



**TECHNISCHE  
UNIVERSITÄT  
DRESDEN**



**Dresden  
Informatik**  
GmbH

**TECHNISCHE UNIVERSITÄT DRESDEN**  
Fakultät Informatik  
Institut für Systemarchitektur  
Professur Rechnernetze

**DRESDEN INFORMATIK GMBH**  
Tannenstraße 2  
01099 Dresden

## **DIPLOMARBEIT**

Zum Thema

### **Entwurf und prototypische Realisierung von einer Migrationsstrategie für ERP-Systeme**

<b>Eingereicht von:</b>	Hicham Nasri Hicham.Nasri@mailbox.tu-dresden.de Matrikel-Nr.: 3287660
<b>Verantwortlicher Hochschullehrer:</b>	Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill
<b>Betreuer TU Dresden:</b>	Dr. rer. nat. Dietbert Gütter
<b>Betreuer Dresden Informatik GmbH:</b>	Dipl.-Ing. Janet Reinert
<b>Begonnen am:</b>	15.02.2011
<b>Eingereicht am:</b>	14.08.2011

## **Selbstständigkeitserklärung**

Ich erkläre hiermit, dass ich die vorliegende Diplomarbeit selbstständig, unter Angabe aller Zitate und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Die Arbeit wurde bisher weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

---

Hicham Nasri

Dresden, den 14. August 2011

## **Danksagung**

Ich bedanke mich bei der Firma Dresden Informatik GmbH, die mir die Möglichkeit gegeben hat, dieses Thema zu bearbeiten.

Außerdem danke ich meiner Betriebsbetreuerin Frau Reinert für die fachliche Betreuung und die zahlreichen Ratschläge und Hinweise.

Nicht zuletzt möchte ich meiner Frau für ihre Unterstützung und ihre Korrekturvorschläge danken.

Weiterhin bedanke ich mich bei allen Personen, die mir durch ihre Unterstützung geholfen haben, diese Arbeit zu erstellen.

# Inhaltverzeichnis

<b>1</b>	<b><i>Einleitung</i></b> .....	<b>1</b>
1.1	Motivation .....	1
1.2	Zielstellung .....	2
1.3	Aufbau der Arbeit .....	2
<b>2</b>	<b><i>Grundlagen des Enterprise Resource Planning</i></b> .....	<b>4</b>
2.1	Definition des ERP-Systems.....	4
2.2	Ziele der ERP-Systeme .....	6
2.3	Aufbau und Funktionsweise von ERP-Systemen.....	7
2.4	Systemarchitektur von ERP-Systemen.....	9
2.5	ERP Open Source.....	11
<b>3</b>	<b><i>Das ERP-System DiPPS</i></b> .....	<b>14</b>
3.1	Analyse der Systemarchitektur von DiPPS .....	14
3.2	Module und Funktionalität von DiPPS .....	18
3.3	Zusammenfassung und Fazit .....	19
<b>4</b>	<b><i>Migration von ERP-Systemen</i></b> .....	<b>20</b>
4.1	Begriff und Möglichkeiten der Migration .....	20
4.2	Ablauf der Datenmigration.....	22
4.2.1	Datenmigration .....	22
4.2.2	Planung der Datenmigration.....	24
4.2.3	Zusammenfassung .....	26
<b>5</b>	<b><i>Anforderungen an ERP-Systeme</i></b> .....	<b>27</b>
5.1	Funktionale Anforderungen.....	27
5.1.1	Mehrschichtenmodell.....	27
5.1.2	Modularer Aufbau .....	29
5.1.3	Mandantenfähigkeit .....	29
5.1.4	Datenbankunabhängigkeit.....	30
5.1.5	Internetfähigkeit .....	30
5.1.6	Releasefähigkeit .....	31
5.2	Nicht funktionale Anforderungen.....	31

5.3	Zusammenfassung.....	33
<b>6</b>	<b>Analyse bestehender Ansätze zur Auswahl eines OS-ERP-Systems.....</b>	<b>34</b>
6.1	Open-Source-Software für das Enterprise Resource Planning.....	34
6.2	ERP auf Basis Freier Software- Zukunftsvision und Realität .....	36
6.3	Zusammenfassung und Fazit .....	38
<b>7</b>	<b>Analyse und Konzeption .....</b>	<b>39</b>
7.1	Konzept für das Auswahlverfahren eines OS-ERP-Systems.....	39
7.1.1	Problemstellung und Zieldefinition .....	42
7.1.2	Zustandsanalyse .....	42
7.1.3	Pflichtenheft.....	43
7.1.4	Durchführung der Marktanalyse und Vorauswahl.....	48
7.1.5	Evaluierung und vergleichende Gegenüberstellung .....	51
7.1.6	Vergleich mit Pflichtenheft und endgültige Auswahl .....	54
7.1.7	Adempiere .....	56
7.1.8	Zusammenfassung.....	57
7.2	Konzept einer kundenspezifische Anpassung in ADempiere .....	58
7.2.1	Allgemeine Anpassungsarten.....	58
7.2.2	Anpassungen in ADempiere.....	59
7.3	Prototypische Implementierung .....	67
7.4	Auswertung.....	73
<b>8</b>	<b>Zusammenfassung und Ausblick.....</b>	<b>75</b>
<b>9</b>	<b>Anhang.....</b>	<b>78</b>
	<i>A01 Inhalt der DVD.....</i>	<i>78</i>
	<i>A02 verschiedene Lizenztypen.....</i>	<i>79</i>
	<i>A03 Umfrage über die Benutzung der freien ERP-Lösungen bei kleinen und mittleren Unternehmen [ECKR].....</i>	<i>80</i>
	<i>A04 DiPPS Module .....</i>	<i>82</i>
	<i>A05 Ansatz für OS-ERP-Auswahl.....</i>	<i>83</i>
	<i>A06 Vergleich ADempiere und Openbravo mit Pflichtenheft.....</i>	<i>85</i>
	<i>A07 AD Validation Fenster.....</i>	<i>90</i>
	<i>A08 AD Rule Engine Fenster.....</i>	<i>91</i>
	<i>A09 Verweis der Datenfelder- und Anzeigetypen auf Datenquelle .....</i>	<i>92</i>

<i>A10 Erstellung der Fenster „Workshop“ und „Workshopzustand“ in AD .....</i>	<i>93</i>
<i>A11 Verbindung der zwei Fenster „Workshop“ und „Workshopzustand“ .....</i>	<i>97</i>
<i>A12 Die Logik für erstellte Fenster .....</i>	<i>99</i>
<i>A13 Anlegen von Callout in Tab „W Thema“ für das Datenfeld „item“ .....</i>	<i>100</i>
<i>A14 Generierung der Model Validator für Business-Logik.....</i>	<i>101</i>
<i>A15 Speicherung der Anwendung durch Packout.....</i>	<i>102</i>
<i>A16 Anpassung durch QuellCode.....</i>	<i>103</i>
<i>Literaturverzeichnis .....</i>	<i>104</i>
<i>Abbildungsverzeichnis.....</i>	<i>108</i>
<i>Tabellenverzeichnis .....</i>	<i>110</i>
<i>Abkürzungsverzeichnis.....</i>	<i>111</i>
<i>Glossar.....</i>	<i>112</i>



# 1 Einleitung

Das Kapitel verdeutlicht die Motivation der Arbeit, führt in ERP-Systeme ein, und beschreibt die Problemstellung. Im Anschluss wird die Zielstellung definiert und abschließend der Aufbau der Arbeit beschrieben.

## 1.1 Motivation

Heutzutage spielt der elektronische Geschäftsverkehr im Unternehmen eine wichtige Rolle. Dies wird mittels besonderer Software realisiert. Sie heißen Enterprise Resource Planning Systeme (ERP-Systeme). Sie beschreiben eine Abbildung und Automatisierung von Geschäftsprozessen und Geschäftsregeln und unterstützen Unternehmen durch eine geeignete IT-Infrastruktur bei einer möglichst effizienten Allokation der zur Verfügung stehenden Ressourcen (z.B. Personal, Geschäftspartner, Produktion oder Kapital) in den verschiedensten betrieblichen Abläufen. Sie sammeln Informationen über den Zustand und die Aktivitäten der verschiedenen Abteilungen eines Unternehmens und übermitteln sie an andere Stellen (an anderen Abteilungen), die z.B. durch Wiederverwendung davon profitieren können. Die Informationen werden von den Benutzern in Echtzeit aktualisiert und sind jederzeit allen zugänglich, die an das System angeschlossen sind.

Zurzeit gibt es zahlreiche ERP-Systeme auf dem Markt (Proprietäre, Open Source und Individualsoftware). Deswegen ist die Auswahl und Einführung eines passenden Systems schwierig. Dabei müssen verschiedene Aspekte wie Lizenzen, Kosten, Funktionalitäten berücksichtigt werden.

Neben der Auswahl eines ERP-Systems ist das Thema Migration zu diskutieren.

In allen Großunternehmen und vielen klein- und mittelständischen Unternehmen wurden seit den 90er Jahren ERP-Systeme oder Warenwirtschaftssysteme eingeführt. Nach einigen Jahren der Benutzung eines Systems soll das ERP-System die Expansion der Geschäftsprozesse und Modernisierung der IT-Infrastruktur des Unternehmens folgen.

Die Dresden Informatik GmbH bietet seit zehn Jahren das ERP-System DiPPS an, das in dieser Arbeit migriert wird.

## **1.2 Zielstellung**

Ziel dieser Arbeit ist es, einen effizienten und kostengünstigen Weg zur Erneuerung des DiPPS-Systems der Firma Dresden Informatik GmbH aufzuzeigen. Dabei soll DiPPS nach Systemarchitektur und Funktionalitäten analysiert werden. Dann sollen die Migrationsmöglichkeiten dieses Systems untersucht werden, die im Rahmen der System-Migration diskutiert werden. Darauf folgt die Evaluierung und Auswahl eines geeigneten ERP-Systems, das mit DiPPS verglichen werden soll.

Danach wird ein Konzept für kundenspezifische Anpassungen des ausgewählten Systems in dieser Arbeit erstellt und umgesetzt.

Zuletzt soll eine Systemdokumentation hinsichtlich der Installation, Benutzung und Anpassung des ausgewählten Systems realisiert werden.

## **1.3 Aufbau der Arbeit**

Die Arbeit gliedert sich in zwei Hauptteile. Der erste Teil behandelt die theoretischen Grundlagen der ERP-Systeme, ihre Aufbaustruktur, ihre Anforderungen und ihre Migration. Der zweite Teil thematisiert die konzeptuellen Entwürfe und deren Implementierungen. Es wird hier ein Konzept sowohl für die Auswahl eines ERP-Systems durch ein Auswahlverfahren als auch für die Anpassung des ausgewählten ERP-Systems und seine Implementierung erstellt.

Im ersten Teil werden die wissenschaftlichen Grundlagen für den weiteren Verlauf erarbeitet. Es werden Begriffe definiert, erläutert und sowohl funktionale als auch nicht funktionale Anforderungen an ERP-Systeme festgelegt, die im zweiten Teil für das Konzept zur Auswahl eines geeigneten Systems verwendet werden. Die Auswahl wird in einem Verfahren durchgeführt, durch das sich eine Vorauswahl einiger potentieller Systeme ergeben wird. Diese werden mit einem erstellten Pflichtenheft nach detaillierten Funktionalitäten verglichen. Es erfolgt die

Entscheidung für ein System (wenn es den definierten Anforderungen genügt), für das im letzten Teil dieser Arbeit ein Konzept für die kundenspezifischen Anpassungen erstellt und dann implementiert wird. Zum Schluss werden die Ergebnisse der Anpassung zusammengefasst und bewertet.

## 2 Grundlagen des Enterprise Resource Planning

Dieses Kapitel gibt einen Überblick über die ERP-Systeme, ihre Ziele, Funktionalitäten sowie ihren Aufbau.

### 2.1 Definition des ERP-Systems

ERP-Systeme sind Anwendungssoftwares, die die Abläufe eines Unternehmens unterstützen und dadurch den effektiven Einsatz der vorhandenen Ressourcen ermöglichen. [ERPPro]

Im Wirtschaftslexikon wird ein ERP-System wie folgt definiert:

„Ein Enterprise Resource Planning-System oder kurz ERP-System dient der funktionsbereichsübergreifenden Unterstützung sämtlicher in einem Unternehmen ablaufenden Geschäftsprozesse. Entsprechend enthält es Module für die Bereiche Beschaffung, Produktion, Vertrieb, Anlagenwirtschaft, Personalwesen, Finanz- und Rechnungswesen usw., die über eine (in Form einer relationalen Datenbank realisierte) gemeinsame Datenbasis miteinander verbunden sind. Durch die unternehmensweite Konsolidierung der Daten ist eine Unterstützung der Planung über sämtliche Unternehmensebenen hinweg (von der Konzernebene über verschiedene Werke, Sparten und Abteilungen bis hin zu einzelnen Lagerorten) möglich.“

[lexi]

Die ERP-Systeme sind aus Programmen für die Produktionsplanung und -steuerung (PPS) mit Anbindung an das Rechnungs- und Finanzwesen entwickelt worden. Später kamen viele weitere Funktionen hinzu. Beispiele sind Zusätze für das Personalwesen, die Logistik, das E-Business und das Customer Relationship Management (CRM).

ERP-Systeme basieren auf einer einheitlichen Datenbank. Sie stellen Informationen für alle Unternehmensbereiche zur Bearbeitung oder Analyse zu Verfügung. Sie verknüpfen insbesondere Informationen über Finanzen, personelle Ressourcen, Produktion und Vertrieb. Sie umfassen Lagerverwaltungssysteme, Kundendatenbanken, Auftragsverfolgungssysteme, Kreditorenbuchhaltung und

vieles andere mehr. Sie bieten je nach Bedarf auch Schnittstellen für Zulieferer und Kunden.



**Abbildung 2.1: Aufgabenkategorien und Einsatzbereiche von ERP-Systemen [GrAnd]**

Die Abbildung 2.1 stellt zusammenfassend die Aufgabenkategorien und Einsatzbereiche von ERP-Systemen dar. Sie unterstützen unterschiedlich diese Bereiche: Einige Systeme fördern nur einen Teil dieser Bereiche und einige unterstützen mehr Bereiche. [AnbERP]

## **2.2 Ziele der ERP-Systeme**

Im vorigen Abschnitt wurde eine Definition für ERP-System gegeben. Diese enthält bereits die Ziele solcher Systeme: Sie haben die Aufgabe, Prozesse innerhalb eines Unternehmens zu organisieren und zu systematisieren und die Daten zu Analyse Zwecken aufzubereiten.

Die ERP-Systeme dienen der Darstellung, Optimierung und Unterstützung der innerbetrieblichen Geschäftsprozesse eines Unternehmens. Dabei können sie Funktionalitäten der PPS- Systeme, wie Produktion- und Arbeitsplanung, integrieren.

Außerdem sollen durch ERP-Systeme kürzere Durchlaufzeiten, höhere Termintreue und höhere Flexibilität bei Änderungen erreicht werden.

Die allgemeinen Ziele, die mit der Einführung eines ERP-Systems realisiert werden sollen sind:

- „Optimierung der betriebswirtschaftlichen Ressourcenplanung unter verschiedenen Randbedingungen (Globalisierung der Wirtschaft und rasche Veränderungen der Märkte).
- Integration der wichtigen Geschäftsprozesse in ein Gesamtsystem
- Bearbeitung der gesamten Bandbreite betriebswirtschaftlicher Fragestellungen in einem Unternehmen“ [W&W]

Die folgende Abbildung 2.2 stellt die verschiedenen Faktoren dar, die den Nutzen einer PPS-Einführung beeinflussen. Diese sind von PPS-Investitionen zu erwarten.

Ein betriebswirtschaftlicher Erfolg wird durch die Kontrolle und Steuerung dieser Faktoren erreicht:

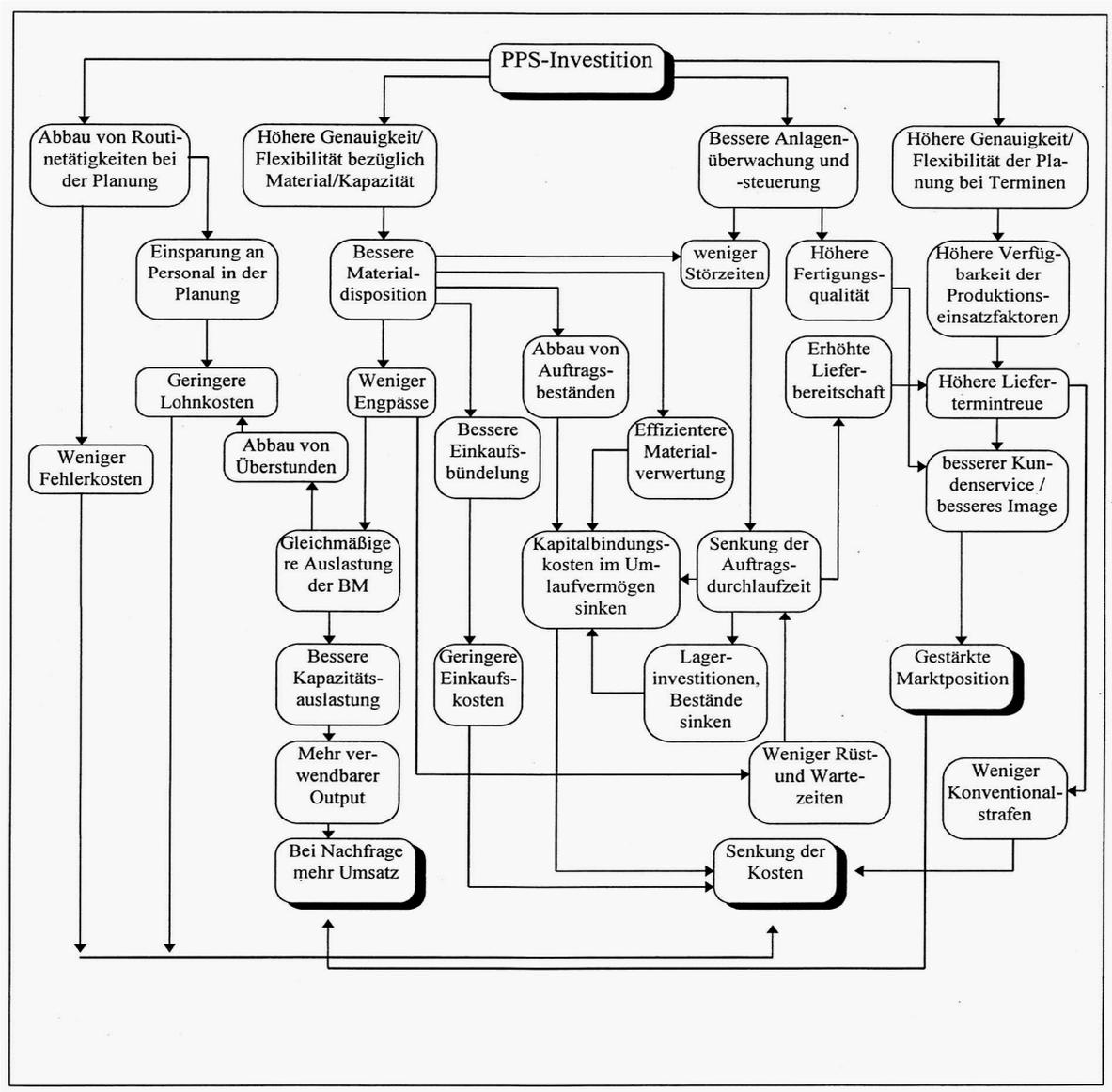


Abbildung 2.2: Nutzeffekte von CIM-Komponenten und Integrationskonzepten (Teil 1) [nutzEf]

### 2.3 Aufbau und Funktionsweise von ERP-Systemen

Die ERP-Systeme bestehen aus einer Vielzahl von Modulen, die in ihrer Gesamtheit die wesentlichen Geschäftsprozesse in einem Unternehmen unterstützen. Die nächste Abbildung stellt die Struktur und typische Komponenten eines ERP-Systems dar:

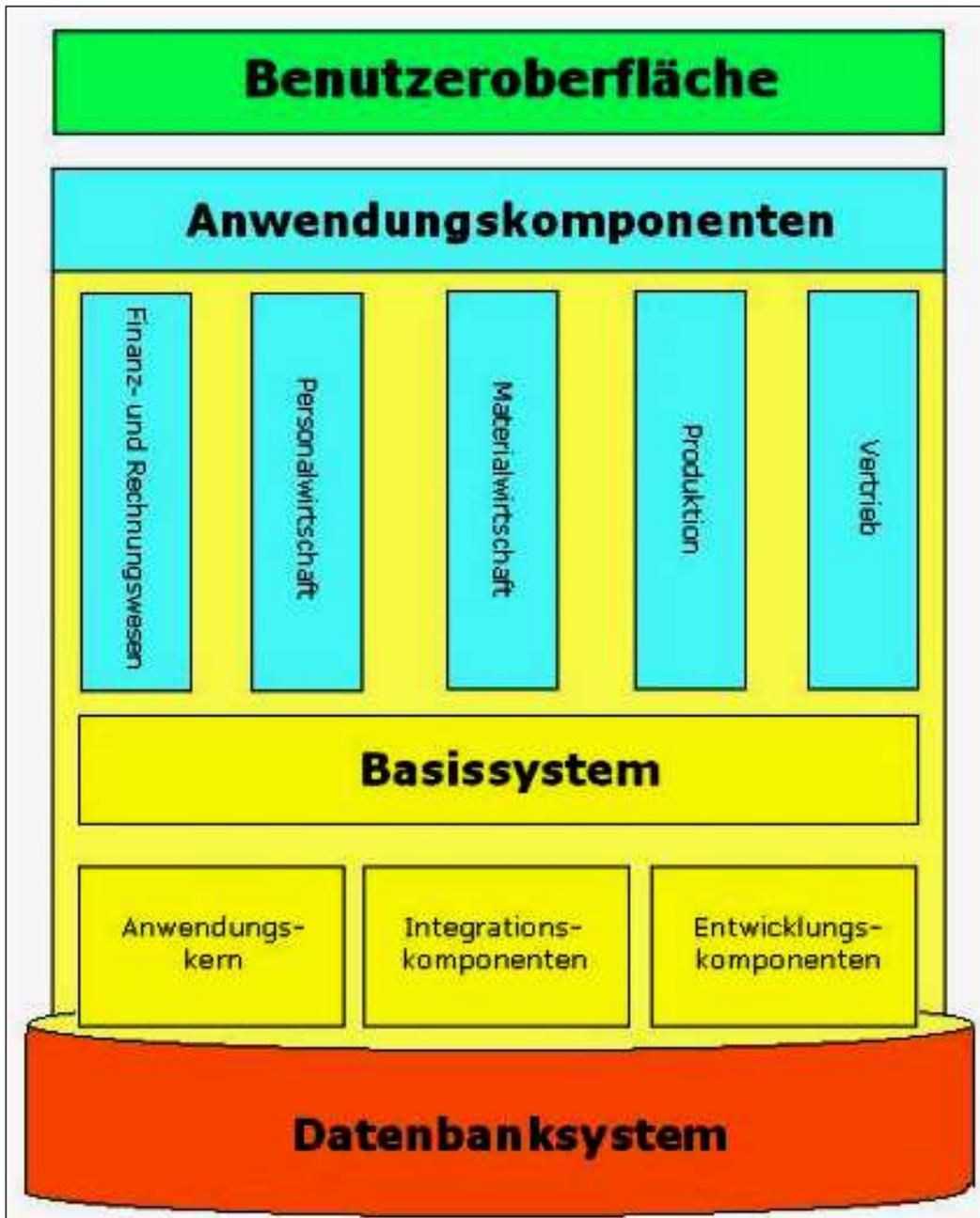


Abbildung 2.3: Aufbau und Anwendungsbereiche eines ERP-Systems

Neben der Benutzeroberfläche und dem Datenbanksystem basiert ein solches System auf einem Basissystem, in das verschiedene Anwendungskomponenten implementiert werden können.

Die Benutzeroberfläche erlaubt die Interaktion zwischen Mensch und Maschine (meist PC oder mobiles Endgerät). Sie kann bei ERP-Systemen an den Benutzer angepasst werden und somit auf dessen Bedürfnisse und Ansprüche ausgerichtet sein.

Die ERP-Systeme enthalten Anwendungskomponenten, auch Module genannt, die wieder in Unterkomponenten geteilt werden können. Jedes dieser Module bzw. jede Unterkomponente beinhaltet bestimmte betriebswirtschaftliche Funktionalitäten des Anwendungsbereichs, die logisch zusammengehören [MAN].

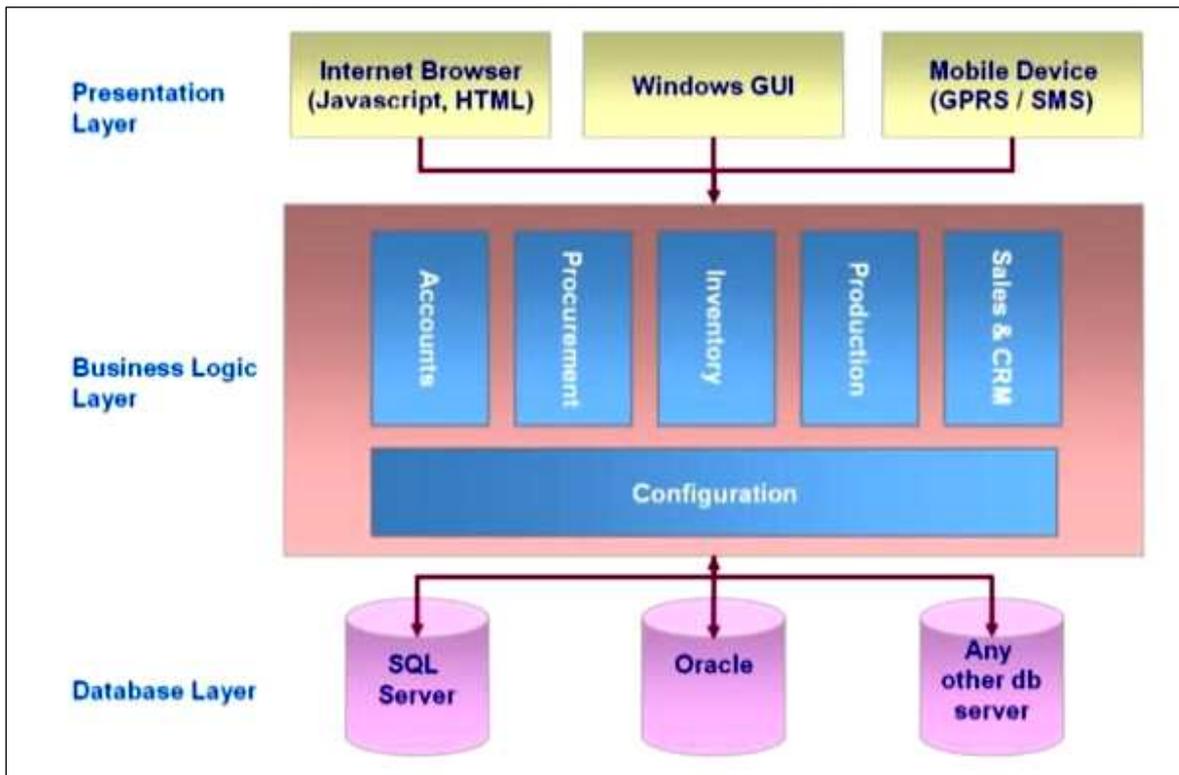
Das Basissystem bildet die Infrastruktur für die Module und realisiert die Schnittstellen zu der Datenbank und der Benutzeroberfläche. Es besitzt Funktionen zur Steuerung und Administration des Systems (wie Benutzerverwaltung, Systemeinstellungen), Schnittstellen zum Betriebssystem, verschiedene Funktionen für die Anpassung, die Entwicklungsumgebung, Programmierschnittstellen sowie den Zugriff auf die Datenbanken [HNWirt].

ERP-Systeme verfügen über ein Datenbanksystem als ein wichtiger Bestandteil. Damit wird ermöglicht, Daten zu speichern, zu verändern, zu löschen, sowie Abfragen zu stellen. Die Datenbank eines ERP-Systems dient vor allem als Speicherungsgrundlage für seine Komponenten und soll dabei Redundanz und Dateninkonsistenz vermeiden.

## **2.4 Systemarchitektur von ERP-Systemen**

Die ERP-Systeme bestehen, wie im vorigen Abschnitt erläutert, aus verschiedenen Komponenten. Sie zeichnen sich immer durch eine 3-Ebenen-Architektur aus und sind heutzutage auf einer Client-Server-Architektur lauffähig. Der Server verarbeitet die von den Clients erstellten Abfragen und Transaktionen. An der Bearbeitung einer Geschäftstransaktion sind mindestens zwei Computerprozesse (Client und Server) beteiligt, die unterschiedliche Aufgaben wahrnehmen [vgl. Abb.1.11, in: [Gronau]]. Der Ablauf der Kommunikation zwischen Client und Server basiert folglich auf Transaktionen.

Die 3-Ebenen-Architektur besteht, wie in der Abbildung 2.4 [ERPSolu] verdeutlicht wird, aus der Präsentationsebene, der Anwendungsebene und der Datenhaltungsebene:



**Abbildung 2.4: 3-Ebenen-Architektur eines ERP-Systems**

Alle drei Ebenen haben unterschiedliche Aufgabenbereiche, die ein ERP-System in seiner Ganzheit benötigt.

Die Präsentationsebene (Presentation Layer) stellt eine Schnittstelle zwischen einem Client und einer Anwendung dar. Sie ist somit die Verbindung zwischen dem ERP-System und dem Benutzer. Sie beschreibt den statischen Aspekt der Benutzeroberfläche, steuert die Interaktion in Verbindung mit Ein-/Ausgabe-Geräten und stellt die Ergebnisse der Anfragen dar.

Die Anwendungsebene (Business Logic Layer) kümmert sich um die betriebswirtschaftliche Logik. Sie ist verantwortlich für die Bearbeitung der Benutzeranfragen und Rückgabe der Ergebnisse an die Präsentationsebene. Die betriebswirtschaftliche Logik stellt die Implementierungen der im Abschnitt 2.3 genannten Komponenten.

Die Datenhaltungsebene (Database Layer) speichert und verwaltet alle vom ERP-System benötigten Daten in einer Datenbank. Der Zugriff auf diese Daten erfolgt mit SQL. Meist unterschiedliche Datenbanksysteme können in einem ERP-System verwendet werden.

## 2.5 ERP Open Source

Der Begriff Open Source Software oder Quelloffene Software ist problematisch, da er häufig ausschließlich in Verbindung mit den Lizenzen der FSF (GNU-GPL, GNU LGPL und GNU-Lizenz für freie Dokumentation) verwendet wird. Im Anhang A 02 zeigt eine Tabelle der möglichen Lizenzen und ihre Rechte. Diese Lizenzen sind zwar frei, aber sie fordern, dass abgeleitete Werke die gleichen Freiheiten gewähren. Die GNU-Lizenz für freie Dokumentation bietet die Möglichkeit, die Änderung ganz bestimmter Abschnitte zu verbieten. Am weitesten verbreitete freie Softwarelizenz ist die GNU GPL. Sie hat folgende Eckpunkte:

- Das Programm darf ohne Einschränkungen, auch kommerziell, genutzt werden.
- Der unveränderte Quelltext des Programms darf kopiert und verbreitet werden.
- Das Programm selbst oder Teile davon dürfen verändert werden.
- Das veränderte Programm darf vervielfältigt und verbreitet werden. Es muss Hinweise zur Veränderungen enthalten.

Open-Source-ERP-Systeme sind auch zusätzlich zu den Proprietären Lösungen wie SAP oder Oracle verfügbar.

Für diese Systeme fallen die Lizenzkosten nicht an oder sie sind sehr gering, da diese Systeme durch eine Gemeinschaft von Entwicklern oder durch Software-Firmen konzipiert werden. Sie ermöglichen einen freien Zugang zu den Quelltexten von ERP-Systemen, erlauben es Unternehmen, diese Quelltexte zu verändern und geben ihnen die Möglichkeit, das System an andere Bedürfnisse anzupassen.

Die folgende Tabelle stellt Unterschiede zwischen OS-ERP-Systemen und Proprietären Systemen dar, abgeleitet von der Tabelle der Firma Catura [CatAD]. Mithilfe dieser Tabelle kann beantwortet werden, ob die OS-ERP-Systeme die Proprietären ersetzen können.

	<b>Proprietäre ERP-Systeme</b>	<b>Open-Source-ERP-Systeme</b>
Entwicklungskosten	In den Lizenzkosten enthalten	Nein
Quellcode-Verfügbarkeit	Nein	Ja
Konsistentes, regelbasiertes System	Ja	Ja
Lizenzkosten	Ja	Nein
Sicherheit	Mittel	Hoch.  Eine weltweite Entwickler Community schließt Sicherheitslücken bei Bekanntwerden
Individuelle Funktionen	Nein	Ja
Erweiterbarkeit	Mittel	Hoch
ERP-Know-How	Ja	Ja
Kosten für Wartung & Support	Mittel bis Hoch	Mittel

**Tabelle 2.1: Vergleich OS-ERP-Systeme und Proprietäre Systeme**

Aus dieser Tabelle lässt sich schlussfolgern, dass man mit Proprietären ERP-Systemen von einem Anbieter abhängig ist und hohe Lizenzkosten zahlt. Im Gegenteil dazu hat man mit Open-Source-ERP-Systemen volle Kontrolle über das System, das auf einer standardisierten, bewährten Lösung beruht. Mittels des Quelloffenen Codes besitzen sie eine große Flexibilität bei der Entwicklung. Darüber hinaus sind spezifische Anpassungen möglich.

Die meisten Open-Source-ERP-Systeme zielen die kleinen und mittleren Unternehmen (KMU), die selbst das System einführen und an den eigenen Bedarf anpassen oder durch IT-Firmen, die OS-ERP-Systeme erweitern und verkaufen, lassen machen, wenn die KMU keine DV-Abteilung haben.

Die OS-ERP-Systeme können kostenfrei von den Internetseiten der Hersteller oder der Entwicklergemeinschaft heruntergeladen werden. D.h. der Quellcode ist frei verfügbar und man hat das Recht ihn zu verändern und weitergeben. Es entstehen keine Lizenzkosten für die Nutzung der Open-Source-ERP-Systeme. Es

entstehen nur niedrige Implementierungskosten- und Zeit für die IT-Firmen und Wartungskosten für KMU. Es fallen die Kosten der Dienstleistungen (Supports, Abschluss eines Servicevertrags oder der Kauf von Zusatzmodulen), die von den IT-Firmen oder von Entwickler des OS-ERP-Systems unterstützt werden.

