



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

André Schmer

Weiterentwicklung, Evaluierung und Einführung
eines prototypischen Open Source
Vertriebssystems

André Schmer

Weiterentwicklung, Evaluierung und Einführung
eines prototypischen Open Source Vertriebssystems

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. rer. nat. Jörg Raasch
Zweitgutachter: Prof. Dr. Olaf Zukunft

Kooperation mit Firma: novomind AG

Abgegeben am 26. August 2008

André Schmer

Thema der Bachelorarbeit

Weiterentwicklung, Evaluierung und Einführung eines prototypischen Open Source Vertriebssystems

Stichworte

EJB, JSF, JBoss, Customer Relationship Management, Vertriebssystem, Evaluierung

Kurzzusammenfassung

Diese Arbeit beschäftigt sich mit der Überführung eines Explorativen Prototypen eines Open Source Vertriebssystems in einen Evolutionären Prototypen. Ziel ist es, unternehmensrelevante Funktionen und Module zur Verfügung zu stellen, um eine Einführung des Systems unter Gewährleistung qualitativer Ansprüche mit Hilfe von Methoden zur Evaluierung zu ermöglichen.

André Schmer

Title of the paper

Further development, evaluation and implementation of a prototypical open source distribution system

Keywords

EJB, JSF, JBoss, Customer Relationship Management, Distribution system, Evaluation

Abstract

The context of this thesis is to be engaged to advance an explorative prototype of an open source customer relationship management system to an evolutionary prototype with the goal to allocate relevant functions and moduls to take care of the adoption of the system with qualitative claim with the cooperation some methods to evaluate.

Danksagung

Mein herzlichsten Dank gilt Herrn Prof. Dr. Jörg Raasch für die hervorragende und intensive Betreuung und Begleitung bei dieser Arbeit.

Ebenso danke ich Herrn Prof. Dr. Olaf Zukunft für die Betreuung als Zweitgutachter.

Ein ganz besonderer Dank geht an Herrn Peter Samuelsen von der novomind AG, der mir erst dieses spannende Thema ermöglicht hat und niemals müde wurde, sämtliche Iterationsschritte intensivst und gemeinsam zu reflektieren.

Nicht zu vergessen natürlich Carola Lilienthal. Sie hat mir mit ihrem umfangreichen Fachwissen über Softwarearchitekturen wertvolle Tipps und Hilfestellung für die Evaluation des Projekts gegeben.

Außerdem danke ich Dennis Winter, Markus Wernicke, Romana Bannuscher, Fahim Aleaf, Dennis Roehrs, Jonas Clusen, Mahmut Cevik und natürlich Boris Klengel und Florian Jüttner, die mir sowohl mit Rat und Tat als auch als Versuchspersonen bei der Durchführung des Usability-Tests zur Seite standen. Allerbesten dank an Carsten German, der etwas seiner kostbaren Zeit erübrigen konnte, die restlichen Fehler aufzuspüren.

Martin Sadowski und Fahim Aleaf: danke für die gute Zeit!

André Schmer, August 2008

Inhaltsverzeichnis

Abbildungsverzeichnis	7
1 Einleitung	8
1.1 Motivation	8
1.2 Umfeld	8
1.3 Problemstellung und Zielsetzung	9
1.4 Aufbau der Arbeit	9
2 Grundlagen	11
2.1 Customer Relationship Management	11
2.2 JavaServer Faces	11
2.3 EJB 3.0	12
2.3.1 Eingesetzte EJB-Komponenten	13
2.4 JBoss	13
3 Analyse der Spezifikation	14
3.1 Funktionale Anforderungen	14
3.2 Nichtfunktionale Anforderungen	16
4 Vision	18
4.1 Kurzfristiges Ziel	18
4.2 Zukünftiges Ziel	19
5 Architektur	20
5.1 Fachliche Architektur	20
5.1.1 Datenmodell	21
5.1.2 Generisches Datenmodell	23
5.1.3 Sichtbarkeitskonzept	30
5.1.4 Rollen-/Rechtesystem	31
5.1.5 Fazit	32
5.2 Technische Architektur	33
6 Realisierung	35

6.1	Vorgehensweise	35
6.1.1	Ziele	35
6.1.2	Proof of Concept	35
6.1.3	Evolutionäres Prototyping	37
6.2	Aufteilung in Module	37
6.3	Contact-Modul	40
6.4	Account-Modul	44
6.5	Activity-Modul	49
6.6	Opportunity-Modul	53
6.7	Administrations-Modul	57
7	Evaluierung	64
7.1	Vorgehensweise	64
7.2	Architekturanalyse	64
7.3	Usability-Test	67
7.3.1	Ergebnis des Usability-Tests	69
7.4	Fazit	70
8	Methodische Abstraktion	72
9	Zusammenfassung	73
9.1	Stand	73
9.2	Erfahrungen	74
9.3	Ausblick	75
A	Aufgaben des Usability-Tests	76
B	Auswertung der Fragebögen	80
	Glossar	87
	Abkürzungsverzeichnis	91
	Literaturverzeichnis	92

Abbildungsverzeichnis

5.1	Datenmodell der Benutzer-Struktur	22
5.2	Datenmodell der Contact-Struktur	24
5.3	Datenmodell der Account-Struktur	25
5.4	Tabellen des Generischen Datenmodells	27
5.5	Generische Felder im Einsatz am Beispiel eines Account	30
5.6	Businessprofil	31
5.7	Schichtenmodell - schematisch	33
6.1	Autor-Kritiker-Zyklus (Züllighoven)	36
6.2	Entwicklungsprozess mit Prototypen (vgl. Pomberger und Weinreich, 1994)	38
6.3	Contact - Detailansicht	42
6.4	Contact - Editier-Maske	43
6.5	Account - Detailansicht	47
6.6	Account - Editier-Maske	48
6.7	Activity - Detailansicht	51
6.8	Activity - Editier-Maske	52
6.9	Opportunity - Detailansicht	55
6.10	Opportunity - Editier-Maske	56
6.11	Benutzerverwaltung	58
6.12	Rollenverwaltung	59
6.13	Businessprofilverwaltung	60
6.14	Administration eines Generischen Feldes	62
6.15	Meine Daten	63
7.1	Drei-Schichten der Architektur	66

1 Einleitung

1.1 Motivation

Ein marktorientiertes Unternehmen erkennt sehr schnell das Potenzial, das durch Wissen um seine Kunden und deren Bedürfnisse, durch das Wissen über seine Mitbewerber und durch das Wissen über den Markt, entsteht. Dieses Wissen ist ein überlebenswichtiges Kapital im Beziehungsmanagement zwischen Kunden und Lieferanten. Um so wichtiger ist es, dieses Wissen transparent gestalten und klassifizieren zu können. Dafür ist es notwendig, geeignete Werkzeuge zur Verfügung zu stellen, die abteilungsübergreifend Zugriff auf diese Informationen ermöglichen und so die Kommunikation nach Innen und nach Außen optimieren.

Diese Arbeit entstand im Rahmen der Anforderung der novomind AG, ein bestehendes proprietäres Vertriebssystem durch ein neues webbasiertes Vertriebssystem abzulösen und ist die zweite von drei Arbeiten, die sich mit diesem Vertriebssystem von der Analyse über die Entwicklung bis hin zu speziellen Modulen für die gezielte Datenauswertung, beschäftigen. Als solche ist sie für die Implementierung der Kernfunktionalitäten bis zur Erreichung der Produktreife und somit Einführung in die Unternehmensstruktur verantwortlich.

Des Weiteren will die novomind AG als aktives Mitglied der Open Source Community Erfahrungen sammeln, mit den Anforderungen, ein Open Source Vertriebssystem zu entwickeln und dieses kostenlos zur Verfügung zu stellen.

1.2 Umfeld

Diese Arbeit entstand im Auftrag der novomind AG und wurde durch das Unternehmen beaufsichtigt. Die novomind AG ist ein mittelständisches Unternehmen mit Sitz in Hamburg und gilt als einer der führenden Softwareanbieter für die prozess- und kostenoptimierende Kundenkommunikation. Die Anfertigung dieser Arbeit fand in den Räumen der novomind AG und in direkter Zusammenarbeit mit der Geschäftsführung und den Mitarbeitern des Unternehmens statt.

1.3 Problemstellung und Zielsetzung

Wie bereits erwähnt, soll ein bestehendes Vertriebssystem abgelöst werden, da es zum einen nicht mehr den heutigen technologischen Anforderungen entspricht, und zum anderen auch nicht mehr in die Vertriebsorganisation der novomind AG passt.

Die Herausforderung bestand darin, eine serverbasierte Webanwendung zu entwickeln, die ein bestehendes proprietäres Vertriebssystem, das ausschließlich als Rich Client Anwendung konzipiert und eingesetzt wurde, abzulösen und den Funktionsumfang zu übernehmen. Ebenso ist es notwendig, das Datenmodell neu zu entwerfen und die bestehenden Daten aus dem Altsystem in vollem Umfang zu übernehmen. Da die Altdaten und deren Formate so nicht mehr den neuen Anforderungen entsprachen, mussten diese an das neue Datenmodell angepasst und verlustfrei übernommen werden. Es mussten also Strategien entwickelt werden, um die Altdaten korrekt zu übernehmen und diese an die neuen Anforderungen anzupassen. Außerdem hatte das abzulösende System trotz seiner Schwächen eine recht hohe Akzeptanz bei den Anwendern. Diese Akzeptanz auch für das neue System zu erreichen und vielleicht sogar zu übertreffen, war eine weitere Herausforderung, die es zu meistern galt.

Diese Arbeit wird mit einem Evolutionären Prototypen, der bis zur Marktreife entwickelt wird, abgeschlossen. Danach folgt die Einführung im Unternehmen der novomind AG und die endgültige Ablösung des Altsystems.

1.4 Aufbau der Arbeit

In dieser Arbeit wird die Entwicklung einer serverbasierten Webanwendung unter Verwendung moderner Technologien behandelt. In Kapitel [1] wurde auf die Motivation und die Zielsetzung für diese Arbeit eingegangen.

Im folgenden Kapitel [2] werden dem Leser einige Grundlagen näher gebracht, die für das Verständnis dieser Arbeit von Bedeutung sind. Es wird kurz auf die Bedeutung eines Vertriebssystems und auf den Nutzen für ein Unternehmen eingegangen und es werden die wichtigsten Technologie Standards wie EJB 3.0, JSF und JBoss kurz erläutert.

In Kapitel [3] werden die Anforderungen an das zu entwickelnde System diskutiert, die vom Unternehmen vorgegeben wurden. Es enthält eine Analyse der Spezifikationen, unterteilt in funktionale und nichtfunktionale Anforderungen, die im Kapitel ausführlich beschrieben und diskutiert werden.

Auf die Ziele wird in Kapitel [4] ausführlich eingegangen. So werden kurzfristige Ziele definiert, die mit dem Abschluss dieser Arbeit zu erreichen sind und es werden zukünftige Ziele definiert, um den Weg für das Vertriebssystem zu ebnet.

Die Architektur wird in Kapitel [5] behandelt. Hierbei wird die Architektur zum einen aus der fachlichen Sicht erläutert, was das Datenmodell, die Idee und Umsetzung des Generischen Datenmodells beinhaltet, sowie das Sichtbarkeitskonzept für das Generische Datenmodell und das Rollen- und Rechtekonzept. Zum anderen wird die Architektur aus der technischen Sicht betrachtet.

Die Realisierung wird ausführlich in Kapitel [6] geschildert. Skizziert werden dabei die Ziele einer Vorgehensweise und dies wird mit einem Proof of Concept das Prototyping für diese Arbeit einleiten. Ebenso wird das Evolutionäre Prototyping erläutert, sowie die Module des Systems im Einzelnen.

Das Thema der Evaluierung wird in Kapitel [7] behandelt. Hier wird unterscheiden zwischen einer Architekturanalyse und einem Usability-Test, mit dem die Akzeptanz der Anwender an diesem System nachgewiesen werden soll.

Im Kapitel [8], methodische Abstraktion, werden die Erfahrungen dieser Arbeit zu einem Leitfaden ausgearbeitet.

Kapitel [9] fasst die Ergebnisse zusammen und reflektiert die Arbeit. Mit einem Ausblick auf Erweiterungen sollen Möglichkeiten der Erweiterungen des Vertriebssystems erläutert werden.

2 Grundlagen

2.1 Customer Relationship Management

Zunächst soll hier bewusst ein Begriffswandel von *Vertriebssystem* zu *Customer Relationship Management (CRM)* vollzogen werden. Der Grund wird deutlich, wenn man den Begriff Vertriebssystem der deutschen Übersetzung von CRM, Kunden-Beziehungs Marketing (oder Management), gegenüberstellt. Während der Begriff Vertriebssystem im betriebswirtschaftlichen Umfeld als ein Element der Vertriebspolitik eines Unternehmens betrachtet wird (vgl. Winkelmann, 2003, S.32), so bezeichnet der Begriff des CRM ein integriertes Kundenmanagement, das eindeutig Bestandteil dieser Arbeit ist.

Customer Relationship Management (CRM) ...

... ist ein ganzheitlicher Ansatz zur Unternehmensführung. Er integriert und optimiert auf der Grundlage einer Datenbank und Software zur Marktbearbeitung sowie definierter Verkaufsprozesse abteilungsübergreifend alle kundenbezogenen Prozesse in Marketing, Vertrieb, Kundendienst, F&E, u.a. Zielsetzung von CRM ist die gemeinsame Schaffung von Mehrwerten auf Kunden- und Lieferantenseite über die Lebenszyklen von Geschäftsbeziehungen. Das setzt voraus, dass CRM-Konzepte Vorkehrungen zur permanenten Verbesserung der Kundenprozesse und für ein berufslebenslanges Lernen der Mitarbeiter enthalten (vgl. Winkelmann, 2003, S.186).

CRM unterstützt die Kommunikation mit dem Kunden, mit verlässlichen Daten, Zahlen, Fakten, es hilft beim Sammeln von Informationen und klassifizieren. Zielsetzung von CRM ist dabei die Schaffung von Mehrwerten auf Kunden- und Lieferantenseite im Rahmen von Geschäftsbeziehungen. Es ist so ein ganzheitlicher Ansatz zur Unternehmensführung.

2.2 JavaServer Faces

JavaServer Faces (JSF) sind ein standardisiertes Framework zur Entwicklung von Benutzeroberflächen für Webapplikationen. Mit JSF kann der Entwickler auf einfache Art die Komponenten einer HTML- oder XHTML- Seite zusammenbauen. Es basiert auf der JSP

Model 2 Architektur und beruht auf den Erfahrungen, die mit Struts gemacht wurden. Seit Java EE 5 ist es Bestandteil der J2EE-Spezifikation.

Mit JavaServer Faces wird es dem Entwickler leicht gemacht, sich an die Model 2 Konventionen zu halten, also die Präsentationsschicht von der Logik zu trennen.

Manche Webapplikationen basieren auf mehreren Formularseiten, die auszuwerten sind. Das zustandslose HTTP-Protokoll bietet keine Möglichkeit, die Anwenderdaten zu persistieren; JSF unterstützt diese und andere Eigenschaften.

JSF arbeiten mit den so genannten *Backing Beans* oder auch *Managed Beans* zusammen. Das sind Java-Klassen, die *Properties* besitzen und auf *Events* reagieren können. Eine Property ist eine Variable, die durch einfache Setter- und Getter Methoden von JSF gelesen und/oder geschrieben werden kann (vgl. Geary und Horstmann, 2004).

Eine Auflistung der Merkmale von JSF:

- Verwaltung der Zustände über mehrere Seiten über einen Komponentenbaum
- Browser-spezifische Generierung von HTML
- Server seitige Formularüberprüfung durch Standard-Validatoren
- Einheitliche Ereignisbehandlung durch swingähnliche Events
- Fehlerbehandlung durch Messages
- Vorhandene UI- Komponenten, die erweiterbar sind
- Internationalisierung durch Ressource Bundles

2.3 EJB 3.0

In der Konzeptionsphase für dieses Projekt wurde bereits von Martin Sadowski (Sadowski, 2008) beschrieben, dass das neue Vertriebssystem über eine Mehrschichten-Architektur verfügen soll. Um das von JSF [2.2] geforderte MVC- bzw. Model 2 Muster einzuhalten, ist es angebracht ein Komponentenmodell einzusetzen. Enterprise Java Beans (EJB) 3.0 gilt als komponentenorientiertes Programmiermodell. Da es außerdem Bestandteil der J2EE/JEE Architektur ist (vgl. Sun Microsystems Inc., 2008b), bietet es sich hier in besonderem Masse an (vgl. Ihns u. a., 2007).

Mit EJB lässt sich die Entwicklung mehrschichtiger Softwarearchitekturen stark vereinfachen.

2.3.1 Eingesetzte EJB-Komponenten

Persistence Entities kommen zum Einsatz, um die fachlichen Objekte des Systems abzubilden. Ein fachliches Objekt repräsentiert eine Tabelle der Datenbank, wie z.B. eine Tabelle *Kunde*. Persistence Entities werden als Plain Old Java Object (POJO) implementiert, ein einfaches Java-Objekt, das keine weiteren Abhängigkeiten zu anderen Objekten besitzt. Die Persistence Entities werden mit Hilfe des Hibernate-Dienstes, einem Mechanismus zum objektrelationalen Mapping, des Application Servers dauerhaft gespeichert. Eine Instanz eines Objektes dieser Klasse entspricht einer Zeile in der entsprechenden Datenbank-Tabelle.

Des Weiteren werden Session Beans eingesetzt, um die Geschäftsobjekte abzubilden. Diese besondere Art der Objekte sind in der Geschäftslogik angesiedelt, die als eigenständige Schicht in der Architektur repräsentiert wird (siehe Kapitel 5). Hier finden die Vorgänge statt, die ein Benutzer mit dem System durchführt, wie z.B. die Persistenz der Entities sicherzustellen.

Session Beans werden ausschliesslich über Interfaces angesprochen. Das bietet den Vorteil, dass sie leicht innerhalb eines Systems verteilt werden können. Mit Hilfe eines Namensdienstes können die Session Beans sehr komfortabel angesprochen werden. So ist es auch möglich diese Schicht z.B. als Webservice zur Verfügung zu stellen.

2.4 JBoss

JBoss ist eine Implementierung eines Application Servers. Er ist Teil des J2EE-Standards und unterliegt der LPGL-Lizenz als freie Software und erfüllt somit die Anforderungen der novomind AG nur Open Source-Komponenten zu verwenden. JBoss ist vollständig in Java implementiert und dient den EJB [2.3] als Laufzeitumgebung. In ihm enthalten ist ein Webserver und er stellt verschiedene Dienste bereit, wie z.B. JAAS, einem Authentifizierungsdienst und einem Hibernate-Modul, mit dem es möglich wird, die Persistence Entities dauerhaft zu speichern (vgl. Red Hat Inc.).

3 Analyse der Spezifikation

Bevor mit der Implementierung begonnen werden kann, müssen die Anforderungen an das zukünftige System analysiert werden. In diesem Kapitel werden die Vorgaben der novomind AG für die Anforderungen an das Vertriebssystem untersucht. Die Basis für die Analyse liefert die erste Arbeit in diesem Kontext, die sich mit der Konzeption des Vertriebssystems befasst. Durch eine Analyse des Altsystems werden die Anforderungen an das neue System festgestellt und somit der Weg für eine Neuentwicklung geebnet (Sadowski, 2008). Es werden hier im Folgenden die funktionalen und nichtfunktionalen Anforderungen, die Martin Sadowski in seiner Arbeit identifiziert hat, aufgenommen, diskutiert und ggf. erweitert.

Die grundlegendste Anforderungen der novomind AG war, das Vertriebssystem mit Java zu realisieren und ausschliesslich solche Komponenten zu verwenden, die unter einer freien Software Lizenz angeboten werden.

3.1 Funktionale Anforderungen

Verwalten von Accounts

Das System soll dem Benutzer die Möglichkeit geben, *Accounts* (Unternehmen) zu verwalten. Dazu gehören das Anlegen und das Bearbeiten von Accounts. Außerdem soll es möglich sein, über die Accountverwaltung *Activities* (Aktivitäten) und *Opportunities* (Geschäftsgelegenheiten) anzulegen, sowie Beziehungen zu anderen Accounts zu hinterlegen. Für einen Account soll es möglich sein, beliebig viele Adressen zu hinterlegen, die von einem bestimmten Adresstyp (Main, Private, Billingaddress, POB, Other) sind. Für die Accounts soll es möglich sein, benutzerdefinierte Felder anzulegen, diese zu gruppieren (Cluster), zu verwalten und mit Werten vorzubelegen. Es soll außerdem eine Basis-Suche für Accounts geben, mit der man nach Accounts suchen und diese z.B. nach Inhaber auflisten kann.

Verwalten von Contacts

Das System soll dem Benutzer die Möglichkeit geben, *Contacts* (Kontaktpersonen) zu verwalten. Zur Verwaltung gehören das Anlegen, Bearbeiten und das Löschen von *Contacts*. *Contacts* müssen einem Account zugeordnet werden, ebenso soll es möglich sein,

einen Contact umzuhängen, d.h. ihn nach einem anderen Account zu verschieben, ohne dass die Daten des Contacts beeinträchtigt werden. Einem Contact sollen auch direkt Activities zugeordnet werden können. Ebenso wie für die Accounts sollen für die Contacts benutzerdefinierte Felder angelegt und verwaltet werden, sowie mit Werten vorbelegt werden können. Diese Felder sollen ebenfalls in benutzerdefinierte Cluster gruppiert werden können. Contacts haben ebenso eine Basis-Suche, in der man nach Contacts suchen kann.

Verwalten von Activities

Das System soll die Möglichkeit bieten, *Activities* (Aktivitäten) zu hinterlegen und zu verwalten. Dazu gehören das Erfassen einer Activity, das Bearbeiten und das Löschen. Eine Activity wird einem Account, einem Contact oder einer Opportunity zugeordnet. Eine Activity soll einen *Activity Type* (Aktivitätstyp) haben und ein oder mehrere *Buying Interests* (Kaufinteresse). Zu einer Activity soll es möglich sein, eine *Follow Activity* (Folgeaktivität) anzulegen. Eine Basis-Suche stellt die Filterfunktion für die Activities dar.

