den Kern der unternehmerischen Tätigkeit dar und sind für dessen Existenz ein kritischer Faktor.

Managementprozesse umfassen planerische Tätigkeiten hinsichtlich der Gestaltung, Lenkung und Entwicklung eines Unternehmens. Ein Geschäftsprozess beschreibt also Tätigkeiten. Diese können fachabteilungsübergreifend und sogar unternehmensübergreifend sein. Die Business Software hat nun die Aufgabe, diesen Prozessen die Möglichkeit zur Effizienzsteigerung und zur besseren Koordination der durch sie beschriebenen Tätigkeiten zu geben. Die fachliche Spezialisierung der Mitarbeiter und die interne Aufbauorganisation gerade älterer Unternehmen stellen dabei Hindernisse dar.

Um eine agile Prozessgestaltung zu unterstützen, ist es hilfreich, eine komponentenbasierte Softwareentwicklung zu betreiben [Acker et al., 2004, S. 8]. Anpassungen der Prozesse bedürfen meist immer noch einer manuellen Reorganisation der darunter liegenden Anwendungen, um die neue Struktur umzusetzen. Moderne WfMS, Workflow-Management-Systeme (vgl. Kapitel 3.2.1.4), unterstützen dabei zwar die Modellierung, Überwachung und Steuerung von Geschäftsprozessen, allerdings ist meist keine direkte Verbindung zwischen Prozesslogik und Anwendungscode gegeben [Acker et al., 2004, S. 11]. Datenabhängigkeiten, interne Zustände oder die Transaktionsfähigkeit von Komponenten müssen bei der Prozessmodellierung beachtet werden.

3.2.1.2 Darstellungsmöglichkeiten

Darstellungsmöglichkeiten von Prozessen gibt es viele. Weit verbreitet ist die EPK, die ereignisgesteuerte Prozesskette. Es gibt aber auch noch zusätzlich Blockdiagramme, Hierarchiediagramme, Datenflussdiagramme usw. [Stahlknecht & Hasenkamp, 2005, S. 236-242].

Alle liefern dem Nutzer Informationen über die Tätigkeiten innerhalb eines Unternehmens bzw. wer diese ausführt und wann. Verschiedene graphische Modellierungstools wie z.B. Visio von Microsoft geben dem Benutzer die Möglichkeit, sich die Prozesse in der von ihm gewünschten Form zu visualisieren.

Der nächste, weit ambitioniertere Schritt ist die Einbindung verschiedener Standorte bzw. Systeme in eine Prozesssicht. Standortübergreifende Darstellungen von Aktivitäten spielen auch im mittelständischen Bereich immer öfter eine Rolle. Man sieht sich zum einen gezwungen, flexibel und reaktionsschnell auf dem Markt zu agieren, und auf der anderen Seite sorgt die Anbindung an einen Partner oder eine Tochterfirma oder evtl. steuerliche/rechtliche Gründe dafür, dass man keine zentrale Informationsgestaltung durchführen kann [Stalder, 2007, S. 29]. Die Notwendigkeit einer geschäftsübergreifenden Planung wird größer.

BPM, Business Process Management, hat den Anspruch, eine solche Planung nicht nur graphisch anzubieten, sondern noch einen Schritt weiter zu gehen und die erstellten Prozesse durch Analyse, Design, Modellierung, Implementierung und Management hindurch zu steuern [Strnadl, 2006] (vgl. Kapitel 3.2.1.5). Das umfasst in letzter Konsequenz auch das automatisierte Umsetzen der Prozesse in Applikationen bzw. Services.

3.2.1.3 Workflows

Wie im späteren Verlauf der vorliegenden Arbeit deutlich wird (vgl. Kapitel 4), sind es nicht Geschäftsprozesse, die eine graphische Repräsentation erfahren, sondern Workflows. Der Unterschied zwischen beiden scheint auf den ersten Blick nicht groß zu sein, da sowohl Prozesse als auch Workflows Arbeitsabläufe beschreiben. Allerdings gibt es in ihren Zielen, die sie verfolgen, wichtige Differenzierungsmerkmale. Gadatsch definiert den Workflow folgendermaßen [Gadatsch, 2007, S. 41]:

„Ein Workflow ist ein formal beschriebener, ganz oder teilweise automatisierter Geschäftsprozess. Er beinhaltet die zeitlichen, fachlichen und ressourcenbezogenen Spezifikationen, die für eine automatische Steuerung des Arbeitsablaufes auf der operativen Ebene erforderlich sind.“

Das bedeutet, dass der Geschäftsprozess die fachlich-konzeptionelle Ebene (das „was“), der Workflow hingegen die operative Ebene (das „wie“) beschreibt [Gadatsch, 2007, S. 47].

3.2.1.4 Workflow-Management-Systeme - WfMS

Workflow-Management-Systeme (WfMS) übernehmen die automatisierte Steuerung von Arbeitsabläufen. Sie dienen dazu, elektronische Dokumente über mehrere Mitarbeiter und Anwendungsgrenzen hinweg nach der Bearbeitung einer Funktion weiterzuleiten [Allweyer, 2005, S. 323]. Die Weiterleitung erfolgt aufgrund eines definierten Geschäftsprozesses, wobei während der gesamten Bearbeitung des Arbeitsablaufs eine Kommunikation mit dem beteiligten Anwendungssystem stattfinden kann. Die Vorteile liegen in einem solchen System vor allem bei der Vermeidung von Medien- und Systembrüchen, der Parallelisierung von Teilprozessen und der frei und flexibel änderbaren Prozesssteuerung [Allweyer, 2005, S. 327]. Besonders stark strukturierte Abläufe lassen sich mit einem solchen System gut organisieren. Schwach strukturierte Abläufe, bei denen sich ein Vorgang nie oder kaum ein zweites Mal in der gleichen Weise wiederholt, sind weniger gut geeignet [Allweyer, 2005, S. 328f].

3.2.1.5 Business Process Management - BPM

Eine Ebene über den WfMS steht das Business Process Management (BPM). Sein Anspruch besteht in der ganzheitlichen Betrachtung der Unternehmenstätigkeit. Das Ziel ist es, Effizienz und Effektivität der Geschäftsprozesse und -funktionen zu verbessern [Liebhart, 2007, S. 90]. Dazu werden Prozesse über ihren ganzen Lebenszyklus hinweg beschrieben, gesteuert, ausgeführt und überwacht. Werkzeuge zur Modellierung gibt es viele, z.B. EPK (vgl. Kapitel 3.2.1.2), Petri-Netze oder UML (Unified Modelling Language), allerdings sind diese Repräsentationen der Modelle meist nur graphischer Natur. Die Umsetzung der Prozesse in die Informationssysteme erfolgt oft noch manuell [Liebhart, 2007, S. 91], da Datenabhängigkeiten, interne Zustände oder die Transaktionsfähigkeit von Komponenten beachtet werden müssen [Acker et al., 2004, S. 11].

3.2.1.6 Business Process Execution Language - BPEL

Die service-orientierte Architektur (SOA, vgl. Kapitel 3.2.2.7) erlaubt es, dank der Bereitstellung der Anwendungen durch Web Services (vgl. Kapitel 3.3.2.4) in Zusammenarbeit mit der Business Process Execution Language (BPEL), einer standardisierten Beschreibung der

Prozessdefinitionen [Allweyer, 2005, S. 347], ausführbaren Code (die Steuerung der Workflow Engine) aus den graphisch modellierten Prozessen gewinnen zu können [Liebhart, 2007, S. 91]. Ein BPEL-Prozess besteht dabei aus einem Prozess-Interface (in WDSL formuliert (vgl. Kapitel 3.3.2.4), da jeder BPEL-Prozess selbst einen Web Service darstellt) und einem Prozess-Schema, das den eigentlichen Ablauf, beteiligte Partner, die Art und Weise der Instanziierung und die Fehlerbehandlung definiert [Liebhart, 2007, S. 92].

Es gibt jedoch zwei Aspekte, die BPEL-Werkzeuge nicht oder nur unzulänglich behandeln: die Ausführung von Prozess-Schritten durch den Menschen und das Management ganzer Geschäftsprozessmodelle (z.B. Prozess-Metamodelle, Lifecycle Management usw.) [Liebhart, 2007, S. 93]. Entsprechende Erweiterungen bieten die großen Hersteller wie SAP oder IBM als proprietäre Lösung im „Human Interaction“ Bereich an. Für das Management von Geschäftsprozessmodellen werden ebenfalls Standards von Organisationen wie der OMG (BPML, Business Process Management Initiative) oder der BPM Group (BPMN, Business Process Modelling Notation) entwickelt und vertrieben.

3.2.1.7 Prozessstandards

Standards in den Prozessabläufen selbst sind abhängig von den betrachteten Unternehmen schwierig umzusetzen. Es gibt aber Bestrebungen, einheitliche Standards, zumindest in der unternehmensübergreifenden Prozessbeschreibung, in dieser Richtung einzuführen. GS1, eine Organisation aus Deutschland, ist nach eigenen Angaben „das Dienstleistungs- und Kompetenzzentrum für unternehmensübergreifende Geschäftsabläufe in der deutschen Konsum- und Gebrauchsgüterwirtschaft“ [GS1, 2008].

Allgemeinere Ansätze verfolgen RosettaNet oder ebXML (vgl. Kapitel 3.3.2.3). Die auf XML basierenden Prozesssammlungen versuchen, eine einheitliche Grundlage der Prozessmodellierung für Unternehmen zu schaffen. RosettaNet konzentriert sich dabei auf eine allgemeingültige Beschreibung der Abläufe und Regeln einer unternehmensübergreifenden Interaktion und die daran beteiligten Dokumente [Alt & Österle, 2004, S. 138]. Im Gegensatz zu ebXML ist die Initiative auf bestimmte Industrien (z.B. Hightech) fokussiert. ebXML formuliert standardisierte Geschäftsprozessmodelle und in ihnen verwendete Datenaustauschformate, allerdings ausgerichtet auf alle Branchen im Rahmen einer globalen Verwendung. Trotzdem haben nur sehr wenige Anbieter von ERP-Systemen eine Prozessstandardisierung weder intern noch extern umgesetzt, wie die Ergebnisse zeigen werden (vgl. Kapitel 4).

3.2.2 Systemarchitektur

Betriebliche Anwendungssysteme sind immer nach bestimmten Grundprinzipien aufgebaut, nach denen die Daten, Verarbeitungslogik und die Präsentation miteinander interagieren. Zusammen mit den in den Interviews abgefragten grundlegenden Systemkonzepten bilden sie den logischen Kern eines betrieblichen Anwendungssystems.

3.2.2.1 Architekturformen

Die Architektur einer Komplettanwendung für ein Unternehmen entscheidet darüber, wie flexibel und mit welchem Aufwand eine Aufgabe im System erledigt werden kann. Es gibt drei wichtige Architekturtypen, die sich bis heute im Bereich der betrieblichen Anwendungssysteme herausgebildet haben [Masak, 2005, S. 40]: Monolithen, Client/Server und Service-Architektur.

3.2.2.2 Monolithisch

Datenhaltung, Verarbeitungslogik und Präsentation waren früher und sind teilweise immer noch in einem großen Block zusammengefasst, einer monolithischen Architektur [Overhage et al., 2007, S. 38] (vgl. Abbildung 3.6). Terminals ohne eigene Rechenkapazität greifen auf einen Rechner oder großen Rechnerverbund zu. Ein Ersetzen von Teilen des Systems oder Eingliederung von neuen Softwarekomponenten ist sehr schwierig und nur unter einem nicht zu vernachlässigendem Aufwand möglich. Wiederverwendbarkeit, Portabilität oder Flexibilität sind bei solchen Systemen aufgrund der herstellerbezogenen Eigenschaften wie Rechnerarchitektur oder Datenorganisation kaum denkbar [Engel et al., 2002, S. 6].

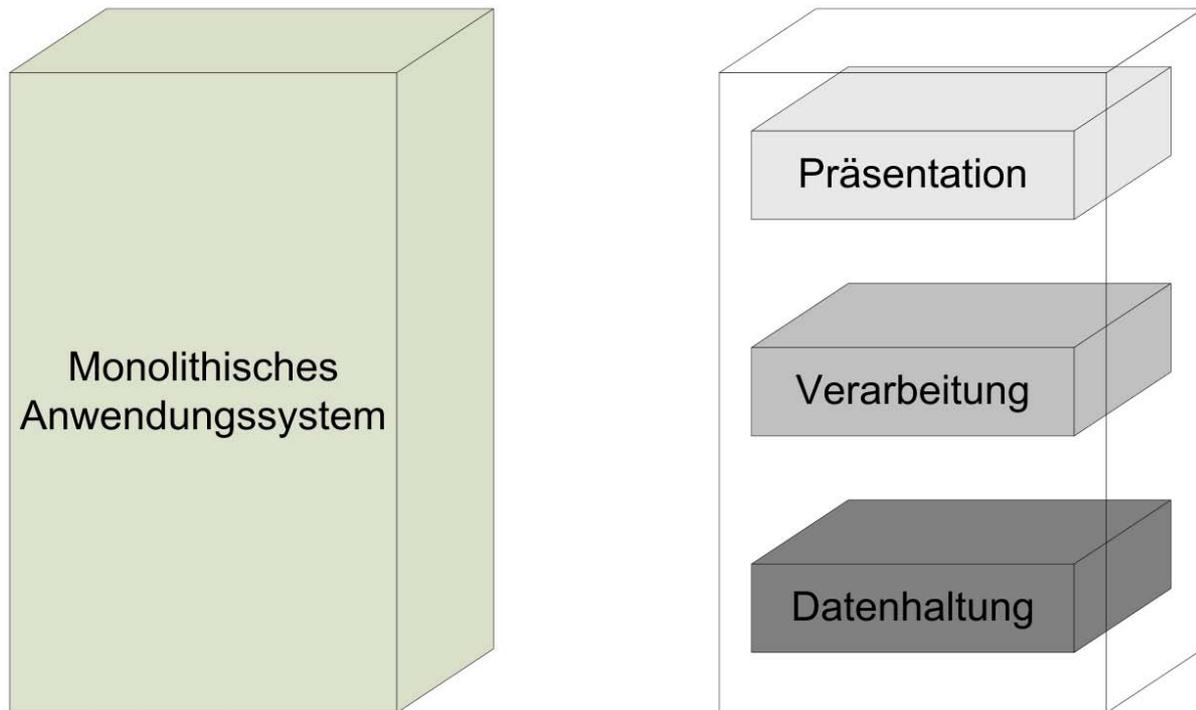


Abbildung 3.6 Monolithische Architektur, Quelle: Overhage et al., 2007

3.2.2.3 Client-Server

Mit der zunehmenden Sensibilisierung hinsichtlich der Wiederverwendbarkeit und Flexibilität eines Systems und den immer stärker zunehmenden verteilten Anwendungen wurde eine neue Sicht auf betriebliche Anwendungssysteme gewonnen [Masak, 2005, S. 112]. Man zerlegte diesen starren Funktionsblock in mehrere Schichten und die drei grundsätzlichen Ebenen Datenhaltung, Verarbeitungslogik und Präsentation auf verschiedene Standorte verteilt [Overhage et al., 2007, S. 44], (vgl. Abbildung 3.7). Ein Server verwaltet die Dienste, die über einen angeschlossenen Client abgerufen werden können. Dabei kann ein Client auch zum Server werden. Wichtig ist der Bedarf, einen Service abzurufen und die Bereitschaft, einen Service anzubieten [Geihs, 1995].

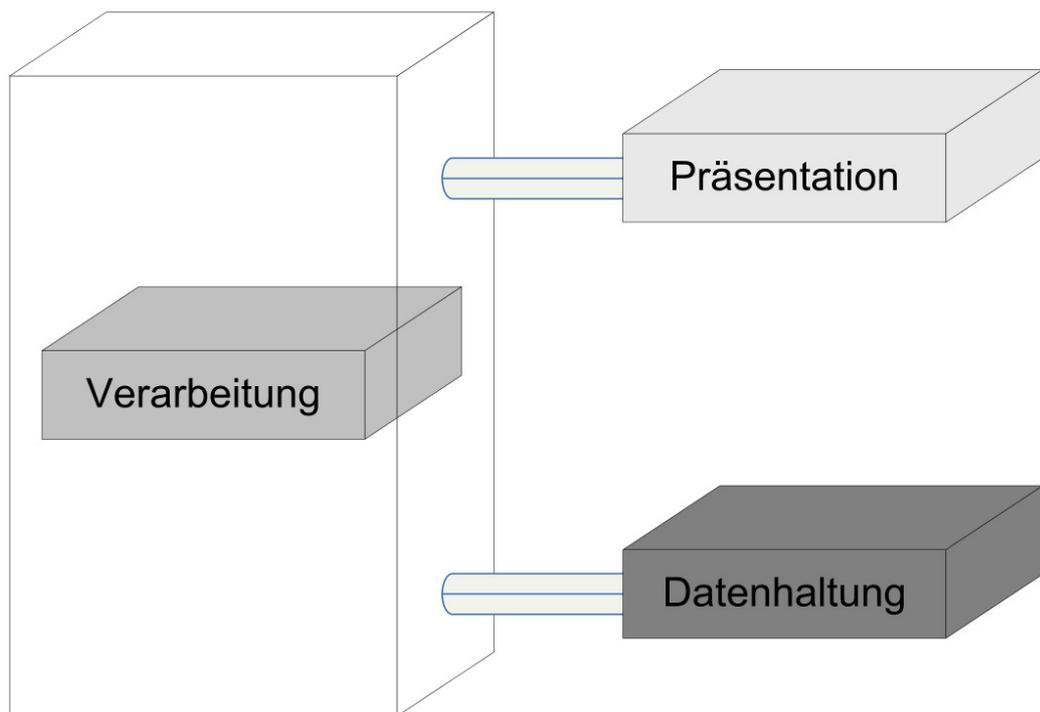


Abbildung 3.7

Client-Server Architektur, Quelle: Overhage et al., 2007

Man kann außerdem zwischen so genannten Fat-Clients, Rich-Clients und Thin-Clients unterscheiden (vgl. Abbildung 3.8).

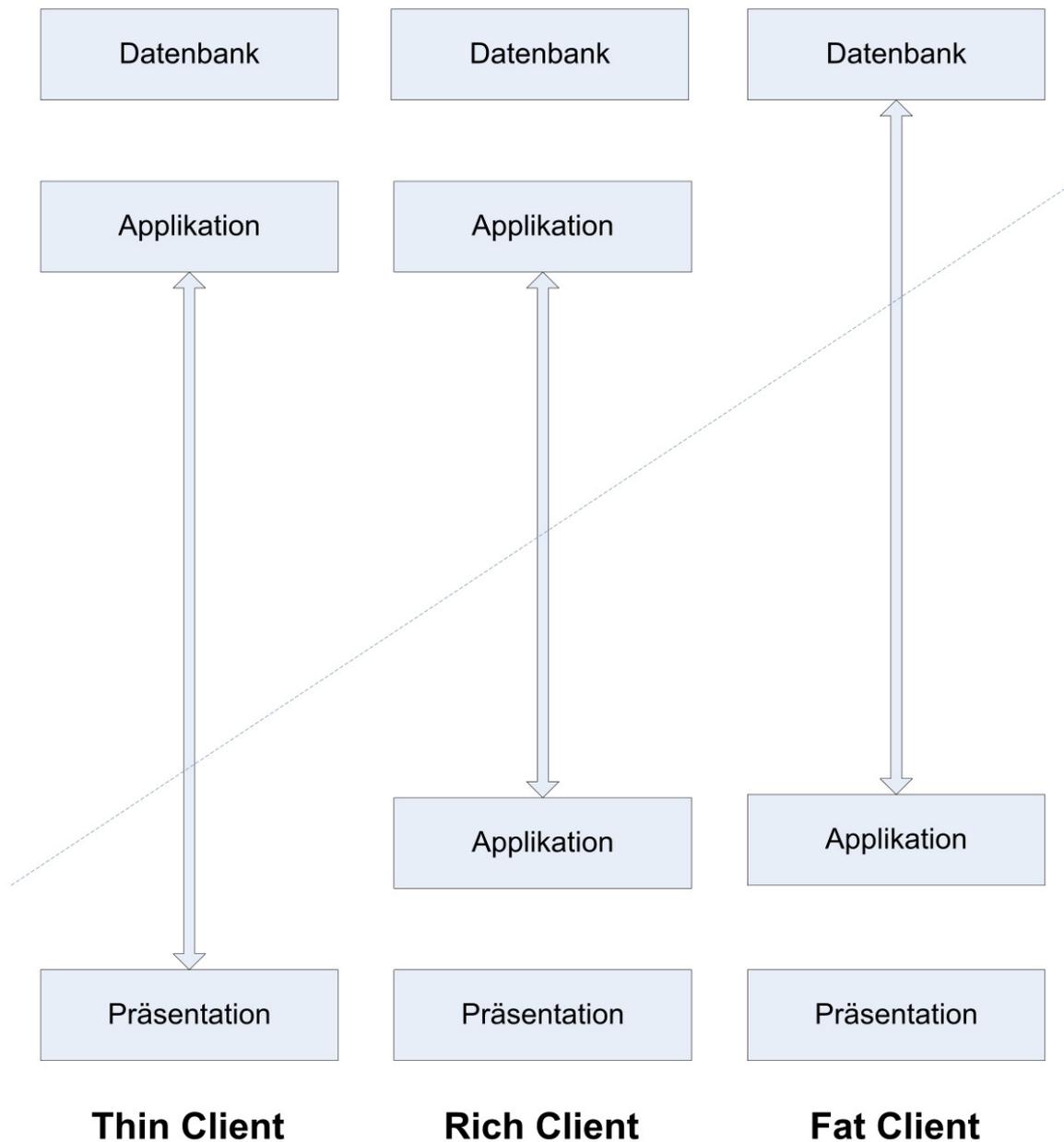


Abbildung 3.8 Verschiedene Client-Ausprägungen, Quelle: Overhage et al., 2007

Thin-Clients haben typischerweise nur eine Ausgabemöglichkeit der Daten, beispielsweise Browserfenster. Rich-Clients besitzen einen Teil der Funktionalität bereits auf ihrer Seite. Beispiel hierfür sind kleinere Java-Programme, die im Browser verankert sind. Fat-Clients können nicht nur die Ein- und Ausgabe wie Thin-Clients erledigen, sondern besitzen eine eigene Rechenleistung zur Verarbeitung von Aufgaben, Konfigurationsmöglichkeit der

Anwendungen usw. [Masak, 2005, S. 117]. Hier sind die graphische Oberfläche und die Applikationen typischerweise aus einem Guss. SAP R/3 basiert z.B. auf dieser Architektur.

3.2.2.4 Spezialisierungen

Man kann die Ebene „Verarbeitung“ nun in weitere Schichten unterteilen: nach dem Funktionsbereich, der Dienste im System anbietet, und der Ablauflogik, die bestimmt, in welcher Reihenfolge die einzelnen Schritte abgearbeitet werden [Overhage et al., 2007, S. 48], (vgl. Abbildung 3.9). Hier findet das Workflow-Management Anwendung, das nach Allweyer die Aufgabe hat, „die Bearbeitung von Geschäftsprozessen über mehrere Mitarbeiter und Software-Anwendungen hinweg zu unterstützen“ [Allweyer, 2005, S. 323].

Innerhalb des WfMS, des Workflow-Management-Systems (vgl. Kapitel 3.2.1.4), kann man die Prozesse definieren, die das System unterstützen soll. Die Workflow-Engine koordiniert die Ausführung der modellierten Prozesse und organisiert die Kommunikation mit den verschiedenen Clients. Benötigen die zuständigen Clients gegebenenfalls für die Ausführung ihrer Tätigkeit andere Anwendungen, koordiniert die Workflow-Engine ebenfalls diesen Vorgang [Allweyer, 2005, S. 331].

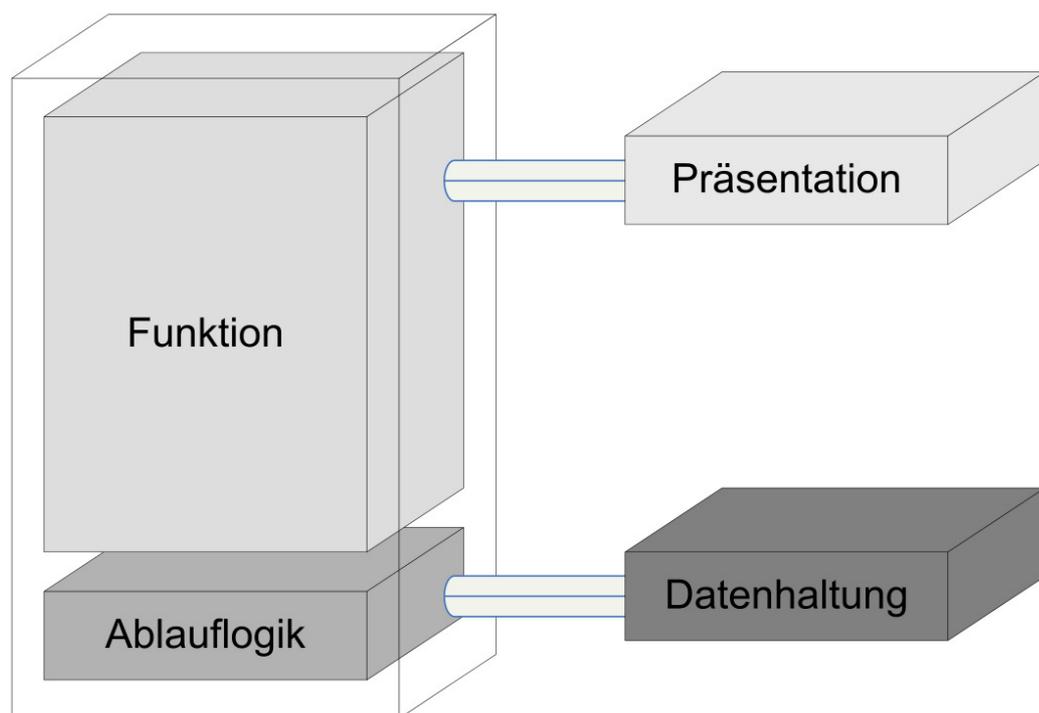


Abbildung 3.9 Workflow Spezialisierung, Quelle: Overhage et al., 2007

Als Mediär zur Kommunikation zwischen den Ebenen „Präsentation“, „Ablauflogik“, „Datenhaltung“ und „Funktion“ kann man außerdem einen Vermittler identifizieren, der Transaktionskonsistenz und eine eventuelle Konvertierung von Datenformaten übernimmt. In diesem Beispiel der Object Request Broker von CORBA [Overhage et al., 2007, S. 51], (vgl. Abbildung 3.10).

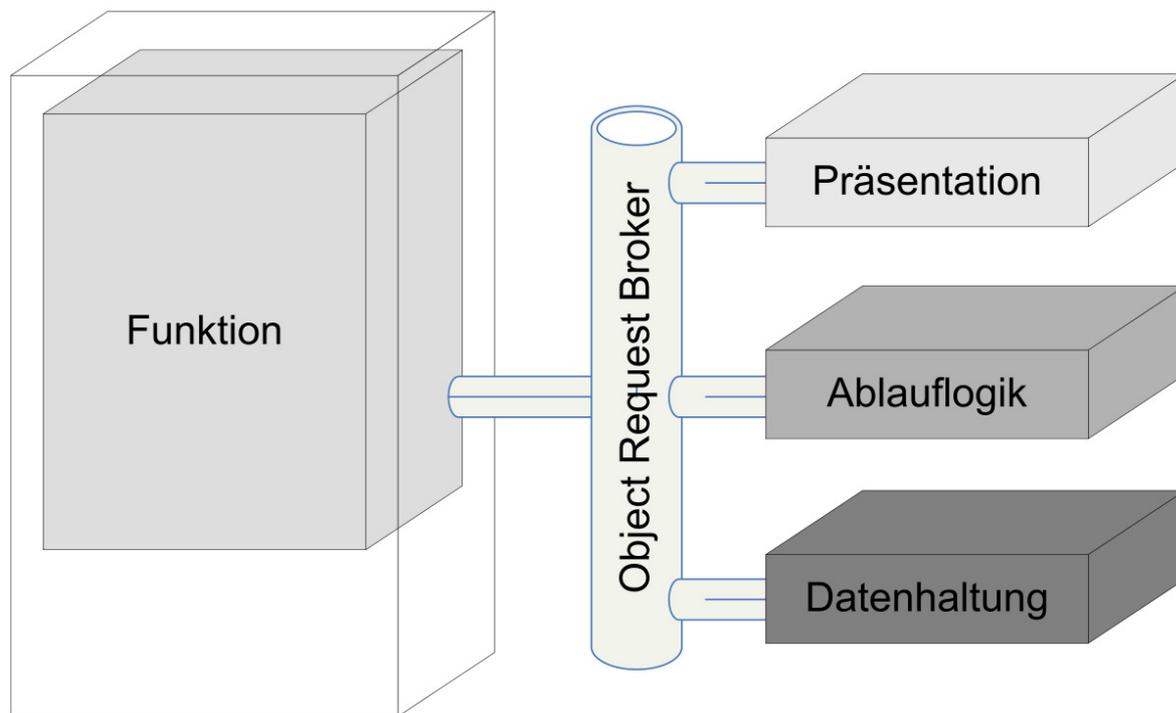


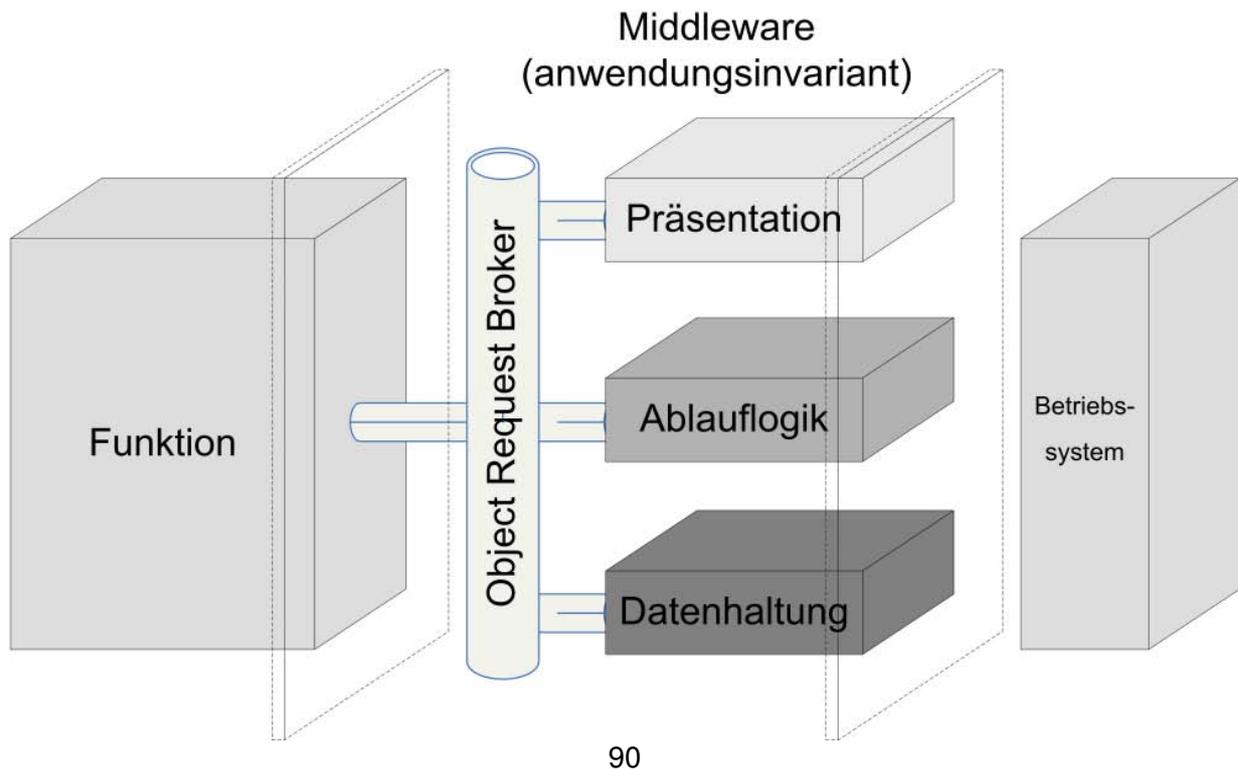
Abbildung 3.10 Vermittler Spezialisierung, Quelle: Overhage et al., 2007

Man erkennt, dass sich im Verlauf der Entwicklung von monolithischen Systemen hin zur Client-Server Architektur bereits eine Komponenten-Sichtweise durchgesetzt hat. Die in dieser Phase entstandene Middleware trägt dem Rechnung.

3.2.2.5 Middleware

Im Zuge der immer stärker werdenden Verteilung von Anwendungen auf mehrere Standorte wurde es notwendig, eine weitere Ebene an Unterstützungssoftware einzuführen, deren Hauptaufgabe das Sicherstellen der Kommunikation der verschiedenen Anwendungen in heterogenen Netzen (insbesondere Client-Server Systeme) ist [Stahlknecht & Hasenkamp, 2005, S. 75].

Die so genannte Middleware fügt sich zwischen dem Betriebssystem und der eigentlichen Anwendungsschicht als Mittler ein [Overhage et al., 2007, S. 53], (vgl. Abbildung 3.11), deren Aufgaben darin bestehen, den Transport komplexer Daten sicherzustellen (Messaging), Funktionsaufrufe zwischen verteilten Komponenten zu organisieren und Transaktionssicherheit herzustellen [Stahlknecht & Hasenkamp, 2005, S. 20]. Im Hinblick auf verteilte Komponenten war ein früherer Ansatz zur Umsetzung das so genannte RPC (Remote Procedure Call) von Sun Microsystems für dessen NFS (Network File System).



90

Abbildung 3.11

Middleware, Quelle: Overhage et al., 2007

Clients haben Funktionsaufrufe bei einem Server getätigt, der die benötigte Prozedur implementiert hatte. Um einen Funktionsaufruf zu starten, musste man eine Nachricht mit der Funktions-ID und dem Parameter an den Server senden.

Eine Verbesserung brachten die Object-RPC-Frameworks wie CORBA, DCOM oder RMI. Im Grunde ist allen gleich, dass sie ein lokales Interface des geforderten Dienstes auf dem Client anbieten, so als wäre die Applikation auf dem Clientrechner. Sie setzen einen zentralen Server als Vermittler in ihr Framework, der als Broker die Kommunikation zwischen den verschiedenen Anwendungen organisiert [Serain, 2002]. Allerdings ist DCOM (Microsoft) nur auf Windows Systemen einsetzbar und RMI (Java) nur in der Lage, Java-Plattformen zu unterstützen.

3.2.2.6 Applikationsserver

Im Zuge der konzeptionellen und auch physischen Trennung von Client, Anwendung und Datenbank entstand der Begriff Applikationsserver, der jedoch im Prinzip dieselben Aufgaben übernimmt wie die zuvor beschriebene Middleware. Laut Masak bestehen die wichtigsten Aufgaben eines Applikationsservers darin [Masak, 2005, S. 161],

- transaktionsorientiert zu arbeiten
- effektive Persistenz durch Integration bestehender Datenhaltungssysteme zu erreichen
- ein Messaging-System und einen Object Request Broker zur Verfügung zu stellen und

- einen Webserver zu stellen.

Die Aufgaben sind demnach fast gleichwertig mit denen der Middleware aus Kapitel 3.2.2.5. Hinzu kommt noch der Webserver, der in Zeiten der globalen Vernetzung seine Existenzberechtigung in der allgemeinen Architektur erhält. Sehr weit verbreitet und oft genutzt sind für Applikationsserver das J2EE (Java Enterprise Edition) Framework und das .NET Framework von Microsoft. Beide sollen nun im Folgenden kurz vorgestellt werden. Außerdem soll als Wegbereiter dieser Technologien CORBA (Common Object Request Broker Architecture) erwähnt werden, die früh sprachunabhängig objekt-orientierte Methodenaufrufe über Prozess- und Maschinengrenze hinweg erlaubte [Engel et al., 2002, S. 76].

CORBA

Die 1989 gegründete Object Management Group (OMG) versuchte, mit CORBA eine standardisierte Architektur zu erstellen, die Plattformneutralität und Interoperabilität sicherstellen sollte [Masak, 2005, S. 145]. Die drei wesentlichen Mechanismen sind IDL (Interface Definition Language), IOP (Internet Inter-Orb Protokoll) und ORB (Object Request Broker).

Die IDL beschreibt die Schnittstellen eines Service oder einer Komponente. Daraus werden beim Client ein so genannter Stub und beim Server ein Skeleton generiert. Der Stub dient als lokales Interface, über den der Client die Funktion beim Server aufruft, als läge sie bei ihm lokal vor. Das Skeleton wiederum beherbergt die eigentliche Funktion. Via dem IOP erfolgt die Übertragung der Aufrufe. Der ORB wird bei Client und Server als Service installiert und leitet die Nachrichten an die betreffenden Stellen weiter. Dies ist unabhängig davon, wo die beiden Kommunikationspartner sich im Netzwerk befinden oder unter welchem Betriebssystem sie laufen (vgl. Abbildung 3.12). Die aktuelle Spezifikation findet man unter [OMG, 2008].

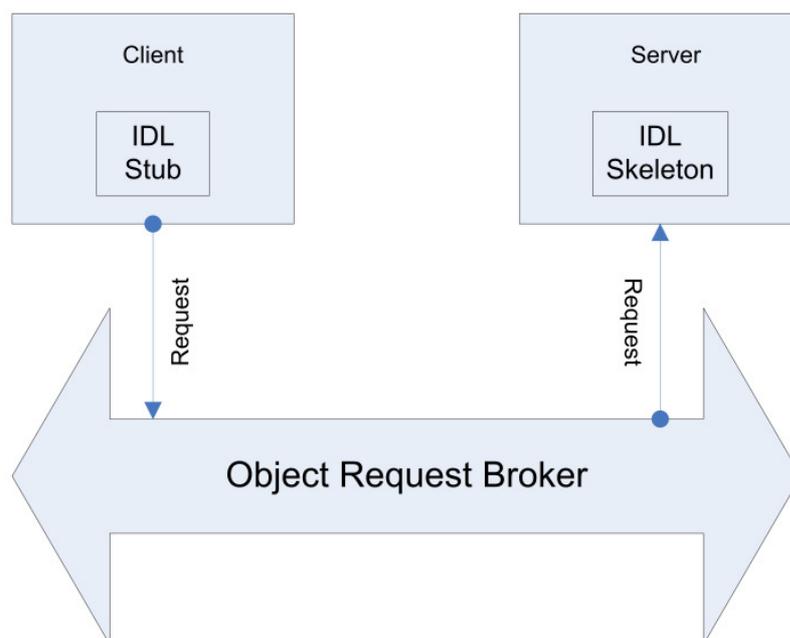


Abbildung 3.12 Beispielhafter Request-Aufruf eines Clients an einen Server unter CORBA

J2EE

J2EE liegt die Sprache Java zugrunde, die von der Firma Sun Anfang der 90er Jahre mit dem Ziel entwickelt wurde, eine Softwareplattform zu schaffen, die auf verschiedenen Rechnersystemen lauffähig sein sollte [Engel et al., 2002, S. 1].

Es wurde schnell klar, dass Java auf der Browserseite als Ausführungsstelle ungeeignet war, da die entsprechenden Bibliotheken und Klassen erst einmal heruntergeladen werden mussten, bevor sie zur Ausführung kommen konnten. Daraufhin wurden die Servlets eingeführt. Dies waren Java-Programme, die auf der Serverseite von Clients gestartet wurden und Ergebnisse zurückliefern konnten [Masak, 2005, S. 184]. Mit der Zeit kamen weitere Komponenten wie Enterprise Java Beans (EJB) hinzu, bis schließlich die J2EE Spezifikation als standardisierte, komponentenbasierte Architektur eingeführt wurde.

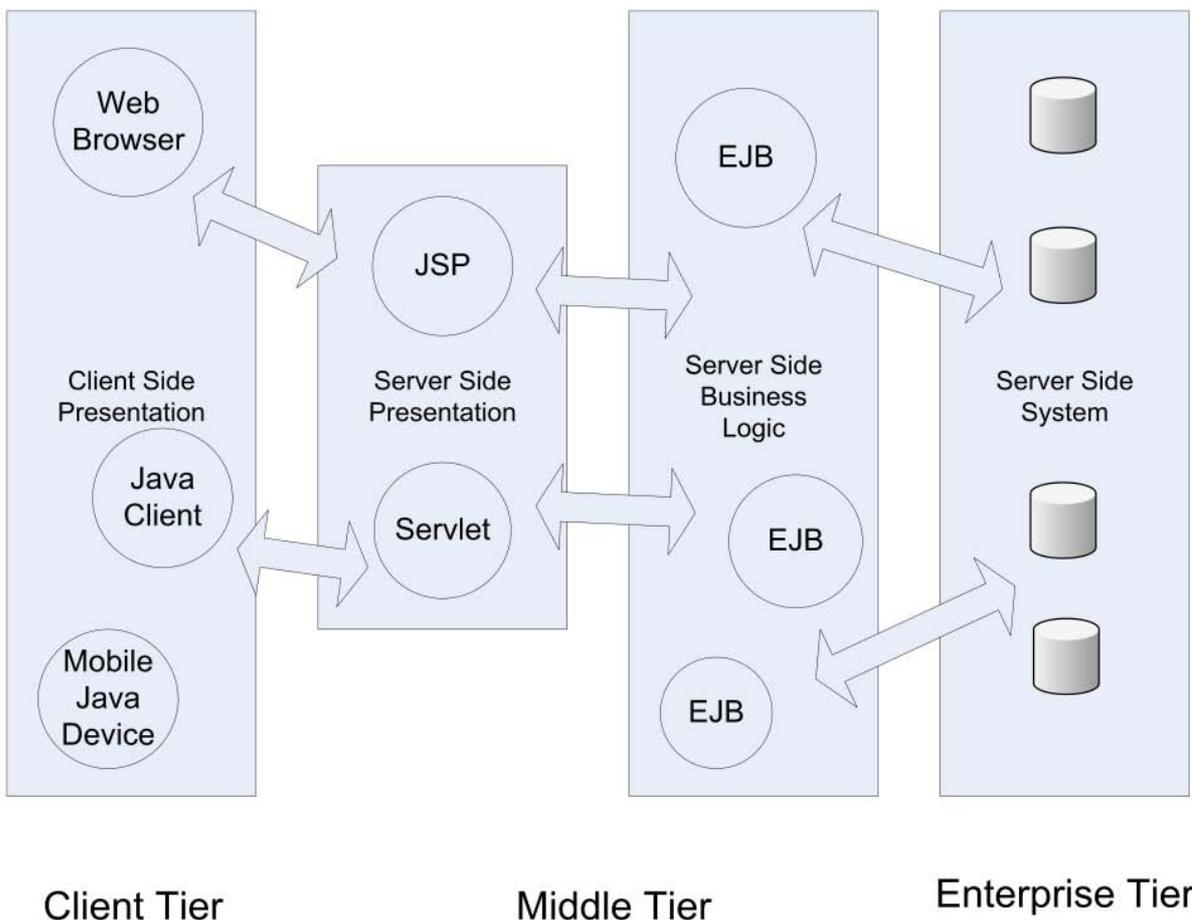


Abbildung 3.13 Schichten-Sichtweise auf J2EE

Der Blick auf die einzelnen Komponenten erfolgt in Schichten (vgl. Abbildung 3.13). Die Client-Schicht können Java fähige Browser, mobile Devices oder richtige Java Programme bilden. Die Enterprise-Schicht repräsentiert das eigentliche System, auf dem die J2EE Umgebung läuft. Die mittlere Schicht setzt die komponentenorientierte Sicht der J2EE

Spezifikation um. Dabei unterscheidet man zwischen einer Präsentationsschicht und einer Business-Logik-Schicht für den Server:

Java Server Pages (JSP) (kleine Javaprogramme innerhalb des Clients) und Servlets (webfähige Programmfragmente auf der Serverseite) können Clientanfragen erst einmal auf Konsistenz (z.B. bei der Eingabe von Finanzdaten) prüfen, bevor sie an die eigentliche Anwendungen, die Enterprise Java Beans (EJB), weitergeleitet werden [Masak, 2005, S. 189]. Diese wiederum sind in so genannten Containern gekapselt, die infrastrukturelle Aufgaben wie Transaktionen und Sicherheitsmerkmale für die Anwendungen organisieren. Die Funktion, in der die eigentliche Geschäftslogik untergebracht ist, kann nur über ein Interface, das vom Container nach außen hin publiziert wird, erreicht werden.

So ist die Wiederverwendung der EJBs auf anderen Systemplattformen und deren lose Kopplung sowohl zur Clientseite als auch zur darunter liegenden spezifischen Plattform garantiert. Die Produktpalette reicht von Open Source Umsetzungen wie JBOSS bis hin zu proprietären Angeboten von IBM Websphere oder BEA Weblogic.

.NET

Im Jahr 2000 brachte Microsoft .NET auf den Markt, um größere Anteile auf dem Servermarkt vor allem gegenüber J2EE zu gewinnen [Masak, 2005, S. 208]. Grundidee ist es, sämtliche Datenübertragungen zwischen den einzelnen Komponenten via Web Services und dem damit verbundenen Dokumentenstandard XML zu bewältigen. So benötigt man keine homogene Infrastruktur mehr, wie bei der vorhergehenden Middleware DCOM. Das Ziel dieses Frameworks ist es, verschiedene Microsoft Produkte in einem Produkt bündeln zu können. Dazu gehören MS SQL, MSMQ als Messaging Lösung oder der Enterprise Server als Application-Server (z.B. SQL Server oder Biz-Talk Server).

Die so genannte Microsoft-Strategie zur Erreichung des Marktzugangs von .NET besteht aus vier Pfeilern [Masak, 2005, S. 209f]:

- .NET Plattform: Eine Laufzeitumgebung, die es vielen verschiedenen Sprachen erlaubt (orientiert nach einem Common Type System (CTS)), in eine Intermediate Language (IL) compiliert zu werden, die wiederum auf einer für alle Sprachen gleichwertigen Common Language Runtime (CLR) zur Ausführung kommt.
- .NET Devices: Entwicklungsumgebung für spezielle Endgeräte wie Smartphones, PDA usw.
- .NET My Services: Eine benutzerorientierte Sammlung von Web Services wie z.B. myCalendar, myWallet usw.
- .NET Enterprise Server: Eine Sammlung von Server-Produkten wie z.B. MS BizTalk, MS SQL-Server, MS Application Center

Produkte auf Basis von .NET gibt es ebenfalls im Open Source Bereich, z.B. Yuhana (seit 2004 [Sourceforge, 2008]). Im proprietären Feld gibt es bislang nur den unter Windows laufenden COM+, der noch auf der DCOM Technologie beruht, allerdings .NET Anwendungen mit einbinden kann [Westphal, 2003].

3.2.2.7 Service-Orientierte Architektur (SOA)

Um den Begriff SOA genauer einzugrenzen, werden im folgenden zwei Referenzmodelle einer service-orientierten Architektur vorgestellt und anschließend das Verständnis der großen Softwareanbieter wie SAP oder Microsoft kurz erläutert. Denn wie sich in den Interviews gezeigt hat (vgl. Kapitel 4), ist SOA kein allgemein präzise definierter Begriff.

SOA ist vielmehr im Bereich der betrieblichen Anwendungssysteme ein oft verwendetes Schlagwort, um die Qualität eines Systems und die Beteiligung an den neuesten Entwicklungen klar herauszustellen. Es wird meist der Eindruck erweckt, dass es sich um eine Technologie oder eine konkrete Architektur handelt.

Referenzmodelle

Dabei ist die Service-Orientierte Architektur eigentlich ein Paradigma, wie die Oasis (Organization for the Advancement of Structured Information Standards) in ihrer Definition beschreibt [Oasis, 2008, S. 8]:

„Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains.”

Es geht also darum, Dienste zu organisieren und den Anwendern zugänglich zu machen. Dabei spielen fünf Aspekte eine wichtige Rolle [Liebhart, 2007, S. 32]:

- Sichtbarkeit der Dienste. Man benötigt Beschreibungen der Funktionalität, technische Voraussetzungen und Mechanismen für den Zugang. Außerdem muss die Beschreibung derart transformierbar sein, dass Syntax und Semantik weithin verstanden werden können.
- Interaktion zwischen Anbieter und Anwender muss sowohl auf technischer als auch auf geschäftlicher Ebene organisiert sein.
- Die Service Beschreibung soll die Interaktion und Sichtbarkeit von Services vereinfachen und automatisieren.
- Vertrag und Richtlinie legen die Bedingungen und deren Rahmen für die Nutzung eines Service fest.
- Effekt resultiert aus der Interaktion. Dabei können neue Informationen entstehen oder Zustandsänderungen folgen.

Wie dieses Paradigma umgesetzt wird, ist in der Definition selbst unerheblich.

Ein weiteres Referenzmodell ist das SOA Meta Model des W3C (World Wide Web Consortium). Dort wird SOA als Ausprägung einer Web Service Architektur beschrieben [Liebhart, 2007, S. 33]. Das Meta-Modell umfasst dabei vier miteinander interagierende Modelle:

- Message Oriented Model: Das Modell beschreibt, wie eine konkrete Implementation eines Service Meldungen sendet und empfängt.
- Resource Oriented Model: Das Modell beschreibt die existierenden Ressourcen.
- Policy Model: Das Modell definiert Grundsätze und Richtlinien für die Interaktion zwischen konkreten Services und Ressourcen.

- Service Oriented Model: Das Modell ist eine Form eines verteilten Systems.

Diese Referenzmodelle stellen eine idealisierte Sicht auf eine service-orientierte Architektur da und beschreiben technologie- und herstellerunabhängige Konzepte, die konkretisiert werden [Liebhart, 2007, S. 30].

SOA Definitionen der großen Hersteller

Die vier großen Softwarehersteller SAP, Oracle, Microsoft und IBM haben jeweils in ihrem Umfeld eine konkrete Definition und Sichtweise auf eine service-orientierte Architektur [Liebhart, 2007, S. 7]:

- SAP definiert SOA als Enterprise Service-Oriented Architecture (ESOA). Modularität der Komponenten und eine Abstraktion der Details der Realisierung liegen der ESOA zugrunde. Über eine standardisierte Konnektivität der Komponenten soll eine Zusammenstellung zu neuen, größeren Prozessen und betrieblichen Abläufen möglich sein.
- Für Oracle ist SOA eine technologische Plattform, um Geschäftsprozesse einfacher in eine IT abbilden zu können. Die Möglichkeit zur Nutzung bestehender Systeme und die lose Kopplung und Kapselung von Services über standardisierte Schnittstellen sind wichtige Eigenschaften einer SOA.
- Microsoft beschreibt SOA als Architektur einer Applikations-Plattform, über die bestehende und neue Anwendungen als normierte Services via Präsentations- und Kollaborationsmechanismen zusammengestellt werden.
- IBM versteht SOA sowohl als Architektur als auch als Programmiermodell. SOA bestimmt die Art und Weise, wie man Software baut. Der Service wird als wiederholbarer Schritt innerhalb eines Geschäftsprozess verstanden.