Die niedrigen Implementierungskosten und geringen Entwicklungszeiten sind Gründe für IT-Firmen, um sich für OS-ERP-Systeme zu entscheiden. Im Anhang A 03 ist beispielhaft dafür eine Umfrage des „Netzwerk Elektronischer Geschäftsverkehr“ (NEG) über die Erwartungen des Unternehmens der Benutzung von OS-ERP-Systemen [UmfNEG] angeführt.

Der obere analytische Vergleich liefert die Ergebnisse, dass OS-ERP-Systeme einige Vorteile gegenüber Proprietären haben. Aus diesem Grund sollen solche Systeme betrachtet werden. Es stellen sich wichtige Fragen: Besitzen die OS-ERP-Systeme schon die Reife für den produktiven Einsatz im Unternehmen? Können die OS-ERP-Systeme die Proprietären Lösungen ersetzen?

Diese Fragen werden im Laufe dieser Arbeit diskutiert und beantwortet.

### 3 Das ERP-System DiPPS

In diesem Kapitel wird das ERP-System DiPPS, seine Architektur und seine Funktionalitäten diskutiert.

#### 3.1 Analyse der Systemarchitektur von DiPPS

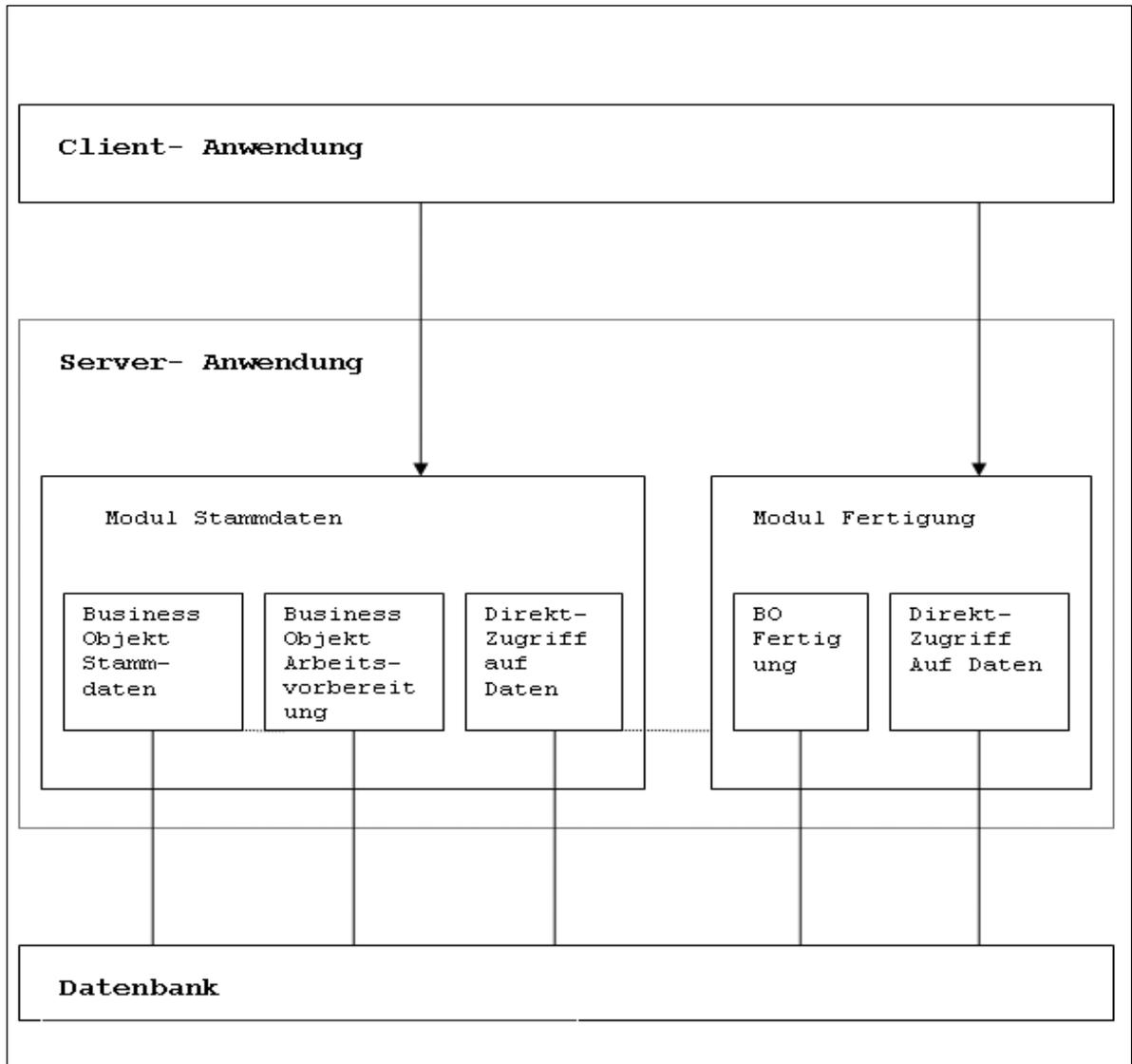
Das DiPPS- System wurde mit der Programmiersprache Delphi 7 entwickelt. Es basiert auf einem dreischichtigen Modell. Dieses besteht aus logischen Einheiten, die auf unterschiedlichen Rechnern ausgeführt werden können und zusammenarbeiten. Dadurch können die Vorteile der zentralisierten Anwendungslogik und der Thin-Client-Anwendungen (Endgerät-Anwendungen im Netzwerk) genutzt werden. Die mehrschichtige Anwendung besteht aus folgenden drei Bereichen:

**1. Client-Anwendung:** erlaubt die Verbindung zwischen dem Endbenutzer und der Anwendung, indem sie die Benutzeroberfläche auf dem Rechner des Benutzers bereit stellt.

**2. Anwendungsserver:** bearbeitet die vom Client angegebene Daten und liefert die Ergebnisse zurück und stellt allgemeine Datendienste, wie Verwaltung und Überwachung der Datenbank, zur Verfügung. Er befindet sich an einer zentralen Position im Netzwerk, auf die alle Clients zugreifen.

**3. Datenbank-Server:** speichert und verwaltet alle von DiPPS benutzten Daten. Er stellt das relationale Datenbankverwaltungssystem (RDBMS) bereit.

Diese Architektur ermöglicht, dass die Client-Anwendungen alle auf dieselbe Mittelschicht, den Anwendungsserver, zugreifen können. Dies wird durch die Kapselung der Anwendungslogik in einer freigegebenen Ebene ermöglicht. Dadurch können Redundanzen vermieden und Wartungskosten gesenkt werden. Zusätzlich kann die Verteilung der Anwendungsoperationen auf mehrere Rechner die Lastenverteilung erhöhen.



**Abbildung 3.1: Logische Darstellung der Client-Server-Architektur von DiPPS**

Für die Umsetzung einer mehrschichtigen Anwendung bietet Delphi die MIDAS Technologie (Multi-tier Distributed Application Services Suite) auf der Basis von DCOM. Diese erlaubt es, mit Hilfe komplexer, beliebig konfigurierbarer und ausbaufähiger Komponenten den Datenaustausch von Datenbankinformationen zwischen Client-Anwendungen und Anwendungsserver zu realisieren. Zusätzlich kann der als „Daten-Broker“ bezeichnete Anwendungsserver durch Business-Objekte für Logik erweitert werden.

Im Folgenden werden die drei Schichten ausführlich beschrieben:

### 1. Client-Anwendungen

Die Benutzerschnittstelle ist als MDI (Multi Document Interface) gestaltet. Es können mehrere Fenster innerhalb des Hauptfensters geöffnet und verwaltet werden. So können die Unterfenster beliebig angeordnet werden, ohne das Main-Fenster zu verlassen. D.h. die Unterfenster verhalten sich zum Programmfenster so, wie sich das Programmfenster zum Desktop verhält und es gibt die Möglichkeit an mehreren Dokumenten zu arbeiten.

Die Fensterhierarchie beginnt mit dem Hauptfenster, dann können Formulare aktiviert werden. Danach wird ein komplexer Dialog realisiert. Dort können Detailformulare geöffnet werden. Diese Dialoge sind modal, d.h. der Nutzer muss die Interaktion im modalen Fenster erst beenden, um in das auslösende Formular bzw. in das Menü zu gelangen.

Da mehrere Fenster einer Bedienoberfläche ein ähnliches Layout besitzen und gemeinsame funktionelle Eigenschaften haben, ist eine Klassifizierung von Formularen und die darauffolgende Entwicklung einer Objekthierarchie notwendig. Das Layout und die Eigenschaften werden nur einmal implementiert und an untergeordnete Objekte weitergegeben.

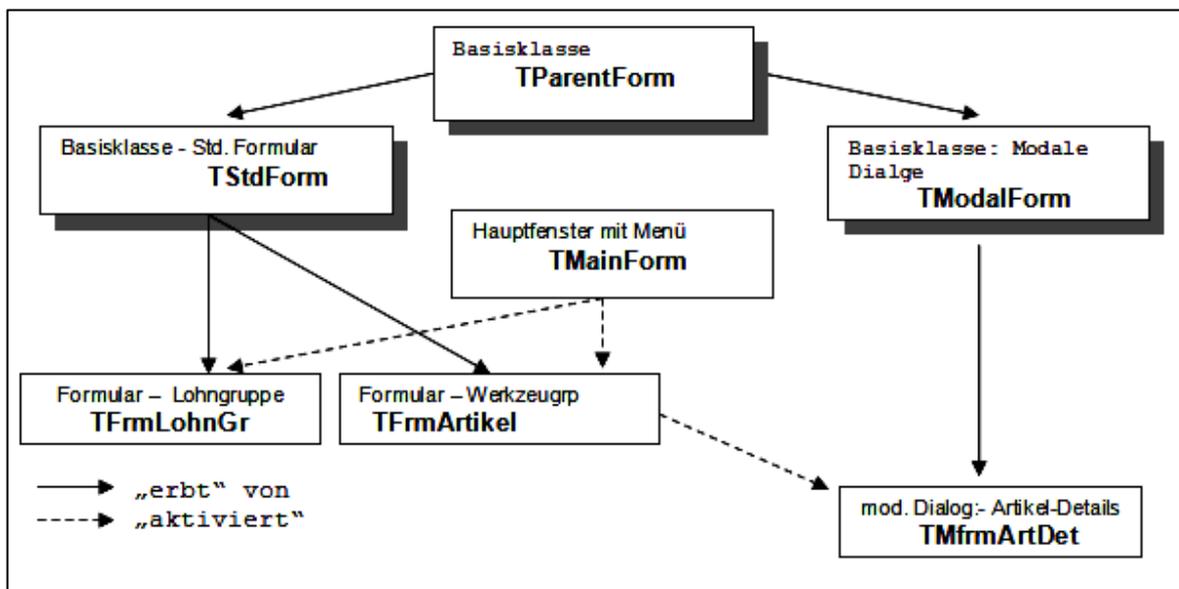


Abbildung 3.2: Objekt- und Fensterhierarchie der Benutzeroberfläche

## **2. Server-Anwendung**

Sie besteht aus zwei Modellen: einem logischen und einem physischen. Das logische Modell besteht aus sogenannten Business-Objekten, die nach Modulen aufgeteilt sind. Alle Business-Objekte stellen die Anwendungslogik dar. Einerseits enthalten sie komplexere Funktionalitäten und stellen Funktionen für die Client-Anwendung zur Verfügung. Andererseits stellen sie Datenmengen zur Darstellung bzw. zur Bearbeitung bereit.

Das physische Modell beschreibt, wie das logische Modell technisch umgesetzt wird, wie also der Zugriff auf Datenmengen und Business-Objekte realisiert werden kann.

Hierfür wird DCOM in Verbindung mit einigen MIDAS-Komponenten verwendet. Das erlaubt die Ausführung der entsprechenden Objekte auf einem entfernten Rechner. Die Kommunikation zwischen einer Datenmenge und einem Anwendungsserver erfolgt über eine IProvider-Schnittstelle, die Teil des Remote-Datenmoduls des Anwendungsservers ist.

Die Server-Anwendung besteht aus vielen Serverprogrammen, die COM-Objekte beinhalten. Jedes COM-Objekt umfasst die Funktionalität eines Moduls und stellt diese über Methoden der Client-Anwendung zur Verfügung.

## **3. Datenbank-Server**

Als Anforderung an das DiPPS wurde eine unabhängige Datenbank-Konnektivität spezifiziert, da das Datenbanksystem nach Kundenwunsch als SQL-Server oder Oracle ausgelegt werden soll. Dies wird durch ausschließliche Verwendung von Aliassen ermöglicht, die auf entsprechende BDE (Borland Database Engine)-Treiber Bezug nehmen.

Die Datenunabhängigkeit erfordert bei der DB-Programmierung den Verzicht auf Programme in datenbankspezifischer Sprache (Bsp. Stored Procedures, Trigger).

DiPPS verfügt über drei Datenbanken, nämlich eine System-Datenbank, eine Text-Datenbank und eine Daten-Datenbank: Die System-Datenbank verwaltet z. B. die Parameter, Nutzer, Mandanten, Zugriffsrechte, Mails, Schnittstellen. Die Text-Datenbank organisiert die Text-Sprachen, Beschriftungen, Message und Messagetexttabellen. Die Daten-Datenbank sorgt für die Verwaltung der DiPPS-Daten, wie Artikel, Konten, Kalender, etc.

### **3.2 Module und Funktionalität von DiPPS**

DiPPS unterstützt verschiedene Module, die auch verschiedene Funktionalitäten besitzen. Diese Module decken den gesamten Geschäftsablaufprozess eines Unternehmens ab.

Die Abbildung 9.1 am Anhang A04 fasst die Module und Funktionalitäten von DiPPS zusammen.

Die Module Kernsystem, Stammdaten, Materialwirtschaft, Einkauf, Verkauf, Fertigung, Kalkulation, Auswertungen und Schnittstellen sind in DiPPS vorhanden. Sie leisten unterschiedliche Aufgaben. Diese müssen bekannt sein, um letztendlich feststellen zu können, ob das neue System den Anforderungen gerecht wird und möglichst alle bisherigen Aufgaben erfüllt.

Im Kernsystem werden die Konfigurationsparameter, die Dokumente, Nutzer sowie die Datensicherung definiert und verwaltet.

Das Modul Stammdaten ist verantwortlich für die Einführung und Bearbeitung der Daten im System (Artikeldaten, Firmen/Personen, Kunden- und Geschäftspartnerdaten, Stücklisten, Arbeitszeiten, Kostenstellen, Lohn, etc).

Im Materialwirtschaftsmodul wird die Bestandsführung, Inventur, Lagerverwaltung, Chargenführung, Qualitätskontrolle für den Wareneingang, Materialbedarfsermittlung und Chargenverfolgung durchgeführt.

Das Einkaufsmodul führt Anfragen und Preisvergleiche, Bestellungen, Wareneingänge, Rechnungsprüfungen, Rahmenverträge aus.

Im Verkaufsmodul werden Angebote, Kundenaufträge, Warenausgänge, Ausgangsrechnungen, Rahmenverträge und Reklamationen erstellt.

Die Fertigungsaufträge, Werkstattaufträge, Kooperationsaufträge, Grobplanungsfertigung und Material- und Zeitbuchung werden im Modul Fertigung behandelt.

Die Kalkulation führt die Angebotskalkulation sowie die mitlaufende Kalkulation und Nachkalkulation für den Fertigungsauftrag aus.

Das Modul Auswertungen fördert den Druck und Export in verschiedene Formate und unterstützt das Management Informations-System und den Abfragegenerator.

Das Schnittstellen-Modul verbindet DiPPS mit anderen Schnittstellen, wie DATEV, ProFib, erlaubt den Datenexport in andere Formate, z.B. XML oder Excel, und den Datenabgleich mit externen Arbeitsplätzen.

### **3.3 Zusammenfassung und Fazit**

Es wurde in diesem Kapitel ein Überblick über die Systemarchitektur, die Module sowie die Funktionalitäten von DiPPS gegeben.

DiPPS unterstützt auf der Basis von DCOM MIDAS-Technologie die Mehrschichtenarchitektur. Die Client-Anwendung ist in Form von MDI bedienbar. D.h. es können gleichzeitig mehrere Dokumente geöffnet werden, die dann in separaten Unterfenstern angezeigt werden.

Die Anwendungslogik besteht aus allen Business-Objekten, die in Module aufgeteilt sind. Sie erlauben die Kommunikation zwischen der Client-Anwendung und der Mittelschicht, um Tabellen-Daten direkt zu visualisieren oder zu modifizieren.

DiPPS ist datenbankunabhängig, indem BDE zum Zugriff auf relationale Datenbanken benutzt wurde.

DiPPS unterstützt verschiedene Module und Funktionalitäten, die in Proprietären ERP-Systemen vorhanden sein sollen.

Die DCOM weist einige Nachteile auf: sie ist proprietär, von speziellen Microsoftplattformen abhängig und läuft zurzeit auf diesen (NT ab 4, Windows ab 95) am stabilsten. Der Einsatz von DCOM außerhalb des Unternehmens wird verhindert, wenn die Firewall die Übertragung blockiert. Dazu ist DCOM aufwendig zu installieren und zu konfigurieren. Die Programmiersprache Delphi 7 ist veraltet und dazu kostenpflichtig. Zusätzlich benutzt die BDE 32 Bit-Komponenten, die nicht mehr weiterentwickelt werden. Das alles weist darauf hin, dass das jetzige System nicht zukunftsweisend ist und ersetzt werden muss.

## 4 Migration von ERP-Systemen

Dieses Kapitel beschäftigt sich erstens mit den Migrationsmöglichkeiten der ERP-Systeme und zweitens mit dem Ablauf der Datenmigration innerhalb der ERP-Systeme.

### 4.1 Begriff und Möglichkeiten der Migration

Der Begriff „Migration“ leitet sich vom lateinischen *migrare* ab, was „Wanderung“ oder „Übersiedlung“ bedeutet. Hieraus abgeleitet bezeichnet IT-Migration den Wechsel oder Übergang zu einer Lösung. [Mig]

Dafür haben Firmen, die vor einer ERP-System-Migration stehen, die folgenden vier Möglichkeiten: „[...]“

1. Das alte System beibehalten wie bisher.
2. Das alte System in eine moderne Umgebung migrieren.
3. Das alte System in einer modernen Umgebung neu entwickeln.
4. Das alte System durch Standardsoftware ersetzen.“ [SoftMig]

Jede Möglichkeit hat Vor- und Nachteile. Bei der ersten vermeidet man alle Risiken (Datenintegration, Kosten und Zeitaufwand), die mit einer Migration verbunden sind. Die zweite und dritte Alternative haben der Vorteil, dass das neue System modern wird: mit der neuen Systemarchitektur können die Probleme der Wartung und Administration gelöst werden. Außerdem werden die Prozesse effizienter und schneller.

Für die neue ERP-Systementwicklung braucht man viel Geduld. Man benötigt viel Zeit für die Konzeption der Geschäftsprozesse, den Entwurf und die Implementierung. Die Kosten sind auch hoch. Eine Neuentwicklung sowie eine Reimplementierung in einer neuen Technologie sind stets kostenintensive und risikoreiche Unterfangen. Generell sind individuelle ERP-Systeme nicht zukunftsorientiert. Es werden lediglich neue Elemente gemäß der benötigten Bedürfnisse immer neu implementiert.

Die letzte Variante hat den Vorteil, dass das alte ERP-System in Betrieb bleibt, während eine Standard-ERP-Lösung eingeführt wird. Diese Systeme sind schon implementiert. Sie benötigen zwar unterschiedlich Zeit für Einarbeitung, Einführung

und Anpassung, aber man spart dafür die Implementierungskosten und -zeit. Generell sind diese Systeme deshalb global und zukunftsorientiert.

Man könnte die IT-Migration in vier Arten unterscheiden [KHMIg]:

1. Software-Migration
2. Datenmigration
3. Hardware-Migration
4. Anwendungsmigration

Bei der Software-Migration geht es um einen grundlegenden Wechsel der Software-Infrastruktur. Sie hat das Ziel, die alte Anwendung in einer neuen technologischen Umgebung ablauffähig zu erhalten. Die Datenmigration bezeichnet die Extraktion der Daten vom alten System und das anschließende Laden in das Zielsystem. Dadurch wird die Übernahme vorhandener Daten aus bestehenden Anwendungen mit hoher Qualität und Effizienz gewährleistet. Bei der Hardware-Migration handelt sich um die Migration von vorhandenen Systemen auf neue Hardware, z.B. der Umzug auf neue Server. Diese Art von Migration ist mit Software-Migration verbunden.

Unter Anwendungsmigration versteht man das vollständige Ersetzen des alten Systems durch ein neues. Diese Migration enthält sowohl Elemente der Software-Migration als auch der Datenmigration. Es könnte auch Hardware-Migration einschließen, wenn das neue System auf einer anderen Hardware läuft.

Die Migration eines ERP-Systems kann für Unternehmen aus verschiedenen Gründen notwendig werden [KHMIg]:

1. Hardware fällt aus, z.B. wenn neue Treiber nicht unterstützt werden.
2. Software benutzt veraltete Technologien, z.B. bei Datenübertragungen und Benutzeroberflächen
3. Software ist neuen, gehobenen Ansprüchen nicht mehr gewachsen

Zu einem gewissen Zeitpunkt nach der Einführung trägt das alte System den gehobenen Ansprüchen der laufenden Prozesse nicht mehr Rechnung und kann nur wenig bis gar keinen Spielraum mehr für nachträgliche Erweiterungen und Anpassungen aufweisen.

Die Dresden Informatik GmbH führt seit 2000 Soft- und Hardwaremigrationen bei den Anwendern von DiPPS durch. Zunehmend wird beobachtet, dass

Anwender die Migration v. a. wegen Benutzung veralteter Technologien in Betracht ziehen. Will die Firma am ERP-Markt erfolgreich bleiben, muss sie dieser Tendenz Rechnung tragen. Sie muss ihren Kunden die Anwendungsmigration auf ein neues ERP-System anbieten können. Für Dresden Informatik ist von Vorteil, dass die Firma die Möglichkeiten der Anwendungsmigration nutzt. So werden einerseits Implementierungskosten und -zeit gespart. Andererseits kann das Unternehmen durch DiPPS auf dem ERP-System-Markt bestehen bleiben, bis die Durchführung und Einarbeitung des neuen ERP-Systems komplett und erfolgreich erfolgt ist.

Damit die Migration zu einem neuen ERP-System gelingen kann, müssen folgende Punkte berücksichtigt werden:

1. Klare und vollständige Anforderungsdefinition: aktuelle und zukünftige funktionale und nicht funktionale Anforderungen festlegen
2. Realistische Erwartungen und klare Ziele: z.B. Kosten, Offenheit des Quell-Codes und Flexibilität berücksichtigen
3. Realistischer Zeitplan: genauen und umsetzbaren Zeitstruktur für die Einführung des neuen OS-ERP-Systems erstellen
4. Qualifikation der Mitarbeiter durch Schulungen gewährleisten

## **4.2 Ablauf der Datenmigration**

Alle ERP-Einführungen oder Implementierungen verlangen eine Migration der Daten aus einem oder mehreren Altsystemen.

In diesem Kapitel (basiert auf [DaMig]) wird ein Überblick über die Datenmigration, ihre Planung und Einsetzung gegeben.

### **4.2.1 Datenmigration**

Die Datenmigration ist notwendig bei der Einführung eines neuen Systems, das existierende Aufgaben eines oder mehrerer anderer Systeme übernimmt. Sie erfolgt in bestimmten Schritten, die in folgender Abbildung schematisch dargestellt sind:



Abbildung 4.1: Diagramm für die Schritte der Datenmigration

**Datenerfassung oder -sammlung (Data Collection):** Hierbei werden ganze Dateneinheiten hinsichtlich bestimmter Bereiche und Attribute in Abhängigkeit existierender Datensätze gesammelt. In der Regel wird dieser Schritt manuell durchgeführt.

**Datenextraktion (Data Extraction):** Die Daten werden aus den Quellen (Legacy-Systeme) extrahiert und für die Bereinigung sowie die anschließende Transformation bereitgestellt.

**Datenbereinigung und Harmonisierung (Data Cleansing & Harmonisation):** In der Datenbereinigung werden die falschen Daten entfernt oder korrigiert. Dies geschieht in zwei Etappen:

1. Bereinigung der Daten im Legacy-System vor der Extraktion, z.B. Schließung offener Bestellungen oder Deaktivierung veralteter Produkte.
2. Manipulation der Daten nach der Extraktion.

Datenharmonisierung ist die Standardisierung der Daten von verschiedenen Quellen zu einem einheitlichen Format. So können beispielsweise Produkt-,

Kunden- und Lieferanten-Codes in unterschiedlichen Formaten im bestehenden System vorliegen und müssen für das neue System vereinheitlicht werden.

**Datentransformation (Data Transformation):** extrahierte und bereinigte Daten müssen in ein einheitliches Datenschema umgewandelt werden. Die Transformation findet in einem eigenen Arbeitsbereich statt, der ETL-Technologie<sup>1</sup> benutzt.

**Datenladen (Data Loading):** Nach der Bereinigung und Transformation müssen die Daten aus dem Arbeitsbereich ins Zielsystem eingebracht werden.

**Datenmigration-Testen (Data Migration-Testing):** Datenmigration-Testen kann sowohl manuell als auch durch ein Programm realisiert werden. Eine Kombination aus beidem wird generell benutzt. Es sollen Etappen für den Test definiert und anschließend verfolgt werden.

**Datenarchivierung (Data Archival & Data Retention):** Die archivierten Daten werden nicht in die Datenmigration zu ERP-Systemen einbezogen. Es ist notwendig, eine Methode zur Speicherung von historischen Daten zu definieren, um verschiedene Anforderungen in der Organisation zu erfüllen.

#### 4.2.2 Planung der Datenmigration

Dieser Abschnitt (gestützt auf [DaMig]) beschreibt einige der wichtigsten Anliegen für den Projektmanager eines Datenmigrations-Projektes. Nur mit einer guten Planung für eine Datenmigration kann diese auch erfolgreich sein.

**Migrationsstrategie:** Zunächst soll eine Strategie für die Datenmigration entwickelt werden, um zu definieren, was zu migrieren ist. Die Strategie soll sowohl eine Datenbreite als auch -tiefe abdecken, d. h. es müssen bestimmte Entitäten und zugehörige Datenmengen zur Migration zur Verfügung gestellt werden. Beispielsweise können Daten eines ausgewählten Lieferanten migriert

---

<sup>1</sup> Erklärung s. 4.2.2

werden (Breite), aber nur die, die den Einkauf der drei letzten Jahre betreffen (Tiefe).

**Datenmigration-Teams:** Der Aufbau eines Datenmigration-Teams mit den entsprechenden Fähigkeiten für das Migrationsprojekt in Firmen allgemein wird zu einem großen Teil den Erfolg oder Misserfolg bestimmen.

Das Team kann wie folgt gegliedert werden:

1. Projektmanager: idealerweise sollte er Erfahrungen in Datenmigration und ERP haben.
2. Datenbankentwickler: er führt die programmatische Datenmigration durch. Im Idealfall sollte er mit den Legacy-Systemen vertraut sein.
3. Unternehmer: Anspruchs- oder Interessengruppe wird benötigt, um die Phasen der Migration freizugeben und zu schließen.
4. Business-Analytiker: er soll an verschiedenen Stellen der Migration beteiligt werden, nämlich von der Definition der Datenfelder bis zum Testen und zur Validierung der Datenmigration.

**Mögliche Technologien zur Datenmigration:** Spreadsheets und ETL-Programme sind die zwei Technologien, die in der Vorbereitung der Daten zum Einsatz kommen, damit diese vom ERP-System geladen werden können.

1. Spreadsheets: werden in der Regel für die Datenerfassung und -überprüfung verwendet. Sie sind einfach zu bedienen und verständlich. Das Risiko bei der Verwendung von Spreadsheets ist ihre Flexibilität: die Daten werden bei der Zuordnung zu verschiedenen Nutzern häufig bedeutungslos, sodass die eindeutige Zuweisung eines Datensatzes zu einem bestimmten Benutzer oft nicht möglich ist.
2. ETL (Extract Transform Load): die ETL-Tools vereinfachen die Aufgabe der Extraktion der Daten aus Legacy-Systemen und erlauben den Analytikern, programmatische Datenbereinigung, Transformation und Harmonisierung der Daten vorzunehmen. Tools wie z.B. Microsoft SSIS und Open Source Pentaho [Penta] bieten leistungsstarke grafische Daten-Modellierungs-Tools an, die Datentransformationen vereinfachen können.

**Planung für die Prüfung:** Die ERP-Implementierung soll prüffähig sein. Dazu gehört auch die Prüfbarkeit der Datenmigration. Deshalb wird eine Dokumentation von Beginn des Projektes an erstellt, sodass die künftige Prüfung erleichtert und jede Etappe der Datenmigration überprüfbar wird, z. B. die Prüfung der Datenintegrität gleich nach der Schemamigration.

**Dual-Wartung:** Es gibt immer eine zeitliche Verzögerung, oft eine Frage von Wochen, zwischen der Einnahme von Daten aus den Legacy-Systemen, der Umwandlung in ein anderes Format und dem Hochladen ins neue ERP-System. Die Daten in den Legacy-Systemen werden ständig bereinigt und Skripte zur Harmonisierung der neuen Daten vorbereitet. Deshalb werden die Änderungen im Legacy-System im neuen System widergespiegelt.

### **4.2.3 Zusammenfassung**

Datenmigration in ERP ist ein in sich komplexes Projekt. Die Qualität und das Know-how des Migration-Teams und des Projektleiters sind der Schlüssel für ein erfolgreiches Migrationsprojekt.

Auf Grund der Themenanforderung und des -umfangs kann die Datenmigration in dieser Arbeit nicht bearbeitet werden.

## **5 Anforderungen an ERP-Systeme**

In diesem Kapitel werden verschiedene Anforderungen an ERP-Systeme, die der Auswahl eines geeigneten Systems dienen, festgelegt und erklärt. Diese können in zwei Kategorien geteilt werden: funktionale und nicht funktionale Anforderungen. Sie bestimmen, welche Funktionalitäten und Module das neue System enthalten soll.

Moderne ERP-Lösungen müssen nicht nur eine bestmögliche, funktionale Unterstützung anbieten, sondern dies auch bei einer sogenannten schlanken Implementierung gewährleisten und offen bleiben, um auf zukünftige Abläufe ohne große personelle oder finanzielle Belastungen angepasst werden zu können.

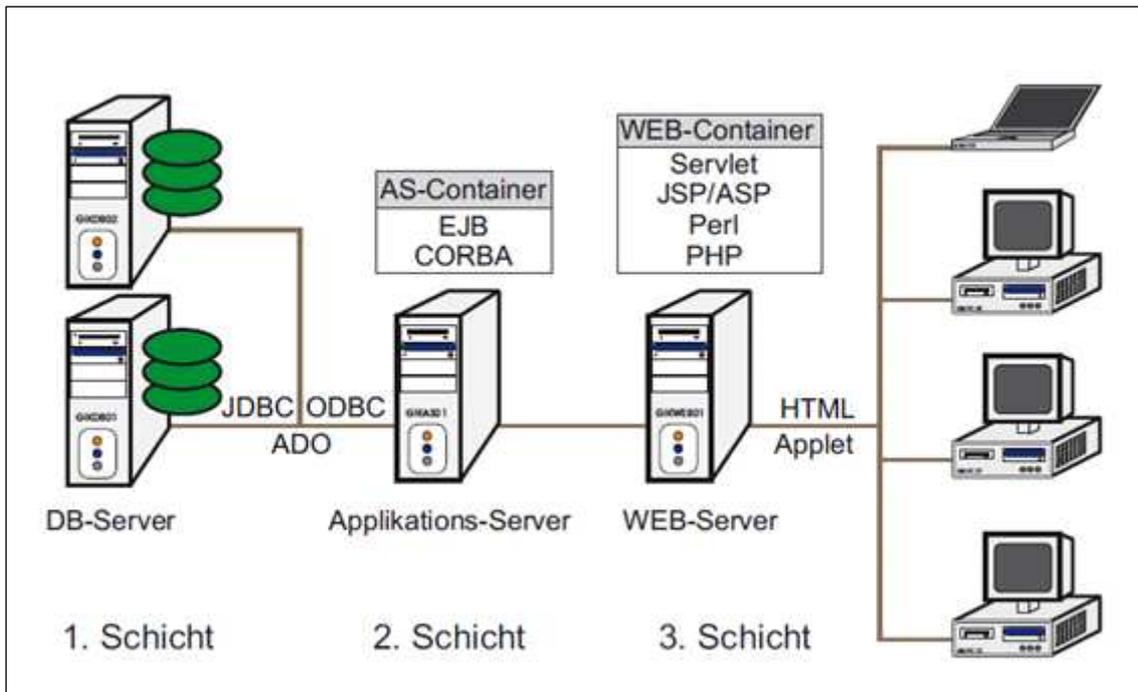
### **5.1 Funktionale Anforderungen**

Im Folgenden werden die wichtigsten Funktionalitäten eines ERP-Systems dargestellt und erklärt. Diese unterteilen sich in notwendige und in optionale Anforderungen.

#### **5.1.1 Mehrschichtenmodell**

Im Allgemeinen sind Mehrschichtenmodelle die Grundlage für moderne Client-Server-Anwendungen. Dadurch wird die Komplexität der Abhängigkeiten innerhalb des Systems reduziert und somit eine geringere Kopplung bei gleichzeitig höherer Kohäsion der einzelnen Schichten erreicht [ModStr]. Dazu sind mindestens zwei Schichten notwendig: Anwendung und Datenbank. Um flexibler in der Rechner- und Netzarchitektur zu sein, sollten aber mindestens drei Schichten angestrebt werden. Diese drei Schichten bestehen aus:

- der Client- Anwendung, die vorrangig die Anzeige der Daten und die Realisierung der Maskenfunktionalität enthält.
- der Server-Anwendung, die komplexe Datenbank- und Rechenoperationen ausführt.
- dem Datenbanksystem, das ausschließlich die Daten bereitstellt.