Verwalten von Opportunities

Das System soll dem Benutzer die Möglichkeit geben, *Opportunities* (Geschäftsgelegenheiten) zu verwalten. Dazu gehören das Anlegen, Bearbeiten und das Löschen von Opportunities. Opportunities werden einem Account zugeordnet und sind ggf. einem *Buying Center* (Einkaufsstelle) zugeordnet. Eine Basis-Suche ermöglicht das Suchen nach Opportunities.

Verwalten von Businessprofilen

Das System soll einem Administrator Funktionen zur Verfügung stellen, mit denen es möglich ist, *Businessprofile* zu hinterlegen. Ein Businessprofil besteht aus einer *Group* (Unternehmensgruppe), einem oder mehrere *Companies* (Unternehmen), sowie die Unterteilung der Companies in *Business Units* (Geschäftsbereiche). Ein Unternehmen hat zudem mehrere *Lokationen* (Locations). Die Businessprofile sollen angelegt, bearbeitet und gelöscht werden können.

Verwalten von Benutzern

Das System soll einem Administrator die Verwaltung von Benutzern ermöglichen. Dazu gehören das Anlegen, Bearbeiten und Löschen von Benutzern, sowie die Vergabe von *Rechten* bzw. *Rollen* und damit die Vergabe von Zugangsrechten zu dem Vertriebssystem. Benutzer werden dem zugrunde liegenden Businessprofil zugeordnet, sind also in einer Group, Company und Business Unit angesiedelt.

Rollen-/Rechteverwaltung

Das System soll einem Administrator die Möglichkeit geben, Rollen anzulegen, zu bearbeiten und zu löschen. Hierbei sind die Rollen, eine sinnvolle Sammlung von im System verteilten Rechten, mit denen der Zugang geregelt wird. Diese Rollen sollen in der Benutzerverwaltung den Benutzern zugeteilt werden.

Mandantenfähigkeit

Das System soll mandantenfähig sein. Das bedeutet in erster Linie, dass mit einer Installation des Systems es möglich sein soll, die Daten eines Mandanten von anderen Mandanten zu kapseln und somit zu verbergen. In einer nächsten Ausbaustufe soll es möglich sein, die Mandanten zu verwalten.

3.2 Nichtfunktionale Anforderungen

Aussehen und Handhabung

In der ersten Ausbaustufe, also bis zur Einführung bei der novomind AG, soll das Vertriebssystem an den „Stil“ der novomind AG angepasst werden. Das bedeutet, dass weitestgehend die Corporate Identity erhalten bleibt und sich im Design widerspiegelt. Im späteren Verlauf, wenn das System für eine Veröffentlichung in den Open Source Sektor vorbereitet wird, sollte das System mit einem anpassbaren Design versehen werden, mit dem ein potentieller Interessent seine eigene Designmarke in das Vertriebssystem integrieren kann. Die Handhabung sollte so gestaltet sein, dass gleiche Funktionen gleich aufgebaut sind. Somit soll eine Konsistenz der Benutzerführung erreicht werden, die es dem Benutzer erleichtert, sich im System zurecht zu finden.

Performance

Das System soll etwa 10 Benutzern gleichzeitig Zugriff ermöglichen.

Das bedeutet, dass die Antwortzeiten der Kommunikation zwischen einem Benutzer und dem System in akzeptabler Zeit erfolgen soll.

Portierbarkeit

Das System soll unabhängig bzgl. der Betriebssystemplattform und der Hardware sein. Außerdem soll das System leicht auf andere Systemumgebungen portierbar sein und die darunter liegende Datenbank mit überschaubarem Arbeitsaufwand austauschbar sein. Da das System als Webanwendung ausgelegt ist und der Benutzer das System ausschliesslich über ein Web Browser bedient, soll die Wahl des Web Browsers beim Anwender liegen.

Das System muss also browserunabhängig bedienbar sein.

Vertraulichkeit

Die Vertraulichkeit der Daten ist ein KO-Kriterium und verdient besondere Aufmerksamkeit. Die Daten müssen deshalb vertraulich behandelt werden. Die Daten des Systems sollen nur authentifizierten und autorisierten Benutzern zugänglich gemacht werden. Ebenso sollen die Daten der Mandanten gekapselt werden und ein Zugang anderer Mandanten nicht gestattet werden, nach Außen Vertraulichkeit der Daten bzgl. der Kunden- und Unternehmensinformationen. Nach Innen, Sichtbarkeit der Daten abhängig vom Rollen-/Rechtesystem.

Erwartungskonformität

Suggestierte Funktionalität muss auch erfüllt werden. Beispiel: Beim Löschen eines Datensatzes soll dieser und nur dieser gelöscht werden. Andere, nicht unmittelbar damit zusammenhängende „Dinge“, sollten außen vor bleiben.

4 Vision

Bei der Vision geht es darum, Ziele zu definieren, die kurzfristig erreicht werden müssen, um das System erfolgreich einzuführen und in das Unternehmensumfeld der novomind AG zu integrieren. Außerdem sollten Ziele definiert werden, die über die Einführung und das Ziel dieser Arbeit hinaus gehen.

4.1 Kurzfristiges Ziel

Das kurzfristige Ziel kann klar definiert werden, denn die Grundlagen dafür wurden in der Analysephase (Sadowski, 2008) bereits geschaffen. Oberstes Ziel ist die erfolgreiche Einführung des Vertriebssystems in die Vertriebsstruktur der novomind AG. Da das Altsystem zur Einführung des neuen Systems abgeschaltet werden soll, ist es zwingend erforderlich den geforderten Funktionsumfang gemäß der Anforderungsspezifikation [3] sicherzustellen, mit besonderem Fokus auf die Sicherheit, was sowohl die Datensicherheit als auch den Datenschutz angeht. Ausserdem ist der Fokus auf eine deutliche Verbesserung der Bedienbarkeit auszuweiten. Dies kann sichergestellt werden, in dem die zukünftigen Anwender aktiv in den Entwicklungsprozess mit eingebunden werden und der Prozess so in einen Autor-Kritiker-Zyklus (Abb. 6.1) eintritt.

Darauf wird in Kapitel [6] „Realisierung“ ausführlicher eingegangen. Des Weiteren muss gewährleistet werden, dass die Daten aus dem Altsystem erfolgreich und verlustfrei in das neue System übernommen werden, damit ein reibungsloser Übergang stattfinden kann. Der Arbeitsablauf im Unternehmen darf nicht unterbrochen oder gar unmöglich gemacht werden.

Im Einzelnen kann man folgende Hauptziele für die Einführung definieren:

- Sicherstellung der zugesicherten funktionalen und nichtfunktionalen Anforderungen
- Sicherstellung der Korrektheit und Vollständigkeit der Daten
- Sicherstellung der Vertraulichkeit der Daten durch Einführung eines Rechte-/Rollensystems

4.2 Zukünftiges Ziel

Das zukünftige Ziel geht nicht nur über die Einführung des Systems hinaus, es beschreibt auch die Visionen, die mit diesem erreicht werden könnten. So ist es nicht zuletzt auch ein Abbild der Unternehmenspsychologie. Es enthält unter Umständen Zustände, die erreicht werden sollen, um das Unternehmen standhaft zu machen. So ist es nur nahe liegend, das System als eine fest integrierte Unternehmensanwendung kontinuierlich weiter zu entwickeln, um es so auf neue Herausforderungen und Vorgaben anzupassen.

Das System als Open Source Vertriebssystem zur Verfügung zu stellen und so einer Gemeinde von Interessenten zugänglichen zu machen, ist einer der nächsten Schritte. In einer eventuellen gemeinsamen Plattform zum Austausch von Erfahrungen und Erkenntnissen verbirgt sich das Potenzial, diese Software konkurrenzfähig zu machen, Entwicklungen der Gemeinde einfließen zu lassen und langfristig Schnittstellen zu anderen Unternehmensanwendungen zu schaffen.

5 Architektur

Es gibt keine einheitliche Definition von Software-Architekturen. Die folgende Definition, wie sie im Standard 1471 der IEEE beschrieben ist, soll hier als Anhaltspunkt dienen.

architecture - *The fundamental organization of a system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution* (vgl. IEEE, 2000).

Eine Softwarearchitektur besteht demnach aus Komponenten und ihren Beziehungen in verschiedenen Strukturen, so wie ein Gebäude eine tragende Struktur, ein Rohrleitungs- und ein Elektronetz enthält (vgl. Ludewig und Lichter, 2007).

5.1 Fachliche Architektur

Die fachliche Architektur wurde bereits in der Konzeptionsphase dieses Systems von Martin Sadowski (Sadowski, 2008) ausgearbeitet und spezifiziert. Hier sollen kurz noch einmal die wesentlichen Merkmale, die eine Umsetzung im vorliegenden Projekt fanden, zusammengefasst werden.

Die Komponenten aus denen sich das Vertriebssystem zusammen setzt können einfach aufgeteilt werden.

Authentifikation und Autorisation

Diese Komponente ist dafür verantwortlich, den Benutzern Zugang zum System zu gestatten und ihnen gemäß der Rollen- und Rechteverteilung, eine entsprechende Sicht auf die Daten zu ermöglichen, sowie den Zugang zu notwendigen Funktionen bereitzustellen.

Modulkomponenten

Hinter dem Begriff der Modulkomponenten verbergen sich die Module zu den *Accounts*, *Contacts*, *Activities* und *Opportunities*. Dies sind die Bereiche des Systems, in denen die eigentliche Arbeit mit dem System stattfindet, und in denen die Zugriffe auf die einzelnen

Objekte des Systems realisiert sind. Die Module im Einzelnen werden im Kapitel [6] „Realisierung“ vorgestellt.

Administration

Das System ist so ausgelegt, dass einzelne Komponenten des Systems administriert werden können. Dazu gehören die Benutzerverwaltung, die Rollenverwaltung, die Steuerung der Sichtbarkeit der Generischen Felder und der Cluster, sowie die Verwaltung des Generischen Datenmodells, des Businessprofils und ein Bereich für die eigenen Daten des am System angemeldeten Benutzers. Zum Administrations-Modul wird noch genauer im Kapitel [6] „Realisierung“ eingegangen.

Datenhaltung

Diese Komponente ist verantwortlich für die dauerhafte Speicherung der Daten und wird über ein Relationales Datenbank-Management-System realisiert.

5.1.1 Datenmodell

Das Konzepts des Datenmodells setzt ein relationales Datenbank-System voraus. In diesem Projekt wurde eine MySQL-Datenbank eingesetzt, da diese als Open Source-Version den Anforderungen vollkommen genügt.

Im Folgenden wird das Datenbank-Schema in Form von Entity-Relationship-Modellen erläutert. Wegen der Komplexität des Datenmodells, werden die Kernsegmente aufgeteilt in ein Modell für die Benutzer-Struktur, das die Beziehungen der Benutzertabelle zu den Mitarbeiterinformationen darstellt, sowie die Eingliederung in die Unternehmensstruktur.

Als weitere Kernsegmente wird ein Modell für die Contact-Struktur, sowie ein Modell für die Account-Struktur dargestellt. Beide sind ähnlich aufgebaut. In diesen beiden Modellen kann man die Beziehungen zu den anderen beiden wichtigen Komponenten des Vertriebssystems, die Opportunities und die Activities erkennen, so dass auf eine explizite Darstellung deren Strukturen verzichtet wird.

Abbildung 5.1 zeigt das Modell der Benutzer-Struktur. Man kann erkennen, dass ein Benutzer (*user*) ein Mitarbeiter (*employee*) ist und das der wiederum einem Mandanten (*tenant*) zugeordnet ist. So wird sichergestellt, dass das System mandantenfähig ist, denn letztendlich wird jeglicher Zugriff auf das Datenmodell über den Benutzer bzw. Mitarbeiter durchgeführt.

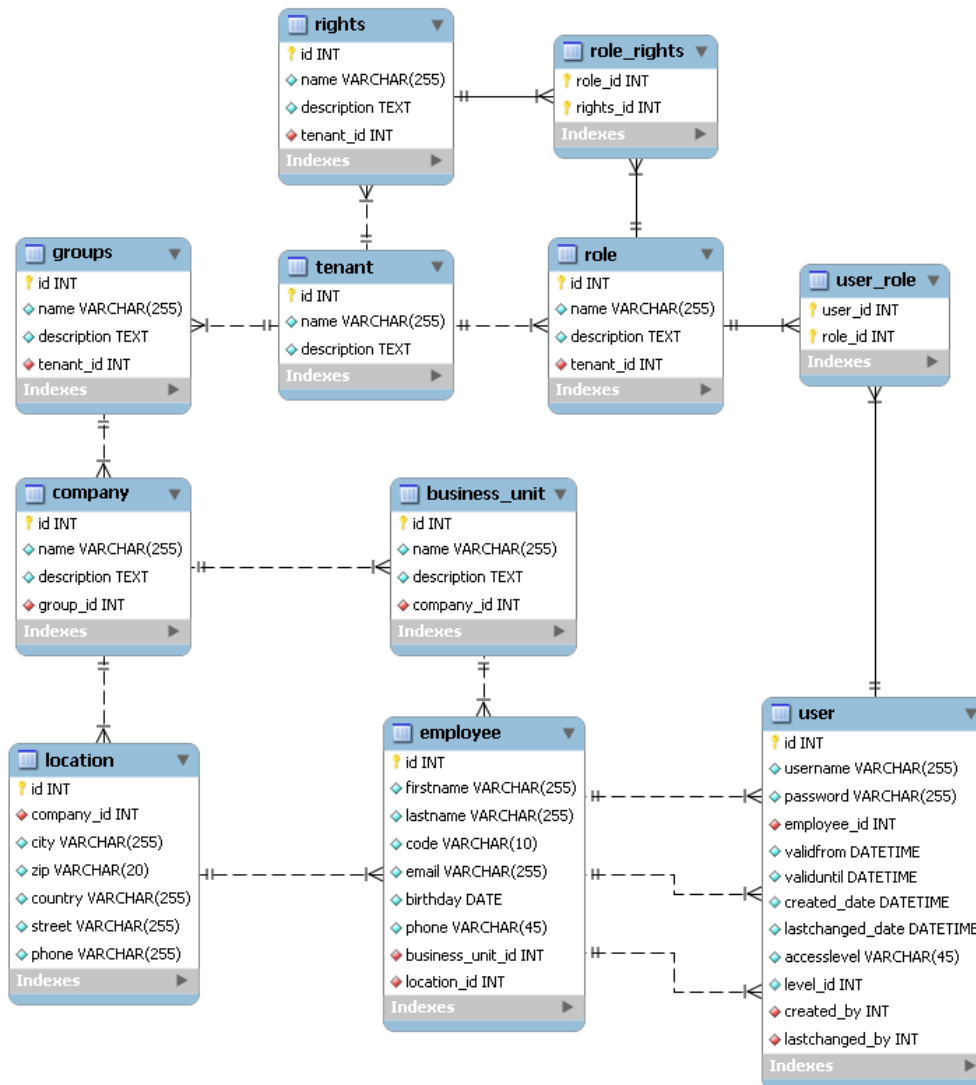


Abbildung 5.1: Datenmodell der Benutzer-Struktur

Abbildung 5.2 zeigt das Modell der Contact-Struktur. Auffällig hierbei ist, dass die üblichen Schlüssel-Tabellen, die Schlüsselwerte enthalten, so nicht mehr vorhanden sind. Stattdessen gibt es eine Tabelle, *enumerationtype*, in der die Schlüsselwerte eingetragen sind. Die Tabellen, die einen Schlüssel referenzieren müssen, referenzieren so nur noch einen Datensatz aus der Tabelle *enumerationtype*.

Den Abbildungen sind aus dem Grund der Übersichtlichkeit die Kardinalitäten vorenthalten. Daher sollen die Grade der Komplexität zu den wichtigsten Tabellen hier aufgeschrieben werden. Ein Contact kann über eine Relationen-Tabelle 0-n Activities referenzieren und über die Tabelle *buyingcenter*, 0-n Opportunities. Ein Contact kann 0-n Adressen haben aber nur einem Account zugeordnet sein.

Abbildung 5.3 zeigt das Modell der Account-Struktur. Hier fehlen aus oben genanntem Grund ebenfalls die Kardinalitäten. Man kann aber erkennen, dass ein Account 0-n Activities, sowie 0-n Opportunities referenzieren kann. Ausserdem kann ein Account 0-n Contacts referenzieren, sowie mehrere Adressen haben. Ebenso kann ein Account 0-n Beziehungen zu anderen Contacts haben, die in der Tabelle *account_relationships* enthalten sind.

Beide Abbildungen enthalten Schnittmengen wie *opportunity*, *activity*, *enumerationtype*, *employee* und *address*. Dennoch wird deutlich, wie die Verbindung zwischen den einzelnen Tabellen hergestellt ist.

5.1.2 Generisches Datenmodell

Das Datenmodell stellt in seiner herkömmlichen Weise die Grundlage für das Datenbanksystem dar. So sind in ihm Entitäten definiert mit Feldern und Datentypen, unter der Voraussetzung, dass es entsprechend der fachlichen Analyse entworfen und implementiert wird. Darüber hinaus gibt es jedoch die Anforderung, ein Datenmodell zu entwerfen, dessen Komponenten, also Entitäten, Felder, Datentypen, vorher nicht bekannt sind. Daher wurde es erforderlich ein Generisches Datenmodell zu konzipieren mit dem es möglich sein soll, flexibel auf die Anforderungen einer Datenstruktur zu reagieren.

Problemstellung

Ein Systemadministrator sollte in der Lage sein, für bestimmte Entitäten, je nach Bedarf ein oder mehrere Felder anzulegen, deren folgende Eigenschaften berücksichtigt werden müssen:

- Auswahl eines Datentyps, der einer der HTML-Komponenten Checkbox, Combobox, Textbox, entspricht
- Festlegen einer Bezeichnung

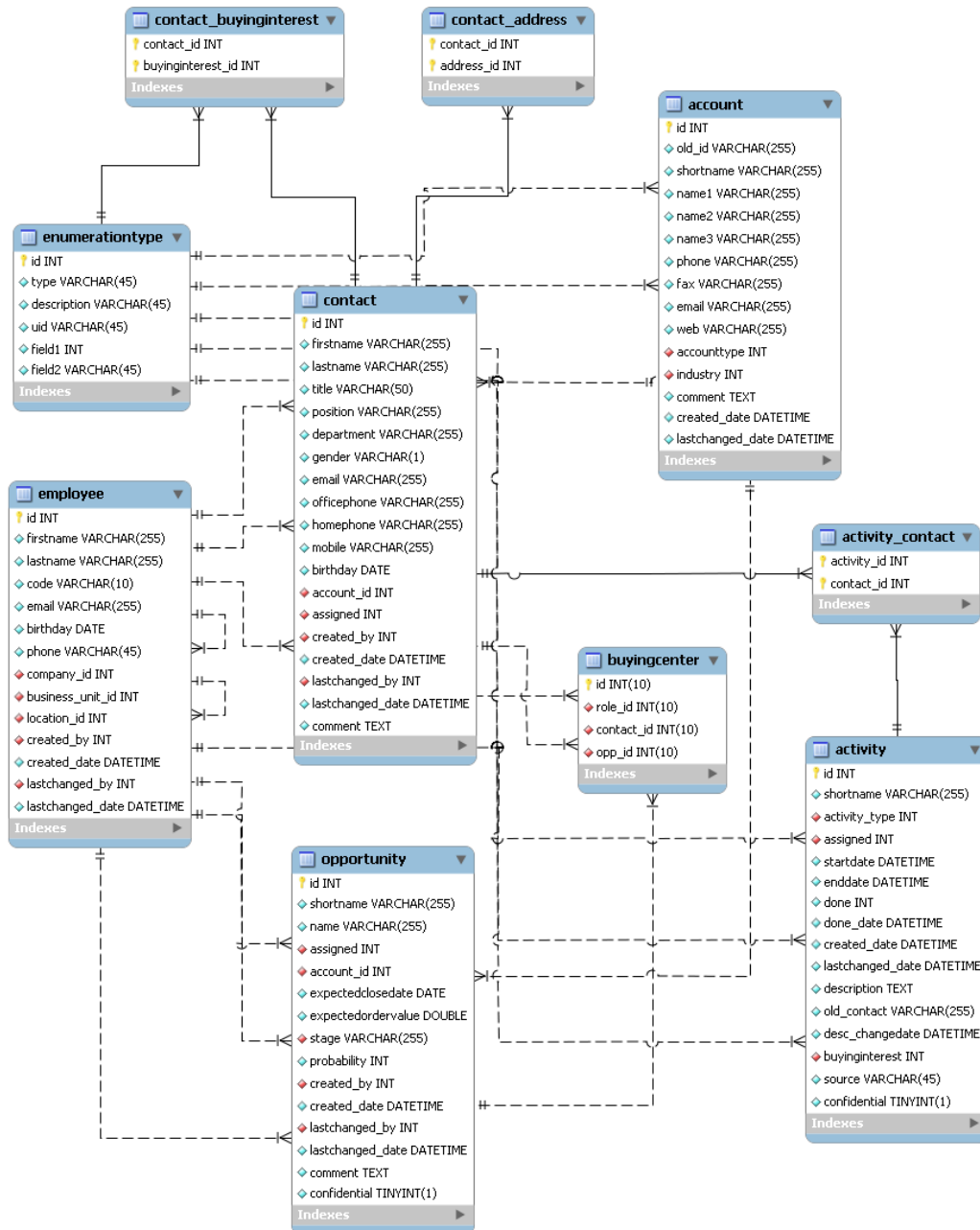


Abbildung 5.2: Datenmodell der Contact-Struktur

5 Architektur

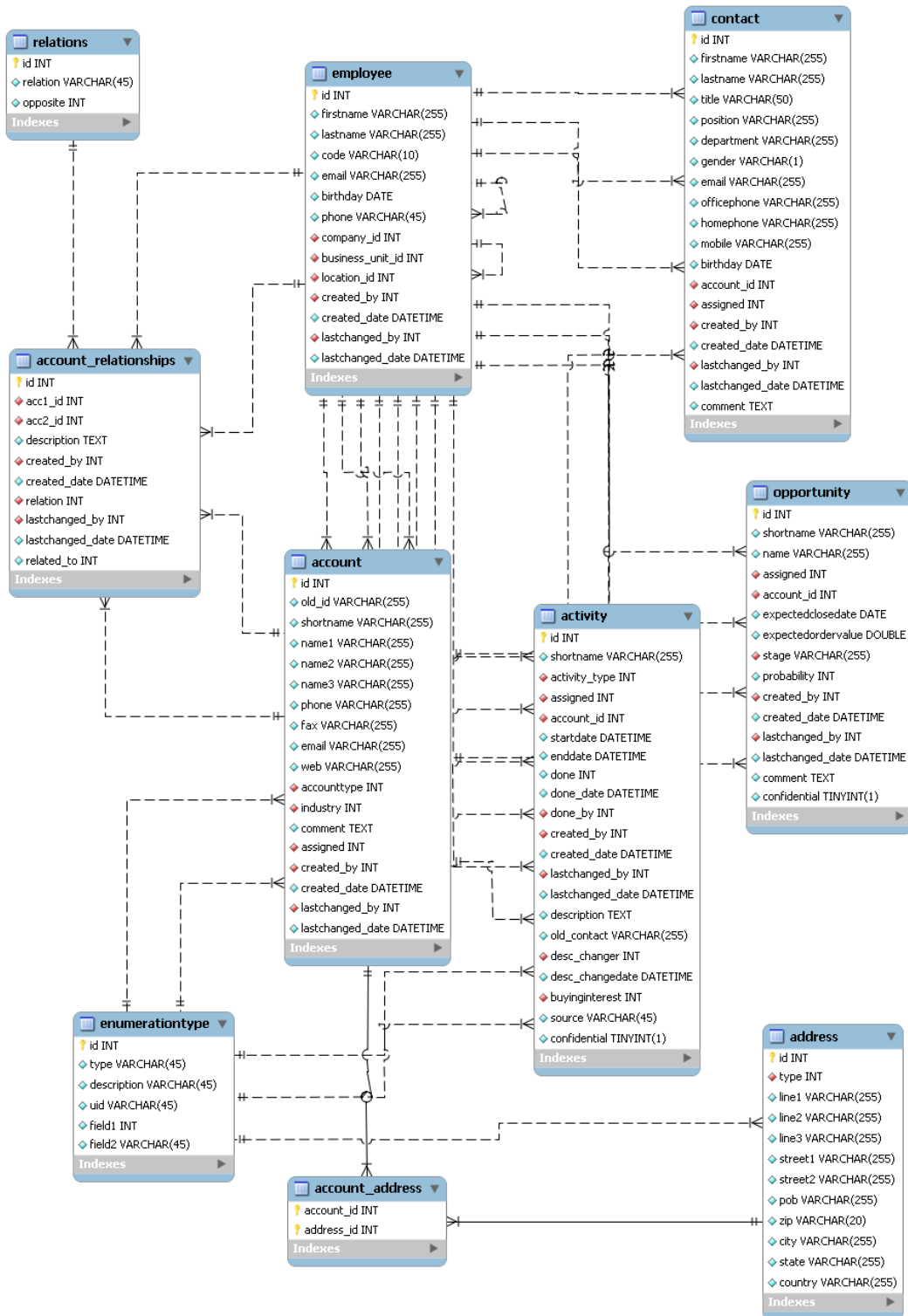


Abbildung 5.3: Datenmodell der Account-Struktur

- Festlegen eines vorgelegten Wertes
- Deklaration zu einem erforderlichen Feld
- Im Falle einer Combobox müssen indizierte Werte hinterlegbar sein
- Im Falle einer Checkbox müssen die Werte „true“ und „false“ vorhanden sein
- Im Falle einer Textbox muss eine Zeichenkette eingebbar sein