Die Anbieter haben durchweg eine teilweise sehr ähnliche Definition einer SOA getroffen. Gemeinsam ist allen die Vorstellung eines Service als standardisierte Grundkomponente [Liebhart, 2007, S. 8]. Zudem unterscheiden alle zwischen einer Integrations-Ebene, einer Service-Ebene, einer Orchestrations-Ebene und einer Präsentations-Ebene.

Architektur eines herstellerunabhängigen SOA-Modells

Liebhart beschreibt die Architektur eines herstellerunabhängigen SOA-Modells, welches sich in mehrere Ebenen gliedern lässt [Liebhart, 2007, S. 26]:

- Presentation: Hier ist die Client-Applikation in einer SOA zu finden, die man auch als Servicenehmer bezeichnen kann.
- Orchestration: Hier werden Geschäftsprozesse und Geschäftsregeln abgebildet. Modellierung und Ausführung der Geschäftsprozesse erfolgt durch BPM (Business Process Management, vgl. Kapitel 3.2.1.5) und BPEL (Business Process Execution Language, vgl. Kapitel 3.2.1.6). Geschäftsregeln zur Steuerung des Ablaufverhaltens einer Anwendung werden durch Rule Engines modelliert und ausgeführt.
- Services: In dieser Ebene sind die Mechanismen zur Dienstverwaltung, standardisierte Serviceschnittstellen sowie spezielle Dienste untergebracht. Als Schnittstellentechnik werden Web Services empfohlen (vgl. Kapitel 3.3.2.4).

- **Integration Architecture:** Hier ist die Infrastruktur zur Verknüpfung verschiedener Dienste und zur Verbindung der Dienste mit darunter liegenden Anwendungen angesiedelt.
- **Applications:** In dieser Ebene finden sich bestehende oder neue Anwendungen.

Die wichtigsten Eigenschaften bestehen dabei in der Wiederverwendung gerade älterer Systeme (Legacy-Systeme, vgl. Kapitel 3.1.7), der Modellierung von ausführbaren Prozessen und Regeln und der Verwendung von Web Services und BPEL (vgl. Kapitel 3.3.2.4 und 3.2.1.6) zur Unterstützung der beiden Aspekte [Liebhart, 2007, S. 26].

Eignung

SOA ist somit sehr gut als Erweiterung von bereits bestehenden Anwendungen und Architekturen geeignet, indem durch eine zusätzliche Service-Ebene und Orchestrierungs-Ebene eine heterogene Landschaft geordnet werden kann [Liebhart, 2007, S. 26]. Durch die Kombination mit dem Business Process Management (BPM) (vgl. Kapitel 3.2.1.5) ist eine Trennung von Prozessbeschreibung und Anwendungslogik möglich, wo durch eine neue Flexibilität in der Prozessorchestration entsteht [Allweyer, 2005, S. 345]. Dank der Wiederverwendung können Legacy-Systeme weiterhin eingesetzt werden, wodurch Kosten gespart werden können. Hinzu kommt, dass durch die Ähnlichkeit der verschiedenen Herstellermodelle und deren gemeinsames Verständnis eines Service eine Kombination ihrer Produkte deutlich leichter geworden ist.

Meist werden Web Services als Service-Schnittstelle eingesetzt, aber es können ebenso CORBA oder J2EE-Frameworks als Mittel dienen, um eine service-orientierte Architektur zu verwirklichen. Wichtig ist die Beibehaltung von Interoperabilität und Wiederverwendbarkeit, um die Eigenschaften einer SOA zu bewahren.

3.2.2.8 Plattformunabhängigkeit

Die Plattformunabhängigkeit eines Systems ist eine Eigenschaft, die bereits bei seiner Entwicklung berücksichtigt werden muss. Performance, Interoperabilität und Kosten sind wichtige Aspekte, die für oder gegen die Entscheidung für eine plattformunabhängige Software beachtet werden müssen [Balzert, 2000, S. 990].

Plattformunabhängigkeit kann nur dann erreicht werden, wenn die Software nicht direkt in den Maschinencode übersetzt wird. Nach diesem Prinzip funktionieren Java und auch .NET. Java geht dabei so vor, dass für die verschiedenen Plattformen wie Windows oder Linux speziell darauf abgestimmte Laufzeitumgebungen ausgeliefert werden, die wiederum einen Bytecode ausführen, der aus dem Quellcode in Java kompiliert wurde [Arnold & Gosling, 1998, S. XVI]. So können die Programme in Java beliebig über die Plattformen hinweg ausgetauscht werden.

.NET geht ähnlich vor, allerdings ist man dort nicht auf eine Programmiersprache festgelegt, sondern hat viele verschiedene zur Auswahl wie z.B. C, Visual Basic, Cobol etc. Einschränkungen ergeben sich dabei aus dem CTS, dem Common Type System, dass für jede Sprache eine gemeinsame Typ-Definition zugrunde legt, um die Interoperabilität zu sichern. Anschließend wird der Quellcode in eine IL, Intermediate Language, übersetzt und in der CLR, Common Language Runtime, in den jeweiligen Maschinencode übersetzt [Masak, 2005, S. 212f].

Theoretisch ist .NET plattformunabhängig, wird jedoch im Rahmen der Microsoft Strategie zusammen mit Windows als Betriebssystem vertrieben bzw. ist in der neuen Version bereits

eng damit verwoben. Die Microsoft Produkte sind dadurch natürlich vorab schon gut auf einander abgestimmt, jedoch ist man eben auf diesen Anbieter festgelegt [Balzert, 2000, S. 991].

Die möglichen Betriebssysteme im Serverbereich reichen dabei von UNIX, über Windows bis hin zu Linux. 2007 hatte sich laut Discherl Windows gegen über Linux und UNIX im Serververkauf mit 67 Prozent zu 22,8 bzw. 7,6 Prozent deutlich abgesetzt [Discherl, 2007].

3.2.3 Modularchitektur

Das dritte Unterkapitel der Architekturbetrachtung untersucht die funktionale Mächtigkeit eines betrieblichen Anwendungssystems. Insbesondere sind in diesem Zusammenhang Customer Relationship Management (CRM) und Business Intelligence (BI) als Zusatzfunktionen im Fokus der Betrachtung.

3.2.3.1 Funktionsumfang

Nach Gronau lassen sich ERP-Systeme nach Aufgabenkategorien und Einsatzbereichen einteilen [Gronau, 2004, S. 4]. Es werden nicht nur die betrieblichen Funktionen in einem System zusammengefasst, sondern auch basierend auf einer gemeinsamen Datenbasis die Geschäftsprozesse im Unternehmen über die vorher meist isolierten Fachabteilungen hinaus abgebildet.

ERP Systeme sind also integrierte Anwendungssysteme, die es sich zur Aufgabe gemacht haben, die funktionsorientierte Sicht auf ein Unternehmen zumindest teilweise aufzubrechen und eine prozessorientierte Sicht hinzuzufügen. Trotz ihres Anspruchs auf eine Komplettlösung für Unternehmen werden oft andere Anwendungssysteme neben ihnen eingesetzt, sei es z.B. aus Spezialisierungsbedarf des Kunden, des Einsatzes von Alt-Systemen [Allweyer, 2005, S. 20], oder fehlenden Funktionalitäten innerhalb des Systems.

Anfang des neuen Jahrhunderts führte Gartner, eine weltweit operierende Analystengruppe (vgl. [Gartner, 2008]), den Begriff von ERP II sein. Unter diesem neuen Schlagwort sollte die Integration von unternehmensübergreifenden Systemen zusammengefasst werden, denn eBusiness hatte zu diesem Zeitpunkt den Vormarsch in die Geschäftswelt angetreten und die elektronische Kommunikation sowohl mit Verbrauchern als auch mit geschäftlich verbundenen Unternehmen wurde zu einem wichtigen Faktor im täglichen Geschäft. So wurden neue Funktionalitäten (z.B. Customer Relationship Management) zu den klassischen Aufgaben eines ERP-Systems (Produktion, Finanzen etc.) hinzugefügt [Fuchs, 2008].

Eine detaillierte Modulbetrachtung der untersuchten ERP-Systeme ist aufgrund der teilweise unterschiedlichen Namensgebungen für die gleiche Funktionalität und der bei manchen Anbietern überwältigenden Anzahl an Modulen nicht praktikabel. Um nun ein möglichst umfassendes Bild des Funktionsumfangs eines ERP-Systems zu erhalten, werden die Tätigkeiten eines Unternehmens basierend auf Porter [Porter, 2000, S. 66] in primäre und sekundäre Aktivitäten in der Wertschöpfung eingeteilt. Porter definiert diese folgendermaßen:

„Jedes Unternehmen ist eine Ansammlung von Tätigkeiten, durch die sein Produkt entworfen, hergestellt, vertrieben, ausgeliefert und unterstützt wird.“ [Porter, 2000, S. 67].

Diese Aktivitäten wurden als Grundlage für die Ermittlung des Funktionsumfangs der betrachteten ERP-Systeme gewählt. Die primären Aktivitäten

- Kundenservice

- Beschaffung/Einkauf
- Produktion/betriebliche Leistungserstellung
- Marketing und Vertrieb
- Materialwirtschaft/Warenwirtschaft
- Produktentwicklung

beschreiben die physische Herstellung des Produktes und die dafür notwendigen Maßnahmen, dessen Verkauf bzw. Übermittlung an den Kunden sowie den Kundenservice [Porter, 2000, S. 69].

Die sekundären Aktivitäten

- Finanz- und Rechnungswesen
- Personalwesen
- Geschäftsführung/Management
- Interne Services

beschreiben die unterstützenden Aktivitäten, welche primäre Aktivitäten in ihrer Funktion derart aufrecht erhalten, indem sie unter anderem für menschliche Ressourcen und verschiedene Funktionen für das ganze Unternehmen sorgen [Porter, 2000, S. 69]. In ihrer Grundform besitzen sie für eine große Anzahl an Unternehmen ähnliche oder sogar gleiche Ausprägungen.

3.2.3.2 Customer Relationship Management - CRM

Seit es Unternehmen gibt, ist man darauf bedacht, möglichst viele Kunden zu gewinnen und auch für eine lange Zeitspanne zu binden. Eine Unterstützung dieser Tätigkeiten durch betriebliche Anwendungssysteme war jedoch nicht von Beginn an gegeben [Wölfle & Schubert, 2005, S. 9]. Somit ist eine gesonderte Betrachtung dieser speziellen Funktion innerhalb eines ERP-Systems gerechtfertigt. CRM Systeme bieten den Unternehmen eine bedeutende Unterstützung bei ihrer Kundenorganisation. Dabei verfolgen sie mehrere Ziele [Bach & Österle, 2000, S. 19]:

- Steigerung der Kundenprofitabilität unter
- Nutzung von Potenzialen zur Kundenbindung, -selektion und -gewinnung, wobei gleichzeitig
- Effizienzsteigerungen in CRM Prozessen erzielt werden.

Man kann drei Prozesse in diesem Zusammenhang definieren, denen alle Kundenkontakte unternehmensseitig zugeordnet werden können [Bach & Österle, 2000, S. 23]: Marketing, Verkauf und Service.

Diese Prozesse sind bei ihrer Tätigkeit jedoch nicht in reiner Interaktion mit den Kunden zu sehen. Sie benötigen Daten aus internen Funktionsbereichen des Unternehmens wie z.B. Einkauf, Produktentwicklung, Leistungserstellung etc., um z.B. den Mitarbeitern im Call-Center

die nötigen Informationen zu beschaffen, die den Kunden weiter gegeben werden sollen. Umgekehrt sollen auch Beschwerden oder Anregungen in die internen Prozesse zurück fließen können.

Integration in die bestehende IT-Landschaft ist somit eine essentielle Eigenschaft, die CRM Systeme erfüllen müssen. Die benötigten Informationen, um die drei CRM-Prozesse effektiv gestalten zu können, kommen aus unterschiedlichen Datenquellen wie z.B. Dokumentenmanagement-Systemen oder aus einem Data Warehouse. Über verschiedene Kanäle werden die so gesammelten und zu Informationen verdichteten Daten als Basis zur Kommunikation mit dem Kunden genutzt. Beispiele für Kommunikationskanäle sind Mitarbeiter im Call-Center, E-Mail, mobiler Service etc. [Bach & Österle, 2000, S. 30].

3.2.3.3 Business Intelligence - BI

Neben dem Customer Relationship Management stellt Business Intelligence eine weitere wichtige Säule als informationstechnisches Unterstützungswerkzeug für ein Unternehmen dar.

Das Ziel von Business Intelligence ist es, den Datenbestand, der in einem Unternehmen durch seine operativen Geschäfte anfällt, zu analysieren und aus den so gewonnen Daten den Entscheidungsträgern einen aggregierten Überblick oder sogar Vorhersagen über die Entwicklung ihres Unternehmens zu liefern [Azvine et al., 2005, S. 214].

Der Begriff wurde 1989 von Howard Dresner eingeführt, einem Analysten der Gartner Group (Kotadia, 2006). Die Definition ist in der Literatur schwierig, da sich unter Business Intelligence eher Konzepte verbergen als klare Begriffe. Manche verstehen darunter Report- und Visualisierungssysteme, andere statistische Analysen und Data Mining. Azvine beschreibt Business Intelligence Systeme mit deren typischen Eigenschaften [Azvine et al., 2005, S. 215]:

- Reports und Visualisierung
- Trend Analyse
- Analyse des Kundenverhaltens
- Predictive Modelling

Gluchowski sieht in der gegenwärtigen Praxis ein „weites“ Verständnis von BI etabliert, zu dem alle Systemkomponenten zu Business Intelligence zählen, „die operatives Datenmaterial zur Informations- und letztlich Wissensgenerierung aufbereiten und speichern sowie Auswertungs- und Präsentationsfunktionalität anbieten“ [Gluchowski et al., 2008, S. 91]. Als Grundstruktur sollte ein BI-System dazu ein Data Warehouse, darauf arbeitend analytische Werkzeuge und ein Reporting-System besitzen, das die aggregierten Daten ausgibt.

BI ist somit für die Analyse und Auswertung von Unternehmensdaten, die in einem ERP-System anfallen, essentiell. Aktuelle Entwicklungen auf dem Markt der BI Anbieter zeigen, dass große Anbieter von ERP-Software wie SAP, Oracle oder Microsoft die Notwendigkeit einer dauerhaften Integration von BI-Software in ihre Systeme sehen. Namhafte Anbieter von BI-Lösungen wurden im Jahr 2007 [Zunic, 2008, S. 2], von so genannten „Global Playern“ aufgekauft. Das bedeutet für die ERP-Anbieter, die Lösungen von Business Objects (SAP), Cognos (IBM) oder Hyperion Solutions (Oracle) benutzen, dass sie nun automatisch zu deren Kunden geworden sind.

3.2.4 Integration

In dem letzten Abschnitt der Architekturbetrachtung werden die Systeme auf mögliche Integrationsmerkmale (intern wie auch extern) hin untersucht.

3.2.4.1 Collaborative Business

Betrachtet man nicht nur das Unternehmen selbst, sondern auch seine Partner im Rahmen einer geschäftsübergreifenden Modellierung, so spricht man von Collaborative Business. Kortus-Schultes und Ferfer definieren es folgendermaßen:

„Unter Collaborative Business kann man firmenübergreifende Transaktionen verstehen, die schnittstellenfrei über die gleiche Plattform abgewickelt werden und zugleich standardisierte, automatisierte, korrespondierende Transaktionen bei der jeweils angesprochenen Gegenpartei auslösen. Die Kollaboration, d.h. die Zusammenarbeit, erfolgt online, in Echtzeit, rollenbasiert und in einer automatisierten Übersetzung für alle in einem Geschäftsprozess angesprochenen Partnerrollen.“ [Kortus-Schultes & Ferfer, 2005, S. 158]

Die Kollaboration bezieht sich dabei nicht nur auf die Interaktion zwischen mehreren Unternehmen, sondern auch auf die unternehmensinterne Zusammenarbeit verschiedener Standorte. [Wölfle, 2007, S. 9]

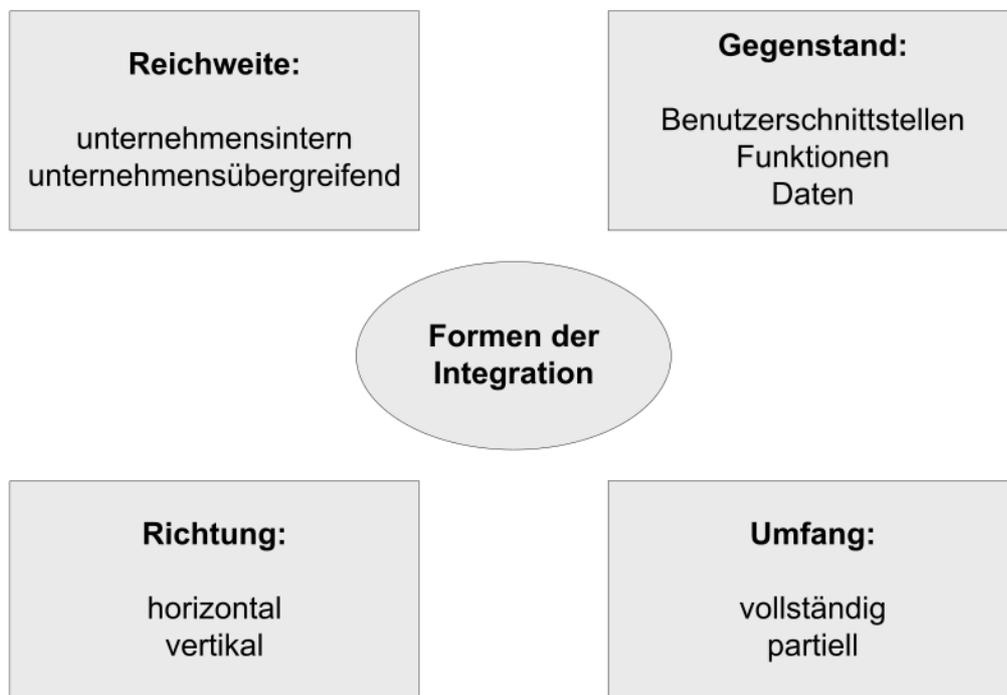


Abbildung 3.14 Formen der Integration

Dies kann von der redundanten Datenhaltung zweier Systeme über die gemeinsam genutzte Datenbank bis hin zu vollintegrierten Anwendungen reichen. Value Added Networks, standardisierte Dokumentformate und neuerdings auch Web Services erleichtern den Informationsaustausch und somit die Möglichkeit, neue Geschäftsbeziehungen und neue Geschäftsprozesse entstehen zu lassen.

3.2.4.2 Integration

Grundsätzlich kann man verschiedene Formen der Integration laut [Mertens et al., 2005, S. 6] unterscheiden: (vgl. Abbildung 3.14)

Eine horizontale Integration findet über mehrere Funktionen bzw. Fachbereiche statt, während die vertikale Integration zu analytischen Zwecken geeignet ist.

Die Reichweite gibt an, ob innerhalb eines Fachbereichs Funktionen miteinander verbunden werden, fachbereichsübergreifend Informationen bereitgestellt werden oder sogar zwischen Unternehmen integriert wird. Letzteres wird aufgrund der globalen Vernetzung und den damit verbundenen Möglichkeiten des eBusiness immer wichtiger.

Der Gegenstand einer Integration kann von den Daten über Funktionen bis hin zu ganzen Programmen gehen. Im Falle der Daten wird dabei eine gemeinsame konsistente Datenbasis herzustellen versucht. Funktionen werden so aufeinander abgestimmt, dass Ergebnisse der einen Funktion direkt von der nächsten weiterverarbeitet werden. Programme zuletzt erhalten eine gemeinsame Schnittstelle, mit deren Hilfe sie miteinander Informationen austauschen können.

Der Umfang beschreibt, in welchem Maße eine Integration stattgefunden hat. Das Ideal ist ein vollständig integriertes Anwendungssystem, in dem sich das Unternehmen korrekt abgebildet wiederfindet. In der Realität jedoch sind meist nur Teilbereiche wie Produktion und Logistik oder Rechnungswesen und Finanzen miteinander verbunden.

3.2.4.3 Enterprise Application Integration - EAI

Betrachtet man die horizontale Integration, so hat man zwei grundsätzliche Möglichkeiten, Funktionen oder Datenbestände usw. zu integrieren: intern und extern.

Die interne Integration nennt man EAI (Enterprise Application Integration). Hauptsächlich kommen solche EAI-Systeme bei bereits vorhandenen älteren Systemen (Legacy-Systeme, vgl. Kapitel 3.1.7) bzw. in einer heterogenen Systemlandschaft zum Einsatz [Masak, 2005, S. 169]. Das Ziel von EAI-Systemen ist die Reduzierung des Aufwandes der Schnittstellenprogrammierung [Allweyer, 2005, S. 341]. Anstatt für je zwei Anwendungssysteme eine separate Schnittstelle zu programmieren, die eine Kommunikation zwischen den Applikationen erlaubt, wird eine zentrale Konvertierungsstelle eingerichtet, die Kommunikationsstandards von verschiedenen Systemen ineinander überführen kann. Doch diese Art der Integration betrachtet nur die Ebene der Datentransformation und -Übertragung. Eine Berücksichtigung des Kontrollflusses, der übergeordneten Geschäftsprozesse, findet nicht statt [Allweyer, 2005, S. 342].

Diesen Nachteil sollen BPMS (Business Process Management Systeme) beseitigen. Sie vereinen Eigenschaften einer EAI und eines WFMS (vgl. Kapitel 3.2.1.4) und vermögen es, die Prozessbeschreibung von der eigentlichen Anwendungslogik zu trennen und eine komponenten- und service-orientierte Neuordnung der Anwendungen in Services zu orchestrieren (vgl. Kapitel 3.2.2.7).

3.2.4.4 Business to Business Integration - B2B

Die externe Integration zielt auf die überbetriebliche Unterstützung von Geschäftsprozessen zwischen mehreren Geschäftspartnern [Allweyer, 2005, S. 348]. Linthicum definiert B2B Integration folgendermaßen [Linthicum, 2001, S. 10]:

„[B2B Application Integration] is, at its foundation, the mechanisms and approaches to allow partner organizations, such as suppliers and consumers, to share information in support of common business events.”

Dabei kann man prinzipiell drei Ausprägungen von Integrationen unterscheiden [Schubert et al., 2005, S. 13]:

- 1:1 Basis, 2 Partner vereinbaren über eine Schnittstelle die Kommunikation
- 1:n Basis, 1 Partner kommuniziert über Schnittstellen, die mit einem (oder mehreren) Standards definiert sind, mit n Partnern
- Die Kopplung von n:m Partnern über einen Intermediär.

Wie auch bei EAI, wird hier mehr als ein einfacher Datenaustausch angestrebt, da die überbetrieblichen Geschäftsprozesse nicht berücksichtigt werden [Allweyer, 2005, S. 354]. Es gibt Initiativen, die eine Standardisierung von unternehmensübergreifenden Geschäftsprozessen anstreben, darunter ebXML (vgl. Kapitel 3.3.2.3) oder RosettaNet (vgl. Kapitel 3.2.1.7).

In Bezug auf die Reichweite steht in dieser Arbeit die unternehmensübergreifende Integration auf dem Prüfstand. Dabei ist die Richtung der Integration horizontal zu sehen, da der Gegenstand im B2B Bereich Daten- und Applikationsebene beinhaltet. Prozessstandards werden in der Planungsebene (vgl. Kapitel 3.2.1.7) extra angesprochen. Der Umfang der Integration kann je nach System von der Bereitstellung standardisierter Dokumentenformate bis hin zu Funktions- und Datenbankschnittstellen reichen.

3.3 Technologie

Der Technologieabschnitt beleuchtet die intern und extern eingesetzten Hilfsmittel, die zum Betrieb eines ERP-Systems notwendig sind. Die interne Betrachtung konzentriert sich dabei auf die technische Basis, auf der ein System zum Einsatz kommt wie auch die Werkzeuge zu dessen Erstellung. Der Fokus im externen Bereich liegt auf Standards, die in der Datenübermittlung zwischen mehreren Kommunikationspartnern zum Einsatz kommen.

3.3.1 Technologie - intern

Im Folgenden werden die Angaben, die besonders oft in den Interviews zu den Bereichen Betriebssystem, Datenbank, Programmiersprache und Entwicklungsumgebung genannt wurden, kurz vorgestellt.

3.3.1.1 Betriebssysteme

Betriebssysteme stellen die Basissoftware für die darauf aufsetzenden Systeme. Sie verwalten die Betriebsmittel und organisieren den reibungslosen Ablauf der Programme [Kofler, 2005, S. 37]. Im Falle von ERP-Systemen muss man zwischen den Betriebssystemen der Clients und der Applikations- bzw. Datenbankserver unterscheiden. Die meisten Clients verwenden Windows-Versionen (Windows NT, Windows XP oder neuerdings Windows Vista) für klassische PCs. Server hingegen benötigen ein entsprechendes Betriebssystem, das die serverspezifischen Betriebsmittel (z.B. Mehrfach-Prozessoren) verwalten kann. Solche Betriebssysteme sind für den Datendurchsatz optimiert, während sich Client Versionen auf die Antwortzeit im Desktop-Bereich konzentrieren [Russinovich & Solomon, 2005, S. 49]. Die

Verteilung zugunsten Windows im Serverbereich (z.B. Windows Server 2003) ist nicht so überwältigend wie im Falle der Clients. Trotzdem haben Windows-Server eine deutliche Mehrheit gegenüber Linux bzw. Unix [Discherl, 2007].

Im Folgenden werden die genannten Betriebssysteme für Server kurz erwähnt, für die klassischen Betriebssysteme im Client Bereich wird auf die gängige Literatur verwiesen:

- 1. Windows: Das von Microsoft angebotene Betriebssystem für Server ist in mehreren Editionen erhältlich, angefangen bei einer Web Edition bis hin zur Datacenter Edition [Russinovich & Solomon, 2005, S. 48]. Die aktuelle Version ist Windows Server 2008.
- 2. Linux Server: Die wohl bekanntesten Server im Linux Bereich stammen von Red-Hat (Enterprise Server, kostenpflichtig) und Fedora-Linux (experimentelle Linux-Distribution, kostenfrei) [Kofler, 2005, S. 1211].
- 3. Unix: Bekanntester Vertreter hier ist HP mit seinem auf UNIX basierendem HP-UX (HP-Unix). Aktuelle Version ist HP-UX 11i v3.

3.3.1.2 Datenbanken

Wie aus den Befragungen zu ersehen ist, werden fast nur relationale Datenbanken in Verbindung mit ERP-Systemen eingesetzt. Die meist genannten waren:

- 1. SQL Server sind relationale Datenbankmanagementsysteme (RDBMS) [Riordan, 2000, S. 11]. Zum einen wird eine Version, MSSQL, von Microsoft vertrieben, zum anderen existiert eine Version des schwedischen Herstellers MySQL AB, MySQL. Sowohl von Microsoft als auch von MySQL AB kann man eine kostenlose, mit Einschränkungen versehene und eine kostenpflichtige Version mit freigegebenen Funktionen beziehen.
- 2. Oracle vertreibt die hauseigene Entwicklung Oracle Database. In der Datenbank können nicht nur relationale, sondern auch objekt-relationale Daten abgespeichert werden [Loney & Theriault, 2002, S. 4f]. Zudem ist eine Verarbeitung von XML Datenstrukturen möglich.
- 3. DB2 wird von IBM vertrieben. Die Datenbank kombiniert ebenso wie die Oracle Database relationale Daten mit einem objekt-orientierten Ansatz [Chamberlin, 1999, S. 26].

3.3.1.3 Programmiersprachen

Im Bereich der Programmiersprache gab es deutlich mehr unterschiedliche Angaben, da Systeme oft nicht nur mit einer Sprache erstellt wurden. Die folgende Liste erläutert kurz die häufigsten Nennungen. Für eine tiefer gehende Beschäftigung mit Programmiersprachen wird auf die klassische Literatur verwiesen:

- 1. Java: Die objekt-orientierte Sprache erzeugt so genannten Bytecode, der auf jedem System durch eine dort vorhandene virtuelle Laufzeitumgebung ausgeführt werden kann. Java ermöglicht die plattformunabhängige Nutzung von Applikationen [Arnold & Gosling, 1998].

- 2. C/C++: Im Gegensatz zu Java ist C eine prozedurale Programmiersprache. C++, das C um objekt-orientierte Konzepte erweitert, bezeichnet man als hybride Sprache [Balzert, 2005, S. 830].
- 3. CSharp: CSharp ist eine von Microsoft entwickelte Programmiersprache. Sie ist auf Microsoft-Plattformen am weitesten verbreitet, so wie Java auf Nicht-Microsoft-Plattformen [Balzert, 2005, S. 829]. CSharp vereint Eigenschaften von Java und C/C++.
- 4. Visual Basic: VB ist eine von Microsoft entwickelte proprietäre, objektorientierte Programmiersprache.
- 5. PL/SQL: Eine proprietäre prozedurale Programmiersprache von Oracle mit starker Datenbanknähe.

Zudem wurden teilweise noch in Cobol oder RPG verfasste Applikationen eingesetzt.

3.3.1.4 Entwicklungsumgebungen

Entsprechend der genannten Programmiersprachen wurden auch die verwendeten Entwicklungsumgebungen angeführt. Besonders häufig wurden folgende Umgebungen genannt:

- 1. Eclipse: 2001 von IBM und anderen Firmen als Open Source Projekt eingeführt, hat sich Eclipse zu einer Plattform für zahlreiche Entwicklungswerkzeuge entwickelt, nicht nur als Java Umgebung [D'Anjou et al., 2005, S. 5].
- 2. Visual Studio: Die von Microsoft entwickelte und in das .NET Framework eingebettete Entwicklungsumgebung unterstützt ebenfalls verschiedene Sprachen, z.B. Visual Basic oder CSharp [Liberty, 2005, S. 16].
- 3. IntelliJ: IntelliJ ist eine proprietäre Entwicklungsumgebung der Firma JetBrains für die Programmiersprache Java.

In diesem Abschnitt wurden recht häufig Eigenentwicklungen angegeben, die zur Erstellung einer Applikation verwendet werden.

3.3.2 Technologie – extern

Ebenso wie bei den technischen Grundlagen, auf denen ein Anwendungssystem beruht, gibt es zahlreiche Möglichkeiten, wie und mit welchen Mitteln man eine Anbindung an ein anderes System realisieren kann. Betrachtet man die verschiedenen Standards, die für eine Integrationslösung nutzbar gemacht werden können, so ist es sinnvoll, diese nach den Einsatztypen zu kategorisieren. Berlecon Research klassifiziert diese nach Geschäftsdokumenten (openTRANS, cXML, xCBL, ebXML, RosettaNet, Biz-Talk, ...), dem Format der auszutauschenden Daten (XML, EDIFACT, SWIFT, Idoc, CSV...), und Klassifikationsschemata (EANCOM, eClass, UN/SPSC, ...) [Quantz & Wichmann, 2003, S. 34].

| | | Anforderungen | Fragestellungen | Standards |
|------------------------|-----------------------|---------------|---|---|
| Transaktionsmanagement | Geschäftsprozesse | | Wie ist das Zusammenspiel zwischen Dokumenten und Prozessen geregelt? Integrität der Dokumente? | ebXML, RosettaNET, Odette, OBL, ... |
| | Datenübermittlung | | Transportmedium Sicherheit Transport-Nachrichtenprotokolle | X.25, OFTP, TCP/IP, ... SSL, ... HTTP, FTP, SMTP, SOAP, ... |
| | Datenaustauschformate | | Welche Inhalte und welche Form müssen/können Geschäftsdokumente aufweisen? | cXML, xCBL, openTRANS, iDOC, EDIFACT, ANSI X.12, ... |
| | Produktidentifikation | | Ist eine Standard-Identifikation der Produkte erforderlich? | EAN, UPC, ... |
| | Transaktionsdaten | | Was verlangen die Standards, was die Kunden? Wo liegen die Daten, in welchen Formaten? | Keine Standards regulative Anforderungen |
| | | ERP | div. DB | |

Abbildung 3.15 Ein Klassifikationsschema von elektronischen Standards nach einer Studie von Berlecon Research, vgl. [Quantz und Wichmann, 2003]

Im Folgenden sollen einige der dort aufgeführten Standards (vgl. Abbildung 3.15) im Rahmen eines Electronic Data Interchange kurz vorgestellt werden.

3.3.2.1 Electronic Data Interchange - EDI

Im Hinblick auf B2B Integration ist die Kommunikation zwischen zwei oder mehreren Systemen untereinander Grundlage für eine erfolgreiche Zusammenarbeit. Heutzutage ist das in den 60er Jahren entwickelte EDI (Electronic Data Interchange) immer noch ein fester Bestandteil, um Daten einem bestimmten Format entsprechend zur maschinellen Verarbeitung zu versenden [Priesnitz und Teille, 2003, S. 18]. Dabei ist es entscheidend, dass der Austausch der geschäftlichen Daten semantisch und syntaktisch korrekt stattfindet. Sender und Empfänger müssen sich über den Aufbau (Syntax) und den Inhalt (Semantik) der Dokumente vor der Kommunikation im Klaren sein, da sonst eine maschinelle Verarbeitung der Daten unmöglich wäre [Priesnitz und Teille, 2003, S. 20]. Dies wird umso schwieriger, je mehr Teilnehmer in einem Netz Daten untereinander austauschen, denn jeder Kommunikationspartner hat evtl. andere Vorstellungen von der Art und Weise, wie und welche Daten in einem Dokument angeordnet werden sollen. Folglich müssen vorher Vereinbarungen auf drei Ebenen getroffen werden, um eine Kommunikation überhaupt zu ermöglichen [Huemer, 2001, S. 16]. Abbildung 3.16 visualisiert die Ebenen:

Auf der Ebene des *Geschäftskontextes* muss der Informationsfluss mit allen möglichen Szenarien und Daten, die eine Geschäftstransaktion betreffen, definiert werden. Hier wird der inhaltliche Grundstein einer EDI Transaktion gelegt (Semantik).

Die *Transfersyntax* auf der nächsten Ebene strukturiert diese Daten und ermöglicht ihre Erkennung innerhalb des Nachrichtendesigns, das basierend auf der Syntax ein Geschäftsdokument modelliert.

Die dritte und letzte Ebene schließlich definiert das *Protokoll*, mit dem die Daten an den Partner übertragen werden.

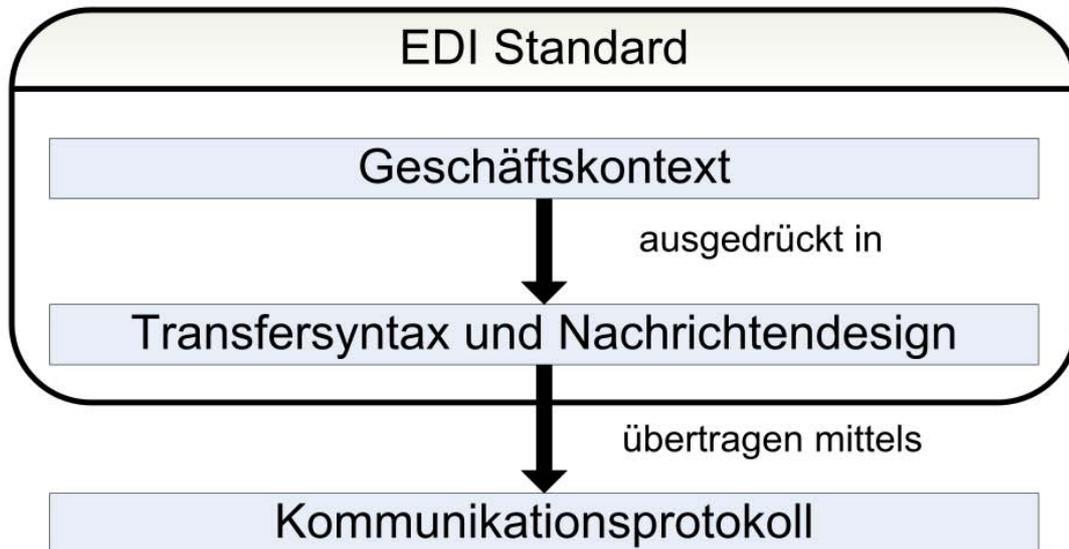


Abbildung 3.16 EDI-Standards im Rahmen der drei Ebenen

Seit der Einführung von EDI vor etwa 50 Jahren haben sich zahlreiche unterschiedliche EDI-Standards herausgebildet, die je nach Branche bzw. Nationalität stark differieren [Steen, 2001, S. 14]. In der Automobilindustrie ist z.B. ODETTE/VDA ein weit verbreiteter Standard, SWIFT im Bankensektor und SEDAS im Bereich Handel in Deutschland. In den USA ist ANSI X.12 als branchenneutrales Regelwerk im Einsatz [Niggel, 1994, S. 50].

Konverter

Wollen nun Unternehmen mit unterschiedlichen Standards geschäftliche Dokumente miteinander austauschen, so benötigen sie meist eine Konvertierungsmöglichkeit der auf maschinelle Verarbeitung ausgerichteten Nachrichten [Niggel, 1994, S. 17]. In heutigen ERP-Systemen sind entsprechende Konverter schon manchmal im Standardumfang anzutreffen. Anwender, die keinen Konverter besitzen, sich nicht leisten können oder deren EDI Nachrichtenaufkommen zu gering ist, haben eventuell auch die Möglichkeit über EDI-Interfaces, die im Netz von ihrem Kommunikationspartner bereitgestellt werden, solche Nachrichten zu verschicken.

Zudem gibt es noch die Möglichkeit, über B2B-Netzwerke wie z.B. Crossgate oder Seeburger einen kostenpflichtigen Anschluss zu einem Kommunikationspartner zu erlangen. Die zweite Möglichkeit einer funktionierenden Kommunikation sind Standards, auf die sich die Partner einigen. Branchenabhängige oder -unabhängige, nationale oder internationale Standards, je größer die Menge an potentiellen Teilnehmern einer Kommunikation wird, desto schwieriger wird es, eine gemeinsame Basis zu finden, mit der alle Beteiligten zufrieden sein können. Im Folgenden werden zwei wichtige Standards vorgestellt, mit denen man EDI verwirklichen kann, EDIFACT und XML.

3.3.2.2 EDIFACT (Datenaustauschformat)

Erst seit 1987, als es viele verschiedene EDI-Standards bereits gab, wurde ein internationaler Standard, EDIFACT (EDI For Administration, Commerce and Transport), verabschiedet, der von der UN/CEFACT gepflegt wird. In diesem Regelwerk werden so genannte Datenelemente (die kleinste unteilbare Dateneinheit mit einer exakten Bedeutung, z.B. Datum, Kundennummer etc.) zu Segmenten zusammengefasst, in denen die Reihenfolge der Datenelemente fest definiert ist und die den Syntaxregeln entsprechend mit anderen möglichen Segmenten eine zu übermittelnde Nachricht bilden [Niggel, 1994, S. 38]. Beispielhafte Nachrichten finden sich auf EDIFactory: <http://www.edifactory.de/index.php?p=unedifact/example>.

Um nun die Kommunikation zwischen mehreren Unternehmen weiter zu standardisieren, wurden Nachrichtentypen erstellt. Die Vorschriften zur Verwendung von Datenelementen, Segmenten und Nachrichten werden in den so genannten „Directories“ aufgeführt [Niggel, 1994, S. 39].

Ein wichtiger Unterschied zu dem im folgenden Kapitel 3.3.2.3 beschriebenen XML ist die Entwicklung des Standards hauptsächlich auf einen textbasierten Austausch von Informationen hin. Multimediale Daten wie z.B. Bilder oder CAD-Daten sind schwer einzubauen [Huemer, 2001, S. 18]. Hinzu kommen die schlechte Lesbarkeit für Menschen und die traditionelle Verbreitung der Nachrichten über kostenpflichtige VANs.

3.3.2.3 XML (Basisformat)

Eine andere Möglichkeit, Geschäftsdokumente zu beschreiben, ist auf der Basis von XML (eXtensible Markup Language). Auch hier kann man den Dienst eines Mediärs wie z.B. eines Value Added Networks (VAN) wahrnehmen oder die direkte Kommunikation über das Internet wählen. Die Nachteile von früheren EDI-Standards (hohe Kosten bei der Implementierung, mangelnde Flexibilität bei sich ändernden Bedingungen und strukturell bedingt geringe Freiheitsgrade [Priesnitz und Teille, 2003, S. 18]) scheinen durch XML umgangen werden zu können.

Vielmehr ist XML dazu geeignet, eine verbesserte Umsetzung zahlreicher Anwendungen zu erreichen. Beispiele sind das Dokumentenmanagement, eine effizientere Gestaltung web-basierter Dienste oder die Automatisierung zwischenbetrieblicher Kommunikation [Turowski & Fellner, 2001, S. V]. Der syntaktische Aufbau wird hierbei durch die XSDs (XML-Schema-Definitionen) semantisch angereichert und eignet sich so zur maschinellen Verarbeitung [Mertens, 2003, S. 258]. Sie bieten die Möglichkeit zur effektiven Festlegung einer Element- und Attributstruktur eines XML-Dokuments [Mertens, 2003, S. 290]. So können Struktur und Inhalt der Dokumente getrennt voneinander betrachtet werden. Durch die Nähe zu HTML ist XML zum einen gut lesbar für Menschen, zum anderen gut darstellbar in Browserfenstern. Auch hier haben sich Organisationen zusammengetan, um EDI-Standards mit XML als Grundlage zu bilden. Bekannte Beispiele sind:

- RosettaNet, eine Sammlung von unternehmensübergreifenden Geschäftsprozessen, vgl. Kapitel 3.2.1.7. (vgl. <http://www.rosettanet.org/cms/sites/RosettaNet/>)
- xCBL (XML Common Business Library), eine Sammlung von Business Dokumenten und ihren Komponenten [Schubert, 2002, S. 16] (vgl. <http://www.xcbl.org/>)

- BMEcat (Bundesverband Materialwirtschaft, Einkauf und Logistik), ein Standard für elektronische Produktkataloge [Schubert, 2002, S. 15] (vgl. <http://www.bmecat.org/deutsch/index.asp>)

Ein weiterer Standard, der auf XML beruht und Geschäftsprozesse beschreibt, soll im Folgenden näher vorgestellt werden: ebXML (electronic business using XML).

ebXML (Geschäftsprozess-Format)

1999 wurde von UN/CEFACT und OASIS die ebXML (electronic business using eXtensible Markup Language) Initiative ins Leben gerufen. Ziel war es, ein offenes und technisches Framework zu schaffen, auf dessen Basis Standardkomitees Datenaustauschformate bzw. Geschäftsprozessmodelle entwickeln, die wiederum dann Softwareanbieter in ihrer Software für Anwender umsetzen [Huemer, 2001, S. 27]. Die so erstellten Modelle sollten in einem globalen Repository abgelegt und für jeden verfügbar gemacht werden. Der aktuelle Standard ist unter <http://www.ebxml.org/> zu finden.

Wie auch bei EDIFACT ist hier nur die Dokumentstruktur ausschlaggebend und nicht die Übertragungsweise. Die gute Lesbarkeit und die Vorteile bei der Darstellung von XML sollten allerdings nicht darüber hinweg täuschen, dass ebenso wie bei EDIFACT ohne die Mitarbeit der zahlreichen Unternehmen, die Kommunikationsstandards einsetzen, einheitliche Dokumententypen kaum realisierbar sind. Es gibt auch hier die Möglichkeit, Spezialisierungen aus diesem Regelwerk zu erzeugen, die eine einheitliche Kommunikationsgrundlage in Frage stellen können.

3.3.2.4 Web Services

Web Services sind eine weitere Möglichkeit, einen elektronischen Datenaustausch zu realisieren. Sie finden immer häufiger Anwendung in betrieblichen Anwendungssystemen und bieten eine weitere Alternative zu den bekannten Frameworks Corba, DCOM oder RMI. Ein großer Unterschied zu diesen bekannten Middlewares besteht in der Nutzung der allgemeinen Internet Standards wie HTTP und XML zur anwendungsinternen Kommunikation [Kuschke & Wölfel, 2002, S. 100]. Dadurch können sie sehr einfach in die existierende Infrastruktur des Netzes integriert und innerhalb dieser genutzt werden.

Definition

Eine Definition aus Business-Sicht ist zum Beispiel folgende:

„Web Services ist eine mächtige Integrationsarchitektur, welche eine dynamische Zusammenarbeit von Applikationen über Netzwerke hinweg unter Nutzung von offenen Internet-Technologien erlaubt.“ [Kuschke & Wölfel, 2002, S. 3].

Strnadl sieht Web Services als wichtigen Bestandteil eines service-orientierten Anwendungssystems [Strnadl, 2006, S. 345]. Chen geht noch weiter und beschreibt Web Services aus drei Blickwinkeln [Chen, 2005, S. 267f]:

- Technologische Sicht: Web Services sind unabhängig von Programmiersprachen, Plattformen oder Objekt-Modellen.
- Architektur-Sicht: Web Services ermöglichen service-orientierte und komponentenbasierte Anwendungssysteme.

- Planungssicht: Web Services können zur Entwicklung dynamischer e-Business Prozesse beitragen.

Die Erwartung an Web Services ist hoch. Doch nicht alle der befragten Unternehmen teilen diese Ansicht, wie die Auswertung zeigen wird. Trotzdem werden sie in den gängigen Applikationsservern, die auf J2EE und .NET beruhen, inzwischen auf breiter Basis unterstützt und genutzt.

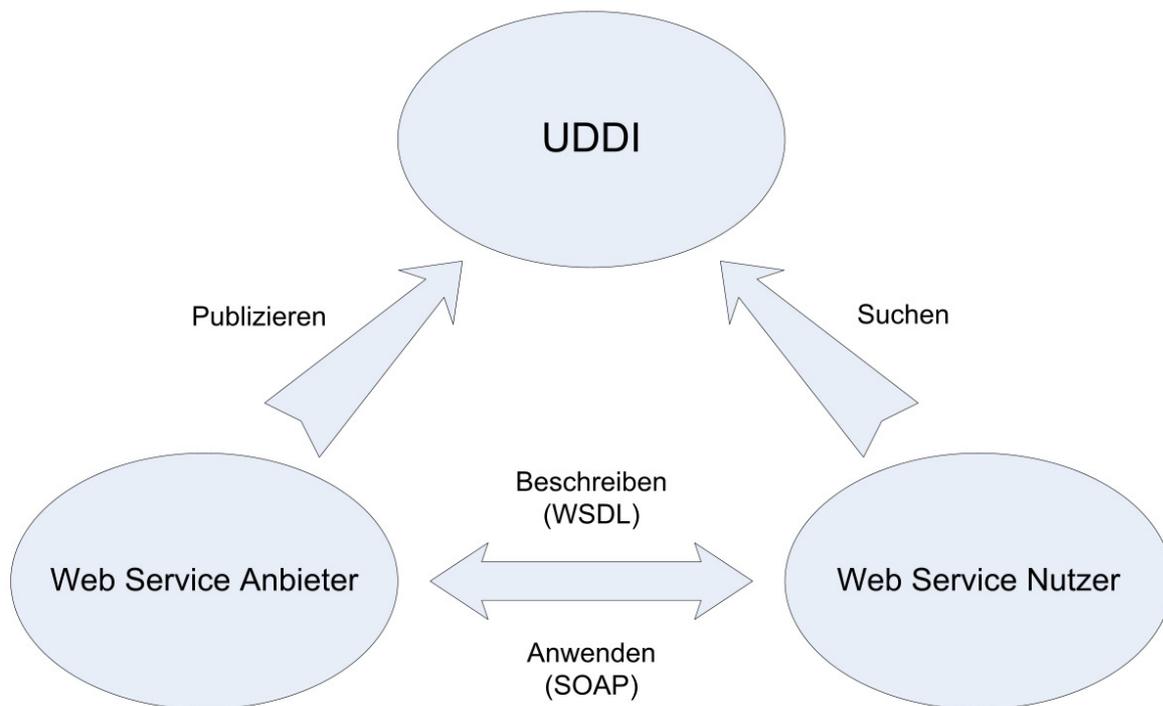


Abbildung 3.17 Web Services im Überblick

Technologische Basis

Technisch gesehen werden in dieser Integrationsarchitektur Softwarekomponenten in einer XML-Notation beschrieben (WSDL, Web Service Description Language), die wiederum mittels XML Nachrichten über die Internet-Infrastruktur austauschen (SOAP). Zu finden sind diese Komponenten in speziellen Verzeichnissen (UDDI, Universal Description, Discovery and Integration) [Kuschke & Wölfel, 2002, S. 3].

Wie in Abbildung 3.17 dargestellt, beschreiben Web Service Anbieter ihre angebotenen Dienste mit Hilfe von WSDL. Dort wird festgeschrieben, wie das Interface des Dienstes aussieht und welche Datentypen ausgetauscht werden können, mit welcher Methode diese Daten ausgetauscht werden und wo der Service zu finden ist.

Das UDDI Verzeichnis erlaubt es anschließend den Anbietern, ihre Dienste zu veröffentlichen und den Anwendern, danach suchen zu können. Dort findet man eine allgemeine Beschreibung des Anbieters, eine Auflistung der von ihm angebotenen Dienste und eine detaillierte Beschreibung, wo dieser Dienst zu finden und wie er anzusprechen ist [Eberhart & Fischer, 2003, S. 304].

Ist der Nutzer bereit, den Dienst zu nutzen, so findet die Kommunikation zwischen den beiden Parteien statt. Dazu wird meistens SOAP als Schnittstellenbeschreibung genutzt, es können auch HTTP GET/POST oder MIME verwendet werden. SOAP beschreibt, wie XML Dokumente von einem Kommunikationspartner zum anderen transportiert werden. Dabei können nicht nur einfache Dokumente ausgetauscht werden, sondern auch systemübergreifende Methodenaufrufe mit übergebenen Parametern verwirklicht werden [Eberhart & Fischer, 2003, S. 180f].

Risiken und Möglichkeiten

Das Ziel von Web Services ist die dynamische Einbindung von Diensten zur Laufzeit einer Anwendung. Um dies zu gewährleisten, müssen weitere Standardisierungsbemühungen voran getrieben werden. Noch gibt es keinen globalen Verzeichnisdienst, bei dem man sich auf die Verlässlichkeit und Vollständigkeit der Informationen verlassen kann. Signaturen und Verschlüsselungsmechanismen müssen sich erst noch durchsetzen, sind aber seit 2002 bereits durch die W3 (World Wide Web Consortium) standardisiert [Eberhart & Fischer, 2003, S. 268].

Ebenso ist Transaktionssicherheit ein immer noch kritischer Punkt [Kuschke & Wölfel, 2002, S. 100]. Allerdings schreiten die Standardisierungsbemühungen immer weiter voran. Einen aktuellen Überblick (Stand 2007) findet man in folgender Quelle: [innoQ Deutschland GmbH & AG, 2007].

Trotzdem eignen sie sich sehr gut für die unkomplizierte Integration verschiedener Systeme, da sie auf den gängigen Internet-Standards aufsetzen und den sehr weit verbreiteten XML-Standard als Beschreibungssprache nutzen. Ein tieferes Verständnis der zu integrierenden Systeme ist nicht mehr erforderlich, da mittels Web Services eine neue Schicht auf die Systeme gelegt wird. Der Integrationsmechanismus besteht nicht mehr aus der manuellen Auscodierung einer Punkt-zu-Punkt Integration, sondern aus einer Orchestration der Services mittels BPM (vgl. Kapitel 3.2.1.5) [Miller, 2003, S. 64].