**Abbildung 5.1: Darstellung der Drei-Schichten-Architektur**

Eine Vermischung der Funktionsumfänge der drei Schichten sollte unbedingt vermieden werden, da sonst die Schichten nicht mehr austauschbar sind. Wenn beispielsweise eine Funktionalität über „Trigger“ oder „Stored Procedures“ in der Datenbank realisiert werden würde, könnte diese Funktionalität beim Austausch des Datenbanksystems nicht mehr erhalten bleiben.

Ein weiterer Vorteil dieses Schichtenmodells ist die Verteilung von Rechenkapazitäten. Die Client-Anwendung ist nicht rechenintensiv und kommt daher mit älteren Computern aus. Die Belastung durch die Server-Anwendung und das Datenbanksystem hängt von der Nutzeranzahl und der Datenmengen ab. Diese Anwendungen können zusammen auf einem Rechner oder getrennt laufen. Die Kapazität des Rechners kann dabei individuell den jeweiligen Bedürfnissen angepasst werden.

Bei der Weiterentwicklung des Systems kann man außerdem noch entscheiden, ob zusätzliche Module aus dem oben genannten Grund in weitere Schichten implementiert werden. Dies hat wiederum den Vorteil, dass stark dynamische oder kundenspezifische Programmteile in eine extra Schicht (Serveranwendung) ausgelagert werden können, um somit nur Änderungen oder Anpassungen für diesen speziellen Teil vorzunehmen.

### **5.1.2 Modularer Aufbau**

ERP-Systeme bestehen aus verschiedenen Modulen. Jedes dieser Module dient einem bestimmten Unternehmensbereich und ist auf die entsprechenden Bedürfnisse und Anforderungen zugeschnitten.

Diese Module verwalten und bilden die Prozesse des Unternehmens ab. Sie können von einem ERP-Anbieter zu einem anderen variieren und unterschiedlich strukturiert sein. Die einzelnen Module bestehen selbst auch wieder aus Komponenten, die mehr oder weniger individuell passend auswähl- und kombinierbar sind. Das Modul „Einkauf“ z.B. besteht aus Anfrage-/Angebot-Abwicklung, Bestellabwicklung, Rechnungsprüfung und Lieferantenbeurteilung.

Die Module unterscheiden sich grundsätzlich in zwei Arten, zum einen die operativen und zum anderen die analytischen. Operative Module kommen bei konkreten Geschäftsfällen zum Einsatz, während analytische Module der Erfassung, Generierung und Auswertung von Daten dienen.

ERP-Systeme müssen folgende Module enthalten: Stammdatenverwaltung, Verkauf, Einkauf, Materialwirtschaft, Rechnungswesen, Vertrieb, Produktionsplanung, Fertigung, Kostenrechnung, Analyse und Leistungsmessung, Projektverwaltung und Administration sowie Systempflege.

Durch die Modularisierung soll einerseits die Möglichkeit bestehen, das Komplettsystem in Teilschritten zu realisieren, andererseits soll der Kunde die Möglichkeit erhalten, nur die benötigten Teile zu erwerben. Deshalb muss schon während der Auswahl und der Entwicklung darauf geachtet werden, dass alle Module in einzelne Programmteile getrennt werden.

### **5.1.3 Mandantenfähigkeit**

Das ERP-System soll in der Lage sein, für verschiedene Firmen gleichzeitig zu arbeiten, d. h. die Geschäftsvorgänge mehrerer Kunden (Unternehmen) unabhängig und getrennt voneinander abbilden zu können, ohne dass seitens der Mandanten ein gegenseitiger Einblick oder Zugriff auf deren Stammdaten (z.B. Adresse und Artikel) bzw. Bewegungsdaten möglich ist.

Die Mandanten sind folglich diejenigen, die die Nutzungsstruktur des Softwaresystems vorgeben. In der Praxis kommt die Mandantenfähigkeit z.B. bei

der chaotischen Lagerverwaltung zum Tragen. Wenn in einem Lager die Bestände verschiedener Firmen gelagert werden, können mit der richtigen Software jedem einzelnen Mandanten verschiedene Bestände unterschiedlicher Firmen zugeordnet werden. Umgekehrt kann die ERP-Software auch anhand verschiedener Bestände den einzelnen Mandanten erkennen. Hierfür werden mandantenabhängige und mandantenübergreifende Daten unterschieden. Mandantenabhängige Daten sind Angaben, die für jeden Mandanten individuell erfasst und verwaltet werden. Mandantenübergreifende Daten sind Informationen die das Management zur Ausübung seiner Aufgaben benötigt [Mand].

#### **5.1.4 Datenbankunabhängigkeit**

Da es keine 100%ige Datenbankunabhängigkeit gibt, sollen so viele verschiedene Datenbanksysteme wie möglich unterstützt und eine kurzzeitige Adaption auf weitere Datenbanken ermöglicht werden.

Diese Anforderungen sollen gewährleisten, dass jeder Kunde sein bereits existierendes Datenbanksystem nutzen kann oder das für ihn passende erwirbt. Ebenso erleichtert sie die Daten-Migration von einem Datenbanksystem zu einem anderen.

#### **5.1.5 Internetfähigkeit**

Unternehmen können sich durch die Nutzung des ERP-Systems über Internetbrowser die aufwändigen Installationen für jeden einzelnen Client sparen. Niederlassungen sowie Tochterfirmen im Ausland können jederzeit flexibel via Internet in vollem Umfang auf die eingesetzte ERP-Lösung zugreifen. Ebenso lassen sich Geschäftspartner direkt in das ERP-System einbinden. Mit einer solchen ERP-Lösung, auch ERP II genannt, wird die Zusammenarbeit in der Lieferkette erheblich vereinfacht; ebenso bietet die Internet-Architektur eine ideale Basis, um moderne flache Unternehmensstrukturen zu unterstützen sowie Prozesse durchgängig und transparent zu gestalten.

### 5.1.6 Releasefähigkeit

Die Releasefähigkeit optimiert die Gesamtkosten eines ERP-Systems, weil sie einen langen Lebenszyklus des Systems sichert. Dadurch werden die Funktionalitäten und Technologien erneuert, während kundenspezifische Änderungen und Erweiterungen erhalten bleiben. Dies vermeidet manuelle Nacharbeit beim Release-Update. [Release]

## 5.2 Nicht funktionale Anforderungen

Nicht-funktionale Anforderungen an ERP-Systeme beschreiben die folgenden Merkmale:

**Anpassungsfähigkeit (Customizing):** Softwareanpassungen an individuelle Kundenwünsche nennt man „Customizing“. Dies wird durchgeführt, ohne den Programmcode zu verändern.

Customizing muss releasefähig sein: Auch Jahre alte individuelle Anpassungen müssen nach dem Einspielen des nächsten Programm-Releases automatisch wieder verfügbar sein.

Anpassungen an Kundenwünsche passieren ausschließlich außerhalb des Programmcodes:

- durch Änderung von Bildschirmmasken
- durch Änderung des Datenbanklayouts
- durch Änderung von Formulardefinitionen
- durch Erstellen von Listen und Auswertungen
- und durch Scripting

ERP-Systeme werden immer in einem unternehmensneutralen Zustand ausgeliefert, daher sind Anpassungen bei der Einführung (Customizing) an die individuellen Aufbau- und Ablauforganisationen eines Unternehmens zwingend erforderlich.

**Zuverlässigkeit** (Systemreife, Wiederherstellbarkeit, Fehlertoleranz)

**Aussehen und Handhabung** (Look and Feel)

**Benutzbarkeit** (Verständlichkeit, Erlernbarkeit, Bedienbarkeit): Je verständlicher und übersichtlicher die Benutzeroberfläche gestaltet ist, desto schneller finden sich die Anwender mit dem neuen System zurecht.

**Leistung und Effizienz** (Antwortzeiten, Ressourcenbedarf, Wirtschaftlichkeit)

#### **Betrieb und Umgebungsbedingungen**

**Wartbarkeit, Änderbarkeit** (Analysierbarkeit, Stabilität, Prüfbarkeit, Erweiterbarkeit)

**Portierbarkeit, Übertragbarkeit** (Installierbarkeit, Konformität, Austauschbarkeit) [MAN]

#### **Anpassbarkeit**

**Sicherheitsanforderungen** (Vertraulichkeit, Informationssicherheit, Datenintegrität, Verfügbarkeit)

**Korrektheit** (Ergebnisse fehlerfrei)

**Flexibilität** (Unterstützung von Standards): Die Änderungen der Kundenbedürfnisse, der wirtschaftlichen Anforderungen an ein Unternehmen, des Lieferanten- oder Partnerumfelds fordern, dass das ERP-System flexibel auf Veränderungen reagieren kann. Es sollte sogar die Flexibilität eines Unternehmens erhöhen. Flexibilität eines ERP-Systems beschreibt im Kern die Möglichkeit, Prozesse und deren IT-Unterstützung schnell und effizient optimieren zu können, indem Veränderungen am System vorgenommen werden.

**Skalierbarkeit** (Änderungen des Problemumfangs bewältigen)

**Mehrsprachigkeit:** Das System muss erlauben die Maskenbedienung in mehreren Sprachen durchzuführen.

#### **Randbedingungen**

**Offenheit für Integration und Standards:** Die Integration eines ERP-Systems ist ein sehr kritischer Faktor, da sie aus einer komplexen Kombination von Daten-, Funktions-, Prozess- und Anwendungsintegrationen besteht. Das System muss Standardschnittstellen vorweisen, um eine einfache und schnelle Integration von zusätzlichen Softwarelösungen zu garantieren.

### **5.3 Zusammenfassung**

Es wurden in diesem Kapitel die wichtigen funktionalen und nicht funktionalen Anforderungen an ERP- Systeme definiert. Diese werden zur Auswahl eines geeigneten ERP-Systems dienen. Die funktionalen Anforderungen müssen außer der Internetfähigkeit alle im zukünftigen System enthalten sein.

## **6 Analyse bestehender Ansätze zur Auswahl eines OS-ERP-Systems**

In diesem Abschnitt werden die Ergebnisse zweier wissenschaftlicher Arbeiten zu OS ERP-Systemen vorgestellt und analysiert, um mit Hilfe der Ergebnisse eine geeignete Systemauswahl für die eigene praktische Arbeit treffen zu können.

Die erste Arbeit mit dem Thema „Open-Source-Software für das Enterprise Resource Planning“ von der Institut für Wirtschafts- und Verwaltungsinformatik, Fachbereich Informatik, Universität Koblenz-Landau, behandelt die Auswahl und Evaluierung von OS-ERP-Systemen auf effiziente Anwendbarkeit in Unternehmen zum Zwecke des Aufbaus eines „ERP-Future-Labs“, in welchem mittelständische Handelsunternehmen die dort installierten Systeme testen können. Diese Arbeit wurde im Jahr 2009 realisiert.

Die zweite Arbeit hat den Titel „Enterprise Resource Planning (ERP) auf Basis Freier Software – Zukunftsvision und Realität“ von M. Sc. Falk Neubert, BWL/Produktions-Management und Wirtschaftsinformatik, Universität Osnabrück. Diese Arbeit behandelt die Auswahl und die Bewertung der OS-ERP-Systeme und wurde im Jahr 2008 geschrieben.

### ***6.1 Open-Source-Software für das Enterprise Resource Planning***

Dieses Projekt wurde in sechs Phasen realisiert. Nach der Bildung eines Projektteams von drei Mitgliedern erfolgten Voruntersuchungen zu der Frage, welche Open-Source ERP-Systeme bereits existierten. Es schlossen sich Überlegungen zur Projektdefinition und –abgrenzung sowie der Entwurf des Projektplans an. Die dritte Phase wurde der fachlichen und DV-technischen Anforderungsanalyse gewidmet und im Zuge dessen eine Ist-Analyse (Problem- und Bedarfsanalyse) erstellt. Darauf aufbauend wurde ein Soll-Konzept mit den fachlichen Anforderungen an Funktionen und Prozesse entwickelt sowie eine Marktrecherche verfügbarer Open-Source ERP-Systeme durchgeführt. Im vierten Schritt folgte die Erstellung eines DV-Pflichtenhefts für Soft- und Hardware unter gleichzeitiger Festlegung der nötigen Evaluationskriterien. In der fünften Phase

wurde zunächst eine Grobauswahl nach KO–Kriterien vorgenommen und von einer Feinselektion anhand von Wunschkriterien vervollständigt. Dabei wurden die verschiedenen ERP-Systeme in Bezug auf die unterschiedlichen Wunschkriterien verglichen, beurteilt und bewertet. Daraus resultierte eine Tabelle (siehe Anhang A 05), in der die Ergebnisse der Untersuchung zusammengefasst wurden: jedes System erhielt Punkte entsprechend des Erfüllungsgrades des jeweiligen Kriteriums. Anhand der Summen der jeweiligen Einzelpunktzahlen wurde das System ausgewählt, welches den höchsten Erfüllungsgrad im Vergleich zu allen anderen untersuchten ERP-Systemen hatte. Im letzten Schritt wurde dieses System installiert und an die Bedürfnisse des Kunden angepasst.

Der Anforderungskatalog wurde auf Basis der bereits notwendigen Anforderungen an ERP-Systeme erstellt. Zwei Module, Buchhaltung und Vertrieb, wurden als zwingend erforderlich gezählt. Hinsichtlich dieser Kriterien wurden die auszuwählenden ERP-Systeme evaluiert und prototypisch dahingehend getestet.

Das Pflichtenheft wurde nach Balzert erstellt [Balz]. Es dient der Definition der Zielbestimmung, des Produkteinsatzes, der Produktumgebung etc.

Zur Durchführung der Recherche wurde die Webseite <http://www.sourceforge.net> benutzt. Von mehr als 100 000 Projekten wurden insgesamt 49 Software-Projekte in der Grobevaluation nach K.O.-Kriterien<sup>2</sup> beurteilt. Danach kamen für die Feinevaluation die OS-ERP-Systeme Openbravo, ADempiere, Lx-Office, Opentaps und TinyERP in Frage. Diese Systeme wurden installiert, anhand des Kriterienkatalogs getestet und erneut beurteilt. Die im Anhang A 05 dargestellten Tabellen fassen, wie zu Beginn dieses Kapitels erwähnt, die erarbeiteten Ergebnisse zusammen.

Festzuhalten ist, dass keine Evaluation des Systems TinyERP durchgeführt wurde. Das lag daran, dass die Übersetzung der Software ins Deutsche nicht möglich war.

Die Tabelle 9.8 im Anhang A 05 stellt den Vergleich der Erfüllungsgrade der evaluierten Systeme dar. Openbravo ERP belegt den ersten Platz mit der Erfüllung von 90,24% der K.O.-Kriterien und 67,29% der Wunschkriterien. Auf dem zweiten Platz liegt das System ADempiere mit 89,29% der Erfüllung der K.O.-Kriterien und

---

<sup>2</sup> Folgende K.O.-Kriterien wurden aufgeführt: das System soll Open Source sein, die Module „Vertrieb“ und „Buchhaltung“ sollen enthalten sein, das System soll in deutscher Sprache sein, die Software muss sich in produktivem Einsatz befinden und es muss ein aktives Projekt sein.

35,56% der Wunschkriterien. Mit etwas deutlicherem Abstand, besonders hinsichtlich der Wunschkriterien, nämlich mit 87,62% der Erfüllung der K.O.-Kriterien und 20,33% der Wunschkriterien, ist Lx- Office an der dritten Stelle. Auf dem letzten Platz, deutlich abfallend im Vergleich zu Openbravo, steht Opentaps mit dem Erfüllungsgrad von 78,10% bei den K.O.-Kriterien und 24,63% bei Wunschkriterien.

Auf Basis dieser Ergebnisse wurde die Installation von Openbravo ERP R2.33 beschlossen.

## **6.2 ERP auf Basis Freier Software- Zukunftsvision und Realität**

In diesem Projekt wurden einige Systeme für die weitere Analyse der OS-ERP-Systeme ausgewählt. Dafür wurde ein Kriterienkatalog entwickelt und eine Marktübersicht erstellt, um anschließend die OS-ERP-Systeme nach Funktionalität zu vergleichen, zu klassifizieren und schließlich zu bewerten.

Der Kriterienkatalog enthielt die folgenden Punkte:

1. Erfüllt die Lösung die Kriterien der Open Source Definition?  
Lizenzkosten, Downloadverfügbarkeit, Aktivitätsindex der Community, Dokumentationsqualität
2. Welchen Funktionsumfang bietet die Lösung?  
Vertrieb und CRM-Materialwirtschaft, Produktion, Rechnungswesen, Personalwesen
3. Kontinuierliche Entwicklung in den letzten Jahren
4. Gibt es professionellen Support?
5. Gibt es installierte Lösungen?

In der Marktübersicht wurden 13 OS-ERP-Systeme<sup>3</sup> identifiziert, von denen acht für die weitere Arbeit ausschieden, weil sie den Anforderungen des Kriterienkatalogs nicht genügten. Die übrigen ERP-Systeme Lx-Office, TinyERP, Openbravo, Opentaps und ADempiere wurden nach Funktionalität bewertet.

---

<sup>3</sup> ADempiere, Apache OFBiz, Compiere, ERP5, IntarS, Limbas, Lx-Office, Openbravo, Opentaps, SQL-Ledger, OpenMFG ERP, TinyERP und WebERP.

Zum Vergleich der Funktionalität wurden Fallstudien auf Basis „echter“ Daten benutzt (Mandanten, Benutzer, Produkt, Fertigungsauftrag). Es wurde der Lagerbestand geprüft und automatische Warenbewegungen angezeigt.

Zur Präsentation der Ergebnisse wurde eine Tabelle angelegt, in der entsprechend der Funktionalitäten der einzelnen Systeme Punkte eingetragen wurden. Bei einer Maximalpunktzahl von 20 beim Vertrieb, 12 bei der Materialwirtschaft, 10 für die Produktion, 9 beim Rechnungswesen und 3 für das Personalwesen ergab sich folgende Verteilung:

<b>Funktionalität</b>	<b>ADempiere</b>	<b>openbravo</b>	<b>opentaps</b>	<b>TinyERP</b>	<b>Lx-Office</b>
<b>Vertrieb/20</b>	17	15	15	12	11
<b>Materialwirtschaft/12</b>	9	9	7	7	3
<b>Produktion/10</b>	8	8	6	7	0
<b>Rechnungswesen/9</b>	7	8	7	6	6
<b>Personalwesen/3</b>	3	3	2	2	0

**Tabelle 6.1: Funktionalitätsvergleich einiger OS-ERP-Systeme**

Anhand der markierten Zahlen ist feststellbar, dass ADempiere und Openbravo deutliche Führungspositionen im Vergleich zu anderen Systemen einnehmen, ADempiere hat gegenüber Openbravo einen leichten Vorsprung. Es folgen Opentaps, TinyERP und schließlich Lx-Office auf dem letzten Platz.

### **6.3 Zusammenfassung und Fazit**

Beide Arbeiten zusammen genommen haben bewiesen, dass einige OS-ERP-Systeme offenbar einen bestimmten Reifegrad erreicht haben, um in Konkurrenz zu Proprietären Systemen zu treten. Aufgrund der Übernahme von Konzepten kommerzieller Software und Anpassungen an Kundenwünsche können OS-ERP-Systeme mittlerweile als Alternative zu Proprietären Systemen angesehen werden.

Bei der ersten Arbeit verblieben lediglich fünf Systeme von 49 in der Grobevaluation. Die 44 anderen schieden aus verschiedenen Gründen aus. Openbravo hat einen kleinen prozentualen Vorsprung vor ADempiere und Lx-Office. Diese drei Systeme können in diesem Fall folglich als die einzigen ernstzunehmenden Konkurrenten zu Proprietären ERP-Systemen bezeichnet werden.

In der zweiten Arbeit belegen die OS-ERP-Systeme ADempiere und Openbravo Spitzenplätze.

Die Ergebnisse beider Arbeiten weisen diesen beiden Systemen Führungspositionen im Vergleich zu anderen OS-ERP-Systemen zu, obwohl in jeder Arbeit verschiedene Anforderungen gestellt wurden. Zudem sind beide Arbeiten noch recht jung, sodass die Schlussfolgerung nahe liegt, dass sich v. a. Openbravo und ADempiere hinsichtlich ihrer Funktionalitäten weiterentwickelt haben.

## 7 Analyse und Konzeption

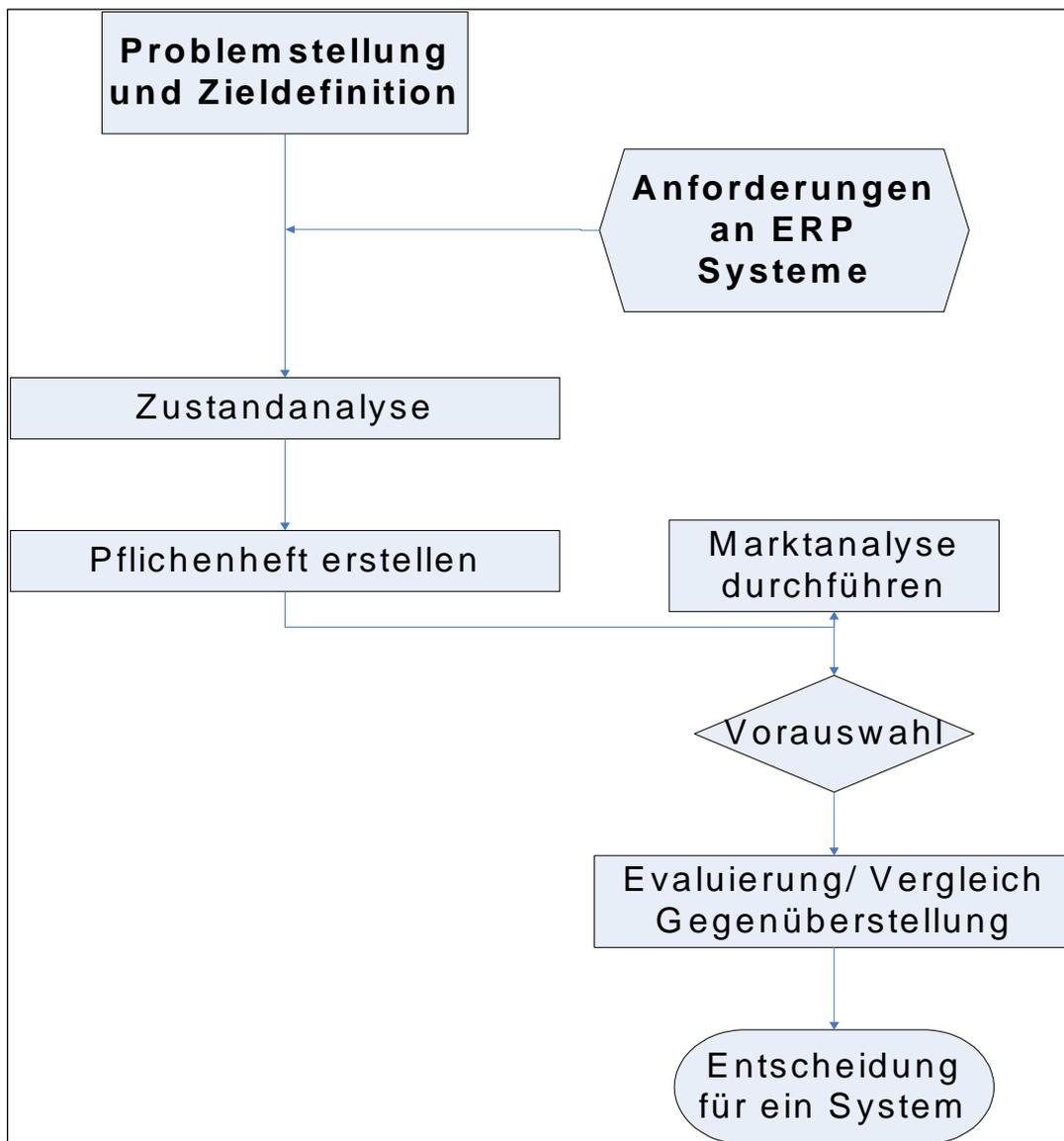
In diesem Kapitel werden Konzepte sowohl für die Auswahl eines ERP-Systems als auch für die Einführung einer kundenspezifischen Anpassung im ausgewählten System entworfen. Abschließend werden eine prototypische Implementierung und eine Auswertung der Ergebnisse vorgenommen.

### 7.1 Konzept für das Auswahlverfahren eines OS-ERP-Systems

Zu Beginn der Auswahlphase sollen die Anforderungen im Kapitel 5 betrachtet werden. Dann müssen die bestehenden Funktionalitäten und Module des DiPPS-Systems analysiert und die Ergebnisse in einem Pflichtenheft dokumentiert werden, um als Grundlage für die Anbietersuche genutzt werden zu können [ReLö]. Im Pflichtenheft sollen auch die Anforderungen, denen das bestehende System (in diesem Fall DiPPS) nicht genügen kann, aufgeführt werden. Open-Source spezifische Anforderungen (Dokumentationsqualität, Breite der Gemeinschaft...) fließen ebenfalls in die System-Auswahl anhand des Pflichtenhefts ein.

Nach dem Vergleich zwischen Proprietären und OS-ERP-Systemen in Abschnitt 2.5 sind vermeidbare Kosten, größere Flexibilität sowie die Unabhängigkeit von einem Hersteller die Hauptgründe für die Wahl eines OS-ERP-Systems. Durch die Verfügbarkeit des Quellcodes kann das System an die sich ändernden Anforderungen und Gegebenheiten innerhalb des Unternehmens angepasst werden. Deswegen wird sich die Auswahl in dieser Arbeit auf die Auseinandersetzung mit OS-ERP-Systemen beschränken.

Die folgende Abbildung stellt das Konzept für die Auswahl eines geeigneten ERP-Systems dar:



**Abbildung 7.1: Konzept für die Auswahl des OS-ERP-Systems**

Am Anfang des Auswahlverfahrens steht die Frage, warum das vorhandene System ersetzt werden sollte. Damit kann einerseits die Problemstellung umrissen und andererseits eine Zielvorstellung formuliert werden. Zweitens wird im Vorfeld aufgrund der in Kapitel 5 definierten Anforderungen an ERP-Systeme eine Zustandsanalyse unabhängig vom späteren System des Unternehmens durchgeführt. Darauf aufbauend wird dann ein Pflichtenheft erstellt. Es folgt die Marktanalyse zur Ermittlung potentieller Anbieter. Schließlich werden einige ERP-Systeme, die den Anforderungen nicht entsprechen, herausgefiltert, bevor die restlichen Systeme verglichen und evaluiert werden. Daraus resultiert die Entscheidung für ein System.

Es existieren im Internet einige Portale, die bei der Auswahl eines geeigneten ERP-Systems helfen können, z.B. <http://www.it-matchmaker.com>.

Bevor auf die Beschreibung der einzelnen Arbeitsetappen eingegangen werden kann, müssen notwendige Projektschritte mit ihren zugehörigen Aufgaben und der jeweiligen Zeitbeschränkung vorgestellt werden. Folgende Tabelle gibt einen Überblick über den geplanten Projektablauf:

Schritte	Aufgabe	Starttermin	Endtermin
Zielbestimmung	<ul style="list-style-type: none"> <li>• Zielgruppe: IT-Firma, die OS- ERP einsetzen und verkaufen will</li> <li>• Zielprodukt: OS-ERP-System auswählen und einführen</li> </ul>	-	-
Zustandsanalyse	DiPPS analysieren, dabei die Anforderungen an ERP-Systeme (s. Kapitel 5) berücksichtigen	15.02.2011	14.03.2011
Pflichtenheft anlegen	Prioritätenliste für die Anforderungen (funktional, nicht funktional) an das neue System erstellen.	15.03.2011	31.03.2011
Marktanalyse	Position der OS-ERP-Systeme auf dem IT-Markt bestimmen	01.04.2011	15.04.2011
Vorauswahl	<ul style="list-style-type: none"> <li>• OS-ERP-Systeme filtern und Vorauswahl treffen</li> <li>• Endet mit Auswahl von 2 bis 3 Systemen</li> </ul>	16.04.2011	30.06.2011
Evaluierung und Vergleich	<ul style="list-style-type: none"> <li>• die ausgewählten Systeme miteinander vergleichen, Auswahl einschränken</li> <li>• Endet mit 1 oder 2 Systemen</li> </ul>		
Endauswahl	<ul style="list-style-type: none"> <li>• Vergleich mit Pflichtenheft</li> </ul>		

	<ul style="list-style-type: none"> <li>• Auswahl eines Systems</li> </ul>		
Einführung des neuen Systems	<ul style="list-style-type: none"> <li>• Installation</li> <li>• Kundenspezifische Anpassungen</li> </ul>	01.07.2011	31.07.2011
Dokumentation	<ul style="list-style-type: none"> <li>• Systemdokumentation</li> <li>• Benutzerhandbucherstellung</li> </ul>	01.08.2011	14.08.2011

**Tabelle 7.1:Projektschritte**

### **7.1.1 Problemstellung und Zieldefinition**

Zu Beginn wird die Problemstellung erläutert, indem sowohl das Ziel dieses Konzepts als auch die Gründe für eine Systemverbesserung bestimmt werden. Zur Zieldefinition gehört eine klare Zielsetzung des Projektes und eine Bestimmung der Projektdauer. Letztere wird bei der Erstellung des Pflichtenheftes in Abschnitt 7.1.3 bestimmt.

Dieses Konzept hat das Hauptziel, ein auf Open Source basierendes ERP-System zu suchen und auszuwählen. Es ist für kleine und mittelständische ERP-Software-Anbieter als Zielgruppe (in diesem Fall für die Dresden Informatik GmbH) bestimmt. Die Dresden Informatik GmbH entwickelt das System DiPPS seit mehr als zehn Jahren kontinuierlich weiter. Das ERP-System DiPPS ist veraltet, denn es basiert auf ziemlich alter Technologie, wie z. B. BDE. Deshalb wird ein Ersatz für DiPPS gesucht, so dass das neue ERP-System den Änderungen und Entwicklungen folgt und die aktuelle Technologie sowohl für die Benutzeroberfläche als auch für die Datenübertragung unterstützt. Damit kann die DI auf dem ERP-Markt bestehen bleiben. Außerdem erhofft man sich mit einem neuen ERP-System bessere Chancen bei der Neukundengewinnung.

### **7.1.2 Zustandsanalyse**

Diese Analyse ist vor der Suche und Auswahl eines ERP-Systems unbedingt erforderlich. Sie dient der Erkennung bestehender und fehlender Funktionalitäten des aktuellen Systems und basiert auf der in Kapitel 5 erarbeiteten Grundlagen und Anforderungen an ERP-Systeme.

Es wurde mit DiPPS gearbeitet, um seinen Zustand analysieren zu können. Dabei wurde jede Funktionalität einzeln getestet.

Folgende Punkte fassen die Ergebnisse des DiPPS- Zustandes zusammen:

**Mehrschichtensystem:** Die Systemarchitektur von DiPPS ist mehrschichtig.

**Modularer Aufbau:** Das System verfügt über die Module Stammdatenverwaltung, Arbeitsvorbereitung, Produktionsplanung, Fertigung, Fertigungskalkulation, Materialsbedarfsermittlung, Administration/Systempflege, Einkauf, Verkauf, Angebots- und Auftragskalkulation sowie Materialwirtschaft. Es fehlen einige Module, wie z.B. Workflow für die Steuerung und Überwachung von Geschäftsprozessen.

**Mandantenfähigkeit:** Es ermöglicht, dass verschiedene Mitarbeiter an verschiedenen Mandanten parallel arbeiten.

**Datenbankunabhängigkeit:** DiPPS ist datenbankunabhängig.

**Internetfähigkeit:** Das System ist nicht internetfähig, d.h. es ist nicht möglich, mit DiPPS online zu arbeiten.

**Releasefähigkeit:** Es können in DiPPS Standardänderungen vorgenommen werden, ohne dass kundenspezifische Anpassungen geändert werden. Folglich ist DiPPS releasefähig.

DiPPS ist anpassungsfähig, zuverlässig, einfach zu benutzen und zu verstehen, effizient, skalierbar, flexibel, änderbar, erweiterbar und anpassbar, offen (bietet Interface zur Integration anderer Standards), prüfbar, mehrsprachig, vertraulich, informationssicher und arbeitet korrekt. Es unterstützt die Datenintegrität und Verfügbarkeit. Es ist nicht portierbar.

### 7.1.3 Pflichtenheft

Um eine rationale Entscheidung über die Auswahl und den Einsatz von ERP-Systemen treffen zu können, ist es unbedingt notwendig, dass zunächst ein Pflichtenheft erarbeitet wird, um es dann als Grundlage für die Suche und Auswahl eines geeigneten Systems zu benutzen.

Das Pflichtenheft ist eine ausführliche Beschreibung der Leistungen, die von der Einführung eines neuen Systems erwartet werden. Aus der vorherigen Zustandsanalyse sollen hier Anforderungen abgeleitet werden.

Neben den funktionalen Anforderungen sind in dem Pflichtenheft auch Restriktionen, z.B. hinsichtlich der Hardware, der Datenbank oder des Betriebssystems, zu berücksichtigen und zu dokumentieren.

Es soll alle Anforderungen der Zustandanalyse aufführen, auch die, die zurzeit nicht zu erfüllen sind. So wird bei der Auswahl eines neuen ERP-Systems gewährleistet, dass dieses mindestens die gleichen Funktionalitäten und Module wie DiPPS haben wird.

Um die Software direkt zu benutzen, sollte sie in einer für jede leicht verständliche Form verfügbar sein, also einen Quellcode einer höheren Programmiersprache nutzen.

Die Lizenz des Systems soll das Kopieren, das Verbreiten und die Nutzung erlauben, um der Dresden Informatik GmbH unnötige Kosten zu ersparen.

Der Quellcode sollte frei zugänglich sein, damit beliebige Modifikationen der Software durchgeführt werden können. Einschränkungen sind hierbei weder in Hinblick auf bestimmte Nutzergruppen noch auf bestimmte Nutzungszwecke zulässig.

Das Client-/Server-Konzept soll vollständig über alle Module hinweg unterstützt werden. Das Konzept wird über eine 3-Ebenen-Architektur, welche eine Trennung des ERP-Systems in Benutzungsschicht (Benutzeroberfläche), Applikationsschicht (Programmierungsumgebung, Applikationskern) und Datenhaltungsschicht (Datenbank) vorsieht, realisiert

In das Pflichtenheft müssen auch Anforderungen an OS-ERP-Systeme einfließen. Dazu gehören der jeweilige Reifegrad der Systeme und die Position im Produktionsbereich. Es sollen keine Systeme Verwendung finden, die sich im Alpha-Stadium der Entwicklung befinden. Die Qualität hinsichtlich Dokumentation und Feedback sowie die Breite der Community und die Community selbst sind ebenso wichtige Anforderungen, die sich im Pflichtenheft befinden müssen.