Darüber hinaus soll es möglich sein, diese Felder in bestimmte Cluster zu gruppieren, die ebenfalls flexibel angelegt werden können. Als Cluster sind in diesem System in der Darstellung Boxen gemeint, die sich klar abgrenzen und über eine eigene Bezeichnung verfügen.

Lösung

Es musste eine Lösung gefunden werden, die es ermöglicht, das bestehende Datenmodell flexibel zu erweitern, ohne dass ein manipulierender Eingriff in das spezifizierte Datenmodell vorgenommen wird. Abbildung 5.4 zeigt den Ausschnitt des Datenmodells, das als Basis für die Umsetzung des Generischen Datenmodells dient. Die aktuelle Implementierung sieht vor, benutzerdefinierte Felder und Cluster anzulegen. Die Cluster dienen zur Gruppierung der Felder.

Folgende Tabellen gehören zu den Kerntabellen des Generischen Datenmodells:

Tabelle *cust_fields_metadata*

Beinhaltet die Metadaten eines benutzerdefinierten Feldes. Dazu gehören:

- id: Index des Datensatzes
- name: Name des Feldes für interne Benutzung
- entity: enthält die Entität für die das Feld definiert ist (contact, account)
- required: enthält Information darüber, ob das Feld als „required“ gekennzeichnet wird oder nicht (bisher noch nicht genutzt)
- label: Bezeichnung des Feldes in der Benutzungsschnittstelle
- ordering: enthält die Sortier-Reihenfolge des Feldes innerhalb des Clusters
- cluster_id: enthält die *ID* des Clusters, in dem das Feld gruppiert ist
- datatype_id: Typ des Feldes. Entspricht einer HTML-Komponente (Textbox, Checkbox, Combobox)
- defaultvalue: Standard-Wert des Feldes. Je nach Datentyp ist der Wert erforderlich (Checkbox, Combobox) oder im Falle eines Textfeldes frei definierbar

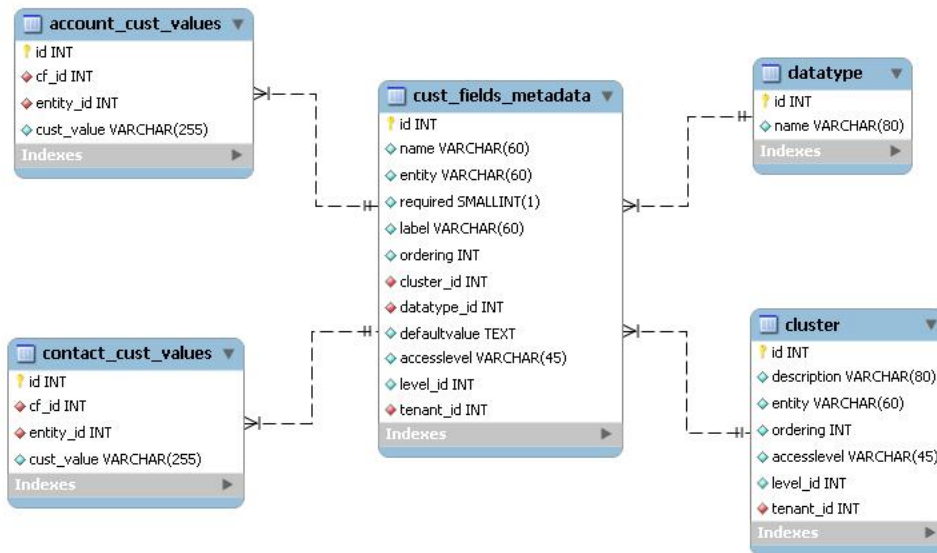


Abbildung 5.4: Tabellen des Generischen Datenmodells

- `accesslevel`: enthält die Zugriffsebene für die Sichtbarkeit des Feldes
- `level_id`: enthält die *ID* des Wertes der Zugriffsebene
- `tenant_id`: *ID* des Mandanten

Tabelle *account_cust_values*

Diese Tabelle enthält die Werte der Entitätsobjekte der Accounts bzgl. der benutzerdefinierten Felder. Dazu gehören:

- `id`: Index des Datensatzes
- `cf_id`: enthält die *ID* des benutzerdefinierten Feldes
- `entity_id`: enthält die *ID*, den Tabelleneintrag in der Tabelle Account
- `cust_value`: Inhalt des benutzerdefinierten Feldes

Tabelle *contact_cust_values*

Diese Tabelle enthält die Werte der Entitätsobjekte der Contacts bzgl. der benutzerdefinierten Felder. Dazu gehören:

- `id`: Index des Datensatzes
- `cf_id`: enthält die *ID* des benutzerdefinierten Feldes
- `entity_id`: enthält die *ID* des Tabelleneintrags der Tabelle Contact
- `cust_value`: Inhalt des benutzerdefinierten Feldes

Tabelle *datatype*

Diese Tabelle enthält Informationen über die verfügbaren Datentypen, die einer HTML-Komponente entsprechen:

- `id`: Index des Datensatzes
- `name`: Name des Datentyps (Textbox, Checkbox, Combobox)

Tabelle *cluster*

Hierin sind die Informationen bzgl. der Cluster, die ebenfalls generisch sind, hinterlegt.

- `id`: Index des Datensatzes
- `description`: Beschreibung des Clusters, dient als Überschrift
- `entity`: enthält die Entität, für die das Feld definiert ist (contact, account)

- ordering: enthält die Sortier-Reihenfolge des Clusters
- accesslevel: enthält die Zugriffsebene für die Sichtbarkeit des Feldes
- level_id: enthält die *ID* des Wertes der Zugriffsebene
- tenant_id: *ID* des Mandanten

In der vorliegenden Version des Vertriebssystems ist das Generische Datenmodell für Accounts und Contacts umgesetzt. Um das Modell für weitere Entitäten wie etwa Activities oder Opportunities zu erweitern, ist für jede dieser Entitäten lediglich eine Tabelle der Form *contact_cust_values* oder *account_cust_values* anzulegen, die schließlich die Werte der Felder aufnimmt. Das Konzept dieses Datenmodells sieht vor, für jede Entität, die durch das Generische Datenmodell ergänzt werden soll, eine eigene Tabelle anzulegen, die die Werte der Felder aufnimmt. Umgesetzt sind die Tabellen für Contacts als *contact_cust_values* und für Accounts als *account_cust_values*. Die Tabellen sind offensichtlich redundant. Den Grund kann man leicht einsehen, wenn man bedenkt, dass für jeden Eintrag einer Entität n-Einträge in der Wertetabelle für die Felder eingetragen werden müssen.

Am folgenden Beispiel soll eine mögliche Tabellenauslastung skizziert werden:

Die Tabelle Contact umfasst etwa 15000 Einträge.

Für die Contacts sind 10 benutzerdefinierte Felder hinterlegt.

In der Wertetabelle für die Generischen Felder müssen dann $15000 * 10 = 150000$ Einträge gespeichert werden.

Nimmt man die Tabelle mit den Activities hinzu, ergibt sich folgendes:

Die Tabelle Activity umfasst etwa 45000 Einträge.

Hat man auch hierfür 10 benutzerdefinierte Felder, müssen in der Wertetabelle für die Activities $45000 * 10 = 450000$ Einträge gespeichert werden. Zuzüglich zu den Contacts ergeben sich etwa 600000 Einträge.

Dies ist ein Beispiel mit nur zwei berücksichtigten Entitäten. Da man aber nicht vorhersehen kann wie viele Entitäten einmal berücksichtigt werden müssen, wie viele Felder es geben wird und welches RDBMS eingesetzt wird, muss man dieses Opfer bringen und offensichtliche Redundanzen zulassen.

Abbildung 5.5 zeigt am Beispiel eines Accounts die Darstellung der Generischen Felder und ihre Aufteilung in Cluster.

Phone	089 / 123456	-0
Fax		
E-Mail	info@b-i-b.de	
Web	http://	
Facts & Figures		
Klassifizierung	A	
Werbe-Etat/Jahr	- Choose -	
Jahres-Umsatz	10 is 50 Mio Euro Umsatz / Jahr	
Unique Visitors	- Choose -	
Anzahl Mitarbeiter	51-100	
Software Environment		
Online-Account?	<input type="checkbox"/>	
Schnittstelle vorhanden?	<input type="checkbox"/>	
Telefonanlage	ISDN	
Comment		

Abbildung 5.5: Generische Felder im Einsatz am Beispiel eines Account

5.1.3 Sichtbarkeitskonzept

Das Generische Datenmodell unterliegt zusätzlich der Anforderung, an ein Sichtbarkeitskonzept gekoppelt zu sein. Die zugrunde liegende Idee ist die Berücksichtigung der Unternehmensstruktur (Abbildung Businessprofil 5.6).

Problemstellung

Nicht alle Generischen Felder und Cluster sind für jeden Anwender des Systems relevant. Teilweise sind es Gründe, die nach Schutz der Daten verlangen, teilweise können die Gründe aber auch trivialer Natur sein. Beispiel: diese Information bringt keinen Vorteil für die Arbeitsverrichtung. Wie auch immer die Entscheidung für eine Einschränkung des Zugangs zu diesen Feldern ausfällt, braucht man eine Strategie, um diesen Zugang zu administrieren und den Zugang der Anwender abzufragen.

Lösung

Die Administration der Generischen Felder und Cluster sieht vor, einen *Access Level*, sowie einen *Access Scope* festzulegen. Der Access Level entspricht einer Ebene des Businessprofils, also Group, Company oder Business Unit. Der Access Scope ist innerhalb des Access Levels ein Element dieser Ebene. Das bedeutet also, dass der Zugang zu diesem Feld nur demjenigen gestattet ist, der innerhalb dieser Struktur tätig ist.

Auf der anderen Seite kann ein Administrator aber auch einem Mitarbeiter Zugang zu diesen Feldern verschaffen, der nicht direkt in dem definierten Unternehmensbereich tätig ist. Beispielsweise kann ein Mitarbeiter, der in Business Unit A, der Company A eingegliedert ist, das Recht bekommen, Felder zu sehen, die nur für Mitarbeiter der Business Unit B, der Company A, sichtbar sind.

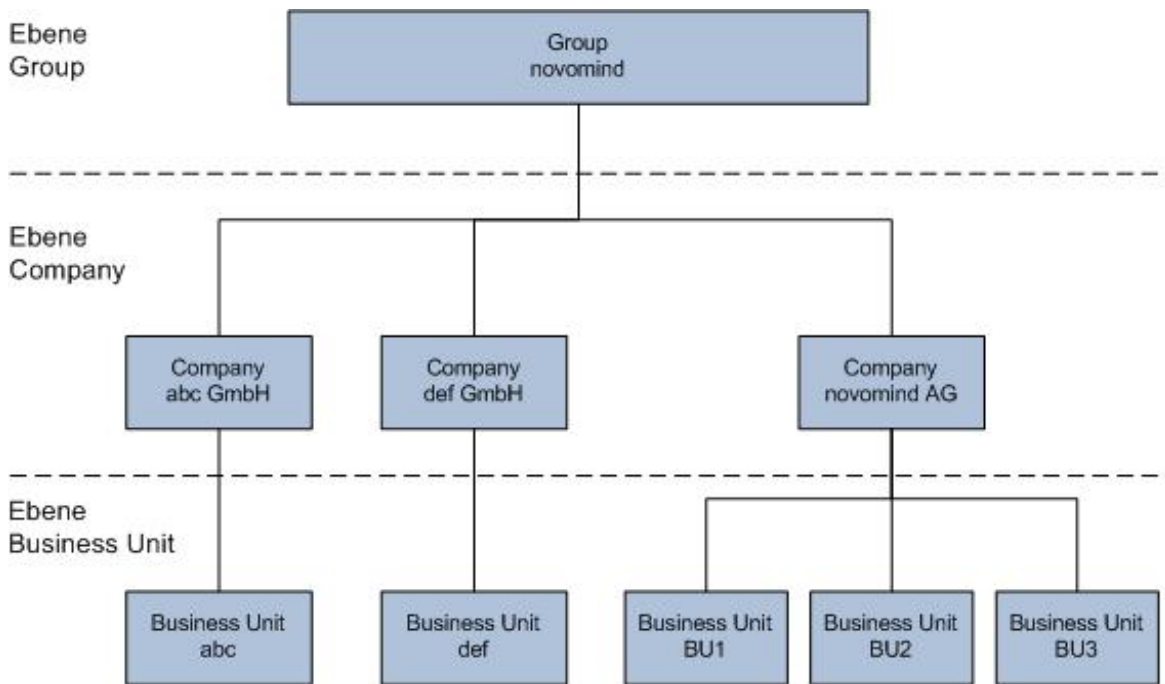


Abbildung 5.6: Businessprofil

5.1.4 Rollen-/Rechtesystem

Das Rechtesystem des vorliegenden Vertriebssystems beruht auf der Verteilung von atomaren Rechten im System. Atomar heißt in diesem Zusammenhang, dass ein Recht nicht weiter aufgebrochen werden kann.

Die Rechte können mannigfaltig sein und überall dort platziert werden, wo eine Einschränkung bzw. Kontrolle des Datenflusses sinnvoll erscheint.

Verteilung der Rechte

Die Vergabe der Rechte an Benutzer des Systems gestaltet sich über die Ergänzung eines

Rollensystems. Das Rollensystem in diesem System ist nichts anderes als eine Gruppierung von Rechten.

Die Rechte werden in der Rollen-/Rechteverwaltung der Administration vergeben, indem eine bestehende Rolle ergänzt bzw. in ihren Rechten reduziert wird, oder eine neue Rolle angelegt wird und ihr die notwendigen Rechte zugeteilt werden. Eine Rolle kann 1-n Rechte besitzen.

Die Benutzer wiederum erhalten ihr Recht, das System zu benutzen, über diese Rollen. Die Rollen werden in der Benutzerverwaltung der Administration vergeben. Ein Benutzer kann 1-n Rollen haben.

Vorteile dieses Systems

Als Vorteil ist hier klar die Freiheit bei der Verwaltung der Rollen zu nennen. Ein Administrator hat so die Möglichkeit, die vorhandenen Rechte zu gruppieren und eine passende Bezeichnung zu hinterlegen. Außerdem können so einem Benutzer auch mehrere Rollen zugewiesen werden.

An der entsprechenden Stelle im System wird unabhängig von der Rolle überprüft, ob der Benutzer das Recht hat, am definierten Prozesskontext teilzunehmen.

5.1.5 Fazit

Mit der Implementierung des Generischen Datenmodells erreicht das Vertriebssystem eine enorme Flexibilität, die einen Vergleich zu eingesessenen Systemen auf dem Open Source Sektor nicht zu scheuen braucht. Die Abbildung des Businessprofils und die damit verbundene Einführung des Sichtbarkeitskonzepts stellen eine Erweiterung dar, die in dieser Form umgesetzt nicht zu finden ist und sie stellt in Ergänzung zum Rechtesystem ein effektives Werkzeug zur Verfügung, dem notwendigen Datenschutz und der Vertrauenswürdigkeit Tribut zu zollen.

5.2 Technische Architektur

Bereits im Kapitel [2], „Grundlagen“, wurde auf die eingesetzten Technologien eingegangen. Dort wurde u.a. EJB 3.0 als Komponentenmodell vorgestellt und die Entscheidung für eine Mehrschichtenarchitektur erläutert. Eine Schichtenarchitektur ist ein Strukturierungsprinzip für ein Software-System. Das vorliegende Software-System besteht aus Komponenten, die modulare Einheiten bilden, die in ihrer Umgebung ersetzbar sind. Siehe hierzu z.B. die Session Beans aus der Schicht der Geschäftslogik (vgl. Kapitel 2.3).

Dabei ist die grundlegende Idee, die Komponenten schichtweise anzuordnen, wobei eine Komponente einer Schicht von einer Komponente der darüber liegenden Schicht aufgerufen wird (vgl. Tanenbaum und van Steen, 2008, S. 53).

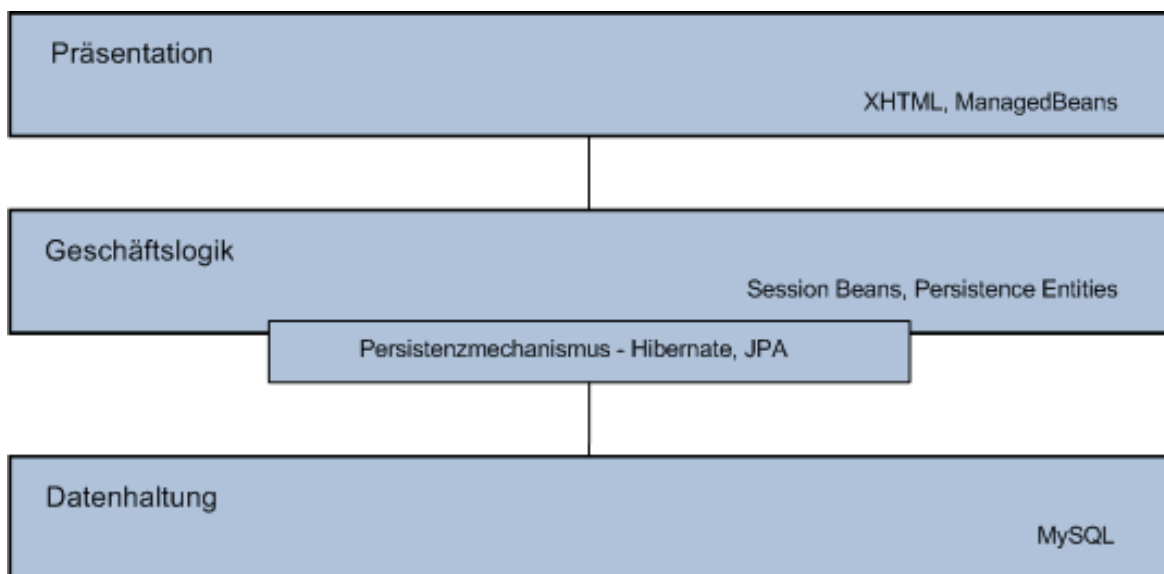


Abbildung 5.7: Schichtenmodell - schematisch

Abbildung 5.7 zeigt schematisch den Aufbau der Applikation in einer Mehrschichtenarchitektur. Es wird die Aufteilung nach dem bekannten Model View Controller (MVC) Architekturmuster deutlich. So teilt sich die Architektur in eine Präsentationsschicht (View), einer Schicht für die Geschäftslogik (Controller) und einer Schicht für die Datenhaltung (Model) auf.

Die **Präsentationsschicht** enthält die JSF-Komponenten. Diese setzen sich aus den JSF-Seiten, die durch HTML/XHTML-Seiten repräsentiert werden, und den Managed Beans zusammen. Managed Beans sind Java-Klassen, die die Daten der UI Komponenten

sammeln und z.B. ActionListener-Methoden bzw. Events behandeln. Sie nehmen bspw. die Daten eines Formulars entgegen und leiten diese an die Geschäftslogik weiter. Über das *Service Locator Pattern* (vgl. Sun Microsystems Inc., 2008a) wird die Verbindung zu den Geschäftsmodellen (vgl. Ihns u. a., 2007) in der Geschäftslogik-Schicht hergestellt. So wird es möglich, die Geschäftslogik-Schicht physisch von der Anwendung zu trennen und z.B. das Geschäftsmodell auf einen anderen Server auszulagern.