3.3.2.5 Client-Zugriffe

Der Client-Zugriff auf eine Anwendung, die auf einem Server verwaltet wird, z.B. bei einem ASP-Anbieter, kann auf zwei Arten erfolgen [Salmen-Fuchs, 2007, S. 26]:

Beim so genannten ClientBased ASP wird ein kleines Programm auf den Client geladen, das sich über Plug-Ins zum Server verbinden und so die Anwendungen starten kann.

Beim ServerBased ASP verbleibt die gesamte Applikationslogik auf dem Server. Man erhält über Java-Applets ein Sichtfenster auf den Server, in dem alle Vorgänge dargestellt werden. Dieses Verfahren ermöglicht den Einsatz der Thin-Clients, die sich durch eine einheitliche Darstellung und eine geringe Bandbreite bei der Datenübermittlung auszeichnen. Das ServerBased ASP kann nun entweder mit den bereits vorhandenen Ressourcen (Browser, Java, etc.) umgesetzt werden, oder mit Hilfe proprietärer Protokolle wie z.B. das ICA-Protokoll (Independent Software Architecture) des Unternehmens Citrix [Salmen-Fuchs, 2007, S. 28]. Das Protokoll erlaubt einen schmalbandigen Zugriff auf so genannte Citrix-Metaframe Server, die Anwendungen für Nutzer zur Verfügung stellen.

Zielgruppe von Terminallösungen sind bevorzugt Anwender, die auf Basis einer schmalbandigen Leitung auf eine nicht lokal stehende Anwendung zugreifen wollen [Joos, 2004, S. 3]. Zusätzlich bietet das Unternehmen weitere Dienstleistungen wie Datensicherheit,

Verschlüsselung oder auch für Unternehmen eine Prozessoptimierung an (vgl. <http://www.citrix.de/produkte/loesungen/unternehmensgroesse/kmu/>).

3.3.2.6 Asynchronous Javascript and XML - Ajax

Klassischer Weise werden Webseiten und deren Inhalte bei einer Veränderung beim Client, z.B. einer Neuansteuerung einer Unterseite oder einer Veränderung innerhalb der Seite, neu vom Server zum Client geschickt [Carl, 2006, S. 2]. Ajax ermöglicht es nun, nur diejenigen Daten senden zu müssen, die wirklich neu sind.

Dadurch ergibt sich ein enormer Geschwindigkeitsvorteil gegenüber der klassischen Variante [Carl, 2006, S. 4]. Ajax ist weniger eine Technologie, sondern vielmehr eine Technik. Funktionen, die bislang in der Form nur auf so genannten Thick-Clients möglich waren, können nun ganz auf dem Server verbleiben. Google und Yahoo waren Vorreiter in der Ajaxnutzung [Asleson & Schutta, 2006, S. 20f].

3.4 Betreibermodelle

Die Art und Weise, wie das betriebliche Anwendungssystem eingesetzt wird, entscheidet über Verantwortlichkeit, Funktionsmächtigkeit und Kosten. Für einen potentiellen Nutzer von betrieblichen Anwendungssystemen gibt es vier Möglichkeiten, seinen Bedarf an informationstechnischer Unterstützung zu decken [Keller, 2006]:

- 1. Build: Die Lösung individuell für das Unternehmen entweder selbst anfertigen oder von einem IT-Haus anfertigen lassen
- 2. Buy: Eine vorgefertigte Lösung von einem Anbieter kaufen
- 3. Rent: Eine Lösung von einem Anbieter mieten (Bsp.: ASP, SaaS (siehe weiter unten))
- 4. Outsource: Einem Anbieter die Organisation und Pflege der eigenen Geschäftsprozesse überlassen

Meist ist nicht nur eine Möglichkeit für Unternehmen interessant, es kann z.B. eine ERP-Lösung intern betrieben werden, die Finanzbuchhaltung wird jedoch als SaaS-Angebot genutzt.

Im Folgenden werden die drei meist genutzten Möglichkeiten vorgestellt, wie ein fertiges Anwendungssystem aussehen kann.

3.4.1 In-House

Bis Mitte/Ende der neunziger Jahre war es üblich, dass das Anwendungssystem inklusive der Hardware beim Kunden selbst vor Ort betrieben wurde. Die Software war dabei auf einem oder mehreren Rechnern installiert, die beim Kunden standen, die Wartung musste ebenfalls vor Ort erfolgen. Kosten entstanden dem Kunden für die Hardware, die Lizenz für die Nutzung der Business Software und durch Service-Verträge, die eine Wartung (Updates, Patches, Releases etc.) regelten. Das klassische Lizenzgeschäft von damals ist auch heute noch weit verbreitet und nach Meinung vieler Anbieter nicht obsolet.

3.4.2 Application Service Providing - ASP

Salmen-Fuchs liefert eine Definition für ASP [Salmen-Fuchs, 2007, S. 11]:

Unter ASP versteht man, „[...] dass ein Unternehmen (oder Endkunde, das auch eine Privatperson sein kann) die benötigte Software nicht mehr in einer Lizenz erwirbt, sondern die benötigten Applikationen (Software) für eine gewisse Zeit (und gegen eine Gebühr) über das Internet oder andere Netze gewissermaßen anmietet.“

Application Service Provider, kurz ASP, traten zum Ende der 90er Jahre zum ersten Mal in Erscheinung. Durch angestrebte Outsourcing Projekte und dem Aufkommen des Electronic Commerce durch die Verbreitung des Internets war dieses neue Betreibermodell eine notwendige Entwicklung [Daylami et al., 2005, S. 1].

ASP Anbieter boten Business Software über das Medium Internet jedem an, der bereit war, einen monatlichen Beitrag für die Nutzung der Software zu leisten. Die vor allem auf kleine und mittlere Unternehmen ausgerichteten Lösungen boten den potentiellen Nutzern von Business Software nun eine weitere Möglichkeit, wie sie ihre IT-Landschaft umsetzen konnten. Neben dem klassischen „make“ (selbst erstellen) und „buy“ (einkaufen) gab es nun die Option „rent“ (mieten) [Fortune & Aldrich, 2003, S. 104].

Der grundsätzliche Aufbau bestand dabei aus einer Datenbank, einem Webserver, der Daten aus der Datenbank an den Client weiterleitete und dem Client selbst, der als Fat-Client Applikationen verwaltete. [Furht et al., 2000] schlug im Jahr 2000 vor, diese 3-Tier Architektur um einen Applikationsserver zu erweitern, der die Anwendungslogik vollständig übernehmen sollte. Die Clients sollten auf Basis von Citrix-Technologie (vgl. Kapitel 3.3.2.5) mit den Servern kommunizieren. Die Auswahl an möglichen Anwendungen reichte dabei von Finanzbuchhaltung über Human Resources bis zu CRM. Für jeden Kunden wurde entsprechend eine Lösung verwaltet, die vom Umfang und Aufbau her der In-House Lösung entsprach [Staff, 2007]. Das ASP Modell wurde als „The next big thing“ gehandelt und dementsprechend waren die Erwartungen groß. Ende 2001 wurde klar, dass ASP nicht das leisten konnte, was von den Analysten vorhergesagt wurde. Es fand ein Wandel in der Struktur der ASPs statt, denn zum einen verschob sich der Anteil der vielen selbständigen ASP Anbieter hin zu Tochterfirmen bzw. Abteilungen von größeren IT Firmen, zum anderen wurde der Anspruch, die komplette IT Landschaft eines Unternehmens abbilden zu wollen, aufgegeben. Der Fokus lag nun eher auf einer Integration bzw. Unterstützungsleistung zu bestehenden Systemen [Fortune & Aldrich, 2003, S. 106].

Die Gründe für das damalige Scheitern werden der oft noch zu geringen Bandbreite zugeschrieben, die für das Datenvolumen nicht ausgelegt war, das versendet werden musste. Ebenso wird es den ASP Anbietern zur Last gelegt, zwar die Hosting Lösung angeboten, allerdings nicht das benötigte Know-how zur Wartung mitgebracht zu haben (z.B. Internet Service Provider).

Aktuell ist SaaS (Software as a Service) als neues Schlagwort in Erscheinung getreten, dem sich sogar die Systems 2007 mit einem eigenen Forum gewidmet hat (vgl. <http://www.saas-pavilion.de/>).

3.4.3 Software as a Service - SaaS

Software as a Service beschreibt ein dem ASP sehr ähnliches Modell, dass sich nur auf den ersten Blick in kleineren Aspekten unterscheidet. Auch hier wird die Software als Mietlösung zur Verfügung gestellt. Wartung und Installation auf der technischen Seite sowie Lizenzkosten auf der wirtschaftlichen Seite entfallen dem Anwender.

Im Vergleich zum früheren ASP sind SaaS-Lösungen jedoch in der Regel mandantenfähig (multi-tenancy), das heißt, eine Instanz der Software bedient mehrere Kunden gleichzeitig [Henschen, 2006, S. 1]. Somit entfällt die Pflege von mehreren Systemen, die im ASP-Modell für jeden Kunden je einmal vorhanden waren [Staff, 2007]. Von der Konzeption her unterscheiden sich die im ASP-Betrieb laufenden Systeme nicht von der In-House Lösung, die bei einem Kunden vor Ort betrieben wird. SaaS-Software ist im Gegensatz dazu auf eine Nutzung von vielen Nutzern im Internet ausgelegt ("One-to-many", [Lixenfeld, 2008]). Beide Betreibermodelle profitieren zudem von dem deutlich besser ausgebauten Verbindungsnetz im Vergleich zu Ende der 90er Jahre.

Laut einer Studie von Gartner werden die Einnahmen mit SaaS Produkten von 5,1 Mrd. US-Dollar (2007) auf 11,9 Mrd. US-Dollar Prozent (2011) anwachsen [Gartner Press, 2007].

Die Zahlen beziehen sich dabei auf CRM, ERP, SCM und Content Management Systeme. Sharon Mertz, Forschungsdirektorin von Gartner, beschreibt die Zukunft von SaaS wie folgt: „Die Akzeptanz von SaaS ist dort am höchsten, wo standardisierte Geschäftsprozesse oder große, räumlich verteilte Arbeitsgruppen vorhanden sind“.

3.5 Markt

Der Markt für betriebliche Anwendungssysteme, im speziellen ERP-Systeme, gibt zuletzt wertvolle Hinweise auf wirtschaftliche Faktoren, die Entwicklungen in und außerhalb der Systeme beeinflussen können. Markttrends und die breite Unterstützung der Industrie entscheiden oft darüber, ob Standards eine große Verbreitung finden oder sinnvolle Ideen verworfen werden. Die in letzter Zeit erbrachten Anstrengungen großer Softwareanbieter wie SAP in den Markt für kleine und mittlere Unternehmen einzudringen, gibt Anlass, einen genaueren Blick auf deren Bedeutung zu werfen.

3.5.1 KMU

Mit KMU bezeichnet man kleine und mittlere Unternehmen. Sie stellen die oder zumindest einen Teil der Zielgruppe der in dieser Arbeit untersuchten ERP-Systeme dar. Die Kriterien, nach denen man einen Betrieb als „klein“ oder „mittel“-groß einstuft, können eindimensional oder vieldimensional sein [Mugler, 2005, S. 23]. Eindimensional wäre z.B. nur die Anzahl der Mitarbeiter als Kriterium zu nehmen, da z.B. ein Unternehmen mit sehr wenigen Mitarbeitern durchaus extrem hohe Umsätze produzieren kann. Mehrere Dimensionen entzerren solche simplen Sichtweisen, erhöhen aber die Anforderungen an die Messgrößen.

3.5.1.1 Klassifikation - öHGB

Nach § 221 öHGB wird im Rahmen der Regelung der Rechnungslegung, Anhangerstellung, Lageberichterstattung, Pflichtprüfung und Pflichtveröffentlichung eine Einteilung in kleine, mittelgroße und große Kapitalgesellschaften vorgenommen [Mugler, 2005, S. 25]:

Kleine Gesellschaften überschreiten zwei der folgenden drei Merkmale nicht:

- 3,125 Mio. Euro Bilanzsumme
- 6,250 Mio. Euro Umsatzerlöse pro Jahr
- im Jahresdurchschnitt 50 Arbeitnehmer

Mittelgroße Gesellschaften überschreiten zwei der drei Kriterien für kleine Gesellschaften und überschreiten nicht zwei der drei folgenden:

- 12,5 Mio. Euro Bilanzsumme
- 25 Mio. Euro Umsatzerlöse pro Jahr
- im Jahresdurchschnitt 250 Arbeitnehmer

| Unternehmenskategorie | Zahl der Mitarbeiter | Umsatz | | Bilanzsumme |
|-----------------------|----------------------|----------------------------|------|----------------------------|
| mittelgroß | < 250 | kleiner/gleich 50 Mio Euro | oder | kleiner/gleich 43 Mio Euro |
| klein | < 50 | kleiner/gleich 10 Mio Euro | oder | kleiner/gleich 10 Mio Euro |
| mikro | < 10 | kleiner/gleich 2 Mio Euro | oder | kleiner/gleich 2 Mio Euro |

Abbildung 3.18 Klassifikation von kleinen und mittelgroßen Betrieben der Europäischen Union, Stand 2003

3.5.1.2 Klassifikation - EU

Die EU hat im Mai 2003 im Rahmen der Förderung kleiner und mittelgroßer Betriebe eine allgemeine Kategorisierung vorgenommen, um solche Unternehmen identifizieren zu können [Mugler, 2005, S. 26] (vgl. Abbildung 3.18). KMU stellen laut den Berechnungen für die Generaldirektion „Unternehmen und Industrie“ der Europäischen Kommission (Stand 2003) 99,8 Prozent aller Unternehmen in der EU (inklusive Schweiz) [Mugler, 2005, S. 34]. Sie sind eine treibende wirtschaftliche und forschungsorientierte Kraft. Sie stellen rund zwei Drittel aller Arbeitsplätze und erwirtschaften mehr als 50 Prozent der gesamten Wertschöpfung. Große Anbieter wie SAP oder Microsoft haben das Potenzial erkannt, dass in dem Business Software Markt der KMU liegt und unternehmen Anstrengungen, neben ihrem bisherigen Standbein der Konzernlösungen Anbietern von KMU-Lösungen Konkurrenz zu machen.

3.5.2 Branchen

ERP-Systeme unterstützen als integriertes Informationssystem die Wertschöpfung eines Unternehmens. Allerdings ist nicht jede Branche für ein System geeignet bzw. ein System nicht für alle Branchen.

Auch unter den befragten Unternehmen gab es Spezialisierungen z.B. auf Produzierende Gewerbe oder Handelsunternehmen. Um die Einteilung der Systeme den Ergebnissen der Befragung entsprechend übersichtlich zu gestalten, wurde anlehnend an die Klassifikation der Wirtschaftszweige (WZ 2003) (vgl. [Deutschland, 2003]) eine deutlich einfachere Gliederung in der Auswertung benutzt:

- Dienstleistende Unternehmen
- Produzierende Unternehmen
- Handelnde Unternehmen

3.5.3 Anbieter

Kleine und mittlere Betriebe profitieren ebenso von betrieblichen Anwendungssystemen wie große Konzerne. Allerdings benötigen sie nicht immer den ganzen Umfang, den ein ERP-System unter Umständen anzubieten hat [David et al., 2003, S. 65] bzw. besitzen zwar eine ähnliche Prozesskomplexität wie Konzerne, haben jedoch deutlich weniger Mitarbeiter für deren Beherrschung [Gronau, 2004, S. 14]. Einige schrecken deshalb vor der großen Investition zurück, die ein klassischer In-House Betrieb mit Lizenzmodell mit sich bringt. Trotzdem bleibt die Notwendigkeit für viele unbestritten, dass eine informationstechnische Unterstützung ihrem Unternehmen zugute kommen würde. Der Markt der ERP-Anbieter befindet sich laut einigen Befragten etwa seit dem Jahr 2000 in einer Umgestaltung hinsichtlich technologischer und konzeptioneller Aspekte der Systemgestaltung.

Schlagworte wie SOA, Web Services oder seit neuem SaaS sorgen für Bewegung auf dem Anbietermarkt. Branchengrößen wie SAP versuchen, mit rein webbasierten Lösungen in das Kundensegment der kleinen und mittelgroßen Betriebe einzudringen und so gegenüber den kleineren Anbietern Marktanteile zu gewinnen. Ob diese neue Strategie Erfolg hat oder nur ein Versuch bleibt, wird die Zukunft zeigen.

3.5.3.1 Open Source Anbieter

Neben den proprietären Anbietern auf dem Markt gibt es noch die so genannten Open Source ERP-Systeme. Der wesentliche Unterschied zwischen beiden Anbietergruppen ist, dass Open Source Anbieter den Sourcecode ihrer Systeme öffentlich machen, d.h. freien Zugang zu dem eigentlich geschriebenen Programm ermöglichen. Proprietäre Anbieter verlangen für die Nutzung ihrer Systeme in der Regel eine Lizenzgebühr.

Serrano und Sarriegi führen drei wichtige Unterschiede auf [Serrano & Sarriegi, 2006, S. 94]:

- 1. Erhöhte Anpassbarkeit: ERP-Systeme benötigen immer ein Customizing (vgl. Kapitel 3.1.6). Der Zugang zum Sourcecode kann diesen Schritt erleichtern.
- 2. Schwächere Bindung an den Hersteller: Man ist als Kunde an proprietäre Hersteller gebunden, die ihren Sourcecode nicht freigeben. Sollten sie oder ihre Partner nicht mehr verfügbar sein, so ist eine Weiterentwicklung und Wartung des Systems schwierig.
- 3. Verringerte Kosten: Lizenzgebühren betragen erfahrungsgemäß etwa 1/6 bis 1/3 der Einführungskosten [Serrano & Sarriegi, 2006, S. 94], wie auch in einem Interview versichert wurde (vgl. A.1). Diese Kosten würden wegfallen.

Die Komplexität der ERP-Systeme ist mit denen der proprietären Anbieter durchaus gleichzusetzen. Allein der Sourcecode führt nicht zu einem einsatzfähigen System, Applikationsserver und Datenbank müssen unter Umständen ebenfalls eingerichtet werden [Serrano & Sarriegi, 2006, S. 95]. Hinzu kommt, dass eine Einführung und Betrieb der Open Source Software nicht ohne eine fachgerechte Beratung auskommt [Serrano & Sarriegi, 2006, S. 96].

4 Auswertung

Im Folgenden werden die Ergebnisse der Interviews aus den vier Aspekten Architektur, Technologie, Betreibermodell und Markt aufgeführt und bewertet. Die Auswertungen in den einzelnen Fragebereichen gliedern sich derart, dass zu Beginn die Fragestellungen noch einmal thematisch kurz erläutert werden. Anschließend folgt eine dem Fragenbereich zugeordnete Beschreibung des Verlaufs der Befragung mit einer daran anknüpfenden aggregierten Zusammenfassung der gewonnenen Ergebnisse.

Die detaillierte Auswertung findet sich im Anschluss an die jeweiligen Zusammenfassungen.

4.1 Architektur

Die Architektur eines ERP-Systems beginnt mit der planerischen Gestaltungsmöglichkeit innerhalb dieses Systems (Planungsebene), womit sich der Anwender sein Unternehmen informationsstechnisch abbilden kann. Dies macht er unter den vorgegebenen konzeptionellen und architekturellen Voraussetzungen (Architektur), die er nur sehr begrenzt beeinflussen kann. Das Modulangebot und die Flexibilität in der Funktionsauswahl (ERP) bestimmt seine Funktionstiefe, die er zum Schluss in der Zusammenarbeit mit anderen Unternehmen (Integration) anwenden muss.

Die folgenden Kapitel des Architekturabschnittes beschreiben die Ergebnisse, die in allen vier Unterkapiteln Planungsebene, Architektur, ERP und Integration gesammelt wurden und bewerten sie.

4.1.1 Planungsebene

Mit dem Wandel der Anwendungssysteme kam auch das Umdenken weg von der Funktionssicht in den einzelnen Abteilungen hin zur Prozesssicht. Geschäftsprozesse und deren graphische Darstellungs- und betriebliche Einsatzmöglichkeiten werden ebenso begutachtet wie die Verwendung von Prozessstandards als mögliche „Best-Practice“ Vorlagen.

4.1.1.1 Verlauf der Befragung

Einigen Teilnehmern der Befragung war der Begriff „Referenzprozess“ nicht geläufig. Auf die Nachfrage der Interviewpartner, ob es sich um Geschäftsprozesse oder Workflows handelte, wurden Geschäftsprozesse angegeben. Auf die zweite Frage der automatisierten Codeerzeugung fügten manche Befragte zu ihren negativen Antworten noch die entschlossene Meinung hinzu, dass eine solche Möglichkeit aufgrund der hohen Komplexität gar nicht existieren könne.

4.1.1.2 Zusammenfassung der Ergebnisse

- Abbildung von Geschäftsprozessen: 38 Prozent, Workflows: 13 Prozent, Insgesamt: 51 Prozent.
- Automatisierte Codeerzeugung nutzen 25 Prozent (nach Anbieteraussage zu kompliziert und unrealistisch umzusetzen).

- Prozesseinsatz beim Kunden: 41 Prozent (meist nur Visualisierungscharakter).
- Verwendung von Eigenentwicklungen im Prozessmodell: 70 Prozent.
- Eine Anpassung der Geschäftsprozesse durch den Kunden bei 88 Prozent möglich (allerdings oft mit Einschränkungen/Vorwissen in der Entwicklung behaftet).

4.1.1.3 Auswertung

Graphische Umsetzung von Geschäftsprozessen

Es war von großem Interesse zu sehen, ob dem Nutzer seine Prozesse in graphischer Form verfügbar gemacht werden. In diesem Fall hätte er zumindest auf Modellebene ein anschauliches Bild seiner möglichen Aktivitäten zur Verfügung. Etwas mehr als die Hälfte der Befragten haben eine solche graphische Repräsentation (vgl. Abbildung 4.1) in ihrem System umgesetzt.

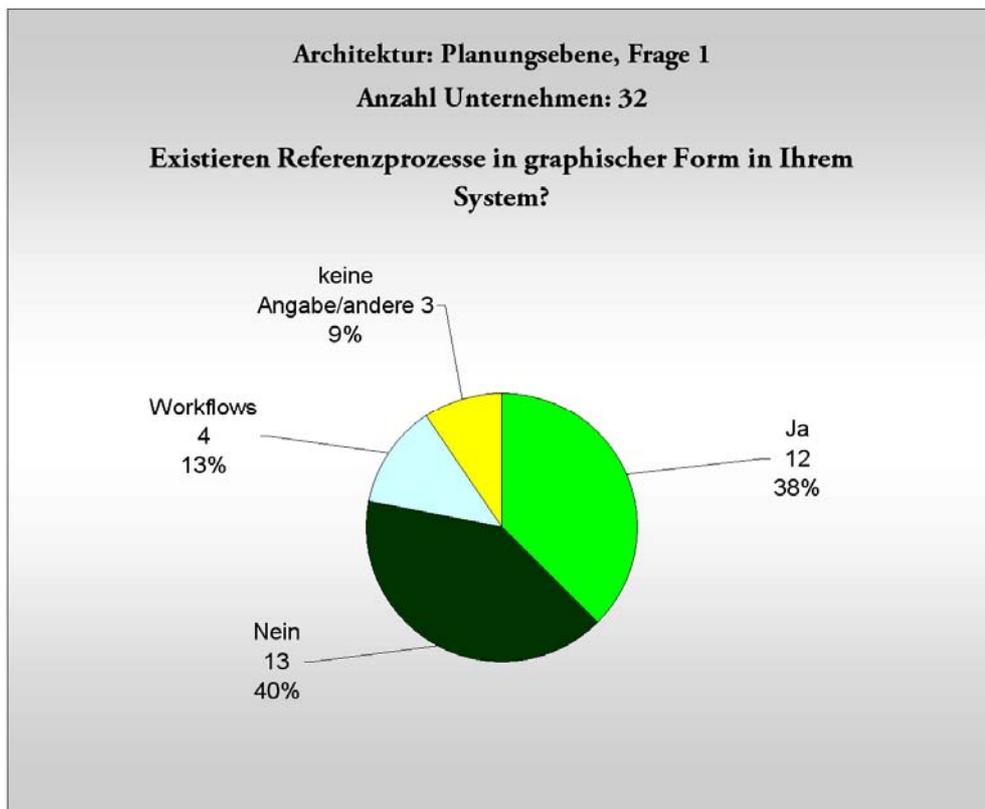


Abbildung 4.1 Graphische Umsetzung von Prozessen

Zwölf Unternehmen bilden Geschäftsprozesse ab, vier Anbieter haben eine visuelle Repräsentation ihrer Workflows umgesetzt und ein Softwarehaus, Alea, verwendet als graphische Planungsmöglichkeit Use-Cases. Dabei spielt eine Branchenzugehörigkeit oder Größe der Unternehmen keine Rolle. Egal ob Produktion oder Handel, ob 10000 Kunden oder nur 50, es lässt sich kein Muster erkennen, ob bestimmte Anbieter eher dafür prädestiniert

sind, eine solch planerische Oberfläche zu bieten. Die meisten nutzen dafür eine selbst entwickelte Modellierungsoberfläche. IFS hingegen bedient sich der UML (Unified Modelling Language), während Informing einen Fremdanbieter für seine Prozessmodellierung anbindet.

Auffällig ist, dass mehr als die Hälfte derjenigen Anbieter, die nach eigenen Angaben eine service-orientierte Architektur (SOA) besitzen bzw. SOA-fähig sind (vgl. Kapitel 4.1.2), eine graphische Prozessmodellierung unterstützen.

Nur fünf Unternehmen entsprechen nicht dieser Eigenschaft: Godyo und Informing bieten zwar eine Prozessmodellierung an, das Systemkonzept ist jedoch eine Eigenentwicklung, SoftM und Seat-1 bezeichnen sich als SOA-fähig, besitzen jedoch wie auch Microsoft (Axapta) als SOA Vertreter keine Prozessmodellierung. Die restlichen 13 Anbieter (40 Prozent) bieten keine graphische Prozessabbildung innerhalb ihres Systems.

Bei vielen der Befragten werden Prozesse als Planungs- und Systemgrundlage genutzt, sind jedoch nicht visuell abrufbar oder editierbar. Es gibt Stimmen, die gerade den Prozessverantwortlichen ein technisches Verständnis der Modellierungswerkzeuge absprechen:

„Geschäftsprozessexperten kommen wesentlich leichter mit einem linearen Text zurecht als mit der Modellierungssprache, in der die Prozesse niedergeschrieben werden sollen.“ (GAL1)

Automatisierte Codeerzeugung

Es gibt nur acht Unternehmen, die noch einen Schritt weiter gehen und eine automatisierte Umsetzung der Prozesse in Code verfolgen (vgl. Abbildung 4.2). Die Reichweite erstreckt sich dabei von Rahmenwerken, die noch ausprogrammiert werden müssen bis hin zu lauffähigen Workflow-Prozessen.

Ramco ist dabei am weitesten fortgeschritten. Hier werden etwa 80 Prozent des Rahmenwerks der Applikation bereits im Vorfeld automatisch generiert. IFS ermöglicht es, über die UML aus einer Prozessdarstellung vorläufigen Code zu erhalten. Lawson erstellt einige Services, während die GUS Group nur Workflow Steuerungsparameter generiert. Intraprend erzeugt HTML-Code, Oracle erhält aus der Prozessmodellierung einen lauffähigen Workflow-Prozess, Alea generiert aus der eigenen Use-Case Darstellung Code und SynERP kann Prozesse teilweise zur Codeerzeugung nutzen.

Ebenso wie bei der Prozessmodellierung sind es überwiegend service-orientierte Systeme, die eine automatisierte Codeerzeugung unterstützen. Im Besonderen sind es nur diejenigen Systeme, die nach eigenen Angaben nicht nur SOA-fähig sind, sondern intern nach der SOA konzipiert wurden. GuS und SynERP bilden eine Ausnahme mit ihren Systemen. GuS beinhaltet nach eigenen Angaben nur SOA-Ansätze und SynERP besitzt eine Eigenentwicklung als Konzeptgrundlage. IAS als SOA-zertifiziertes System bietet keine automatisierte Codeerzeugung an.

Mehr als die Hälfte der Anbieter (neun), die eine graphische Repräsentation der Geschäftsprozesse anbieten, setzt keine automatisierte Code-Erzeugung um. Insbesondere haben diejenigen Anbieter, die nur eine Workflow-Darstellung unterstützen, keine Unterstützung in diesem Bereich.

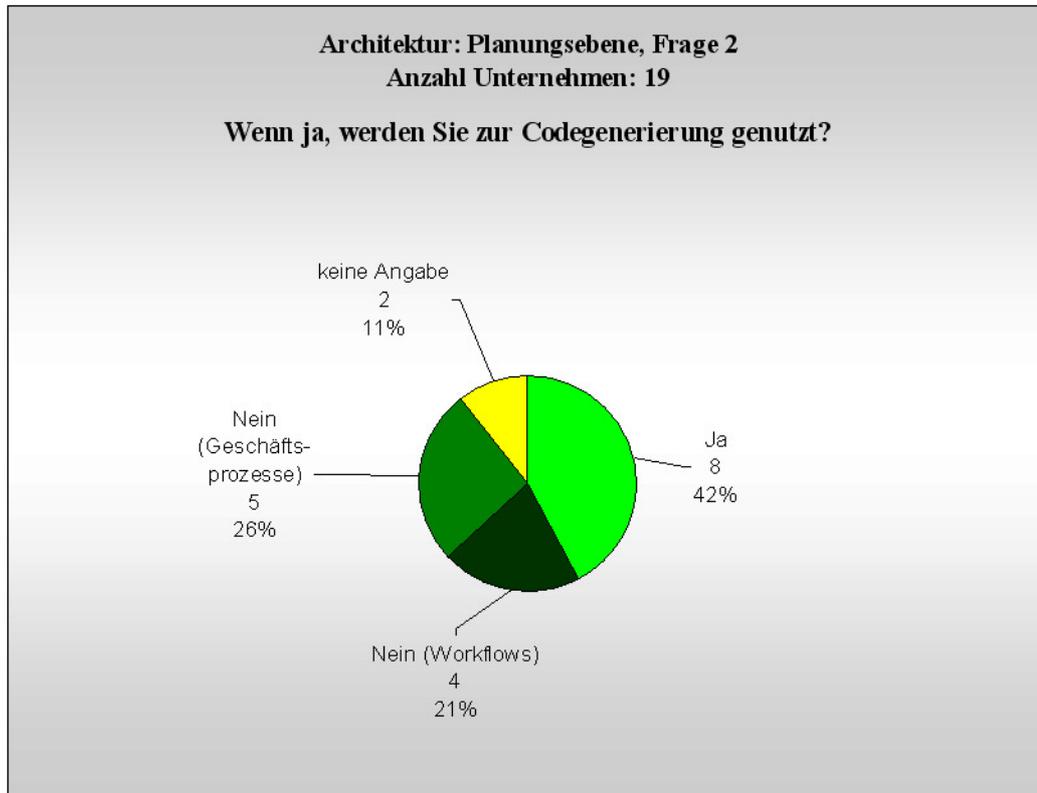


Abbildung 4.2 Automatisierte Umsetzung der Prozesse in Code

Insgesamt ist jedoch die Bereitschaft, hier mehr Innovationskraft zu zeigen, gering. Ein Anbieter gab technische Bedenken an: „In Standard-Software gibt es keine Codegenerierung aus einem Prozessmodell heraus, um die Wartbarkeit zu sichern.“ (GA1)

Die Mehrheit der Unternehmen versicherte im Interview, dass die Komplexität einer kompletten Unternehmenssoftware zu groß sei, als dass man aus der Prozessdarstellung automatisierten Code erzeugen könnte. Vielmehr ist die manuelle Anpassung der vorliegenden Standardsoftware die gängige Methode, neuen Code zu erzeugen.

Einsatz beim Kunden

Der Einsatz der Prozessdarstellung beim Kunden ist in der Menge etwas größer als die Anzahl der graphischen Umsetzungen (vgl. Abbildung 4.3). IAS, eBootis, Godyo und Informing nutzen trotz visueller Darstellung der Prozesse diese nicht beim Kunden. Ansonsten kommen bei allen Systemen, die eine graphische Umsetzung anbieten, diese auch beim Kunden zur Anwendung. Dabei hat ein solcher Einsatz, wie in mehreren Interviews versichert wurde, oft nur den Charakter einer Visualisierung.

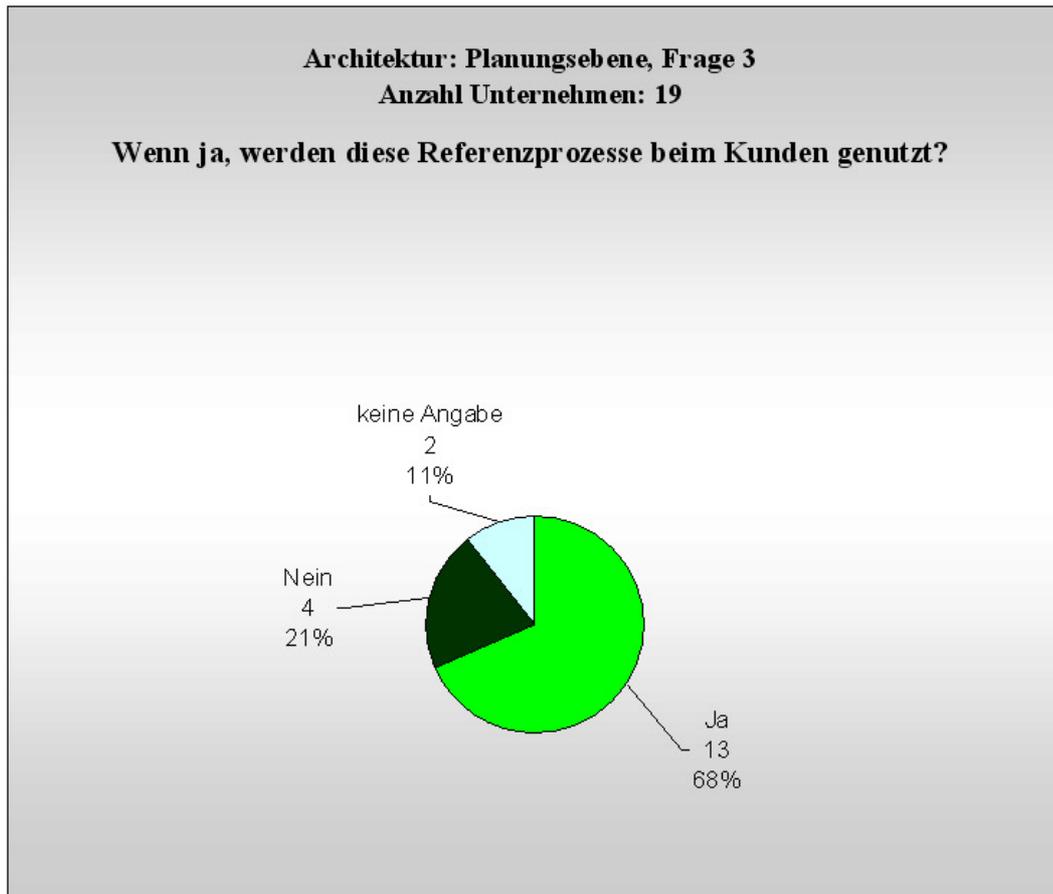


Abbildung 4.3 Verwendung der Prozesse beim Kunden

Es ist daher auch kein Muster zu erkennen, ob besonders SOA-basierte Systeme einen Kundeneinsatz vorsehen oder eher die Anbieter mit einem eigenentwickelten Systemkonzept.

Standards in der Prozessgestaltung

Etwa zwei Drittel der Anbieter setzt in Sachen Prozess-Gestaltungsstandards auf eigene Entwicklungen und nicht auf vorgegebene Referenzprozesse (vgl. Abbildung 4.4). Die häufigste Begründung war die individuelle Ausrichtung auf den Kunden und seine Bedürfnisse. Nur in Bereichen, in denen es bereits etablierte Standards gibt (z.B. Pharmazie), werden diese auch umgesetzt (Bsp. GuS mit der Umsetzung von GAMP (Good Automation Manufacturing Practice)). Unternehmensübergreifende Standards wurden in diesem Zusammenhang kaum genannt. Abacus unterstützt als einziger Anbieter Prozessstandards von GS1. Die restlichen sechs Unternehmen nutzen nach eigenen Angaben eine Auswahl an Branchenstandards bzw. modulspezifische Standards z.B. für die Logistik.

Die Größe eines Unternehmens spielt ebenfalls eine Rolle. Microsoft gab an, dass es für sein ERP-System Axapta keine pauschale Antwort auf diese Frage gebe, da durch das Partnernetzwerk branchenspezifischen Lösungen nicht alle berücksichtigt werden könnten. Doch dies verlagert die Problematik einer standardisierten Prozessgestaltung nur eine Ebene nach unten.

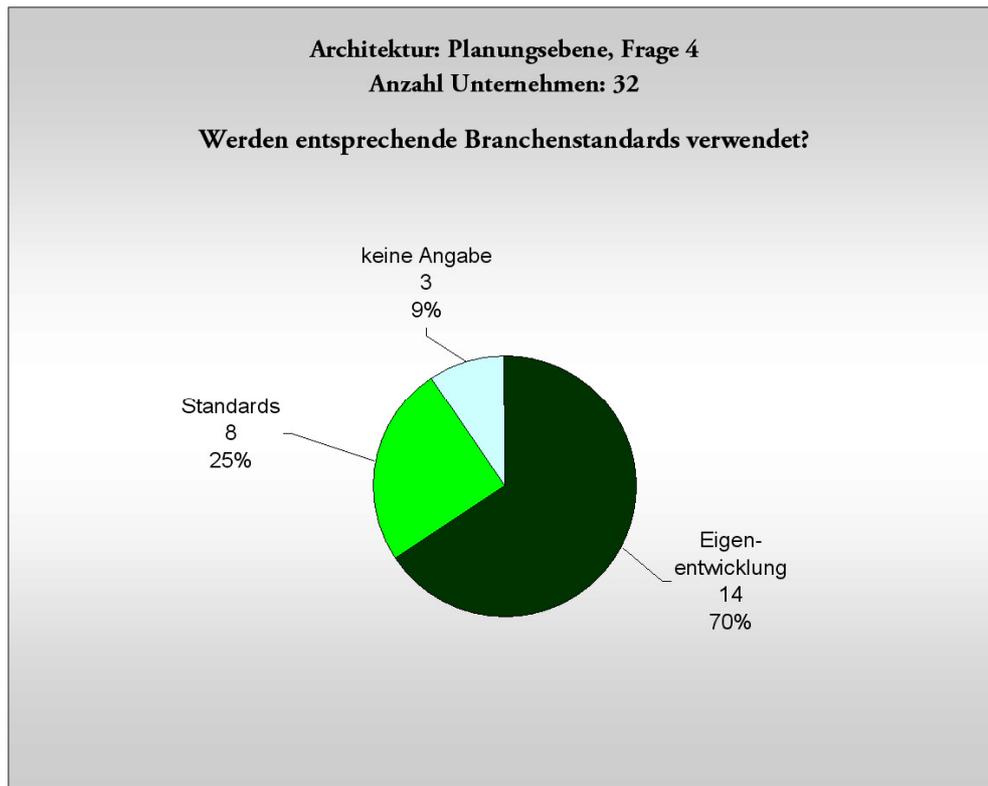


Abbildung 4.4 Branchenstandards im Einsatz

Der Zwang von außen (ob durch festgelegte oder Best Practice Standards) ist ausschlaggebend, entsprechende Prozesse einzuführen. Ansonsten werden immer Individuallösungen bevorzugt.

Modifikationsmöglichkeit des Benutzers

Zuletzt stellte sich die Frage der Umgestaltungsmöglichkeit, die der Kunde innerhalb seiner Prozessgestaltung hat. Unerwarteter Weise haben alle Anbieter bis auf Semiramis, Informing und Oracle eine Möglichkeit, mit der Kunden ihre Prozesse modifizieren oder sogar neu gestalten können (vgl. Abbildung 4.5). Diese Option ist nach Ansicht einiger Anbieter „im Normalfall in Unternehmen nicht erwünscht.“ (FO1)

Die große Mehrheit der Softwarehäuser lässt eine Prozessmodifikation trotzdem zu. Allerdings ist dies bei einigen Systemen nicht ganz unproblematisch für den Anwender. Microsoft gestattet dies mit dem Kauf einer Entwicklerlizenz, IAS und Compra setzen tiefere Entwicklungskennntnisse voraus, GUS lässt es nur in Ausnahmen zu, Abacus lässt dies normalerweise durch Partner erledigen und SHD steckt genau definierte Grenzen, innerhalb derer sich der Kunde mit seinen Änderungen bewegen darf.

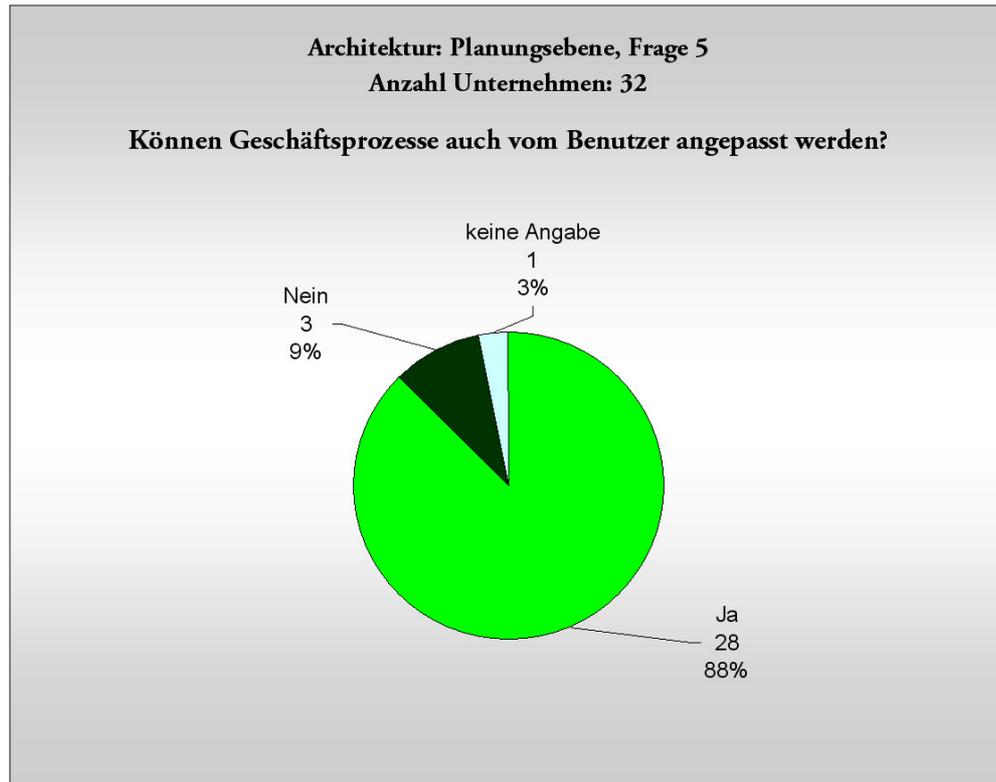


Abbildung 4.5 Anpassungsmöglichkeiten

Trotzdem ist es erstaunlich, dass, obwohl nur die Hälfte der Anbieter eine graphische Repräsentation unterstützt, fast alle eine Modifikation in der Geschäftsprozesslogik erlauben.

4.1.1.4 Fazit

Man kann festhalten, dass trotz der fehlenden visuellen Umsetzung die Möglichkeit zum Eingriff in die Prozessabfolge für den Benutzer ein nützliches Gestaltungswerkzeug bietet. Trotzdem werden diejenigen Kunden, deren Prozessgestaltung sich nicht graphisch abbilden lässt, es schwer haben, ohne Hilfe des IT-Partners Änderungen im Prozessmodell vornehmen zu können. Bei mehreren Anbietern sind außerdem Entwicklungskennntnisse erforderlich, die sicher nicht jeder Kunde mit sich bringt.

Aber das ist auch nicht erforderlich, wie die Meinung vieler Systemanbieter ist: „Eine Standardlösung anzupassen, wird teurer sein, als eine Individuallösung zu erzeugen“. (B14)

Diese Einstellung scheint sich schon in der Prozessplanung widerzuspiegeln. Eine automatisierte Codegenerierung aus Prozessmodellen heraus ist laut Anbietern unrealistisch, denn der Aufwand, entsprechende Codegeneratoren zu entwickeln, ist aufgrund der zahlreichen internen Abhängigkeiten, die in einem ERP-System beachtet werden müssen, viel zu groß. Vielmehr ist es wichtig, durch eine individuelle Prüfung Gegebenheiten und Anforderungen von Kunde zu Kunde neu zu ermitteln und die speziellen Problemstellungen zu lösen.

In der Vision des Business Process Managements (BPM) soll es am Ende möglich sein, lauffähigen Code allein aus der Modellierung heraus zu gewinnen. Davon ist die Realität noch

sehr weit entfernt. Zwar unterstützen einige Anbieter branchenspezifische Prozessstandards, trotzdem wurde der Eindruck gewonnen, dass Standardisierungen in der Prozessgestaltung, eine Grundvoraussetzung für diese Vision, bereits in der Konzeption einer betrieblichen Software hinter die Notwendigkeit einer individuellen Prozessanpassung beim Kunden gestellt werden.

Die meisten Hersteller sehen den Bedarf mehr in der kundennahen Betreuung und in der Berücksichtigung ihrer individuellen Prozessabläufe als in der Umsetzung von vorgegebenen Standards. Das dürfte es Vertretern wie GS1 (vgl. Kapitel 3.2.1.7) deutlich schwieriger machen, für eine generelle Standardisierung einzutreten.

Zwei Drittel der Anbieter gestalten die Prozesse ihrer Kunden nach dem aktuellen Bedarf. Und das aus einem einfachen Grund: „Kleinere Unternehmen brauchen eher lösungsorientierte und einfach einzusetzende Anwendungen für einzelne Problemstellungen.“ (ME3)

Dies lässt sich mit Standardsoftware nur über entsprechende Anpassungen oder direkt als Individuallösung realisieren. Deswegen ist es nicht verwunderlich, dass alle bis auf eine handvoll Softwarehäuser eine Codegenerierung erst gar nicht versuchen umzusetzen. Zusammenfassend lässt sich sagen, dass eine visuelle Prozessdarstellung umgesetzt ist, Prozessänderungen (wobei teilweise beschränkt) ebenfalls möglich sind, Prozessstandards in der Modellierung jedoch die Ausnahme bilden.

4.1.2 Systemarchitektur

In diesem Abschnitt wird das konzeptionelle und architekturelle Rahmenwerk (Systemkonzept und Systemarchitektur) eines ERP-Systems näher betrachtet. Das Konzept beschreibt die Anforderungen, die ein System erfüllen muss. SOA ist in diesem Zusammenhang das Schlagwort, um das es geht. Viele Anbieter benutzen das Kürzel als Technologie-Erklärung, viele aber auch nur als Werbewort. Auf diesem Konzept baut die eigentliche Architektur auf, die physische Realisation des Systems auf Clients und Servern.

Plattformunabhängigkeit, Middleware und die verwendete Browsertechnologie beschreiben eher technologische Aspekte, die jedoch für die logische Struktur ebenso wichtig sind wie das zugrunde liegende Konzept. Die freie Wahl der Plattform bedeutet für das System zum einen eine deutlich größere Flexibilität aber auch zugleich eine fehlerfreie Umsetzung auf verschiedenen Betriebssystemen. Die Middleware ist die technische Grundlage für die Applikationen. Der Einsatz eines Browsers schließlich bedeutet eine Abkehr von Fat-Clients hin zu portablen Thin-Clients.

4.1.2.1 Verlauf der Befragung

Der recht technisch durchsetzte Fragenblock hat niemandem der Befragten wirkliche Schwierigkeiten bereitet. Besonders bei den letzten vier Fragen hinsichtlich Systemarchitektur, Plattformunabhängigkeit, Middleware und Browsertechnologie gab es klare Antworten. Erstaunlich war, dass bei der Frage hinsichtlich einer SOA teilweise gezögert wurde. Während der Interviews entstand der Eindruck, dass SOA als Konzept evtl. nicht ganz verstanden wurde bzw. man sich nicht direkt als „nicht-SOA-fähig“ verstanden wissen wollte.

4.1.2.2 Zusammenfassung der Ergebnisse

- SOA-Ansatz: 25 Prozent, SOA: 28 Prozent

- 2-Tier: 31 Prozent, 3-Tier: 56 Prozent, 4-Tier: 13 Prozent
- 27 Thin-Clients werden angeboten, davon 8 Browser.
- Plattformunabhängigkeit ist bei 59 Prozent der Unternehmen umgesetzt.
- J2EE ist bei 36 Prozent im Einsatz, .NET bei 19 Prozent, Eigenentwicklungen bei 29 Prozent.
- Java und HTML sind bei browserbasierten Systemen weit vorne im täglichen Einsatz.

4.1.2.3 Auswertung

Systemkonzept

Der Anteil von service-orientierten Konzepten nimmt in der Gestaltung der ERP-Systeme immer weiter zu.

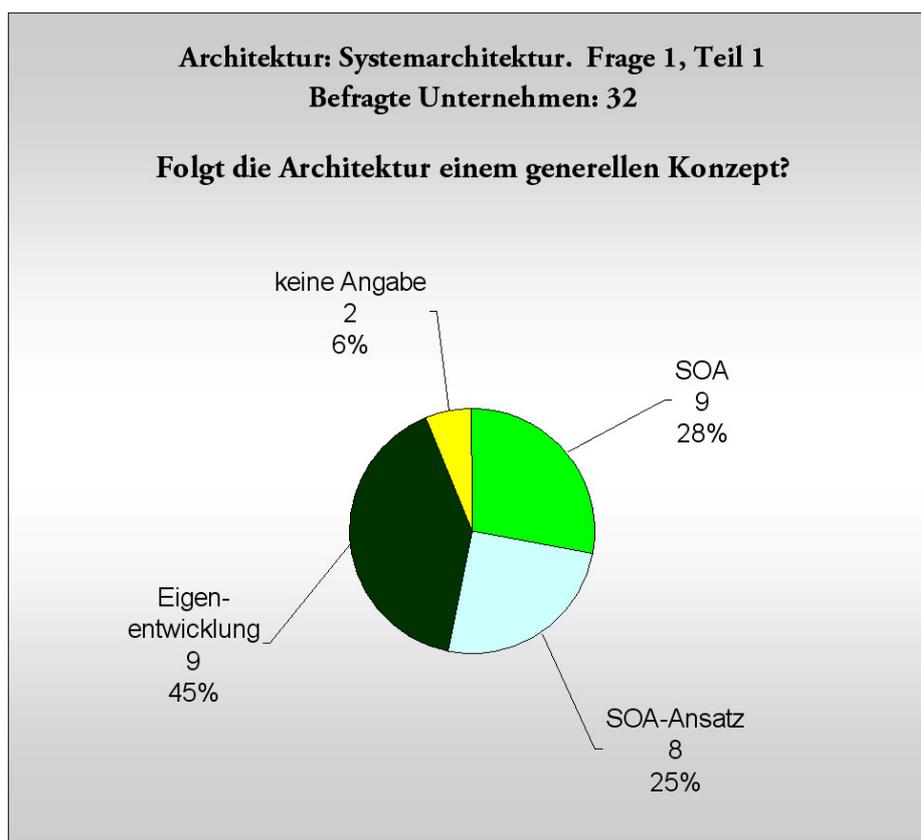


Abbildung 4.6 Umgesetzte Konzepte

Etwas weniger als die Hälfte der Befragten gab an, ein eigenes Systemkonzept entwickelt zu haben, nach dem ihre Software arbeitet, ohne nähere Angaben zu machen (vgl. Abbildung 4.6). Neun Anbieter richten sich dagegen nach den Prinzipien einer Service orientierten Architektur: Microsoft (Axapta), IAS, Lawson, IFS, Ramco, SAP, Intraprend, Alea und Oracle.