Die nächste Tabelle stellt die Kriterien mit ihre Prioritäten im Pflichtenheft zusammengefasst dar:

<b><u>Anforderungen</u></b>
<b>Funktionale Anforderungen</b>
Mehrschichtensystem
Modularer Aufbau
Mandantenfähigkeit

Datenbankenunabhängigkeit
Internetfähigkeit
Releasefähigkeit
<b>Nicht funktionale Anforderungen</b>
Anpassungsfähigkeit
Zuverlässigkeit
Aussehen und Handhabung
Benutzbarkeit
Leistung und Effizienz
Betriebs- und Umgebungsbedingungen
Wartbarkeit, Änderbarkeit
Portierbarkeit und Übertragbarkeit
Sicherheitsanforderungen
Korrektheit
Flexibilität
Skalierbarkeit
Mehrsprachigkeit
Offenheit für Integration und Standards
<b>Open Source Anforderungen</b>
Lizenz soll das Kopieren, Nutzen und Verbreiten der Software erlauben
System in deutscher Sprache vorhanden
Stable: im Produktionsbereich (nicht in Alpha-Bereich der Entwicklung)
Aktives Projekt, aktive Community und Dokumentation
<b>Hardware Anforderungen</b>
Auf windowsplattformen lauffähig (Client- Server)
Auf Linuxplattformen lauffähig (Client- Server)

**Tabelle 7.2: Pflichtenheft, Teil 1**

Da fast alle gelisteten Kriterien sowohl für den Entwickler als auch für den Endanwender erforderlich sind, sollen diese Kriterien im zukünftigen System nach Möglichkeit vorhanden sein. Die rot markierten Kriterien zählen als Wunschkriterien, d.h. auch ohne sie ist die Nutzung und Weiterentwicklung des Systems möglich.

Die nächste Tabelle stellt detaillierte Funktionalitäten, die im zukünftigen System vorhanden sein sollen, dar. Diese Tabelle wurde aus der Dokumentation der Projekte von DI erarbeitet.

<b>Funktionalität</b>	<b>Beschreibung</b>
<b>Artikel</b>	Mind. zweistufige Klassifikation (Produktklassen, Produktgruppen)
	Art der Lagerverwaltung wählbar (Einlager, Mehrlager, Chargen, Seriennummern)
	Wareneingangsprüfung wählbar
	Überlieferungssatz wählbar
	Administrator kann Eigenschaften selbst definieren
	kunden-/lieferantenspezifische Artikelnummern und -beschreibungen
	Losgrößen für Beschaffung (Einkauf/Fertigung) wählbar
	Gruppenänderung
	mittlerer Einkaufspreis
	Zeichnungsverwaltung zum Artikel, Auftrag oder Kunden
<b>Lager</b>	variable 3-dim. Struktur
	Lagerrechte für Buchungsarten
	Lagerrechte für Nutzer
<b>Materialwirtschaft</b>	Inventur
	Umlagerung
	Zu-/Abgangsbuchungen immer gegen Lagerbestand
	Buchen in negativen Bestand
	Buchung/Bestandskorrekturen gegen Kostenstellen
	Chargenverwaltung mit MHD und Rückverfolgung
	Qualitätskontrolle, Prüfpläne
<b>Stücklisten</b>	Status und Änderungsindex
	Phantombaugruppen verwaltbar (werden bei Fertigungsplanung in ihre Bestandteile aufgelöst)
	Komponententausch
	Strukturansicht
<b>Arbeitspläne</b>	Status und Änderungsindex
	Alternivarbeitspläne, losgrößenabhängig
	Arbeitsfolgen mit Bezug auf Arbeitsgang (verbal), Arbeitsgruppe (Maschine – Kapazität, Kosten) und Lohngruppe (Mitarbeiter – Kosten)
	Kooperationsarbeitsfolgen
	Arbeitsgruppentausch
<b>Bedarfsplanung</b>	Umwandlung von Fertigungsvorschlägen
	<b>MRP II :</b>
	- Berücksichtigung von Bedarf aus Meldebestandsunterschreitung, Kundenaufträgen, Werkstattaufträgen
	- Berücksichtigung von geplanten Zugängen aus Bestellungen und
	- aus Stücklistenauflösung
- Umwandlung von Bestellvorschlägen	
<b>Fertigung/ Produktion</b>	Zuordnung zum Kundenauftrag möglich
	Behandlung von Erzeugnisstrukturen als Fertigungsstruktur
	Auftragseinplanung für Fertigerzeugnisse und Baugruppen
	Kopie von Stückliste und Arbeitsplan zum Auftrag speichern
	Terminierung über Arbeitsplan
	Terminierung mit endlichen/unendlichen Kapazitäten (Arbeitsgruppen, Mitarbeiter) über Kalender
	Ausschussplanung
	Fertigungsgrobplanung mit Materialbeschaffungskontrolle
	Arbeitspapiere erzeugen
	Rückmeldung Materialentnahmen
	Rückmeldung Arbeitszeiten
	Rückmeldung Ausschuss
Buchen Fertigungszugang	

	Kooperationsaufträge für Kooperationsarbeitsgänge
	Auftragsüberwachung
	Auftragstyp Service/Reparatur (keine planbare Arbeit, nur Ist-Buchungen)
<b>Geschäftspartner</b>	Klassifikation: Interessent, Kunde, Lieferant, Referent, Mitarbeiter
	Adressstamm
	Kontaktpersonenstamm
	<b>Kundendaten :</b>
	- Kreditlimit
	- Umsatzkategorien
	- Preise und Rabatte
	- Rahmenverträge
	<b>Lieferantendaten:</b>
	- Preise und Rabatte
	- Rahmenverträge
	<b>Referentendaten:</b>
	- Provision
<b>Beschaffung</b>	<b>Mitarbeiter:</b>
	- Schichtplan
	- Kommt/Geht-Erfassung
	Administrator kann Eigenschaften selbst definieren
	Serienbriefe
	Anfragen, Preisvergleich
	Kopie Anfrage zu Bestellung
	Kopie Kundenauftrag zu Bestellung
	<b>Bestellung:</b>
	- Position mit Liefersplittung
	- Nebenkosten
	- Abruf aus Rahmenvertrag
	<b>Wareneingang:</b>
wahlweise Bestandserhöhung oder sofortige Entnahme gegen Fertigung (Doppelbuchung)	
Rechnungsprüfung	
Überweisung	
Lieferantenbewertung	
Anbieterverzeichnis (wer kann was liefern)	
<b>Verkauf</b>	<b>Angebot:</b>
	- Position ohne Artikel
	- Mengenstaffelung
	- Alternativpositionen
	- Nebenkosten
	Kopie Angebot zu Kundenauftrag
	Kopie Bestellung zu Kundenauftrag
	<b>Kundenaufträge:</b>
	- Positionen mit Liefersplittung
	- Nebenkosten
	- Abruf aus Rahmenvertrag
	- Zahlplan
	<b>Buchen Warenausgang:</b>
	- Lieferscheinerstellung
	<b>Reklamationsbearbeitung:</b>
	- Eingang, Bearbeitung, Rücklieferung
	<b>Rechnungslegung:</b>
	- Auftragsrechnung (Kundenauftrag, Reklamation)
- Sammelrechnung (Kundenaufträge)	
- Vorkasserechnung (Kundenauftrag, Reklamation)	
- Kostenstellenrechnung	
- Fertigungsrechnung für Service/Reparatur	
- Zahlungsüberwachung	
<b>Kalkulation</b>	Zuschlagskalkulation

	Angebotskalkulation
	<b>Fertigungskalkulation:</b>
	- Vorkalkulation über Planmenge Fertigerzeugnis
	- mitlaufende Kalkulation über Planmenge eines jeden Auftrages
	- Nachkalkulation über Ist-Menge Fertigerzeugnis
	- Kostenarten: Materialkosten, Materialgemeinkosten, Lohnkosten, Lohngemeinkosten, Kooperationskosten, Fertigungsgemeinkosten
	- Führung der Kostenarten einzeln über Erzeugnisstruktur
	Kundenauftragkalkulation
	<b>Produktionskalkulation:</b>
	- Soll: Standardkostenermittlung über Stückliste und Arbeitsplan
	- Ist: Fertigungsnachkalkulation
<b>Dokumenten-Verwaltung</b>	Erstellung, Zuordnung von Dokumenten zu Stammdaten und Vorgängen: Dateien, E-Mails
	Dokumentensuche mit Referenzierung der ERP-Daten
<b>Projekt-Management</b>	Projekte/Budget/Ausgaben
	Mitarbeiterplanung, Planzeiten/Ist-Zeiten, Beteiligungen
	Zuordnung beliebiger Vorgänge (Angebot, Kundenauftrag, Bestellung, Fertigung,...)
<b>Schnittstellen</b>	Finanzbuchhaltung
	Lohnbuchhaltung
	Export (xml, cvs...)
<b>Auswertungen und Statistiken</b>	Darstellung: Tabellen, Pivots, Diagramme
<b>Workflow</b>	-
<b>Archivierung</b>	-
<b>Report-Generator</b>	-

Tabelle 7.3: Pflichtenheft, Teil 2

#### 7.1.4 Durchführung der Marktanalyse und Vorauswahl

In diesem Abschnitt werden Marktinformationen über OS-ERP-Systeme, unabhängig vom zukünftigen System und von den Anforderungen im Pflichtenheft, gesammelt und eine Vorauswahl der auf dem OS-ERP-Markt wichtigsten Systeme getroffen. Diese Systeme werden anschließend evaluiert und miteinander verglichen. Es wird insbesondere ein Überblick über ERP Open Source gegeben.

Die folgende Tabelle [Reto] stellt die möglichen Quellen für die Recherche nach OS-ERP-Systemen dar. Diese Quellen werden danach beurteilt, ob sie geeignet oder ungeeignet für die Produktsuche, Produkt-Vorselektion und Produktbeurteilung sind.

Informationsquellen	Produktsuche	Vorselektion	Produkt-beurteilung
Softwarekataloge (z.B. ISIS-Report)	●	⊙	○
Fachzeitschriften	⊙	⊙	⊙
Messen (z.B. CeBIT)	⊙	⊙	○
World Wide Web	●	●	⊙
Konferenzen/Seminare/Tagungen	○	⊙	⊙
Fachliteratur	○	⊙	●
Herstellerinformationen (Informationsmaterial)	○	⊙	⊙
Beratungsunternehmen	⊙	⊙	⊙
Marktforschungsinstitute (z.B. Gartner Group)	●	●	○
Universitäten (Fallstudien, empirische Studien)	⊙	●	⊙
Erfahrungsberichte von Anwendern (Usergroups)	–	○	●
Testinstallation	–	○	●

○ ungeeignet    ⊙ teilweise geeignet    ● gut geeignet

**Tabelle 7.4: Mögliche Informationsquellen zur Auswahl eines ERP-Systems**

Festzustellen ist, dass das Internet die geeignetste Quelle für die Suche und Vorauswahl einer Software ist, da es besondere Webseiten für Open-Source Produkte gibt. Eine weitere Recherchemöglichkeit bieten Marktforschungsinstitute wie z.B. „Gartner Group“.

Zur Durchführung der Suche nach diesen Produkten wird in dieser Arbeit auf die zwei bekanntesten Seiten <http://sourceforge.net> und <http://www.heise.de/software> Bezug genommen, weil sie mehr als 300 OS-ERP-Systeme enthalten.

Es werden bei der Marktanalyse folgende Kriterien berücksichtigt:

- Das System muss in deutscher Sprache vorhanden sein.
- Es muss Stable sein, d.h. es soll nicht im Alpha-Bereich der Entwicklung sein.

<http://sourceforge.net> beinhaltet 30 Systeme, <http://www.heise.de/software> noch 10 weitere, die den o. g. Kriterien entsprechen und in die engere Vorauswahl genommen wurden.

Anschließend wurden diese 40 Systeme nach Betriebssystem, Art der Anwendung (Desktop oder Web) und Bemerkung über die Community und/oder

die Aktivität des Projektes sortiert. Das Ergebnis wird in folgender Tabelle veranschaulicht:

ERP-System	Plattform	User Interface	Bemerkung
<b>Openbravo</b>	<b>BS unabhängig</b>	<b>Web basiert</b>	<b>Breite Community-aktives Projekt: letztes Release-Datum 22.02.2011</b>
Vtiger	Window/Linux/mac	Plug-in/ web basiert/ Desktop	Nur CRM, enthält nur zwei ERP-Module Inventur und Projekt
<b>ADempiere</b>	<b>BS unabhängig</b>	<b>Desktop/ web basiert</b>	<b>Große Community/aktives Projekt: letztes Release-Datum 16.01.2011</b>
Dolibarr	BS unabhängig	Web basiert	-
Compiere	BS unabhängig	Gnome, KDE, Web-based, Win32 (MS Windows)	Module gegen entgelt Inaktiv: Release Datum: 2010-06-01
webERP	BS unabhängig	Web basiert	DBMS beschränkt sich auf MySQL
EBI Neutrino R1	BS unabhängig	Desktop/ web basiert	Mehr CRM als ERP orientiert
jAllinOne	BS unabhängig	Desktop/ web basiert	Mehr CRM als ERP/ kleine Community
<b>Opentaps</b>	<b>BS unabhängig</b>	<b>Desktop/ web basiert</b>	<b>Unterstützt verschiedene DBMS Release Datum: 23.02.2011 Dokumentation gegen Geld</b>
FacturaLUX	BS unabhängig	Desktop	Ist auf Finanzen beschränkt
JFire	BS unabhängig	Desktop/ web basiert	Die Module sind nicht oder teilweise nutzbar
Ext Rails	-	Web basiert	Kein Download verfügbar
TUTOS	-	-	Nur CRM
EdgeERP	BS unabhängig	Web basiert	Inaktives Projekt Letztes Release-Datum 13.08.2007
MAXdev	-	Web basiert	CMS
Zelico	Linux/ VMware	Desktop/web basiert	Läuft nicht auf Windows
Tryton	-	-	Quellcode in Download nicht verfügbar

Ifriqiya	BS unabhängig	Desktop	Inaktives Projekt. Letztes Release-Datum 10.04.2007
PhPepperShop	-	Web basiert	Für E-Commerce und Shopping
Linux Kontor	-	Web basiert/ X Window system(X11)	Inaktives Projekt. Letztes Release-Datum: 09.01.2003
Nseererp	Windows	-	Seit der Registrierung am 17.07.2007 kein Release
JasperSoft	BS unabhängig	Desktop	Download nur für Linux verfügbar
Tine 2.0	BS unabhängig	Web basiert	Download nicht verfügbar
SQL-Ledger	BS unabhängig	Web basiert	Seit der Registrierung am 04.04.2000 kein Release verfügbar+ Download nicht verfügbar
epesiBIM	BS unabhängig	Web basiert	CRM , wenig ERP Module
IntarS 5.1	-	Web basiert	branchenspezifische Repositories gegen Entgelt
JTL Mage Connect	-	-	Framework für Integration zwischen ERP und Online Shop Magneto
Bearstep	-	-	Download nur für Linux verfügbar
pERP	BS unabhängig	Web basiert	Enthält wenig Module
Jerpa	BS unabhängig	Web basiert	Inaktives Projekt: 31.01.2002
Erp4linux	BS unabhängig	-	- Download nicht mehr verfügbar - Alpha-Produktion
Small-erp	-	-	Inaktives Projekt- Download nicht verfügbar
Nmwerp	-	.Net/Mono	Inaktives Projekt: 06.07.2006
OpenEnterprise	-	-	Inaktives Projekt- Download nicht verfügbar
Open shoe managment system	-	-	Inaktives Projekt- Download nicht verfügbar
DevEnhancer	-	-	Nur Framework für Entwickler von SAP Business One ERP

			CRM Solution
Soul	-	Web basiert	Inaktives Projekt: letztes Release- Datum 14.04.2003
SAO2000	-	Desktop	Download nicht verfügbar
Lx-Office	Nur auf Linux lauffähig	Web basiert	Module, Dokumentation gegen Entgelt
kmuERP OSS	Plattformunabhän- gig	-	Download nur unter Linux verfügbar

**Tabelle 7.5: Selektierte OS-ERP-Systeme auf dem Markt**

Anhand dieser Tabelle müssen 37 Systeme aus unterschiedlichen Gründen, die in der Spalte „Bemerkung“ aufgelistet sind, aussortiert werden: Einige Systeme aus der Tabelle unterstützen nur eCommerce-Anwendungen, Katalogaustausch, Marketing, CMS oder CRM, z.B. das System vtiger CRM oder nature CRM. Einige sind nicht mehr zum Download verfügbar, z.B. ZaraStar ERP. Einige Projekte sind nie benutzt worden, werden nur als Framework angeboten oder sind veraltet, d. h. inaktiv. Andere Systeme müssten durch zu viele Implementierungen weiterer Funktionalitäten ergänzt werden oder haben eine zu geringe Benutzer- und Entwickleranzahl (kleine Community).

Nur die drei bereits in der Tabelle hervorgehobenen OS-ERP-Systeme Openbravo, ADempiere und Opentaps sind als potentielle Software geblieben. Sie laufen auf verschiedenen Betriebssystemen und sind außerdem internetfähig. Vor allem aber verfügen sie über eine deutsche Übersetzung und zählen zu aktiven Projekten, die sich nicht mehr im Alpha-Bereich ihrer Entwicklung befinden.

Deshalb werden diese drei Systeme in der nächsten Etappe miteinander verglichen, um die Auswahl auf zwei oder ein System zum letztendlichen Vergleich mit DiPPS beschränken. Der Vergleich mit DiPPS wird zeigen, inwiefern sich die Systemauswahl an die DiPPS-Funktionalitäten annähert oder von ihnen abweicht.

### **7.1.5 Evaluierung und vergleichende Gegenüberstellung**

Es wurden in der vorigen Etappe drei OS-ERP-Systeme aus 40 möglichen ausgewählt: Openbravo, ADempiere und Opentaps.. Der Funktionsumfang dieser drei Systeme ist in etwa gleich.

Zunächst Sie sollen im Folgenden miteinander nach allgemeinen Funktionalitäten verglichen werden. Der Vergleich setzt sich aus zwei Teilen zusammen: für die Systeme Adempiere und Openbravo wurde ein Service Provider für Compiere, Openbravo und Adempiere [evalGd] verwendet. Der andere Teil des Vergleiches gründet sich auf der Benutzung der Online Demo-Versionen von Opentaps, Openbravo und Adempiere. Zur Veranschaulichung der Untersuchungsergebnisse dient die folgende Tabelle:

<b>Features</b>	<b>Openbravo</b>	<b>Adempiere</b>	<b>opentaps</b>
<b>Rechnung</b>	j	j	j
<b>Auftragsverwaltung</b>	j	j	j
<b>Beschaffung (Material)</b>	j	j	j
<b>Inventur</b>	j	j	j
<b>(Personen/Firmen) Verwaltung</b>	j	j	j/n
CRM	j	j	j
Fertigung	j	j	n
<b>Verkauf- und Marketingverwaltung</b>	j	j	j
<b>Projektverwaltung</b>	j	j	n
MIS	j	j	j
<b>Workflow</b>	j	j	j
Datenbank Oracle XE/10g/11g	j	j	j
Datenbank PostgreSQL	j	j	j
Application Dictionary	j	j	n
Java Hot-swap Debug	n	j	n
Open Source und kostenfreie: Server-Seite	j	j	j
Open Source und kostenfreie Java-Client	n	j	n
Open Source and Free: Web UI Client	j	j	j
Lightweight Scripting Support	n	j	n

First Public release	April 2006	Oktober 2006	August 2005
Controlled and Driven by	Openbravo, S.L.	ADempiere Community	Opentaps OFBiz

**Tabelle 7.6: Vergleich der drei ausgewählten Systeme**

In der Evaluation haben die drei OS-ERP-Systeme einen ausgereiften Eindruck hinterlassen. Einige Schwächen sind jedoch vorhanden. So fehlt allen drei Systemen ein Modul für die Dokumentenverwaltung.

Da wichtige Funktionalitäten, wie Fertigung und Projektverwaltung bei Opentaps fehlen, reicht sein Funktionsumfang noch nicht vollständig an kommerzielle ERP-Systeme heran. Außerdem hat Opentaps [Otaps] im Gegensatz zu Openbravo [Obravo] und ADempiere [ADem] keine deutsche Community. Es existiert nur eine (englische) Dokumentation des OFBiz Frameworks, diese ist jedoch für Endanwender nicht problemlos nutzbar. Ferner müssen sich die Anwender aus insgesamt 244 Dokumenten die passende Anleitung für Nutzung und Entwicklung heraussuchen. Daher entspricht dieses System nicht genügend den o. g. Anforderungen und wird an dieser Stelle aus der engeren Auswahl entfernt.

Es bleiben ADempiere und Openbravo als mögliche OS-ERP-Systeme. Beide wurden aus einer früheren Compiere-Version [Cpiere] entwickelt, die heute nicht über Open-Source-Module verfügt, nur gegen Bezahlung verwendbar ist und deren Datenbank nicht frei wählbar ist. Dennoch legt diese Tatsache in Zusammenhang mit dem Ergebnis der Vorauswahl die Vermutung nahe, dass beide Systeme über eine ähnliche Grundstruktur verfügen, die erfolgsversprechend ist.

Deshalb kann der Einsatz von Adempiere und Openbravo bereits heute eine interessante und effiziente Alternative zu kommerziellen Systemen für kleinere und mittlere Unternehmen darstellen. Zudem unterstützen sie die gleichen Module wie Proprietäre Systeme und basieren außerdem auf neuen Technologien.

### **7.1.6 Vergleich mit Pflichtenheft und endgültige Auswahl**

In der vorigen Phase wurde Opentaps wegen fehlender Funktionalitäten und Module ausgeschlossen. Es werden nun die zwei verbliebenen Systeme

Openbravo und Adempiere installiert und den detaillierten Funktionalitäten des Pflichtenhefts gegenüber gestellt. Die Tabelle 9.9 im Anhang A 06 beinhaltet 115 Funktionalitäten, die im Pflichtenheft aufgelistet sind.

Die Ergebnisse dieses Vergleiches werden in Form einer Statistik in folgender Tabelle dargestellt.

<b>System</b>	<b>Unterstützte Funktionalität</b>	<b>Nicht unterstützte Funktionalität</b>
<b>Adempiere</b>	83/115 - 72%	32/115 - 28%
<b>Openbravo</b>	69/115 - 60%	46/115 - 40%

**Tabelle 7.7: Ergebnis des Vergleichs**

ADempiere deckt 83 Funktionen des Funktionsumfangs im Pflichtenheft, Openbravo dagegen nur 69 der 115 Funktionalitäten. Das entspricht einem Prozentsatz von 72%, mit dem Adempiere die aufgeführten Funktionalitäten unterstützt. Dagegen schneidet Openbravo mit nur 60% deutlich schlechter ab.

Die untersuchten OS-ERP-Systeme ADempiere und Openbravo bieten derzeit ausreichende Unterstützung für betriebswirtschaftliche Funktionen an, um für kleine und mittlere Unternehmen eine Alternative zu marktgängigen Produkten darzustellen. Da beide, wie schon erwähnt, von dem System Compiere abstammen und das Tool Application Dictionary<sup>4</sup> übernommen haben, besitzen sie neben einer ähnlichen Systemarchitektur wie Compiere auch eine hohe Flexibilität für weitere Anpassungen. ADempiere enthält allerdings mehr Funktionalitäten als Openbravo und unterstützt nicht nur Web-Client wie Openbravo, sondern auch die auf Java Swing basierte Desktop-Anwendung.

Damit fällt die Wahl eines OS-ERP-Systems auf ADempiere. Es ist das einzige System, das alle Anforderungen erfüllt hat und einen großen Teil der von DiPPS unterstützten Funktionalitäten abdeckt. Außerdem verfügt es über viele Anpassungsmöglichkeiten, die im nächsten Kapitel analysiert und diskutiert werden.

---

<sup>4</sup> vgl. Kapitel 7.1.7

### 7.1.7 Adempiere

[ProjAd] Das ADempiere-Projekt wurde im September 2006 nach der Abspaltung von Compiere Inc. erstellt. Ein Teil der Entwicklergemeinde von Compiere war nicht mit der Geschäftsführung einverstanden, weil sie mehr Wert auf die kommerzielle Seite der Software gelegt hat. Sie wollte Compiere nicht mehr als Open Source anbieten.

Der Name des Projekts leitet sich von dem italienischen Wort *adempiere*, welches „erfüllen“ bedeutet, ab. Alle Community-Mitglieder nehmen an der Entwicklung und Erweiterung dieses Projekts teil.

Die Community setzt das Prinzip Bazaar [BazRay] ein, um ihre Ziele zu verwirklichen. Dieses Prinzip besagt, dass der Quellcode in jedem Entwicklungsstadium über das Internet einsehbar sein und die Software von verschiedenen Entwicklern weiterentwickelt werden soll. So werden auch große Projekte, wie Linux-Kernel koordiniert.

Das Projekt hat einen Rat, der sich um die Organisation, die Ziele (Entwicklung einer hochwertigen Applikation - maximale Entwickler-Teilnahme) und Abstimmung neuer Funktionalitäten kümmert. Dieser Rat spielt keine Führungsrolle.

Die nächste Tabelle stellt technische Daten von ADempiere dar [ADWiki]. Die Daten in der Tabelle wurden nach der offiziellen Webseite von ADempiere sortiert.

<b>ADempiere: Technische Daten</b>	
Java Code	Ca. 1.5 Mio. Zeilen
Datenbank Tabellen	Ca. 720 Tabellen
Server und Client	Windows, Linux und Mac
Mehrere Server in verteilter Umgebung	Mit Synchronisierung möglich
Prozessorientiertes Konzept	Ablaufsteuerung, Alarmfunktion usw.
Rollenbasiertes User- und Zugriffsrechtekonzept	Flexible Zugriffskontrolle und Benutzerverwaltung
6 Multis	Multi-Mandantenfähigkeit Multi-Organisationsfähigkeit Multi-Sprachfähigkeit Multi-Währungsfähigkeit Multi-Lagerfähigkeit Multi-Produktionstandortfähigkeit

Flexibles Berichtswesen	Interner Berichtsgenerator Externe Berichte über Jasper Report
-------------------------	---

**Tabelle 7.8: Technische Daten von ADempiere**

Bei der Abspaltung von Compiere wurde das Data Dictionary [DDict] (oder Application Dictionary) übernommen. Es ist ein Katalog von Metadaten, der in einer Datenbank die Definition und Darstellungsregeln von Datenelementen enthält.

Diese Architektur erlaubt die Erweiterung und Änderung des Datenkatalog-Konzepts aus der Datenbank in die Applikation hinein. Das wiederum sorgt für die Verwaltung von Applikationsobjekten und ihrer Gültigkeitsregeln, ebenso wie des Bildschirmlayouts und der Darstellungslogik aus der Applikation selbst heraus.

ADempiere ist nicht nur ein aktives Projekt, das eine starke Entwicklergemeinschaft hat, sondern auch ein ERP-System, das sich im produktiven Einsatz befindet.

Es gibt verschiedene IT-Firmen in Deutschland, die dieses System benutzen, an kleine und mittelständige Unternehmen verkaufen und diese entsprechend betreuen. Dazu gehören z. B. die Firma Catura AG in Passau, die Schaeffer AG in Berlin, die metas GmbH und evenos IT Solution in Bonn, die integration GmbH in Böblingen sowie die Objectcode GmbH in Lünen. [Refer]

### **7.1.8 Zusammenfassung**

In diesem Kapitel wurde ein Konzept für das Auswahlverfahren eines OS-ERP-Systems durchgeführt und brachte das Ergebnis, dass Adempiere ein passender Ersatz für DiPPS sein kann. Im nächsten Schritt wird ein Konzept für die Einführung einer kundenspezifischen Anpassung in ADempiere entwickelt und umgesetzt.

## 7.2 Konzept einer kundenspezifische Anpassung in ADempiere

### 7.2.1 Allgemeine Anpassungsarten

Im Allgemein gibt es vier Möglichkeiten zur Anpassung eines ERP-Systems an die kundenspezifischen Bedürfnisse (gestützt auf [HesGör]). Sie sind: [vgl. [HesGör], S.223ff]

- Customizing
- Personalisierung
- Eigenentwicklung
- Modifikation

**Customizing:** Die Aufbau- und Ablauforganisation eines Unternehmens muss im System abgebildet sein, um das ERP-System produktiv einsetzen zu können. Dies wird durch das so genannte Customizing möglich. Dazu gehören alle Einstellungen und Anpassungen am System, die sich allein durch die Einstellung von Parametern, d. h. ohne Programmierung, vornehmen lassen. [vgl. [HesGör], S.224]

**Personalisierung:** Mit Hilfe der Möglichkeiten der Personalisierung kann die Bedienung des ERP-Systems optimal an die Anforderungen jedes Nutzers angepasst werden. Dazu zählen z. B. arbeitsplatzspezifische Menüs und Shortcuts, die sich alle, wie beim Customizing, ohne Programmierung ausführen lassen. [vgl. [HesGör],S.233ff]

**Eigenentwicklung:** Durch integrierte Entwicklungsumgebungen und ihre Werkzeuge werden Eigenentwicklungen in Form von eigenen Programmen und Objekten unkompliziert möglich. Aufgrund ihrer Einbindung in das System stellen sie eine unkritische Anpassungsmöglichkeit dar. Wirtschaftliche Aspekte sollten allerdings bezüglich des Umfangs berücksichtigt werden. [vgl. [HesGör], S. 241ff.]

**Modifikation:** Anders als bei Eigenentwicklungen werden hierbei durch die integrierte Entwicklungsumgebung programmtechnische Veränderungen an Standardobjekten eines ERP-Systems vorgenommen. Deshalb sind

Modifikationen auch als äußerst kritisch zu bewerten, da sie mit einigen Risiken bzw. wirtschaftlichen Gefahren verbunden sind. Sie sollten deshalb möglichst nur in Ausnahmefällen umgesetzt werden. [vgl. [HesGör], S. 244ff.]

## 7.2.2 Anpassungen in ADempiere

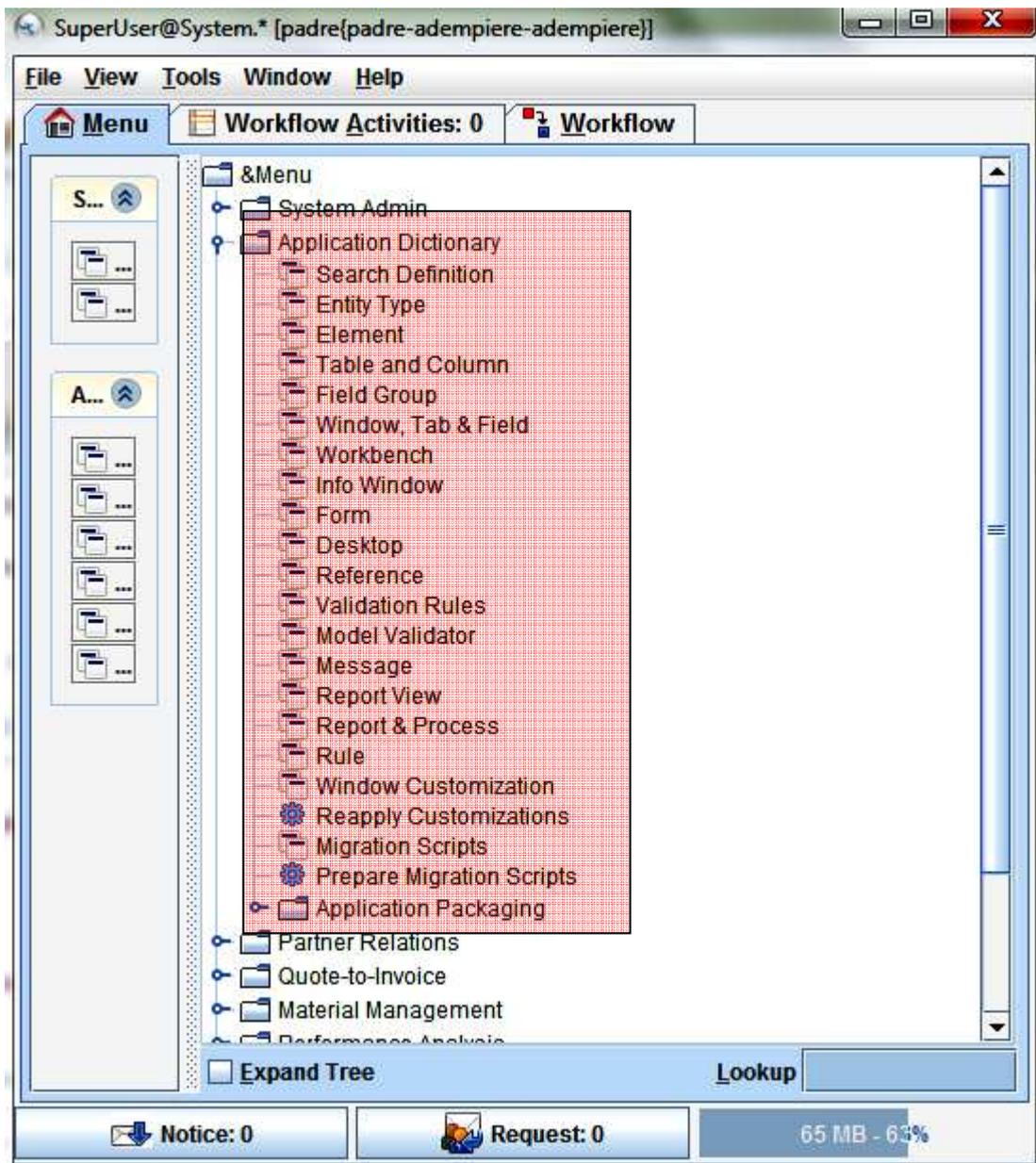
Es gibt grundsätzlich zwei Möglichkeiten der Realisierung einer Anpassung in ADempiere. Die erste kann mit weniger Programmierung umgesetzt werden, die zweite erfordert die Änderung des Quellcodes.