In der Schicht der **Geschäftslogik** befinden sich die Geschäftsmodelle. Diese sind einfache Java-Klassen (POJO), in denen die Geschäftsmethoden untergebracht sind. Diese so genannten zustandslosen Session Beans laufen in einem Applicationserver, in diesem Fall JBoss, und bedienen sich der Persistence Entities. Die Abfragen an die Persistence Entities erfolgt mit Hilfe der so genannten Java Persistence Query Language (Java Persistence Query Language (JPQL)), eine Abfragesprache, die sich syntaktisch SQL-Anfragen ähnelt. Für tiefer gehende Informationen dieser Technologie sei an dieser Stelle auf weiterführende Literatur verwiesen (vgl. Ihns u. a., 2007).

In der **Datenhaltungsschicht** werden die Zustände der Objekte bzw. der Persistence Entities über ein objektrelationales Mapping via Hibernate dauerhaft gespeichert. Der Zugriff erfolgt in den Session Beans über die Java Persistence API (JPA). Eingesetzt wird hier eine MySQL- Datenbank. Ein entscheidender Vorteil der Verwendung von Hibernate ist die Unabhängigkeit der Anwendung vom SQL-Dialekt der eingesetzten Datenbank.

6 Realisierung

6.1 Vorgehensweise

6.1.1 Ziele

Wie bereits in der Einleitung erwähnt, liegt dieser Arbeit ein ganz bestimmtes Vorhaben zu Grunde, nämlich ein bestehendes proprietäres Vertriebssystem durch ein neu konzipiertes webbasiertes Vertriebssystem abzulösen. Dem folgt die Bedingung, innerhalb eines vorgegebenen Zeitraumes das alte System vollständig zu ersetzen, um so die ungehinderte Fortführung der damit verbundenen Tätigkeiten in der Abteilung sicherzustellen. Um dieses Ziel zu erreichen, wird die Realisierung einer bestimmten Vorgehensweise unterworfen, die in diesem Kapitel erläutert wird.

6.1.2 Proof of Concept

Ein bestehendes Softwaresystem durch ein neu zu konzipierendes System abzulösen, ist kein trivialer Vorgang. Daher ist es notwendig, sich an Regeln zu halten, die eine klare Struktur der Vorgehensweise vorgeben. Für diese Arbeit erwies es sich als sinnvoll, eine Vorgehensweise zu wählen, mit der man die fachlichen Funktionen abbilden und die Ergebnisse der Arbeit sehr zeitnah reflektieren und diskutieren kann. Recht früh wurden daher die Vertriebsmitarbeiter und die zukünftigen Anwender des Systems mit in die Entwicklung eingebunden. So konnten etwaige Schwachstellen und Lücken in der gewünschten bzw. geforderten Funktionalität aufgedeckt werden und durch definierte Autor-Kritiker-Zyklen (Abb. 6.1) eliminiert werden.

Diese Zyklen werden durch einen stetig weiter zu entwickelnden Prototypen erreicht. Damit wird die Durchführbarkeit demonstriert, bis zum Erreichen der Marktreife.

Autor-Kritiker-Zyklus

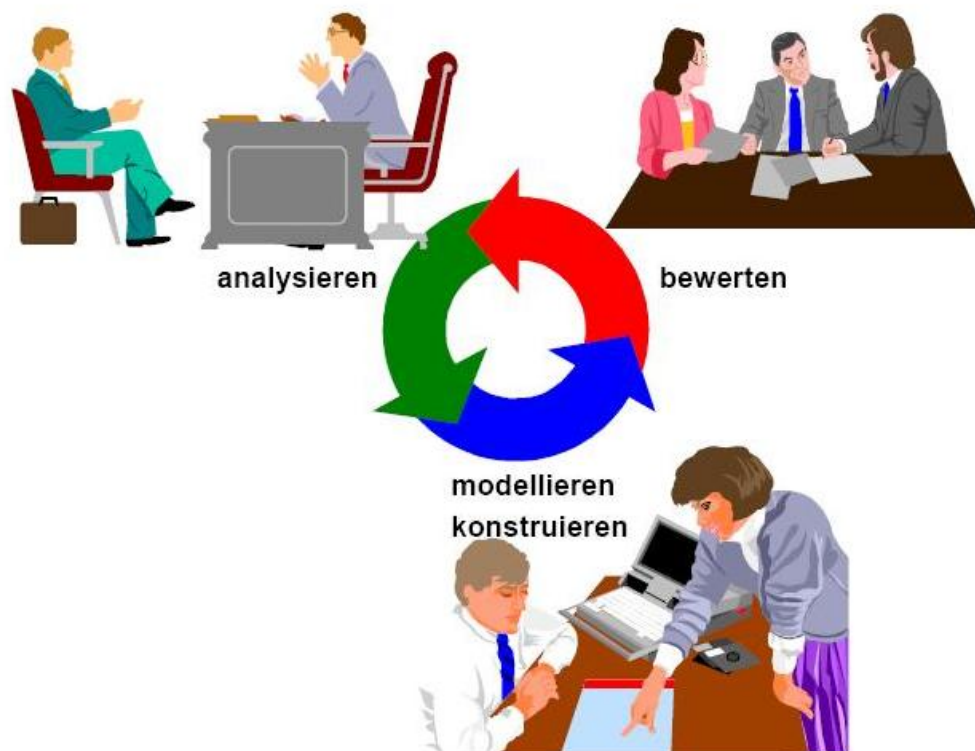


Abbildung 6.1: Autor-Kritiker-Zyklus (Züllighoven)

6.1.3 Evolutionäres Prototyping

Um den Anspruch gerecht zu werden, die Mitarbeiter aktiv mit in den Entwicklungsprozess einzubinden, bietet es sich an, ein Evolutionäres Prototyping zu wählen. Damit wird die Weiterentwicklung des Prototyps in Zyklen fortgeschrieben bzw. ein schrittweiser Ausbau des Systems erreicht. Das Ziel ist eine inkrementelle Systementwicklung, das heißt eine schrittweise aufbauende Entwicklung nach folgender Vorgangsweise: Entwicklung eines Prototyps für die von vornherein klaren Benutzeranforderungen. Das Ergebnis dient als Basissystem für den Anwender und für den nächsten Schritt, in dem neue Benutzeranforderungen integriert und der Prozess von neuem beginnt (vgl. Pomberger und Pree, 2004, S. 28).

Die Prototypen bekommen jeweils eine Versionsnummer und werden vor jeder Aktualisierung einer Validierung unterworfen. Erst wenn die Validierung erfolgreich verlaufen ist und durch den Anwender definierte Abnahmetests erfolgreich durchlaufen hat, wird der Stand des Prototypen eingefroren und dieser als Basis für den nächsten Prototyp zur Verfügung gestellt.

Den Vorteil dieses Verfahrens erkennt man leicht, wenn man sich klar macht, dass auf diese Weise Funktionalitäten kurzfristig geplant, implementiert und zur Diskussion bzw. zum Testen freigegeben werden können. So hat man die Möglichkeit, sehr schnell auf Komplikationen, Missverständnisse oder fehlerhaftes Verhalten zu reagieren.

Der Umfang eines Prototyps ist nicht klar gegliedert. Er wird individuell geplant. Dies ist abhängig vom Zeitrahmen, der zur Verfügung steht, und vom Umfang an Funktionen, die zur Implementation anstehen. So ist man flexibel bezüglich der Gestaltung der einzelnen Versionen. Das vorliegende Vertriebssystem wurde grob in zehn Prototypen eingeteilt mit den Versionsnummern V0.1-1.0, wobei die Version V1.0 die finale Version ist, die zur Freigabe in den produktiven Einsatz gestellt wurde.

Abbildung 6.2 zeigt die Aktivitäten eines Prototyping-orientierten Prozessmodells.

6.2 Aufteilung in Module

Die Kernkomponenten des Vertriebssystems können klar identifiziert und leicht in Module aufgeteilt werden. Dies ist sinnvoll, da so eine logische Trennung vorgenommen werden kann, die zum einen eine klare Benutzerführung erlaubt und sich zum anderen auf die Qualität der Entwicklung bzw. Wartung des Systems niederschlägt. Somit wird auch dem geforderten Model-View-Controller Ansatz (siehe Kapitel 5 „Architektur“) gerecht.

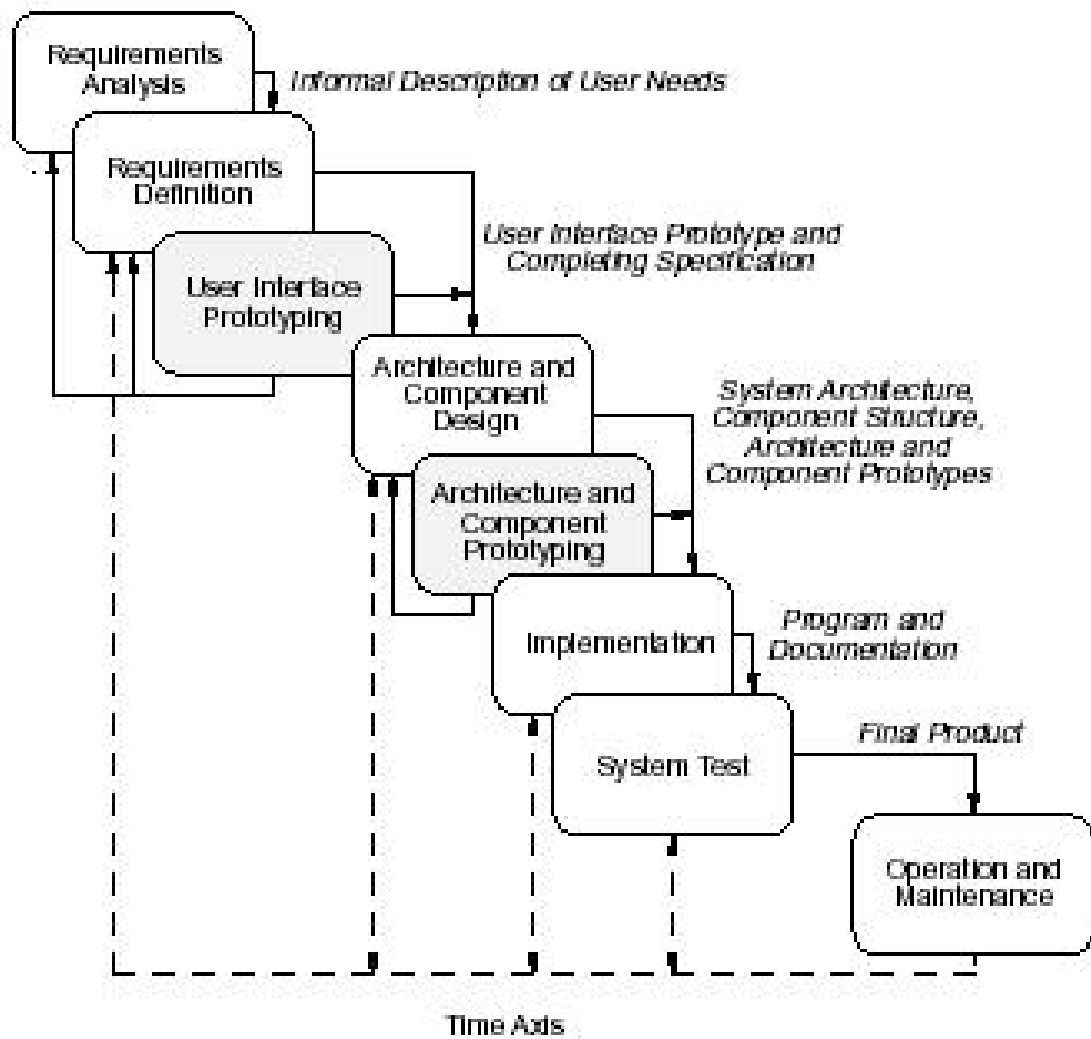


Abbildung 6.2: Entwicklungsprozess mit Prototypen (vgl. Pomberger und Weinreich, 1994)

Die Aufteilung in Module wird auch den Prinzipien des objektorientierten Architekturentwurfs gerecht. So beschreiben Ludewig und Lichter (Ludewig und Lichter, 2007, S. 380) ein Modul als eine Menge von Operationen und Daten, die nur so weit von außen sichtbar sind, wie dies die Programmierer explizit zugelassen haben. Sie ebnet somit den Weg zum objektorientierten Entwurf in dem, dem Prinzip des Information Hiding Rechnung getragen wird als auch die lose Kopplung eines Moduls zu anderen Modulen und der interne Zusammenhalt eines Moduls eine tragende Rolle spielt. Diese Entwurfsprinzipien werden stark durch das MVC-Architekturmuster, sowie den Einsatz von EJB 3.0 (siehe Kapitel 2) unterstützt.

Um diesem Anspruch gerecht zu werden, sind folgende Module im System realisiert:

- Contact-Modul
- Account-Modul
- Activity-Modul
- Opportunity-Modul
- Administration-Modul

Jedes dieser Module beinhaltet den Umfang an Funktionen, um die gegebenen Anwendungsfälle abzubilden. Die Module sind auch im Sinne der Objektorientierung mit starker Kohäsion ausgelegt, so dass in jedem Modul, nur die für diese Aufgaben notwendigen Funktionalitäten zur Verfügung stehen. Eine einheitliche Gestaltung der einzelnen Module sorgt für einen hohen Wiedererkennungswert und unterstützt den Anwender bei der Bewältigung seiner Aufgaben. Die angezeigten, modulabhängigen Daten werden in Bereiche zusammengefasst, die sich in diesem System *Cluster* nennen. In diesen Clustern werden verwandte Informationen übersichtlich präsentiert. Eine Durchsetzung der Cluster ist in jedem Modul realisiert.

6.3 Contact-Modul

Das Contact-Modul beinhaltet sämtliche Informationen rund um einen Contact und die Möglichkeit, diese zu pflegen. Ein Contact ist ein Mitarbeiter eines *Accounts*, der als Kontaktperson für den Lieferanten dient. Sämtlicher Geschäftsverkehr zu einem Account läuft über diesen Contact. Eine Kontaktperson hat somit sämtliche Personen relevante Stammdaten (Master Data), wie Name, Vorname, Geburtstag, Email, Telefon etc. Diese sind in einem Cluster *Master Data* (Abb. 6.3) zusammengefasst. Additiv zum Cluster *Master Data* gibt es noch das Cluster *Owner/Usage*, das Informationen über den Inhaber (Owner) eines Datensatzes enthält, sowie Informationen zum letzten Änderungsdatum bzw. zum Erstellungsdatum, und das Cluster *Addresses* mit den Adressinformationen des Contacts.

Als besonderes Merkmal ist hervorzuheben, dass man die Möglichkeit hat, zu den Contacts selbst definierte Felder zu erstellen (siehe Kapitel 5) und diese ebenfalls in Cluster zusammenzufassen. Die Contact-Detailansicht beinhaltet darüber hinaus Informationen zu dessen Activities, sowie Opportunities, an denen der Contact beteiligt ist, sowie ein Kommentar.

Das Contact-Modul teilt sich in verschiedene Ansichten/Masken auf:

- Contact-Übersicht inkl. Suchmöglichkeiten
- Contact-Detailansicht, Übersicht sämtlicher Informationen zu einem Contact
- Anlegen, bzw. Editier-Maske

Erreicht wird das Contact-Modul über den Reiter *Contacts*. Hier gelangt man auf die Übersicht sämtlicher Contacts, mit der Möglichkeit nach bestimmten Contacts zu suchen. Die Suche beinhaltet verschiedene Suchkriterien, die miteinander verknüpft werden können. Gesucht werden kann nach Inhaber (Owner), Kaufinteresse (Buying Interest), Vorname (Firstname) oder Nachname (Lastname). Die Anlegen- bzw. Editier-Ansicht (Abb. 6.4) sind gleich aufgebaut und unterscheiden sich lediglich in der Überschrift.

Das Contact-Modul bildet folgende Anwendungsfälle ab:

Anlegen eines Contacts

Ein Contact wird über den Button *add Contact* aus der Contact-Übersicht heraus angelegt. Notwendige Felder, die zum Anlegen erforderlich sind, werden fett gedruckt dargestellt, eine entsprechende Validierung dieser Felder weist den Anwender nötigenfalls auf das Fehlen hin. Die Felder teilen sich auf in die Stammdaten, die selbst definierten Felder und die Adressdaten.

Bearbeiten eines Contacts

Bearbeitet wird ein Contact über den Button *edit Contact* aus der Detailansicht des Contacts heraus. Das Bearbeiten erfolgt in gleicher Weise wie das Anlegen eines Contacts. Notwendige Felder sind entsprechend markiert und mit Validatoren versehen. Im Bearbeiten-Modus besteht zudem die Möglichkeit, den Contact einem anderen Account mit seinen sämtlichen Daten zuzuweisen.

Löschen eines Contacts

Ein Contact kann gelöscht werden, wenn dieser nicht mehr benötigt wird. Dies geschieht über den Button *delete Contact* aus der Detailansicht. Das Löschen ist mit einem dauerhaften Entfernen aus der Datenbank verbunden und immer mit Vorsicht zu bedienen. Deshalb ist dieser Vorgang auch nur dann möglich, wenn keine relevanten Referenzen zu Activities und Opportunities mehr bestehen.

Ändern des Inhabers (Owner)

Es ist möglich, den Inhaber eines Contacts zu ändern und einem anderen Anwender zuzuweisen. Dazu gibt es in der Detailansicht die Möglichkeit, einen anderen Anwender des Systems auszuwählen.

Hinzufügen einer Activity

Über das Contact-Modul kann eine Activity angelegt werden, dabei wird die Activity direkt dem Contact zugeordnet. Dafür steht in der Detail-Ansicht des Contacts ein Button *add Activity* zur Verfügung.

Contact einem anderen Account zuweisen

Der Benutzer hat die Möglichkeit, einen Contact einem anderen Account zuzuweisen. Dies ist z.B. dann notwendig, wenn der Contact das Unternehmen wechselt. Die ihm zugewiesenen Activities bleiben dabei erhalten. Ebenso die Opportunities, an denen er im Sinne des Buying Centers beteiligt ist.

Contact einem Buying Center hinzufügen

Dieser Anwendungsfall wird nur indirekt über das Contact-Modul durchgeführt. In der Contact-Detailansicht ist lediglich eine Auflistung der Opportunities, an denen er als Mitglied eines Buying Center beteiligt ist, zu finden. Ein Buying Center ist in einem Unternehmen eine Gruppe von Personen, die an einer Kaufentscheidung beteiligt ist. Ein Contact wird einem Buying Center über eine Opportunity hinzugefügt. Ein Contact nimmt im Buying Center eine spezielle Rolle (Buyer, Decider, Gatekeeper, Influencer, User) ein.

6 Realisierung

The screenshot displays the novomind iSELL CRM interface in a Windows Internet Explorer browser. The page title is "novomind iSELL - Contact Details Herr Oliver Müller". The browser address bar shows "https://localhost:8443/contacts/contactdetail.faces". The interface includes a navigation menu with options like Home, Activities, Accounts, Contacts, Opportunities, Campaigns, Reports, and Admin. The user is logged in as André Schmer, Products, novomind AG, Hamburg. The main content area shows the contact details for "Herr Dipl. Betriebswirt Oliver Müller".

Contact Details Herr Dipl. Betriebswirt Oliver Müller

Master Data

Account	B-I-B GMBH
Title	Dipl. Betriebswirt
Firstname	Oliver
Lastname	Müller
Department	Vertrieb
Position	Leiter Einkauf
Email	o.muller@bib.de
Office Phone	089 / 123456 -7
Home Phone	
Mobile	0176-7654321
Birthday	02. April 1973
Buying Interests	email

Owner / Usage

Owner	André Schmer / novomind AG [re-own]
Created By	André Schmer (13.07.2008)
Last Changed By	André Schmer (16.08.2008)

Facts & Figures

Newsletter	<input checked="" type="checkbox"/>
Broker	<input type="checkbox"/>
Interessen	Segeln

Addresses

Private: Oliver Müller
Münchener Strasse 17
89409 München
GERMANY
[Show Location](#)

Comment

Opportunities / BuyingCenter [1]

Activities [2] [add Activity](#)

Shortname	End Date	Assigned To	Done
email - Follow Up	30.07.2008	André Schmer	<input type="checkbox"/>
Partner Contact - email	25.07.2008	André Schmer	<input type="checkbox"/>

Page 1 of 1

novomind iSell v1.0-000
2008-07-04 13:00

© 2008 novomind AG

Abbildung 6.3: Contact - Detailansicht

6 Realisierung

The screenshot shows a web browser window titled "novomind iSELL - Contacts - Windows Internet Explorer". The address bar shows "https://localhost:8443/contacts/edit_contact.faces". The browser interface includes a search bar with "Google" and a "Zertifikatfehler" warning. The application header features the "iSELL" logo, a navigation menu (Home, Activities, Accounts, Contacts, Opportunities, Campaigns, Reports, Admin), and a user profile for "André Schmer" (Products, novomind AG, Hamburg) with a "Logout" button. Below the header, a breadcrumb trail reads "Last Viewed: Partner Contact - email | email - Follow Up | B-I-B GMBH | Müller, Oliver Dipl. Betriebswirt | BIB". A search bar is also present.

The main content area is titled "Edit Contact Oliver Müller" and contains a form with the following sections:

- Contact Master Data:** A table of fields for account and personal information. Fields in bold are required.
- Facts & Figures:** A section for newsletter, broker, and interests.
- Comment:** A text area with a rich text editor toolbar.