Der Rest gab an, zumindest Teilprinzipien einer SOA zu verfolgen bzw. SOA-fähig zu sein, d.h. sich gut in eine bestehende SOA-Landschaft einfügen zu können.

Auffällig dabei ist: Drei Softwarehäuser, IAS, Lawson und SAP haben nach eigenen Angaben keine oder nur eine sehr begrenzte Verwendung für Web Services als Integrationsmechanismus. Als Middlewareplattformen kann ebenfalls keine Präferenz erkannt werden, z.B. nutzt Ramco sowohl J2EE als auch .NET, Lawson und IFS nur J2EE und IAS benutzt eine Eigenentwicklung. Es zeigt sich, dass weder Web Services noch eine Middleware ausschlaggebend oder notwendige Bedingung sind, um eine SOA umzusetzen.

Systemarchitektur - Client/Applikation/Datenbank

Abbildung 4.7 beschreibt die physische/logische Umsetzung der Systemarchitektur, das heißt 2-Tier Systeme lassen sich in zwei Einheiten teilen, Client und Applikationsserver/Datenbank, 3-Tier verteilt sich auf drei (Client - Applikationsserver - Datenbank) und 4-Tier auf vier Einheiten (Client - Webserver - Applikationsserver - Datenbank).

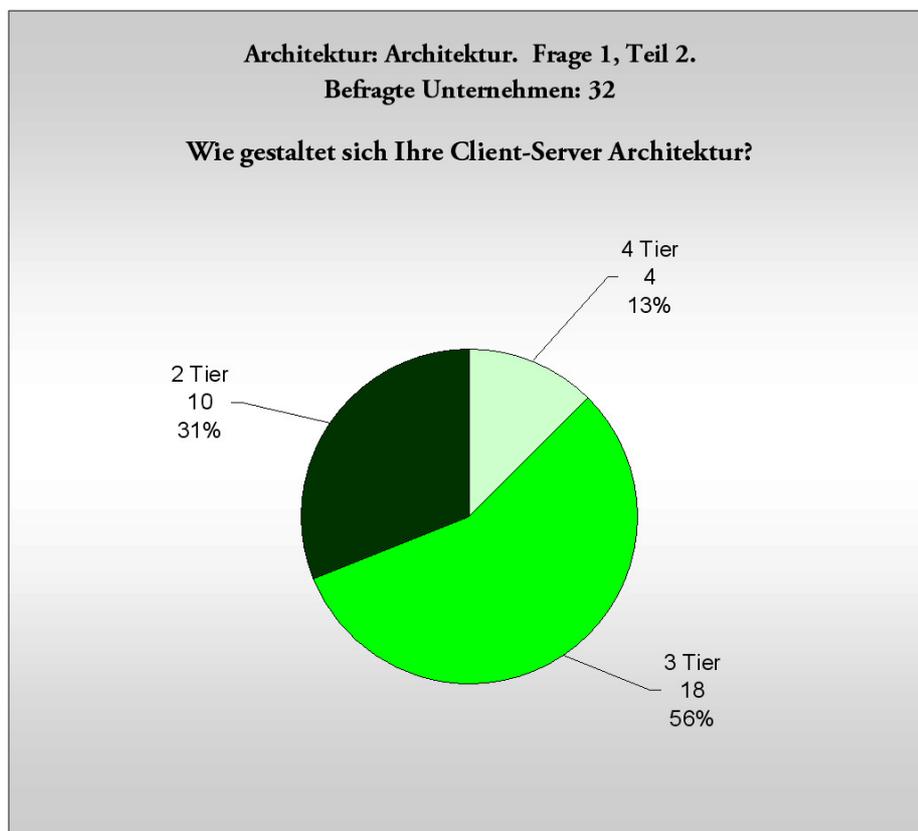


Abbildung 4.7 Verwendete Architekturen

Die größte Verbreitung haben Systemarchitekturen mit drei Komponenten. Die klassische drei Schichten Architektur (Client - Applikationsserver - Datenbankserver) wird etwa von der Hälfte der Anbieter umgesetzt. Diese bieten eine Thin-Client Lösung an, acht noch zusätzlich eine Fat-Client Lösung. Auffällig ist: nur vier der fünfzehn 3-Tier Architekturen basieren auf einer selbst entwickelten Middleware (SAP, Informing, Abacus und ams). Die anderen Architekturen nutzen bereits entwickelte Middlewareprodukte (z.B. Websphere (J2EE) oder BizTalk (.NET)).

Überrascht hat der hohe Anteil an 2-Tier Architekturen. Zehn Anbieter nutzen eine solche 2-Tier Lösung. Vier davon bieten nur einen Fat-Client an, während die anderen eine Thin-Client Variante bevorzugen. Hier ist der Einsatz von kommerzieller Middleware deutlich größer, nur SHD verwendet eine Eigenentwicklung. Intraprend besitzt unter den 2-Tier Lösungen eine Sonderstellung, da das System nur aus der Client Seite (HTML-gestützt) und einer Datenbank besteht.

Die 4-Tier Systeme haben als Erweiterung einen Webserver in ihre Struktur mit aufgenommen. Er dient als Intermediär zwischen Client und Applikationsserver. Nur einer der vier Anbieter setzt hier eine Eigenentwicklung als Middleware für sein System ein: IAS.

Einen starken Zusammenhang zwischen dem Systemkonzept und der Systemarchitektur kann man nur bedingt erkennen. Alle 4-Tier Systeme besitzen nach eigenen Angaben eine serviceorientierte Architektur (IAS, Lawson, Ramco, Oracle), vier SOA-basierte Systeme verwenden eine 3-Tier Architektur (IFS, SAP, Alea, Microsoft (Axapta)), während nur ein 2-Tier System eine SOA umsetzt (Intraprend). Die lose Kopplung der drei bzw. vier logischen Bausteine Client, Webserver, Applikationsserver und Datenbank, scheinen eine SOA zumindest zu begünstigen.

Systemarchitektur - Clients

Es lässt sich feststellen, dass der Browser als Client eine immer wichtigere Rolle spielt (vgl. Abbildung 4.8). Etwa ein Drittel der Thin-Clients werden nur durch Browser repräsentiert, was zum Schluss führt, dass im Bereich der Front-Ends die Nutzung von bereits vorhandener Technologie (HTTP, HTML) als möglicher Ersatz zu eigenen User Interfaces als echte Alternative angesehen wird.

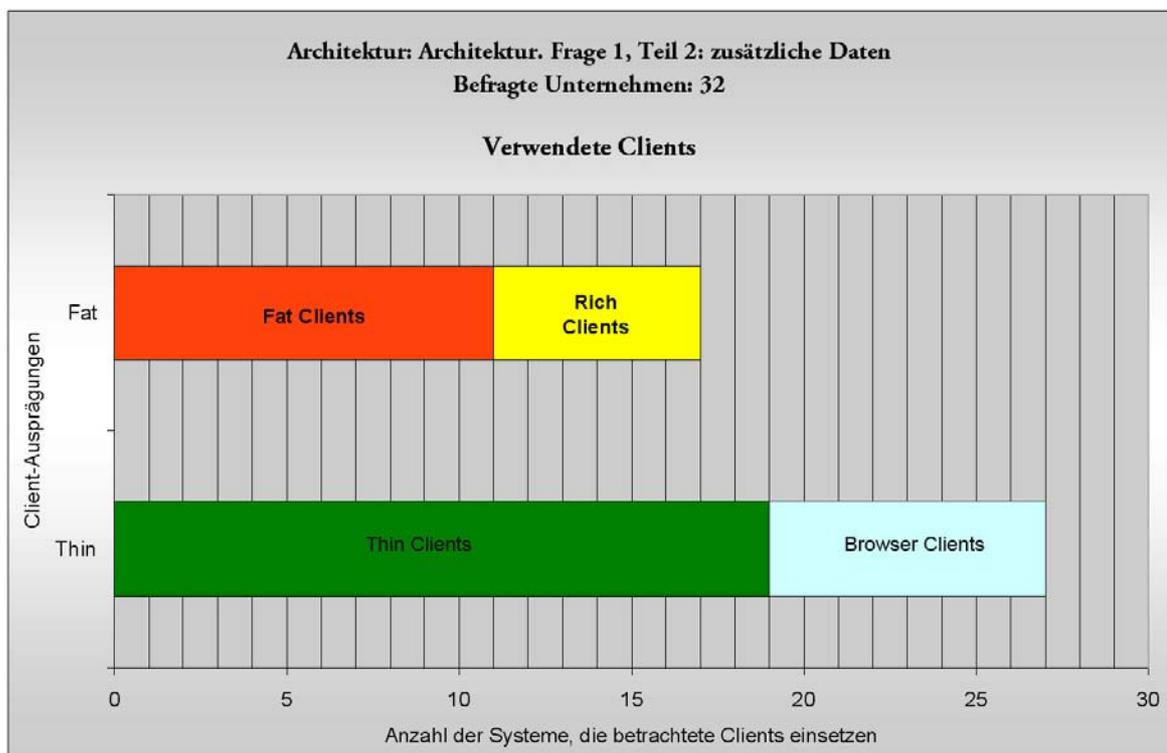


Abbildung 4.8 Verwendete Clients

Insgesamt ist die Anzahl der Thin-Clients zusammen mit den Browsern deutlich höher als die Anzahl der Fat-Clients zusammen mit den Rich-Clients. Trotzdem werden angesichts der scheinbar Web-orientierten Clients immer noch von fast der Hälfte der Anbieter Front-Ends mit zusätzlicher Applikationslogik angeboten.

Plattformunabhängigkeit

Die Zahl der plattformunabhängigen Systeme ist etwas größer als die derjenigen Anwendungen, die bestimmte Betriebssysteme oder Middlewaretechnologien benötigen (vgl. Abbildung 4.9). Bis auf eine Anwendung, USE-21 von ERP-21, die Linux als Betriebssystem voraussetzt, sind die restlichen der plattformabhängigen ERP-Systeme auf Windows bzw. (MS) SQL als Datenbank angewiesen. Günther-BS und Topix lassen zusätzlich noch Macintosh als Plattform zu.

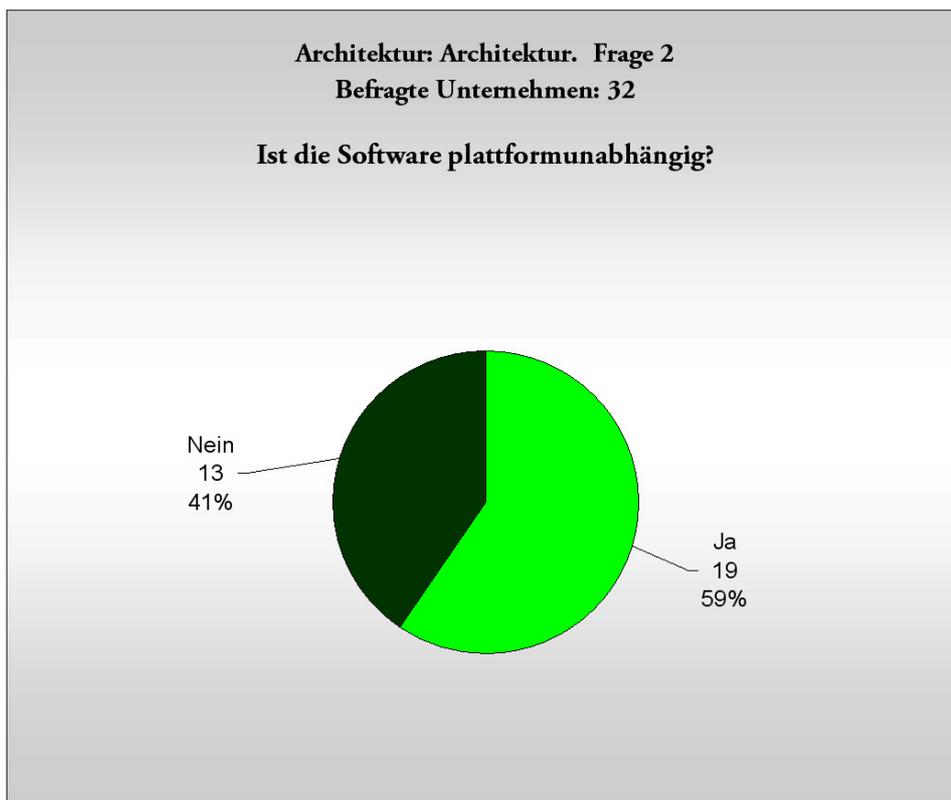


Abbildung 4.9 Verbreitung der Plattformunabhängigkeit

Alle 4-Tier Systeme sind unabhängig in ihrer Plattformwahl. Das spricht für ihre auf Webanwendungen ausgerichtete Strategie. Dabei ist die Plattformunabhängigkeit nicht an eine Programmiersprache gebunden, allerdings verwenden 13 der 19 Anbieter, die ihr System als plattformunabhängig charakterisiert haben, Java als grundlegende Programmiersprache (vgl. Kapitel 3.3.1.3). Sechs Anbieter (Ramco, IAS, IFS, SAP, Oracle und SynERP) verwenden außerdem C++, PL/SQL, Visual Basic, ABAP oder Delphi.

Middleware

Fast drei Viertel der Befragten nutzen bereits existierende Middleware Produkte als Applikationsgrundlage (vgl. Abbildung 4.10). Dabei ist relativ gut zu differenzieren, welche der beiden grundsätzlichen Frameworks (J2EE oder .NET) die größte Verbreitung hat: J2EE. Drei Anbieter, Ramco, eBootis und IFS, nutzen beide Plattformen, wobei eBootis und IFS .NET nur für ihre Clients nutzen.

Somit sind zwei der drei Systeme, die beide Plattformen unterstützen, primär zu J2EE zu zählen, was dessen Anteil auf elf Systeme erhöhen würde. Neun Anbieter geben an, eine eigene Middleware-Lösung implementiert zu haben. Unter den restlichen fünf ist eine Runtime von Acucorp (Hilmer) für die Cobol-Anwendung und GSWeb (Seat-1) als OS Lösung. Nur bei Semiramis wird zusätzlich noch CORBA angeführt, welches genutzt werden kann. Somit ist CORBA als Middleware-Lösung praktisch kaum noch vorhanden. Günther-BS basiert auf der proprietären Software Filemaker und Topix auf 4D-Software.

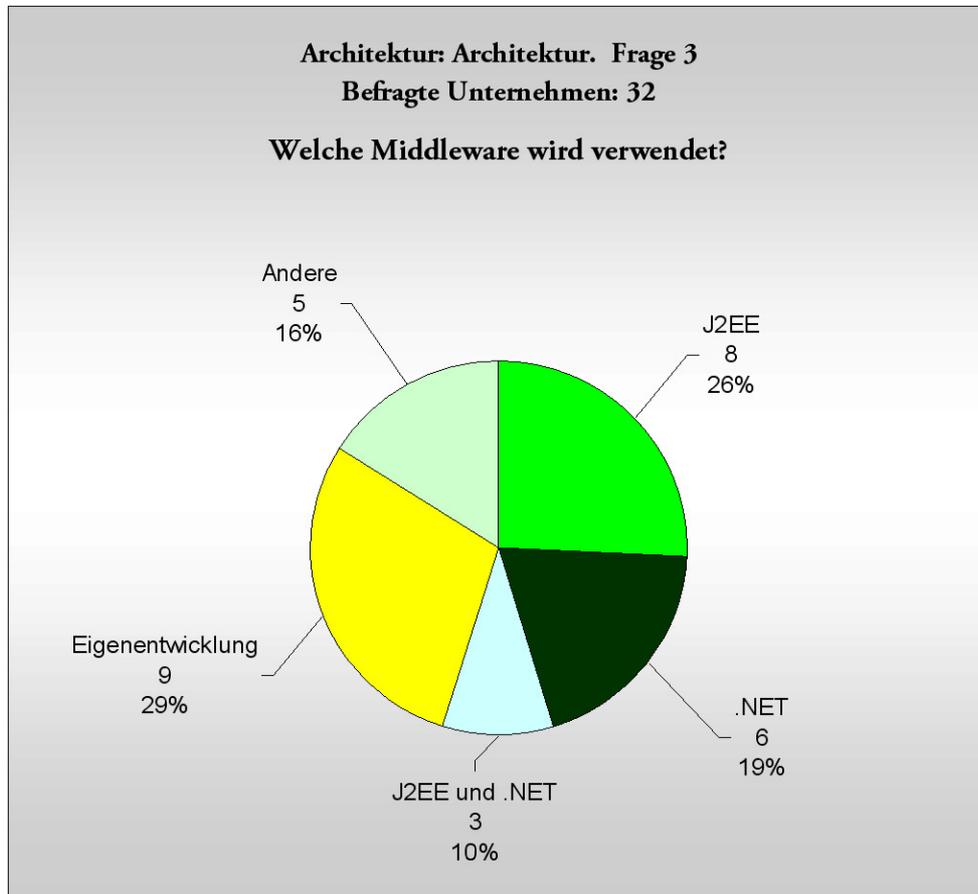


Abbildung 4.10 Verwendete Middleware

Interessant ist der relativ hohe Anteil (29 Prozent) an Eigenentwicklungen in diesem Bereich. Von diesen sind alle bis auf AMS, Informing und Microsoft (Navision und Axapta) plattform-unabhängig.

Browsertechnologie

„Damals (2000) war Java für die Clients noch zu langsam und unflexibel“ (WE2), kommentierte ein Interviewpartner die frühen Einsatzszenarien der Programmiersprache. Browser und insbesondere Java scheinen diese ehemalige Einstellung endgültig zu widerlegen. Überall da, wo Browser als Darstellungsmittel genutzt werden, wird auch Java als Unterstützungswerkzeug angeführt. HTML scheint als reines Darstellungswerkzeug ebenfalls eine breite Nutzung zu erfahren.

Sieben Systeme bieten keinerlei Browserzugriff an: B.I.M., SynERP, nGroup, ERP4all, Alea, Hilmer und SHD. Godyo bietet nur beschränkte Optionen an. Alle anderen, auch Bäurer, PSIPenta und AMS als Fat-Client Anbieter, bieten die Möglichkeit, via Browser auf Daten zugreifen zu können. Alle verwenden dabei unter anderem Java als Zugriffsmöglichkeit (vgl. Abbildung 4.11). Eine normale Darstellung via HTML wird von sieben Befragten explizit angegeben.

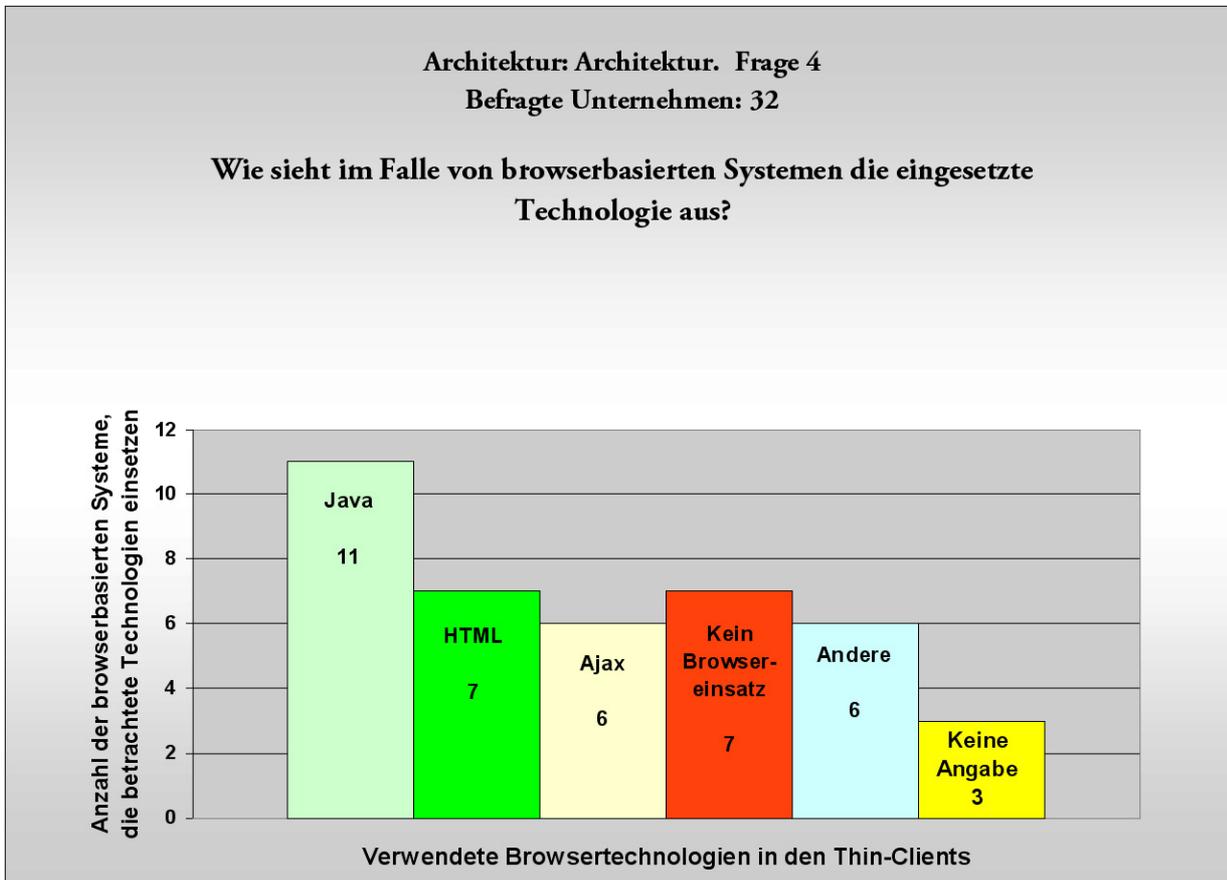


Abbildung 4.11 Browsertechnologien im Einsatz

Ajax wird in sechs Fällen als zusätzliches Werkzeug genutzt. Ramco sowie Microsoft haben als zusätzliche Technologie noch XML angeführt, SAP nutzt eine selbst entwickelte Technologie (Web Dynpro).

4.1.2.4 Fazit

Die reale Umsetzung einer SOA in einem System ist nur bei neun der Befragten geschehen. Zumindest haben mehr als die Hälfte aller Anbieter Tendenzen in diese Richtung. Das zeigt, dass sich, unabhängig von der Branchenausrichtung, Service-Strukturen langsam durchsetzen. Etwa die Hälfte aller SOA Vertreter haben eine 4-Tier Architektur mit einem Webserver als neues Bindeglied.

„Wir haben HTML-Sichten auf die Daten, dadurch sind wir hoch flexibel“ (FI5), versicherte ein Anbieter. Der Browser als einzige Client-Lösung gewinnt an Bedeutung. Nur etwa ein Fünftel der Befragten setzen keinen Browser als Zugriffsmöglichkeit ein. Thin-Clients wandeln sich von Terminal-Lösungen zur webbasierten Darstellung, gleichzeitig scheint sich Java endgültig als nützliches und performantes Werkzeug auch auf der Client-Seite etabliert zu haben. Die Ausrichtung des Geschäfts geht somit mehr zu webbasierten Business Modellen hin.

Plattformunabhängigkeit ist ebenfalls stark vertreten. Einer der Befragten nannte einen wichtigen Grund dafür: „Plattformunabhängigkeit eröffnet ein breites Kundenspektrum.“ (GA2)

Das Festlegen auf eine Plattform verringert die Flexibilität und verschließt einen Zugang zu potentiellen Kunden. Weniger als die Hälfte derjenigen, die an eine Plattform gebunden sind, nutzen das .NET Framework.

Die Verbreitung von Microsoft als Plattformlieferant ist wider Erwarten eher gering. Der Anteil von J2EE an den Middlewaresystemen ist etwas höher als der von .NET. Zählt man die beiden Anbieter eBootis und IFS mit den .NET Clients und ihrer J2EE basierten Middleware ebenfalls zu J2EE, beträgt das Verhältnis zwischen beiden etwa eins zu zwei.

Es lässt sich feststellen, dass der Zugang zum Internet fast überall zu einer entsprechenden zusätzlichen Möglichkeit geführt hat, Daten via Browser auszutauschen. Java hat sich in diesem Zusammenhang als nützliches Werkzeug entpuppt. Ob dies auch die Verbreitung von SOA als Systemkonzept fördert, konnte nicht geklärt werden.

Fest steht, dass die Hälfte der SOA Anbieter einen Webserver mit in ihre Architektur aufgenommen haben. Acht Anbieter haben den Browser bereits als einzigen Client eingeführt und die Tendenz scheint sich in diese Richtung weiter fortzusetzen. Im Middleware-Bereich ist es offen, wer von den beiden großen Plattformen für Middlewaresysteme, J2EE oder .NET, am Ende den Markt eventuell dominieren wird. J2EE hat den Vorteil, dass es auf dessen Basis bereits viele ausgereifte Applikationsserver am Markt gibt und die Plattformunabhängigkeit gewährleistet ist. Trotzdem hat .NET inzwischen viele Nutzer gefunden, da eine reibungslose Integration aller Microsoft-Produkte, vor allem auch auf Client-Seite, möglich ist.

4.1.3 Modularchitektur

Im Folgenden stehen die Module eines ERP-Systems im Vordergrund. Ausprägungen des Moduleinsatzes in den jeweiligen Systemen der Befragten wird als erstes untersucht. Nicht alle Systeme haben unbedingt den gleichen Funktionsumfang. Dies gilt besonders für CRM- und BI-Module, für die es neben den klassischen ERP-Häusern viele Anbieter gibt, die entsprechende Software als Einzellösung anbieten. Neben der Frage, ob solche Systeme sowohl in der Komplettlösung integriert sind als auch selbst entwickelt wurden, wird zuletzt die Releasefähigkeit der Systeme im Falle einer Individualisierung durch den Kunden selbst untersucht.

4.1.3.1 Verlauf der Befragung

Die Befragung in diesem Abschnitt gestaltete sich insofern schwierig für die Befragten, da die Anbieter anhand einer auf Porter (Porter, 2000, S. 66) basierenden Übersicht der primären und sekundären Aktivitäten eines Unternehmens die Funktionalitäten ihrer Software beschreiben sollten.

4.1.3.2 Zusammenfassung der Ergebnisse

- Mängel in der Funktionsunterstützung herrschen im Personalwesen, Finanzbuchhaltung, Produktentwicklung und Geschäftsführung/Management bei einigen ERP-Anbietern.
- Customer Relationship Management (CRM) ist intern umgesetzt, Business Intelligence (BI) wird von Partnern bezogen.
- Die Releasefähigkeit bleibt bei Änderungen im System meist erhalten, aber unter Einschränkungen.

4.1.3.3 Auswertung

Moduleinsatz

Abbildung 4.12 zeigt, dass besonders in den sekundären Aktivitäten, die ein ERP-System abdecken soll, häufig Fremdanbieter hinzugezogen werden. Im Personalwesen und in dem Finanz- und Rechnungswesen gibt es dabei besonders viele fehlende Funktionalitäten.

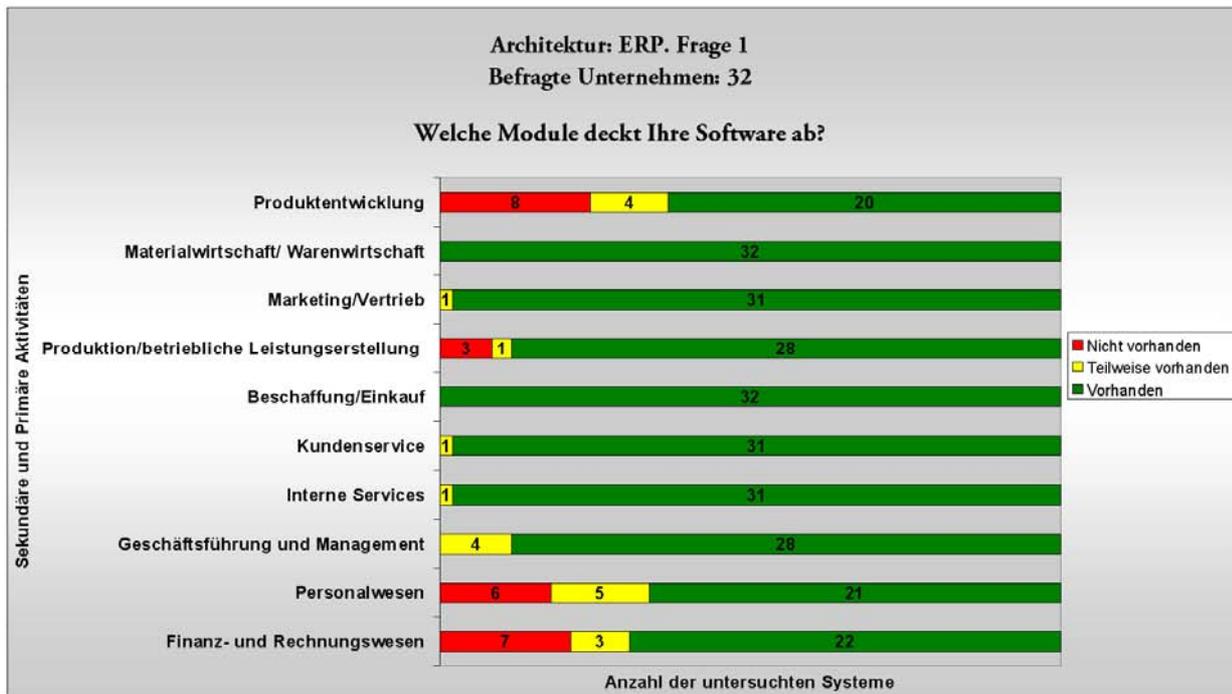


Abbildung 4.12 Besonders im sekundären Bereich Fremdanbieter benötigt

Für Geschäftsführung/Management werden ebenfalls ausgiebig hauserne Programme genutzt, wie die Anzahl der Partnerlösungen in den folgenden zwei Paragraphen zeigt. Welcher Fremdanbieter auch genutzt wird, meist erscheinen die Anwendungen nicht als fremde Software für den Kunden, sondern werden in die hauserne Lösung integriert.

Im primären Bereich bieten die meisten Anbieter die benötigte Funktionalität an. Auf den ersten Blick scheint die fehlende Funktionalität in der Produktentwicklung überraschend zu sein. Der Grund dafür liegt jedoch in der Branchenausrichtung einiger Interviewpartner. ERP-Anbieter, die sich z.B. auf den Einzelhandel spezialisiert haben (wie SHD), benötigen kein solches Modul. Topix ist auf Handel und Dienstleistungsunternehmen spezialisiert, wo durch Bemühungen sowohl in der Produktentwicklung als auch in der Produktion selbst sekundäre Bedeutung haben. Alea benötigt als Versandhandels-Spezialist keine Produktion im Portfolio. Die restlichen Anbieter unterstützen dagegen auch Anwender, die im produzierenden Bereich tätig sind. Nur jeweils ein Anbieter hat nach eigenen Angaben Defizite im Bereich Marketing/Vertrieb (IntarS), Interne Services (myFactory) und im Kundenservice (Hilmer).

Insgesamt gaben zwölf Anbieter an, alle Aktivitäten durch ihr System unterstützen zu können. Elf davon haben eine Kundenanzahl, die 150 beträgt bzw. deutlich höher ist. Nur Jentech entspricht mit aktuell 15 Kunden nicht dieser Eigenschaft.

CRM/BI Integration

Die früheren Zeiten, in denen CRM bzw. BI noch nicht standardmäßig in ERP-Systemen integriert waren, sind vorbei. CRM und BI sind als Module beide in den ERP-Systemen fast durchgängig vorhanden (vgl. Abbildung 4.13). Allerdings nicht unbedingt als eigene Lösung, wie man im folgenden Abschnitt sehen wird.

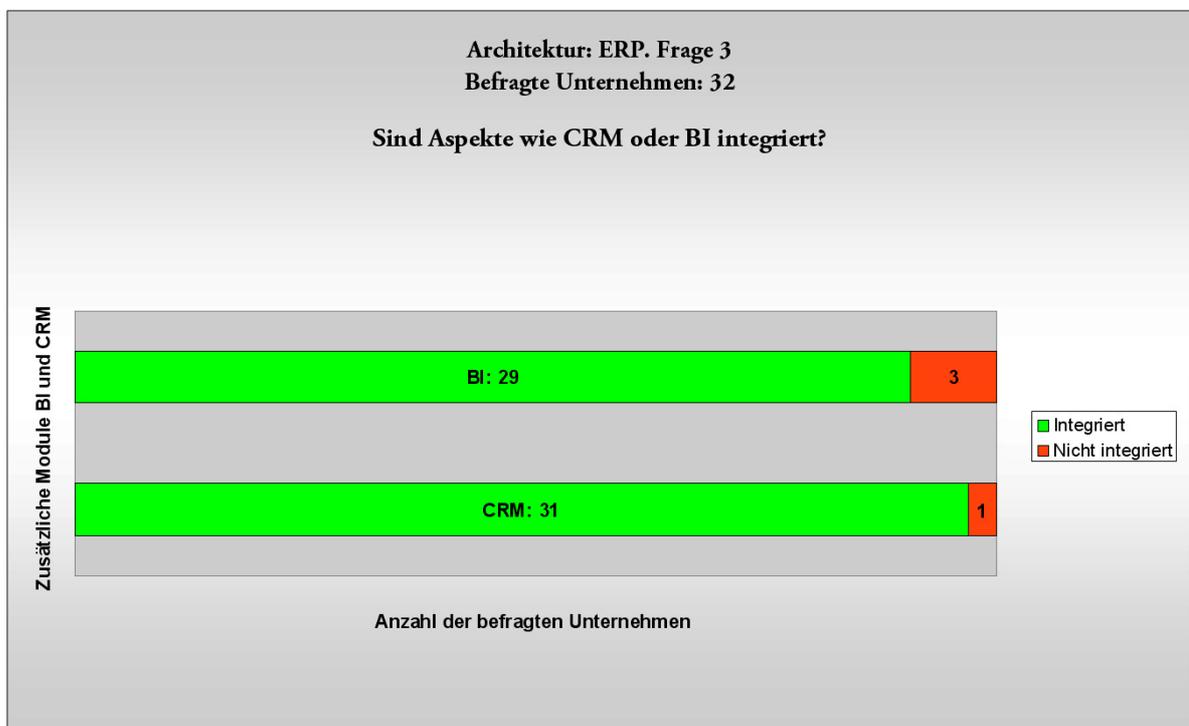


Abbildung 4.13

Einsatz von CRM und BI in den Systemen

CRM findet mit 31 Unternehmen die häufigste Verbreitung. Im BI Bereich sind es nur drei Anbieter, die keine Funktionalität umsetzen: Hilmer Software bietet weder eine CRM- noch eine BI-Lösung an. ERP4all hat keine vollwertige BI-Lösung, da die Unternehmensgröße der Kunden zu klein ist, stattdessen werden hier einfache Reportingsysteme eingesetzt. Günther-BS besitzt ebenfalls keine Business Intelligence Lösung.

CRM/BI Eigenentwicklung

Im CRM-Bereich sind fast alle Module selbst entwickelt (vgl. Abbildung 4.14), bis auf das Modul von Bäurer, hier wird eine Sage Applikation genutzt, und das Modul von nGroup, das auf einer Microsoft CRM-Lösung aufsetzt. BI-Module hingegen werden sehr oft von einem Fremdanbieter bezogen. Nur etwa ein Drittel der Anbieter kann auf eine Eigenentwicklung zurückgreifen. Den größten Anteil bei fremden BI-Lösungen besitzt Cognos mit fünf Systemen.

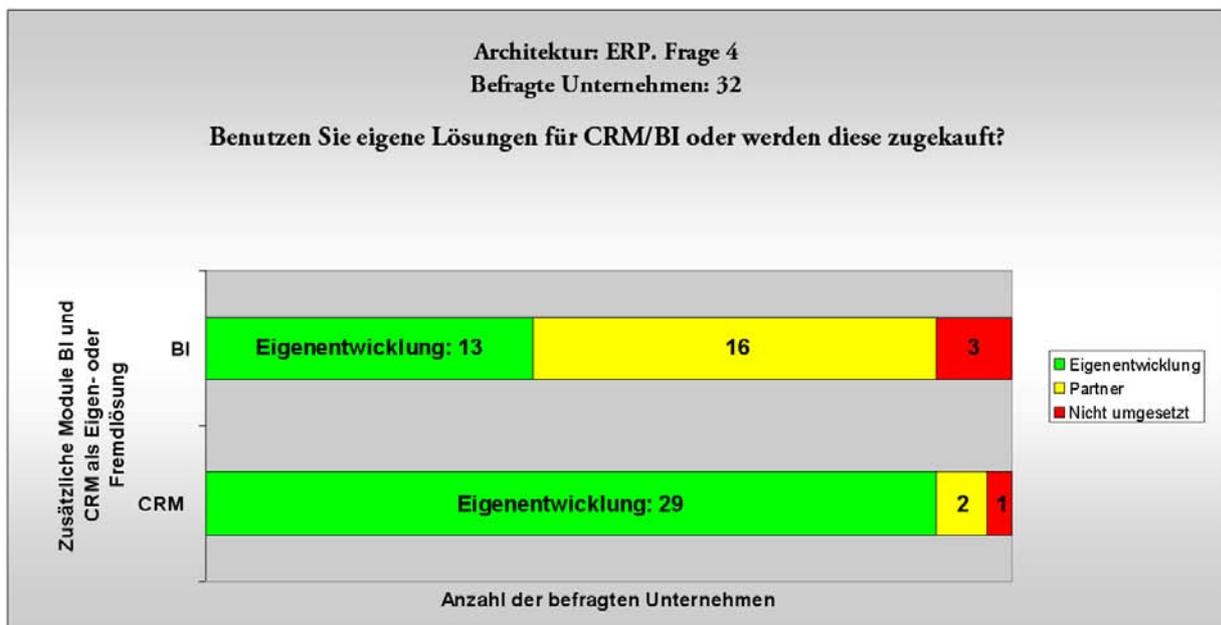


Abbildung 4.14 Eigenentwicklungen im CRM/BI Bereich

CRM weist aufgrund seiner direkten Auswirkung auf die Unternehmensleistung einen starken integrativen Funktionscharakter für ein ERP-System auf. Deswegen werden die meisten CRM-Systeme von den jeweiligen ERP-Anbietern selbst entwickelt. BI-Systeme werten in spezieller Form Datenbestände aus, die während des operativen Geschäftes anfallen. Komplexität in den dazu verwendeten Techniken bzw. bereits bestehendes Know-how von Fremdanbietern lässt es einfacher erscheinen, eine bereits existierende Lösung einzubinden anstatt selbst neu zu entwerfen.

Releasefähigkeit trotz Customizing

78 Prozent der Anbieter garantieren eine Releasefähigkeit trotz Customizing durch den Kunden (vgl. Abbildung 4.15). Meist werden eigene Datenbankfelder für Kunden definiert, die sie zusätzlich nutzen können. Es gibt aber z.B. auch Repositories, die eine Konfiguration des

Systems speichern und dem Kunden die Möglichkeit geben, außerhalb zu agieren. Prämisse ist immer, dass sich der Kunde an vorgegebene Schnittstellen halten muss, wenn er eigene Entwicklungen vornimmt.

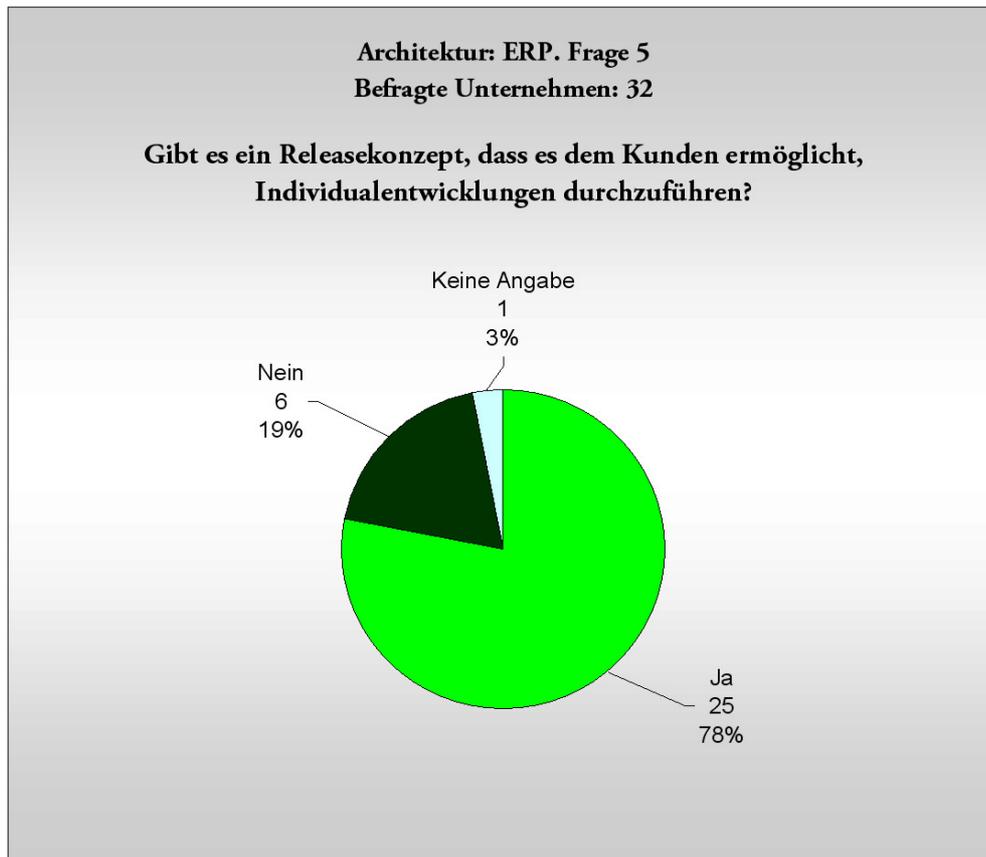


Abbildung 4.15 Releasefähigkeit größtenteils vorhanden

Das andere Drittel erlaubt keine Eingriffe bzw. sieht für ihre Kunden keinen Bedarf, da sie entweder selbst durch individuelle Weiterentwicklungen gewünschte Anpassungen vornehmen (Günther-BS) oder wie Alea nur einen Release von Standardsoftware vorsehen.

4.1.3.4 Fazit

Im sekundären Bereich gibt es viele Anbieter, die auf Partner-Software zurückgreifen. Das ist durchaus verständlich, da es sich dabei um unterstützende Prozesse handelt, die von Unternehmen zu Unternehmen ähnliche Ausprägungen haben und von spezialisierten Entwicklern angeboten werden. Die mangelnde Unterstützung der Produktplanung kann zu einem Teil auf den fehlenden Zielmarkt zurückgeführt werden, da sich manche Anbieter nur auf Handels- bzw. Dienstleistungsunternehmen spezialisiert haben.

Mit knapp 30 Prozent ist dieser Anteil trotzdem überraschend hoch. Der Weg von einer produkt-orientierten hin zu einer kunden-orientierten Sicht scheint endgültig abgeschlossen zu sein. Erwartungsgemäß war die Aussage, dass CRM-Systeme überwiegend aus dem jeweiligen Systemhaus selbst stammen, da der Kundenservice als primäre Aktivität eng mit

den anderen Aktivitäten wie z.B. Produktion oder Verkauf im täglichen Geschäft verbunden ist.

Dies ist für BI-Systeme nicht zwangsläufig der Fall, da sie in der Regel aus einem Data Warehouse bereits abgespeicherte Daten auswerten. Hier wird eine Expertenlösung deutlich häufiger eingesetzt. Die höhere Anzahl der von außen bezogenen BI-Lösungen im Vergleich zu den fehlenden Funktionalitäten im „Geschäftsführung/Management“-Bereich erklärt sich dadurch, dass Anbieter von ERP-Systemen meist Komplettpakete inklusive der Partnersoftware vertreiben, so dass dem Kunden die entsprechende Funktionalität direkt zur Verfügung steht.

Erstaunlich ist die Anzahl der Anbieter, die eine Releasefähigkeit trotz Customizing von Kundenseite her beibehalten können. Hier wird dem Kunden trotz proprietärer Software ein vermeintlich großes Maß an Eigenständigkeit zugebilligt. Eine Eigenständigkeit, die er in der Prozessgestaltung nicht unbedingt hatte und die auch in diesem Fall eher mit Einschränkungen (z.B. Namenskonventionen, extra bereit gestellte Datenfelder etc.) verbunden ist, wie ein Anbieter bestätigte: „Solange sich der Kunde an die Schnittstellen-Vorgaben hält, bleibt seine Version releasefähig.“ (ZB1)

4.1.4 Integration

Den Abschluss in der Architekturbetrachtung liefert der Bereich Integration, der sich sowohl mit internen Abhängigkeiten der Module als auch mit den Anbindungsmechanismen an Fremdsysteme beschäftigt. Im Fokus der Betrachtung stehen in diesem Zusammenhang die Daten- und Applikationsebene, über die integriert werden soll. Zuletzt soll der Aspekt eines rollenspezifischen Berechtigungssystems untersucht werden.

Im Zusammenhang mit Integration spielt dies eine wichtige Rolle, da mögliche Kooperationspartner zur besseren Eingliederung in das eigene System Rechte zugewiesen bekommen sollten.

4.1.4.1 Verlauf der Befragung

In diesem Abschnitt gab es teilweise sehr unterschiedliche Antworten auf die gestellten Fragen. In einigen Fällen wurden sie ausgelassen aufgrund von mangelnden Kenntnissen. Das traf zum einen auf die Frage nach der Anbindung an ein Fremdsystem auf Applikations- und Datenebene zu und zum anderen auf die Frage nach verwendeten Standards und dem evtl. damit verbundenen Einsatz von Konvertern (da diese Frage mit der ersten Frage des Technologie/Systemextern verschmolzen wurde, vgl. Kapitel 4.2.2).

4.1.4.2 Zusammenfassung der Ergebnisse

- Der direkte Zugang zu Datenbanken findet über einfache Schnittstellen statt, Web Services dienen im Applikationsbereich als Zugriffswerkzeug.
- 84 Prozent der Systeme haben im primären Bereich interne funktionale Abhängigkeiten, es herrscht ein unattraktives Aufwand-Nutzen Verhältnis zur Neuentwicklung.
- Ein Berechtigungskonzept ist in jedem System umgesetzt.

4.1.4.3 Auswertung

Anbindung an Fremdsysteme

Insgesamt präsentieren sich viele verschiedene Möglichkeiten der Systemintegration. Oft ist es nicht nur eine Schnittstellenart, die Systeme zu Verfügung stellen, sondern eine ganze Reihe an Zugangsarten. Auf Applikationsebene werden dabei sehr oft Web Services eingesetzt, um Funktionalitäten für Nutzer bereitzustellen, auf Datenbankebene eher selten (vgl. Abbildung 4.16). Nur Godyo, Oracle und Intraprend ermöglichen einen Datenbankzugriff via Web Services. Ebenfalls kaum genannt ist XML als Dokumentensprache im Bereich Datenbank-Kommunikation, Vertreter sind hier die Systeme von nGroup, Informing, Intraprend und Microsoft (Navision).

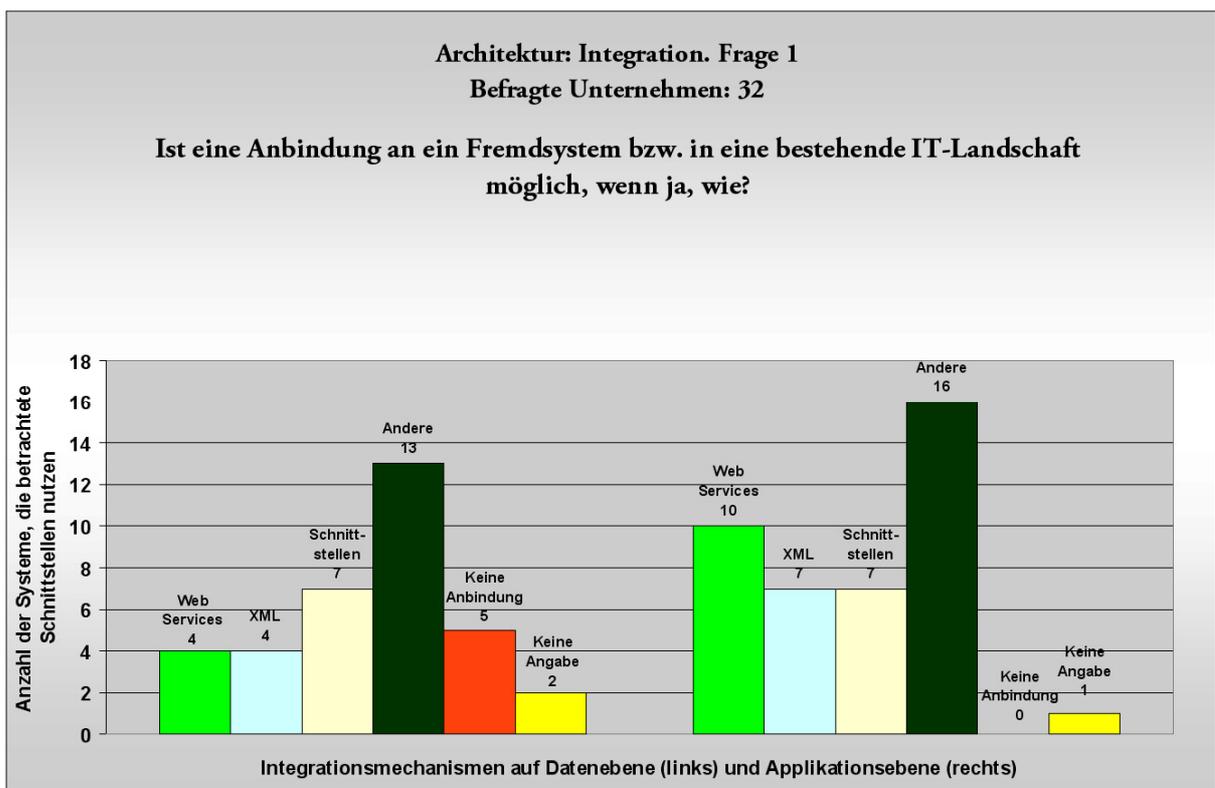


Abbildung 4.16 Anbindungsmöglichkeiten an Fremdsysteme auf Applikations- bzw. Datenbankebene

Im Applikationsbereich hingegen findet man XML durch den Einsatz in den Web Service Komponenten deutlich häufiger vor. Die allgemein angegebenen Schnittstellen wurden nicht näher konkretisiert, sondern als gegeben angenommen. Interessant sind die sehr verschiedenen Angaben zur Integration, die weder XML noch Web Services umfassten.