Für die erste Möglichkeit wird das ADempiere's Application Dictionary [AD] eingesetzt. Es ist ein integraler Bestandteil der Anwendung und enthält Informationen über die Daten, ihre Natur und wie diese Daten miteinander verbunden sind. Damit definiert es die Struktur der einzelnen Module und Fenster. Dieses Tool ermöglicht außerdem die Verwaltung von Entitäten der Anwendung und deren Überprüfung (Validation Rules), sowie die Bildschirmlayouts und die Displaylogik innerhalb der Anwendung.

Application Dictionary ist durch den Benutzer erweiterbar, d.h. es ist dem berechtigten Benutzer erlaubt, neue Tabellen, Fenster, Datenfelder hinzuzufügen. Die Application Dictionary löst die Model-View-Logic-Architektur des modernen Software-Designs ab.

Die gesamte Anwendung kann durch dieses Tool erweitert, konfiguriert und manipuliert werden, ohne dass sie neu kompiliert oder neu ausgeführt werden muss. Das Schreiben oder Ändern des Codes ist somit drastisch reduziert worden.

Application Dictionary ist folglich ein leistungsstarkes und flexibles Framework innerhalb von ADempiere.



**Abbildung 7.2: Elemente der Application Dictionary**

Application Dictionary ADempiere verfügt über „Table and Column“, bezieht sich damit direkt auf die ADempiere-Datenbank und erlaubt das Einfügen oder Ändern:

- Der Datenfelder, Tabellen und Fenster ohne Business-Logik
- Der Gültigkeitsregeln für Datenfelder
- Der Standardwerte für Datenfelder
- Der Logik für Felder (z.B. nur zum Lesen oder zur Sichtbarkeit)
- Des Layouts (die Reihenfolge der Datenfelder) der Tabelle

„Table and Column“ gestattet außerdem das Übersetzen oder Umbenennen der GUI- Elemente.

Die folgende Abbildung stellt die Architektur von ADempiere mit Application Dictionary vor.

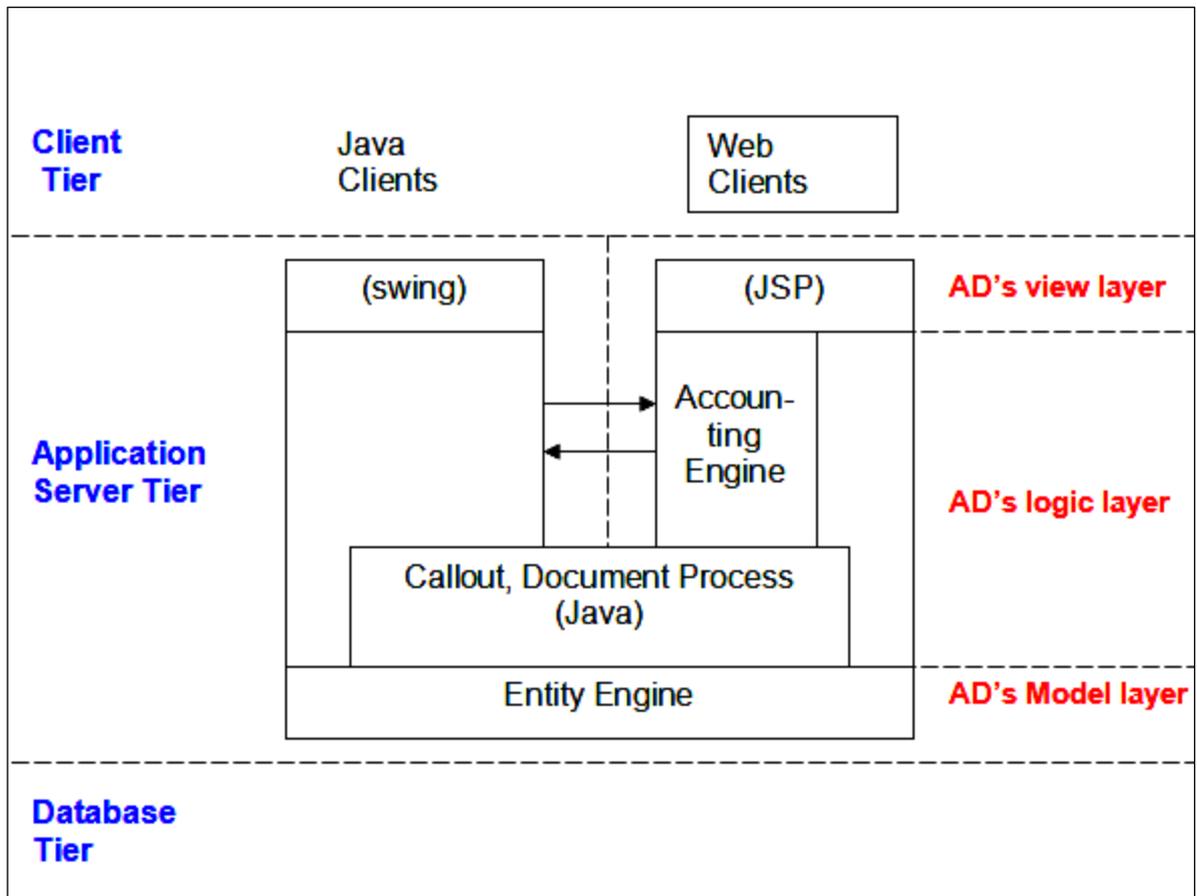


Abbildung 7.3: Architektur von ADempiere und die Schichten der Application Dictionary

Wie der Grafik zu entnehmen ist, befindet sich das Application Dictionary im Application Server von ADempiere, aufgeteilt in drei Schichten: Model-Schicht, View-Schicht und Logik-Schicht. Diese sollen kurz vorgestellt werden.

### 1. AD Model-Schicht

Die Model-Schicht kümmert sich um alle Datensätze der Anwendung wie die Datenbank-Tabellen, die Datenfeld-Struktur und deren Datentypen, Referenz- und Validierungsregeln. Tabellen, Datenfelder und ihre Eigenschaften sind im AD Table & Column Fenster definiert. Die Datenfelder können durch Validierungsregeln im AD Validation Fenster (siehe Anhang A 07) definiert werden.

Jede Validierung legt JavaScript, Java Code oder SQL Script in der Anwendung zur Filterung des Wertes der Referenzen an.

Via Callouts kann ein Datenfeld andere Datenfelder in einem Fenster oder in der grundlegenden Datenbank aktualisieren. Der AD Engine bereitet Tabellen und Datenfelder-Struktur für die View-Schicht vor (z.B. Anzeige über die Standard-Benutzeroberfläche oder Berichtformate).

## **2. AD View-Schicht**

Die View-Schicht ist verantwortlich für die Darstellung sowohl im Java Swing Client als auch im HTML Client. Dies läuft durch das Application Dictionary ohne mühevollen Programmierung ab. Diese Schicht stellt die in der Model-Schicht angelegten Tabellen in der Benutzeroberfläche innerhalb von Fenstern dar. Alle CRUD-Funktionen werden in dieser Schicht für die Verwendung des erstellten Fensters generiert. Letztes erlaubt die Speicherung, das Löschen und Aktualisieren der Datensätze. Die View-Schicht ermöglicht leistungsstarke Suchfunktionen und wird durch das AD Menü, das AD Fenster, sowie den AD Prozess und den Bericht gesteuert.

## **3. AD Logik-Schicht**

In der Logik-Schicht wird die gesamte Business-Logik definiert und alle Zugriffsfunktionen, wie Benutzer-Login und Zugriffsrolle, verwaltet. Diese Schicht erlaubt eine schnelle Anpassung der Anwendung auf geschäftliche Anforderungen je nach aktueller Entwicklung.

Das ADempiere-Projekt hat den Logik-Teil bei der Einführung:

- der Trennung von erweiterter Logik und Core-Logik:
- von JSR223 Scripting in Rule-Engine (Anhang A 08)
- und des Callout

verfeinert. Die Logik-Trennung erfolgt durch strikte Anwendung des Model-Validators, um dynamische Aktionen abzudecken.

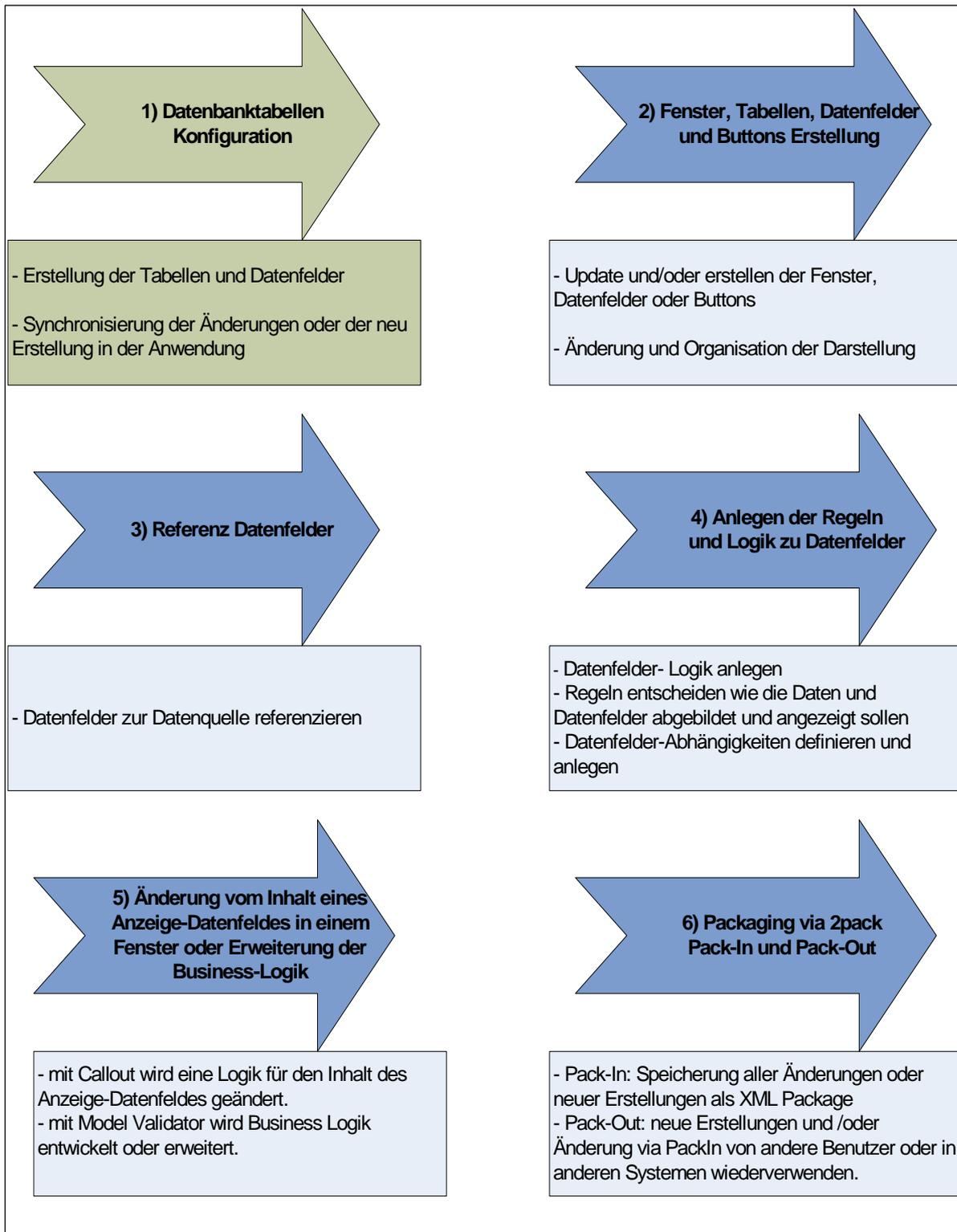
ADempiere basiert auf modellgetriebener Architektur (MDA). Dies führt zu einfacherer Anpassbarkeit, schneller Einsetzbarkeit und geringeren Betriebskosten. MDA vereinfacht die Anpassung der Anwendung an die Bedürfnisse der Kunden.

Somit können einfache Anpassungen in kürzester Zeit vorgenommen werden. Ohne aufwendige Programmierung oder Erstellung von SQL-Datenbankabfragen ist es beispielsweise möglich:

- Eine Datenbanktabelle um eine Spalte zu erweitern
- Der Spalte bestimmte Restriktionen zuzuweisen
- Das Design einer Eingabemaske entsprechend anzupassen
- Daten über das modifizierte Fenster in die neue Spalte einzugeben
- Einen Bericht, der die neuen Daten verwendet, zu erstellen.

Im Folgenden wird ein allgemeines Konzept zur kundenspezifischen Anpassung (Neu-Erstellung eines Fensters mit einem oder mehreren Datenfeldern in ADempiere) durch das Application Dictionary ADempiere vorgestellt.

Dieses Konzept ist auch für die Änderung eines existierenden Fensters anwendbar.



**Abbildung 7.4: Anpassungskonzept in Application Dictionary**

Dieses Konzept besteht aus sechs Phasen:

1. Die erste Phase kümmert sich um die neue Erstellung der Datenbanktabellen in der ADempiere-Datenbank. Es können auch Views in der Datenbank erstellt werden.

Es gibt zwei Möglichkeiten, neue Datenbanktabellen in ADempiere zu erstellen: entweder direkt in der ADempiere-Datenbank oder in der ADempiere-Anwendung durch Application Dictionary. Die erste Möglichkeit benötigt dann die Aktualisierung der Tabellen in Application Dictionary, damit von der Anwendung auf sie zugegriffen werden kann. Die zweite Möglichkeit fordert die Synchronisierung der neu erstellten Tabellen, damit sie von der ADempiere-Datenbank aus zugreifbar wird. Die Tabellen sollen immer folgende Datenfelder enthalten, damit sie von der Application Dictionary verstanden und anerkannt und in AD\_Window (beinhaltet Informationen über alle ADempiere-Fenster) sowie in AD\_PrintFormat (für Druckformate) verwendet werden können:

- **ad\_client\_id**: verweist auf die Tabelle **ad\_client**, die Informationen des Mandanten enthält.
- **ad\_org\_id**: bezieht sich auf die die Tabelle **ad\_org**, die die Daten der Organisation beinhaltet.
- **created**: Datum
- **createdby**: Name
- **updated**: Datum
- **updatedby**: Name
- **isactive**: das ist eine Checkbox, die bestimmt, ob das Datenfeld aktiv oder inaktiv sein soll.

**Createdby** und **updatedby** nehmen generell Bezug auf die Benutzer, die in der ADempiere-Anwendung sind, d.h sie verweisen die Tabelle **ad\_user**.

Für die Änderung einer Applikation in ADempiere muss allerdings mit der zweiten Etappe begonnen werden.

2. In der zweiten Phase werden Fenster, Datenfelder oder Buttons aktualisiert bzw. neu erstellt.

Fenster werden hier neu erstellt oder geändert, indem die neu erstellten Datenbanktabellen hier angenommen und zu einem Fenster zugeordnet werden.

Es können Namen zur Darstellung im Fenster eingesetzt werden. Datenfelder sollen Parameter haben z.B. Character, Numeric. Normalerweise verfügen die Datenfelder schon bei der Erstellung der Datenbanktabellen über Parameter. Die ADempiere Application Dictionary erlaubt die Änderung diesen Parameter. Die Datenfelder sollten sortiert und im Fenster nach Wunsch oder Bedarf platziert

werden. Zuletzt muss das neue Fenster im Menü aktiviert werden, damit es zugänglich wird.

Es können zwei Tabellen auf einem Tab im Fenster erstellt werden. Die Darstellung der erstellten Fenster ist sowohl als Listenansicht als auch als Detailansicht möglich. Beide verfügen über eine automatische Druckfunktion, die sie sich anpassen lässt. Application Dictionary ermöglicht mittels der Funktion „Lookup Record“ die Suche nach Daten in einem Fenster. Sie muss nur aktiviert werden.

3. Die dritte Phase erlaubt den Verweis der Datenfelder- und Anzeigetypen auf die Datenquelle. Die Abbildungen im Anhang A 09 veranschaulichen dies. So können die Datenfelder eines Fensters miteinander und mit den Datenfeldern eines anderen Fensters in der Anwendung verbunden werden.

4. Die vierte Phase beschäftigt sich mit der Definition und dem Anlegen von Regeln und Logik für die Datenfelder. Die Logik bestimmt, wie die Datenfelder dargestellt und dynamisch angezeigt oder versteckt werden.

5. Die fünfte Phase ist mit dynamischen Aktionen verbunden. In dieser Phase werden die Änderungen des Inhaltes der Darstellung eines Datenfeldes bearbeitet und /oder die Business-Logik definiert.

6. In der letzten Phase werden die realisierten Anpassungen gespeichert, damit man sie für weitere Verwendung durch eine andere ADempiere-Instanz verwenden kann. Dafür wird das Tool Application Packaging, das ein Bestandteil von Application Dictionary ist, verwendet. Die erstellten Tabellen, Spalten und Datenfelder in der Datenbank sind im Pack-out in einer einzigen XML-File vorhanden und werden dann von Pack-In verarbeitet.

Das genierte Package ist in zwei Formaten gespeichert: .zip für Windows-Plattformen und tar.gz für Linux. Diese beiden Dateien beinhalten den kompletten Quellcode, um die Anpassungen für andere ADempiere-Instanzen zu installieren. Deshalb muss nur diese Datei auf alle Zielbenutzer verteilt werden, um die vorgenommenen Änderungen für alle zugänglich zu machen.

### **7.3 Prototypische Implementierung**

In diesem Abschnitt wird das im vorigen Paragraph vorgestellte Konzept zur kundenspezifischen Anpassung des OS-ERP-Systems ADempiere mittels der Implementierung eines Prototyps validiert.

Es wird eine Anwendung für die Verwaltung des Firmenworkshops in ADempiere realisiert werden. Diese Anwendung erlaubt die Versammlung von verschiedenen Firmen oder Abteilungen, die über bestimmte Themen und Probleme diskutieren wollen. Es wird gezeigt, wie eine entsprechende Anpassung in ADempiere abläuft.

Dafür soll zunächst ein Datenmodell für diese Anwendung angelegt werden, dann folgt die Erstellung einer Datenbank, die mit der ADempiere-Datenbank verbunden wird.

#### **Semantisches Modell:**

Das Datenmodell soll einen Workshop für Firmen verwalten. Es sollen Datum, Beginn und Ende, Agenda, Titel und Vorsitzende des Workshops gespeichert werden. Jeder Workshop besteht aus einem oder mehreren verschiedenen Themen bzw. Teilnehmern. Jedes Thema ist mit Namen, Autor und Zustand gekennzeichnet.

Das entsprechende semantische Modell ist folgendes: Es werden die Objekte Workshop, Thema, Teilnehmer und Zustand erkannt. Dafür müssen nachfolgende Attribute in der Datenbank gespeichert sein:

- **F\_Workshop** (Titel, Datum, von ..., bis ..., Agenda, Vorsitzender)
- **F\_Workshop\_Thema** (Item, Thema, Autor)
- **F\_Workshop\_Teilnehmer** (Name, Firmenname)
- **F\_Workshop\_Zustand** (Zustand, Beschreibung)

Die folgende Abbildung fasst das Datenmodell für einen solchen Workshop zusammen:

es wird zwei Fenster mit einander verbunden aufgebaut. Das erste für die Organisation eines Workshops (Teilnehmer, Themen, Firmen) und das zweite wird den Zustand eines Workshops zeigen (offen, geklärt, gelöscht und in Warteschleife)

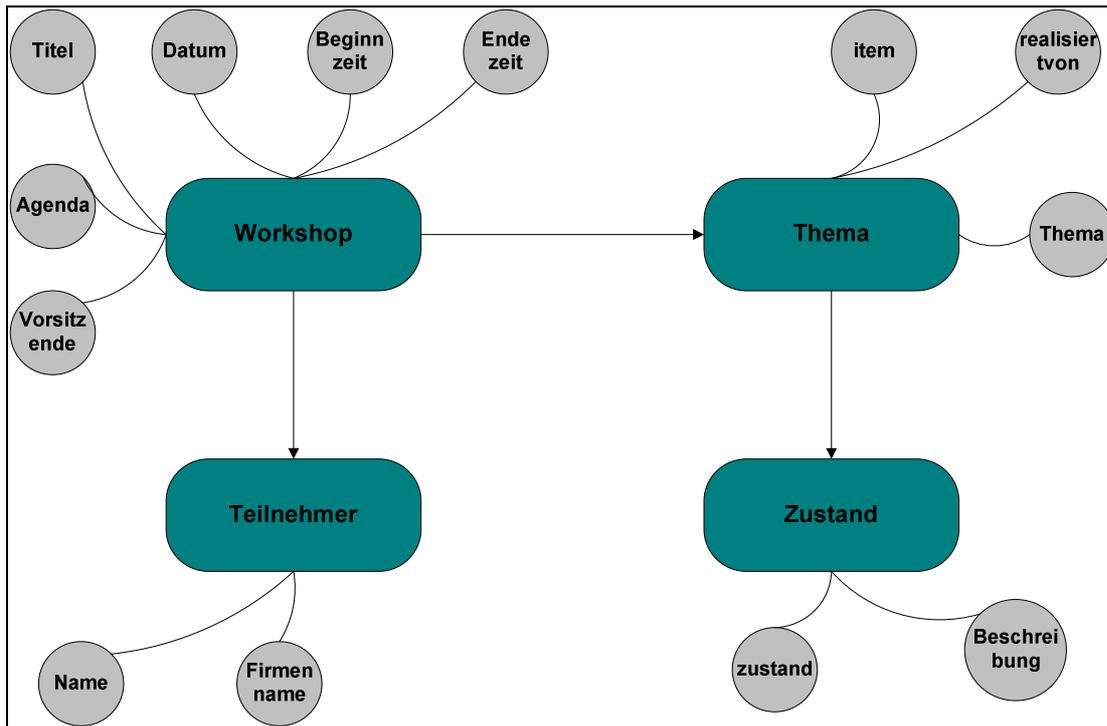


Abbildung 7.5: Datenmodell für die Anwendung "Workshop"

Es werden zwei miteinander verbundene Fenster aufgebaut. Das erste für die Organisation eines Workshops (Teilnehmer, Themen, Firmen) und das zweite für die Anzeige des Zustands eines Workshops (offen, geschlossen, in Warteschleife, u. ä.).

Anschließend werden die Datentypen den Attributen zugeordnet:

**F\_Workshop** {**Titel**: Character Varying (255), **Datum**: Date, **von**: Timestamp, **bis**: Timestamp, **Agenda**: Character Varying (4000), **Vorsitzende**: Character Varying}.

**F\_Workshop\_Thema** {**item**: Numeric (10, 0), **Thema**: Character varying (2000), **Autor**: Character Varying (80)}.

**F\_Workshop\_Teilnehmer** {**Name**: Character varying (60), **Firmenname**: character Varying (80)}.

**F\_Workshop\_Zustand** {**Zustand**: Character Varying (60), **Beschreibung**: Character varying (255)}.

### Logisches Modell:

Im dritten Schritt ergibt sich folgendes logisches Modell:

Bei der Erstellung der Datenbanktabellen sollen die Attribute AD\_Client\_ID, AD\_Org\_ID, isactive, created, createdby, updated und updatedby, die in allen Tabellen der ADempiere-Datenbank existieren, in die Tabellen des Workshop hinzugefügt werden.

Die AD\_Client\_ID bezieht sich auf die Tabelle Client (mandant), AD\_Org\_ID auf die Tabelle Organisation (Firma)

So entsteht die folgende Datenbank:

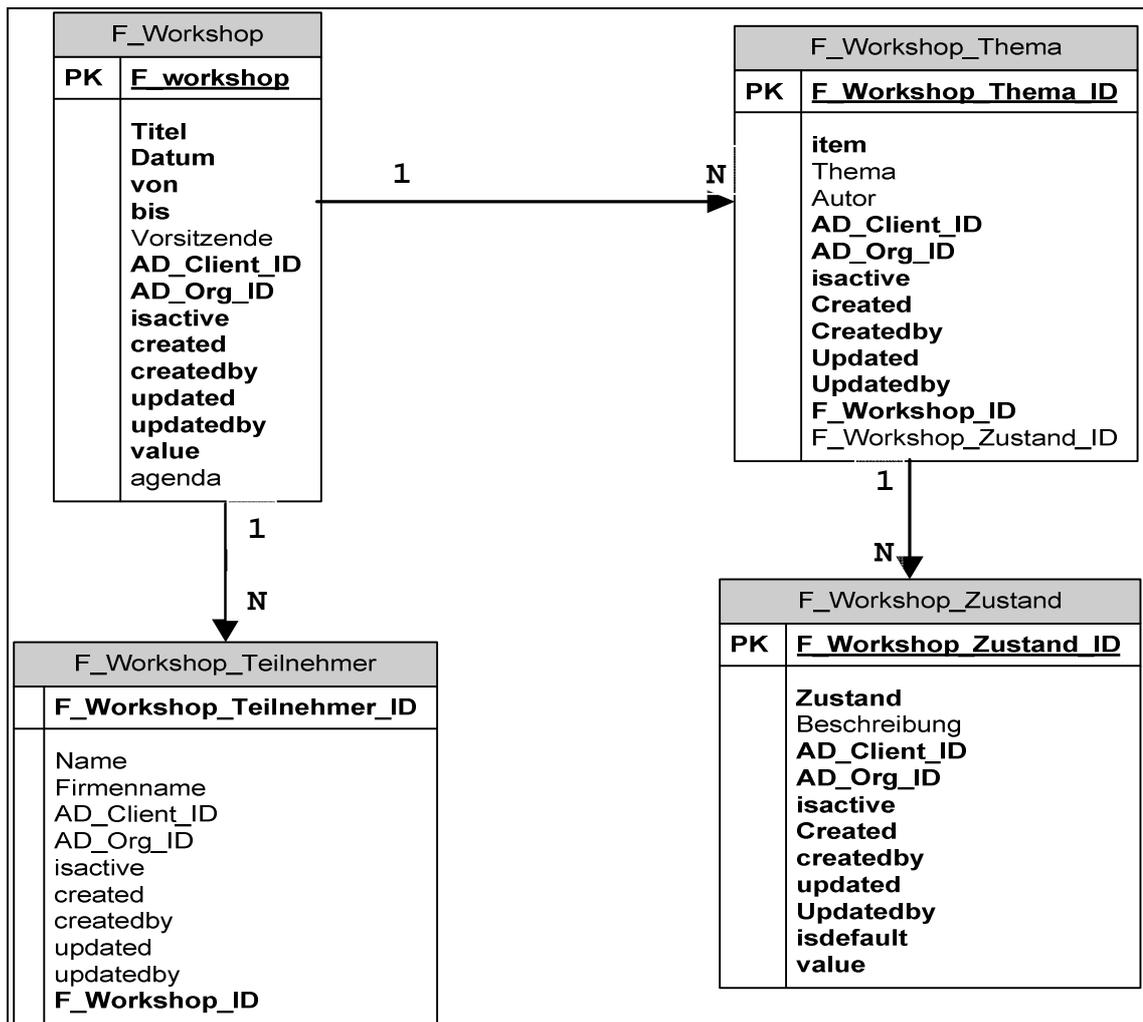


Abbildung 7.6: Datenbanktabellen für die Anwendung "Workshop"

Die vier erstellten Tabellen beinhalten folglich AD\_Client\_ID und AD\_Org\_ID, sodass sie automatisch auf Mandanten und Organisation (Firma) verweisen.

Nun werden die Schritte des in der Abbildung 7.4 erstellten Konzeptes durchgeführt.

1. Im ersten Schritt werden die Datenbanktabellen der Abbildung 7.6 in die ADempiere-Datenbank integriert. Dann werden diese Tabellen in Application Dictionary eingefügt.

Dazu muss man als Systemadministrator eingeloggt sein und das Application Dictionary aufrufen. In „Table and Column“ legt man einen neuen Datensatz mit dem gleichen Namen wie die Tabellen an. Zuletzt generiert man die Spalten, indem man den Button „Create columns from DB“ anklickt. (Siehe Abbildung 9.5 im Anhang 10) Dies wird für alle vier Tabellen durchgeführt.

2. Im zweiten Schritt werden die gewünschten Fenster angelegt, in diesem Fall „Workshop“ und „Workshopzustand“. In „Window, Tab & Field“ von AD werden die zwei Fenster erstellt, die die im ersten Schritt erzeugten Tabellen als Tab<sup>5</sup> enthalten (Siehe Abbildungen im Anhang A10).

sie zeigen die Erstellung des Fensters „Workshop“ mit den Tabs „Workshop“, „W Teilnehmer“ und „W Thema“, die die Datenbanktabellen „f-workshop“, „f\_workshopteilnehmer“ und „f\_workshophema“ präsentieren. Das Fenster „Workshop“ hat einen zusätzlichen Tab, nämlich „ausstehende Themen“, der aus dem Tab „W Thema“ kopiert wurde, d.h. er verweist auf die Datenbanktabelle „f\_workshophema“. Mit gleichen Schritten wird das Fenster „Workshopzustand“ erstellt.

3. Im dritten Schritt werden die Datenfelder der einzelnen Fenster verknüpft, d.h. die Datenfelder, die mit bestimmten ADempiere-Datenbanktabellen verbunden sein sollen, müssen hier festgelegt werden. Beispielsweise verweist im Fenster „Workshop“ das Datenfeld „Firma“ im Tab „W Teilnehmer“ auf die ADempiere-Datenbanktabelle „AD\_Org“ für die Organisation.

Es wird in diesem Schritt die Verknüpfung der beiden Fenster vorgenommen. Das Fenster „Workshopzustand“ wird im Tab „W Thema“ auf das Fenster „Workshop“ verwiesen. So kann man beim Erfassen eines Themas im Workshop seinen Zustand lesen oder eingeben. Dies wurde in AD unter Reference realisiert. Die Abbildungen in Anhang A11 stellen den Verweis des Workshopzustandes zum Workshop dar.

---

<sup>5</sup> Tab stellt eine Tabelle in einem Fenster dar.

4. im vierten Schritt werden die Regeln und Logik für die Datenfelder angelegt. Dabei muss zwischen zwei Logik-Arten unterschieden werden, nämlich die Default-Logik und die Anzeige-Logik. Erstere findet Verwendung, wenn man einen Standardwert auf einem Datenfeld zuweisen möchte, z.B. das Datum im Fenster „Workshop“ bekommt als Standardwert das heutige Datum, wenn man die Wert „@Date@“ im „Default Logic“ eingibt (Siehe Abbildung 9.16 am Anhang A12). Die zweite wird verwendet, um einige Datenfelder oder Tab ein- oder auszublenden, z.B. der Tab „ausstehende Themen“ wird im Fenster „Workshop“ eingeblendet, wenn der Datenfeld „Workshopzustand“ die Werte „offen“ oder „in Warteschleife“ zuweist. Er wird ausgeblendet, wenn eine der Werte „gelöscht“ oder „geklärt“ für den Workshopzustand ausgewählt ist.

Dies wird mit der Anzeige Logik **@f\_workshopzustand\_ID@!1000003 & @f\_workshopzustand\_ID@!1000004** realisiert. Dieser Ausdruck bedeutet, dass die Zustände in der Tabelle „f\_workshopzustand“ mit den ID- Nummern 1000003 und 1000004 in der Datenbank ausgeblendet werden. Diesen zwei ID-Nummer entsprechen den Zuständen „gelöscht“ und „geklärt“. (Siehe Abbildung 9.17 am Anhang A12)

Die Logik hat das folgende Format:

**Format:= {expression} [{logic}{expression}]**  
**Expression:= @ {context} @ {operand} {value} or**  
**@ {context} @ {operand} {value}**  
**Logic: = {!} | {&}**  
**Context: = any global or window context**  
**Value: = strings or numbers**  
**Logic operators: = AND or OR with the previous result from left to right**  
**Operand: = eq{=}, gt{>}, le{<}, not{~!}**

5. Im fünften Schritt wird die dynamische Logik definiert und entwickelt. In diesem Schritt kann die Business-Logik mit Callout [CaOut] oder mit Model Validator [ModVal] realisiert werden. Die Callout-Klasse (extend CalloutEngine) besitzt verschiedene Methoden, die aufgerufen werden, wenn ein Datenfeld in einem Fenster geändert wurde. Callout ist für die einfache Logik und Model Validator für die komplexe Logik anwendbar.

Das Datenfeld „item“ im Tab „W Thema“ muss man per Hand eingeben. Mit der Benutzung von Callout wird dieses Datenfeld bei Anlegen eines neuen Datensatzes im Tab „W Thema“ automatisch generiert und die Reihenfolge von „item“ beibehalten. Dies wird realisiert wenn man eine Java-Klasse namens

CalloutWorkshop implementiert. Diese Klasse erweitert den CalloutEngine und implementiert die Methode „getNextItem“, die die nächste item-Nummer bei Neuanlegen eines Datensatzes im Tab „W Thema“ automatisch generiert. Sie selektiert den maximalen Wert von item in der Datenbanktabelle und inkrementiert sie beim Anlegen eines neuen Themas. Nach der Implementierung der „CalloutWorkshop“-Klasse loggt man sich als Systemadministrator in ADempiere ein. In „Table and Column“ von AD öffnet man die Tabelle „f\_workshopthema“, dann das Datenfeld „f\_workshop\_ID“. Danach gibt man den Pfad vom implementierten Callout, **org.compiere.model.CalloutWorkshop.getNextItem** ein. (Siehe Abbildungen am Anhang A13)

Model Validatoren kommen erst zum Einsatz, wenn man bei der Anpassung komplexe Business-Logik implementieren wollen. Sie erlauben das Persistence Objekt für die Tabellen zu erzeugen. In dieser Anwendung wird keine Business-Logik entwickelt, aber sie wird dafür vorbereitet, in dem man die Modelle für die vier erstellen Tabellen generiert (Anhang A14). Dies läuft in der Entwicklungsumgebung Eclipse. Es wird eine Java Applikation namens „generate Model“ mit der Hauptklasse **org.adempiere.util.GenerateModel** im Adempiere Projekt angelegt. Die Ausführung dieser Applikation generiert die vier Klassen: X\_f\_workshop.java, X\_f\_workshopteilnehmer.java, X\_f\_workshopthema und X\_f\_workshopzustand. Diese Klassen befinden sich in **org.compiere.model** und können nicht geändert, sondern nur durch die Implementierung einer Java-Klasse, die die zu entwickelnde Logik enthalten muss, erweitert werden.

6. Die gesamte Anwendung wird im letzten Schritt durch das sogenannte 2Pack gespeichert. Es ist eine Packaging-Applikation unter ADempiere AD. Diese Applikation erlaubt den Benutzern und Entwicklern auf eine einfache Weise, die Anpassungen und Daten mit anderen ADempiere-Anwendern zu teilen.