Contact Master Data	
Account	B-I-B GMBH
reallocate to Account	<input type="text"/>
Title	Dipl. Betriebswirt
Firstname	Oliver
Lastname	Müller
Department	Vertrieb
Position	Leiter Einkauf
Gender	male
Email	o.müller@bib.de
Office Phone	089 / 123456 - 7
Home Phone	<input type="text"/>
Mobile	0176-7654321
Birthday (DD.MM.YYYY)	02.04.1973
Buying Interest	- Choose Buying Interest -
List of Buying Interests	email [remove]

Facts & Figures	
Newsletter	<input checked="" type="checkbox"/>
Broker	<input type="checkbox"/>
Interessen	Segeln

Comment

- Kompetent
- Teamfähig
- Kontaktfreudig

Abbildung 6.4: Contact - Editier-Maske

6.4 Account-Modul

Ein *Account* ist in der Vertriebsstruktur ein Unternehmen, das mit dem Lieferanten in irgendeiner Beziehung steht. Die Accounts im vorliegenden Vertriebssystem werden im Account-Modul gepflegt. Ein Account besteht im wesentlichen Sinne aus den Daten, die ein Unternehmen ausmachen. Dazu gehören Name des Unternehmens, Kontaktdaten wie Telefonnummer, Faxnummer, E-Mail-Adresse und Internetadresse sowie Anschriften(n). Darüber hinaus können Unternehmen eingeordnet werden in einen so genannten *Account Type*. Ein Account Type ist die Einordnung des Accounts aus Sicht des Lieferanten, wie z.B. Competitor oder Customer.

Folgende Account Types stehen in diesem Vertriebssystem zur Verfügung:

- Competitor
- Customer
- Customer from SP
- End Cus Competitor
- Potential CUS
- Potential SP
- Potential Technology Partner
- Press
- SP Competitor
- Solution Provider
- Supplier
- Technology Partner
- not active

Ein Account gehört außerdem einer Branche oder auch Industriezweig an. Dieses wird als *Industry* bezeichnet und steht in folgenden Ausprägungen zur Verfügung:

- Advertising Agency
- Automotive
- Call Center
- Finance

- Government
- IT
- Industry
- Insurance
- Media
- Other
- Press
- Telecommunication
- Tourism
- Wholesale

Die Daten des Accounts sind in so genannte *Cluster* angeordnet, eine Gruppierung von ähnlichen Daten, die sich logisch von anderen Bereichen abgrenzen. Im Account-Modul gibt es das Cluster *Master Data*, das die Stammdaten zum Account enthält, das Cluster *Owner/Usage*, mit Informationen über den Inhaber des Datensatzes und Zeitangaben über die letzte Bearbeitung sowie das Erstellungsdatum, das Cluster *Addresses*, mit den Adressinformationen zu dem Account und zudem noch die Generischen Cluster, die erstellt werden können, um die Generischen Felder zu gruppieren (Abb. 6.5).

Das Account-Modul bietet ebenso die Möglichkeit, selbst definierte Felder zu erstellen (siehe Kapitel 5) und diese ebenfalls in Cluster zusammenzufassen.

Das Account-Modul teilt sich in verschiedene Ansichten/Masken auf:

- Account-Übersicht inkl. Suchmöglichkeiten
- Account-Detailansicht, Übersicht sämtlicher Informationen zu einem Account
- Anlegen bzw. Editier-Maske

Erreicht wird das Account-Modul über den Reiter *Accounts*. Hier gelangt man auf die Übersicht sämtlicher Accounts, mit der Möglichkeit nach bestimmten Accounts zu suchen. Die Suche beinhaltet verschiedene Suchkriterien, die miteinander verknüpft werden können. Gesucht werden kann nach Inhaber (Owner), Kurzbezeichnung (Shortname), Name, Industriezweig (Industry) oder Typ (Account Type). Die Anlegen- bzw. Editier-Ansicht (Abb. 6.6) sind gleich aufgebaut und unterscheiden sich lediglich in der Überschrift.

Das Account-Modul bildet folgende Anwendungsfälle ab:

Anlegen eines Accounts

Ein Account wird über den Button *add Account* aus der Account-Übersicht heraus angelegt. Notwendige Felder, die zum Anlegen erforderlich sind, werden fett gedruckt dargestellt, eine entsprechende Validierung dieser Felder weist den Anwender nötigenfalls auf das Fehlen hin. Die Felder teilen sich auf in die Stammdaten, die selbst definierten Felder und die Adressdaten.

Bearbeiten eines Accounts

Bearbeitet wird ein Account über den Button *edit Account* aus der Detailansicht des Accounts heraus. Das Bearbeiten erfolgt in gleicher Weise wie das Anlegen eines Accounts. Notwendige Felder sind entsprechend markiert und mit Validatoren versehen.

Ändern des Inhabers (Owner)

Es ist möglich, den Inhaber eines Accounts zu ändern und einem anderen Anwender zuzuweisen. Dazu gibt es in der Detailansicht die Möglichkeit, einen anderen Anwender des Systems auszuwählen.

Hinzufügen einer Activity

Über das Account-Modul kann eine Activity angelegt werden, dabei wird die Activity direkt dem Account zugeordnet. Dafür steht in der Detailansicht des Accounts der Button *add Activity* zur Verfügung.

Hinzufügen einer Opportunity

Über das Account-Modul kann eine Opportunity angelegt werden, dabei wird die Opportunity direkt dem Account zugeordnet. Dafür steht in der Detailansicht des Accounts ein Button *add Opportunity* zur Verfügung.

Erstellen einer Relationship

Accounts können mit anderen Accounts in einer Beziehung stehen. Zum Beispiel könnte ein Konzern ein oder mehrere Zweigstellen haben. Diese Beziehungen kann man abbilden, indem man eine Relation zu einem anderen Account anlegt. Der Button *add Relationship* leitet den Anwender auf eine Seite, auf der er einen anderen Account auswählen kann und den aktuellen Account mit dem ausgewählten Account und einer Relation (ist Mutter von, gehört zu, arbeitet für, beauftragt) verknüpft. Die Abbildung einer Relation ist stets in beide Richtungen aktiv. So wird zu einer Beziehung „ist Mutter von“ ebenso in die andere

Richtung die Beziehung „gehört zu“ abgebildet.

Hinzufügen eines Contacts

Ein Contact kann direkt über Account-Detailansicht hinzugefügt werden. Dafür steht der Button *add Contact* zur Verfügung.

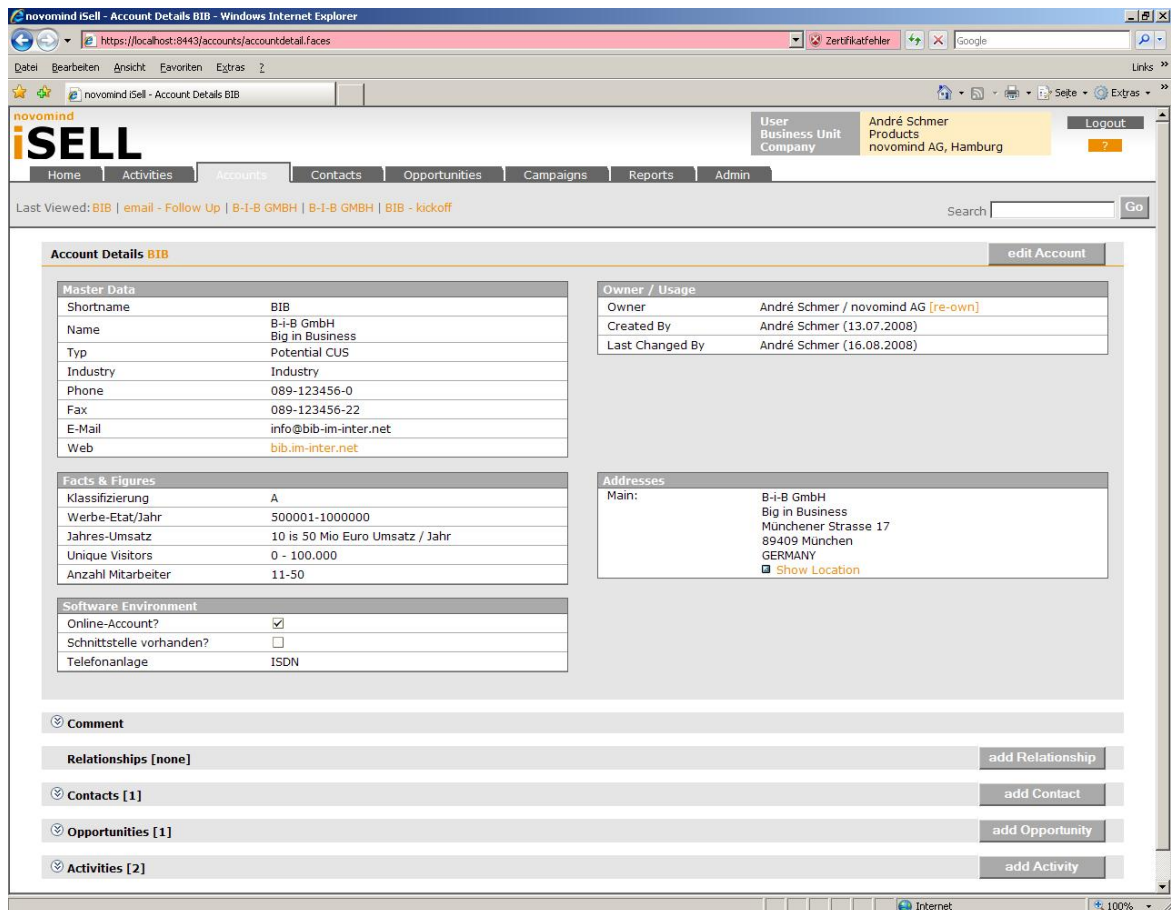


Abbildung 6.5: Account - Detailansicht

6 Realisierung

novomind iSELL

User: André Schmer
Business Unit: Products
Company: novomind AG, Hamburg

Home | Activities | Accounts | Contacts | Opportunities | Campaigns | Reports | Admin

Last Viewed: BIB | email - Follow Up | B-I-B GMBH | B-I-B GMBH | BIB - kickoff

Search [] Go

Edit Account BIB

Fields in **bold** are required

Account Master Data	
Shortname	BIB
Name	B+B GmbH Big in Business
Type	Potential CUS
Industry	Industry
Phone	089-123456 -0
Fax	089-123456-22
E-Mail	info@bib-im-inter.net
Web	http://bib-im-inter.net

Facts & Figures	
Klassifizierung	A
Werbe-Etat/Jahr	500001-1000000
Jahres-Umsatz	10 is 50 Mio Euro Umsatz / Jahr
Unique Visitors	- Choose -
Anzahl Mitarbeiter	10 is 50 Mio Euro Umsatz / Jahr Weniger als 10 Mio Euro Umsatz / Mehr als 50 Mio Euro Umsatz / Jahr

Software Environment	
Online-Account?	<input checked="" type="checkbox"/>
Schnittstelle vorhanden?	<input type="checkbox"/>
Telefonanlage	ISDN

Comment

B-i-B scheint recht finanziell sehr gut dazustehen.
Potential ist vorhanden, könnte für alle Beteiligten eine win-win Situation werden.
Bleibe am Ball.

Abbildung 6.6: Account - Editier-Maske

6.5 Activity-Modul

Das Activity-Modul beinhaltet sämtliche Informationen rund um eine Activity und die Möglichkeit, diese zu pflegen (Abb. 6.7). Activities sind Aufgaben und Ereignisse, die frei beschrieben werden können. Activities werden durch einen Activity Type, einem Buying Interest und einem Shortname beschrieben. Ausserdem können sie einem oder mehreren Contacts, sowie einem Mitarbeiter zugewiesen werden und haben einen Erledigtstatus. Activities werden in den verknüpften Datensätzen (Accounts, Contacts, Opportunities) angezeigt und von dort auch angelegt.

Die Daten einer Activity sind der Übersichtlichkeit halber in Cluster gruppiert, so dass zusammengehörige Daten übersichtlich zusammengefasst sind. Die Cluster sind hier das Cluster *Master Data* für die Stammdaten und das Cluster *Owner/Usage* mit den Informationen über den Inhaber der Activity, sowie Informationen zum letzten Änderungsdatum bzw. zum Erstellungsdatum.

Erreicht wird das Activity-Modul über den Reiter *Activities*. Hier gelangt man auf die Übersicht sämtlicher Activities, mit der Möglichkeit nach bestimmten Activities zu suchen. Die Suche beinhaltet verschiedene Suchkriterien, die miteinander verknüpft werden können. Gesucht werden kann nach Inhaber (Owner), Erledigt (Done), oder nach einem Intervall des Enddatums (End Date). Die Anlegen- bzw. Editier-Ansicht (Abb. 6.8) sind gleich aufgebaut und unterscheiden sich lediglich in der Überschrift.

Die Activities-Detailansicht beinhaltet darüber hinaus Informationen zu dessen Activities, sowie verknüpfte Datensätze der Contacts.

Das Activity-Modul teilt sich in verschiedene Ansichten/Masken auf:

- Activity-Übersicht inkl. Suchmöglichkeiten
- Activity-Detailansicht, Übersicht sämtlicher Informationen zu einer Activity
- Anlegen, bzw. Editier-Maske

Das Activity-Modul bildet folgende Anwendungsfälle ab:

Anlegen einer Activity

Eine Activity wird über den Button *add Activity* aus den verknüpfenden Datensätzen (Account, Contact, Opportunity) heraus angelegt. So wird die Activity direkt dem verknüpfenden Datensatz zugeordnet. Notwendige Felder, die zum Anlegen erforderlich sind, werden fett gedruckt dargestellt, eine entsprechende Validierung dieser Felder weist den Anwender nötigenfalls auf das Fehlen hin.

Bearbeiten einer Activity

Bearbeitet wird eine Activity über den Button *edit Activity* aus der Detailansicht der Activity heraus. Das Bearbeiten erfolgt in gleicher Weise wie das Anlegen einer Activity. Notwendige Felder sind entsprechend markiert und mit Validatoren versehen.

Löschen einer Activity

Eine Activity kann gelöscht werden, wenn diese nicht mehr benötigt wird. Dies geschieht über den Button *delete Activity* aus der Detailansicht.

Ändern des Inhabers (Owner)

Es ist möglich, den Inhaber einer Activity in der Bearbeiten-Maske zu ändern und einem anderen Anwender zuzuweisen.

Hinzufügen einer Follow Activity

Es ist möglich einer zu einer bestehenden Activity eine Follow Activity anzulegen. Eine Follow Activity ist ebenfalls eine Activity, die sämtliche Daten der Mutter-Activity übernimmt.

6 Realisierung

The screenshot displays the 'novomind iSELL' web application interface. The browser window title is 'novomind iSell - Activity details BIB, Partner Contact - email - Windows Internet Explorer'. The address bar shows 'https://localhost:8443/activities/activitydetail.faces'. The application header includes the 'iSELL' logo, a navigation menu (Home, Activities, Accounts, Contacts, Opportunities, Campaigns, Reports, Admin), and a user profile for 'André Schmer' (Products, novomind AG, Hamburg) with a 'Logout' button. Below the header, a breadcrumb trail reads 'Last Viewed: BIB | Partner Contact - email | email - Follow Up | Müller, Oliver Dipl. Betriebswirt | BIB - kickoff'. A search bar is also present.

The main content area is titled 'Activity details BIB, Partner Contact - email' and includes 'edit Activity' and 'delete Activity' buttons. It is divided into two tables:

Master Data	
Shortname	Partner Contact - email
Activity Type	Partner Contact
Buying Interest	email
Account	BIB
End Date	25.07.2008
Done	<input type="checkbox"/>
Confidential	<input type="checkbox"/>

Owner / Usage	
Owner	André Schmer / novomind AG
Created By	André Schmer (13.07.2008)
Last Changed By	André Schmer (13.07.2008)

Below the tables is a 'Contacts [1]' section with a table listing one contact:

Name ↑	Position	Email	Office Phone
Müller, Oliver Dipl. Betriebswirt	Leiter Einkauf	o.muller@b-i-b.de	089-123456-7

A 'Comment [André Schmer (13.07.2008 15:14)]' section contains the text: 'Kunde erbittet um Zusendung von Informationsmaterial zu iMail. Zusendung alsbald, spätestens bis zum 25.7.2008.'

The footer shows 'novomind iSell v1.0-000' and '2008-07-04 13:00' on the left, and '© 2008 novomind AG' on the right.

Abbildung 6.7: Activity - Detailansicht

6 Realisierung

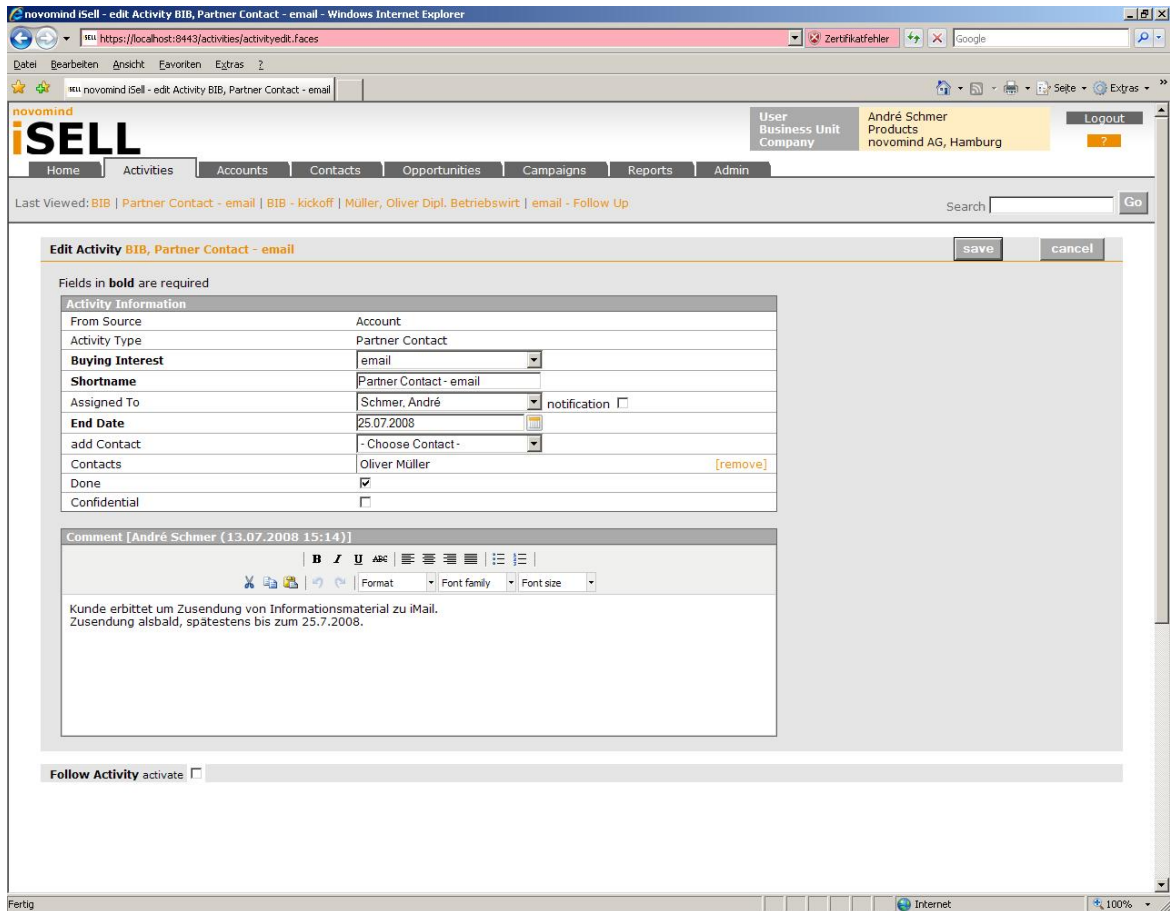


Abbildung 6.8: Activity - Editier-Maske

6.6 Opportunity-Modul

Das Opportunity-Modul beinhaltet sämtliche Informationen rund um eine Opportunity und die Möglichkeit, diese zu pflegen (Abb. 6.9). Opportunities sind die Umsätze und noch in der Schwebe befindlichen Geschäfte. Eine Opportunity besteht aus den sie beschreibenden Daten, die Stammdaten der Opportunity. Die Daten einer Opportunity sind der Übersichtlichkeit halber in Cluster gruppiert, so dass zusammengehörige Daten übersichtlich zusammengefasst sind. Die Cluster sind hier das Cluster *Master Data* für die Stammdaten und das Cluster *Owner/Usage* mit den Informationen über den Inhaber der Opportunity, sowie Informationen zum letzten Änderungsdatum bzw. zum Erstellungsdatum. Opportunities haben eine *Stage*, die Phase in der sich eine Opportunity befindet (*AK - Akquisition*, *AN - Angebot*, *AU - Auftrag*, *LO - Lost Order*) und eine *Probability*, die den Fortschritt bzw. die Abschlusswahrscheinlichkeit des möglichen Geschäfts darstellt.

Erreicht wird das Opportunity-Modul über den Reiter *Opportunities*. Hier gelangt man auf die Übersicht sämtlicher Opportunities, mit der Möglichkeit nach bestimmten Opportunities zu suchen. Die Suche beinhaltet verschiedene Suchkriterien, die miteinander verknüpft werden können. Gesucht werden kann nach Inhaber (Owner), Status (Stage), oder nach einem Intervall des Abschlussdatums (Closing Date). Es kann auch eine Auflistung sämtlicher Opportunities des aktuellen, der letzten drei oder zukünftigen Quartale durchgeführt werden. Die Anlegen- bzw. Editier-Ansicht (Abb. 6.10) sind gleich aufgebaut und unterscheiden sich lediglich in der Überschrift.

Opportunities werden ausserdem in den verknüpften Datensätzen (Accounts, Contacts) angezeigt.

Die Opportunity-Detailansicht beinhaltet darüber hinaus Informationen zu dessen Activities, sowie etwaige Buying Center und ein Kommentar.

Das Opportunity-Modul teilt sich in verschiedene Ansichten/Masken auf:

- Opportunity-Übersicht inkl. Suchmöglichkeiten
- Opportunity-Detailansicht, Übersicht sämtlicher Informationen zu einer Opportunity
- Anlegen, bzw. Editier-Maske

Das Opportunity-Modul bildet folgende Anwendungsfälle ab:

Anlegen einer Opportunity

Eine Opportunity wird über den Button *add Opportunity* aus den verknüpfenden Datensätzen (Account, Contact) heraus angelegt. So wird die Opportunity direkt dem verknüpfenden Datensatz zugeordnet. Notwendige Felder, die zum Anlegen erforderlich

sind, werden fett gedruckt dargestellt, eine entsprechende Validierung dieser Felder weist den Anwender nötigenfalls auf das Fehlen hin.

Bearbeiten einer Opportunity

Bearbeitet wird eine Opportunity über den Button *edit Opportunity* aus der Detailansicht der Opportunity heraus. Das Bearbeiten erfolgt in gleicher Weise wie das Anlegen einer Opportunity. Notwendige Felder sind entsprechend markiert und mit Validatoren versehen.