Im Folgenden werden sie aufgelistet (vgl. Abbildung 4.17): Dabei reichen die Angaben in der Applikationsebene von einer Web-Shop Anbindung bis zu SOA als Integrationswerkzeug oder eine einfache Textdatei im CSV-Format (Comma Separated Value).

| <u><i>Applikationsebene</i></u> | <u><i>Datenbankebene</i></u> |
|-----------------------------------|-----------------------------------|
| Eigenentwicklung | SQL |
| Web Shop | Pearl-Schnittstellen |
| Übergabedateien | offen dokumentiert |
| BAPI/WSAPI | EAI Framework |
| EAI Framework | Direktanbindung |
| Java Access | BDI |
| BPI | MyODBC |
| SOA | Protokolladapter |
| CSV | BAPI |
| ESB | JDBC |
| Batch-Interfaces | Application Integration Framework |
| JMS | |
| HTTP- | |
| Synchronisationsdienste | |
| API | |
| Application Integration Framework | |

Abbildung 4.17 Sehr verschiedene Angaben zu Integrationsmöglichkeiten

Greenax benutzt den selbst entwickelten Business Process Integrator (BPI) (vgl. Kapitel A.1) als Funktionsintermediär, GuS ist in der Lage, Business Process Application Interfaces (BAPIs) in Zusammenarbeit mit SAP-Systemen anzusteuern. Es fällt auf, dass für die Interviewpartner als Schnittstelle nicht nur eine Software (z.B. Web-Shop) oder ein Dokumentenstandard (z.B. CSV) in Frage kommt, sondern auch Konzepte wie die SOA oder Frameworks.

Ähnlich verhält es sich auf Datenbankebene. Ob es sich um SQL-Schnittstellen, EAI-Frameworks oder der Business Data Integrator (BDI) (vgl. Kapitel A.1) von Greenax handelt, es gibt keine allgemein verbindlichen oder anerkannten Methoden, Funktionen oder Dokumentenstandards, die allen Anbietern zueigen sind. Ein Anbieter beschreibt die Situation als Mittler zwischen den unzähligen Standards folgendermaßen: „[Wir] verstehen uns als Spinne im Netz.“ (EG2)

Tendenziell ist ein direkter Zugang über einfache Schnittstellen im Datenbankbereich und Web Services im Applikationsbereich erkennbar. Die Verbreitung von XML und, wie man später noch sehen wird, das Verstehen von EDI-Standards als Grundformat scheint die einzige gemeinsame Grundlage zu sein, die bei fast allen Systemen vorhanden ist.

Abhängigkeiten von Modulen untereinander

Im Vergleich zu der bereits weiter fortgeschrittenen Service-Orientierung der Anbieter ist eine lose Kopplung der Module in den ERP-Systemen noch sehr selten verbreitet (vgl. Abbildung 4.18). Achtzehn der befragten Unternehmen haben eine enge Integration zwischen ihren einzelnen Funktionen, die einen modularen Aufbau nicht zulässt. Miclas z.B. bietet nur ein Modul an, in dem alle Funktionalitäten untergebracht sind.

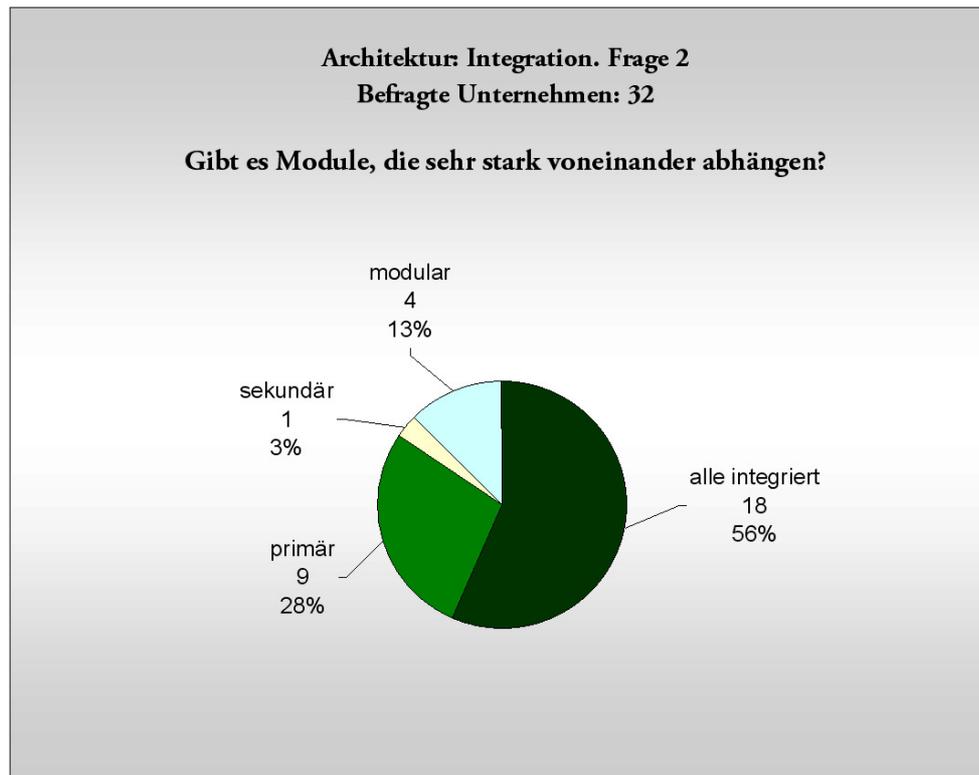


Abbildung 4.18 Nur etwa 10 Prozent der Systemlandschaften sind modular aufgebaut

Im primären Bereich, also bei den Aktivitäten, die das Kerngeschäft des Unternehmens unterstützen, ist es etwa ein Drittel. Das heißt, bei mehr als 80 Prozent der Anbieter ist eine lose gekoppelte Funktionsstruktur nicht denkbar. Das ist, so wurde es in den Interviews ein paar Mal deutlich, auch gar nicht erwünscht: „Wir bieten eine Lösung aus einer Hand.“ (SS1)

Das beinhaltet auch, mögliche Konkurrenz mit ihren Lösungen auf Abstand zu halten bzw. deren Integration in die bestehende IT-Landschaft zu erschweren, wenn nicht sogar unmöglich zu machen, ohne einen großen Aufwand. Abacus gibt an, dass im sekundären Bereich die Finanzbuchhaltung und die Debitorenbuchhaltung stark voneinander abhängen, der Rest jedoch modular aufgebaut ist. Abacus ist somit der einzige Anbieter, der im sekundären Bereich Abhängigkeiten beinhaltet.

Vier Anbieter, Ramco, IFS, Alea und Oracle, sind nach eigenen Angaben rein modular ausgerichtet. Trotzdem sollte man tendenziell auch noch Abacus hinzuzählen, da dort nur Funktionalitäten aus dem Bereich „Finanz- und Rechnungswesen“ stark voneinander abhängen. Alle Anbieter mit einer modularen Funktionsstruktur basieren ebenfalls auf der SOA. Sie kombinieren einen komponenten- und service-orientierten Ansatz.

Berechtigungskonzept

Hier haben alle ein vollwertiges Berechtigungskonzept anzubieten, welches zwischen Benutzer und Rolle unterscheiden und bei den meisten Anbietern bis hinunter auf die Feldebene umgesetzt werden kann. Nur drei Anbieter erlauben keinen direkten Zugriff auf die Datenbank, das sind GuS, nGroup und Oracle. Hier reicht die Berechtigung nur bis auf die

Funktionsebene. Somit sollten in diesem Bereich hinsichtlich der Anbindung von Fremdsystemen und der internen Rollenverteilung keine Probleme entstehen.

4.1.4.4 Fazit

Die Art der Schnittstellen zu anderen Systemen ist ebenso vielfältig wie die Anzahl der Systeme selbst. Ein Drittel der Befragten setzt Web Services ein, allerdings auch nur teilweise oder für ganz bestimmte Anwendungen. Trotzdem ist die Anbindung an ein anderes System ohne großen Aufwand kaum zu erledigen.

Angesichts der vielen verschiedenen so genannten Standards, die sich im Umlauf befinden, ist es nicht verwunderlich, dass sich die meisten Anbieter auf die Kommunikationsformen in ihrem Zielsegment konzentrieren und andere im Hinblick auf eventuelle Integrationsunterstützung erst einmal vernachlässigen. Zumal eine zu leichte Integration im Hinblick auf Systemgeschlossenheit nicht erwünscht zu sein scheint. Eine Anbindung an Fremdsysteme ist oft nicht vorgesehen. Das heißt, der Anspruch an eine Komplettlösung, die den Kunden bindet, ist höher als der Anspruch an ein vereinfachtes Collaborative Business.

Das spiegelt sich auch in der Eigenschaft der Modularität von Komplettlösungen wider. Das Aufwand-Nutzen Verhältnis, ein komponenten-orientiertes System zu schaffen, ist oft unattraktiv, da der Hersteller daran interessiert ist, sein ganzes System zu vertreiben und nicht nur Teile davon. 84 Prozent der untersuchten Systeme sind zumindest im primären Bereich und damit im Kerngeschäft intern miteinander verwoben.

Die überall durchgängig umgesetzten Berechtigungskonzepte, die primär für die interne Anwenderorganisation beim Kunden gedacht sind, bieten zumindest so als Zugangportal für externe Nutzer ein Potenzial zur Anbindung.

4.2 Technologie

Der Abschnitt Technologie beleuchtet die Werkzeuge, mit denen die im vorigen Kapitel 3.2.2 „Architektur“ vorgestellten Konzepte und Architekturen umgesetzt werden. Die Gliederung erfolgt dabei in den systeminternen und systemexternen Teil.

4.2.1 Systemintern

Systeminterne Aspekte betreffen technologische Grundlagen wie das Betriebssystem oder genutzte Datenbanken. Ebenso wichtig für die Fähigkeiten und Eigenschaften eines betrieblichen Anwendungssystems sind Programmiersprache und die Entwicklungsumgebung, in der Code geplant und erzeugt wird. Da die meisten der befragten ERP-Systeme proprietäre Software darstellen, wird in der letzten Frage der Einsatz von Open Source Werkzeugen in der Entwicklung und auch in der Funktionalität als Integration untersucht.

4.2.1.1 Verlauf der Befragung

Hier waren die Angaben im Vergleich zum vorigen Kapitel 4.1.4 trotz der ebenso technologischen Fragen vollständig. Einige konnten leider auf die Frage, warum eine Programmiersprache benutzt wurde, keine oder nur unvollständige Antworten liefern, da die Konzeption zu lange zurücklag.

4.2.1.2 Zusammenfassung der Ergebnisse

- Windows ist als Betriebssystem breit unterstützt, allerdings sind nur 25 Prozent der Systeme rein Windows basiert.
- SQL bewegt sich vor Oracle vor DB/2 in der Datenbankverbreitung.
- Knapp 50 Prozent der Systeme verwenden unter anderem Java als Programmiersprache, sieben C/C++, sechs CSharp.
- Eclipse ist als Entwicklungsumgebung sehr beliebt (10 von 32), Visual Studio hat 6 Anwender, es gibt viele Eigenentwicklungen.
- Open Source Programme zur Erstellung bzw. internen Unterstützung von ERP-Systemen sind von zwei Dritteln gern gesehen.

4.2.1.3 Auswertung

Betriebssysteme

Die Frage zielte in erster Linie auf Server-Betriebssysteme ab. Die unterschiedlichen Versionen des Betriebssystems, z.B. Windows XP oder Windows Server 2003, wurden bei der Befragung vernachlässigt. Windows ist erwartungsgemäß das meist unterstützte Betriebssystem, das 27 Anbieter nutzen können (vgl. Abbildung 4.19).

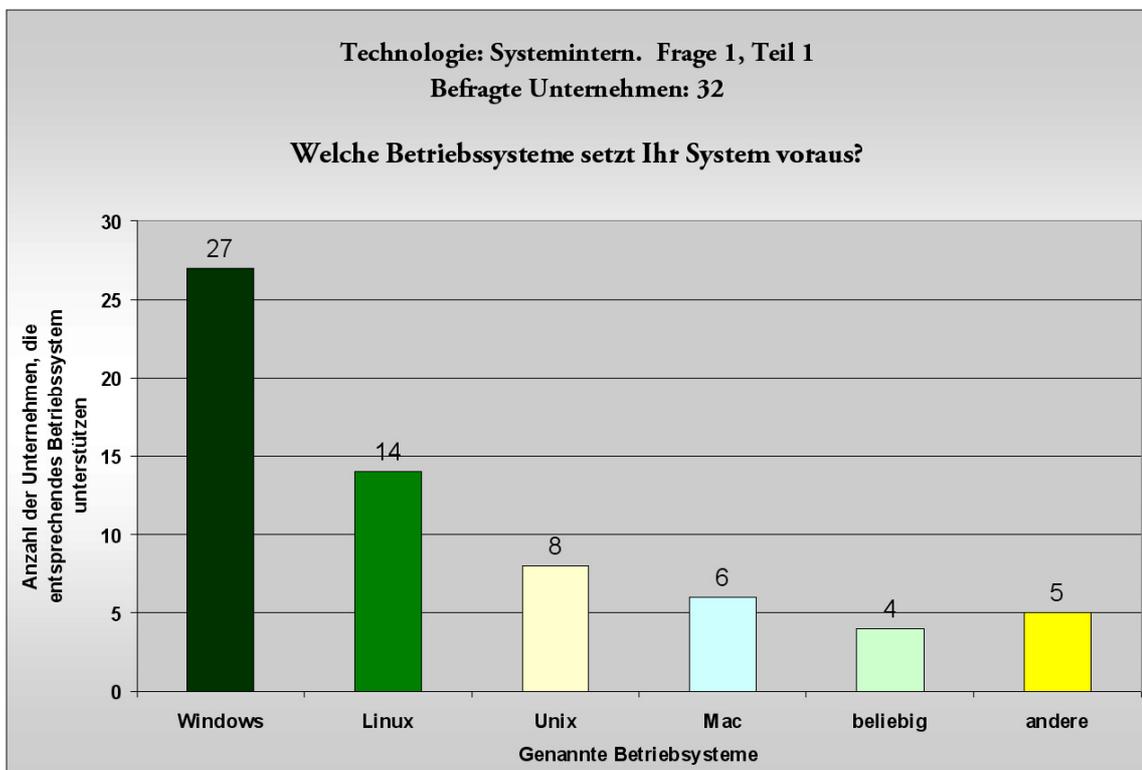


Abbildung 4.19 Möglichkeiten für Betriebssysteme

Dies spiegelt auch die aktuelle Studie wider [Discherl, 2007], die Windows im Server-Bereich mit einem deutlichen Vorsprung gegenüber Linux bzw. Unix sieht. Neun der 27 Systeme sind allein auf Windows-Plattformen ausgerichtet, darunter natürlich Microsoft selbst. Aber auch im Client-Bereich ist Windows stark vertreten. Zwingende Voraussetzung ist Windows für Clients jedoch nur bei Godyo, IFS und Jentech.

Linux unterstützen nur noch 14 Systeme, Unix und Macintosh werden immerhin noch von acht bzw. sechs Anbietern miteinbezogen. Die freie Wahl der Betriebssysteme scheint immer noch eine sehr seltene Option für Kunden zu sein. Vier Anbieter geben den Kunden diese Möglichkeit. IAS und SHD benötigen nur eine Java-lauffähige Plattform als Grundvoraussetzung, SAP unterstützt nach eigenen Angaben jedes Betriebssystem und Intraprend braucht aufgrund seines HTML-basierten Ansatzes nur einen Browser. Die restlichen Softwareplattformen, die genannt wurden, sind i5-OS von IBM (Alea), AS400 von IBM (PSI), Solaris von Sun Microsystems und AIX von IBM (Oracle).

Datenbanken

Es zeigt sich, dass bei den ERP-Anbietern die bewährten relationalen Datenbanken weit verbreitet sind. Als Datenbank hat SQL (hierbei wird nicht unterschieden zwischen MSSQL und dem frei erhältlichen MySQL) die größte Verbreitung mit 20 Systemen, Oracle Datenbanken liegen auf Platz 2, DB/2 von IBM ist noch mit 12 Umsetzungen vertreten (vgl. Abbildung 4.20).

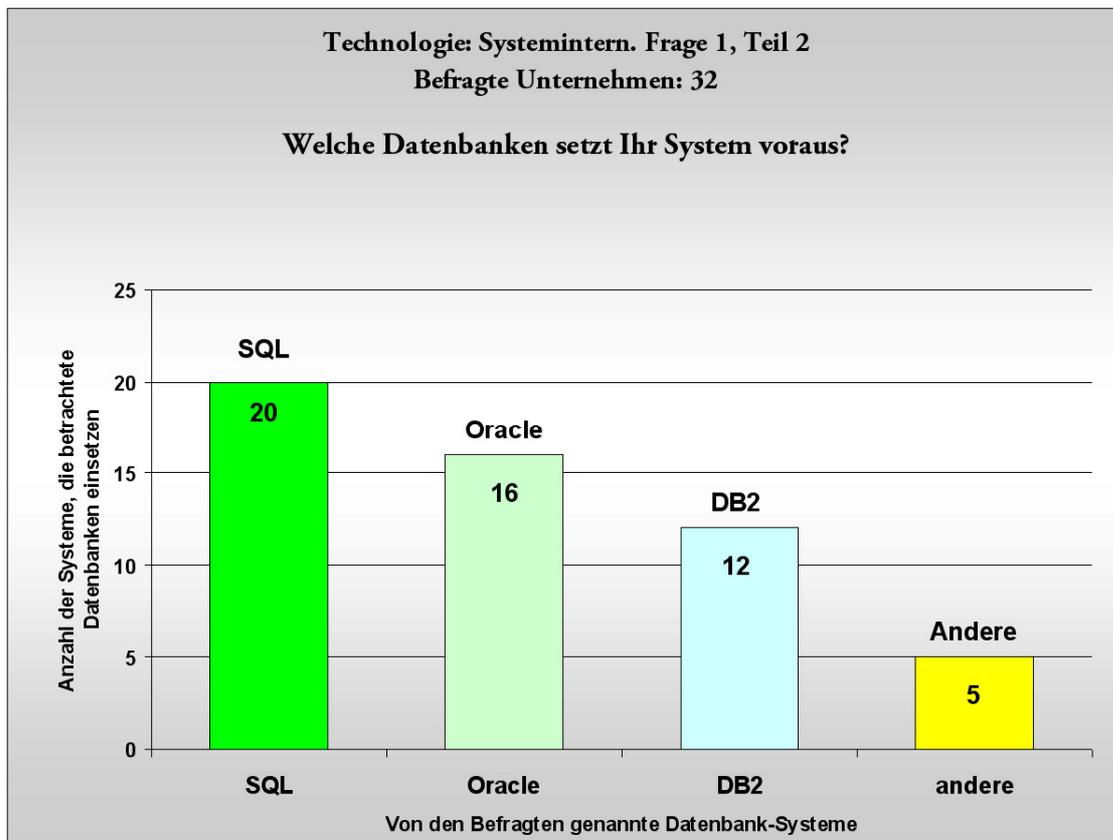


Abbildung 4.20

Relationale Datenbanken als Grundlage von ERP-Systemen

Die restlichen fünf Angaben beziehen sich auf Caché, eine postrelationale Datenbank (Intraprend), Filemaker, eine relationale Datenbank (Günther-BS), 4D, ebenfalls eine relationale Datenbank (Topix), Firebird, eine relationale Open Source Datenbank (SynERP) und generell JDBC-fähige Datenbanken (eBootis).

Diejenigen Systeme, die nur Microsoft als Grundlage benutzen, verwenden in der Regel auch MSSQL als Datenbank, nur Godyo und nGroup verwenden noch zusätzlich eine Oracle Lösung.

Programmiersprache

Die meist benutzte Programmiersprache ist Java (vgl. Abbildung 4.21). Allerdings sind nur sechs der dreizehn Systeme nach eigenen Angaben komplett in Java geschrieben worden. Bei dem Rest wird Java entweder neben einer anderen Programmiersprache oder nur für die neu entwickelten Softwareteile genutzt, während die anderen in einer älteren Sprache weiter gepflegt werden. Zudem wird von einem Anbieter (eBootis) die Applikationen in Java und die Clients in C/C++ geschrieben.

Der Hauptgrund für die Nutzung von Java war dessen Plattformunabhängigkeit. Die zweite Begründung liegt eher im argumentativen Bereich: es gilt als zukunftsweisend und vorausschauend. Obwohl Java bereits fast 20 Jahre alt ist, hat es bei den Befragten immer noch einen hohen Stellenwert. Vor allem in der Applikationsentwicklung gilt es bei einigen als performante Sprache, welche jedoch im Client-Bereich noch Defizite aufweist: „Damals (2000) war Java für die Clients noch zu langsam und unflexibel.“ (WE2)

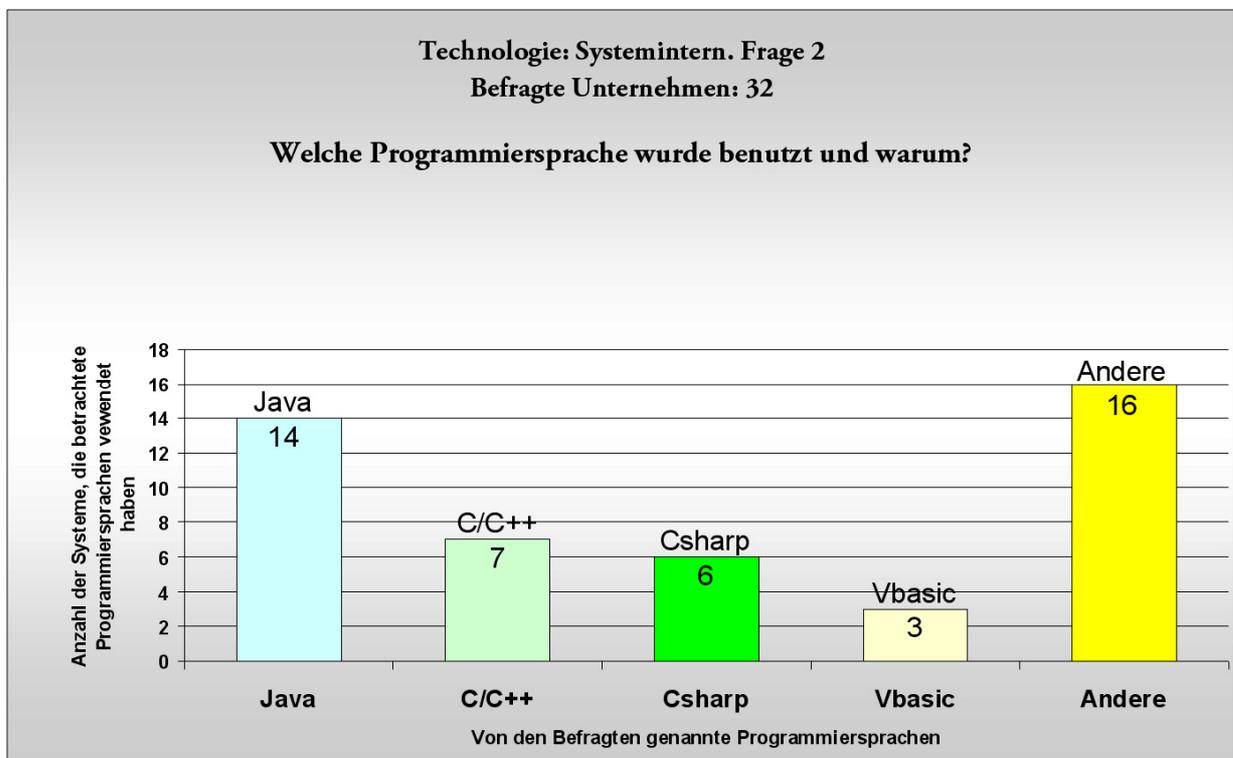


Abbildung 4.21 Java bei mehr als 50 Prozent im Einsatz

C/C++ und CSharp haben eine etwa gleichstarke Vertretung bei den Anbietern. Diese sind besonders bei älteren Systemen zu finden und existieren fast immer neben einer weiteren Sprache. Der Einsatz von CSharp wurde damit begründet, dass diese relativ junge Programmiersprache aufgrund ihrer Modernität, dem guten Erfolgspotenzial und der technologischen Basis von .NET ausgesucht wurde. Daher ist CSharp auch auf allen .NET Plattformen vertreten.

Unter den anderen Sprachen (vgl. Abbildung 4.22) sind unter anderem RPG Artefakte, Cobol, Objective C, PL/SQL, IntarSkript (Eigenentwicklung von Seat-1), eine Eigenentwicklung auf Basis von Visual Objects (AMS), ABAP, die eigenentwickelte Programmiersprache von SAP oder C/AL, eine eigenentwickelte Sprache für Microsoft (Navision).



Abbildung 4.22 Sechzehn unterschiedliche Angaben zur Programmiersprache

Bis auf wenige alte Sprachen wie z.B. Cobol oder RPG haben fast alle einen objekt-orientierten Charakter. Auffällig sind die teilweise noch alten Programmfragmente, die weiterhin einen Teil mancher Systeme bilden. Lawson besitzt z.B. noch ältere RPG-Artefakte, Hilmer schreibt seine Software in Cobol.

Entwicklungsumgebung

Wie auch bei den Programmiersprachen zeigt sich hier eine sehr große Auswahl an unterschiedlichen Produkten, die Verwendung finden. Bei den Entwicklungsumgebungen führt klar Eclipse gefolgt von Visual Studio. Zusammen mit IntelliJ fordert Java hier etwa ein Drittel aller Entwicklungsumgebungen (vgl. Abbildung 4.23). Ein Drittel der Anbieter hat eine eigene

Umgebung geschaffen. Der Rest greift auf Produkte wie z.B. eine Runtime von Acucorp für Cobol (Hilmer Software) oder den ProjectBuilder und Notepad++ zurück (Seat-1).

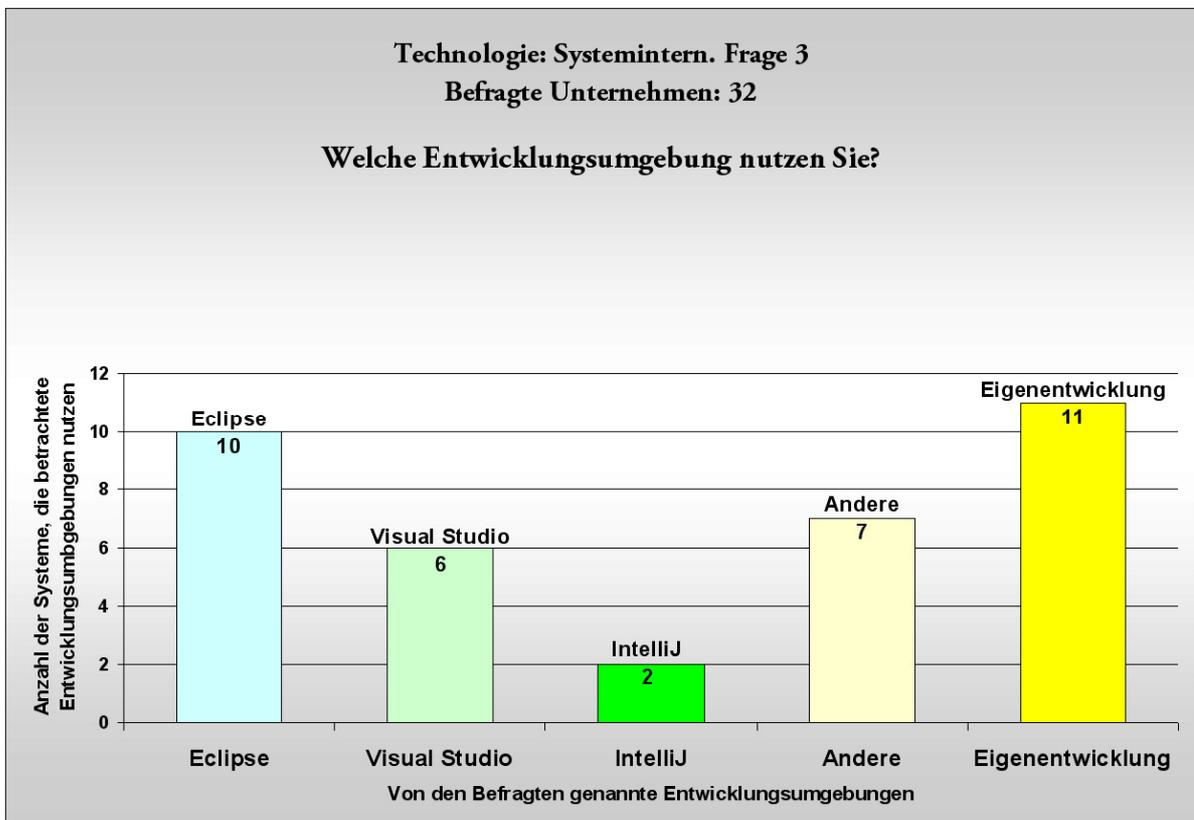


Abbildung 4.23 Eclipse und IntelliJ als Java-Entwicklungsumgebung weit verbreitet

Java ist bei mehr als einem Drittel der Befragten im Einsatz oder sogar Hauptsprache geworden. Das Argument der Plattformunabhängigkeit und die breit gefächerte Community scheinen Anlass zu geben, die ERP-Systeme neuerer Generation direkt oder zumindest die neuen Applikationen in Java zu verfassen. Hinzu kommt durch den Einsatz von Eclipse als OS-Programm eine stärkere Orientierung hin zu freien Entwicklungswerkzeugen. Visual Studio ist jedoch als proprietäre Entwicklungsumgebung nicht zu vernachlässigen. Im .NET Bereich ist sie unangefochten der stärkste Vertreter.

Open Source (OS) als Werkzeug

Zwei Drittel der Befragten sehen OS, sei es als Entwicklungsumgebung (z.B. Eclipse), Servergrundlage (z.B. JBoss) oder als Textverarbeitung (z.B. openoffice), als ein wichtiges Hilfsmittel in der Erstellung und im Betrieb ihres ERP-Systems. Das andere Drittel setzt interessanter Weise vollständig auf Windows (vgl. Abbildung 4.24).

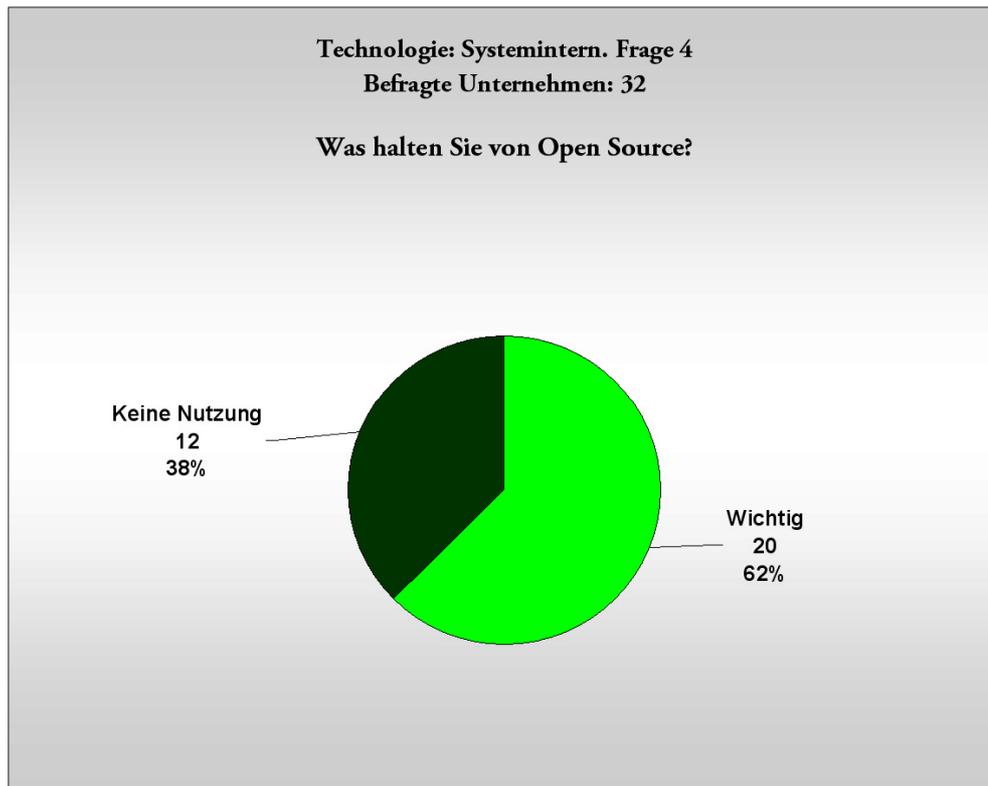


Abbildung 4.24 Zwei Drittel sehen OS als wichtiges Entwicklungswerkzeug an

Hier scheinen sich zwei Lager zu bilden. Auf der einen Seite gibt es die Microsoft Anhänger, die keinerlei OS einsetzen, und auf der anderen Seite die OS Nutzer, die aber nicht zwangsläufig Java einsetzen müssen. Von diesen entwickeln sogar einige aktiv an Projekten mit, da sie ihnen ebenso zum Vorteil gereichen wie anderen auch (z.B. PSIpenta, SHD oder Oracle). OS ist also, was die Nützlichkeit für eine Softwareentwicklung betrifft, durchaus ein gern gesehenes und oft genutztes Instrument zur Erstellung proprietärer Software.

4.2.1.4 Fazit

Es gibt eine deutlich größere Menge an Systemen, die plattformunabhängig sind als Microsoft oder sogar Linux gebundene Produkte. Zusammen mit der starken Verbreitung von Java und der positiven Grundeinstellung zu OS Produkten deutet dies auf eine Ausrichtung hin zu portabler und evtl. sogar mobiler Software. Informationen können heute schon auf kleine, portable Displays, Handhelds, PDAs und ähnliches gebracht werden. Man benötigt nur z.B. einen Browser und/oder Java.

„OS lässt sich nicht mehr aus heutigen Systemen wegdenken.“ (GA4), sagte ein Interviewpartner. Diese Einstellung ist bei der Mehrheit der Anbieter verbreitet. Ob Programmiersprache oder Entwicklungsumgebung, die Nutzung von OS-Software spielt in der Erstellung von betrieblichen Anwendungssystemen eine wichtige Rolle. Dies gilt nicht für Softwarehäuser, die ihre Strategie auf Microsoft ausgerichtet haben:

„Der Einsatz von OS Werkzeugen steht momentan nicht zur Diskussion.“ (PA1)

.NET und die damit verbundenen Produkte decken den kompletten Werkzeugbedarf für die Programmierer ab. So haben sich zwei Lager gebildet: Es gibt diejenigen Anbieter, die Open Source freundlich gesinnt gegenüberstehen und sich teilweise sogar aktiv an der Weiterentwicklung beteiligen und diejenigen Softwarehäuser, die lieber auf einer komplett integrierten Infrastruktur arbeiten und diese ausgiebig nutzen.

Der größte Teil der Clients im KMU Markt basiert immer noch auf Windows. Dies gilt auch für den Serverbereich. Fast alle Anbieter wissen um die große Verbreitung von Microsoft-Produkten und bieten eine entsprechende Unterstützung ihrer Systeme an, jedoch nicht alle auf exklusiver Basis. Die Freiheit, eine Plattform wählen zu dürfen, scheint kundenorientierter zu sein als die Festlegung auf ein bestimmtes Produkt. Deshalb hat Java den größten Anteil an den genutzten Programmiersprachen: „Java ist die beste Sprache für die Plattformunabhängigkeit.“ (GA3)

4.2.2 Systemextern

Der zweite und letzte Abschnitt der Technologie begutachtet vor allem Dokumentenstandards, durch deren Einsatz eine Anbindung an fremde Systeme erleichtert werden kann. Als zweites Integrationswerkzeug werden außerdem Web Services im Einsatz und am Ende die im Browsereinsatz verwendeten Hilfsmittel betrachtet.

4.2.2.1 Verlauf der Befragung

In diesem Abschnitt wurde vor allem bei der ersten Frage nach den Dokumentenstandards eher verhalten geantwortet bzw. die gängigen Standards wie XML, EDI und Branchenspezifika genannt. EDI schien dabei gleichgesetzt zu werden mit EDIFACT, denn oft wurde auf die Frage nach dem Standard EDI geantwortet, obwohl EDI den generellen Austausch meint und EDIFACT einen EDI-Standard. Auf nähere Nachfrage wusste man teilweise nicht genau, welche Dokumente genau ausgetauscht werden könnten, bzw. zählte eine sehr viel größere Menge von abweichenden Nebenstandards auf. Da die Fülle an Austauschmöglichkeiten einfach zu groß ist, als dass man sie alle hier vollständig aufführen könnte, werden nur die häufigsten Nennungen berücksichtigt. Für detaillierte Informationen wird auf den Anhang A.1 verwiesen.

4.2.2.2 Zusammenfassung der Ergebnisse

- Neben EDIFACT und branchenspezifischen Transaktionsstandards wird immer XML angegeben.
- Web Services haben eine große Verbreitung, vor allem in der Funktionsintegration.
- Die Browsertechnologie wird als Thin-Client Basis (9) leicht vor Citrix als Terminallösung (7) eingesetzt.

4.2.2.3 Auswertung

Dokumentenstandards

XML, EDIFACT und je nach Branche spezifische Standards wie z.B. Odette, VDA etc. sind bei jedem der Befragten vertreten. Nur Hilmer nutzt als einziger Anbieter keinerlei XML als Grundlage für einen Dokumentenstandard.

EDIFACT hat sich somit seit 1987 endgültig in allen ERP-Systemen als ein allgemein anerkannter Transaktionsstandard etabliert. Ebenso ist das deutlich jüngere XML als Grundlage für Geschäftsdokumente inzwischen überall verbreitet, wobei alle bis auf einen Anbieter nur XML selbst im Rahmen von Dokumentenstandards benennen.

eBootis hat als einziges Softwarehaus explizit openTRANS als Dokumentenstandard angegeben. Wenn sie eingesetzt werden, dann sind es meist nur EDI-Konverter, die Erwähnung finden. Neben diesen eher allgemein gehaltenen Aussagen haben einige Anbieter noch folgendes angegeben: nGroup betreibt einen Business Server, der die Dokumente verwalten und konvertieren kann, SAP nutzt die eigenentwickelte XI (Exchange Infrastructure) zum Dokumentenaustausch und Microsoft bedient sich des hauseigenen BizTalk-Servers. Trotzdem wurde während der Interviews versichert, dass alle wichtigen Kommunikationsstandards unterstützt werden könnten.

Web Services

Web Services haben in den betrieblichen Anwendungssystemen inzwischen eine große Verbreitung gefunden.

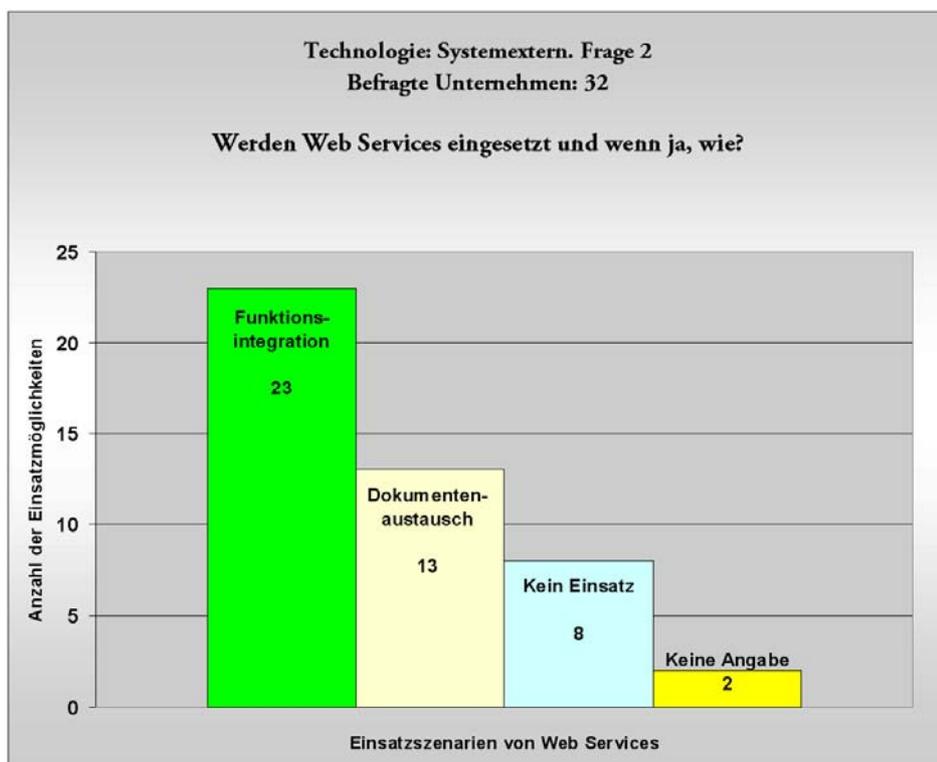


Abbildung 4.25

Web Services als weitflächig genutztes Werkzeug

Das Einsatzgebiet von ihnen ist in diesem Zusammenhang eindeutig die Funktionsintegration (vgl. Abbildung 4.25). Dienste werden via dem Service angeboten und gekapselt. Immerhin ein Drittel verwendet Web Services zum Dokumentenaustausch, nur acht Anbieter setzen sie überhaupt nicht ein. Microsoft (Navision) wird Web Services erst 2009 umsetzen.

Der Einsatz von Web Services ist nicht bei allen Anbietern sowohl systemintern als auch -extern gewünscht. Bei acht Systemhäusern ist die Bereitstellung ihrer Funktionen nach außen für fremde Anwendungen möglich. Immerhin haben neun der Unternehmen, die einen WS-Einsatz angegeben haben, einen breitflächigen Einsatz verfolgt.

WS scheinen somit als Integrationswerkzeug anerkannt und auch verbreitet zu sein. Die restlichen vier Anbieter (PSIPenta, Sage/Bäurer, Lawson und Abacus) nutzen Web Services nur in einem beschränkten Umfang für ausgewählte Module (z.B. die Anbindung eines Personalmanagement Moduls bei Lawson).

Web Anwendungen

Es scheint, als ob die Performanz und die Sicherheit einer Terminal-Lösung immer noch konkurrenzfähig zu den frei erhältlichen Browsern als Zugangportal ist, Browser als Darstellungsmittel jedoch im Vergleich die Oberhand gewinnen (vgl. Abbildung 4.26). 24 von 32 Unternehmen setzen unter anderem auf die normalen Browserfähigkeiten, wenn es um die Übermittlung und Darstellung der Unternehmensdaten geht. Neun Anbieter bieten im Hinblick auf Web Anwendungen die Zugangsmöglichkeiten exklusiv über den Browser an.

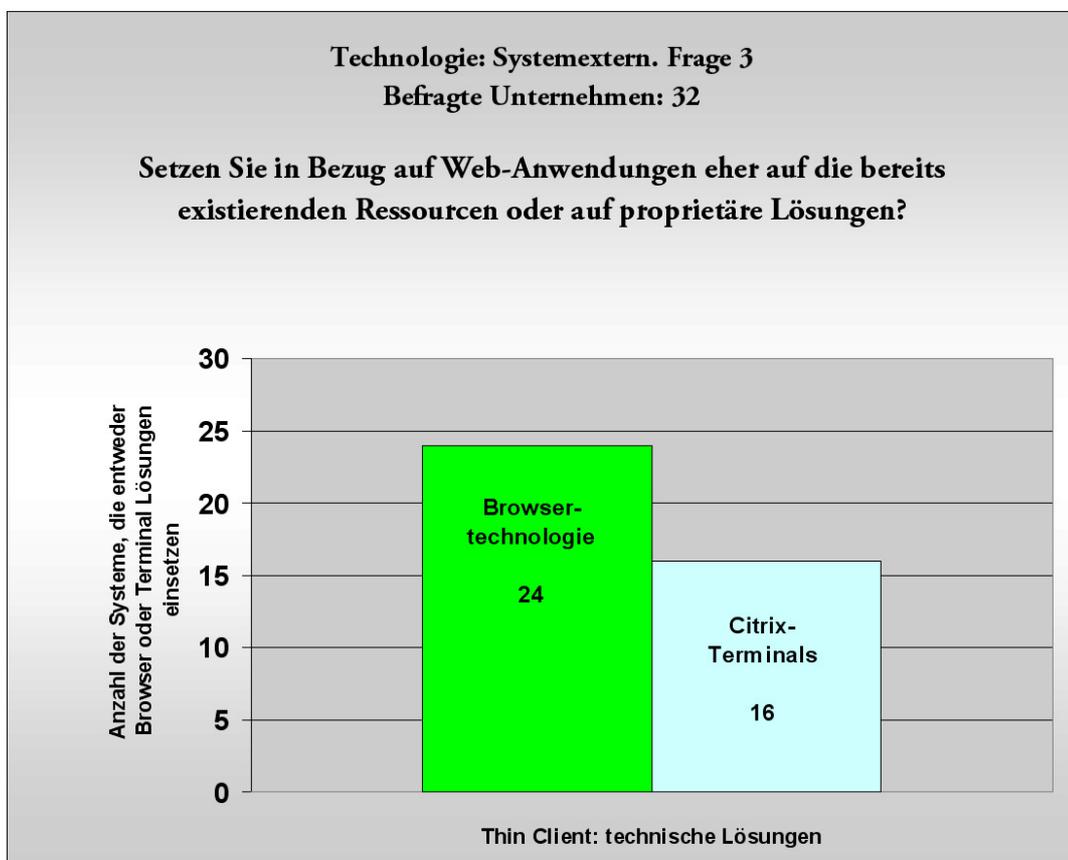


Abbildung 4.26

Klassische Browser öfter im Einsatz als Citrix

„Will man volle Funktionalität haben, so braucht man Citrix“ (BIN3), war die Meinung eines Interviewpartners. Sicherheitsmechanismen, Load-Balancing und viele weitere Services der Terminallösung werden bei der Verrichtung der Anwendungen über das Internet/Intranet von 16 Anbietern genutzt.

Neun Anbieter nutzen Citrix parallel zu klassischen Browsern (z.B. nutzt eBootis Browser für Thin-Clients, Citrix für die Rich-Client Variante). Weitere sieben Systeme verwenden Citrix exklusiv als Terminal Lösung. Darunter sind drei von den vier reinen Fat-Client Anbietern. Nur Godyo setzt auf Browser.

4.2.2.4 Fazit

Im Hinblick auf die Technologien, die für die externe Kommunikation verwendet werden, sind Web Services ein wichtiges und weit verbreitetes Werkzeug, sei es zum Dokumentenaustausch oder zur Integration von Applikationen. Die große Verbreitung von XML als Grundlage für EDI-Standards zeigt klar auf, dass sich diese Metasprache wohl endgültig bei den ERP-Systemen etabliert und eine besondere Rolle eingenommen hat (Zitat aus einem Interview: „XML ist Herzens-Angelegenheit“ (EG3).

Frühere EDI-Konverter und EDI-Infrastrukturen waren sehr kostenintensiv. Web Services in Kombination mit XML bauen auf Standards auf und bieten einen einfachen Mechanismus, Geschäftsdokumente verschicken und Funktionen über Entfernung verfügbar machen zu können. Zusammen mit der großen Anzahl an Systemen, die Browser als Client-Lösung unterstützen, bieten Web Services und XML eine echte Alternative zu den klassischen EDI-Standards.

Interessant ist die immer noch hohe Verbreitung von Terminal Lösungen. Mehrfach wurde in den Interviews das Thema Sicherheit als Grund für den Einsatz benutzt (SAP nutzt Citrix z.B. primär als sicheren, internen Zugang zum System), aber auch eine unzureichende Infrastruktur:

„Aufgrund begrenzter Netzressourcen sieht man keine echte Usability in reinen Web-Anwendungen.“(BP1)

Trotzdem haben Browser im Vergleich zu früheren Systemen eine deutliche Überhand gegenüber Terminal-Lösungen gewonnen.

4.3 Betreibermodell

Der Abschnitt Betreibermodell befasst sich mit den möglichen Einsatzszenarien der betrachteten ERP-Systeme. Die Mandantenfähigkeit wird ebenfalls in diesem Abschnitt beleuchtet, da sie ein wichtiges Hilfsmittel für ein ASP/SaaS Szenario darstellt. Die Vertriebs- und Wartungsmodelle geben einen ersten Hinweis darüber, wie stark die Nähe zum potentiellen Kunden ist und wie weit verbreitet der Auftritt eines Anbieters über ein Partnernetzwerk gestaltet wird.

4.3.1.1 Verlauf der Befragung

Während der Interviews zeigte es sich, dass viele Systeme technisch in der Lage sind, mehrere Betreibermodelle umzusetzen, als es in der betrieblichen Praxis tatsächlich der Fall ist, sei es aus wirtschaftlichen oder konzeptionellen Gründen. ASP und SaaS schienen dabei manchmal als gleichwertig angesehen zu werden bzw. konnten in den Interviews nicht direkt als zwei unterschiedliche Betreibermodelle differenziert werden.

4.3.1.2 Zusammenfassung der Ergebnisse

- In-House ist als klassisches Betreibermodell bei allen vertreten, fast 50 Prozent bieten eine ASP Lösung an, fast ein Drittel beschreibt sich als SaaS-fähig.
- Mandantenfähigkeit ist bei allen gegeben.
- Der Vertrieb zusammen mit Partnern ist die klassische Variante, die Quote der Direktwartung ist sehr hoch.

4.3.1.3 Auswertung

Betreibermodelle

Erwartungsgemäß können alle ERP-Systeme als In-House Lösung betrieben werden. Es gibt keinen der proprietären Anbieter, der sich ganz von dem klassischen Lizenzgeschäft getrennt hat. Interessant war der Anteil der möglichen Hosting-Lösungen (vgl. Abbildung 4.27). Nach Angaben der Hersteller werden mehr als 50 Prozent der Systeme auch als ASP-Lösung betrieben, etwa ein Drittel können als SaaS-Szenario angeboten werden.