Die Abbildungen im Anhang A15 zeigen, wie diese Speicherung abläuft. Sie ist sowohl für Windows als auch für Linux möglich. Im Package wird eine XML-Datei erzeugt, die Anweisungen zur Anwendung oder Anpassung beinhaltet.

Dieses Package kann benutzt werden, wenn man die realisierte Anpassung in andere ADempiere-Instanzen (z.B. andere Mandanten) einfügen will.

## 7.4 Auswertung

Die Installation und Konfiguration von ADempiere ist im Vergleich mit anderen OS-ERP-Systemen leichter. Nur die Einarbeitung in ADempiere selbst ist aufwändig und benötigt eine gewisse Zeit (ungefähr vier Wochen). Das Modul „Produktionsplanung und –steuerung“ benötigt mehr Zeit für die Einarbeitung.

Es ist möglich, ADempiere mit anderer Software zu verknüpfen, z.B. Jasper für Reporting oder mit Pentaho für die Datenintegration.

Die Benutzeroberfläche von ADempiere ist einfach zu bedienen, z.B. für die Erstellung neuer Artikel oder Business-Partner.

Das System enthält viele wichtige Module, aber bei einigen fehlen bestimmte Funktionalitäten, beispielsweise Arbeitsschichten im Modul „Payrol und personal“. Solche Funktionalitäten sollten in der Zukunft implementiert werden, damit ADempiere zum produktiven Einsatz kommen kann.

Die Vorteile der modellgetriebenen Architektur von ADempiere hatten einen positiven Einfluss auf den Ablauf der Anpassung. Folgende Punkte zeigen dies:

- Schnelle Anpassung der Anwendung an neue Anforderungen:  
Die Implementierung der Anpassung läuft schnell, wenn die etwas aufwändige Einarbeitung in ADempiere erfolgt ist.
- Größere Stabilität:  
Die Fenster werden regelbasiert aus einem aktiven Application Dictionary dynamisch generiert, statt individuell programmiert werden zu müssen.
- Höhere Produktivität von Anwendern und Entwicklern:  
Sowohl die Entwickler als auch die Anwender können die implementierten Anpassungen wiederverwenden und mit anderen ADempiere-Instanzen teilen. AD erleichtert die Entwicklung der Anpassungen.
- Konzentration auf Funktionalität und Verhalten der Anwendung:  
Nach der Erstellung der Datenmodelle und Fenster durch ADempiere AD beschränkt sich die Arbeit darauf, die Logiken zu erstellen (Default-Logik, Anzeigelogik und/oder Business-Logik)

Einige Anpassungen in ADempiere lassen sich nicht in Application Dictionary realisieren. Dafür muss man ein separates Projekt anlegen (siehe Abbildung 9.23 im Anhang A16), damit der ADempiere-Quellcode unveränderbar bleibt.<sup>6</sup>

---

<sup>6</sup> Ein Beispiel für eine solche Anpassung ergänzt die bearbeitete Anpassung (Workshop) auf der dieser Arbeit beiliegenden DVD.

## 8 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurden die Möglichkeiten der Migration von DiPPS untersucht. Am Anfang stand dabei die Analyse der Funktionalitäten von DiPPS für die weitere Verwendung.

Da grundsätzlich die zwei Möglichkeiten der Open Source und der Proprietären Systeme bestehen, wurden sie im Abschnitt 2.5 miteinander verglichen. Es hat sich ergeben, dass es sich für IT-Firmen, mit besonderer Betonung auf der Dresden Informatik GmbH, lohnt, die bestehenden OS-ERP-Systeme zu erweitern, an kleine und mittelständische Unternehmen zu verkaufen und diese entsprechend zu betreuen. Aus diesem Grund wurde die Suche auf ERP-Systeme, die auf Open Source basieren, beschränkt.

Nach der Diskussion über die Migrationsmöglichkeiten eines ERP-Systems in Kapitel 4 wurde beschlossen ein neues ERP-System auszusuchen und im Rahmen eines Beispiels für eine Anpassung einzusetzen.

Die Migration und Einführung jedes ERP-Systems ist mit Datenmigration verbunden. Es wurden daher in Kapitel 5 die Strategie der Datenmigration, ihre Implementierung und die dafür möglichen Technologien diskutiert. Dieses Thema wurde auf Grund der Themenanforderung nicht in dieser Arbeit betrachtet, könnte aber Inhalt einer weiterführenden Arbeit sein.

Bevor ein Konzept für die Auswahl eines geeigneten OS-ERP-Systems erstellt und umgesetzt wurde, sind in Kapitel 6 zwei bestehende Ansätze für die Auswahl eines OS-ERP-Systems analysiert worden. Beide haben zu dem Ergebnis geführt, dass einige Open-Source basierte ERP-Systeme den Reifegrad erreicht haben, um eine ernstzunehmende Konkurrenz für Proprietäre Systeme darzustellen. Außerdem stellten sich dabei v. a. die Systeme Openbravo und ADempiere als potenzielle Kandidaten heraus.

Das Auswahlkonzept wurde in sieben Etappen durchgeführt. Auf die Herausarbeitung der Problem- und der Zielstellung folgte eine Zustandsanalyse für DiPPS. Auf Basis dieser Analyse und der in Kapitel 5 definierten Anforderungen an ERP-Systeme wurde ein Pflichtenheft erstellt. Danach wurde unabhängig von diesem Pflichtenheft eine Marktanalyse für OS-ERP-Systeme durchgeführt, die mit der Vorauswahl von drei Systemen aus 40 möglichen endete: Opentaps, Openbravo und ADempiere. Die drei Systeme wurden installiert und getestet.

Nach einem Vergleich der Ergebnisse schied Opentaps aus. Zuletzt wurden die zwei Systeme Openbravo und Adempiere nach detaillierten Funktionalitäten (115 Funktionalitäten) mit dem Pflichtenheft verglichen. Die Ergebnisse zeigten, dass Adempiere im Vergleich zu Openbravo mit 72% deutlich mehr Funktionalitäten abdeckte und somit am besten für die weitere Arbeit geeignet war.

Im nächsten Abschnitt wurde das System Adempiere eingeführt, indem das Beispiel eines Workshops für Firmen für die Anpassung im System implementiert und dann evaluiert wurde.

Adempiere beinhaltet das Application Dictionary, das die Einführung der kundenspezifischen Anpassungen erleichtert. Dafür mussten die Datenmodelle als Tabellen und Datenfelder innerhalb von Fenstern angelegt, die Logiken definiert und ebenfalls angelegt werden. Für das Beispiel Workshop wurden zwei miteinander verbundene Fenster angelegt. Es wurden die Default Logik für das Datum und die Anzeigelogik für den Tab „W Thema“ im Fenster Workshop, die das Ausblenden von Tab „ausstehende Themen“ erlaubt, wenn der „Workshopzustand“ geschlossen oder geklärt ist, definiert. Die Verbindung der Anlage eines Themas mit der Inkrementierung des Datensatzes „item“ erfolgte durch Callout.

Allgemein kann Adempiere mit einer starken Entwicklergemeinschaft auftrumpfen. Das System hat in Version 4.0.0 die Ziele:

- Migration zur modernen n-tier Architektur ermöglichen
- Ajax Client einzusetzen
- Neue Reporting Engine zu verwenden
- 100%ige Datenbankunabhängigkeit zu gewährleisten
- Neue Komponenten zur Verfügung zu stellen, z. B.:
  - neue Persistence engine (Einsatz von JPA)
  - Integration der Business Rules: Modifikation von Business-Logik ohne Änderung des Quell-Codes.
  - Integration der Business-Rules in Application Dictionary

Dies wird unter dem Link [http://www.adempiere.com/Road\\_Map](http://www.adempiere.com/Road_Map) noch detaillierter und in Hinblick auf zukünftige Entwicklungen dargestellt.

Aufgrund seiner Systemarchitektur ist ADempiere das passende System für Dresden Informatik GmbH. ADempiere wächst mit dem Unternehmen und ist in jeder Zeit auf neue Funktionalitäten erweiterbar. Durch seine GPLv2-Lizenzierung kann ADempiere in der Gegenwart und in der Zukunft ohne Lizenzkosten eingesetzt werden.

## 9 Anhang

### *A01 Inhalt der DVD*

- **/Diplomarbeit/ PDF/**  
Die Diplomarbeit als PDF-Dokument
- **/Diplomarbeit/Word/**  
Die Diplomarbeit als Word-Dokument
- **/Adempiere360Its Quellcode/**  
Das im Rahmen der Arbeit ausgewählte Open Source ERP-System
- **/Code/ Anpassung über Application Dictionary**  
Der im Rahmen der Arbeit entwickelte Prototyp für die Anpassung:
  - Datenbank Tabellen,
  - Entwickelte Benutzeroberflächen als XML-Datei, durch Packout komprimiert
  - Entwickelte und generierte Java-Klassen (Callout und Model Validator)
- **/Code/ Anpassung über Quellcodeänderungen**
- **/System Dokumentation**  
Installationsanweisung  
Benutzerhandbuch - erste Schritte  
Vorgehensweise bei Softwareanpassung

## A02 verschiedene Lizenztypen

Lizenzen	mit proprietärer Software kombinieren?	Mit anderer Software kombinieren und verbreiten?	QuellCode ändern und verbreiten?
<b>GNU General Public License (GPL)</b>	Nein	Ja, wenn die gesamte Software unter GPL ist	Ja, wenn die Software unter GPL lizenziert
<b>GNU Lesser general Public License (LGPL)</b>	Ja	Ja	Ja, wenn die Software unter LGPL oder GPL lizenziert
<b>Apache License</b>	Ja	Ja	Ja, aber "Apache" kann nicht im Namen des modifizierten Codes verwendet werden
<b>BSD License</b>	Ja	Ja	Ja
<b>Mozilla Public License (MPL)</b>	Ja	Ja	Ja, wenn die Software unter MPL lizenziert ist.
<b>MIT (X11) License</b>	Ja	Ja	Ja

Tabelle 9.1: Verschiedene Lizenzen und ihre Einschränkungen

**A03 Umfrage über die Benutzung der freien ERP-Lösungen bei kleinen und mittleren Unternehmen [ECKR]**

Bereits ein Drittel der befragten 33% der Unternehmen setzen lizenzkostenfreie ERP-Lösungen ein.

43% können sich einen Einsatz vorstellen

13% sind unentschlossen

4% lehnen den Einsatz eines OS ab.

Erwartungen mit dem Einsatz von ERP-Lösungen:

Befragte Unternehmen	Erwartung
98,4%	Flexibilität (Möglichkeit der Anpassung)
89,4%	Wegfall von Lizenzkosten
87,7%	Unabhängigkeit von einem bestimmten Hersteller
77,2%	Offenheit der Quellcodes

**Tabelle 9.2: Erwartungen des OS-ERP-System-Einsatzes**

Bei Unternehmen, die bereits OS-ERP-Systeme einsetzen

Unternehmen, die bereits OS-ERP-Systeme benutzen	Erfüllte Erwartungen
72,4%	Kostenvorteile vor der Implementierung
93,5%	Flexibilität
83,8%	Herstellerunabhängigkeit
83,8%	Quelloffenheit

**Tabelle 9.3: Erfüllte Erwartungen nach dem OS-ERP-System-Einsatz**

"Für die Unternehmen ist es entscheidend, ein ERP-System zu nutzen, das ihren aktuellen und zukünftigen Anforderungen gerecht wird und entsprechend angepasst werden kann. Nicht zuletzt wegen der durchschnittlichen Nutzung von

acht bis zehn Jahren sind die Anschaffungskosten von ERP-Lösungen bei der wirtschaftlichen Betrachtung häufig zu vernachlässigen", bewertet Uwe Salm, ERP-Experte des NEG, dieses Ergebnis.

## A04 DiPPS Module

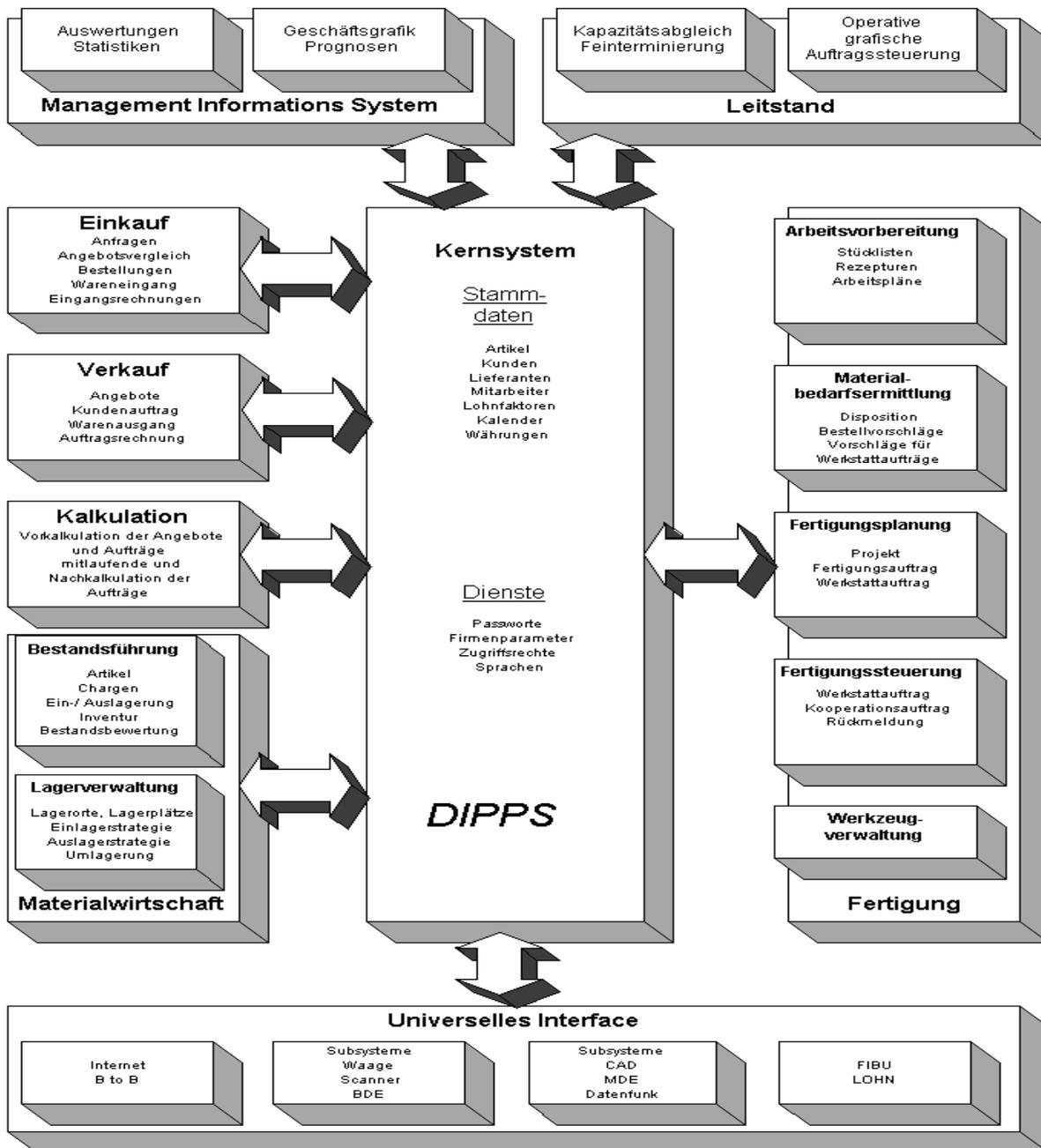


Abbildung 9.1: Übersicht über die DiPPS-Module der Dresden Informatik GmbH

**A05 Ansatz für OS-ERP-Auswahl**

KO-Kriterien	Punkte Soll	Punkte Ist	Erfüllungsgrad in %
Vertrieb	60	60	100
Buchhaltung	80	70	87,50
Systemarchitektur	280	245	87,50
<b>Gesamt</b>	<b>420</b>	<b>375</b>	<b>89,29</b>
Wunschkriterien	Punkte Soll	Punkte Ist	Erfüllungsgrad in %
Vertrieb	1000	166,67	16,67
Buchhaltung	1000	305	30,50
Systemarchitektur	1000	595	59,50
<b>Gesamt</b>	<b>3000</b>	<b>1066,67</b>	<b>35,56</b>

Tabelle 9.4: Ergebnisse der Evaluation von ADempiere

KO-Kriterien	Punkte Soll	Punkte Ist	Erfüllungsgrad in %
Vertrieb	60	60	100
Buchhaltung	80	60	75
Systemarchitektur	280	208	74,29
<b>Gesamt</b>	<b>420</b>	<b>328</b>	<b>78,10</b>
Wunschkriterien	Punkte Soll	Punkte Ist	Erfüllungsgrad in %
Vertrieb	1000	0	0
Buchhaltung	1000	150	15
Systemarchitektur	1000	589	58,90
<b>Gesamt</b>	<b>3000</b>	<b>739</b>	<b>24,63</b>

Tabelle 9.5: Ergebnisse der Evaluation von Opentaps

KO-Kriterien	Punkte Soll	Punkte Ist	Erfüllungsgrad in %
Vertrieb	60	55	91,67
Buchhaltung	80	80	100
Systemarchitektur	280	208	83,21
<b>Gesamt</b>	<b>420</b>	<b>328</b>	<b>87,62</b>

<b>Wunschkriterien</b>	<b>Punkte Soll</b>	<b>Punkte Ist</b>	<b>Erfüllungsgrad in %</b>
Vertrieb	1000	0	0
Buchhaltung	1000	30	3
Systemarchitektur	1000	580	58
<b>Gesamt</b>	<b>3000</b>	<b>610</b>	<b>20,33</b>

Tabelle 9.6: Ergebnisse der Evaluation von Ix-Office

<b>KO-Kriterien</b>	<b>Punkte Soll</b>	<b>Punkte Ist</b>	<b>Erfüllungsgrad in %</b>
Vertrieb	60	60	100
Buchhaltung	80	60	75
Systemarchitektur	280	259	92,50
<b>Gesamt</b>	<b>420</b>	<b>328</b>	<b>90,24</b>

<b>Wunschkriterien</b>	<b>Punkte Soll</b>	<b>Punkte Ist</b>	<b>Erfüllungsgrad in %</b>
Vertrieb	1000	666,67	66,67
Buchhaltung	1000	650	65
Systemarchitektur	1000	702	70,20
<b>Gesamt</b>	<b>3000</b>	<b>2018,67</b>	<b>67,29</b>

Tabelle 9.7: Ergebnisse der Evaluation von Openbravo

<b>Platz</b>	<b>System</b>	<b>KO-Kriterien</b>	<b>Wunschkriterien</b>
1.	Openbravo ERP	90,24%	67,29%
2.	ADempiere	89,29%	35,56%
3.	Lx-Office	87,62%	20,33%
4.	opentaps	78,10%	24,63%

Tabelle 9.8: Vergleich der Erfüllungsgrade

## A06 Vergleich ADempiere und Openbravo mit Pflichtenheft

Funktionalität	Beschreibung	Unterstützt von Adempiere	Unterstützt von Openbravo
<b>Artikel</b>	Mind. 2 stufige Klassifikation (Produktklassen, Produktgruppen )	Ja, in Form von beliebig stufige Produkt-Kategorie	Ja
	Art der Lagerverwaltung wählbar (Einlager, Mehrlager, Chargen, Seriennummern)	Ja, in Materialwirtschaft (lager und Lagerort)/in Los-/Chargen-Nr.	ja
	Wareneingangsprüfung wählbar	Ja, in Beschaffung Modul	ja
	Überlieferungssatz wählbar	Ja, in Modul Rücksendung (Returns)	Nein
	Administrator kann Eigenschaften selbst definieren	nein	Nein
	kunden-/lieferantenspezifische Artikelnummern und -beschreibungen	ja	ja
	Losgrößen für Beschaffung (Einkauf/Fertigung) wählbar	Ja, in Beschaffung und Materialwirtschaft Module	ja
	Gruppenänderung	nein	Nein
	mittlerer Einkaufspreis	Nein	Nein
	Zeichnungsverwaltung zum Artikel, Auftrag oder Kunden	ja	ja
<b>Lager</b>	variable 3-dim. Struktur	ja	ja
	Lagerrechte für Buchungsarten	ja	ja
	Lagerrechte für Nutzer	ja	ja
<b>Materialwirtschaft</b>	Inventur	ja	ja
	Umlagerung	Ja, in Warenbewegung smodule	Nein
	Zu/Abgangsbuchungen immer gegen Lagerbestand	ja	ja
	Buchen in negativen Bestand	ja	ja
	Buchung Bestandkorrekturen gegen Kostenstellen	Nein	Nein
	Chargenverwaltung mit MHD und Rückverfolgung	ja	Nein
	Qualitätskontrolle, Prüfpläne	Ja, in Quality Management Modul	Nein
<b>Stücklisten</b>	Status und Änderungsindex	ja	ja
	Phantombaugruppen verwaltbar (werden bei Fertigungsplanung in ihre Bestandteile aufgelöst)	Nein	Nein
	Komponententausch	ja	Nein
	Strukturansicht	ja	ja
<b>Arbeitspläne</b>	Status und Änderungsindex	nein	ja
	Alternativarbeitspläne, losgrößenabhängig	nein	Nein

	Arbeitsfolgen mit Bezug auf Arbeitsgang (verbal), Arbeitsgruppe (Maschine – Kapazität, Kosten) und Lohngruppe (Mitarbeiter – Kosten)	Ja	Ja
	Kooperationsarbeitsfolgen	Nein	Nein
	Arbeitsgruppentausch	Nein	Nein
<b>Bedarfsplanung</b>	Umwandlung von Fertigungsvorschläge	Ja	Ja
	<b>MRP II :</b> - Berücksichtigung von Bedarfen aus Meldebestandsunterschreitung, Kundenaufträgen, Werkstattaufträgen	ja	Ja
	- Berücksichtigung von geplanten Zugängen aus Bestellungen und	Nein	Nein
	- aus Stücklistenauflösung	Nein	Nein
	- Umwandlung von Bestellvorschlägen	Ja, Bestellung aus Bedarf	Ja, Bestellung aus Bedarf
<b>Fertigung/Produktion</b>	Zuordnung zum Kundenauftrag möglich	Ja	Ja
	Behandlung von Erzeugnisstrukturen als Fertigungsstruktur	Ja, in Module Materialwirtschaft / Produktion	Ja
	Auftragseinplanung für Fertigerzeugnisse und Baugruppen	Ja, durch BOM/ manufacturing management/Bill of Materials & Formulas(Bill of materials)	Ja
	Kopie von Stückliste und Arbeitsplan zum Auftrag speichern	Ja	Ja
	Terminierung über Arbeitsplan	Ja	Ja
	Terminierung mit endlichen/unendlichen Kapazitäten (Arbeitsgruppen, Mitarbeiter) über Kalender	Nein	Nein
	Ausschussplanung	Nein	Nein
	Fertigungsgrobplanung mit Materialbeschaffungskontrolle	Nein	Nein
	Arbeitspapiere erzeugen	Nein	Nein
	Rückmeldung Materialentnahmen	ja	Ja
	Rückmeldung Arbeitszeiten	Ja	Ja
	Rückmeldung Ausschuss	Nein	Nein
	Buchen Fertigungszugang	ja	ja
	Kooperationsaufträge für Kooperationsarbeitsgänge	Nein	Nein
	Auftragsüberwachung	Ja	Nein
	Auftragstyp Service/Reparatur (keine planbare Arbeit, nur Istbuchungen)	Ja	Ja
	<b>Geschäftspartner</b>	Klassifikation: Interessent, Kunden, Lieferant, Referent, Mitarbeiter	ja
Adressstamm		ja	Ja
Kontaktpersonenstamm		ja	Ja
<b>Kundendaten :</b> - Kreditlimit		ja	Ja
- Umsatzkategorien		ja	Ja
- Preise und Rabatte		ja	Ja
- Rahmenverträge		Nein	Nein

	<b>Lieferantendaten:</b> - Preise und Rabatte	ja	Ja
	- Rahmenverträge	ja	Nein
	<b>Referentendaten:</b> - Provision	ja	Ja
	<b>Mitarbeiter:</b> - Schichtplan	nein	Ja
	- Kommt/Geht-Erfassung	nein	Nein
	Administrator kann Eigenschaften selbst definieren	Nein	Nein
	Serienbriefe	Nein	Nein
<b>Beschaffung</b>	Anfragen, Preisvergleich	ja	Nein
	Kopie Anfrage zu Bestellung	ja	Nein
	Kopie Kundenauftrag zu Bestellung	ja	Ja
	Bestellung: - Position mit Liefersplittung	Nein	Nein
	- Nebenkosten	ja	Ja
	- Abruf aus Rahmenvertrag	Nein	Nein
	Wareneingang: wahlweise Bestandserhöhung oder sofortige Entnahme gegen Fertigung (Doppelbuchung)	ja	Nein
	Rechnungsprüfung	ja	Nein
	Überweisung	ja	Ja
	Lieferantenbewertung	ja	Nein
	Anbieterverzeichnis (wer kann was liefern)	nein	Nein
<b>Verkauf</b>	<b>Angebot:</b> - Position, die keine Artikel	Ja, in Vertrieb Modul. Angebot ist eine Art von Auftrag.	Ja
	- Mengenstaffelung	ja	Ja
	- Alternativpositionen	ja	Ja
	- Nebenkosten	ja	Ja
	Kopie Angebot zu Kundenauftrag	Ja, Angebot in Auftrag umwandeln	Ja
	Kopie Bestellung zu Kundenauftrag	Ja, in Beschaffung Modul/ Bestellung/Zeilen aus anderem Auftrag kopieren	Ja. In Modul „Bestellwesen“/Bestellung (Einkauf)
	<b>Kundenaufträge:</b> - Positionen mit Liefersplittung	Nein	Nein
	- Nebenkosten	Ja	Ja
	- Abruf aus Rahmenvertrag	Nein	Nein
	- Zahlplan	ja	Ja
	Buchen Warenausgang: - Lieferscheinerstellung	ja	Nein
	Reklamationsbearbeitung: - Eingang, Bearbeitung, Rücklieferung	ja	Ja
	<b>Rechnungslegung:</b> - Auftragsrechnung (Kundenauftrag, Reklamation)	ja	Ja
	- Sammelrechnung (Kundenaufträge)	ja	Nein
	- Vorkasserechnung (Kundenauftrag, Reklamation)	ja	Ja

	- Kostenstellenrechnung	Ja	Ja
	- Fertigungsrechnung für Service/Reparatur	Ja, Management Maintenance in Manufacturing Management Module	Ja
	- Zahlungsüberwachung	Ja	Ja
<b>Kalkulation</b>	Zuschlagskalkulation	Ja	Nein
	Angebotskalkulation	Ja	Ja
	<b>Fertigungskalkulation:</b> - Vorkalkulation über Planmenge Fertigerzeugnis	Ja,	ja
	- mitlaufende Kalkulation über Planmenge eines jeden Auftrages	Nein	Nein
	- Nachkalkulation über Istmenge Fertigerzeugnis	Ja, in Standard Costing Management	Nein
	- Kostenarten: Materialkosten, Materialgemeinkosten, Lohnkosten, Lohngemeinkosten, Kooperationskosten, Fertigungsgemeinkosten	Ja	Ja
	- Führung der Kostenarten einzeln über Erzeugnisstruktur	Nein	Nein
	Kundenauftragkalkulation	Ja	Ja
	<b>Produktionskalkulation:</b> - Soll: Standardkostenermittlung über Stückliste und Arbeitsplan	Ja, in Planning Management/Forecast Management	Ja
	- Ist: Fertigungsnachkalkulation	Ja, in Manufacturing Management	Nein
<b>Dokumentenverwaltung</b>	Erstellung, Zuordnung von Dokumenten zu Stammdaten und Vorgängen: Dateien, E-Mails	Nein	Nein
	Dokumentensuche mit Referenzierung der ERP-Daten	Nein	Nein
<b>Projektmanagement</b>	Projekte/Budget/Ausgaben	ja	Ja
	Mitarbeiterplanung, Planzeiten/Istzeiten, Beteiligungen	Nein	Nein
	Zuordnung beliebiger Vorgänge (Angebot, Kundenauftrag, Bestellung, Fertigung,...)	Ja, aber als Projektzyklus. Z.B. die Etappen eines Projekt (Konzeption-Implementierung-Testen)	Ja, aber als Projektzyklus. Z.B. die Etappen eines Projekt (Konzeption-Implementierung-Testen)
<b>Schnittstellen</b>	Finanzbuchhaltung	Ja, DATEV Export-Schnittstelle	Ja, in Financial Management Modul
	Lohnbuchhaltung	Ja, DATEV Export-Schnittstelle	Ja, in Financial Management/Accounting
	Export (xml, cvs...)	ja	Ja
<b>Auswertungen und Statistiken</b>	Darstellung: Tabellen, Pivots, Diagramme	ja	Ja
<b>Workflow</b>		ja	Ja

<b>Archivierung</b>		Ja, Dokumente und Reports durch Archiv-Viewer	Ja, Archivierung von Dokumente und Reports
<b>Report-generator</b>		Ja, Komplette und sofort einsetzbare Jasper-Belege für Angebot, Auftrag, Lieferschein, Rechnung, Bestellung	Ja

**Tabelle 9.9: Detaillierter Vergleich nach Funktionalitäten zwischen ADempiere und Openbravo gegenüber des Pflichtenhefts**

## A07 AD Validation Fenster

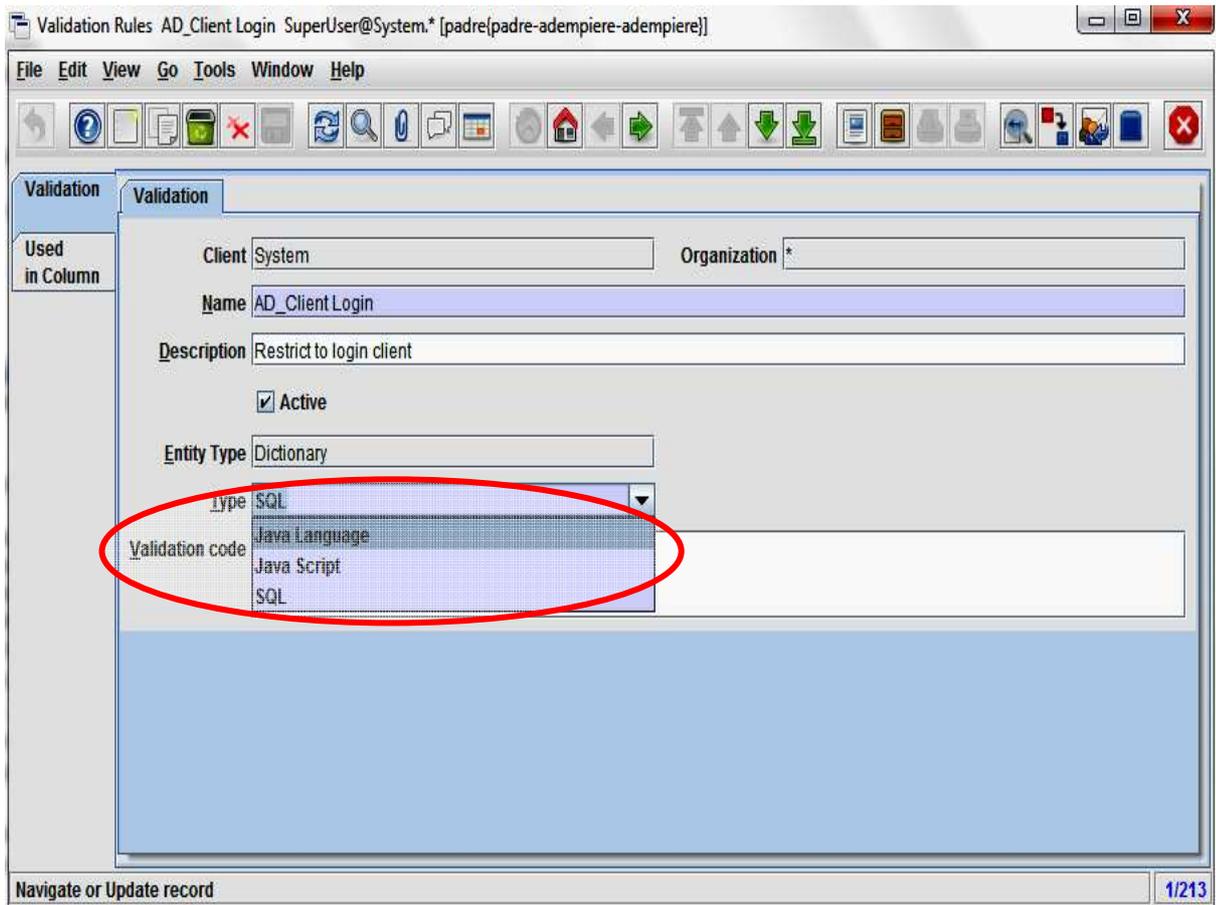


Abbildung 9.2: AD Validation Fenster

## A08 AD Rule Engine Fenster

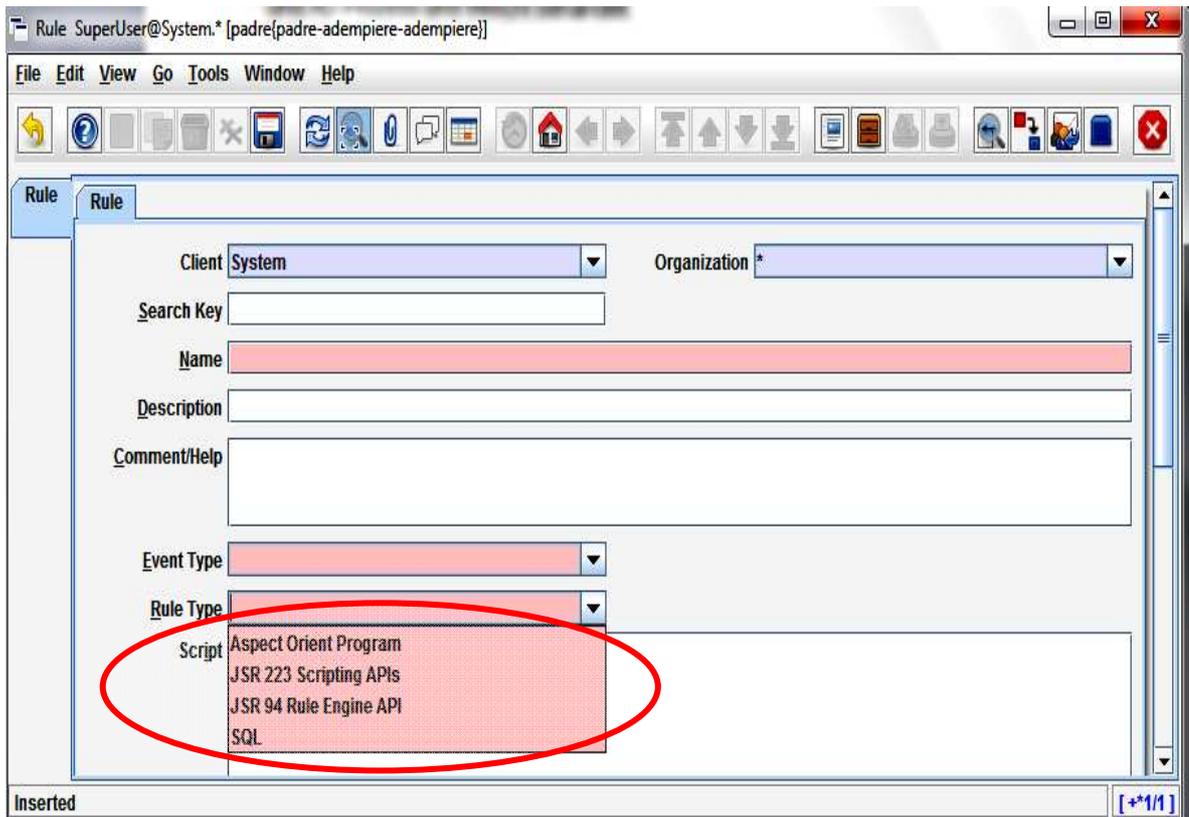


Abbildung 9.3: AD Rule Engine

## A09 Verweis der Datenfelder- und Anzeigetypen auf Datenquelle

The screenshot shows the SAP Data Dictionary configuration for a field named 'firma'. The configuration is as follows:

Field	Value
Mandant	System
Organisation *	
Tabelle	f_workshopteilnehmer_W Teilnehmer
Spaltenname	firma
Column SQL	
System-Element	firma
Name	Firma
Beschreibung	
Kommentar/Hilfe	
<input checked="" type="checkbox"/> Aktiv	
Preislistenversion	
Länge	80
Referenz	Table
Dynamische Validierung	
Referenzschlüssel	AD_Org (all)
Standardwert Logik	

The 'Referenz' and 'Referenzschlüssel' fields are circled in red in the original image.