Löschen einer Opportunity

Eine Opportunity kann gelöscht werden, wenn diese nicht mehr benötigt wird. Dies geschieht über den Button *delete Opportunity* aus der Detailansicht.

Ändern des Inhabers (Owner)

Es ist möglich, den Inhaber einer Opportunity in der Bearbeiten-Maske zu ändern und einem anderen Anwender zuzuweisen.

Hinzufügen einer Activity

Über das Opportunity-Modul kann eine Activity angelegt werden, dabei wird die Activity direkt der Opportunity zugeordnet. Dafür steht in der Detail-Ansicht der Opportunity ein Button *add Activity* zur Verfügung.

Buying Center einer Opportunity hinzufügen

Von der Detail-Ansicht aus, kann man einen oder mehrere Buying Center anlegen. Ein Buying Center ist die Einkaufsstelle eines Unternehmens und besteht aus einer oder mehreren Personen, die in dem Buying Center eine Rollen einnehmen (Buyer, Decider, Gatekeeper, Influencer, User).

6 Realisierung

The screenshot displays the 'novomind iSell' web application interface. The browser window title is 'novomind iSell - Opportunity Details BIB, BIB - kickoff - Windows Internet Explorer'. The address bar shows the URL 'https://localhost:8443/opportunities/opportunitydetail.faces'. The application header includes the 'iSELL' logo, a navigation menu (Home, Activities, Accounts, Contacts, Opportunities, Campaigns, Reports, Admin), and a user profile for 'André Schmer' from 'novomind AG, Hamburg'. Below the header, a breadcrumb trail reads 'Last Viewed: BIB | BIB - kickoff | Partner Contact - email | email - Follow Up | Müller, Oliver Dipl. Betriebswirt'. The main content area is titled 'Opportunity Details BIB, BIB - kickoff' and features two tables: 'Master Data' and 'Owner / Usage'. The 'Master Data' table lists fields such as Shortname, Name, Account, Closing, Order Value, Stage, Probability, and Confidential. The 'Owner / Usage' table lists Owner, Created By, and Last Changed By. Below these tables is a 'Comment' section with a text area containing the text 'Interesse ist sehr gross, unbedingt am Ball bleiben. Info auch an Herrn Dr. Meyer.' At the bottom of the main content area, there are sections for 'Contacts / Buying Center [none]' and 'Activities [none]'. The footer of the application shows 'novomind iSell v1.0-000' and '© 2008 novomind AG'.

Master Data	
Shortname	BIB - kickoff
Name	BIB - kickoff
Account	BIB
Closing	25.08.2008
Order Value	€ 35.000
Stage	AK
Probability	25 %
Confidential	<input type="checkbox"/>

Owner / Usage	
Owner	André Schmer / novomind AG
Created By	André Schmer (13.07.2008)
Last Changed By	André Schmer (13.07.2008)

Comment

Interesse ist sehr gross, unbedingt am Ball bleiben. Info auch an Herrn Dr. Meyer.

Contacts / Buying Center [none] [What does it mean?](#) [add BuyingCenter](#)

Activities [none] [add Activity](#)

Abbildung 6.9: Opportunity - Detailansicht

6 Realisierung

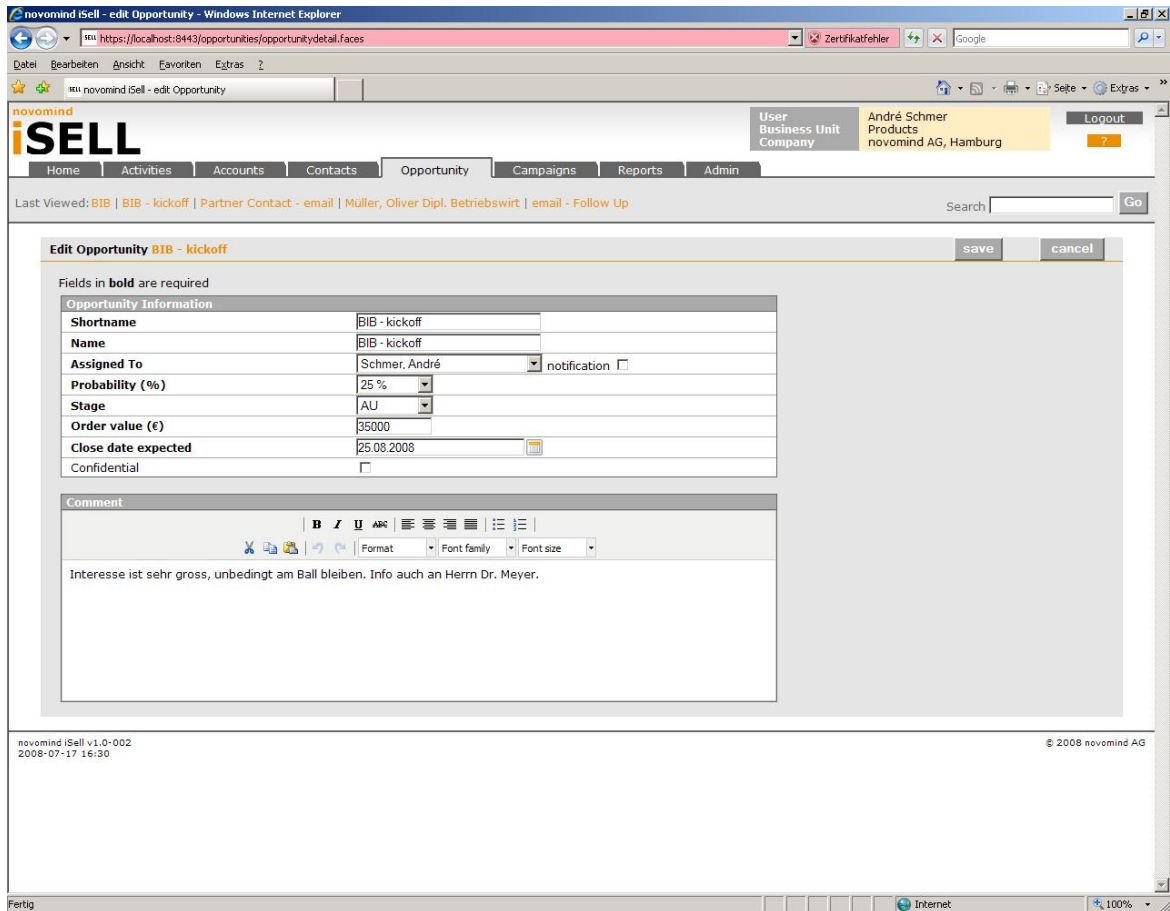


Abbildung 6.10: Opportunity - Editier-Maske

6.7 Administrations-Modul

Eine Besonderheit des Systems ist es, dass man einige Komponenten administrieren kann. So kann ein Administrator Einfluss auf die Werte einiger Tabellen nehmen oder den Zugang zum System steuern. Im Folgenden sollen die administrierbaren Komponenten der Version V1.0 des Vertriebssystem vorgestellt werden.

User (Benutzerverwaltung)

Über die Benutzerverwaltung wird maßgeblich der Zugang zum System geregelt. Und zwar geschieht das in der Weise, dass Benutzer angelegt werden können und diese mit Rechten bzw. Rollen versehen werden, um den Zugang zum System zu steuern (Abb. 6.11).

In der Benutzerverwaltung werden die Benutzer des Systems angelegt und bearbeitet. Die Zugangskontrolle für die Benutzer erfolgt durch Authentifikation mittels Benutzernamen und Passwort, die in der Verwaltung hinterlegt werden. Zu den Daten eines Benutzers gehören ebenso der vollständige Name und Kontaktdaten wie Telefon oder E-Mail Adresse. Außerdem wird die Zugehörigkeit innerhalb des Unternehmens eingestellt, in dem das Unternehmen, der Geschäftsbereich und die Lokation ausgewählt werden können. Die Verwaltung eines Benutzers schließt damit ab, dass dem Benutzer noch mindestens eine Rolle zugewiesen werden muss, die ihn schließlich im System autorisiert. Außerdem kann dem Benutzer seine Reichweite gemäß des Sichtbarkeitskonzepts (siehe Kapitel 5.1.3) eingestellt werden, so hat er z.B. Zugang zu benutzerdefinierten Feldern, die nicht in seinem Geschäftsbereich liegen und für einen unternehmensweiten Zugriff auch nicht vorgesehen wurden. Dies kann z.B. für Projekt- oder Bereichsleiter sinnvoll sein.

Die Benutzerverwaltung bildet folgende Anwendungsfälle ab:

- Anlegen eines Benutzers
- Bearbeiten eines Benutzers
- Löschen eines Benutzers
- Zuteilen von Rollen

Roles (Rollen-/Rechteverwaltung)

In der Rollen-/Rechteverwaltung ist es einem Administrator möglich, Rollen zu verwalten und diese mit Rechten zu versehen. Die Rechte des Systems sind die Zugangssteuerungen zu Modulen bis hin zu einzelnen Feldern. Sie steuern den Zugang der Sichtbarkeit und die zur Verfügungsstellung von Funktionen, die die Bearbeitung der Daten gewähren oder unterbinden. Die Rechte sind systemweit platziert und werden in der Rollen- und Rechteverwaltung lediglich zu Rollen zusammengefasst.

6 Realisierung

The screenshot displays the 'iSELL' user management interface. The top navigation bar includes 'Home', 'Activities', 'Accounts', 'Contacts', 'Opportunities', 'Campaigns', 'Reports', and 'Admin'. The user profile 'André Scher' is shown in the top right corner. The main content area is titled 'Edit User: André Schmer' and contains a form with the following fields:

Step 1: Input Main Data	
Firstname	André
Lastname	Schmer
Email	aschmer@novomind.com
Birthday	
Username	aschmer
New Password	••••••••
Confirm Password	••••••••
Valid from	16.03.2008
Valid until	

Fields in **bold** are required

Company	testcompany GmbH
Business Unit	Technik
Location	Hauptstrasse 66
Zip/City	20255 Hamburg
Country	Germany
or Choose a different	20255 Hamburg, Hauptstrasse
Created By	
Last Changed By	André Schmer (17.08.2008)

cancel go to step 2 >>

novomind iSELL v1.0-000
2008-07-04 13:00

Abbildung 6.11: Benutzerverwaltung

Je nach Aufgabengebiet kann man so Rollen definieren, die exakt auf die Anforderungen zugeschnitten sind. Auf die so angelegten Rollen kann in der Benutzerverwaltung zugegriffen und an die Benutzer verwaltet werden (Abb. 6.12).

Die Rollen-/Rechteverwaltung bildet folgende Anwendungsfälle ab:

- Anlegen von Rollen und Zuteilung von Rechten
- Bearbeiten von Rollen

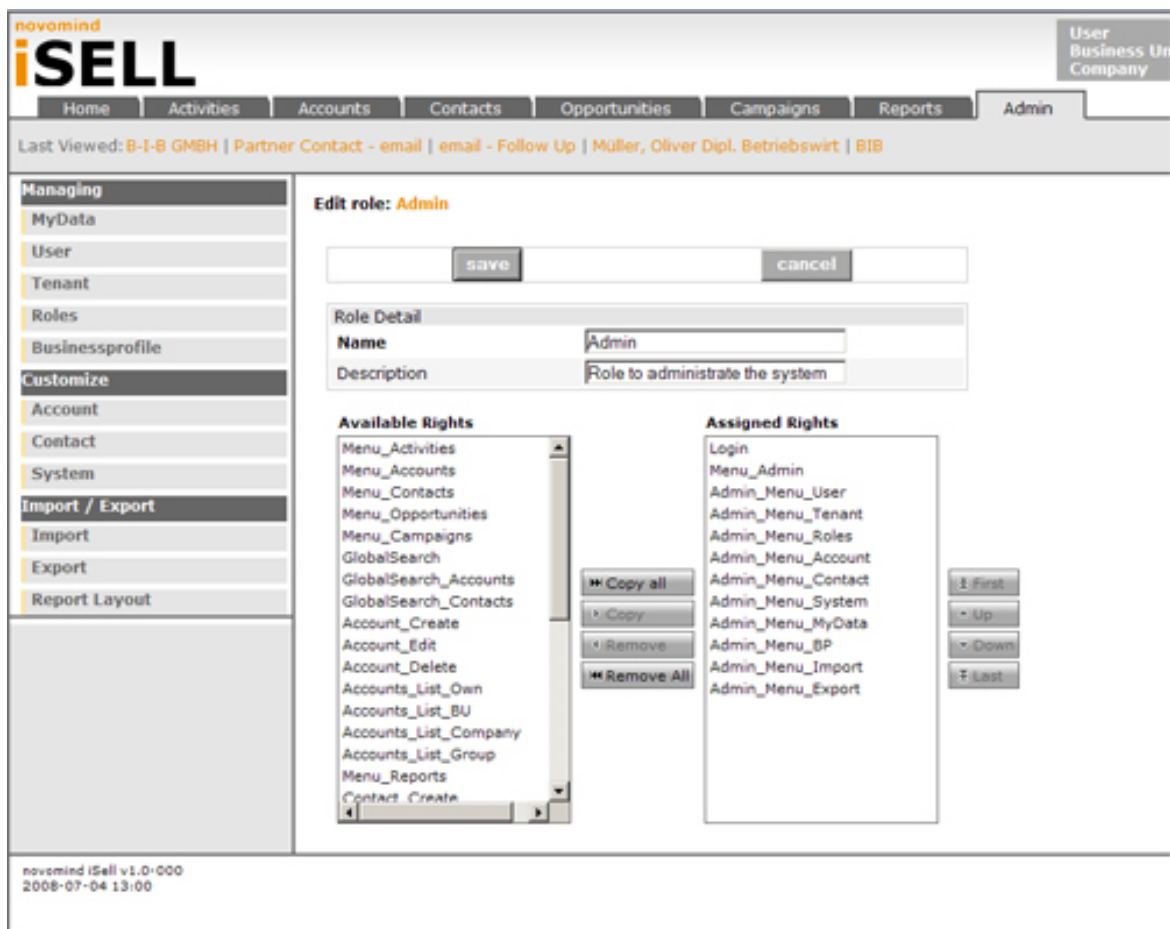


Abbildung 6.12: Rollenverwaltung

Businessprofile (Unternehmensprofil)

Unternehmen sind oftmals eingeteilt in verschiedene Bereiche oder Abteilungen, die helfen, die Organisation des Unternehmens zu vereinfachen. Diese Aufteilung wird in einer Struktur zusammengefasst, die in diesem System unter die Bezeichnung Unternehmensprofil (Businessprofil) fällt (siehe Kapitel 5.1.3). Das System unterscheidet *Group* (Gruppe),

Company (Unternehmen) und Business Unit (Geschäftsbereich) (Abb. 6.13).

Die Verwaltung des Businessprofils bildet folgende Anwendungsfälle ab:

- Anlegen von Gruppen
- Bearbeiten von Gruppen
- Anlegen von Unternehmen und Lokationen innerhalb der Gruppen
- Bearbeiten von Unternehmen und Lokationen innerhalb der Gruppen
- Anlegen von Business Units innerhalb der Unternehmen
- Bearbeiten von Business Units innerhalb der Unternehmen

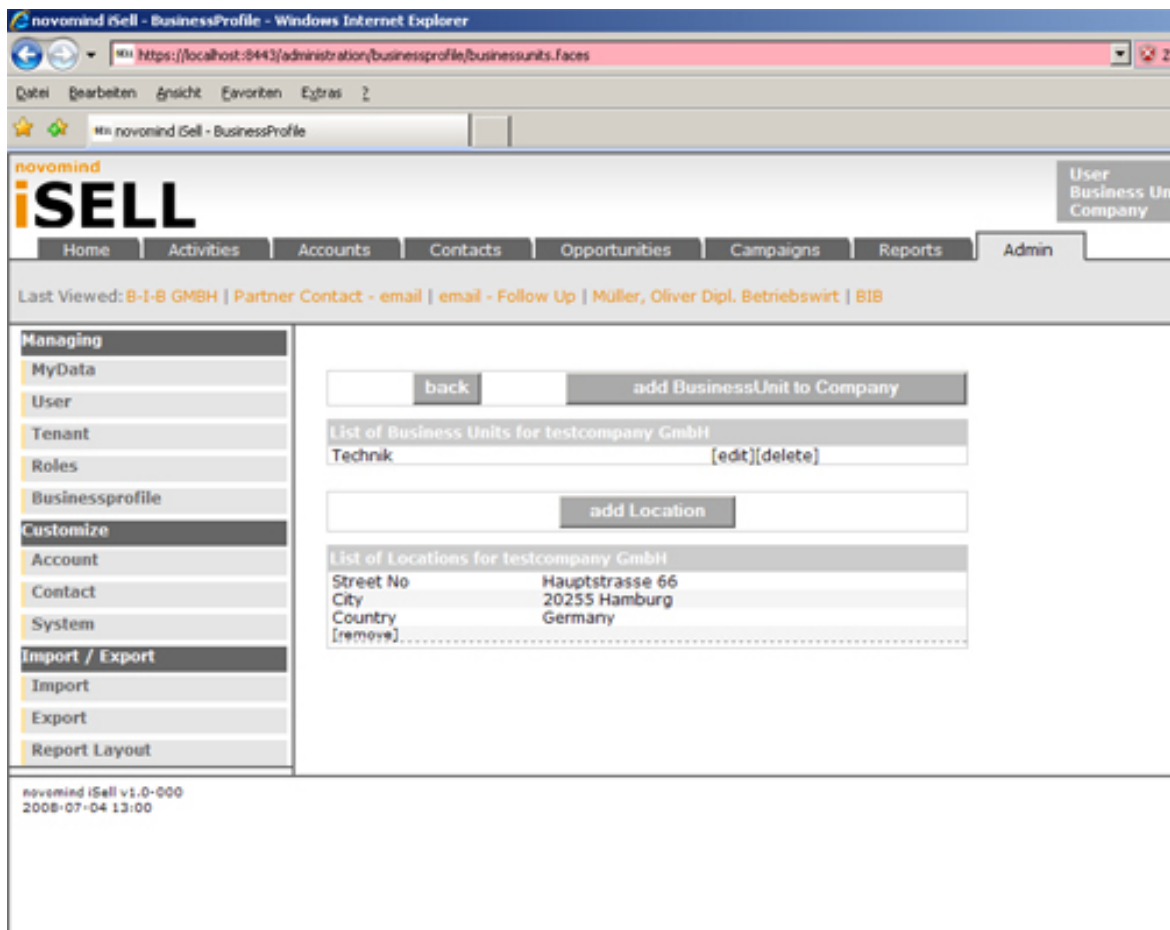


Abbildung 6.13: Businessprofilverwaltung

Customizing (Benutzerdefinierte Felder und Cluster)

Um den Anspruch aus Kapitel 5.1.2 "Generisches Datenmodell" gerecht zu werden, benutzerdefinierte Felder und Cluster anlegen zu können, muss eine Möglichkeit geschaffen werden, diese benutzerdefinierbaren Einheiten verwalten zu können. Dies geschieht in dem Bereich Customizing sowohl für Accounts als auch für Contacts (Abb. 6.14).

So ist es einem Administrator möglich, für die erwähnten Entitäten Accounts und Contacts, Felder anzulegen, die sich aus einem Datentyp, einer Bezeichnung und einem etwaigen vorgelegten Wert zusammensetzen. Die Felder können darüber hinaus in Cluster gruppiert werden und sich dem Sichtbarkeitskonzepts (siehe Kapitel 5.1.3) unterwerfen, das es ermöglicht, die Felder nur für bestimmte Unternehmensbereiche sichtbar und zugänglich zu machen.

Analog zu den Feldern können auch benutzerdefinierte Cluster für Accounts und Contacts angelegt werden. Diese haben eine Bezeichnung und unterliegen ebenso dem Sicherheitskonzept aus Kapitel 5.1.3.

Folgende Anwendungsfälle bildet der Customizing Bereich ab:

- Anlegen von Generischen Feldern
- Bearbeiten von Generischen Feldern
- Löschen von Generischen Feldern
- Einstellen der Sichtbarkeit
- Anlegen von Clustern zur Gruppierung
- Bearbeiten von Clustern
- Löschen von Clustern
- Einstellen der Sichtbarkeit

My Data (Meine Daten)

Einem eingeloggtten Benutzer muss es möglich sein, sein Passwort zu bearbeiten. Dieses wird im Bereich My Data zur Verfügung gestellt (Abb. 6.15).