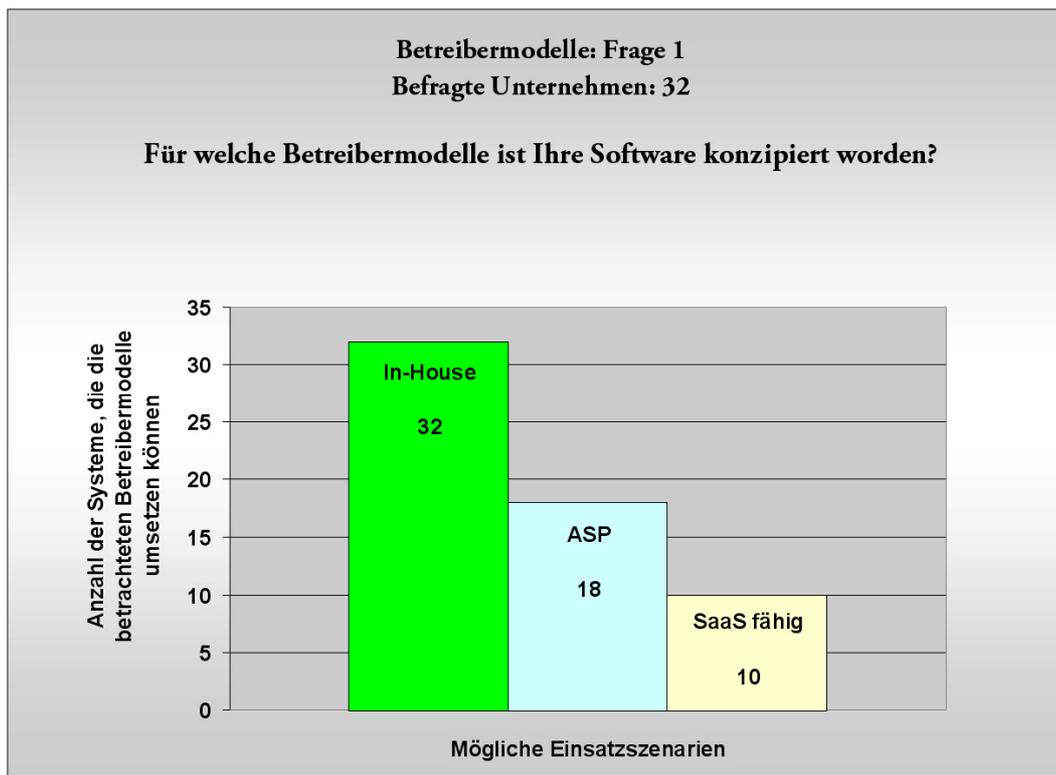


Abbildung 4.27 ASP und SaaS holen auf In-House Lösungen auf

Dabei ist zu beachten, dass es sich um technische Möglichkeiten handelt. Nur drei Anbieter, myfactory, Oracle und Intraprend bieten eine lauffähige Lösung im SaaS Betrieb an. GuS, IAS, Bison, SoftM, Ramco und Abacus nannten dieses Betreibermodell als möglich. Alea gab an, dass ein SaaS-Einsatz geplant sei. Den Grund, warum eine Hosting-Lösung trotz

vorhandener technischer Möglichkeiten nicht angeboten wird, beschreibt ein Anbieter folgendermaßen:

„In-House, ASP und SaaS sind prinzipiell möglich, aber weder ASP noch SaaS sind von Kundenseite her erwünscht.“ (B11)

Eine Service-Orientierung spielt für die möglichen Betreibermodelle In-House und ASP eine sekundäre Rolle. Vier der Systeme mit einem eigenentwickelten Systemkonzept bieten auch ASP-Lösungen an. Dagegen sind service-orientierte Strukturen für ein SaaS-Szenario fast immer Voraussetzung. Nur myFactory ist der einzige Anbieter mit einer eigenen Konzeption und einem reinen SaaS-Angebot.

SaaS ist somit kein neues Konzept mehr und durchaus bei einigen Anbietern prinzipiell möglich umzusetzen, allerdings ist es als Business Modell immer noch im Vergleich zum klassischen Lizenzgeschäft gering vertreten. Mit den Initiativen von SAP, Oracle und anderen Anbietern in Richtung SaaS wird sich dieser Anteil vermutlich bald erhöhen.

Mandantenfähigkeit

Alle Anbieter behaupteten, dass ihr System mandantenfähig sei. Nur ERP 21 und myFactory gaben an, dass es evtl. mehrerer Datenbanken bedürfe, sollte man mehrere Unternehmen z.B. im Rahmen eines Konzerns abbilden wollen. Somit ist die Möglichkeit, mehr als eine Betriebsstätte abbilden zu können, nach eigenen Angaben durchaus bei allen gegeben.

Vertrieb und Support

Eine Kombination aus direktem und Partnervertrieb ist immer noch die bevorzugte Weise, das System dem Kunden verfügbar zu machen.

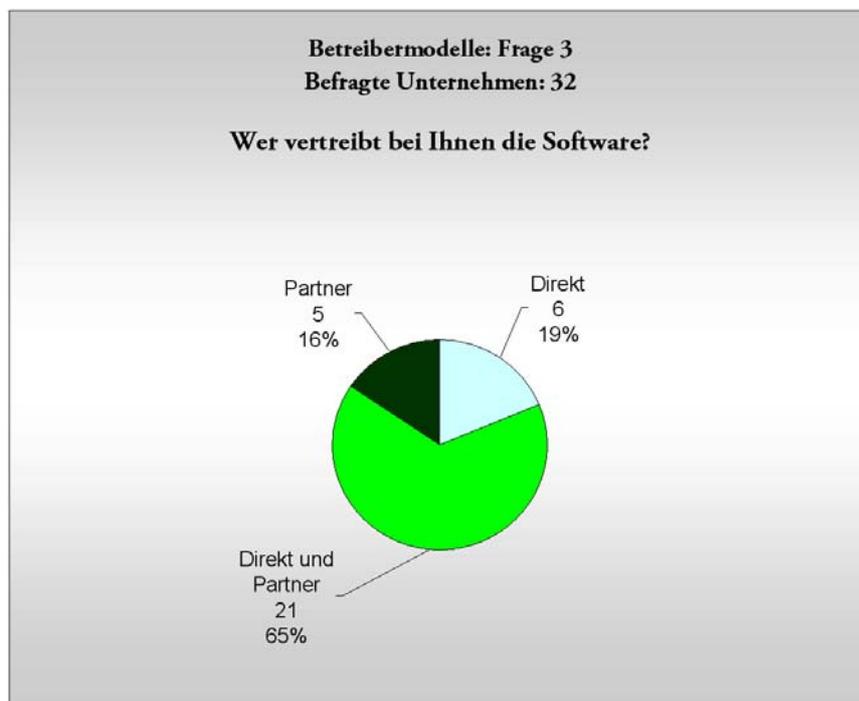


Abbildung 4.28

Kombination aus Direkt- und Partnervertrieb dominant

SHD, Hilmer, Godyo, Topix, Alea und Günther-BS bevorzugen einen Direktvertrieb aus eigenem Haus. Microsoft (Navision), Jentech, Informing, Abacus und nGroup verwenden ausschließlich ihr Partnernetzwerk zum Vertrieb ihrer Software (vgl. Abbildung 4.28).

Partner im Vertrieb eröffnen für die Anbieter zwei Möglichkeiten: Zum einen kann über das Partnernetzwerk eine größere Zahl an Kunden erreicht werden, wenn man selbst keine eigenen Standorte vor Ort besitzt, zum anderen können Partner branchenspezifische oder allgemein kundenindividuelle Wünsche erfüllen, die das Standardsystem nicht bieten kann.

Die oben genannten Anbieter benutzen dementsprechend auch nur eine Direktwartung bzw. Wartung durch den Partner. Einzige Ausnahme bildet Informing, welches in Deutschland einen Partnervertrieb betreibt, die Software allerdings selbst wartet. Diesem unmittelbar erfolgenden Support schließen sich noch Greenax, Ramco, PSIPenta, ERP 21, SAP, Intraprend, B.I.M. und Oracle an (vgl. Abbildung 4.29).

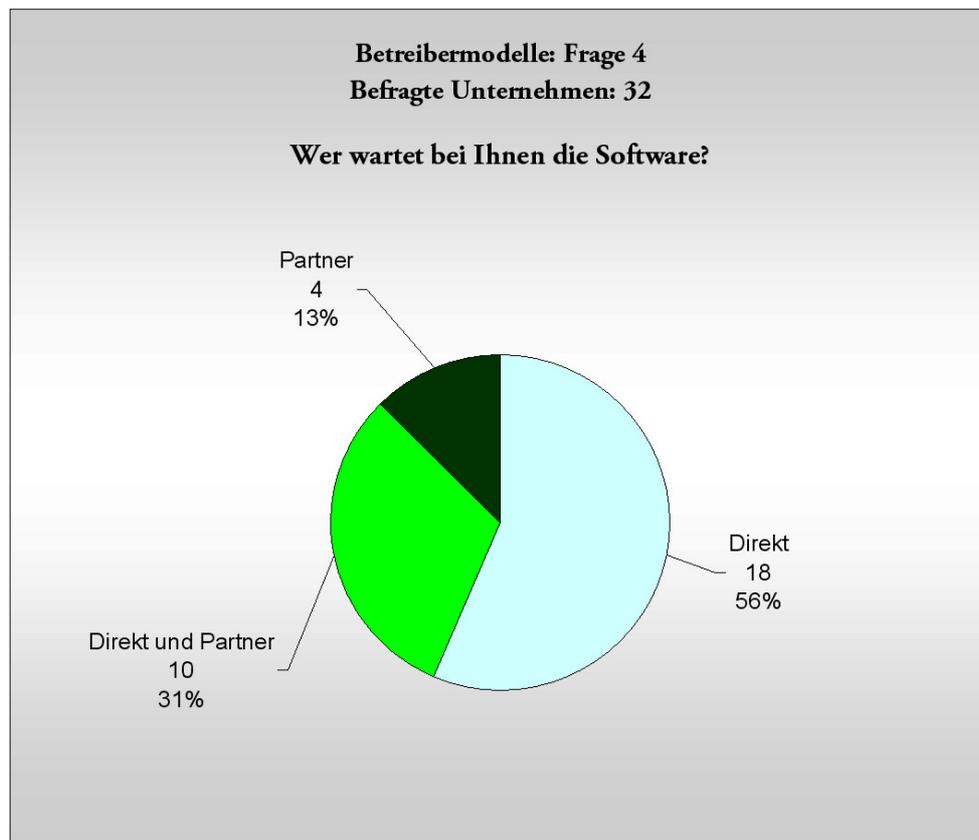


Abbildung 4.29 Direkte Wartung überwiegt Kombinationslösung bzw. reinen Partnervertrieb

Die Anzahl der direkten Wartungen ist somit deutlich höher als die der direkten Vertriebe. Sowohl Kundennähe als auch die direkte Kontrolle über das eigene Produkt scheinen einigen Anbietern in dieser Hinsicht wichtig zu sein.

4.3.1.4 Fazit

In-House Lösungen sind immer noch zusammen mit dem klassischen Lizenzvertrieb weiter verbreitet als gemietete Lösungen. Dabei liegt es nicht an den mangelnden technischen Möglichkeiten vieler Systeme, eine gehostete Lösung umzusetzen, sondern an dem Wunsch der Kunden, es in Anspruch zu nehmen, wie einige Anbieter es formulieren. Standardsoftware bedarf immer noch eines entsprechenden Customizings, dass im Mietmodell nicht oder nur beschränkt umgesetzt werden kann, so einige Interviewpartner. So ist es verständlich, dass ASP im Vergleich zu SaaS eine deutlich größere Verbreitung hat.

Trotzdem scheint SaaS von großen Anbietern wie SAP vorangetrieben zu werden. Für einige der hier befragten Anbieter ist es „eine Entwicklung am Markt vorbei.“ (WS4)

Sie schätzen SaaS als ein Paket aus Standardsoftware ein, das den Anforderungen gerade kleinerer Unternehmen aufgrund von mangelnder Anpassungsfähigkeit nicht gerecht werden kann. In ihren Augen benötigen besonders KMU eine intensive Betreuung bei der Einführung und Nutzung von betrieblichen Anwendungssystemen.

Betrachtet man die Vertriebs- und Support Organisation, so zeigt sich, dass die große Mehrheit eine enge Kundennähe zu suchen scheint bzw. pflegt, zumindest in dem Support-Wesen. Als Anbieter einer Mietsoftware ist der Kontakt zum Kunden vorhanden, wie das Beispiel von myFactory zeigt. Hier findet wie bei vielen anderen eine Kombination aus Direkt- und Partnerbetreuung statt. Oracle und Intraprend nehmen die Wartung sogar direkt vor. Eine mangelnde Kundenbetreuung bei einem Mietmodell ist somit eher unwahrscheinlich als Grund für die geringe Verbreitung.

Auch der Aspekt der Mandantenfähigkeit, der bei jedem umgesetzt ist (auch bei dem SaaS Anbieter), scheint kein Argument gegen eine Mietlösung zu liefern. Es bleibt also die Vermutung, dass es die scheinbar fehlende Individualisierungsmöglichkeit ist, die klassischen Lizenzmodellen immer noch einen Bonus gibt.

4.4 Markt

Im letzten Abschnitt werden die beiden Blickwinkel Kunden (Anwender) und Anbieter auf ein ERP-System betrachtet. Der User-Abschnitt konzentriert sich dabei auf die Anwendersicht auf ein ERP-System, der Markt-Abschnitt auf die ständig im Wandel befindliche Konkurrenz.

4.4.1 Anwender

Die erste Frage zielt darauf ab, eine Klassifikation der ERP-Systeme nach den für sie passenden Kundenzahlen zu erstellen. Aus Anwendersicht ist es außerdem wichtig zu wissen, welche Eigenschaften im Vergleich zu den Konkurrenten eine Entscheidung zugunsten ihres Systems ausmachen. Ebenso wichtig ist der Einfluss, den man inhaltlich auf seinen IT-Partner hat, sowie die Kommunikationswege, die einem potentiellen Kunden offen stehen.

4.4.1.1 Verlauf der Befragung

Branchenzielgruppe und Kundenanzahl sind ausschlaggebend für die unmittelbare Konkurrenz, mit der man um die Aufmerksamkeit der Kunden wirbt. So sind die Vorteile, die ein Nischenanbieter gegenüber seinem universellen Standardsoftware-Konkurrenten sieht, anderer Natur als ein Standardsoftware-Anbieter in derselben Position.

4.4.1.2 Zusammenfassung der Ergebnisse

- Technologischer Vorsprung gilt bei 21 Anbietern als Haupt-Unterscheidungsmerkmal zu ihrer Konkurrenz, Kundennähe nur bei einem Fünftel.
- Der Einfluss der Kunden auf das fertige Produkt ist bei fast allen sehr groß.
- Die Kommunikationswege mit dem Kunden sind nicht entscheidend für den Einfluss auf das Produkt oder die Qualität des Produkts.

4.4.1.3 Auswertung

Anzahl der Kunden

Etwa ein Drittel der befragten Unternehmen haben ein Kundenvolumen, das zwischen 101 und 500 Kunden liegt. ERP21 hat die geringste Kundenzahl mit 7, SAP die größte mit weltweit 44.000 (vgl. Abbildung 4.30). Somit sind von der Verteilung der Kunden her sowohl kleine als auch ganz große Softwarehäuser in dieser Befragung vertreten, so dass angesichts der Abbildung des deutschen ERP-Marktes zwar keine vollständige, dafür aber eine repräsentative Anzahl und Größe von Systemanbietern befragt wurde.

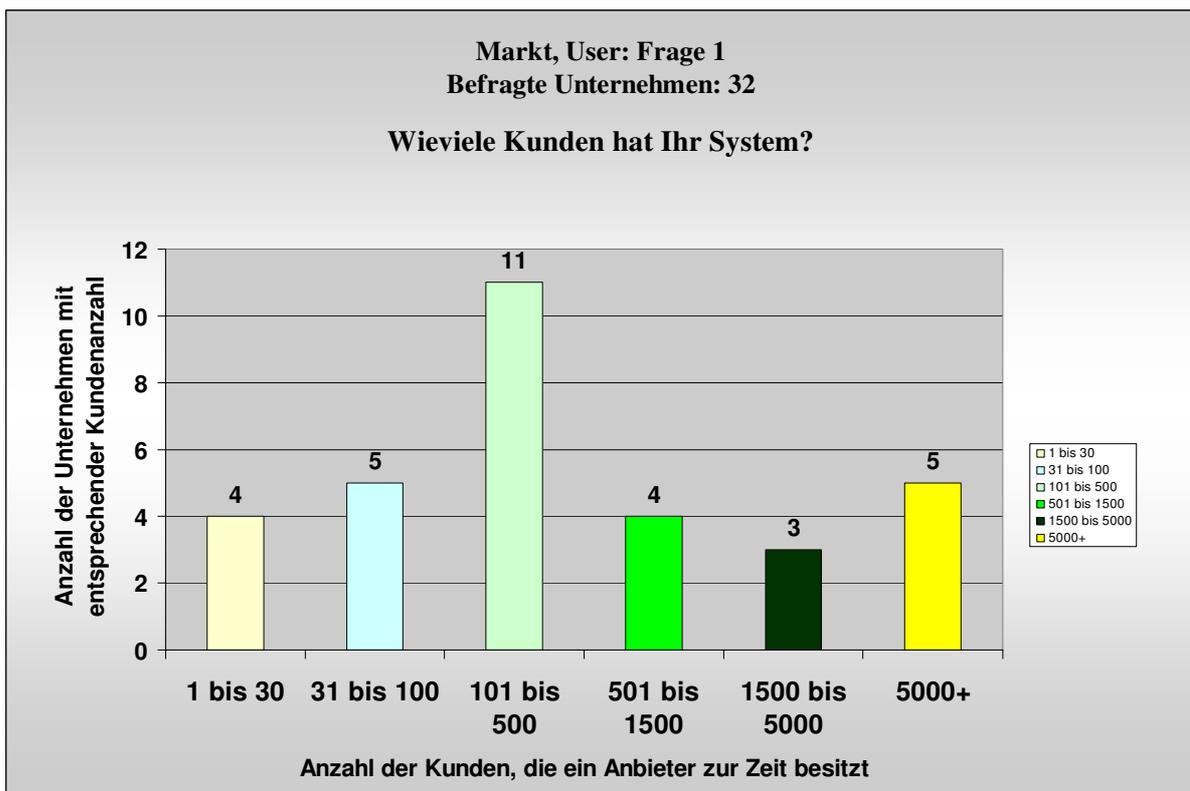


Abbildung 4.30 Abdeckung aller Unternehmensgrößen und -klassen gegeben

Mit zwei Open Source Systemen (Seat-1 und SynERP) ist deren inhaltlicher Beitrag ebenso abgesichert wie die Teilnahme der drei großen Softwarehäuser SAP, Oracle und Microsoft,

deren aktuelle und zukünftige Aktivitäten mit Sicherheit einen nachhaltigen Einfluss auf den gegenwärtigen ERP-Markt haben werden.

Vorteile gegenüber der Konkurrenz

Während der Interviews wurden auf die Frage der Abgrenzung hin zur Konkurrenz mehrfach die in Abbildung 4.31 aufgeführten Merkmale erwähnt:

- Technologischer Vorsprung
- Flexibilität
- Branchenspezialisierung
- Preis/Leistung
- Kundennähe

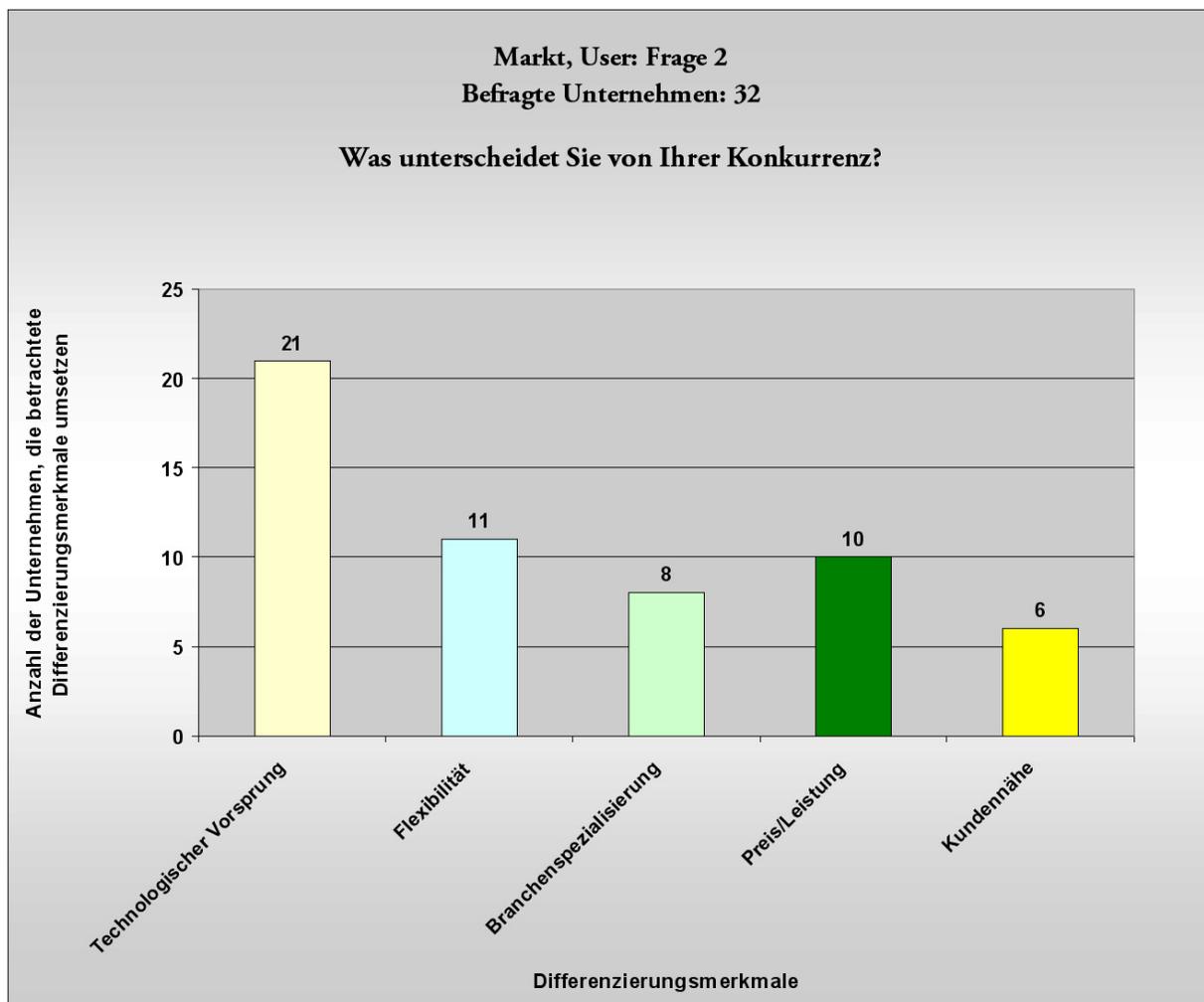


Abbildung 4.31 Technologischer Vorsprung als Unterscheidungsmerkmal Nummer 1

Das Hauptmerkmal, das viele Anbieter zu ihrer Konkurrenz unterscheidet, ist für 21 Unternehmen die eingesetzte Technologie. Modernität ist in allen Anbietergrößen, ob auf MS-Technologie oder J2EE basierend, das entscheidendste Kriterium. Das Spektrum der eigentlichen Nennungen reicht von:

- Spezielle Datenbankstrukturen
- Modul-Eigenentwicklungen
- eigene Systemkonzepte
- innovative Betreibermodelle
- moderne Frameworks
- webbasierte Architekturen
- vollständig integrierte Komplettangebote

bis hin zu

- Programmiersprachen.

Mit einigem Abstand folgen Branchenspezialisierungen, ein gutes Preis/Leistungs-Verhältnis und Flexibilität eines Systems. Die Kundennähe als serviceorientierter Aspekt wird nur von einem Fünftel der Befragten als Unterscheidungsmerkmal angesehen. Branchenspezifische Exzellenz wird verständlicherweise von fast allen Nischenanbietern als hervorragendes Merkmal genannt. Nur Alea (spezialisiert auf Versandhandel) sieht seine Exzellenz in der modernen SOA-Architektur und in der Flexibilität des Systems.

Im Preis/Leistungs-Verhältnis sind die meisten bemüht, die von ihnen gelieferte Leistung im Vergleich zur Konkurrenz als ein preislich sehr günstiges Angebot auszuzeichnen. Nur ein Anbieter sieht seine Stärke in umgekehrter Weise: „Wir sind preislich die teuersten, aber auch Marktführer aufgrund eines hervorragenden Preis-/Leistungsverhältnisses.“ (SS2) Die von ihm gelieferte Leistung wird mit einem entsprechenden Entgelt honoriert.

Die Kundennähe wurde zuletzt von den wenigsten Anbietern als Qualitätsmerkmal genannt. Der Service für den Kunden scheint zumindest von Anbieterseite aus als gegeben angesehen zu werden, so dass eine besondere Konzentration auf ein kundenfreundliches Auftreten nicht notwendig ist.

Einfluss der Kunden

Zwar führen die Befragten Kundennähe selten als Differenzierungsmerkmal an, trotzdem herrscht eine allgemein starke Bindung zwischen Anbietern und Nutzern (vgl. Abbildung 4.32), die es vielen Kunden nach Angaben der Interviewpartner erlaubt, Einfluss auf die Software nehmen zu können.

Dieser Einfluss reicht von Projektwissen, dass bei einem Kunden im Laufe einer Installation gewonnen und in den Standard aufgenommen wurde, über Vorschläge, die Kunden selbst dem Anbieter vortragen, bis hin zu einer Mitsprache bzw. Mitentwicklung von Kundenseite aus. SAP gibt Kunden z.B. die Möglichkeit, für sie Custom Development Projects durchzuführen, die auf individuelle Wünsche besonders eingehen. Daraus entstandene Best-Practice Fälle können anschließend in den Standard einfließen.

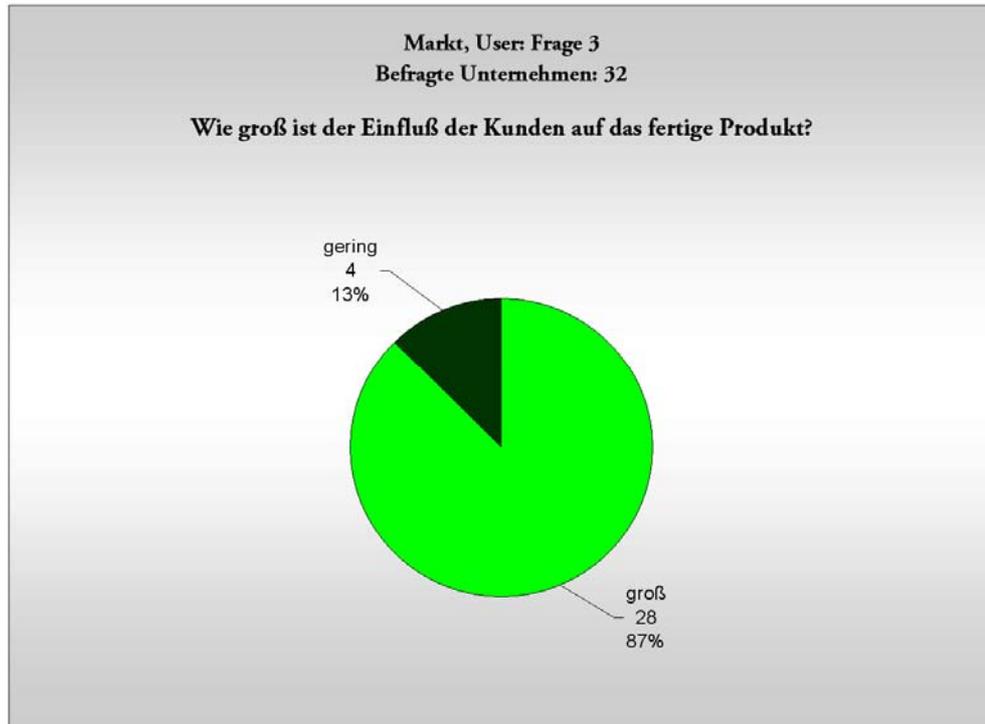


Abbildung 4.32 Kunden haben bei vielen Anbietern Einfluss auf die Systemgestaltung

Nur vier Anbieter lassen eine nutzerseitige Einflussnahme entweder beschränkt oder gar nicht zu: Abacus, ERP21, Jentech und myFactory. 28 von 32 Anbietern gaben an, dass Kunden generell einen großen Einfluss auf ihre Systementwicklung nehmen können. Teilweise resultiert dieser aus einem recht kleinen Kundenstamm, der eine Ausrichtung auf dessen Erfordernisse notwendig macht. Die unternehmerische Größe eines Kunden ist ebenfalls ein wichtiger Faktor, da der finanzielle Einfluss enorm sein kann, wie ein Anbieter versicherte: „Größere Kunden üben natürlich einen größeren Einfluss auf das Endprodukt aus.“ (FR1)

Außerdem bedeutet eine Offenlegung des Codes des Systems automatisch eine Mitentwicklungsmöglichkeit, die Kunden nutzen können.

Kommunikation mit dem Kunden

Kunden Die Kommunikationswege mit den Kunden reichen von einem direkten Kontakt (Mail, Telefon) über die vertriebliche Betreuung, Usergroups, Kundentage und Konferenzen hinweg bis zu einem Interessensverband aus Anwendern. Interessant zu beobachten ist, dass die Anbieter mit geringem Kundeneinfluss ihren Kunden trotzdem verschiedene Austauschmöglichkeiten ermöglichen.

Wie die Kommunikation stattfindet, ist also nicht entscheidend für die Qualität oder den Einfluss auf das Produkt.

4.4.1.4 Fazit

Preis/Leistung spielt in den Augen der Anbieter keine so große Rolle wie die technologischen Eigenschaften eines Systems, um für potentielle Kunden interessant zu sein. Dabei ist es weniger wichtig, ob das System .NET oder J2EE basiert ist, vielmehr sind es spezifische

Eigenschaften wie Datenbankarchitekturen, Betreibermodelle oder eigenentwickelte Systemkonzepte, die von der Konkurrenz abgrenzen sollen.

Plattformunabhängigkeit ebenso wie „moderne“ Programmiersprachen spielen als technologische Aspekte eine wichtige Rolle. Schließlich möchte der Kunde eine möglichst langlebige und auf Veränderungen vorbereitete Software haben. Modernität scheint zu einem Zeitpunkt, an dem alte Systeme ausgetauscht werden müssen und neue Initiativen hin zu neuen Betreibermodellen gemacht werden, mehr denn je gefragt zu sein.

Man lockt also mit dem Produkt und seinen Features, weniger mit Service wie die wenigen Nennungen bezüglich der Kundennähe als Differenzierungsmerkmal zeigen. Trotzdem ist der Service nicht verschwunden, er tritt nur nicht unbedingt als Unterscheidungsmerkmal in Erscheinung.

Dagegen sind Branchenspezifika durchaus noch ein für Kunden interessanter Aspekt, der sie zum Kauf bewegen könnte wie auch, wenn auch nur von einem knappen Drittel angegeben, das Preis/Leistungsverhältnis. Hat sich der Kunde einmal für ein System entschieden, so hat er, abhängig von seiner eigenen Größe und der Entwicklungsstrategie des Anbieters, die Möglichkeit, Einfluss auf das System nehmen zu können. Das zeigt, dass Entwickler von Standardprodukten zumindest teilweise einen kundennahen Kontakt pflegen, so dass neue Anregungen in den Standard einfließen können.

4.4.2 Anbieter

Zuletzt wird die Situation der Anbieter näher beleuchtet. Die Wahl der unterstützten Branchen bestimmt Funktionsumfang und Spezialisierungsgrad. Die Einschätzung ihrer eigenen Position im Vergleich zu ihrer Konkurrenz gibt eine wertvolle Charakterisierung der jeweiligen Interviewpartner ab. Insbesondere Open Source bzw. proprietäre Anbieter als Konkurrenten sollen als Wettbewerber charakterisiert werden, ebenso wie die eigene Zukunft in einem Markt, dessen Entwicklung durch neue Impulse und sich verändernde Bedingungen ständig weiter voranschreitet.

4.4.2.1 Verlauf der Befragung

Prinzipiell wird in dieser Arbeit zwischen drei Branchenzweigen unterschieden: Produktion, Handel und Dienstleistung. Alle drei lassen sich beliebig tief weiter differenzieren, allerdings würde das im Rahmen der untersuchten 32 Unternehmen zu weit führen. Deshalb werden hier den Angaben der Befragten entsprechend die meist genannten Branchenzweige aufgeführt.

Die Produktion spezialisiert sich in diesem Blickfeld noch einen Schritt weiter in den Anlagen- und Maschinenbau, der Handel in Einzel- und Großhandel.

4.4.2.2 Zusammenfassung der Ergebnisse

- Produktionsunterstützung ist immer noch vermehrt gegenüber Handel vertreten, Dienstleistung ist seltener repräsentiert.
- 50 Prozent sprechen Open Source Anbietern kaufmännisches Hintergrundwissen ab.
- Fast 75 Prozent wollen nur in Marktanteilen wachsen.

- Fortschreitende Konsolidierung im Markt und neue Technologien werden hauptsächlich als die zukünftigen Einflussfaktoren im Markt gesehen.

4.4.2.3 Auswertung

Branchenverteilung

Auf eine bestimmte Branche spezialisierte Anbieter bieten trotzdem auch Lösungen für andere Branchenzweige an. Diese Anbieter sind in der jeweiligen Sparte Produktion, Handel oder Dienstleistung eingetragen. Nur diejenigen Anbieter, die ein branchenübergreifendes Produkt anbieten, sind entsprechend vermerkt (vgl. Abbildung 4.33).

So sind laut der getroffenen Aussagen in den Interviews nur fünf Systeme wirklich branchenübergreifend: Abacus, nGroup, Ramco, SAP und Microsoft (Axapta). Der Rest hat entweder im Handel, in der Dienstleistung oder Produktion Schwerpunkte gesetzt.

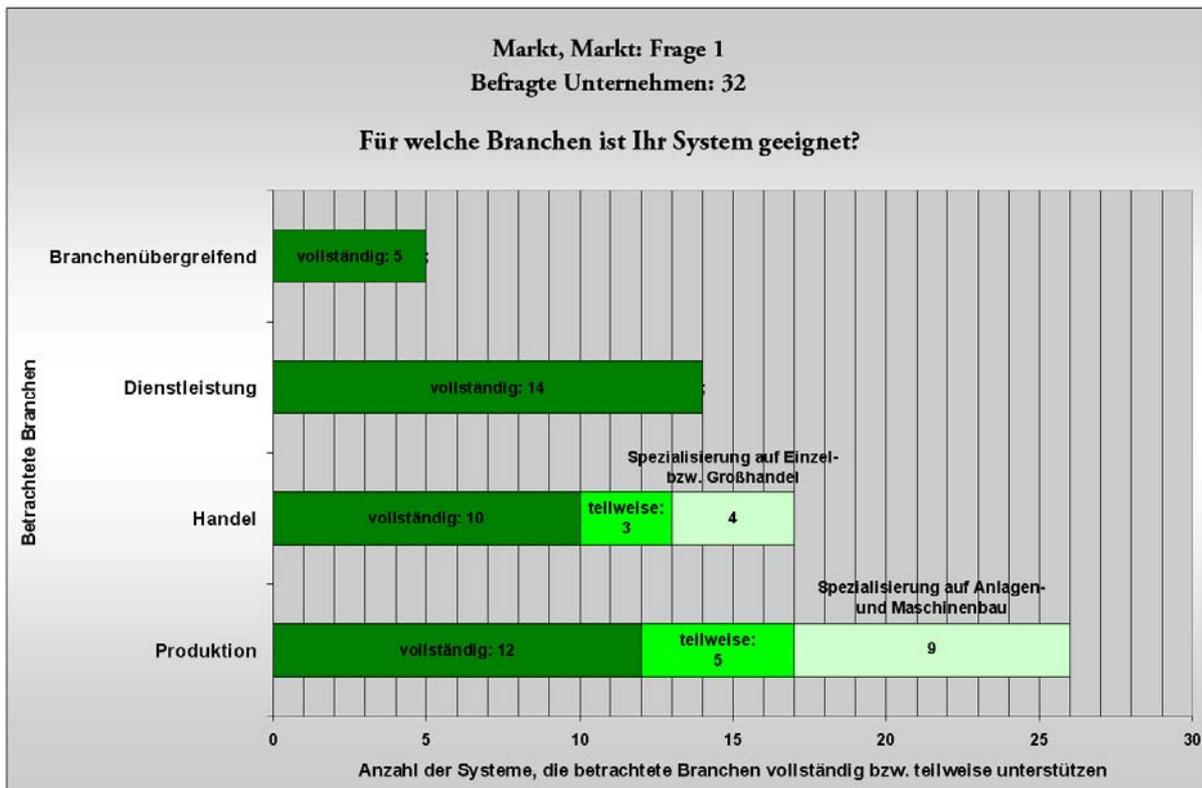


Abbildung 4.33 Das produzierende Gewerbe wird bei den Befragten am meisten unterstützt

Und auch dort werden nicht unbedingt alle produzierenden bzw. Handelsunternehmen unterstützt. Ob es eine Spezialisierung auf den Anlagen- und Maschinenbau oder die Konzentration auf Großhandelsunternehmen ist, auch dort finden sich Einschränkungen. Unter den Befragten zeigt sich, dass die Unterstützung der Produktion eine deutliche Mehrheit gegenüber dem Handel hat. Dienstleistungsunternehmen werden nur von etwa der Hälfte der Anbieter unterstützt.

Marktposition

Die Befragten schätzen ihre Situation im Verhältnis zur Konkurrenz und ihren Kunden teilweise sehr unterschiedlich ein. Folgende Aussagen wurden während der Befragungen getroffen:

- Marktführerschaft (regional begrenzt bzw. branchenspezifisch)
- Hoher Bekanntheitsgrad (regional begrenzt bzw. branchenspezifisch)
- Groß in der Nische
- „Wir sind wir analog zu den Bereichen des VW-Konzerns der Skoda: gleichwertige Technik, aber aufgrund weniger Renommee (und bei uns geringerer Unternehmensgröße) günstigerer Preis“ (ME1)
- Bei Anwendern beliebt (Gute Position in der Trovarit-Studie)
- Stark international vertreten
- Nummer 3 weltweit
- Technologische Vorreiterrolle
- Europäischer Anbieter (Mehrsprachenfähigkeit)
- Challenger im Markt
- Innovativer Anbieter
- Kleiner Marktanteil
- Anbieter für KMU mit hohem Individualisierungsbedarf
- Weltmarktführer in betriebswirtschaftlicher Software
- Marktführer im Bereich Open Source
- Nummer 1 im deutschen Mittelstand

Viele betrachten dabei ihr Unternehmen in Relation zu anderen direkten Konkurrenten. Herausragende Eigenschaften des Systems als Charakterisierungsmerkmal werden eher selten für die Position im Markt benannt. Insgesamt lässt sich sagen, dass die Unternehmen in dieser Befragung ein heterogenes Bild der Anbieterlandschaft liefern.

Es sind sowohl regional begrenzt operierende Systeme dabei als auch europäische und weltweite Anbieter. Es gibt Kandidaten mit recht kleiner Kundenzahl, welche im mittleren Spektrum angesiedelt sind und wenige, bei denen die Zahlen in die zehntausende gehen. Nischenanbieter und branchenübergreifende Systeme sind ebenso vertreten wie Challenger und Marktführer oder Mittelständler und Konzerne.

Open Source/Proprietäre Anbieter als Konkurrenz

Alle Befragten bis auf zwei, Seat-1 und SynERP, sind proprietäre Softwareanbieter, die zusätzlich zum Service auch ein Entgelt für die Software selbst verlangen. Open Source Entwickler sind aus ihrer Sicht keine echte Konkurrenz.

Nur zwei Anbieter (Miclas und B.I.M.) teilen diese Meinung nicht. B.I.M. sieht vor allem in der Bereitstellung von Einzelfunktionen wie z.B. „Einkauf“ ein Potenzial für OS-Systeme. Miclas befindet sich aufgrund der relativ geringen Userzahl seiner potentiellen Kunden in direkter Konkurrenz mit OS.

Trotzdem wird gleichzeitig auch hier der Hauptgrund angeführt, warum OS-Entwickler nicht geeignet sind, eine komplette Unternehmenslösung zu erstellen: der kaufmännische Hintergrund fehlt (vgl. Abbildung 4.34).

Ein Drittel der Befragten hält das betriebswirtschaftliche Grundwissen der Entwickler von OS-Systemen für mangelhaft bzw. nur in einzelnen Bereichen für ausreichend. CRM-Systeme werden z.B. durchaus als ebenbürtige Konkurrenz gesehen, für die sich ein Kunde entscheiden könnte.

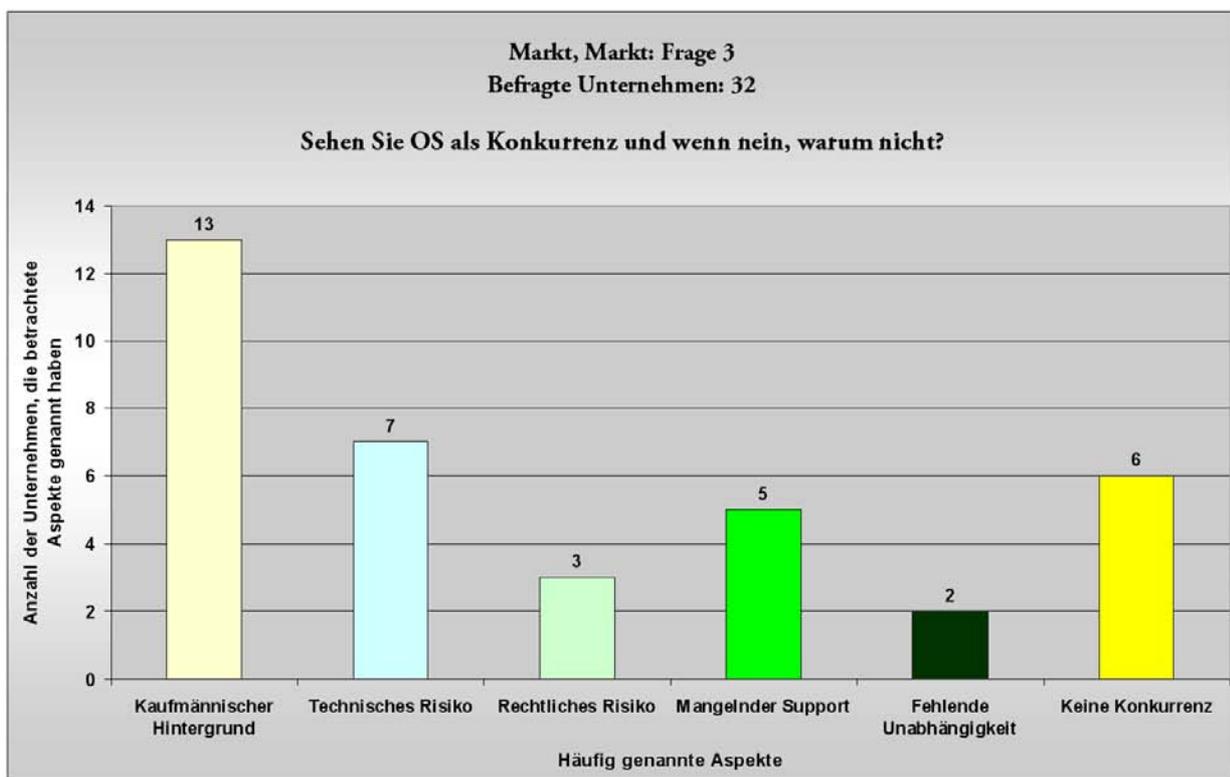


Abbildung 4.34 OS wird nicht als ernsthafter Konkurrent angesehen

Komplettsysteme hingegen erfüllen nicht die funktionalen Anforderungen und scheitern somit in ihrer Aufgabe als Backbone eines Unternehmens, wie ein Anbieter versicherte: „Open Source ERP-Lösungen bedürfen im Moment noch zu viel Anpassungsarbeiten (die bei Standard-Software bereits erledigt ist), die meist eingekauft werden müssen.“ (ME2)

Der nächste, jedoch nicht als so gravierend angesehene Punkt ist das technische Risiko, dass nicht abschätzbar zu sein scheint. Mangelnde Validierbarkeit wird hier als Argument angeführt. Aber es gibt auch Gegenstimmen, die OS technisch gesehen eine gute Qualität zuschreiben, jedoch im fachlichen Sinne absprechen. Das rechtliche Risiko aufgrund nicht beachteter Rechtsprechung z.B. im Steuerwesen wird dreimal angesprochen.

Gravierender scheint die mangelnde Support-Infrastruktur zu sein, die aufgrund schlechter oder nicht vorhandener Schulungen, Workshops und ähnlichem keine Unterstützung im Systembetrieb leistet. Gleichzeitig wird das vermeintliche Versprechen der Unabhängigkeit durch Open Source bemängelt, denn das System ist zwar an sich frei und willkürlich veränderbar, trotzdem bindet man sich an einen Dienstleister, wenn man die Unternehmenssoftware ernsthaft betreiben möchte. Sechs Befragte sehen Open Source einfach nicht als Konkurrenten ohne nähere Angaben zu machen.

Unter den Befragten befanden sich zwei Open Source Anbieter, die eine Einschätzung der Konkurrenzsituation aus ihrer Sicht geben konnten. SynERP sieht durchaus proprietäre Anbieter als Mitstreiter um potentielle Kunden, allerdings konzentriert sich das Unternehmen eher auf die unmittelbare OS-Konkurrenz. Seat-1 sieht die Standardisierungsbemühungen der Konkurrenten als nicht konkurrenzfähig an für die Zielgruppe, die von ihnen bedient wird. Als Anbieter eines betrieblichen Anwendungssystems für eine Nischenbranche ist die Individualisierung für jeden Kunden ausschlaggebend.

Standardsoftware kann die meist speziellen Anforderungen nicht erfüllen. So setzt das Unternehmen auf agile Softwareentwicklung und agiles Projektmanagement, dass basierend auf dem Grundsystem je nach Kunde individuell ein System entwickelt.

In fünf bis zehn Jahren

Etwa zwei Drittel der Befragten sehen bzw. wünschen für sich eine Vergrößerung ihres bisherigen Marktanteils bzw. Kundenstamms (vgl. Abbildung 4.35).

Dabei reichen die Vorstellungen von einem konstanten Wachstum über eine jährliche Steigerungsrate von 20 Prozent bis hin zu einer angestrebten Marktführerschaft bzw. deren Ausbau. Acht Unternehmen haben vor, in weitere Branchenzweige zu expandieren und damit ihre potentielle Zielgruppe deutlich zu vergrößern, fünf planen sogar die Expansion auf globaler Ebene.

Ein weiterer Gedanke, den ein Interviewpartner in näherer Zukunft als aktuell bezeichnet, ist der Zusammenschluss von mehreren Anbietern, um sich vor eventuellen Übernahmen oder zu starkem Konkurrenzdruck schützen zu können:

„Es gibt evtl. die Möglichkeit, sich in einem ERP-Cluster zusammenzuschließen.“ (RG2)

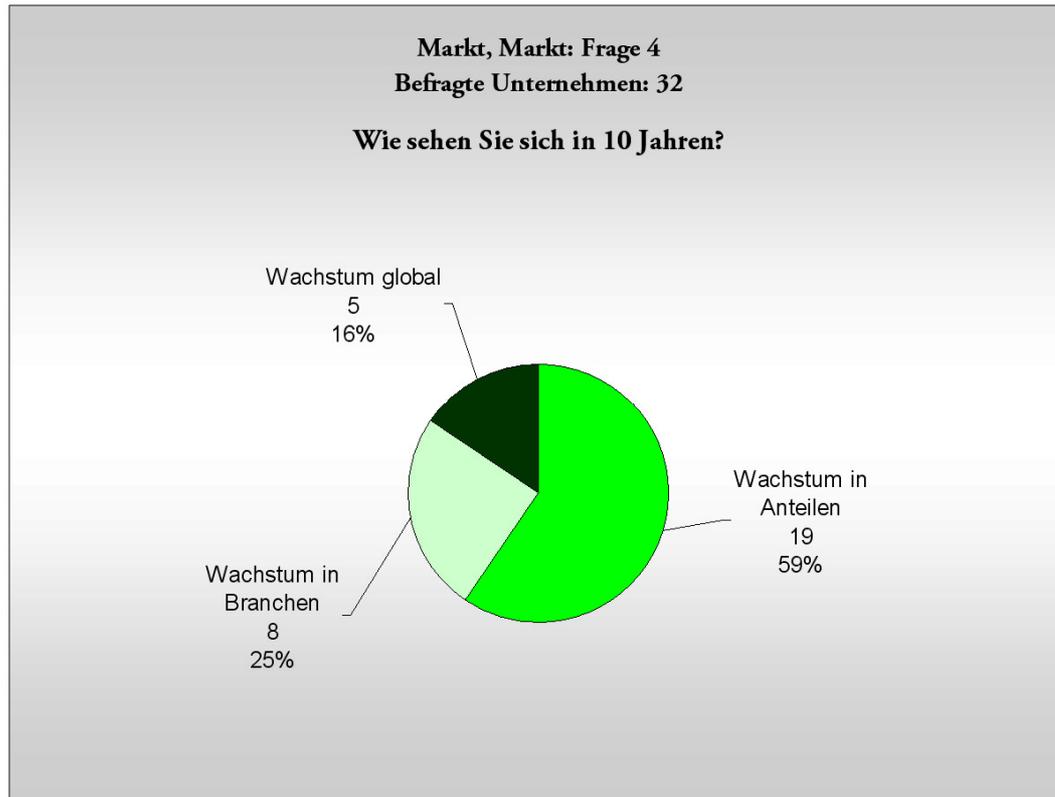


Abbildung 4.35 Wenige mit aggressivem Expansionskurs, eher stetiges Wachstum im Vordergrund

Zukunft des Anbietermarktes

Die beiden größten Einflussfaktoren in der zukünftigen Entwicklung sind in den Augen von fast einem Drittel der Befragten die fortschreitende Konsolidierung des Marktes und neue technologische Entwicklungen (vgl. Abbildung 4.36). Es werden weiterhin Unternehmen aufgekauft bzw. fusioniert. Dies liegt zum einen an einer gewissen Sättigung des Marktes, zum anderen an Finanzdienstleistern, die ERP-Anbieter aufkaufen und zu einem späteren Zeitpunkt wieder abstoßen. Hinzu kommen die strategischen Aufkäufe, die Konzerne wie SAP oder Oracle zur Erweiterung ihres Portfolios tätigen. In diesem Zusammenhang wurde von einem Anbieter die zukünftige Partnerverlässlichkeit in Frage gestellt, wenn IT Partner von Unternehmen plötzlich übernommen oder zerschlagen werden könnten.