Abbildung 9.4: Referenzierte Datenfelder

## A10 Erstellung der Fenster „Workshop“ und „Workshopzustand“ in AD

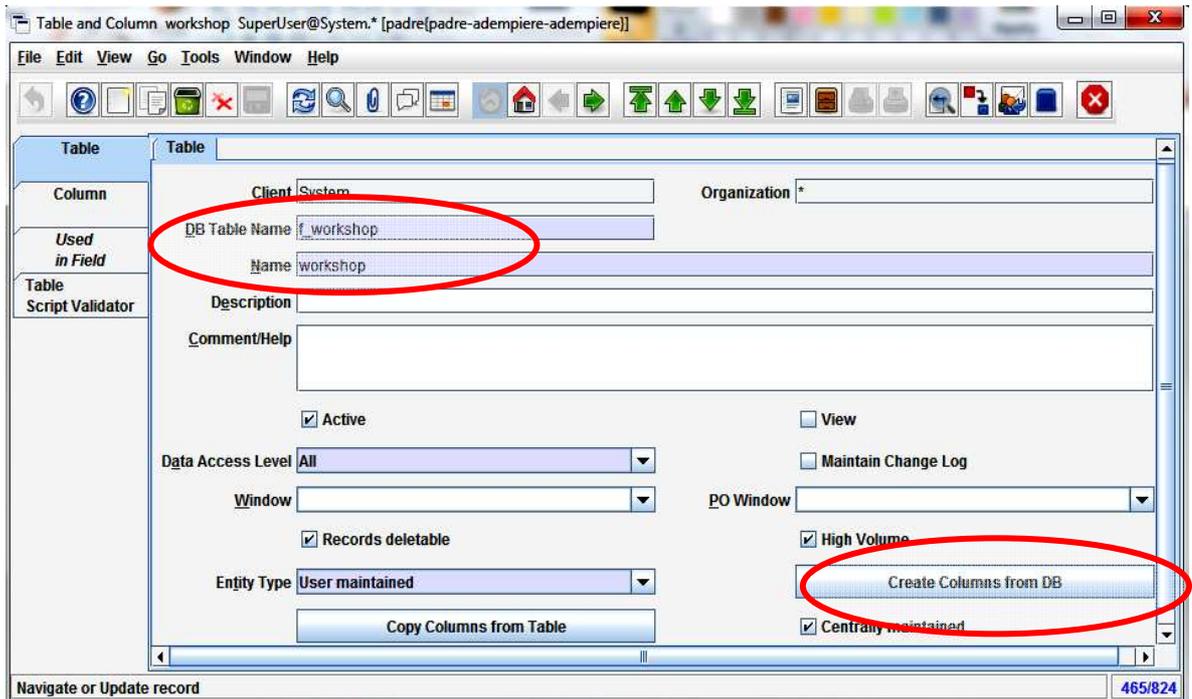


Abbildung 9.5: Tabelle f\_workshop in Application Dictionary

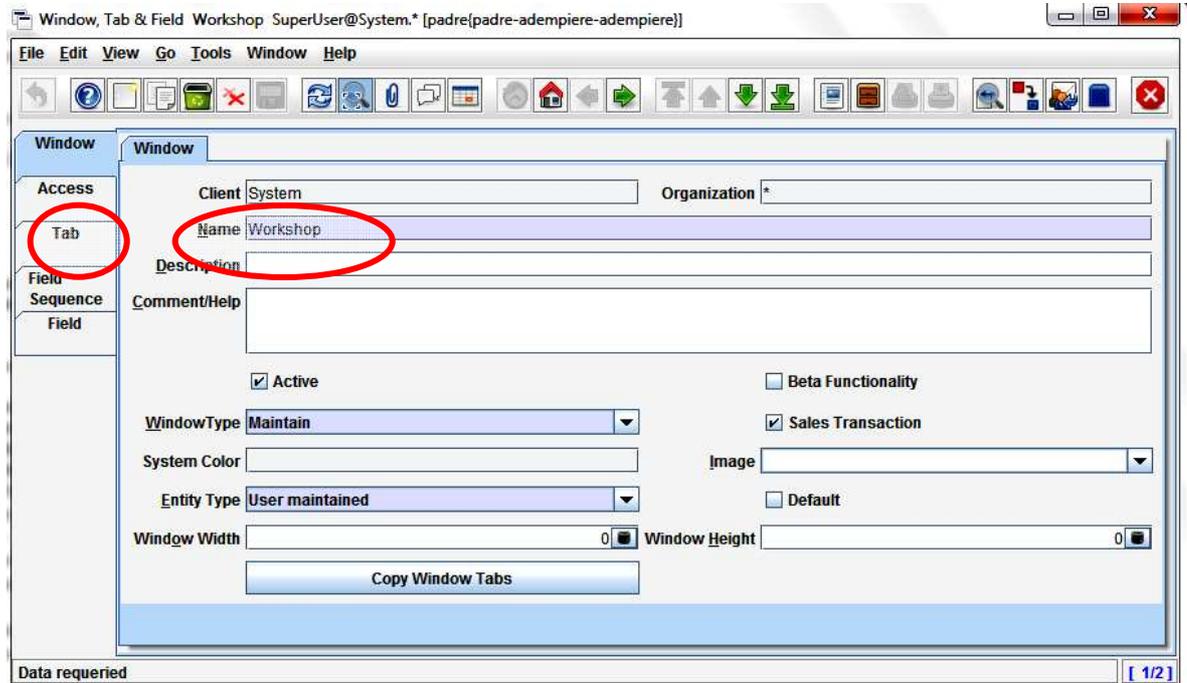


Abbildung 9.6: Erstellung des "Workshop"-Fensters

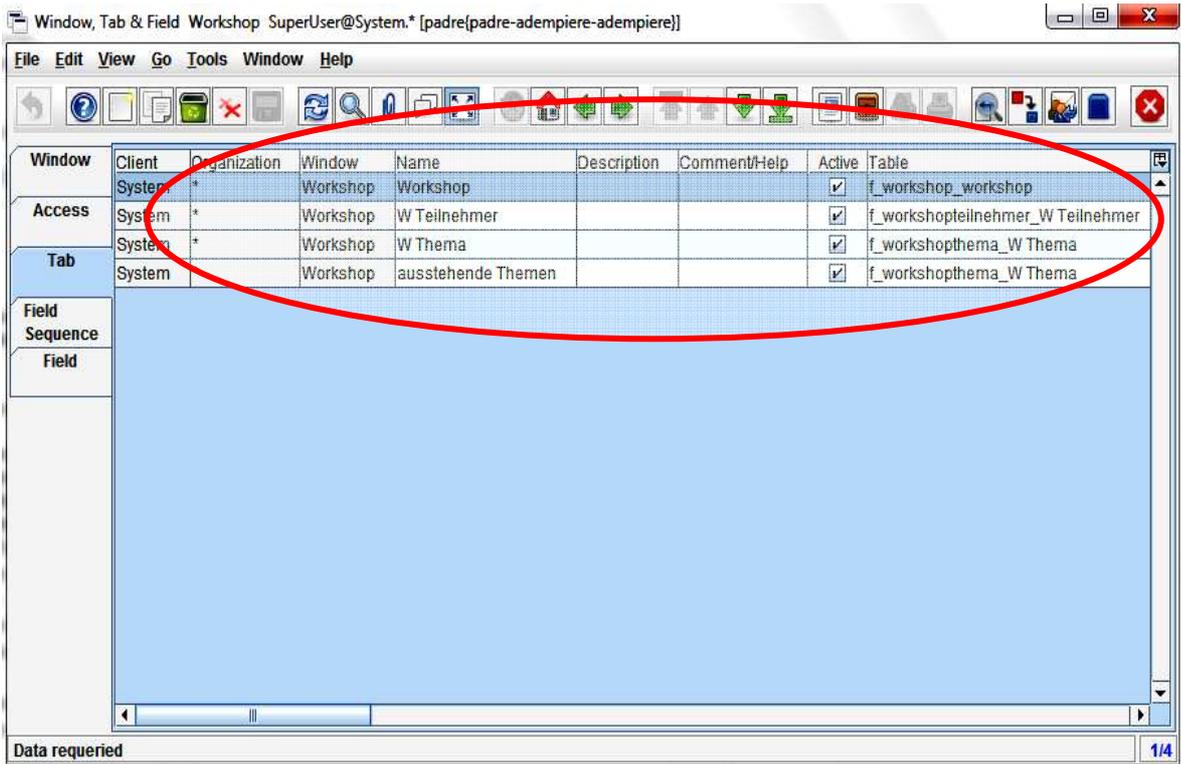


Abbildung 9.7: Datenfelder des "Workshop"-Fensters

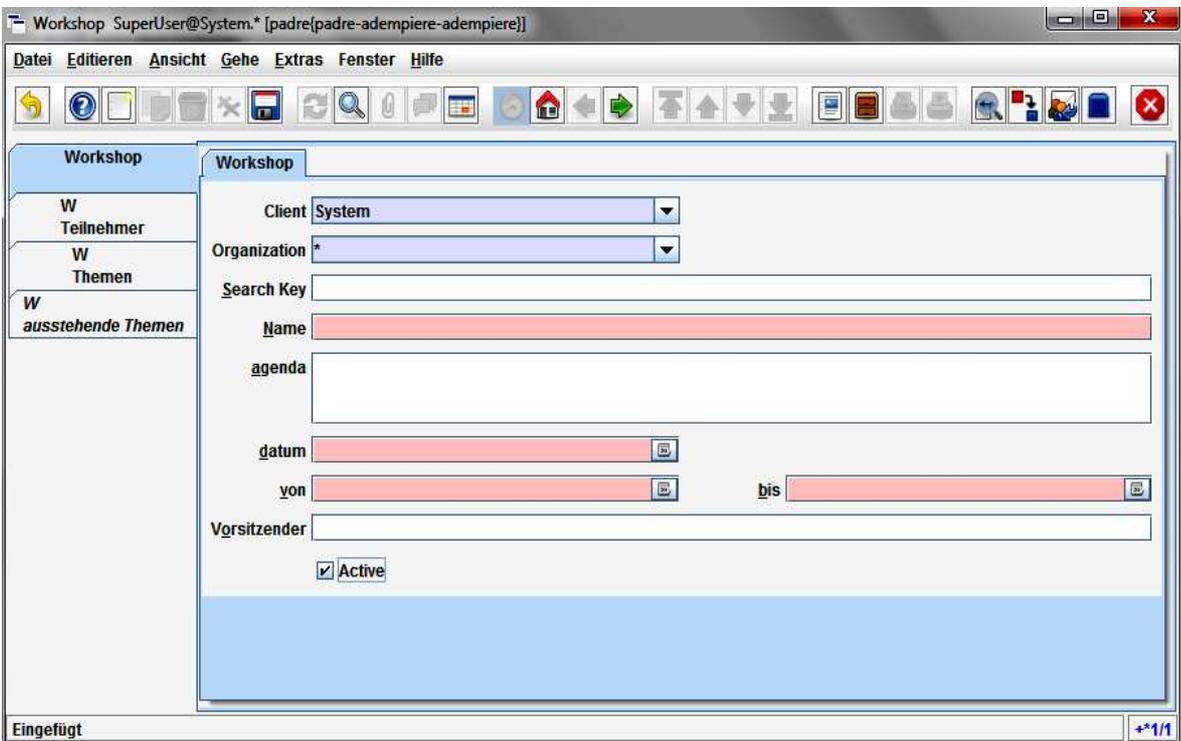


Abbildung 9.8: "Workshop"-Fenster - Tab Workshop

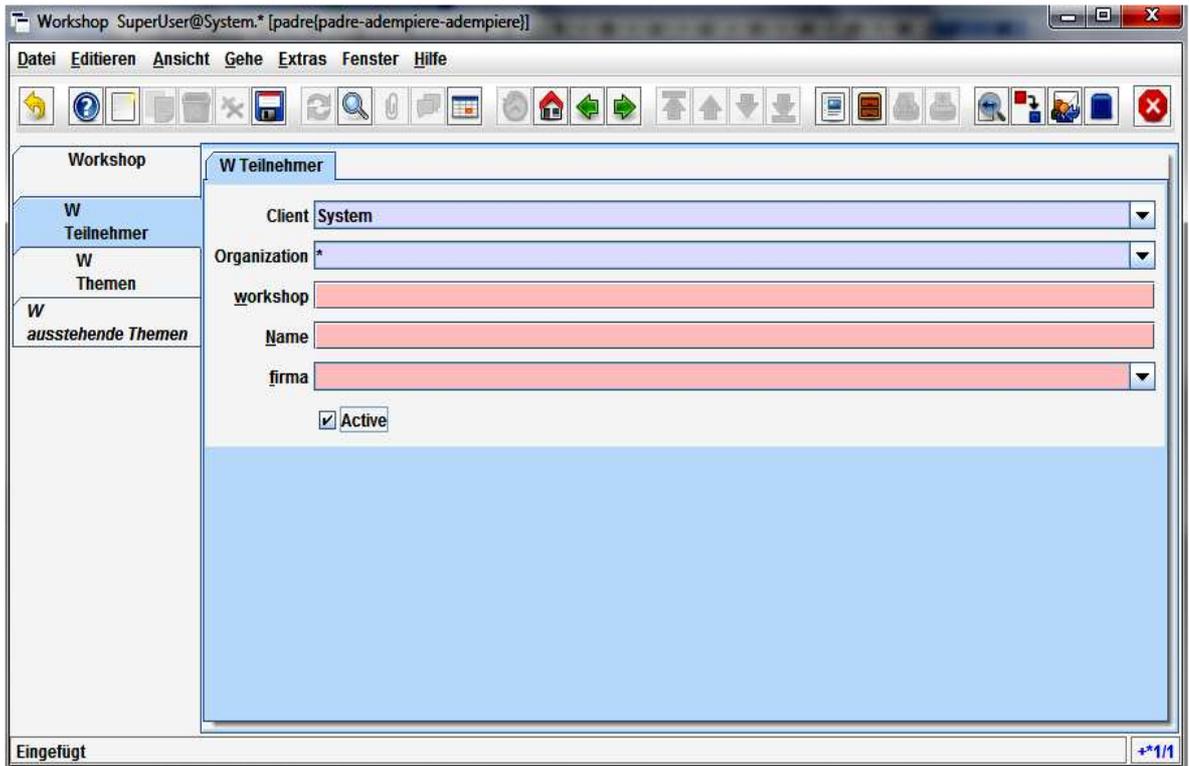


Abbildung 9.9: "Workshop"-Fenster - Tab Teilnehmer

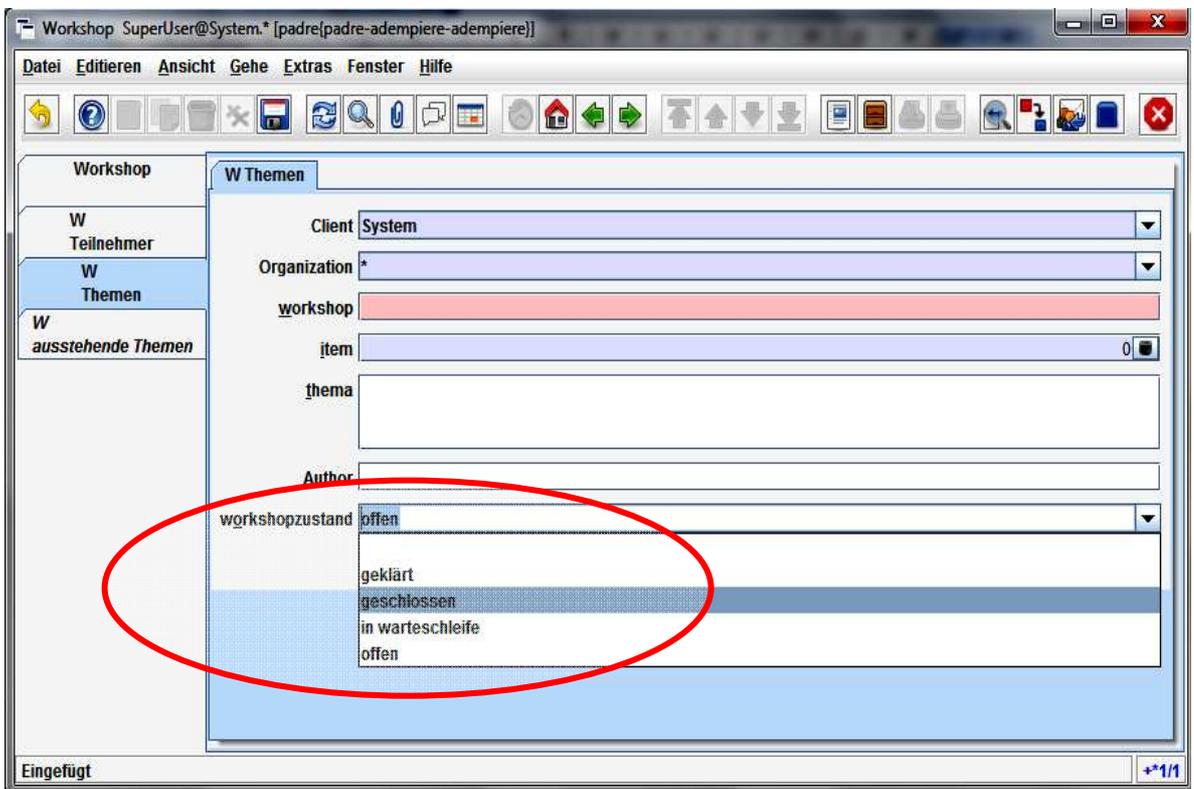


Abbildung 9.10: "Workshop"-Fenster - Tab Thema

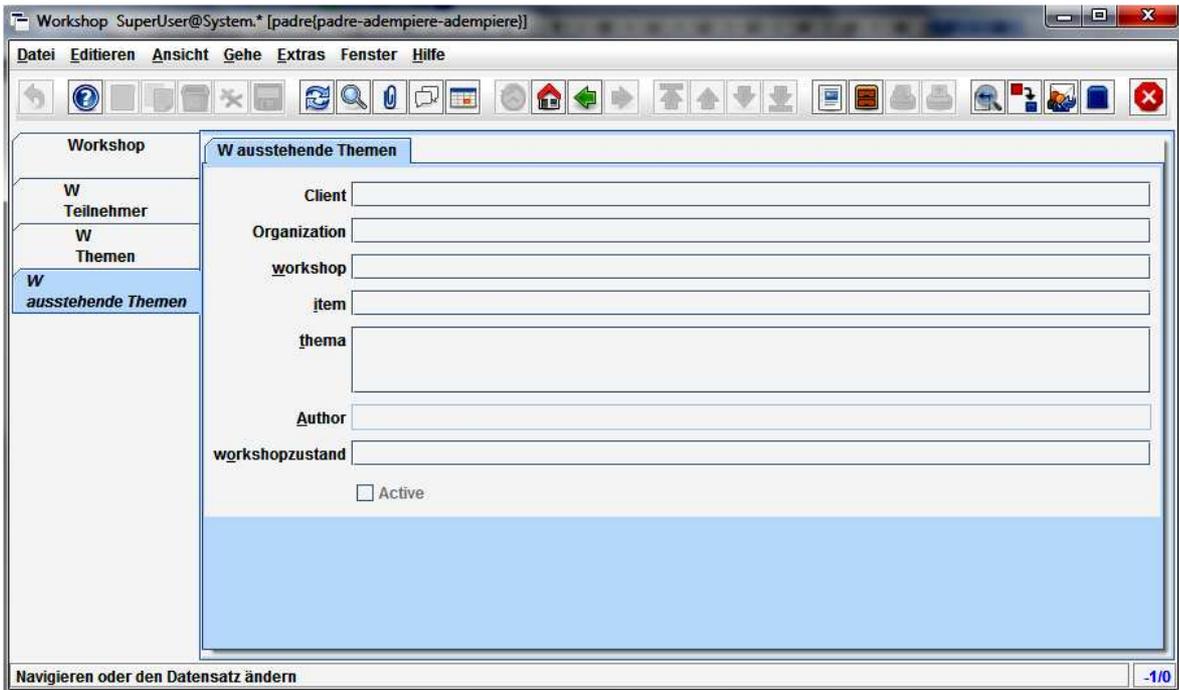


Abbildung 9.11: "Workshop"-Fenster - Tab ausstehende Themen

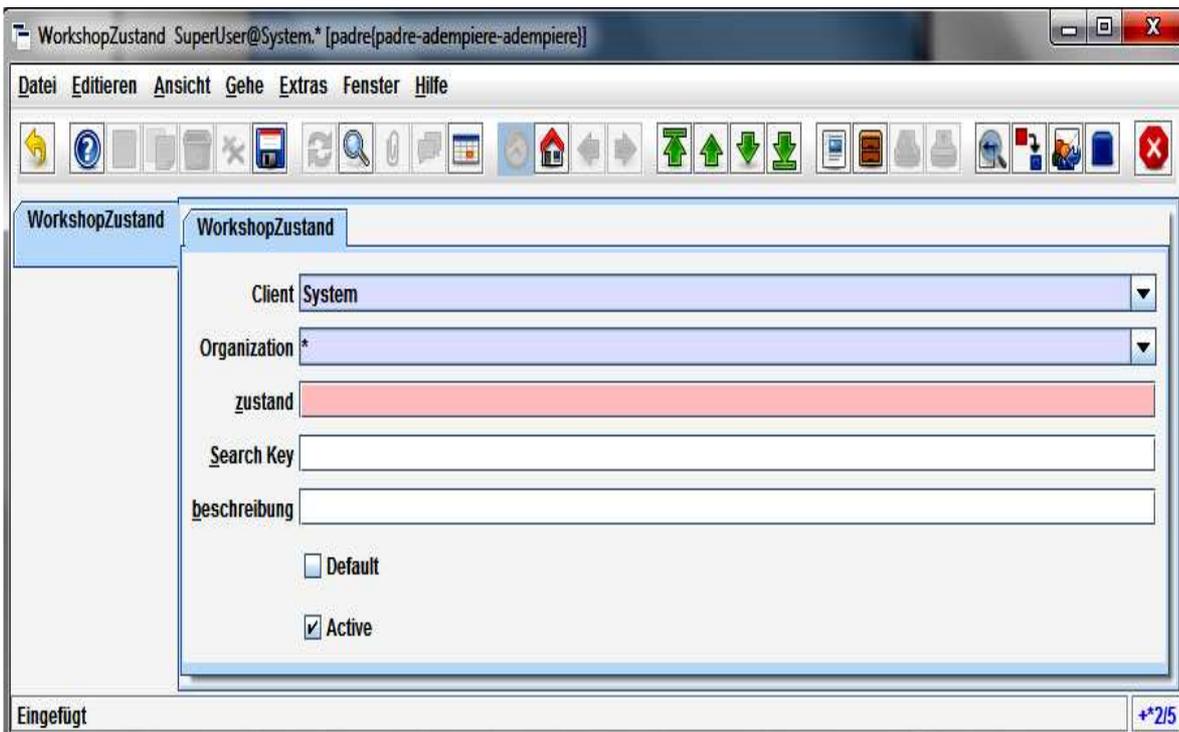


Abbildung 9.12: Workshopzustand-Fenster

## A11 Verbindung der zwei Fenster „Workshop“ und „Workshopzustand“

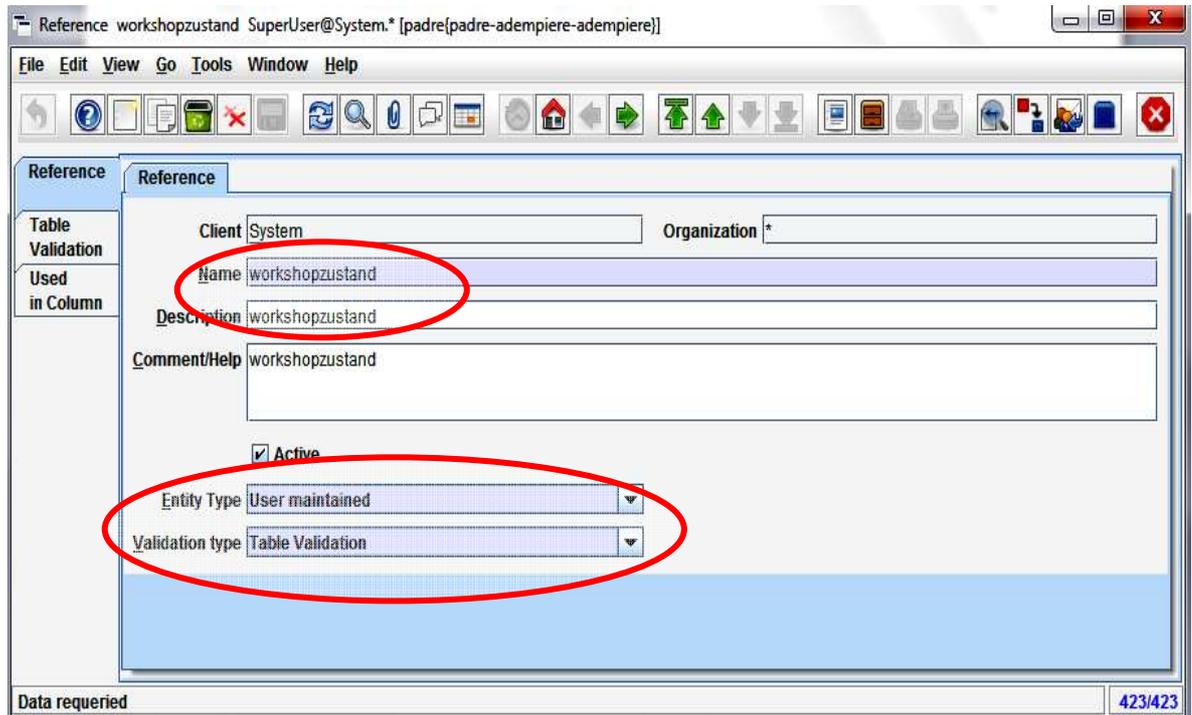


Abbildung 9.13: Verweisung des Fensters "Workshopzustand" auf das Fenster "Workshop" (Teil 1)

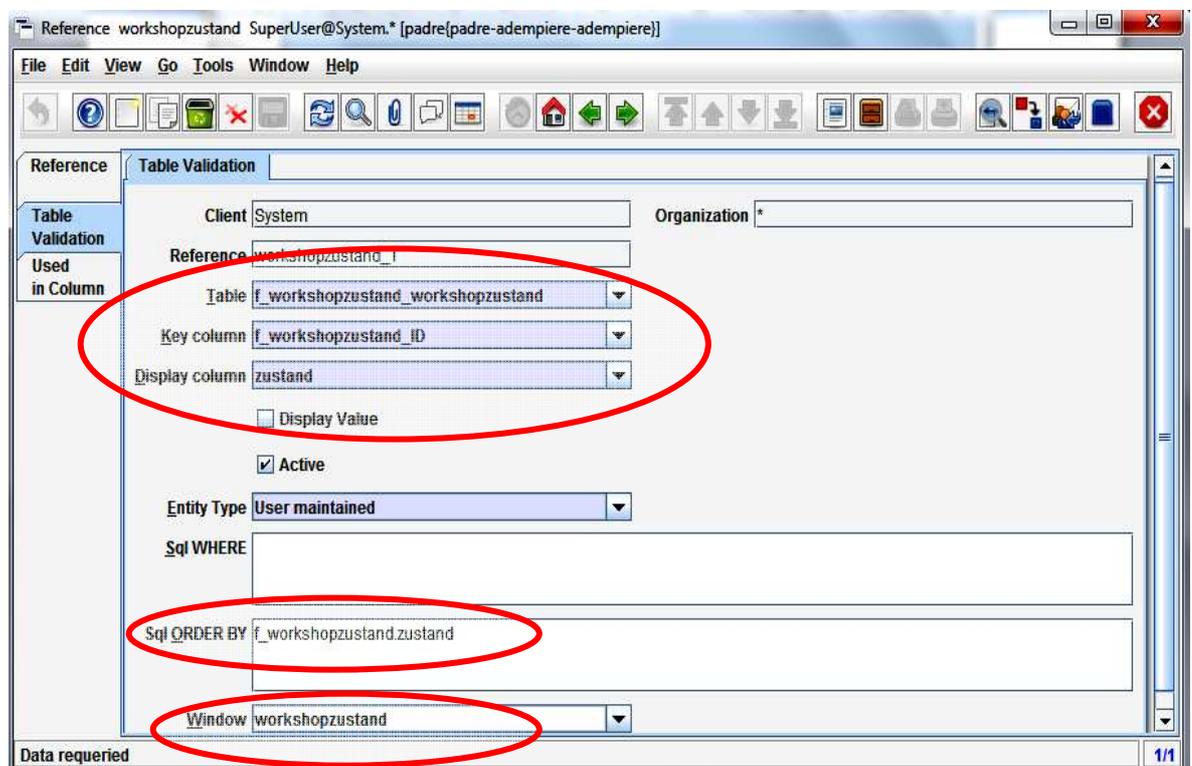


Abbildung 9.14: Verweisung des Fensters "Workshopzustand" auf das Fenster "Workshop" (Teil 2)

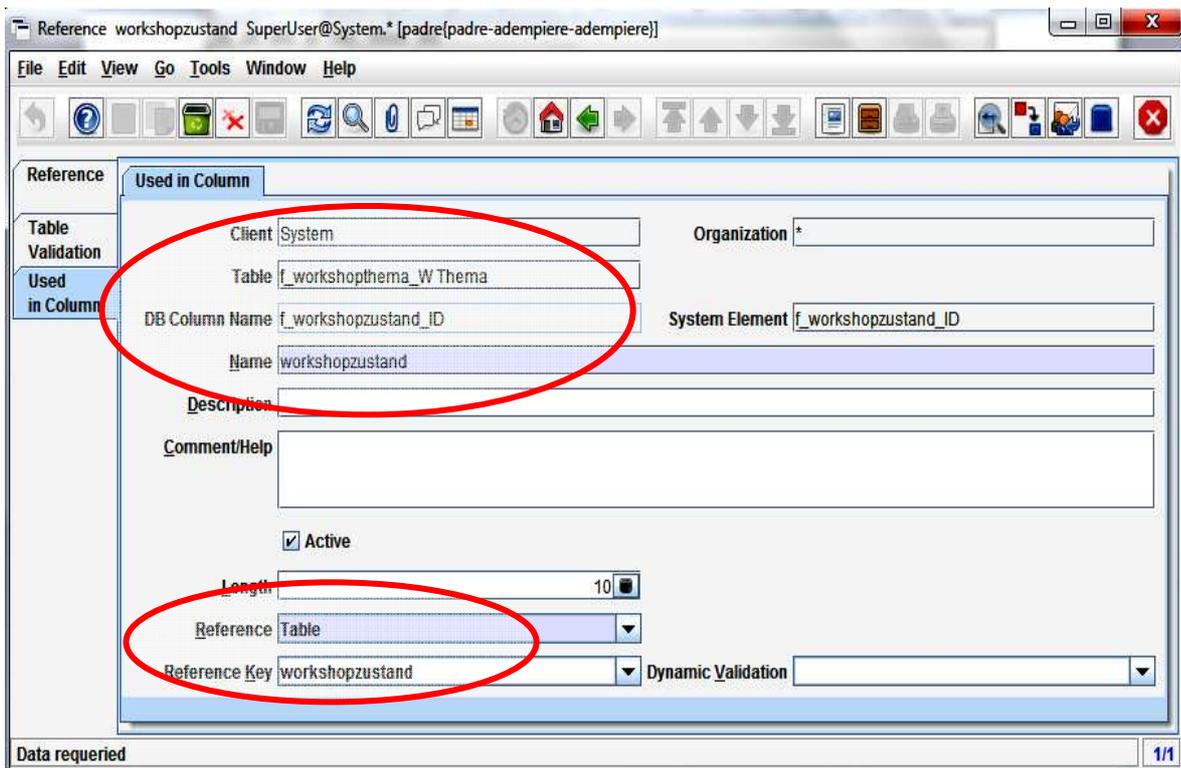


Abbildung 9.15: Verweisung des Fensters "Workshopzustand" auf das Fenster "Workshop" (Teil 3)