Folgender Anwendungsfall wird hier abgebildet:

- Bearbeiten des eigenen Passwortes

The screenshot displays the 'Account Cust Field Edit' configuration page for a field named 'Online-Account?'. The interface includes a navigation menu on the left, a breadcrumb trail at the top, and a main configuration area with various input fields and buttons.

novomind iSELL | User: Business Compa

Home | Activities | Accounts | Contacts | Opportunities | Campaigns | Reports | Admin

Last Viewed: 8-I-B GMBH | Partner Contact - email | email - Follow Up | Müller, Oliver Dipl. Betriebswirt | BIB

Managing

- MyData
- User
- Tenant
- Roles
- Businessprofile

Customize

- Account
- Contact
- System

Import / Export

- Import
- Export
- Report Layout

Account Cust Field Edit: Online-Account?

save cancel

Fields in **bold** are required

Label	Online-Account?
Id	online-account
Data Type	checkbox
Default Value	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
in Cluster	Software Environment
Access Level	BusinessUnit
Access Scope	testcompany GmbH/Technik

novomind iSell v1.0-000
2008-07-04 13:00

Abbildung 6.14: Administration eines Generischen Feldes

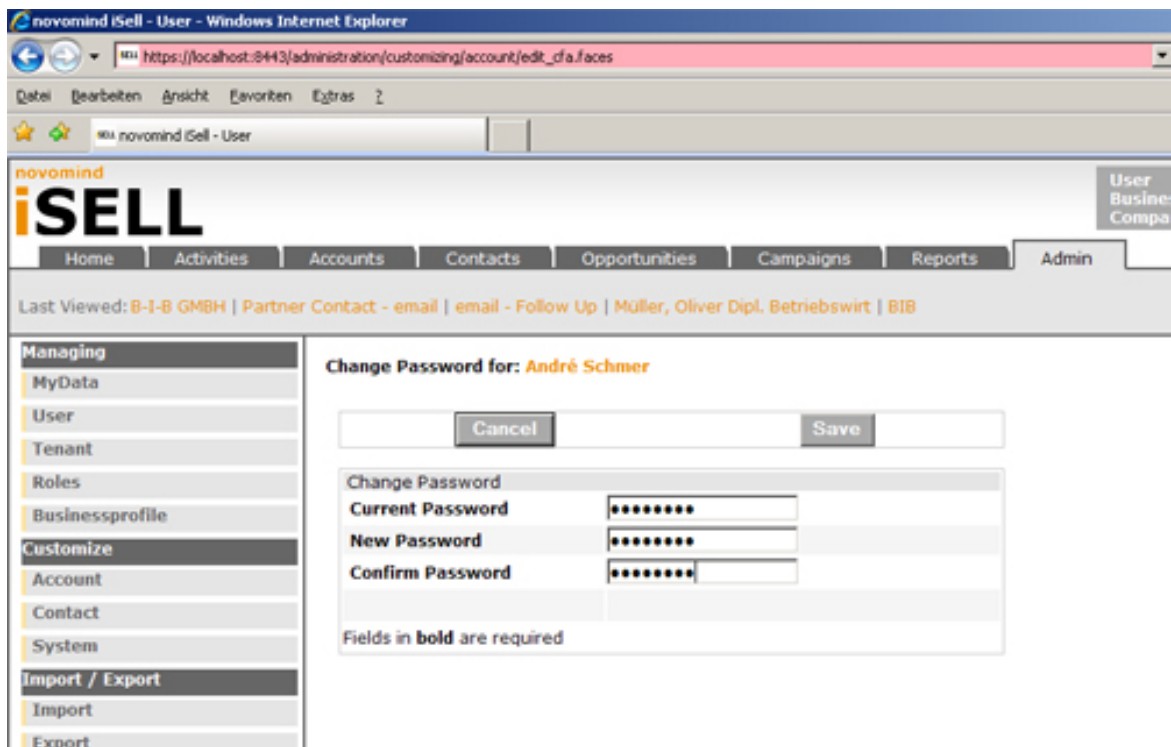


Abbildung 6.15: Meine Daten

7 Evaluierung

Ein großes Thema bei der Entwicklung von Anwendungssoftware und letztendlich ausschlaggebend für eine erfolgreiche Einführung eines Softwareprodukts, ist die Akzeptanz des Produkts bei den zukünftigen Benutzern und denen, die das System warten und weiterentwickeln werden.

Sowohl auf Seite der Benutzer eines Softwaresystems als auch auf der Seite der Anwendungsentwickler kann die Akzeptanz gemessen und sichergestellt werden, indem Verfahren zur Überprüfung der Qualität eingesetzt werden.

Bei den in dieser Arbeit angewandten Verfahren handelt es sich um eine Architekturanalyse mit Hilfe eines Softwareprogramms, das den Code untersucht und um einen Test, der die Software auf ihre Gebrauchsqualität hin untersucht, den so genannten Usability-Test.

Softwarequalität ist die Gesamtheit der Eigenschaften und Merkmale, die Software in Verwendung und (Weiter-) Entwicklung aufweist, um die an sie gestellten Anforderungen zu erfüllen (vgl. DIN, 1987–07).

7.1 Vorgehensweise

Bei der Evaluierung des hier beschriebenen Vertriebssystems werden zwei Ansätze betrachtet. Zum einen wird eine Evaluierung der Architektur durchgeführt. Dies dient primär der Qualitätssicherung nach Innen. Zum anderen wird die Anwenderakzeptanz festgestellt, in dem ein Usability-Test im Labor durchgeführt wird.

7.2 Architekturanalyse

Eine Software-Architektur ist ein wesentliches Merkmal für die Qualität eines Softwareprodukts. Sie bildet das Grundgerüst einer Software auf dem schliesslich sämtliche Komponenten aufbauen. Wenn das Fundament wackelt, kann der Aufbau nicht stabil sein.

Für die Analyse der Architektur wird ein spezielles Programm eingesetzt. Der Software-Tomograph *Sotograph* ist ein kommerzielles Produkt der Firma Software Tomography GmbH. Mit dieser Software ist es möglich automatische Codeanalysen durchzuführen. Es übernimmt das Architekturmodell in Form der Java-Klassen und leitet daraus die Architektur ab.

Eingesetzte integrierte Entwicklungsumgebungen wie etwa Eclipse, bieten nur oberflächliche Unterstützung für die Analyse. Zwar gibt es auch hier Programme, die man einbinden kann und die den Code analysieren, allerdings weicht das Ergebnis stark von einer brauchbaren Architekturanalyse ab, da diese nur auf Code-Ebene stattfinden. Ein Architektur-Entwurf findet jedoch auf einer höheren Ebene als der des Codes statt und dort sollte auch die Analyse durchgeführt werden.

Mit Hilfe des Sotographen ist es möglich, detaillierte Struktur-, Qualitäts- und Abhängigkeitsanalysen auf verschiedenen Abstraktionsebenen durchzuführen. Insbesondere können auch Zyklen analysiert und visualisiert sowie duplizierter Code identifiziert werden.

Der Sotograph kann darüber hinaus Metriken erstellen und somit kritische Codestellen identifizieren. Ebenso lassen sich duplizierte Codeblöcke und unerwünschte Zyklenstrukturen auffinden (vgl. Sotograph, 2008).

Als Bestandteil der Architekturanalyse sind Metriken aufgestellt worden, die die Anzahl der Java-Klassen und Interfaces sowie deren Aufteilung in Packages ermitteln, sowie Metriken zu Codezeilen Lines of Code (LOC), zyklomatische Komplexität, ein- und ausgehende Beziehungen der verschiedenen Klassen sowie die Größen der Klassen erfassen.

Abbildung 7.1 zeigt sehr anschaulich die Schichten und deren Package-Struktur. Außerdem kann man an den rechtsseitigen Verbindungen erkennen, dass es nur wenige Zyklen auf Package-Ebene gibt, bzw. auf Schichtenebene sogar gar keine.

Bad-Smells

Zyklische Beziehungen zwischen Klassen führen dazu, dass diese enger aneinander gekoppelt sind und als Einheit betrachtet werden müssen, was die Übersichtlichkeit und Wartung erschwert, da so nicht gewährleistet werden kann, dass Änderungen nicht ungewollte Nebeneffekte erzielen.

Zyklomatische Komplexität ist ein Hinweis auf die Verzweigungen, die innerhalb einer Methode möglich sind. Eine hohe zyklomatische Komplexität ist ein Zeichen für aufwändige Wartbarkeit und schlechte Testbarkeit. Die Grenze liegt bei 10 (vgl. McCabe, 1976).

Die wichtigsten ermittelten Metriken des Systems im Einzelnen:

Anzahl Java Klassen: 119

Anzahl Interfaces: 22

7 Evaluierung

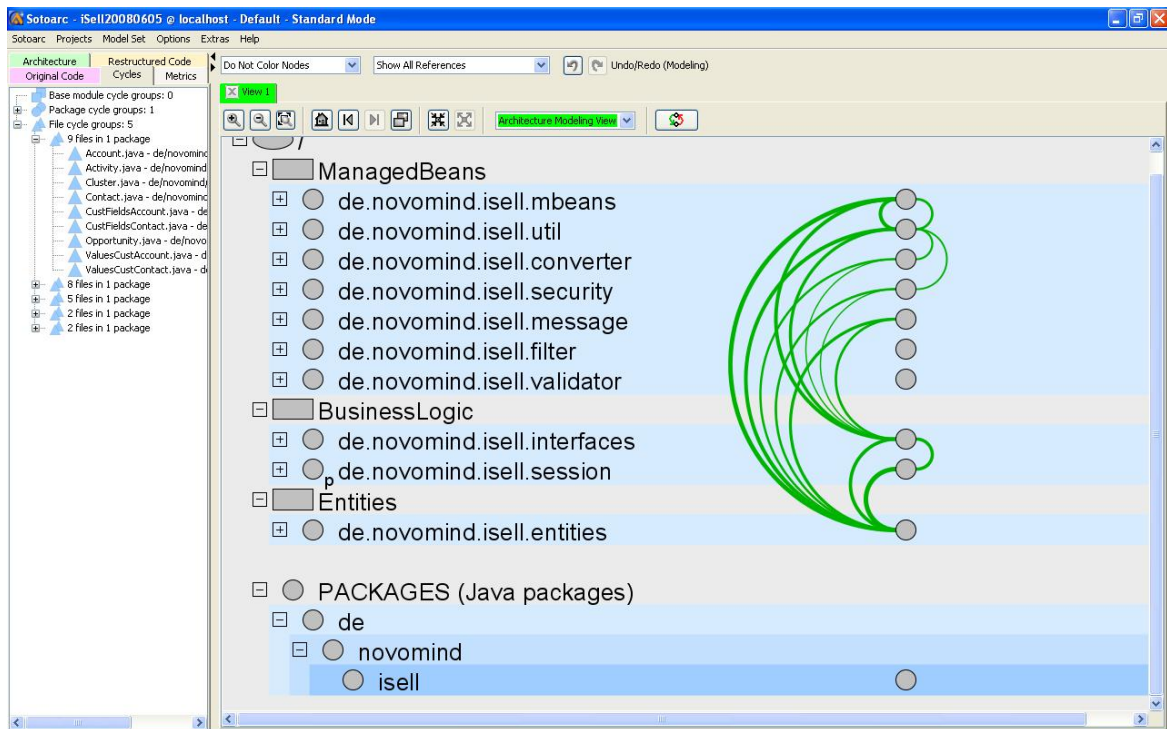


Abbildung 7.1: Drei-Schichten der Architektur

Code Zeilen (LOC): 22000

Klassen mit LOC > 1000: 2

Anzahl der Packages: 19

Anzahl Methoden: 2145

Anzahl Methoden mit einer zyklomatischen Komplexität > 10: 21

Anzahl Klassen, die an Zyklen beteiligt sind: 24

7.3 Usability-Test

Gegenstand des Usability-Tests

Getestet werden soll das vorliegende Vertriebssystem. Bei Durchführung des Usability-Tests befindet sich das Vertriebssystem in der finalen Version V1.0 und ist bereits bei der novomind AG eingeführt.

Ziel des Usability-Tests

Dieser Usability-Test findet im Rahmen der Qualitätssicherung statt. Da der Anwender im Mittelpunkt bei der Entwicklung und der Einführung dieses Systems steht, ist ein Akzeptanztest Maßstab für den Erfolg eines Projekts. Im Vordergrund steht bei diesem Test die Gebrauchsqualität des vorliegenden Vertriebssystems.

Planung und Durchführung

Der durchgeführte Test ist als ein nichtfunktionaler Test klassifiziert. Während ein funktionaler Test das System in Bezug auf funktionale Qualitätsmerkmale wie z.B. Korrektheit und Zuverlässigkeit hin überprüft, werden bei einem nichtfunktionalen Test lediglich jene Qualitätsmerkmale untersucht, die die Benutzbarkeit des Systems beinhalten. Es steht also nicht das „was tut die Software“ im Vordergrund, sondern das „wie tut die Software etwas.“

Der Test ist in zwei Teile aufgeteilt. Er wurde über mehrere Tage und mit 7 Versuchsteilnehmern durchgeführt. Damit liegt der Usability-Test mit der Anzahl an Testpersonen im Rahmen wie er von der Literatur empfohlen wird (vgl. Schweibenz und Thissen, 2003). Der erste Teil beinhaltete den Test an der Software in dem die Teilnehmer eine Aufgabe in 9 Schritten abzuarbeiten hatten, die sie vorher nicht kannten. Dieser Teil endete mit einem Nachgespräch. Im Nachgespräch wurden zunächst die Auffälligkeiten seitens der Versuchsleitung angesprochen, sowie die Ursache geklärt. Außerdem hatte der Teilnehmer die Gelegenheit, seinerseits Bemerkungen sowohl positiver als auch negativer Natur zu äußern und Vorschläge zur Verbesserung anzubringen. Der zweite Teil beinhaltet das Ausfüllen eines Fragebogens nach ISONORM 9241/10 zur Beurteilung von Software auf Grundlage der Internationalen Ergonomie-Norm.

Es wurden bewusst Teilnehmer ausgesucht, die sich zwar im Umgang mit dem Internet und der Bedienung eines Browsers auskennen, darüber hinaus aber über wenig bis keinen betriebswirtschaftlichen Hintergrund bzw. Kenntnisse eines Vertriebs- bzw. CRM Systems verfügen. Hintergrund: die Klassifizierung als nichtfunktionalem Test.

Der Test wurde im Usability Labor der Hochschule für Angewandte Wissenschaften Hamburg durchgeführt und digital sowohl als Videodatei als auch als Audiodatei dokumentiert.

Pilot-Test

Den eigentlichen Tests ging ein so genannter Pilot-Test oder auch Pre-Test voraus. Dieser als „Test für den Test“ ausgelegte Durchgang diente dem Zweck, herauszufinden, ob die Aufgabe für einen branchenfremden Probanden so durchführbar ist und ob der angestrebte zeitliche Aufwand ausreichen würde.

Erwartungen

Im Laufe der Entwicklung des Systems stellte sich heraus, dass die Benutzbarkeit des Vertriebssystems ein sehr wichtiges Thema für die Akzeptanz der Anwender ist. So ergaben sich verschiedene Schlüsselstellen im System, die Bedarf an Diskussionen innerhalb der Gruppe der zukünftigen Anwender erforderten. Es konnten folgende Schlüsselstellen identifiziert werden von denen erwartet wird, dass diese ebenfalls als Hindernis im Usability-Test in Erscheinung treten.

Adresseingabe in Accounts und Contacts

Hier ist die Eingabe der Adresse nur über ein Formular möglich, das vorher aktiviert werden muss. Die Aktivierung geschieht durch einen Hyperlink, der im Rahmen der Account-Erfassen-Maske eher unwahrscheinlich wirkt und leicht übersehen werden kann.

Erfassen einer Follow Activity

Ähnlich wie bei der Adresseingabe muss eine Follow Activity aktiviert werden, in dem man eine bestehende Activity aufruft, in den Edit-Modus wechselt und dort einen Haken setzt. Dieses Element ist im Kontext der Editierfunktion ebenfalls nicht offensichtlich.

Editieren eines Datensatzes

Um einen Datensatz im vorliegenden System im Allgemeinen editieren zu können, muss man den Datensatz aufrufen. In dieser Phase ist ein Editieren allerdings noch nicht möglich. Erst nach einem Klick auf einen entsprechenden Button, gelangt der Anwender in den Edit-Modus. Man kann davon ausgehen, dass für gewöhnlich erwartet wird, einen Datensatz editieren zu können, wenn man diesen bereits aufgerufen hat.

7.3.1 Ergebnis des Usability-Tests

Ergebnis des Pilot-Tests

Das Ergebnis des Pilot-Tests war durchweg positiv. So wurde der angestrebte Zeitrahmen eingehalten und die Aufgabe war auch für eine branchenfremde Person durchführbar. Lediglich ein paar kleinere Änderungen bzgl. der zu verwendenden Daten in der Aufgabenstellung mussten angepasst werden.

Ergebnis der Teilnehmer-Tests

Die wohl auffälligste Beobachtung der Teilnehmer bei ihrer Durchführung der Aufgabe war, dass lediglich einer von 7 Teilnehmer (14,3%) auf Anhieb einen Bezug zwischen Contact und Account herstellen konnten. So interpretierten die Teilnehmer vermeintliche Daten des Contacts wie z.B. dessen E-Mail-Adresse als Daten des Accounts und trugen diese in das für sie am ehesten passende Feld ein. Da aber ausschließlich Teilnehmer rekrutiert wurden, die keinerlei Vorkenntnisse mit einem Vertriebssystem hatten, ist dieses Phänomen auf die mangelnde Erfahrung mit solchen Systemen zurückzuführen. Im Nachgespräch gaben die Teilnehmer auch als Grund dafür an, „sich erst einmal in die Materie einarbeiten zu müssen“.

Als weitere auffällige Beobachtung ist das Übersehen des Adressbereiches bei den Accounts zu erwähnen. Lediglich 2 von 7 Teilnehmer (28,6%) haben den Bereich gefunden in dem die Adresse(n) zu einem Account eingetragen werden konnten. Im Nachgespräch gaben die Teilnehmer an, die Information nicht zur Verfügung gehabt zu haben, bzw. den Zugang zum Bereich, der nur durch einen Hyperlink in der Erfassen-Maske eines Accounts zugänglich war, schlichtweg übersehen zu haben.

Durchweg positiv (100%) beurteilten die Teilnehmer auf Nachfrage den Wiedererkennungswert bei der Durchführung der einzelnen Aufgaben durch die einheitliche Gestaltung der einzelnen Masken. So gaben sie auch an (100%), sich gut zurecht zu finden.

Die Gestaltung der Workflow-Unterstützung, in dem die zugehörigen Angaben zu einem Account, wie Erfassen eines Contacts, Erfassen einer Activity und Erfassen einer Opportunity, von der Detailseite eines Accounts aus vorgenommen werden können, haben 4 von 7 Teilnehmer (54,14%) übersehen und stattdessen die auffälligere Variante über die Reiter gewählt. So scheiterten sie allerdings bei der Erfassung der Activity und der Opportunity, da diese nur im Kontext eines Accounts, wie in dieser Aufgabe, oder eines Contacts angelegt werden können. Im Nachgespräch bestätigten die Teilnehmer auf Nachfrage auch die Vermutung, es sei intuitiver über die Reiter zu gehen.

Auf die Frage, was ihnen maßgeblich negativ an der Software auffiel, antworteten 3 von 7 Teilnehmern (42,8%), dass ihnen die Software nur ungenügend Rückmeldung darüber gebe, ob etwa eine Follow Activity erfolgreich angelegt wurde oder nicht. Der Rest der

Teilnehmer (58,2%) hatte nichts Auffälliges zu bemängeln.

Auswertung der Fragebögen

Unterteilung in Fragenkomplexe

Die Fragen des Fragebogens sind in sieben Fragenkomplexe aufgeteilt. Innerhalb der Fragenkomplexe gibt es verschiedene Bewertungskriterien, um den Komplex zu beurteilen. So sind folgende Bewertungen möglich: [- - - , - - , - , - / + , + , + + , + + +].

Gelesen werden die Diagramme wie folgt: Die Y-Achse stellt die Anzahl der Teilnehmer dar, die sich für ein Bewertungskriterium entschieden haben. Die X-Achse zeigt die möglichen Bewertungen an.

Die Auswertung der Fragebögen fasst die einzelnen Bewertungen in Form von Balkendiagrammen zusammen und ist im Anhang B einsehbar. Die Auswertung erfolgte anonym.

7.4 Fazit

Die Architekturanalyse ergibt auf den ersten Blick einen Architekturentwurf mit kleinen Schönheitsfehlern. Auffällig ist, dass es eine Menge an Klassen gibt, die in Zyklen involviert sind; immerhin wird hier ein Wert von etwa 20% ermittelt. Jedoch muss man diesen Wert mit Vorsicht betrachten. Nicht alles was auf den ersten Blick als Nachteil scheint, ist in der Praxis auch so zu bewerten. So entfallen auf 80% der Klassen, die an Zyklen beteiligt sind, auf Klassen, die als fachliche Objekte implementiert sind, so genannte Persistence Entities des EJB 3.0 Komponentenmodells. Aus Entwicklersicht ist es ein Vorteil auf diese Objekte von den unterschiedlichen Seiten der Kardinalitäts-Endpunkte zugreifen zu können. So betrachtet können diese Zyklen vernachlässigt werden. Die restlichen Zyklen sollte man identifizieren und nach Möglichkeit eliminieren.

Als Empfehlung will ich hier aussprechen, die angesprochenen Zyklen zu beseitigen. Außerdem sind die Klassen bzw. der Code weiterhin zu kommentieren. Die Formulierung von Kommentaren ist ein wesentlicher Bestandteil der Entwicklung und das Weglassen solcher kann das Lesen von Code und schließlich die Wartung erheblich erschweren.

Das Ergebnis des Usability-Test ist eindeutig und lässt keine Zweifel aufkommen, dass die Gestaltung des Vertriebssystems, dem Anspruch der Anwender auf Übersichtlichkeit und Unterstützung in der Bewältigung der Aufgaben gerecht zu werden, durchaus gelungen ist und nur wenig Optimierungspotenzial bietet. So kann man feststellen, dass die meisten Bedienungsschwierigkeiten daher rühren, dass die Teilnehmer sowohl in der Branche als auch in der Bedienung solch eines Systems keinerlei Vorkenntnisse besitzen.

Der Vergleich mit den Erwartungen, die vor dem Test ermittelt wurden, hat sich in zwei der drei Punkte bestätigt. So hatten die Teilnehmer Schwierigkeiten bei der Adresseingabe, da der Link übersehen wurde, die dann auch in den meisten Fällen unberücksichtigt blieb. Bei der Follow Activity erforderte die Aufgabe die Erfassung dieser mit dem Hinweis, in den Edit-Modus einer bestehenden Activity zu wechseln. Hier jedoch brauchten die Teilnehmer eine gewisse Zeit, den Haken zum Aktivieren der Follow Activity zu finden. Der dritte Punkt der Erwartungen wurde problemlos bewältigt. So fanden die Teilnehmer sich recht schnell zurecht und konnten ohne Mühen den Anzeige-Modus vom Editier-Modus unterscheiden.

Als Empfehlung für eine Verbesserung sei die Einführung einer Hilfefunktion genannt und eine noch stärkere Fokussierung auf die Benutzerführung. Darüber hinaus gilt es den Anwender über die Aktivitäten der Software in Kenntnis zu setzen, was diese gerade ausführt oder um mitzuteilen, ob seine Eingaben zu einem positiven bzw. negativen Resultat führen.

8 Methodische Abstraktion

Das vorliegende Vertriebssystem fällt eher unter die Kategorie „Kleinerer Projekte“. Jedoch spielen die Größe eines Projekts und die Größe eines Teams eine untergeordnete Rolle, wenn man sich strikt an eine bestimmte Vorgehensweise hält. Ein Softwaresystem kann in ähnlicher Weise geplant und „gebaut“ werden wie ein Haus. Dort gibt es zunächst die Planer oder Architekten, die einen Entwurf machen. Ein Architekt entwirft einen Bauplan, der das Design des Hauses und seiner Komponenten, etwa die Anordnung und Verteilung der Räume, beschreibt. Er wird stets zur Verfügung stehendes finanzielles Budget berücksichtigen, wie auch rechtliche Vorschriften, ökologische Gesichtspunkte oder Ergonomie der Komponenten. Der Bauingenieur sorgt dafür, dass der Entwurf des Architekten auch umgesetzt wird unter Einhaltung der Ziele und des Budgets. Er plant die Vorgehensweise und setzt seine Ressourcen wie Mitarbeiter oder Material sinnvoll ein.