Einig ist man sich in der Tatsache, dass die Zahl der Anbieter sinken wird, allerdings gibt es unterschiedliche Meinungen, in welcher Sparte sie weniger werden. Auf der einen Seite ist man überzeugt, dass kleine Unternehmen verschwinden werden:

„Kleine Anbieter werden von der Bildfläche verschwinden.“ (HE2)

Auf der anderen Seite räumt man Mittelständlern, vor allem denjenigen, die eine standardisierte Software anbieten, wenige Chancen ein: „Es wird im Mittelstand keine 600 sondern nur noch zwei Dutzend Anbieter geben.“ (WE5)

Es gab nur eine Stimme, die auch einen Global Player, SAP, dank seiner neu entwickelten Business by Design Software keine Chancen auf Überleben eingeräumt hat: „SAP wird es in der Art in zehn Jahren nicht mehr geben.“ (WS3)

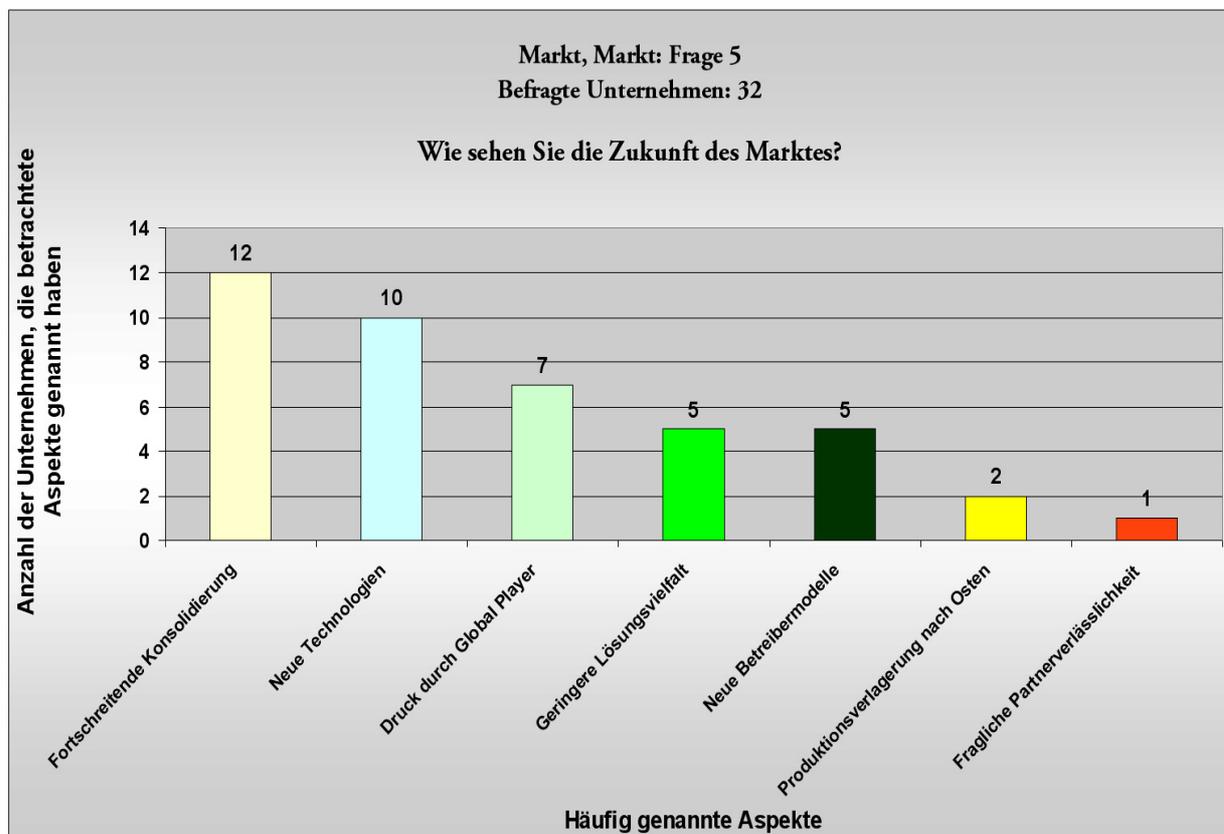


Abbildung 4.36 Konsolidierung des Marktes und neue Technologie-Impulse sehen die Befragten voraus

Zehn Befragte sehen neue Technologien als maßgeblichen Faktor, der zukünftige Entwicklungen beeinflussen wird. Dazu gehören web-basierte Technologien ebenso wie die Erneuerung alter Systeme und die Abkehr von Komplettsystemen hin zu modularisierten Baukästen. SaaS wird mehr Bedeutung zukommen. Fast im gleichen Atemzug werden neue Betreibermodelle als Faktor genannt, die alte Lizenzmodelle über kurz oder lang ablösen werden. Denkbar wäre z.B. eine Sammlung von Lösungskits, die unter einem großen Label wie z.B. Microsoft vertrieben werden, d.h. die Anzahl der Lösungen nimmt zwar nicht ab, jedoch die Anzahl der Anbieter: „Die Lösungsvielfalt im Markt wird erhalten bleiben, die Zahl der Anbieter sinkt aber (Lösungskits)“ (RG3)

Der Druck von oben durch die großen Konzerne wie SAP oder Oracle wird ebenfalls laut einigen Befragten zukünftig eine größere Rolle spielen, da diese durch ihren Vorstoß in den KMU Markt Marktanteile wegnehmen werden. Hinzu kommen neue Impulse aus China und Indien, in die immer mehr Produktion verlagert wird und dadurch auch neue Ideen einen stärkeren Einfluss in Europa haben werden:

„Indien und China sind der Backbone der IT-Industrie, und von ihnen werden neue Impulse erwartet.“ (FR3)

4.4.2.4 Fazit

Open Source ist zwar ein viel diskutiertes Thema, wird als wirkliche Bedrohung jedoch aufgrund des angeblich mangelnden wirtschaftlichen Verständnisses der Systemhersteller ignoriert. Trotzdem existieren diese Anbieter und können durchaus hohe Kundenzahlen aufweisen.

Anbieter setzen vermehrt auf ihre technologischen Neuerungen als Kompetenzmerkmal im Vergleich zur Konkurrenz. Gleichzeitig werden technologische Neuerungen neben der fortschreitenden Konsolidierung als treibende Kraft in der Zukunft des Marktes gesehen.

Die Entwicklung des Marktes scheint in den Augen der Befragten klar darauf hin zu deuten, dass die Anbieterzahl weiter abnehmen wird. Welche Gruppe das genau betrifft, scheint strittig zu sein. Die Nischenanbieter sind davon überzeugt, dass ihre Exzellenz in der jeweiligen Branche sie vor allzu großer Konkurrenz schützen wird, da branchenübergreifende Systeme nicht die Funktionstiefe besitzen können, die ihre Kunden benötigen. Unternehmen, die mehrere Branchen unterstützen, versuchen, sich auf ihre jeweiligen Besonderheiten zu stützen wie z.B. ein innovatives Betreibermodell oder die Konzentration auf Klein-Unternehmen mit 5-25 Usern. Es scheint, dass eine Standardsoftware im KMU Bereich, besonders im unteren Teil bei den kleinen Unternehmen, nicht lukrativ ist:

„Eine Standardlösung anzupassen wird teurer sein, als eine Individuallösung zu erzeugen“.
(BI4)

Individualisierung und das Eingehen auf spezielle Kundenwünsche ist laut vielen Befragten essentiell für ein erfolgreiches Arbeiten in diesem Segment.

5 Prognose - Ein Blick in die Zukunft

Welchen Weg werden zukünftige Konzepte und Architekturen unter gleichzeitiger Betrachtung der anderen Entwicklungen in Technologie, Betreibermodell und Markt einschlagen? Basierend auf den Auswertungen der geführten Interviews in Kapitel 4 folgt nun zuerst die Skizzierung eines auf den Ergebnissen aufbauenden, möglichen Szenarios, wie ERP-Systeme in mehreren Jahren aussehen könnten.

Anschließend wird eine zukunftsorientierte Bewertung der getroffenen Aussagen vorgenommen. Alle vier Kapitel Architektur, Technologie, Betreibermodell und Markt werden hinsichtlich der möglichen zukünftigen Entwicklungen näher betrachtet. Zusätzlich werden aktuelle und auch zukunftsweisende Entwicklungen in den jeweiligen Abschnitten erwähnt, so dass man einen möglichst umfassenden Eindruck möglicher Szenarien gewinnen kann.

5.1 Ein mögliches Szenario

ERP-Systeme haben den Anspruch, eine komplette Unternehmenslösung anzubieten, die sämtliche benötigten Funktionalitäten umsetzen kann. Dies mag in einem Greenfield Ansatz sogar sehr gut anwendbar sein, allerdings bestehen in der Regel neben den neu einzuführenden Systemen bereits Legacy-Systeme vor Ort, deren Integration in das neue ERP-System bewältigt werden muss. Hinzu kommt, dass viele Unternehmen Kommunikationsstrukturen mit betrieblichen Anwendungssystemen ihrer geschäftlichen Partner besitzen, die ebenfalls an das neue System angeschlossen werden müssen. Bislang war dies oft über eine Punkt-zu-Punkt Integration geschehen, indem die Kommunikationskanäle für jede Verbindungsstelle manuell ausprogrammiert wurden. Diese Vorgehensweise resultiert aus einer Funktionssicht auf ein betriebliches Anwendungssystem.

Erst die Prozesssicht ermöglicht es, zumindest auf Modellebene, die verschiedenen Systeme zur Erfüllung eines geschäftlichen Prozesses miteinander in Beziehung treten zu lassen. Im Folgenden wird ein Szenario beschrieben, dessen Realisation über alle Systeme hinweg noch in weiter Zukunft liegt, dessen Wurzeln aber bereits heute erkennbar sind.

Entwicklungspotenzial in der Architektur

SOA und BPM erlauben zusammen eine neue und einfache Möglichkeit, verschiedene Anwendungen (zugänglich über die Services) entsprechend eines Prozessmodells zu orchestrieren. Das Resultat ist eine enorme Flexibilität in der Systemgestaltung. Anwendungen können, wenn nötig, beliebig in das System integriert werden. Funktionsfähige Serviceketten resultieren umgehend aus den neu modellierten Prozessen.

Entwicklungspotenzial in der Technologie

Web Services dienen dafür als allgemeine Schnittstelle unabhängig von den über sie transportierten Dokumenten. Web-basierte Systeme erlauben einen einfachen Zugang über Browser zu den Funktionen, so dass ein weltweiter Zugriff ohne vorherige Installation einer Software möglich ist.

Entwicklungspotenzial im Betreibermodell

Das Potenzial von SOA für das Betreibermodell SaaS ist enorm. SOA liefert die technische Grundlage, um Services zwischen verschiedenen Systemen flexibel miteinander integrieren

zu können. SaaS bietet Anwendungen als Mietobjekt über das Netz an. Die Möglichkeit, nun aus diesen vielen Anwendungen heraus die jeweils richtigen auszusuchen, zu orchestrieren und in sein bestehendes System zu integrieren, realisiert ein flexibles Betreibermodell.

Entwicklungspotenzial im Markt

Diese Flexibilität eröffnet im Hinblick auf die Herstellung und Vermarktung der ERP-Systeme neue Möglichkeiten. Große Anbieter wie SAP, Oracle oder IBM müssen nicht mehr eigene Module konzipieren und programmieren. Durch eine SOA können Fremdprodukte sehr schnell in die bestehende Anwendungslandschaft integriert werden. Somit entstehen Systeme nach dem Baukastenprinzip.

Ein solches Szenario ist in den kommenden Jahren noch nicht wie beschrieben auf einer breiten Basis umsetzbar (wie man in den weiteren Kapiteln sehen kann). Gerade in der Zusammenarbeit zwischen Services und Prozessen liegt noch viel Entwicklungsbedarf. Trotzdem sind die Voraussetzungen bereits heute gegeben. Eine Service-Orientierung hat sich teilweise schon durchgesetzt. Die Prozessmodellierung ist im Ansatz vorhanden. Technisch müssen sich Web Services noch weiterentwickeln, um als allgemein gültiger Intermediär wirken zu können. Der Trend, keine eigenen Lösungen zu entwickeln, sondern das Portfolio durch Aufkäufe zu ergänzen, spiegelt sich in der Konsolidierung des Marktes wider, die weiter fortschreitet.

5.2 Architektur

Im Folgenden wird eine Prognose der planerischen, konzeptionellen, funktionsorientierten und integrationsorientierten Impulse gegeben, die in den kommenden Jahren die Entwicklung von ERP-Systemen bestimmen werden.

5.2.1 Planungsebene

Aus den Interviews gewonnene Erkenntnisse:

- Eine Prozessmodellierung ist nur bei der Hälfte der befragten Unternehmen umgesetzt.
- Die Modellierung selbst erfordert Kenntnisse in der jeweiligen Modellierungssprache.
- Eine automatisierte Codeerzeugung wird zwar ansatzweise umgesetzt, allerdings oft aufgrund zu komplexer interner Abhängigkeiten in einem ERP-System als unmöglich erachtet.
- Der Einsatz beim Kunden hat meist nur einen Visualisierungscharakter.
- Standards in der Prozessmodellierung sind sehr selten im Einsatz.
- Die Anpassung der Geschäftsprozesse durch den Kunden ist oft mit Einschränkungen/Vorwissen in der Entwicklung behaftet.

Die frühere Notwendigkeit, seine eigenen Geschäftsprozesse an das ERP-System anpassen zu müssen [David et al., 2003, S. 67], kann durch eine service-orientierte und komponentenorientierte Systemarchitektur in Zukunft umgangen werden. Die Vision des Business Process Managements, eines Tages aus einer Prozessgestaltung heraus ein lauffähiges System erzeugen zu können, wird jedoch auch in Zukunft eine Vision bleiben. Die

Komplexität und der damit verbundene Aufwand, ein Prozessmanagement dieser Art zu entwickeln, scheint für viele ERP-Anbieter unattraktiv zu sein: „ERP-Systeme sind zu komplex, als dass eine Geschäftsprozessmodellierung sinnvoll wäre.“ (GAL1)

Die zugrunde liegende Architektur, um eine derartige Gestaltungsfreiheit und -mächtigkeit zu erlangen, müsste nach heutigem Stand komponenten-orientiert und auch service-orientiert sein [Strnadl, 2006, S. 345], so dass sich die mit den Prozessen verbundenen Funktionalitäten problemlos nach denen im Modell vorgenommenen Umstrukturierungen neu miteinander anordnen lassen. Selbst dann gilt es noch, die Zustände der Komponenten, Vor- und Nachbedingungen usw. zu beachten. Dieser deutliche Mehraufwand wird ohne entsprechende Nachfrage auf Kundenseite von der Mehrheit der Anbieter in naher Zukunft nicht angegangen werden.

Dabei ist eine flexible und änderungsfreundliche Prozessgestaltung im Vergleich zu den immer noch verbreiteten herkömmlichen Systemen vom Aufwand her deutlich geringer [Strnadl, 2006, S. 409]. Immerhin haben sieben Anbieter erste Ansätze verwirklicht, um solche Systeme in Zukunft anzubieten. Darunter sind auch weltumspannende ERP-Anbieter wie Oracle oder Ramco. Laut einer Studie von BPTrends [Harmon & Wolf, 2007, S. 25] sind 41 Prozent der Anwender betrieblicher Anwendungssysteme interessiert an den Prozessdarstellungs-Möglichkeiten BPMN (Business Process Modelling Notation) bzw. der UML (Unified Modelling Language) zur Darstellung ihrer Prozesse.

Die Mehrheit der Anwender befindet sich in einem Stadium, in dem sie ihre Kernprozesse erfasst haben und nun erst die Zusammenarbeit all ihrer Prozesse im Unternehmen begreifen müssen [Harmon & Wolf, 2007, S. 45]. Eine Studie der FHNW belegt, dass in den kommenden Jahren schlanke und rollenbasierte Prozesse in vielen Unternehmen eine wichtige Rolle spielen werden [Felley, 2007, S. 6]. Somit ist auf Kundenseite noch viel Potenzial und Bedarf an einem Business Process Management.

Ebenso wie bei den Dokumentenstandards kann eine einheitliche oder zumindest artverwandte Prozessdarstellung eine Anbindung älterer Legacy-Systeme oder die Anbindung an einen Geschäftspartner deutlich erleichtern. Die unternehmensübergreifende Integration von Geschäftsprozessen mit einheitlichen Standards ist momentan jedoch erst am Anfang. Es sind Tendenzen zu erkennen, diese sehr heterogene Landschaft zu homogenisieren (RosettaNet, ebXML Business Process), allerdings wird dies wohl noch nicht in den nächsten Jahren gelingen.

Da SOA als grundlegende Architektur ebenfalls eine größere Verbreitung finden wird (vgl. Kapitel 5.2.2), ist eine Orchestration von Services aus der Modellierung heraus in realistischer Reichweite. Anpassungen werden jedoch weiterhin für den Großteil der Kunden nur mit Hilfe der Anbieter möglich sein, da die Nutzer nicht unbedingt ein programmiertechnisches Verständnis haben, geschweige denn einen Zugriff auf den Quellcode, dessen Anpassung nach Prozessänderungen erforderlich ist.

Einzellösungen, wie sie unter anderem in Fallbeispielen erläutert werden [Wölfle & Schubert, 2007], bleiben vorläufig die Regel. Wie schon bei der Prozessmodellierung, sind es vermehrt die großen Anbieter wie SAP oder Microsoft, die eine Einführung von Prozessstandards in ihrem System voran treiben. Es bleibt abzuwarten, ob sich eine solche Standardisierung in der Prozessebene langfristig und auf breiter Basis durchsetzen kann, denn der Erfolg steht und fällt mit dem Willen der einzelnen ERP-Anbieter, sich auf gemeinsame Prozessstandards festzulegen und der Fähigkeit der Kunden, ein Modellierungswissen anzuwenden, welches nicht selbstverständlich ist: „Geschäftsprozessexperten kommen wesentlich leichter mit einem

linearen Text zurecht als mit der Modellierungssprache, in der die Prozesse niedergeschrieben werden sollen." (GAL2)

Prognose:

- Man wird zunehmend Prozessabbildungen bzw. -gestaltungen in Systemen sehen.
- Dank zunehmender Service-Orientierung der Systeme ist eine Orchestrierung der Services aus der Prozessgestaltung möglich, allerdings noch nicht kurzfristig. Die manuelle Anpassung wird noch einige Jahre die Regel bleiben.
- Prozessstandards werden sich nicht gegen die individuelle Prozessgestaltung durchsetzen können.
- Der Kunde wird gegenüber der Prozessgestaltung sensibler.

5.2.2 Systemarchitektur

Aus den Interviews gewonnene Erkenntnisse:

- Service-Orientierung ist kein Schlagwort mehr, sondern wird in Architekturen umgesetzt.
- Service-Orientierung ist nicht einheitlich definiert.
- Der Webserver hält Einzug in die n-Tier-Landschaft.
- Plattformunabhängigkeit wird bevorzugt.
- J2EE als Plattform noch mit Übergewicht zu Microsoft und .NET trotz seiner großen Serververbreitung.
- Browser und somit web-basierte Systeme kommen häufiger zum Einsatz.

Eine service-orientierte Architektur scheint sich unter den ERP-Anbietern durchzusetzen. Obwohl der Begriff SOA noch nicht einheitlich definiert ist (vgl. Kapitel 3.2.2.7), haben neun Anbieter bereits nach eigenen Angaben eine solche Architektur als Konzept ihrer Software umgesetzt, immerhin acht Anbieter bezeichnen sich als SOA-fähig, können also in einer SOA problemlos eingebunden werden, wobei die innere Struktur nicht oder nur teilweise service-orientiert ist.

Diese Entwicklung ist unabhängig von einer komponenten-orientierten Architektur, wie man im Kapitel 4.1 sehen kann. Dort haben, wenn man die Integration des primären Funktionsbereichs hinzuzählt, mehr als zwei Drittel der Anbieter starke interne Abhängigkeiten in ihren Funktionsbereichen. Vier Anbieter (Ramco, IFS, Oracle und Alea), die SOA umgesetzt haben, besitzen zusätzlich ein komponenten-orientiertes System. Somit sind es nur diese vier Softwarehäuser, die einer in der Literatur (vgl. [Strnadl, 2006]; [Turowski, 2004]; [Turowski, 2003]; [Allweyer, 2005]) propagierten Ideal-Lösung nahe kommen. Allerdings wird das für die nächsten drei bis fünf Jahre noch die Ausnahme bleiben.

Zumindest die service-orientierte Sichtweise setzt sich verbreitet durch. Mit Integrationsmechanismen wie Web Services, deren Nutzung vor allem als Funktionsintegrator inzwischen breiten Einsatz erfährt und der Notwendigkeit, ältere Legacy-Systeme an neue Systeme

anbinden zu müssen, bringen in Services gekapselte Funktionalitäten einen erheblichen Mehrwert, der in Zukunft sicherlich genutzt wird.

Unter den Befragten ist SOA selbst noch nicht einheitlich definiert. Konsens herrscht nur darüber, dass es sich im Kern um Services und ihren Einsatz in betrieblichen Anwendungssystemen dreht. Da gerade SAP, Oracle, IBM und Microsoft teilweise verschiedene Ansichten einer SOA haben, wird die etwas ungenaue Begriffserklärung noch weiter vorherrschen.

Betrachtet man die eigentliche Architektur eines Systems, so ist die Plattformunabhängigkeit weit verbreitet. Diese Eigenschaft mag für Clients selbst nicht unbedingt ausschlaggebend sein, da immer noch sehr viele Clientsysteme mit Windows operieren. Für Applikationsserver ist dies jedoch nicht trivial. .NET ist zwar theoretisch plattformunabhängig, in der Praxis jedoch mit Windows Betriebssystemen eng integriert [Masak, 2005, S. 208]. Microsofts Strategie, für .NET im Zusammenspiel mit Windows eine größere Verbreitung zu erreichen, funktioniert. Windows-Server haben laut Discherl [Discherl, 2007] im zweiten Quartal 2007 einen Anteil von über 67 Prozent erreicht.

Betrachtet man .NET im Hinblick auf Middleware-Produkte wie z.B. J2EE-basierte Applikationsserver, so lässt sich zu diesem Zeitpunkt noch nicht feststellen, ob sich ein bestimmtes Produkt dauerhaft durchsetzen wird. Unter den Befragten gibt es viele eigenentwickelte und weniger bekannte Frameworks, die eine untergeordnete Rolle neben Middlewares wie z.B. NetWeaver von SAP, Oracle Fusion oder J2EE-Produkte oder der .NET-Plattform spielen. Diesen Frameworks ist die Eigenschaft einer komponenten- und service-orientierten Architektur gemein, wodurch sie die benötigte Infrastruktur für eine SOA bereitstellen können. Achtet man nur auf diesen Aspekt, so haben in der Summe mehr als die Hälfte der Anbieter bereits ein potentiell Werkzeug, um ein auf Services ausgerichtetes System zu schaffen. In diesem Zusammenhang hat AMR Research in einer Studie untersucht [Burger, 2008], welche Middlewares in SOA-Projekten die größte Verbreitung haben: Microsoft führt mit 59 Prozent gegenüber Websphere von IBM mit 44 Prozent, Bea Weblogic mit 35 Prozent, Oracle Fusion mit 31 Prozent und SAP Netweaver mit 29 Prozent das Feld an. Anfang 2008 ordnete Oracle die Verhältnisse durch den Kauf von Bea jedoch neu. Durch die Akquisition wird Oracle führender Anbieter im Middleware-Markt [Hill, 2008].

Das Verhältnis von Thin- zu Fat-Clients wird sich dagegen voraussichtlich weiter zugunsten der Thin-Clients und im speziellen Browser als User Interface entwickeln. Bereits die Hälfte der Thin-Clients werden durch Browser wie Internet Explorer oder Firefox etc. repräsentiert. HTML, HTTP und XML im Besonderen eignen sich aufgrund der bereits existierenden Dokumenten- und Übertragungsstandards gut zum Informationstransport. Bedenkt man, dass XML in jedem der befragten Systeme (vgl. Kapitel 4) als Dokumentensprache unterstützt wird, scheint es nur eine natürliche Entwicklung zu sein, wie ein Anbieter versicherte: „Wenn man einmal ein web-basiertes System hat, ist alles viel einfacher.“ (F11)

Es findet eine Migration der Clients von ursprünglich klassischen In-House Terminals zu Web Interfaces statt. Dabei scheint der Browser momentan die Oberhand gegen über Terminallösungen wie Citrix zu erlangen.

Prognose:

- Die Service-Orientierung wird sich weiter in den ERP-Systemen verbreiten, jedoch nicht so schnell als konzeptionelle Grundlage, sondern vielmehr als Möglichkeit, ältere Anwendungen via Services leichter wiederverwenden zu können.

- Eine einheitliche Definition einer SOA wird es in kürzerer Zeit noch nicht geben, aber die großen Ähnlichkeiten zwischen den verschiedenen Ansätzen dürften keine Probleme in der Zusammenarbeit bereiten.
- Plattformen wie J2EE und .NET werden durch ihre auf Services und Komponenten ausgerichtete Architektur eine größere Verbreitung finden.
- Browser werden sich im Zuge der häufiger vorkommenden web-basierten Systeme als Client weiter verbreiten.

5.2.3 Modularchitektur

Aus den Interviews gewonnene Erkenntnisse:

- ERP-Systeme sind bei kleineren Anbietern keine Komplettlösungen mehr in diesem Sinne, da nicht alle Funktionalitäten aus eigenem Hause abgedeckt werden.
- Kundenorientierte Systeme sind die Regel.
- Customizings sind an Auflagen gebunden, trotzdem existieren sehr viele releasefähige Systeme. Individualität ist in der Software möglich.

Die Defizite vor allem im sekundären Bereich der unternehmerischen Tätigkeiten nach Porter [Porter, 2000, S. 66] sind zwar auffällig hoch (der größte Mangel herrscht in Personalwesen mit knapp 30 Prozent), allerdings sind es Tätigkeiten, die man durch Partnersoftware zufriedenstellend ausfüllen kann, wie es einige der Anbieter praktizieren. Der Kunde erhält trotzdem fast immer eine komplette Lösung durch den jeweiligen Anbieter, d.h. die Partnersoftware wird nahtlos in das ERP-System integriert und er muss sich um entsprechende Sonderlösungen nur selten selbst kümmern. Diese Vorgehensweise kann durch eine SOA sogar weiter begünstigt werden, da Anwendungen einfacher und flexibler in ein bestehendes System integriert werden können.

Betrachtet man SaaS-Modelle, so wird dort der Funktionsumfang eventuell nicht alle Anforderungen eines potentiellen Kunden abdecken können. Da hier für alle Nutzer brauchbare Module angeboten werden müssen, wird sich die Funktionsbreite für das Basispaket nicht auf Speziallösungen ausdehnen, sondern reduzieren.

Dagegen können mit Hilfe von Partnernetzwerken die fehlenden Funktionen trotzdem bereitgestellt werden. Es ist ein Trend zu erkennen, der eine Neuorientierung zu Individualentwicklungen auf Basis einer Standardplattform beinhaltet, wie mehrere Interviewpartner versicherten: „Aus unserer Sicht wird der Trend zur Spezialisierung hinsichtlich der Branchenausrichtung weiter anhalten.“ (PJ2)

Gerade kleinere Unternehmen benötigen nicht immer die ganze Funktionsbreite einer ERP-Lösung. Gleichzeitig sind sie mit mehreren IT-Partnern verbunden, um ihre betriebliche Leistung zu erbringen [David et al., 2003, S. 65]. Betrachtet man im speziellen BI, so werden mit den Übernahmen der großen BI-Anbieter wie Cognos, Business Objects oder Hyperion Solutions [Alexander, 2007] indirekt SAP, IBM oder Oracle zu neuen Partnern im Bereich BI-Integration. Da SAP und Oracle als große ERP-Anbieter weltweit operieren und IBM in diesem Segment ebenfalls stark aktiv ist, z.B. in Kooperation mit mehreren ERP-Anbietern Hosting-Lösungen anbietet, Applikationsserver stellt usw., bleibt abzuwarten, inwiefern sich das Verhältnis zu kleineren ERP-Anbietern, die nun Lösungen von SAP und IBM nutzen, ändern wird. Die großen Anbieter haben durch diese Aufkäufe ihr Portfolio um einen Funktionsbereich

aus den sekundären Aktivitäten erweitert. Dadurch können sie zum einen bei kleineren ERP-Häusern indirekt als Zulieferer auftreten, zum anderen haben sie mit einer eigenen BI-Lösung in ihrem System einen weiteren Schritt zu einer sehr breiten, wenn nicht sogar angestrebten vollständigen Funktionspalette gemacht.

Das Angebot, releasefähige Customizings von Kundenseite aus zu erstellen, wird auch weiterhin zwar möglich aber von bestimmten Restriktionen abhängig sein. Ein Interviewpartner bestätigte es folgendermaßen: „Der Kunde kann innerhalb der für ihn vorgesehenen Schublade operieren.“ (EG1)

Eine komponenten-orientierte Architektur, die eine solche Entwicklung erleichtern würde, liegt noch in weiter Zukunft.

Prognose:

- Partnersoftware, besonders für sekundäre Aktivitäten, wird weiterhin genutzt. Die Komplettlösung wird es seltener geben, da Systeme dank SOA deutlich einfacher zusammengestellt werden können.
- Auf Basis von Standardlösungen werden vermehrt individuelle Lösungen angeboten und es wird mehr auf den Kunden eingegangen werden.
- Große Anbieter vervollständigen weiterhin ihre Funktionalitäten durch Aufkäufe und werden somit großflächig als indirekter Partner mit vielen betriebswirtschaftlichen Anwendungen in Kontakt kommen.
- Eine freie Möglichkeit, eigene Entwicklungen in ein bestehendes System mit einzubringen, ist zwar mit einer service-orientierten Landschaft einfacher, jedoch im Rahmen des Customizings noch nicht realistisch.

5.2.4 Integration

Aus den Interviews gewonnene Erkenntnisse:

- Unübersichtlich viele Schnittstellen und Dokumentenstandards werden in den Systemen berücksichtigt.
- Die meisten ERP-Systeme sind immer noch, zumindest in den primären Aktivitäten, eng miteinander integriert.
- Berechtigungskonzepte sind in jedem der Systeme umgesetzt.

Die Anzahl der verfügbaren und für das eigene System notwendigen Schnittstellen ist immer noch unübersichtlich groß, wie die Befragung gezeigt hat. Auch auf Prozessebene gibt es keine allgemeinen Standards, die eine Integration erleichtern könnten.

Individuelle Projekte zur Anbindung (vgl. [Wölfle & Schubert, 2007] oder [Wölfle & Schubert, 2005]) bleiben vorerst die Regel. Trotzdem wurde eine Integrationsmöglichkeit am häufigsten genannt: Web Services. Sie scheinen im Zusammenhang mit der weiten Verbreitung von XML als Integrationswerkzeug für Applikationen eine größere Bedeutung zu gewinnen. Dabei spiegelt der Einsatz von Web Services nicht primär eine service-orientierte Architektur als Grundkonzept wider, sondern dient allein dem Zweck des Dokumentenaustausches bzw. als Funktionsschnittstelle zum Anwendungssystem.

.NET basiert hinsichtlich der Kommunikation vollständig auf Web Services, J2EE unterstützt diese ebenfalls. Deren Einsatz und Bedeutung wird also mit großer Wahrscheinlichkeit zunehmen. Ob sie die Probleme der Transaktionssicherheit, der Übertragungssicherheit, der Skalierbarkeit etc. wie in Kapitel 3.3.2.4 angesprochen, dauerhaft überwinden können, bleibt abzuwarten. Allerdings sind Bestrebungen im Gange, die entsprechenden Lücken im Standard auszufüllen (vgl. [innoQ Deutschland GmbH & AG, 2007]).

Wie in Kapitel 5.2.3 bereits angesprochen wurde, steht aufgrund der vielen funktionalen Abhängigkeiten in den betrachteten Systemen momentan eine komponenten-orientierte Architektur der ERP-Systeme nicht im Mittelpunkt der Entwicklung. Hier sind vielmehr service-orientierte bzw. service-fähige Systeme in der Entstehung bzw. Umwandlung begriffen. Die gängigen Middleware-Frameworks wie J2EE und .NET zusammen mit einem vollwertigen Berechtigungskonzept, das eine Rollenverteilung für Fremdnutzer möglich macht, liefern sogar die Infrastruktur, um eine Komponenten-Orientierung konsequent umzusetzen. Vielleicht wird durch den momentan stattfindenden Sichtwechsel auf die Systeme ebenfalls ein Sichtwechsel hin zur lose gekoppelten Modulstruktur angestoßen.

Die internen Abhängigkeiten, die vielen Systemen noch zu Eigen ist, lassen jedoch eher vermuten, dass zwar eine Service-Struktur aufgebaut, eine lose Kopplung der Module jedoch nicht angestrebt wird. Ein Interviewpartner sagte: „Wir können nicht SOA, wir sind SOA.“ (WE1)

Die Integration zu fremden Systemen und eine erleichterte Prozesslandschaft stehen auf einer Seite, das Bestreben, sein komplettes System an einen Kunden zu verkaufen, auf der anderen.

Prognose:

- Web Services werden in Zukunft vermehrt eingesetzt werden, da sie eine einfache und schnelle Anbindung nach außen realisieren können.
- Die technischen Möglichkeiten (Berechtigungskonzept, Middleware-Frameworks, SOA, etc.) für eine lose Kopplung der Funktionen sind zwar gegeben, aber noch nicht so weit verbreitet. Zudem wird man mit dem Anspruch eines Komplettanbieters und im Sinne einer einfacheren Erstellung des Systems weiterhin mehr integrierte als komponenten-orientierte Systeme sehen.

5.3 Technologie

In diesem Abschnitt wird vorgestellt, welche technischen Grundlagen, Eigenschaften und Werkzeuge die Leistungs- und Integrationsfähigkeit von ERP-Systemen in den kommenden Jahren unterstützen werden.

5.3.1 Systemintern

Aus den Interviews gewonnene Erkenntnisse:

- Microsoft gelingt es, mit seinem Betriebssystem sowohl im Client- als auch im Serversegment die größte Verbreitung zu erlangen.
- Relationale Datenbanken sind am beliebtesten.

- Java erfreut sich zusammen mit Eclipse einer großen Beliebtheit.
- Freie Entwicklungswerkzeuge und Anwendungen sind im alltäglichen Gebrauch anzutreffen.

Windows hat in den vergangenen Jahren als Betriebssystem für Server im Vergleich zu Linux bzw. Unix einen beachtlichen Vorsprung gewonnen [Discherl, 2007], obwohl die Verbreitung noch nicht so hoch wie am PC-Markt ist. Trotzdem verwenden nur neun Systeme Windows exklusiv. Plattformunabhängigkeit und der breite Einsatz von Java in den Systemen deuten auf Portabilität hin, die vielen als Qualitätsmerkmal wichtig zu sein scheint. Eine marktdominierende Stellung von Microsoft im Betriebssystembereich für Server scheint sich möglicherweise langfristig abzuzeichnen, momentan ist jedoch für Hersteller die Flexibilität entscheidend.

Im Bereich Datenbanken dominieren weiterhin die relationalen DBs den Markt. MS-SQL wird im Rahmen der Microsoft Strategie weiter an Boden gewinnen, aber im Vergleich zu Oracle oder IBM keine Marktdominanz erreichen. Java hat sich in den vergangenen Jahren als Sprache für serverseitige Applikationen verdient gemacht und wird mit zunehmender Zeit nun auch für die Client-Seite interessant. Trotzdem werden weiterhin in vielen Systemen mehrere Programmiersprachen verwendet, allein aus dem Grund, dass ältere Programmfragmente nicht ausgetauscht werden sollen oder können, ohne das ganze System zu destabilisieren. Dies wird durch die Durchdringung des SOA Gedankens noch gefördert, denn dadurch haben ältere Applikationen dank der Service-Schnittstellen wieder eine Existenzberechtigung neben neueren Applikationen.

Open Source Werkzeuge, seien es Entwicklungsumgebungen wie Eclipse oder evtl. ganze Applikationen (z.B. CRM), werden im Umfeld der ERP-Systeme immer wichtiger und kommen vermehrt zum Einsatz. Die Notwendigkeit, eine eigene Entwicklungsumgebung zu entwerfen evtl. sogar mit eigener Programmiersprache, ist heutzutage aufgrund der umfangreichen Funktionen, die aktuelle Open Source Produkte anbieten, nicht mehr gegeben.

Prognose:

- Microsoft wird den hohen Anteil der Verbreitung seiner Betriebssysteme halten wenn nicht sogar weiter ausbauen können. Allerdings werden weiterhin die meisten Systeme mehrere Betriebssysteme unterstützen.
- Im Bereich der Datenbanken ist kein allgemeiner Trend abzusehen, allerdings scheint sich SQL als meist genutzte Datenbank etabliert zu haben.
- Java wird sich im Zuge der web-basierten Lösungen weiter verbreiten.
- Freie Entwicklungswerkzeuge und Anwendungen sind durchaus ausgereift und bieten viele Funktionen, die in Zukunft vermehrt genutzt werden.

5.3.2 Systemextern

Aus den Interviews gewonnene Erkenntnisse:

- XML hat sich als Metasprache für Geschäftsdokumente in jedem System neben EDIFACT etabliert.

- Web Services sind sowohl als internes als auch externes Integrationswerkzeug im Einsatz.
- Terminallösungen sind immer noch im Einsatz und bekommen durch die klassischen Browser Konkurrenz.

XML hat sich als Grundlage für Datenaustauschformate neben EDIFACT und den anderen branchenspezifischen EDI-Standards etabliert, wird diese jedoch in naher Zukunft nicht ersetzen. XML bietet neben den Eigenschaften, Geschäftsdokumente zu beschreiben (vgl. Kapitel 3.3.2.3), auch die komfortable Möglichkeit, direkt im Browser ohne eine aufwendige Konvertierung dargestellt werden zu können.

XML als Grundlage für geschäftliche Dokumente ist im Vergleich zu den klassischen EDI-Standards Internet-optimiert, ein offener und weltweiter Standard, leicht zu erlernen, leicht verständlich und über die flexible Schema-Definition an jeden Verwendungszweck leicht anpassbar [Priesnitz und Teille, 2003, S. 19]. Dennoch haben sich in der Vergangenheit so wie im klassischen EDI-Bereich mehr als 200 verschiedene, teilweise proprietäre Standards gebildet [Angele, 2003, S. 329], die einen Austausch von Informationen untereinander ebenso wieder erschweren wie früher. XML ist ein allgemein verwendbares, syntaktisches Austauschformat, eine allgemeine Standardisierung setzt die Sprache jedoch nicht um. Eine einheitliche Standardisierung wird zwar angestrebt und ist bereits teilweise (XML) bzw. komplett (EDIFACT) verabschiedet, trotzdem müssen diese EDI-Standards unterstützt und gepflegt werden. Für die nahe Zukunft ist eine Koexistenz von XML-basierten Standards und den älteren Dokumenten am wahrscheinlichsten.

Besonders mit der großen Verbreitung von Web Services scheint sich XML endgültig als Grundlage für Geschäftsdokumente etabliert zu haben. Der jetzt schon breite Einsatz im Hinblick auf Funktionsintegration wird sich weiter ausdehnen, ebenso der reine Dokumentenaustausch (vgl. 4.1.4). Noch scheint Citrix in der Anwendung mit 50 Prozent eine breite Akzeptanz zu finden. Doch nur sieben Anbieter verwenden die Terminallösung exklusiv.

Browser haben als Darstellungsmittel dagegen eine hohe Akzeptanz. Sie werden weiterhin vermehrt als Thin-Client Lösung zum Einsatz kommen, denn die Entwicklung der Systeme hin zu einem web-fähigen und web-orientierten Anwendungspaket involviert automatisch die Browser als mögliches Zugangsportale zu der Software. Web-basierte Systeme werden einen größeren Marktanteil gewinnen: „Die Entwicklung hin zu flexiblen und über das Web erreichbaren Anwendungen wird fortschreiten.“ (KB1)

Prognose:

- XML-basierende Dokumente werden vermehrt eingesetzt zusammen mit einer größeren Verbreitung von Web Services. Sie werden die enorme Anzahl an unterschiedlichen Standards nicht ersetzen können, allerdings bieten sie eine gute Transportmöglichkeit zwischen verschiedenen Systemen.
- Web-basierte Systeme werden vermehrt entstehen bzw. bereits existierende Systeme werden eine Web-basierte Schnittstelle hinzufügen.

5.4 Betreibermodell

Welche Betreibermodelle sind in Zukunft für den Kunden möglich bzw. welche werden wirklich zum Einsatz kommen, wird in diesem Abschnitt vorgestellt.

Aus den Interviews gewonnene Erkenntnisse:

- Das klassische In-House Betreibermodell dominiert weiterhin, SaaS gewinnt jedoch an Bedeutung, sowohl gegenüber der In-House Variante als auch gegenüber ASP.
- Mandantenfähigkeit ist bei allen gegeben.
- Die Kundennähe und der direkte Kontakt zum System des Kunden sind allgemeinüblich.

Alle Anbieter bieten die klassische In-House Lösung als lizenzbasiertes Betreibermodell an. Dieses Standbein wird in Zukunft auch niemand der Anbieter aufgeben wollen. Ein Interviewpartner meinte sogar: „In-House Lösungen wird der deutsche Mittelstand weiterhin präferieren.“ (WE3)

Trotzdem ist die Bereitschaft da, sich neuen Betreibermodellen zu öffnen, und wird neben dem eigenen Engagement auch vom Marktdruck her getrieben werden. Dies gilt vor allem für den Bereich der Standard-Softwarehersteller. Dort werden Initiativen wie die von SAP mit „Business by Design“ durchaus für einen harten Konkurrenzkampf sorgen.

Im Vergleich zu den 90er Jahren, in denen ASP zum ersten Mal mit wenig Erfolg zum Einsatz kam, hat sich die Technologie (vollständige Mandantenfähigkeit in allen Systemen, Breitbandzugang, Service-Orientierung) weiterentwickelt. Im Bewusstsein der Anbieter und Kunden jedoch gilt eine Mietlösung teilweise noch als nicht sicher [Lixenfeld, 2008, S. 1]. Dabei scheint die Sorge, dass die Software evtl. nicht von dem Anbieter selbst gepflegt wird, wie es in frühen ASP-Zeiten teilweise der Fall war, unbegründet. Wartungen finden fast ausschließlich von Anbieterseite aus statt und dort werden die Daten sicher vorgehalten, nur im Vertrieb werden zunehmend Partner bemüht. Zusammen mit einem ergonomischen und leicht verständlichen User Interface können SaaS Modelle attraktiver für kleinere Unternehmensgrößen sein.

Sie können nicht nur Software hosten, sie können auch leicht über ein Portal andere Kunden miteinander in Verbindung treten lassen [David et al., 2003, S. 69]. Eine Studie der FHNW hat ermittelt, dass Anwender in den kommenden Jahren von der In-House Variante abrücken werden [Felley, 2007, S. 5]. Allen Widersprüchen der vor allem kleineren Anbieter zum Trotz wird sich ASP/SaaS als neue, feste Option im Markt etablieren.

Nischenanbieter fühlen sich in ihrer Rolle eines hoch-spezialisierten Anbieters sicher und von den momentanen Entwicklungen nicht bedroht. Standardsoftware-Anbieter bieten meist ein breiteres Spektrum an Funktionen an als von einem kleinen Unternehmen genutzt werden kann. Ein Anbieter war der Ansicht: „Wenn die Anzahl der eingesetzten Module die Anzahl der User überschreitet, sollte man nachdenklich werden.“ (PA3)

Das Geschäftsmodell der Nischenanbieter, genau die benötigten Funktionen zu liefern und von Fall zu Fall noch individuelle Entwicklungen maßgeschneidert dem Kunden anzupassen, ist für einen großen Anbieter schwierig, mit einem Portfolio an Standardfunktionen umzusetzen. Die Klientel wird sich weiterhin auf eine enge Partnerschaft und die persönlichen Anpassungswünsche verlassen können, so der Glaube der Nischenanbieter.

Allerdings wird es auch in diesem Segment zu einer stärkeren Konkurrenz kommen. Einige Anbieter, darunter SAP und auch Microsoft, wissen um die Qualität einer individuellen Betreuung der Kunden und wollen verstärkt auf ihr Partnernetzwerk und die von den Partnern

auf Basis des allgemeinen Standards angebotenen Individualleistungen setzen. Geht man von den großen Systemen aus, die bislang auf dem Markt, vor allem bei Konzernen, so erfolgreich waren, so ist diese Vision unrealistisch. Betrachtet man allerdings die neuen Anwendungssysteme, die modular aufgebaut und service-orientiert sind, so scheint ein solches Geschäftsmodell deutlich realistischer zu sein: „Es wird eine Renaissance der Individualentwicklungen geben.“ (BI5)

Prognose:

- SaaS wird sich von einem so genannten Buzzword (Modewort) zu einem ernstzunehmenden Betreibermodell weiterentwickeln. Der Einsatz wird nicht für jedes Unternehmen sinnvoll sein, aber für Kunden ohne ausgefallene Sonderwünsche wird eine Mietlösung ein attraktives Konkurrenzangebot sein.
- Kundennähe und die Fokussierung auf individuelle Wünsche waren und bleiben ein wichtiges Instrument zur Kundenzufriedenheit, sowohl für Nischenanbieter als auch die Softwarekonzerne.

5.5 Markt

Hier werden die zukünftigen Potenziale für den Aspekt Markt vorgestellt, z.B. welche Eigenschaften sind für die Anbieter im Vergleich zur Konkurrenz entscheidend, welchen Einfluss haben die Kunden auf ihr Produkt, oder können Open Source Anbieter zur ernsthaften Konkurrenz heranwachsen und wie sehen die Anbieter ihre und die Zukunft des Marktes.

5.5.1 Anwender

Aus den Interviews gewonnene Erkenntnisse:

- Der technologische Vorsprung dominiert das Auftreten der Anbieter nach außen hin.
- Der Einfluss der Kunden auf das fertige Produkt ist bei fast allen Anbietern sehr groß.
- Die Kommunikationswege mit dem Kunden sind nicht entscheidend für den Einfluss auf das Produkt oder die Qualität des Produkts.

Technologie ist für die meisten Anbieter das Kriterium, mit dem man sich an Konkurrenten messen kann. Aktuell sind es eben auch technologische bzw. vom Marketing in die Technologie eingeordnete Begriffe wie Web Services, SOA oder auch SaaS, die die Diskussion beherrschen. Kundennähe existiert bei allen Anbietern und wird sorgsam gepflegt, trotzdem zählt sie nicht als mögliches Werbemittel. Sie scheint vielmehr als natürliches Nebenprodukt aus der Anbieter-Kunden Beziehung angesehen zu werden.

Sollten sich Mietlösungen wie SaaS weiter verbreiten, könnte diese persönliche Beziehung in Gefahr geraten. Der Fokus gerade auf den Markt der kleinen und mittelgroßen Kunden beinhaltet auch Anwender, die mit dem Umgang einer betriebswirtschaftlichen Software keine oder kaum Erfahrung haben oder die Software nur selten nutzen. Ein Anbieter hat sein Interesse an dieser Klientel in Worte gefasst: „Wir wollen die Gelegenheitsuser abholen.“ (KS2)

Das deutet auf eine einfachere und leichter zu bedienende Software hin, deren Erfolgspotenzial gerade in diesem Segment hoch sein dürfte. Flexibilität und Branchenausrichtung werden nur von etwa einem Drittel der Anbieter als Unterscheidungsmerkmal gesehen. Dabei ist Flexibilität im Hinblick auf die Entwicklung zu web-basierten Lösungen oder service-orientierten Architekturen eine wichtige, wenn auch nicht präzise definierte Eigenschaft. Die Branchenausrichtung hingegen deutet vor allem weiter in Richtung einer engeren Beziehung zum Kunden. Es wird nicht mehr ausreichend sein, nur eine neue Technologie oder neue Systemeigenschaften einzuführen, die Ausrichtung der Anbieter auf ihre Kunden, zumindest in bestimmten Nischenbranchen, wird sich zu einem wichtigen Unterscheidungsmerkmal herausbilden: „Ein Hauptmerkmal ist die Zusammenarbeit auf Augenhöhe mit den Kunden.“ (RL1)

Prognose:

- Individuelle Kundenbetreuung wird mit einer steigenden Anzahl an standardisierten Mietlösungen zu einem wichtigeren Unterscheidungskriterium als heute.

5.5.2 Anbieter

Aus den Interviews gewonnene Erkenntnisse:

- Alle Branchen können von Anbietern bedient werden, die Produktion ist klassischer Weise immer noch am häufigsten vertreten.
- Open Source Anbietern wird eine betriebswirtschaftliche Expertise aberkannt.
- Fortschreitende Konsolidierung im Markt und neue Technologien werden hauptsächlich als die zukünftigen Einflussfaktoren im Markt wirken.

Die Produktion ist immer noch die Zielgruppe, die von den meisten Anbietern unterstützt wird. Handel und vor allem Dienstleistung liegen zurück. Trotzdem gibt es für den Handels- als auch Dienstleistungssektor Angebote, die nur exklusiv auf ihre jeweilige Branche ausgerichtet sind. Ob besonders ein Branchenzweig in Zukunft mehr oder weniger Unterstützung erfahren wird, ist nicht zu klären.

Open Source Anbieter stellen in den Augen fast aller proprietärer Anbieter keine Konkurrenz dar, weil sie vor allem den betriebswirtschaftlichen Hintergrund nicht beherrschen, wie viele versichern: „Im OS Umfeld hat man technologisch viel Potenzial, fachliches Know-how ist jedoch deutlich schwieriger zu finden“. (RG1)

Auf dem Markt für mittelgroße Unternehmen scheinen Open Source Anbieter keine großen Chancen zu haben. Im kleinen Umfeld jedoch (15-25 User) können und werden sie mit der Zeit zu einer ernstzunehmenden Konkurrenz werden: „Proprietäre Anbieter von Standard-Software gelangen nicht in die Nischenmärkte, die einen hohen Individualisierungsbedarf haben.“ (BI3)

Die Zukunft sehen etwa drei Viertel der Unternehmen für sich in wachsenden Marktanteilen. Dies steht jedoch zu derselben Aussage im Widerspruch, dass eine fortschreitende Marktkonsolidierung erwartet wird. Diese Konsolidierung ist jedoch wahrscheinlicher, denn der Druck durch Global Player wie SAP oder Oracle, die in den Markt für KMU eintreten wollen, wächst: „In der Mitte des Marktes werden viele Anbieter verschwinden, da sie von den „Großen“ geschluckt werden.“ (FR4)

Die SOA und deren Möglichkeit für eine einfache Anwendungsintegration ist einer der Gründe für die momentane Marktkonsolidierung [Hill, 2007]. Allein Oracle hat innerhalb von 48 Monaten 41 Unternehmen akquiriert. Das Szenario, dass sich einige Anbieter im KMU Bereich zu so genannten ERP-Clustern zusammenschließen wollen, scheint somit nicht unrealistisch zu sein, um sich gegen die härter werdende Konkurrenz zu wappnen, wie es einige Anbieter skizzieren: „Es wird in Zukunft wichtiger sein, sich mit anderen zu vernetzen.“ (EG5)

Neue Technologien können diesen Prozess sogar beschleunigen, wie die Verbreitung von Web Services im Zusammenhang mit SOA gezeigt hat.

Prognose:

- Die Konsolidierung am Markt wird weiter voranschreiten. Größere Unternehmen werden zu einen kleinere, spezialisierte Anbieter von Business Software aufkaufen, um ihr Portfolio zu erweitern. Kleinere Anbieter dagegen werden sich zu größeren Gruppen zusammenschließen.
- Open Source Anbieter werden aufgrund ihres lizenzfreien Geschäftsmodells besonders im KMU-Bereich weiterhin eine Alternative bieten.
- Der Druck der Global Player wird weiter steigen, da zum einen neue und auf SaaS-Modellen basierende Systeme im KMU-Bereich eine ernstzunehmende Konkurrenz darstellen werden, zum anderen eine individuelle Kundenbetreuung durch das gut ausgebaute Partnernetzwerk möglich ist.
- Der Markt bleibt weiterhin in der Lösungsvielfalt heterogen, da auch ganz unterschiedliche Anforderungen von Kundenseite her bedient werden müssen. Allerdings wird die Ausrichtung auf den Kunden an Bedeutung zunehmen.