## A12 Die Logik für erstellte Fenster

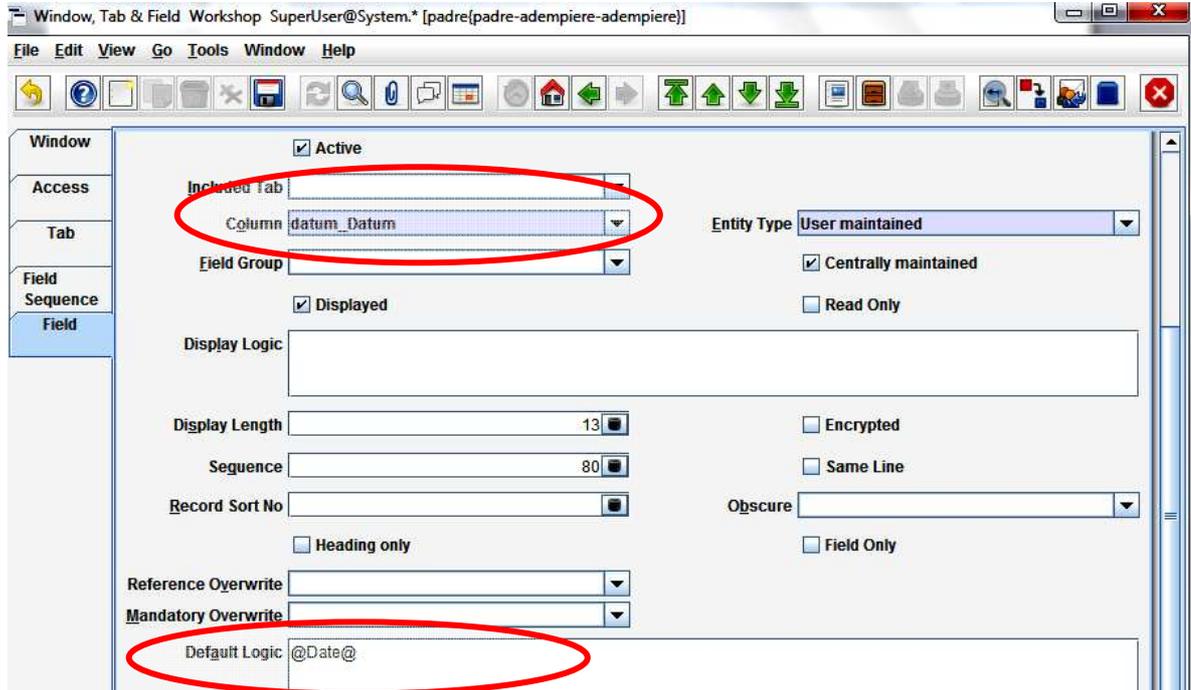


Abbildung 9.16: Default Logik für "Datum" im Fenster "Workshop"

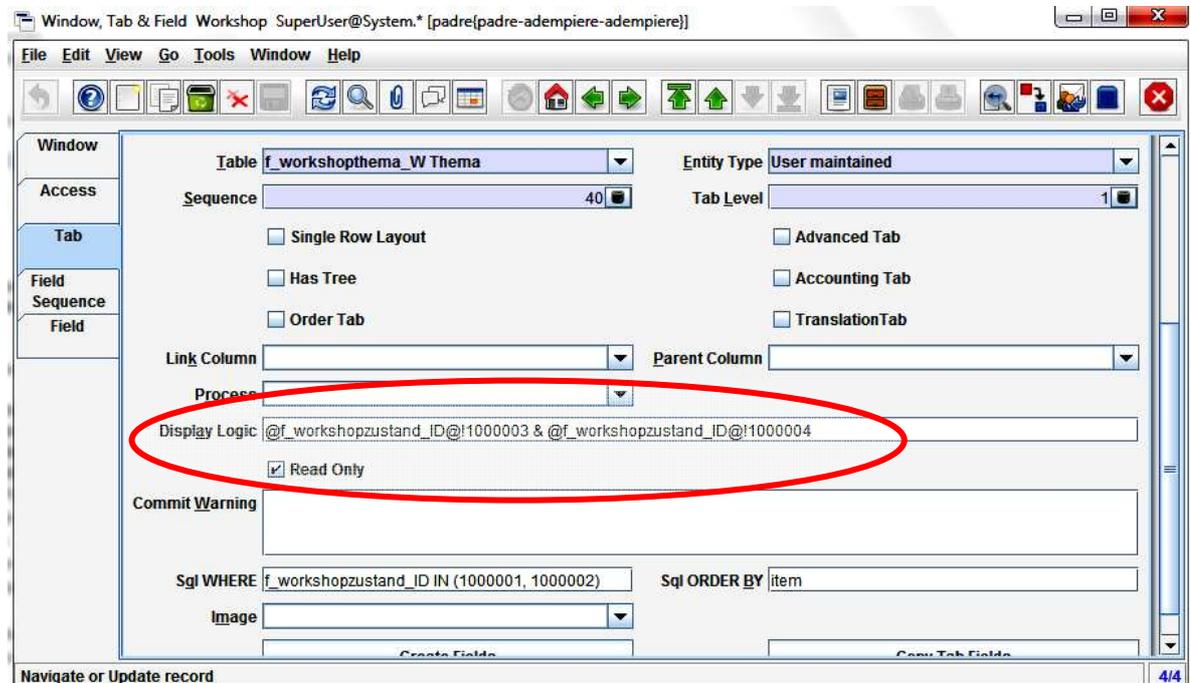


Abbildung 9.17: Anzeige Logik für den Tab "ausstehende Themen" im Fenster "Workshop"

### A13 Anlegen von Callout in Tab „W Thema“ für das Datenfeld „item“

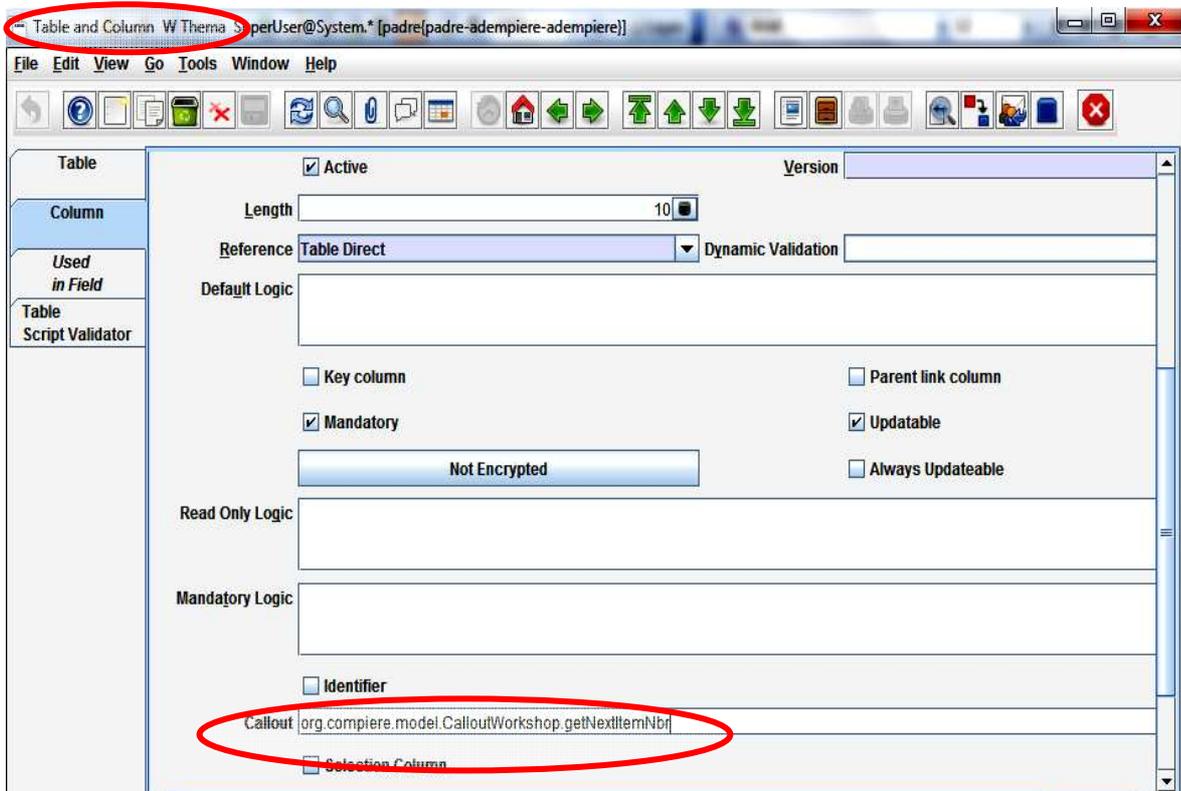


Abbildung 9.18: Callout in Tab "W Thema"

## A14 Generierung der Model Validator für Business-Logik

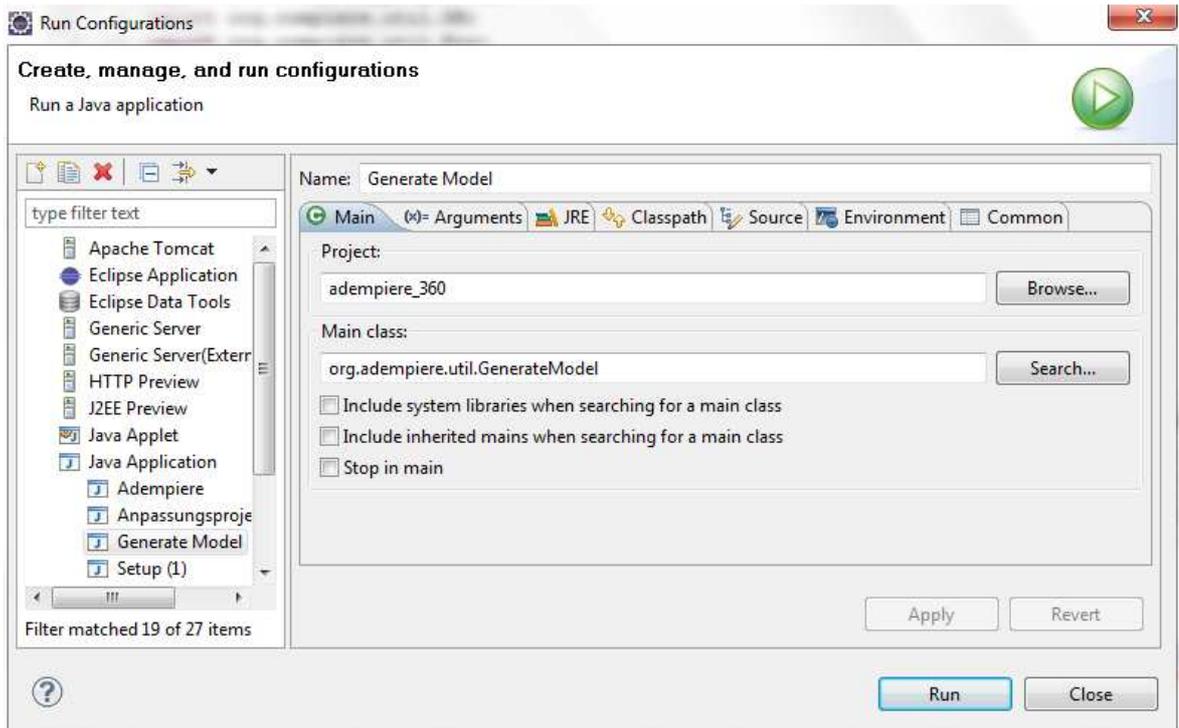


Abbildung 9.19: Java Applikation für die Generierung der Model Validatoren (Teil 1)

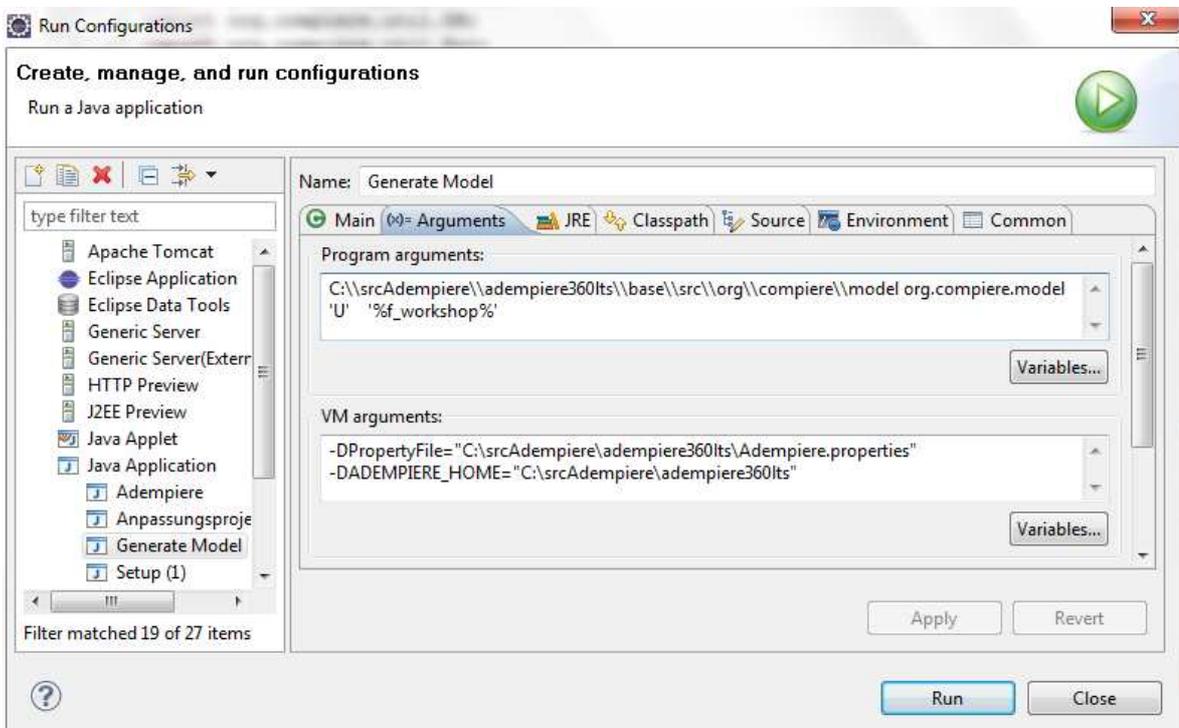


Abbildung 9.20: Java Applikation für die Generierung der Model Validatoren (Teil 2)

## A15 Speicherung der Anwendung durch Packout

The screenshot shows the 'PackOut - Create a package' window. The 'Export Package' dialog is open, displaying the following fields:

Client	System	Organization	*
Name of Package	WorshopPackout	Package Version	1.0
Registered EMail	hicham_nasri@web.de	E-Mail Address	hicham_nasri@web.de
Creation Directory	c:\Packout		
Version	-	Release No	No specific release
Description of Package	dieser Package legt eine Anwendung für Workshop bei Firmen ein. er verbindet zwei fenster workshop und workshopzustand		

Record saved 1/1

Abbildung 9.21: PackOut für die Anwendung "Workshop" (Teil 1)

The screenshot shows the 'PackOut - Create a package' window. The 'Package Details' dialog is open, displaying the following fields:

Client	System	Organization	*
Package Build ID	1000002		
Line No	10		
Type	Application or Module		
Menu	Workshop		
Notes			

Abbildung 9.22: PackOut für die Anwendung "Workshop" (Teil 2)

## A16 Anpassung durch QuellCode

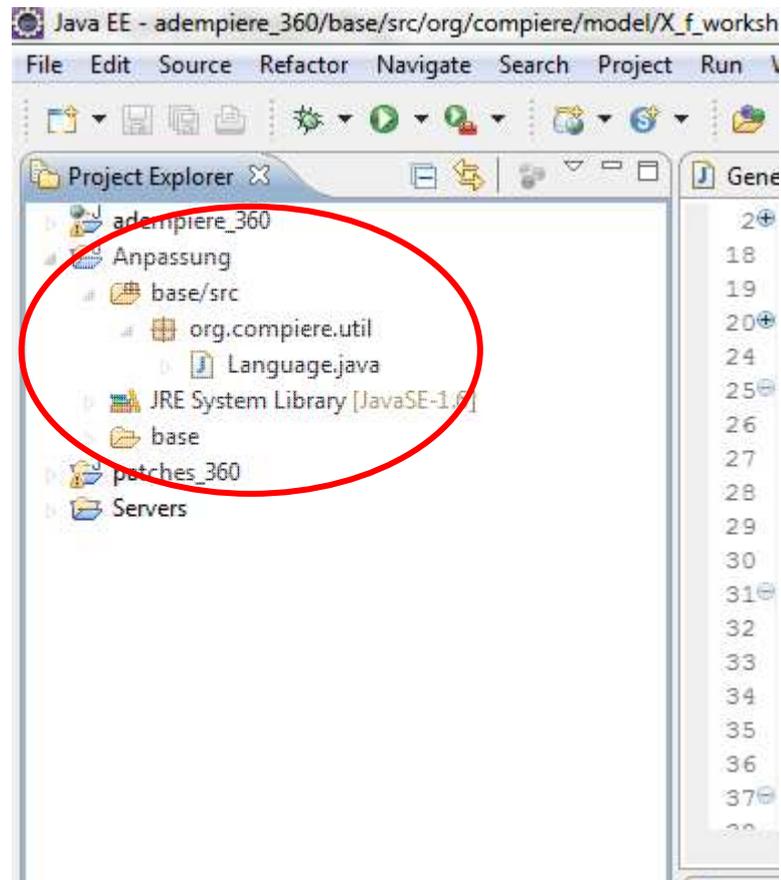


Abbildung 9.23: Anpassung (Sprachauswahl bei der Anmeldung) im separaten Projekt

## Literaturverzeichnis

### Sekundärliteratur:

- [BazRay] The Cathedral & Bazaar- Musings on Linux and Open Source by an Accidental Revolutionary: Eric S. Raymond
- [GrAnd] Enterprise Resource Planning:  
N. Gronau, K. Andresen. 2006, Universität Potsdam
- [Gronau] Enterprise Resource Planning und Supply Chain Management:  
N. Gronau, 2004, S.17
- [HesGör] Basiswissen ERP-Systeme: Auswahl, Einführung & Einsatz betriebswirtschaftlicher Standardsoftware.  
M. Hesseler, M. Görzt, 2007
- [KHMIg] IT-Integration & Migration: Knut Hildebrand  
Dpunkt Verlag, Heidelberg 2007
- [nutzEf] Nutzeffekte von CIM-Komponenten und Integrationskonzepten (Teil 1):  
M. Schumann, P. Mertens, S.46
- [Vwal] Einführungsstrategien für OpenSource-ERP-Lösungen. Ein Praxisbeispiel mit openERP: M. Sc. Falk Neubert,  
Nürnberg, 28. Januar 2009
- [W&W] Enterprise Resource Planning:  
D. Werth, P. Walter, 2006, S.10

### Internetquellen:

- [AD] Application Dictionary Adempiere  
[http://www.adempiere.com/Application\\_Dictionary](http://www.adempiere.com/Application_Dictionary)  
Abruf: 09.06.11
- [ADem] Open source ERP-System  
<http://www.adempiere.de/>  
Abruf: 12.06.11
- [ADwiki] ADempiere ERP Wiki  
[http://www.adempiere.com/ADempiere\\_ERP](http://www.adempiere.com/ADempiere_ERP)  
Abruf: 12.06.11

- [AnbERP] ERP Systeme  
<http://erp-anbieter.de/erp-programme>  
 Abruf : 25.02.11
- [Balz] Pflichtenheft (Aufbau nach Balzert)  
<http://st.inf.tu-dresden.de/SalesPoint/v3.1/tutorial/stepbystep2/pflh.html>  
 Abruf: 03.05.11
- [CaOut] ADempiere- callout  
<http://www.adempiere.com/Callout>  
 Abruf. 01.07.11
- [catAD] Catura AG: ADempiere Software Anbieter- Tabelle: Open Source im Vergleich  
<http://www.catura.de/index.php/de/know-how/open-source-alternativen.html>  
 Abruf: 10.03.11
- [Cpiere] Compiere ERP-System  
<http://www.compiere.com/>  
 Abruf: 24.06.11
- [DaMig] Data Migration in ERP Projects:  
<http://www.slideshare.net/BSMConsulting/data-migration-in-erp-projects-white-paper>  
 Abruf: 20.05.11
- [DDict] Data Dictionary von Compiere  
<http://wiki.compiere.com/display/docs/Data+Dictionary>  
 Abruf:01.06.11
- [ECKR] Freie ERP- Systeme zunehmend attraktiv für KMU: Umfrage von Electronic Commerce- Kompetenzzentrum Ruhr  
[http://www.ec-ruhr.de/news/freie\\_erp\\_systeme\\_zunehmend\\_attraktiv\\_für\\_kmu](http://www.ec-ruhr.de/news/freie_erp_systeme_zunehmend_attraktiv_für_kmu)  
 Abruf: 30.04.11
- [ERPEin] ERP Einführung  
<http://www.erp-software.org/erp-einfuehrung>  
 Abruf:01.03.11
- [ERPPro] ERP Programme  
<http://erp-software.org/erp-programme>  
 Abruf: 01.03.11
- [ERPSolu] ERP Solutions: System Architecture  
[http://www.weblinedia.com/erp\\_modules\\_solutions\\_tech\\_overview.htm](http://www.weblinedia.com/erp_modules_solutions_tech_overview.htm)  
 Abruf: 15.05.11

- [evalGd] Evaluation Guide: Compiere, Openbravo & Adempiere Service Provider  
[http://www.himanflash.com/support/evaluation\\_guide.html](http://www.himanflash.com/support/evaluation_guide.html)  
 Abruf: 27.04.11
- [Lexi] Enterprise Resource Planning-System: Gabler  
 Wirtschaftslexikon  
<http://wirtschaftslexikon.gabler.de/>  
 Abruf: 20.02.11
- [MAN] [http://pi3.informatik.uni-mannheim.de/downloads/grundstudium/erp/hws07/Vorlesung\\_1\\_ERP\\_WS0708.pdf](http://pi3.informatik.uni-mannheim.de/downloads/grundstudium/erp/hws07/Vorlesung_1_ERP_WS0708.pdf)  
 Abruf:02.03.11
- [Mand] ERP Software Auswahl Glossar  
<http://www.erp-software-auswahl.de/erp-software-auswahl-glossar>  
 Abruf:01.03.11
- [Mig] MECK- Mainfränkisches Electronic Commerce Kompetenzzentrum:  
 ERP für den Mittelstand- Migration und Datenübernahme: Prof. Dr. R.  
 Thome Juli 2009  
[http://www.meck-online.de/wp-content/uploads/2010/01/2009-08\\_migration.pdf](http://www.meck-online.de/wp-content/uploads/2010/01/2009-08_migration.pdf)  
 Abruf: 17.03.11
- [ModVal] ADempiere- ModelValidator  
<http://www.adempiere.com/ModelValidator>  
 Abruf: 25.07.11
- [ModStr] Software Engineering: Modularisierungsstrategien  
 F. Raudszus, FH Wiesbaden  
<http://www.ite.fh-wiesbaden.de/~raudszus/download/Modularisierung.pdf>  
 Abruf: 01.05.11
- [Obravo] Open source web-basiertes ERP-System  
<http://www.openbravo.com/de/>  
 Abruf: 13.04.11
- [Otaps] Open Source ERP + CRM solution  
<http://www.opentaps.org/>  
 Abruf: 20.04.11
- [Penta] Pentaho:  
<http://community.pentaho.com/>  
 Abruf: 02.06.11
- [ProjAd] About Adempiere  
[http://www.adempiere.com/What\\_does\\_ADempiere-mean](http://www.adempiere.com/What_does_ADempiere-mean)  
 Abruf: 15.06.2011

- [Refer] Firmenreferenzen (z.B.):  
<http://www.catura.de/>  
<http://www.schaeffer-ag.de/>  
<http://www.metas.de/>  
<http://www.evenos.de/>  
<http://www.integratio.com/>  
<http://www.objectcode.de/>  
Abruf: 01.08.11
- [Release] ERP Lifecycle Management: Releasefähigkeit schützt Ihre Investitionen  
[http://www.greenax.com/infoglueDeliverLive\\_bison/digitalAssets/7/7778\\_Artikel\\_Releasefaehigkeit.pdf](http://www.greenax.com/infoglueDeliverLive_bison/digitalAssets/7/7778_Artikel_Releasefaehigkeit.pdf)  
Abruf:02.03.11
- [ReLö] ERP-Einkauf: Das Pflichtenheft hat Recht  
<http://www.computerwoche.de/management/compliance-recht/1859905>  
Abruf:02.03.11
- [Res] edu BITE: Auswahl von ERP Systemen  
[http://www.dke.univie.ac.at/wirtschaftsinformatik/history/ss06/ekibis/lo\\_erp\\_technik/M4\\_Lektion3/Auswahl\\_von\\_ERPSystemen/Inhalt/user/eduBITE/Lernobjekte\\_ERP-Standardsoftware/m4\\_lo032\\_AuswahlERP-Systeme/content/medien/3\\_2.pdf](http://www.dke.univie.ac.at/wirtschaftsinformatik/history/ss06/ekibis/lo_erp_technik/M4_Lektion3/Auswahl_von_ERPSystemen/Inhalt/user/eduBITE/Lernobjekte_ERP-Standardsoftware/m4_lo032_AuswahlERP-Systeme/content/medien/3_2.pdf)  
Abruf: 10.04.11
- [Reto] RETO von Arb: Vorgehensweisen und Erfahrungen bei der Einführung von Enterprise-Management-Systemen dargestellt am Beispiel von SAP R/3. 1998.  
[http://www.staempfli.com/digital-publications/html\\_d/idv4.html](http://www.staempfli.com/digital-publications/html_d/idv4.html)  
Abruf: 11.05.11
- [SoftMig] Einleitung zum Thema Softwaremigration  
[http://www.dpunkt.de/leseproben/2898/4\\_Kapitel%201.pdf](http://www.dpunkt.de/leseproben/2898/4_Kapitel%201.pdf)  
Abruf: 17.06.11
- [UmfNEG] Mittelstand greift verstärkt zu freien ERP-Systemen:  
[http://www.silicon.de/technologie/software/0,39044013,41537102,00/mittelstand\\_greift\\_verstaerkt\\_zu\\_freien\\_erp\\_systemen.htm](http://www.silicon.de/technologie/software/0,39044013,41537102,00/mittelstand_greift_verstaerkt_zu_freien_erp_systemen.htm)  
Abruf: 05.05.11

# Abbildungsverzeichnis

Abbildung 2.1: Aufgabenkategorien und Einsatzbereiche von ERP-Systemen [GrAnd] .....	5
Abbildung 2.2: Nutzeffekte von CIM-Komponenten und Integrationskonzepten (Teil 1) [nutzEf].....	7
Abbildung 2.3: Aufbau und Anwendungsbereiche eines ERP-Systems.....	8
Abbildung 2.4: 3-Ebenen-Architektur eines ERP-Systems.....	10
Abbildung 3.1: Logische Darstellung der Client-Server-Architektur von DiPPS .....	15
Abbildung 3.2: Objekt- und Fensterhierarchie der Benutzeroberfläche .....	16
Abbildung 4.1: Diagramm für die Schritte der Datenmigration.....	23
Abbildung 5.1: Darstellung der Drei-Schichten-Architektur.....	28
Abbildung 7.1: Konzept für die Auswahl des OS-ERP-Systems.....	40
Abbildung 7.2: Elemente der Application Dictionary .....	60
Abbildung 7.3: Architektur von ADempiere und die Schichten der Application Dictionary ...	61
Abbildung 7.4: Anpassungskonzept in Application Dictionary .....	64
Abbildung 7.5: Datenmodell für die Anwendung "Workshop".....	68
Abbildung 7.6: Datenbanktabellen für die Anwendung "Workshop" .....	69
Abbildung 9.1: Übersicht über die DiPPs-Module der Dresden Informatik GmbH.....	82
Abbildung 9.2: AD Validation Fenster.....	90
Abbildung 9.3: AD Rule Engine .....	91
Abbildung 9.4: Referenzierte Datenfelder .....	92
Abbildung 9.5: Tabelle f_workshop in Application Dictionary.....	93
Abbildung 9.6: Erstellung des "Workshop"-Fensters.....	93
Abbildung 9.7: Datenfelder des "Workshop"-Fensters.....	94
Abbildung 9.8: "Workshop"-Fenster - Tab Workshop .....	94
Abbildung 9.9: "Workshop"-Fenster - Tab Teilnehmer .....	95
Abbildung 9.10: "Workshop"-Fenster - Tab Thema.....	95
Abbildung 9.11: "Workshop"-Fenster - Tab ausstehende Themen .....	96
Abbildung 9.12: Workshopzustand-Fenster.....	96
Abbildung 9.13: Verweisung des Fensters "Workshopzustand" auf das Fenster "Workshop" (Teil 1).....	97
Abbildung 9.14: Verweisung des Fensters "Workshopzustand" auf das Fenster "Workshop" (Teil 2).....	97
Abbildung 9.15: Verweisung des Fensters "Workshopzustand" auf das Fenster "Workshop" (Teil 3).....	98
Abbildung 9.16: Default Logik für "Datum" im Fenster "Workshop" .....	99
Abbildung 9.17: Anzeige Logik für den Tab "ausstehende Themen" im Fenster "Workshop" .....	99
Abbildung 9.18: Callout in Tab "W Thema" .....	100
Abbildung 9.19: Java Applikation für die Generierung der Model Validatoren (Teil 1) .....	101

<b>Abbildung 9.20: Java Applikation für die Generierung der Model Validatoren (Teil 2) .....</b>	<b>101</b>
<b>Abbildung 9.21: PackOut für die Anwendung "Workshop" (Teil 1) .....</b>	<b>102</b>
<b>Abbildung 9.22: PackOut für die Anwendung "Workshop" (Teil 2) .....</b>	<b>102</b>
<b>Abbildung 9.23: Anpassung (Sprachauswahl bei der Anmeldung) im separaten Projekt ....</b>	<b>103</b>

## Tabellenverzeichnis

Tabelle 2.1: Vergleich OS-ERP-Systeme und Proprietäre Systeme.....	12
Tabelle 6.1: Funktionalitätsvergleich einiger OS-ERP-Systeme .....	37
Tabelle 7.1:Projektschritte .....	42
Tabelle 7.2: Pflichtenheft, Teil 1.....	45
Tabelle 7.3: Pflichtenheft, Teil 2.....	48
Tabelle 7.4: Mögliche Informationsquellen zur Auswahl eines ERP-Systems .....	49
Tabelle 7.5: Selektierte OS-ERP-Systeme auf dem Markt.....	51
Tabelle 7.6: Vergleich der drei ausgewählten Systeme .....	54
Tabelle 7.7: Ergebnis des Vergleichs.....	55
Tabelle 7.8: Technische Daten von ADempiere .....	57
Tabelle 9.1: Verschiedene Lizenzen und ihre Einschränkungen .....	79
Tabelle 9.2: Erwartungen des OS-ERP-System-Einsatzes .....	80
Tabelle 9.3: Erfüllte Erwartungen nach dem OS-ERP-System-Einsatz .....	80
Tabelle 9.4: Ergebnisse der Evaluation von ADempiere.....	83
Tabelle 9.5: Ergebnisse der Evaluation von Opentaps .....	83
Tabelle 9.6: Ergebnisse der Evaluation von Ix-Office .....	84
Tabelle 9.7: Ergebnisse der Evaluation von Openbravo.....	84
Tabelle 9.8: Vergleich der Erfüllungsgrade .....	84
Tabelle 9.9: Detaillierter Vergleich nach Funktionalitäten zwischen ADempiere und Openbravo gegenüber des Pflichtenhefts .....	89

## Abkürzungsverzeichnis

<b>BDE</b>	Borland Database Engine
<b>BWL</b>	Betriebswirtschaftslehre
<b>CIM</b>	computer Integrated Manufacturing
<b>CMS</b>	Chain Management System
<b>CRM</b>	Customer Relationship Management
<b>DCOM</b>	Distributed Component Object Model
<b>DiPPS</b>	Dresden Informatik Produktionsplanung und – Steuerung
<b>DV</b>	Datenverarbeitung
<b>ERP</b>	Enterprise Resource Planning
<b>ETL</b>	Extraction Transformation load
<b>FSF</b>	Free Software Foundation
<b>GNU FDL</b>	Free Documentation License
<b>GNU-GPL</b>	General Public License
<b>GNU LGPL</b>	Lesser General Public License
<b>GPRS</b>	General Packet Radio Service
<b>GUI</b>	Graphical User Interface
<b>KMU</b>	kleine und mittlere Unternehmen
<b>MDI</b>	Multi Document Interface
<b>MIDAS</b>	Multi- tier Distributed Application Services Suite
<b>NEG</b>	Netzwerk Elektronischer Geschäftsverkehr
<b>OS</b>	Open Source
<b>PPS</b>	Produktionsplanung und – Steuerung
<b>RDBMS</b>	relationale Datenbankmanagementsystem
<b>SMS</b>	Short Message Service
<b>SQL</b>	Structured Query Language
<b>SSIS</b>	SQL Server Integration Services
<b>XML</b>	Extensible Markup Language

# Glossar

**DCOM:** ist ein objektorientiertes RPC-System, das auf dem DCE-Standard basiert.

**Inkonsistenz:** Widersprüche oder Fehlerhaftigkeiten der Daten

**Model Validator:** ist ein wesentlicher Teil in Adempiere. Es ist Java Klasse, die `org.compiere.model.ModelValidator` im Adempiere Projekt implementiert. Sie enthält zwei wichtige (Callback) Methoden, `modelChange` und `docValidate`, die bei der Änderung des Modells und Dokuments aufgerufen werden

**Redundanz:** Mehrfachspeicherung der Daten

**Spreadsheet:** (deutsch Tabellenkalkulation): Eine Software für die Verarbeitung der numerischen und alphanumerischen Daten in Tabellenform

**Stored Procedures:** unter einem Namen im Datenbankserver gespeicherte Abläufe und Anweisungen in einem DBMS.

**Thin- Client:** Endgerät eines Netzwerkes

**Trigger:** Funktionalität in DBMS. Dadurch werden die Daten, die nur ‚statisches‘ Wissen darstellen können, um Abläufe aus der realen Welt, also dynamisches Verhalten erweitert.