6 Kritische Zusammenfassung und Ausblick

In den zurückliegenden Kapiteln wurde dargelegt, welche künftigen Anforderungen an ERP-Systeme (Fokus auf Deutsche ERP-Anbieter) in den kommenden Jahren zu erwarten sind. Dafür wurde anlehnend an die qualitative Forschungsmethodik von Mayer [Mayer, 2004, S. 29] zuerst eine Betrachtungsebene auf ERP-Systeme modelliert, die vier Aspekte umfasst: Architektur, Technologie, Betreibermodell und Markt.

Anschließend wurden die vier Bereiche weiter unterteilt und zu jedem Abschnitt die theoretischen Grundlagen erläutert. Die darauf folgende Befragung von 32 ERP-Anbietern aus Deutschland lieferte die Datenmenge, die ausgewertet wurde, und auf deren Basis eine Prognose der zukünftigen Entwicklungen stattfand. Bei 130 angefragten Unternehmen und 32 Antworten betrug die Rücklaufquote 24,6 Prozent. Die Menge an Unternehmen ist somit nicht vollständig, aber hinsichtlich der Branchenverteilung, Unternehmensgröße und inhaltlichen Relevanz ergibt sich eine aussagekräftige Stichprobe. Die drei größten ERP-Anbieter in Deutschland (SAP, Microsoft und Oracle) haben ebenso an der Befragung teilgenommen, wie zwei Open Source Vertreter, eine Reihe von Nischenanbietern und Standardsoftware-Anbietern jeglicher Größe.

Im Folgenden sollen zu jedem Aspekt des theoretischen Modells je eine kurze Zusammenfassung und anschließend weiterführende Forschungsfragen vorgestellt werden.

6.1 Ausblick: Architektur

Die Befragung hat gezeigt, welche Konzepte und Systemarchitekturen den betrachteten ERP-Systemen zugrunde liegen. Zudem wurde die Prozessgestaltung ausgiebig betrachtet, der Funktionsumfang ermittelt und Integrationsansätze beurteilt.

6.1.1 Ausblick: Planungsebene

Besonders in einer durchgängigen Prozessgestaltung und der Umsetzung von Prozessstandards liegen noch Defizite, die einer intensiven Weiterentwicklung bedürfen. Die Modellierung sollte nicht bei einem einfachen Visualisierungsbemühen stehen bleiben, sondern zusammen mit einem service-orientierten Anwendungssystem eine Neuordnung von Funktionen über die Planungsebene hinweg ermöglichen. Es gibt bereits entsprechende Ansätze (BPMN, BPEL, etc.), die jedoch nur von einem sehr geringen Teil der Befragten angewandt werden. Hinzu kommen standardisierte Prozesse für eine geschäftsübergreifende Prozessmodellierung, deren Umsetzung ebenfalls kaum vorhanden ist.

Es gilt außerdem, die Nutzer eines Modellierungswerkzeuges für die Potenziale zu sensibilisieren. Hier sind die Prozessverantwortlichen im Unternehmen selbst gefordert, nicht vor den technischen Herausforderungen zurückzuweichen, sondern den Nutzen zu erkennen. Ob die breite Masse der Betroffenen dies bereits getan hat, war in dieser Arbeit nicht zu klären. Hier bieten sich weitere Möglichkeiten zu forschen, z.B. hinsichtlich eines intuitiveren und leichter verständlichen GUI (Graphical User Interface), um den Einstieg in die Modellierung zu erleichtern.

6.1.2 Ausblick: Systemarchitektur

SOA ist noch nicht bei allen Befragten eindeutig definiert. Der Service-Gedanke ist dafür zumindest als grobes Konzept allen Befragten gleich. Aufbauend auf Services können

Legacy-Systeme leichter wiederverwendet werden, neue Betreibermodelle betrieben und Zukäufe leichter in das bestehende System integriert werden. Trotzdem bedarf es einer disziplinierten Service-Verwaltung, um besonders für einen großflächigen Einsatz Eindeutigkeit, Sicherheit etc. zu gewährleisten. Dies könnte Gegenstand weiterer Forschung sein.

Betrachtet man die n-Tier Architektur, so finden sich normalerweise drei Bausteine wieder: Client, Applikationsserver und Datenbank. Angesichts der Service-Orientierung, die sich weiter ausbreiten wird, könnte eine weitere Betrachtungsebene eingeführt werden, welche die Systeme nicht nach der n-Tier Architektur klassifiziert. Stattdessen kann nach einem SOA-Modell wie es z.B. von Liebhart in [Liebhart, 2007, S. 66] vorgeschlagen wird, eine Begutachtung der ERP-Systeme stattfinden.

6.1.3 Ausblick: Funktionsarchitektur

In den vergangenen Kapiteln wurden die wertschöpfenden Tätigkeiten eines Unternehmens als Grundlage für die funktionale Betrachtung der untersuchten ERP-Systeme verwendet. Dies hat entsprechend des Modulangebots der Interviewpartner eine Klassifizierung nach Tätigkeiten erlaubt und lieferte eine gute Übersicht des allgemeinen Angebots. Im Verlauf der Befragung und der späteren Auswertung wurde klar, dass zukünftige Systeme in manchen Bereichen Defizite aufweisen können bzw. weitere Partnerlösungen hinzuziehen müssen. Ein Gegenstand weiterer Forschung kann in diesem Zusammenhang eine genauere Modulbetrachtung der ERP-Systeme sein, besonders derjenigen Systeme, die im SaaS-Bereich arbeiten, um ein genaueres Bild der Modulstruktur zu erlangen.

6.1.4 Ausblick: Integration

Hinsichtlich der B2B (Business to Business)-Integration gibt es noch zu viele unterschiedliche Schnittstellen, die eine einfache Anbindung an Fremdsysteme erschweren.

Besonders auf Prozessebene sind kaum allgemein anerkannte Standards im Einsatz. Hier ist noch viel Potenzial für weitere Entwicklungen. Auf Applikations- und Datenbankebene hingegen bieten Web Services eine Alternative. Sie können zwar nicht die darunter liegende Masse an Austauschstandards ersetzen, sie aber zumindest im Sinne eines Service auf einheitliche Weise zur Verfügung stellen.

Bei der internen Integration (Enterprise Application Integration - EAI) bleibt die konsequente Entwicklung aus einer auf Applikationsebene beschränkten Integration, bei der die Berücksichtigung des Kontrollflusses sowie der übergeordneten Geschäftsprozesse nicht stattfindet [Allweyer, 2005, S. 342], zu einem BPM (Business Process Management) eine der wichtigsten Aufgaben für ERP-Anbieter.

Web-basierte Systeme und die Zusammenführung von mehreren Informationsquellen als Entscheidungsunterstützung führen zu einem weiteren, momentan diskutierten Thema: Mashups [Dornan, 2007].

Sie sind eine weiter entwickelte Form der Web-Integration und präsentieren durch eine Kombination bereits vorhandener Daten neue Inhalte [Schumann, 2008]. Sie können z.B. Entscheidungswege durch das Zusammenführen von verteilten Informationen erheblich beschleunigen. Inwieweit dies für Unternehmen praktikabel ist, muss weitere Forschung zeigen. Ein Anbieter glaubt an die ernstzunehmende Bedeutung dieser Entwicklung: „Die Zukunft liegt in Collaborative Work.“ (KS3)

6.2 Ausblick: Technologie

Der Fokus lag bislang auf den Systemen selbst und ihren technischen Eigenschaften und Neuerungen. Cegielski et al. betrachten zukünftige technische Neuerungen aus Sicht der Anwender und deren Konfrontation mit der Entscheidung, ob und wie man solche Neuentwicklungen wie z.B. XML oder Web Services in seiner gegenwärtigen IT-Landschaft einsetzen soll oder nicht [Cegielski et al., 2005]. Der gegenwärtige Wandel in der ERP-Landschaft erschwert natürlich die Entscheidung, ob evtl. ein neues System für das Unternehmen notwendig ist, um wettbewerbsfähig zu bleiben. Cegielski et al. schlagen in diesen Zusammenhang drei Aspekte vor, die es zu beachten gilt, wenn man sich mit neuen Technologien auseinandersetzt [Cegielski et al., 2005, S. 117]:

Zuerst sollte in Betracht gezogen werden, ob die Technologie für das Unternehmen einen echten Mehrwert für die Wettbewerbsfähigkeit birgt. Ist dies der Fall, so muss der Verantwortliche die Nutzerbasis, die mit dem System täglich arbeiten und die mit der neuen Entwicklung evtl. interagieren, berücksichtigen.

Zuletzt sollte untersucht werden, ob und wie die neue Technologie nicht nur in der gegenwärtigen sondern auch in einer langfristigen Unternehmensstrategie Verwendung finden kann.

Erst, wenn diese auf Geschäftsebene angesiedelten Entscheidungen getroffen wurden, kann über die technische Ebene der Einführung gesprochen werden.

6.2.1 Ausblick: Technologie - Systemintern

Um eine genauere Klassifizierung des Einsatzes der verwendeten Technologien zu erhalten, könnte man in Zukunft noch eine Ebene tiefer gehen und z.B. im Falle von Windows als Betriebssystem die unterschiedlichen Versionen berücksichtigen, die Microsoft sowohl für Clients als auch Server anbietet. Dies gilt ebenso für den Bereich der Datenbanken, bei dem eine Aufteilung z.B. für SQL-Datenbanken in MySQL und MSSQL nicht vorgenommen wurde.

Neueste Entwicklungen, die in dieser Arbeit nicht berücksichtigt werden konnten, sind z.B.:

- Die mögliche Verlagerung der Systeme vom klassischen Betriebssystem hin zu einer Virtualisierungslösung (vgl. [Jarzyna, 2007]; [Hülsebusch, 2008])
- Vorzüge der Virtualisierung für KMU (vgl. [Parbel, 2008])
- Die Einführung eines neuen Business-Servers von Microsoft (vgl. [Siering, 2008])

Sie versprechen, neue Aspekte hinsichtlich der Infrastruktur und allgemeinen Systemgestaltung zu geben. Außerdem scheint sich ein neuer Trend abzuzeichnen, der die vorkonfigurierte Verbindung von Software und Hardware beinhaltet. Zielgruppe sind im Besonderen kleine und mittelständische Unternehmen (vgl. [Schüler, 2008]).

6.2.2 Ausblick: Technologie - Systemextern

XML hat sich als grundlegende Sprache für Geschäftsdokumente etabliert. Allerdings existieren bereits wie im klassischen EDI-Bereich eine Vielzahl von unterschiedlichen Standards. Die Aufgabe besteht darin, ebenso wie bei den älteren Transaktionsstandards, einen einheitlich anerkannten Standard zu finden.

Web Services vermögen es, zwischen unterschiedlichen Systemen eine Kommunikation zu ermöglichen. Allerdings ist dies in Bezug auf Web-Reichweite nur mit einem eindeutigen Verzeichnisdienst möglich. Zudem kommen Sicherheitsaspekte wie Verschlüsselung oder Transaktionssicherheit hinzu. Die Bemühungen müssen fortgesetzt werden, die bereits heute existierenden Standards ausreifen zu lassen.

6.3 Ausblick: Betreibermodell

Neben den bereits bekannten Betreibermodellen In-House und ASP ist nun auch SaaS als neue Variante zu nennen. Zwar handelt es sich wie bei ASP um ein Hosting-Modell, allerdings unterscheidet es sich in mehreren Eigenschaften deutlich von einem ASP-Modell.

Es existieren drei klassische Vorteile von Standardsoftware gegenüber Individualsoftware [Gluchowski et al., 2008, S. 11]:

- 1. Kostengünstigkeit trotz hoher Beschaffungskosten (keine eigenen Softwareentwicklungskosten)
- 2. Zeitersparnis (schnelle Beschaffung und Anpassung im Vergleich zur Entwicklung von Individualsoftware)
- 3. Zukunftssicherheit (seriöse Anbieter nutzen allgemein gültige Standards und entwickeln ihre Produkte ständig weiter)

SaaS-Angebote bieten in jedem der drei Bereiche weiteres Potenzial. Sie sind kostengünstiger aufgrund des reinen Mietpreises, den man für die Nutzung zahlt. Sie ersparen mehr Zeit, da die Installation der Software bereits abgeschlossen ist und sie bieten durch die ständige Wartung und Weiterentwicklung ein großes Zukunftspotenzial. Es sprechen jedoch zwei Punkte gegen einen großflächigen Einsatz:

Zum einen fehlt das Kundenvertrauen in eine reine Mietlösung, da Sicherheitsaspekte und die Herausgabe der Unternehmensdaten als kritisch angesehen werden [Lixenfeld, 2008, S. 4]. Zum anderen ist eine individuelle Anpassung des Systems nicht oder kaum möglich.

Diese beiden Gesichtspunkte bedürfen großer Aufmerksamkeit. Im Hinblick auf mögliche Individualanpassungen wäre es interessant zu untersuchen, inwieweit das Partnernetzwerk mit Speziallösungen (sowohl technologisch als auch organisatorisch) eine standardisierte SaaS-Software für bestimmte Kunden anreichern könnte.

6.4 Ausblick: Markt

Im Folgenden werden eine Auswahl an möglichen und zukünftigen Trends im Bereich Markt aufgezeigt und weiterführende potentielle Forschungsansätze angesprochen.

6.4.1 Ausblick: Anwender

Es gibt noch einen wichtigen Faktor, wenn man ERP-Systeme betrachtet: den Nutzer. Neben all den organisatorischen und technischen Verbesserungen müssen vor allem die Nutzer eines solchen Systems sowohl in der Anwendung geschult als auch auf die Neuerung positiv eingestellt werden. Das Kundenvertrauen zu erlangen, ist die wichtigste Aufgabe. Das Potenzial der Business Software steht und fällt mit der Benutzerakzeptanz [Wölflé & Schubert, 2006, S. 15].

Die Automatisierung, Standardisierung und die mögliche Einschränkung der Entscheidungsgewalt können sich negativ auswirken. Kwahk [Kwahk, 2006] beschreibt dazu eine Studie hinsichtlich der Nutzerakzeptanz. Weitere Forschung in dieser Richtung ist angebracht.

6.4.2 Ausblick: Anbieter

Oracle besitzt mit der Eigenentwicklung Oracle Fusion Middleware eine eigene Plattform zur Integration. Jedoch herrscht die allgemeine Annahme vor, Oracle sei primär ein Applikations- und Datenbankanbieter. Das mittel- bis langfristige Ziel, auf dem Middlewaremarkt einen größeren Marktanteil zu erzielen, hat Oracle trotz der Eigenlösung dazu bewogen, den Erwerb von BEA Systems Anfang 2008 zu finalisieren, einen großen Anbieter für Middleware-Technologie [Hill, 2008].

Zukünftige strategische Entscheidungen wie diese können selbstverständlich in dieser Arbeit nicht skizziert werden, allerdings ist der Trend einer fortschreitenden Markt-Konsolidierung deutlich erkennbar. Besonders die größeren Anbieter versuchen, über eine gezielte Einkaufspolitik fehlende Kompetenzen zu erwerben, insbesondere im kleinen und mittelständischen Bereich. Die Ambitionen sind hoch gesteckt. SAP z.B. will bis 2010 den Anteil der Mittelstandskunden von 65 Prozent auf 75-80 Prozent erhöhen [Wilkens, 2008].

6.5 Fazit

Die zukünftigen ERP-Systeme zeichnen sich, wie auch die gegenwärtigen, durch eine große Vielfalt an unterschiedlichen Systemkonzepten aus. Die Anzahl der Anbieter wird zwar in den kommenden Jahren sinken, aber nicht so dramatisch, wie es einige Interviewpartner prognostiziert haben. Die wichtigsten Trends sollen noch einmal stichpunktartig festgehalten werden:

- Eine weitere Annäherung zwischen BPM und SOA bzw. Ausbau der jeweiligen Ansätze ist zu erwarten.
- Fortschreitende Konsolidierung: große Anbieter werden ihr Portfolio um Kompetenzen erweitern (BI, Middleware etc.).
- Branchenspezifische Entwicklungen werden nicht mehr durch große Anbieter selbst, sondern durch das Partnernetzwerk entwickelt.
- Durch SOA gewonnene Flexibilität und Modularität ermöglichen einen effektiven Einsatz für KMU.
- Mashups besitzen im ERP Bereich ein hohes Potenzial.
- Es wird eine stärkere Fokussierung auf Kunden geben (z.B. verbesserte, intuitive GUI).

Abschließend bleibt zu bemerken, dass es in den kommenden Jahren zu Veränderungen kommen wird, diese sich aber positiv für den Kunden und auch auf die Systemgestaltung auswirken werden. Ein Interviewpartner hat diese Entwicklung treffend kommentiert:

„Für den Markt gilt: Evolution statt Revolution“. (KS4)

Literaturverzeichnis

- Acker, Hilmar; Atkinson, Colin; Dadam, Peter; Rinderle, Stefanie; Reichert, Manfred (2004): Aspekte der komponentenorientierten Entwicklung adaptiver prozessorientierter Unternehmenssoftware. In: (Turowski, 2004). S. 7-24, 2004.
- Alexander, Sascha (2007): IBM kauft Cognos: Konkurrenz für SAP, SAS und Oracle im Markt für Business Intelligence, Computerwoche.de, 12.11.07, 2007.
URL: http://www.computerwoche.de/produkte_technik/businessintelligence/1847569/
Letzter Zugriff: 09.03.08.
- Allweyer, Thomas: Geschäftsprozess-Management, Strategie, Entwurf, Implementierung, Controlling, Herdecke, Bochum: W3L-Verlag, 2005.
- Alt, Rainer; Österle, Hubert: Real-Time Business, Heidelberg: Springer Verlag, 2004.
- Amberg, Michael: Prozessorientierte betriebliche Informationssysteme: Methoden, Vorgehen und Werkzeuge zu ihrer effizienten Entwicklung, Berlin: Springer Verlag, 1999.
- Angele, Jürgen (2003): XML reicht nicht aus. In: (Mertens, 2003). S. 329-336, 2003.
- Arnold, Ken; Gosling, James: The Java Programming Language, California: Addison-Wesley, 2. Auflage, 1998.
- Asleson, Ryan; Schutta, Nathaniel T.: Foundations of Ajax, Berkeley, CA (USA): Apress, 2006.
- Azvine, B.; Cui, Z.; Nauck, D. (2005): Towards real-time business intelligence, BT Technology Journal, Volume 23 (July), S. 214–225, 2005.
- Bach, Volker; Österle, Hubert: Customer Relationship Management in der Praxis, Berlin, Heidelberg: Springer Verlag, 2000.
- Balzert, Helmut: Lehrbuch der Software-Technik: Software-Entwicklung, Heidelberg, Berlin: Spektrum Akademischer Verlag, 2. Auflage, 2000.
- Balzert, Helmut: Lehrbuch: Grundlagen der Informatik, Konzepte und Notationen in UML2 Java 5, CSharp und C++, Heidelberg: Spektrum Akademischer Verlag, 2. Auflage, 2005
- Burger, Dan: SOA remains hard to define, but Projects on the Rise, itjungle.com, 14.01.08, 2008.
URL: <http://www.itjungle.com/tfh/tfh011408-story05.html>
Letzter Zugriff : 09.03.08.
- Campbell-Kelly, Martin: From Airline Reservations to Sonic the Hedgehog, A History of the Software Industry, Cambridge/Massachusetts: The MIT Press, 2004.
- Carl, Denny: Praxiswissen Ajax, Köln: O'Reilly, 2006.
- Cegielski, Casey G.; Reithel, Brian J.; Rebman, Carl M.: Emerging Information Technologies: Developing a Timely IT Strategy, Communications of the ACM, Volume 48 (Nr. 8), S. 113-117, 2005.
- Chamberlin, Don: DB2, Universal Database: Der unentbehrliche Begleiter, Bonn, Reading (Massachusetts): Addison-Wesley-Longman, 1999.
- Chen, Minder (2005): An analysis of the driving forces for Web services adoption, Information Systems and E-Business Management, Volume 3 (Number 3), S. 265-275, 2005.

- D'Anjou, Jim; Fairbrother, Scott; Kehn, Dan; Kellerman, John; Mc-Carthy, Pat: The Java Developer's Guide to Eclipse, Boston: Addison-Wesley, 2005.
- David, Julie Smith; e. McCarthy, William; Sommer, Brian S. (2003): Agility - The Key to Survival of the Fittest in the Software Market, Communications of the ACM, Volume 46(Number 5), S. 65–69, 2003.
- Daylami, Nozar; Ryan, Terry; Olfman, Lorne (2005): Determinants of Application Service Provider (ASP) Adoption as an Innovation, Proceedings of the 38th Hawaii International Conference on System Science, January, S. 259b, 2005.
- Deutschland, Statistisches Bundesamt: Klassifikation der Wirtschaftszweige (WZ), Wiesbaden: Statistisches Bundesamt Deutschland, 2003.
URL: <http://www.destatis.de/jetspeed/portal/cms/Sites/destatis/Internet/DE/Content/Klassifikationen/GueterWirtschaftsklassifikationen/Content75/KlassifikationWZ2003,templateId=renderPrint.psm>
Letzter Zugriff: 09.03.08
- Discherl, Hans-Christian: Servermarkt: Windows legt zu, Linux verliert, Unix liegt weit zurück, PC-Welt.de, 29.08.07, 2007.
URL: http://www.pcwelt.de/start/computer/netzwerk_server/news/92076/
Letzter Zugriff : 09.03.08.
- Dornan, Andy: Half-Baked or Mashed: Is Mixing Enterprise IT And The Internet A Recipe For Disaster? informationweek.com, 08.09.07, 2007.
URL: http://www.informationweek.com/software/showArticle.jhtml;jsessionid=H10CMPC3J_SQMWQSNLPCKH0CJUNN2JVN?articleID=201804743&pgno=1&queryText=saas+asp
Letzter Zugriff : 09.03.08.
- Eberhart, Andreas; Fischer, Stefan: Web Services: Grundlagen und praktische Umsetzung mit J2EE und .NET, München: Carl Hanser Verlag, 2003.
- Engel, A.; Koschel, A.; Tritsch, R.: J2EE kompakt - Enterprise Java, Konzepte und Umfeld, Heidelberg: Spektrum Akademischer Verlag, 2002.
- Felley, Gabriel (2007): Welchen Anforderungen müssen Business Software Systeme in den kommenden 3-5 Jahren Stand halten? Arbeitsberichte der Hochschule für Wirtschaft FHNW, 26.11.07, 2007.
- Ferstl, Otto K.; Sinz, Elmar J.: Grundlagen der Wirtschaftsinformatik, München: Oldenbourg Verlag, 5. Auflage, 2006.
- Fortune, Annetta; Aldrich, Howard E. (2003): Acquiring Competence at a Distance: Application Service Providers as a Hybrid Organizational Form, Journal of International Entrepreneurship, Volume 1 (Number 1), S. 105-121, 2003.
- Fuchs, Christian: Begriffserklärung ERP II, ec-net.de, 2008
URL: http://www.ec-net.de/EC-Net/Redaktion/Pdf/Anwendungssoftware/begleitprojekt-erp-erklarung-erp-ii.property=pdf.bereich=ec_net.sprache=de.rwb=true.pdf
Letzter Zugriff : 09.03.08.
- Furht, Borko; Phoenix, Chris; Yin, John; Aganovic, Zijad (2000): An Innovative Internet Architecture for Application Service Providers, Proceedings of the 33th Hawaii International Conference on System Science, January, S. 10ff, 2000.
- Gadatsch, Andreas: Grundkurs Geschäftsprozessmanagement, Wiesbaden: Vieweg Verlag, 4. Auflage, 2007.

Gartner (2008): Gartner-Group, Gartner-Group.

URL: <http://www.gartner.com/>

Letzter Zugriff : 09.03.08.

Gartner Press (2007): Gartner Says Worldwide Software as a Service Revenue in the Enterprise Application Software Markets to Grow 21 Percent in 2007, Gartner Group.

URL: <http://www.gartner.com/it/page.jsp?id=511899>

Letzter Zugriff: 09.03.08.

Geihs, Kurt: Client/Server-Systeme - Grundlagen und Architekturen, Bonn: International Thomson Publishing GMBH, 1995.

Gluchowski, Peter; Gabriel, Roland; Dittmar, Carsten: Management Support Systeme und Business Intelligence, Computergestützte Informationssysteme für Fach- und Führungskräfte, Berlin, Heidelberg: Springer Verlag, 2. Auflage, 2008.

Gronau, Norbert: Enterprise Resource Planning und Supply Chain Management, München, Wien: Oldenbourg-Verlag, 2004.

GS1 (2008): GS1 Grundlagen, GS1.

URL: http://www.gs1-germany.de/internet/content/ueber_gs1_germany/index_ger.html

Letzter Zugriff : 09.03.08.

Harmon, Paul; Wolf, Celia (2007): The State of Business Process Management, BPTrends.com, Februar 2008.

URL: http://www.bptrends.com/memberssurveys/deliver.cfm?reportid=1003&target=BP Trends%202008%20Survey%20Report%202-11-08%20Final,%20V2.pdf&return=surveys_landing.cfm

Letzter Zugriff : 09.03.08.

Henschen, Doug (2006): SaaS and SOA: Together Forever, intelligententerprise.com, 01.12.06.

URL: <http://www.intelligententerprise.com/showArticle.jhtml;jsessionid=XA4PSCGKLM05OQSNDLRSKH0CJUNN2JVN?articleID=194500397>

Letzter Zugriff : 09.03.08.

Hill, Sidney (2007): Vendors benefit from SOA just as much as users, Manufacturing Business Technology, S. 48, 01.12.07.

Hill, Sidney (2008): Oracle changes tactics, but not intent, Manufacturing Business Technology, 01.02.08.

URL: <http://www.mbtmag.com/article/CA6529697.html>

Letzter Zugriff : 09.03.08.

Hitt, Lorin M.; Wu, D.J.; Zhou, Xiaoge (2002): Investment in Enterprise Resource Planning: Business Impact and Productivity Measures, Journal of Management Information Systems, Volume 19 (Number 1), S. 71–98, 2002.

Hülsenbusch, Ralph (2008): VMworld Europe 2008: Mit VMware ins Bett, heise.de, 27.02.08.

URL: <http://www.heise.de/newsticker/meldung/104149>

Letzter Zugriff: 09.03.08.

Huemer, Christian (2001): Electronic Business XML. In: (Turowski & Fellner, 2001). Chap. 1, S. 13–28.

innoQ Deutschland GmbH; Deutsche Post AG (2007): Web Services Standards Overview. innoq.com, Februar 2007.

URL: <http://www.innoq.com/soa/ws-standards/poster/innoQ%20WS-Standards%20Poster%202007-02.pdf>

Letzter Zugriff : 09.03.08.

Jarzyna, Dirk (2007): BEA sagt dem Betriebssystem Goodbye, networkcomputing.de, 26.09.07.

URL: <http://www.networkcomputing.de/nwc/home/artikel/article/boa-sagt-dem-betriebs-system-goodbye/>

Letzter Zugriff : 09.03.08.

Joos, Thomas: Terminalserver mit Citrix Metaframe XP, Wiesbaden: Vieweg Verlag, 2004.

Jung, Dr. Jakob (2007): SAP stellt Business by Design vor, informationweek.de, 24.09.07.

URL: <http://www.informationweek.de/news/showArticle.jhtml?articleID=201807691>

Letzter Zugriff : 09.03.08.

Keller, Erik (2006): Enterprise Software: a new beginning, Manufacturing Business Technology, 01.07.06.

URL: <http://www.mbtmag.com/article/CA6350389.html>

Letzter Zugriff : 09.03.08.

Kofler, Michael: Linux: Installation, Konfiguration, Anwendung, München: Addison-Wesley Verlag, 7.Auflage, 2005.

Kortus-Schultes, Doris; Ferfer, Ute: Logistik und Marketing in der Supply Chain:

Wertsteigerung durch virtuelle Geschäftsmodelle, Wiesbaden: Gabler Verlag, 2005.

Kotadia, Munir (2006): Business intelligence lies beyond IT: Dresner, zdnet.com.au, 27.02.06.

URL: <http://www.zdnet.com.au/news/software/soa/Business-intelligence-lies-beyond-IT-Dresner/0,130061733,139240318,00.htm>

Letzter Zugriff : 09.03.08.

Kromrey, Helmut: Empirische Sozialforschung, Opladen, Berlin; Leske & Budrich Verlag, 10. Auflage, 2002.

Kuschke, Michael, Wölfel, Ludger: Web Services Kompakt, Heidelberg, Berlin: Spektrum Akademischer Verlag, 2002.

Kwahk, Kee-Young: ERP Acceptance: Organizational Change Perspective, Proceedings of the 39th Hawaii International Conference on System Science, January, p. 172b, 2006.

Liberty, Jesse: Programming CSharp, Sebastopol, California: O'Reilly Media, 4. Auflage, 2005.

Liebhart, Daniel: SOA goes real, München: Carl Hanser Verlag, 2007

Linthicum, David: B2B Application Integration, Amsterdam: Addison-Wesley Longman, 2001.

Lixenfeld, Christoph (2008): SaaS: Lösungen aus der Leitung, Computerwoche.de, 06.02.08.

URL: http://www.computerwoche.de/hp_cw_mittelstand/loesungen/1854899/

Letzter Zugriff : 09.03.08.

Loney, Kevin; Theriault, Marlene: Oracle 9i, DBA-Handbuch, München, Wien: Carl Hanser Verlag, 2002.

Masak, Dieter: Moderne Enterprise Architekturen, Berlin: Springer Verlag, 1. Auflage, 2005.

Mayer, Horst O.: Interview und schriftliche Befragung, München: Oldenbourg Verlag, 2. Auflage, 2004.

- Mertens, Peter: Lexikon der Wirtschaftsinformatik, Berlin, Heidelberg: Springer, 4. Auflage, 2001.
- Mertens, Peter (Hrsg.): XML-Komponenten in der Praxis, Heidelberg: Springer Verlag, 2003.
- Mertens, Peter; Bodendorf, Freimut; König, Wolfgang; Picot, Arnold; Schumann, Matthias; Hess, Thomas: Grundzüge der Wirtschaftsinformatik, Berlin, Heidelberg: Springer Verlag, 9. Auflage, 2005.
- Miller, Gerry (2003): .Net vs. J2EE, Communications of the ACM, Volume 46 (Number 6), S. 64–67, 2003.
- Mühlfeld, Claus; Windolf, Paul; Lampert, Norbert; Krüger, Heidi (1981): Auswertungsprobleme offener Interviews. In: Soziale Welt. Jahrgang 32, S. 325–352.
- Mugler, Josef: Grundlagen der BWL der Klein- und Mittelbetriebe, Wien: Facultas Universitätsverlag, 1. Auflage, 2005.
- Niggel, Johann: Die Entstehung von Electronic Data Interchange Standards, Wiesbaden: Deutscher Universitäts-Verlag, 1994.
- Oasis (2008): SOA Spezifikation der OASIS-Group, Oasis,
URL: <http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf>
Letzter Zugriff : 09.03.08.
- OMG: (2008). Corba Spezifikation, OMG,
URL: http://www.omg.org/technology/documents/corba_spec_catalog.htm
Letzter Zugriff : 09.03.08.
- Overhage, Sven; Skroch, Oliver; Turowski, Klaus (2007): Vorlesung: Komponentenbasierte Anwendungssysteme, Lehrstuhl für Betriebswirtschaftslehre, Wirtschaftsinformatik und Systems Engineering, Universität Augsburg,
URL: http://www.wi2.info/index.php?option=com_docman&task=docdownload&gid=43&Itemid=130
Letzter Zugriff : 09.03.08.
- Parbel, Matthias (2008): VMware: Virtualisierung hilft auch kleinen und mittleren Unternehmen, heise.de, 26.02.08.
URL: <http://www.heise.de/newsticker/meldung/104071>
Letzter Zugriff : 09.03.08.
- Porter, Michael E.: Wettbewerbsvorteile: Spitzenleistungen erreichen und behaupten, Frankfurt/Main, New York: Campus, 6. Auflage, 2000.
- Priesnitz, Michael; Teille, Karl (2003): XML als Grundlage für standardisierte Internetschnittstellen. In: (Mertens, 2003). S. 9-27, 2003.
- Quantz, Joachim; Wichmann, Thorsten: E-Business Standards in Deutschland: Bestandsaufnahme, Probleme, Perspektiven, Berlin: Berlecon Research GmbH, 2003.
- Rüegg-Stürm, Johannes: Das neue St. Galler Management-Modell, Grundkategorien einer integrierten Management-Lehre, Bern, Stuttgart, Wien: Haupt, 2002.
- Rettig, Cynthia (2007): The Trouble with Enterprise Software, MIT Sloan Management Review, S. 21–27, 2007.
- Riordan, Rebecca M.: Microsoft SQL Server 2000 Programmierung, Redmond, Washington: Microsoft Press, 2000.

- Russinovich, Mark E.; Solomon, David A.: Microsoft Windows Internals, Redmond, Washington: Microsoft Press, 4.Auflage, 2005.
- Salmen-Fuchs, Tanja: Application Service Providing. Perspektiven, Anforderungen und Marktentwicklung, München: Grin Verlag, 2007.
- Schüler, Peter (2008): SAP bündelt Hard- und Software, heise.de, 04.03.08.
URL: <http://www.heise.de/newsticker/meldung/104463>
Letzter Zugriff : 09.03.08.
- Schmitt, Kathrin (2007):. ERP kommt in Deutschland langsam, aber gewaltig, silicon.de, Technologie und Business, 20.03.07.
URL: http://www.silicon.de/enid/wirtschaft_und_politik/26094
Letzter Zugriff : 09.03.08.
- Schubert, Petra (2002): E-Procurement: Elektronische Unterstützung der Beschaffungsprozesse in Unternehmen. In: (Wölflé et al., 2002). S. 1-28, 2002.
- Schubert, Petra; Wölflé, Ralf; Dettling, Walter: E-Business Integration: Fallstudien zur Optimierung elektronischer Geschäftsprozesse, München, Wien: Carl Hanser Verlag, 2005.
- Schumann, Rolf (2008): Mashups: Neue Self-Services für Business User? computerwoche.de, 27.02.08.
URL: <http://www.computerwoche.de/soa-expertenrat/2008/02/27/mashups-neue-self-services-fur-business-user/>
Letzter Zugriff : 09.03.08.
- Serain, Daniel: Middleware and Enterprise Application Integration, London: Springer Verlag, 2002.
- Serrano, Nicolás; Sarriegi, Josée Maria (2006): Open Source Software ERPs: A New Alternative for an Old Need, IEEE Software, Volume 3, S. 94–97, 2006.
- Siering, Peter (2008):. Neue Rundumsorglos-Server von Microsoft, heise.de, 20.02.08.
URL: <http://www.heise.de/newsticker/meldung/103831>
Letzter Zugriff : 09.03.08.
- Sourceforge (2008): Yuhana Applikationsserver, Sourceforge.
URL: <http://yuhana.sourceforge.net/>
Letzter Zugriff : 09.03.08.
- Staff (2007): Solutions buyers kick the tires of the software-as-a-service model, Manufacturing Business Technology, 01.06.08.
URL: <http://www.mbtmag.com/article/CA6450626.html>
Letzter Zugriff : 09.03.08.
- Stahlknecht, Peter; Hasenkamp, Ulrich: Einführung in die Wirtschaftsinformatik, Berlin: Springer Verlag, 11. Auflage, 2005.
- Stalder, Renato (2007): Standortübergreifende Prozeßintegration im Industrieunternehmen. In:(Wölflé & Schubert, 2007), S. 29–36.
- Steffen, Thomas (2001): XML/EDI-Standardisierung: Ein Überblick. In:(Turowski & Fellner, 2001), S. 1–12.
- Strnadl, Dr. Christop F. (2006): Einführung in Business Process Management (BPM) und BPM Systeme (BPMS), BPM-Netzwerk.de, 29.06.06.
URL: <http://www.bpm-guide.de/articles/55>
Letzter Zugriff : 09.03.08.

- Turowski, Klaus: Fachkomponenten: Komponentenbasierte betriebliche Anwendungssysteme, Aachen: Shaker Verlag, 2003.
- Turowski, Klaus (Hrsg.): Architekturen, Komponenten, Anwendungen: Proceedings zur 1. Verbundtagung Architekturen, Komponenten, Anwendungen (AKA 2004), Bonn: Gesellschaft für Informatik, 2004.
- Turowski, Klaus; Fellner, Klement J (Hrsg.): XML in der betrieblichen Praxis: Standards, Möglichkeiten, Praxisbeispiele, Heidelberg: dpunkt Verlag, 2001.
- Vilpola, Inka; Kouri, Ilkka; Väänänen-Vainio-Mattila, Kaisa (2007): Rescuing Small and Medium-sized Enterprises from Inefficient Information Systems - a Multi-disciplinary Method for ERP Systems Requirements Engineering, Proceedings of the 40th Hawaii International Conference on System Science, January, S. 242b., 2007.
- Westphal, Ralf (2003): Softwareentwicklung für den Applikationsserver des .NET Frameworks, Teil 1. Objektspektrum, April, 2003.
- Wilkens, Andreas (2008): SAP gewinnt mit Mittelstandsoftware neue Kunden, heise.de, 04.03.08.
URL: <http://www.heise.de/newsticker/meldung/104457>
Letzter Zugriff : 09.03.08.
- Wölfle, Ralf (2006): Business Prozessexzellenz mit Business Software. In: (Wölfle & Schubert, 2006), S. 5-18.
- Wölfle, Ralf (2007): Business Collaboration – Standortübergreifende Geschäftsprozesse. In: (Wölfle & Schubert, 2007), S. 1-16.
- Wölfle, Ralf; Schubert, Petra: Integrierte Geschäftsprozesse mit Business Software, München, Wien: Carl Hanser Verlag, 2005.
- Wölfle, Ralf; Schubert, Petra: Prozessexzellenz mit Business Software, München, Wien: Carl Hanser Verlag, 2006.
- Wölfle, Ralf; Schubert, Petra: Business Collaboration -Standortübergreifende Prozesse mit Business Software, München: Carl Hanser Verlag, 2007.
- Wölfle, Ralf; Schubert, Petra; Dettling, Walter: Procurement im E-Business, München, Wien: Carl Hanser Verlag, 2002.
- Zaha, Johannes Maria; Kelch, Stephan (2004): Komponentenfindung in monolithischen objektorientierten Anwendungssystemen. In: (Turowski, 2004). S. 151-162.
- Zunic, Markus (2008): Gartner: planloser Einsatz von Business Intelligence in Unternehmen, techchannel.de, 06.02.08.
URL: <http://www.tecchannel.de/news/themen/business/1746063/>
Letzter Zugriff : 09.03.08.

Anhang

In diesem Kapitel befinden sich zusätzliche Informationen zu den befragten Unternehmen und das Abkürzungsverzeichnis für die in der Arbeit verwendeten Kurznamen der Anbieter.

Protokolle der Interviews

Da die bei den Interviews angefallenen Daten als ein Kapitel im Anhang den Rahmen dieser Arbeit deutlich sprengen, wurden die Protokolle auf einem Datenträger (CD) gesichert und liegen der schriftlichen Version der Arbeit bei. Die elektronische Version dieser Arbeit wird dagegen die Protokolle im Anschluss enthalten.

Kurzversionen der Unternehmensnamen

Im Folgenden findet sich eine Liste, die benutzte Kurznamen der Unternehmen dem vollständigen Namen zuordnet (vgl. Abbildung A):

| | |
|--|--------------------------------|
| Abacus Research AG | Abacus |
| ALEA GmbH | Alea |
| ams.hinrichs+müller GmbH | ams |
| B.I.M.-Consulting mbH | B.I.M. |
| IAS GmbH | IAS |
| nGroup (Compra) | nGroup |
| e.bootis AG | eBootis |
| ERP-21 GmbH | ERP-21 |
| Godyo AG | Godyo |
| Bison Schweiz AG | Bison |
| Günther Datenverarbeitung GmbH | Günther-BS |
| GUS Group AG | GuS |
| Hilmer Software GmbH | Hilmer |
| IFS Deutschland GmbH | IFS |
| Informing AG | Informing |
| Seat-1 Software GmbH | Seat-1 |
| INTRAPREND Gesellschaft für Intranet Anwendungsentwicklung mbH | Intraprend |
| Jentech Datensysteme AG | Jentech |
| Lawson Software Deutschland GmbH | Lawson |
| ERP4all Business Software GmbH | ERP4all |
| Microsoft Deutschland myfactory Software GmbH | Microsoft (Navision) myfactory |
| PSI AG | PSI |
| Ramco Systems Ltd. | Ramco |
| Sage Baurer | Baurer |
| SAP AG | SAP |
| SoftM Software und Beratung AG | SoftM |
| SHD Holding GmbH | SHD |
| SynERP Y GmbH | SynERP Y |
| Topix Informationssysteme AG | Topix |
| Wika Systems | Microsoft (Axapta) |
| Oracle | Oracle |

Abbildung A Kurzversionen der Unternehmensnamen

Bisher erschienen

Arbeitsberichte aus dem Fachbereich Informatik

(<http://www.uni-koblenz.de/fb4/publikationen/arbeitsberichte>)

Norbert Frick, Künftige Anforderungen an ERP-Systeme: Deutsche Anbieter im Fokus, Arbeitsberichte aus dem Fachbereich Informatik 11/2008

Jürgen Ebert, Rüdiger Grimm, Alexander Hug, Lehramtsbezogene Bachelor- und Masterstudiengänge im Fach Informatik an der Universität Koblenz-Landau, Campus Koblenz, Arbeitsberichte aus dem Fachbereich Informatik 10/2008

Mario Schaarschmidt, Harald von Kortzfleisch, Social Networking Platforms as Creativity Fostering Systems: Research Model and Exploratory Study, Arbeitsberichte aus dem Fachbereich Informatik 9/2008

Bernhard Schueler, Sergej Sizov, Steffen Staab, Querying for Meta Knowledge, Arbeitsberichte aus dem Fachbereich Informatik 8/2008

Stefan Stein, Entwicklung einer Architektur für komplexe kontextbezogene Dienste im mobilen Umfeld, Arbeitsberichte aus dem Fachbereich Informatik 7/2008

Matthias Bohnen, Lina Brühl, Sebastian Bzdak, RoboCup 2008 Mixed Reality League Team Description, Arbeitsberichte aus dem Fachbereich Informatik 6/2008

Bernhard Beckert, Reiner Hähnle, Tests and Proofs: Papers Presented at the Second International Conference, TAP 2008, Prato, Italy, April 2008, Arbeitsberichte aus dem Fachbereich Informatik 5/2008

Klaas Dellschaft, Steffen Staab, Unterstützung und Dokumentation kollaborativer Entwurfs- und Entscheidungsprozesse, Arbeitsberichte aus dem Fachbereich Informatik 4/2008

Rüdiger Grimm: IT-Sicherheitsmodelle, Arbeitsberichte aus dem Fachbereich Informatik 3/2008

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik 2/2008

Markus Maron, Kevin Read, Michael Schulze: CAMPUS NEWS – Artificial Intelligence Methods Combined for an Intelligent Information Network, Arbeitsberichte aus dem Fachbereich Informatik 1/2008

Lutz Priese, Frank Schmitt, Patrick Sturm, Haojun Wang: BMBF-Verbundprojekt 3D-RETISEG Abschlussbericht des Labors Bilderkennen der Universität Koblenz-Landau, Arbeitsberichte aus dem Fachbereich Informatik 26/2007

Stephan Philippi, Alexander Pinl: Proceedings 14. Workshop 20.-21. September 2007 Algorithmen und Werkzeuge für Petrinetze, Arbeitsberichte aus dem Fachbereich Informatik 25/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS – an Intelligent Bluetooth-based Mobile Information Network, Arbeitsberichte aus dem Fachbereich Informatik 24/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS - an Information Network for Pervasive Universities, Arbeitsberichte aus dem Fachbereich Informatik 23/2007

Lutz Priebe: Finite Automata on Unranked and Unordered DAGs Extended Version, Arbeitsberichte aus dem Fachbereich Informatik 22/2007

Mario Schaarschmidt, Harald F.O. von Kortzfleisch: Modularität als alternative Technologie- und Innovationsstrategie, Arbeitsberichte aus dem Fachbereich Informatik 21/2007

Kurt Lautenbach, Alexander Pinl: Probability Propagation Nets, Arbeitsberichte aus dem Fachbereich Informatik 20/2007

Rüdiger Grimm, Farid Mehr, Anastasia Meletiadou, Daniel Pähler, Ilka Uerz: SOA-Security, Arbeitsberichte aus dem Fachbereich Informatik 19/2007

Christoph Wernhard: Tableaux Between Proving, Projection and Compilation, Arbeitsberichte aus dem Fachbereich Informatik 18/2007

Ulrich Furbach, Claudia Obermaier: Knowledge Compilation for Description Logics, Arbeitsberichte aus dem Fachbereich Informatik 17/2007

Fernando Silva Parreiras, Steffen Staab, Andreas Winter: TwoUse: Integrating UML Models and OWL Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 16/2007

Rüdiger Grimm, Anastasia Meletiadou: Rollenbasierte Zugriffskontrolle (RBAC) im Gesundheitswesen, Arbeitsberichte aus dem Fachbereich Informatik 15/2007

Ulrich Furbach, Jan Murray, Falk Schmidsberger, Frieder Stolzenburg: Hybrid Multiagent Systems with Timed Synchronization-Specification and Model Checking, Arbeitsberichte aus dem Fachbereich Informatik 14/2007

Björn Pelzer, Christoph Wernhard: System Description: "E-KRHyper", Arbeitsberichte aus dem Fachbereich Informatik, 13/2007

Ulrich Furbach, Peter Baumgartner, Björn Pelzer: Hyper Tableaux with Equality, Arbeitsberichte aus dem Fachbereich Informatik, 12/2007

Ulrich Furbach, Markus Maron, Kevin Read: Location based Information systems, Arbeitsberichte aus dem Fachbereich Informatik, 11/2007

Philipp Schaer, Marco Thum: State-of-the-Art: Interaktion in erweiterten Realitäten, Arbeitsberichte aus dem Fachbereich Informatik, 10/2007

Ulrich Furbach, Claudia Obermaier: Applications of Automated Reasoning, Arbeitsberichte aus dem Fachbereich Informatik, 9/2007

Jürgen Ebert, Kerstin Falkowski: A First Proposal for an Overall Structure of an Enhanced Reality Framework, Arbeitsberichte aus dem Fachbereich Informatik, 8/2007

Lutz Priebe, Frank Schmitt, Paul Lemke: Automatische See-Through Kalibrierung, Arbeitsberichte aus dem Fachbereich Informatik, 7/2007

Rüdiger Grimm, Robert Krimmer, Nils Meißner, Kai Reinhard, Melanie Volkamer, Marcel Weinand, Jörg Helbach: Security Requirements for Non-political Internet Voting, Arbeitsberichte aus dem Fachbereich Informatik, 6/2007

Daniel Bildhauer, Volker Riediger, Hannes Schwarz, Sascha Strauß, „grUML – Eine UML-basierte Modellierungssprache für T-Graphen“, Arbeitsberichte aus dem Fachbereich Informatik, 5/2007

Richard Arndt, Steffen Staab, Raphaël Troncy, Lynda Hardman: Adding Formal Semantics to MPEG-7: Designing a Well Founded Multimedia Ontology for the Web, Arbeitsberichte aus dem Fachbereich Informatik, 4/2007

Simon Schenk, Steffen Staab: Networked RDF Graphs, Arbeitsberichte aus dem Fachbereich Informatik, 3/2007

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik, 2/2007

Anastasia Meletiadou, J. Felix Hampe: Begriffsbestimmung und erwartete Trends im IT-Risk-Management, Arbeitsberichte aus dem Fachbereich Informatik, 1/2007

„Gelbe Reihe“

(<http://www.uni-koblenz.de/fb4/publikationen/gelbereihe>)

Lutz Priebe: Some Examples of Semi-rational and Non-semi-rational DAG Languages. Extended Version, Fachberichte Informatik 3-2006

Kurt Lautenbach, Stephan Philippi, and Alexander Pinl: Bayesian Networks and Petri Nets, Fachberichte Informatik 2-2006

Rainer Gimnich and Andreas Winter: Workshop Software-Reengineering und Services, Fachberichte Informatik 1-2006

Kurt Lautenbach and Alexander Pinl: Probability Propagation in Petri Nets, Fachberichte Informatik 16-2005

Rainer Gimnich, Uwe Kaiser, and Andreas Winter: 2. Workshop "Reengineering Prozesse" – Software Migration, Fachberichte Informatik 15-2005

Jan Murray, Frieder Stolzenburg, and Toshiaki Arai: Hybrid State Machines with Timed Synchronization for Multi-Robot System Specification, Fachberichte Informatik 14-2005

Reinhold Letz: FTP 2005 – Fifth International Workshop on First-Order Theorem Proving, Fachberichte Informatik 13-2005

Bernhard Beckert: TABLEAUX 2005 – Position Papers and Tutorial Descriptions, Fachberichte Informatik 12-2005

Dietrich Paulus and Detlev Droege: Mixed-reality as a challenge to image understanding and artificial intelligence, Fachberichte Informatik 11-2005

Jürgen Sauer: 19. Workshop Planen, Scheduling und Konfigurieren / Entwerfen, Fachberichte Informatik 10-2005

Pascal Hitzler, Carsten Lutz, and Gerd Stumme: Foundational Aspects of Ontologies, Fachberichte Informatik 9-2005

Joachim Baumeister and Dietmar Seipel: Knowledge Engineering and Software Engineering, Fachberichte Informatik 8-2005

Benno Stein and Sven Meier zu Eißten: Proceedings of the Second International Workshop on Text-Based Information Retrieval, Fachberichte Informatik 7-2005

Andreas Winter and Jürgen Ebert: Metamodel-driven Service Interoperability, Fachberichte Informatik 6-2005

Joschka Boedecker, Norbert Michael Mayer, Masaki Ogino, Rodrigo da Silva Guerra, Masaaki Kikuchi, and Minoru Asada: Getting closer: How Simulation and Humanoid League can benefit from each other, Fachberichte Informatik 5-2005

Torsten Gipp and Jürgen Ebert: Web Engineering does profit from a Functional Approach, Fachberichte Informatik 4-2005

Oliver Obst, Anita Maas, and Joschka Boedecker: HTN Planning for Flexible Coordination Of Multiagent Team Behavior, Fachberichte Informatik 3-2005

Andreas von Hessling, Thomas Kleemann, and Alex Sinner: Semantic User Profiles and their Applications in a Mobile Environment, Fachberichte Informatik 2-2005

Heni Ben Amor and Achim Rettinger: Intelligent Exploration for Genetic Algorithms – Using Self-Organizing Maps in Evolutionary Computation, Fachberichte Informatik 1-2005