Ein Softwarearchitekt sollte im Prinzip genauso handeln. Wie ein Haus muss auch ein Softwaresystem geplant und entworfen werden. So wird man sich für Komponenten entscheiden, etwa für die technische Umgebung, die eingesetzt werden. Es wird eine fachliche Architektur entworfen, die schließlich beschreibt, wie die definierten Anforderungen umgesetzt werden sollen. Daraus wird eine technische Architektur abgeleitet, die den Entwurf einzelner Komponenten vorsieht, sowie die Funktionalität festlegt und das Datenmodell entwirft.

Der Entwickler entscheidet sich für eine Vorgehensweise wie in diesem Projekt ein Exploratives/Evolutionäres Prototyping. Er muss die Ressourcen planen, sein Team organisieren, sinnvolle Projektpläne erstellen, Tests durchführen lassen etc. Letztendlich wird dann ein Abnahmeverfahren eingeleitet, mit dem überprüft wird, ob der zwischen Auftraggeber und Auftragnehmer geschlossene Vertrag und damit die gestellten Anforderungen erfüllt wurden.

Für welche Vorgehensweise man sich auch immer entscheidet, wie umfangreich das Projekt mit all seinen Ressourcen auch sein mag, um ein Projekt erfolgreich zu ende zu bringen. Eine ausgiebige Planung des Systems ist unabdingbar. Es mag durchaus sein, dass es Projekte gibt, die auf sehr unkonventionelle Art erstellt wurden und tatsächlich im Einsatz sind, aber man kann dabei nicht von einem erfolgversprechenden Verfahren sprechen.

9 Zusammenfassung

An dieser Stelle soll auf das Projekt und auf seine Durchführung zurückgeblückt werden. Wie ist der Stand der Entwicklung? Wie bewährt sich das Vertriebssystem im alltäglichen Geschäft? Was sind die wichtigsten Erfahrungen, wie ist die Perspektive? All das soll hier reflektiert und zum Gesamtergebnis zusammenfasst werden.

9.1 Stand

Dieses System wurde auf der Grundlage eines Explorativen Prototypen geplant und entwickelt. Die Daten wurden aus dem Altsystem übernommen. Dafür war es notwendig, Strategien zu entwickeln, die die Übernahme der Daten verlustfrei gewährleisten können. Das Datenmodell wurde um ein Generisches Datenmodell ergänzt, mit dem es möglich wurde, das Datenmodell im gewissen Rahmen jederzeit nach Belieben zu erweitern. Es wurde eine Vorgehensweise gewählt, die es ermöglicht, die zukünftigen Anwender und Mitarbeiter des Vertriebs aktiv in den Entwicklungsprozess einzubinden, um so in einen Autor-Kritiker-Zyklus einzutreten.

Damit wurde ein Evolutionärer Prototyp ins Leben gerufen, der sukzessive zur Produktreife entwickelt wurde. Mit der Evaluation wurde die Entwicklung abgeschlossen. Letztendlich hat sich gezeigt, dass sowohl die Architektur einer Überprüfung standhält und das auch ein Akzeptanztest im Usability-Labor erfolgreich verlief.

Das vorliegende, web basierte Vertriebssystem ist mit Abschluss dieser Arbeit erfolgreich bei der novomind AG in der Version V1.0 eingeführt. Es hat wie geplant das alte proprietäre Vertriebssystem abgelöst, die Daten wurden erfolgreich und vor allem verlustfrei übernommen, die Mitarbeiter der Vertriebsabteilung können ihre Arbeit ungehindert fortführen.

Das System läuft stabil und ist unter verschiedenen Betriebssystem-Plattformen getestet worden. Bisher liegt dem Datenmodell eine MySQL-Datenbank zu Grunde, die sich für diese Anforderungen als leistungsfähig genug erwiesen hat.

Die Akzeptanz des Systems ist recht hoch, das Projekt, so das Fazit, wurde erfolgreich zu Ende geführt.

Offene Probleme

Dennoch gibt es immer noch offene Probleme, die aus zeitlichen Gründen nicht gelöst werden konnten. Darunter fallen:

Browserkompatibilität

Die Browserkompatibilität, die unter das Problem der Portierbarkeit fällt und damit ein nichtfunktionales Kriterium in der Spezifikation darstellt, ist ein Aspekt, der einer Lösung bedarf. Zwar läuft das System stabil unter dem Internet Explorer 7, jedoch sollte das System mindestens noch in den gängigen Versionen des Internet Explorers 6 zuverlässig funktionieren, sowie in den Versionen 2 und 3 des Mozilla Firefox.

Datenbankunabhängigkeit

Das Vertriebssystem basiert derzeit noch auf einer MySQL-Datenbank 5.0. Wie Martin Sadowski (Sadowski, 2008) bereits erläuterte, ist dieses System nicht datenbankunabhängig. Den Grund vermutet Sadowski dafür in den SQL-Abfragen und er gibt einen ersten Anhaltspunkt für die Lösung des Problems. Jedoch muss für eine Datenbankunabhängigkeit ein Konzept entwickelt werden, das eine Abstraktionsebene vorsieht, die datenbankunabhängige Abfragen erstellt.

Benutzerdefinierbares Design

Das Design und Layout des Vertriebssystems sind noch sehr stark an das Corporate Design der novomind AG angepasst. Damit das System in der Open Source-Gemeinde auch Anklang findet, sollte das Design soweit anpassbar sein, dass ein Unternehmen seine eigenen Farben und sein eigenes Logo einbringen kann, ohne dass der Code stark verändert werden muss.

9.2 Erfahrungen

Grundsätzlich sind die Anforderungen an das System erfüllt worden. Kompromisse sind bei der Browserkompatibilität gemacht worden. Die Vereinbarung, nur den Internet Explorer 7 zu benutzen, ist zwar für eine gewisse Übergangszeit akzeptabel, jedoch ist es für die Vision des Systems, als Basis für weitreichende Erweiterung zu dienen, nicht tragbar. Auch haben sich einige fremd eingesetzte Komponenten nicht bewährt. So gibt es hin und wieder noch Probleme mit dem „What You See Is What You Get (WYSIWYG)“-Editor, auch ist die Wahl einiger Asynchronous JavaScript and XML (AJAX) Komponenten nicht sehr glücklich. Die Vorgehensweise des Evolutionären Prototyping hat sich als sehr hilfreich erwiesen. Auch

die Entscheidung für JSF als Framework-Standard für Benutzeroberflächen und JBoss als Applicationserver hat sich als erfolgreich erwiesen.

9.3 Ausblick

Das vorliegende Vertriebssystem existiert in der Version V1.0. Mit dieser Version ist ein Anfang für einen weiteren kontinuierlichen Ausbau des Systems geschaffen worden. Grundlegende Funktionalitäten zur Bewältigung der Aufgaben im Bereich eines Vertriebs eines mittelständischen Unternehmens sind umgesetzt.

Um den Anforderungen eines stets wachsenden Marktes und den Innovationen auf dem Technologie-Sektor gerecht zu werden, muss das System stets aktualisiert und mit neuen Komponenten versehen werden. Ein Reporting- und Selektions-Modul, das bereits in Planung ist (Aleaf, 2008), ist ein erster Schritt. Der Wunsch nach Schnittstellen zu anderen Softwarelösungen, wie etwa MS-Office, oder zu Verzeichnisdiensten wie dem Lightweight Directory Access Protocol (LDAP) liegen bereits vor. In den Erweiterungen liegt das Potenzial, das nur abgeschöpft werden muss. Die Implementierung einer Datenbankabstraktionsschicht, mit der es möglich ist, die darunter liegende Datenbank auszutauschen, erscheint in jedem Fall sinnvoll, wenn man bedenkt, das System in die Open Source-Gemeinde zu stellen. Denn auch hier wird letztendlich die Akzeptanz eine nicht zu unterschätzende Rolle spielen.

A Aufgaben des Usability-Tests

Aufgaben für die Usability-Untersuchung Vertriebssystem iSell

Bitte lesen Sie sich die Aufgaben vor Beginn gründlich durch!

Aufgabe

Ziel der Aufgabe ist es, die grundlegenden Bereiche des Vertriebssystems einmal zu betreten und einen kompletten möglichen Workflow zu simulieren.

Situationsbeschreibung

Sie sind der Mitarbeiter Joseph Weinkübler und im Vertrieb eines mittelständischen Unternehmens verantwortlich für einen bestimmten Kundenbereich. Ihr Aufgabenbereich umfasst u.a. die Betreuung von Kunden und die Pflege von Kundenbeziehungen. Um diese Aufgabe bewerkstelligen zu können, ist ein Vertriebssystem entwickelt worden, das Sie darin unterstützt, die Kunden, deren Kontakte und Aktivitäten zu pflegen. So können gezielt Informationen gesammelt und klassifiziert werden und die Beziehungen transparent gestaltet werden. Es unterstützt Sie in der Kommunikation nach Innen und Außen. Alle **fett** gedruckten Inhalte sagen etwas über die Informationen aus, die einzugeben sind.

1. Einloggen in das System

Loggen Sie sich mit folgenden Daten in das System ein.

Benutzername: tester

Passwort: novomind

2. Anlegen eines Accounts

Sie haben auf einer Messe einen Interessenten kennen gelernt. Dessen Visitenkarte wollen Sie nun als neuen *Account*, also als potentiellen Kunden, in das System eingeben. Als *Account* bezeichnet man ein Unternehmen, mit dem man in irgendeiner Beziehung steht.

Sie sehen hier eine typische Visitenkarte auf der der Name des Kontakts zu finden ist sowie die Adressdaten der Firma, für die er arbeitet.

B-i-B GmbH
Big in Business GmbH
Münchener Straße 17
89409 München

Oliver Müller
- Leiter Einkauf -
Tel.: 089 / 123456 - 7
E-Mail: o.müller@b-i-b.de

Legen Sie also zunächst einen Account an.

Folgende zusätzliche Angaben sollten Sie dabei ebenfalls eingeben:

Type: Potential CUS

Industry: Industry

Comment: Tragen Sie hier ein, dass Sie den Eindruck haben, mit B-i-B einen sehr potenten Kunden gewinnen zu können und eine Zusammenarbeit sehr fruchtbar für den Unternehmenserfolg sein könnte.

Address type: Main

Hinweis zur Telefonnummer: Für gewöhnlich ist die Durchwahl eines Accounts, also einer Firma, immer die Zentrale, also die 0, sofern keine anderen Informationen vorliegen. Die Kontakte einer Firma haben dann eine Telefonnummer, die sich aus der Firmennummer und einer Durchwahl für den Kontakt zusammensetzt.

3. Contact erfassen

Erfassen Sie nun im weiteren Schritt zu diesem Account einen *Contact*. Ein *Contact* ist ein Ansprechpartner oder Repräsentant des Unternehmens mit dem Sie sämtliche Geschäfte abwickeln.

Folgende zusätzliche Angaben sollten Sie dabei ebenfalls eingeben:

Buying Interest: email

Comment: Tragen Sie hier ein, dass Sie den Herrn auf einer Messe am

13.07.2008 getroffen haben und er sehr interessiert schien und beeindruckt von Ihrem Unternehmen war.

4. Activity erfassen

Mit einer *Activity* legt man eine Tätigkeit oder einen Vorgang an, um z.B. den Fortschritt oder den Status einer Beziehung zu dokumentieren.

Der Kunde hat darum gebeten, ihm Informationsmaterial zu *iMail*TM zukommen zu lassen. Sie legen dazu eine Activity im System an. Das Informationsmaterial soll bis spätestens den 25.07.2008 beim Kunden sein.

Folgende zusätzliche Angaben sollten Sie dabei ebenfalls eingeben:

Activity Type: Partner Contact

Buying Interest: email

End date: 25.07.2008

Contact: aktuellen Contact auswählen

5. Account bearbeiten

Sie haben durch rege Kontaktpflege in Erfahrung bringen können, dass ihr Kunde ein monatliches **E-Mail-Volumen von etwa 15.000 Emails** und einen **Jahresumsatz von etwa 10 Mio. Euro** hat.

6. Opportunity zum Account erfassen

Eine *Opportunity* in einem Vertriebssystem ist ein potentielles Geschäft. Der Abschluss eines Geschäfts kann unter Umständen ein langwieriger Prozess sein und verschiedene Stufen haben. Der Fortschritt dabei wird in solchen Opportunities festgehalten.

Da der Kunde sehr großes Interesse am Kauf von *iMail*TM hat, wollen Sie eine Opportunity im System anlegen. Die Entscheidung sollte bis zum 25.08.2008 getroffen sein und der voraussichtliche Zahlungseingang wird 35.000 € betragen.

Folgende zusätzliche Angaben sollten Sie dabei ebenfalls eingeben:

Name: BIB kickoff

Probability: 25

Stage: AK (Akquisition)

Order Value: 35000

Close date expected: 25.08.2008

7. Follow-Up Activity erstellen

Der Kunde hat um eine Vorführung des künftigen Produkts *iMail*TM in seinen Geschäftsräumen gebeten. Erfassen Sie also für die oben bereits erfasste Activity eine *Follow Activity*. Eine *Follow Activity* ist ebenfalls eine Activity, nur dass diese sich an eine bereits vorhandene anschließt. Dazu ist es notwendig, in den Edit-Modus der von Ihnen bereits angelegten Activity zu gehen.

Folgende zusätzliche Angaben sollten Sie dabei ebenfalls eingeben:

Activity Type: Follow Up

Shortname: email - Follow Up

Buying Interest: email

End date: 30.07.2008

Contact: der entsprechende Contact

8. Ändern der Phase einer Opportunity

Wir machen jetzt einen Sprung in die Zukunft.

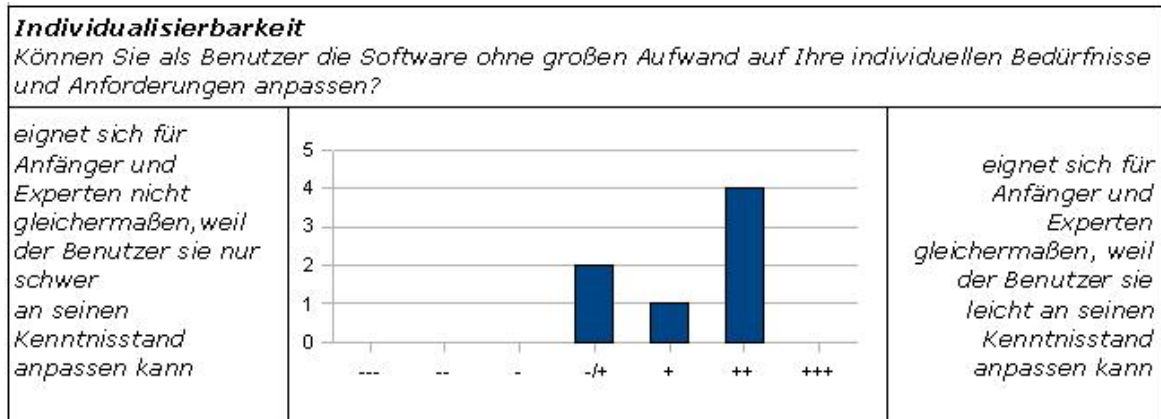
Der Kunde hat sich für Ihr Produkt entschieden.

Sie ändern den **Stage** der Opportunity auf gewonnen (**AU**).

9. Ausloggen aus dem System

Loggen Sie sich bitte aus dem System aus.

B Auswertung der Fragebögen



B Auswertung der Fragebögen

Aufgabenangemessenheit <i>Unterstützt die Software die Erledigung Ihrer Arbeitsaufgaben, ohne Sie als Benutzer unnötig zu belasten?</i>																		
<i>ist kompliziert zu bedienen</i>	<table border="1"> <caption>Data for 'ist kompliziert zu bedienen'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>0</td></tr> <tr><td>-#</td><td>0</td></tr> <tr><td>+</td><td>0</td></tr> <tr><td>++</td><td>6</td></tr> <tr><td>+++</td><td>1</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	0	-#	0	+	0	++	6	+++	1	<i>ist unkompliziert zu bedienen</i>
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	0																	
-#	0																	
+	0																	
++	6																	
+++	1																	
<i>bietet nicht alle Funktionen, um die anfallenden Aufgaben effizient zu bewältigen</i>	<table border="1"> <caption>Data for 'bietet nicht alle Funktionen...'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>0</td></tr> <tr><td>-#</td><td>1</td></tr> <tr><td>+</td><td>0</td></tr> <tr><td>++</td><td>2</td></tr> <tr><td>+++</td><td>4</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	0	-#	1	+	0	++	2	+++	4	<i>bietet alle Funktionen, um die anfallenden Aufgaben effizient zu bewältigen</i>
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	0																	
-#	1																	
+	0																	
++	2																	
+++	4																	
<i>erfordert überflüssige Eingaben</i>	<table border="1"> <caption>Data for 'erfordert überflüssige Eingaben'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>0</td></tr> <tr><td>-#</td><td>0</td></tr> <tr><td>+</td><td>0</td></tr> <tr><td>++</td><td>2</td></tr> <tr><td>+++</td><td>5</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	0	-#	0	+	0	++	2	+++	5	<i>erfordert keine überflüssige Eingaben</i>
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	0																	
-#	0																	
+	0																	
++	2																	
+++	5																	
<i>ist schlecht Anforderungen der Arbeit zugeschnitten</i>	<table border="1"> <caption>Data for 'ist schlecht Anforderungen...'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>0</td></tr> <tr><td>-#</td><td>0</td></tr> <tr><td>+</td><td>1</td></tr> <tr><td>++</td><td>2</td></tr> <tr><td>+++</td><td>4</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	0	-#	0	+	1	++	2	+++	4	<i>ist gut auf die Anforderungen der Arbeit zugeschnitten</i>
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	0																	
-#	0																	
+	1																	
++	2																	
+++	4																	

B Auswertung der Fragebögen

Selbstbeschreibungsfähigkeit Gibt Ihnen die Software genügend Erläuterungen und ist sie in ausreichendem Maße verständlich?		
bietet einen schlechten Überblick über ihr Funktionsangebot		bietet einen guten Überblick über ihr Funktionsangebot
verwendet schlecht verständliche Begriffe, Bezeichnungen, Abkürzungen oder Symbole in Masken und Menüs		verwendet gut verständliche Begriffe, Bezeichnungen, Abkürzungen oder Symbole in Masken und Menüs
liefert in unzureichendem Maße Informationen darüber, welche Eingaben zulässig oder nötig sind		liefert in zureichendem Maße Informationen darüber, welche Eingaben zulässig oder nötig sind
bietet auf Verlangen keine situationsspezifischen Erklärungen, die konkret weiterhelfen		bietet auf Verlangen situationsspezifische Erklärungen die konkret weiterhelfen
bietet von sich aus keine situationsspezifischen Erklärungen, die konkret weiterhelfen		bietet von sich aus situationsspezifische Erklärungen, die konkret weiterhelfen

B Auswertung der Fragebögen

Steuerbarkeit Können Sie als Benutzer die Art und Weise, wie Sie mit der Software arbeiten, beeinflussen?																		
<p><i>bietet keine Möglichkeit, die Arbeit an jedem Punkt zu unterbrechen und dort später ohne Verluste wieder weiterzumachen</i></p>	<table border="1" style="margin: auto;"> <caption>Data for 'bietet keine Möglichkeit...'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>0</td></tr> <tr><td>-/+</td><td>3</td></tr> <tr><td>+</td><td>0</td></tr> <tr><td>++</td><td>3</td></tr> <tr><td>+++</td><td>1</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	0	-/+	3	+	0	++	3	+++	1	<p><i>bietet die Möglichkeit, die Arbeit an jedem Punkt zu unterbrechen und dort später ohne Verluste wieder weiterzumachen</i></p>
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	0																	
-/+	3																	
+	0																	
++	3																	
+++	1																	
<p><i>erzwingt eine unnötig starre Einhaltung von Bearbeitungsschritten</i></p>	<table border="1" style="margin: auto;"> <caption>Data for 'erzwingt eine unnötig starre...'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>0</td></tr> <tr><td>-/+</td><td>1</td></tr> <tr><td>+</td><td>3</td></tr> <tr><td>++</td><td>2</td></tr> <tr><td>+++</td><td>1</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	0	-/+	1	+	3	++	2	+++	1	<p><i>erzwingt keine unnötig starre Einhaltung von Bearbeitungsschritten</i></p>
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	0																	
-/+	1																	
+	3																	
++	2																	
+++	1																	
<p><i>ermöglicht keinen leichten Wechsel zwischen einzelnen Menüs oder Masken</i></p>	<table border="1" style="margin: auto;"> <caption>Data for 'ermöglicht keinen leichten Wechsel...'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>0</td></tr> <tr><td>-/+</td><td>1</td></tr> <tr><td>+</td><td>0</td></tr> <tr><td>++</td><td>1</td></tr> <tr><td>+++</td><td>5</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	0	-/+	1	+	0	++	1	+++	5	<p><i>ermöglicht einen leichten Wechsel zwischen einzelnen Menüs oder Masken</i></p>
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	0																	
-/+	1																	
+	0																	
++	1																	
+++	5																	
<p><i>erzwingt unnötige Unterbrechungen der Arbeit</i></p>	<table border="1" style="margin: auto;"> <caption>Data for 'erzwingt unnötige Unterbrechungen...'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>0</td></tr> <tr><td>-/+</td><td>1</td></tr> <tr><td>+</td><td>0</td></tr> <tr><td>++</td><td>3</td></tr> <tr><td>+++</td><td>3</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	0	-/+	1	+	0	++	3	+++	3	<p><i>erzwingt keine unnötige Unterbrechung der Arbeit</i></p>
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	0																	
-/+	1																	
+	0																	
++	3																	
+++	3																	

B Auswertung der Fragebögen

Erwartungskonformität <i>Kommt die Software durch eine einheitliche und verständliche Gestaltung Ihren Erwartungen und Gewohnheiten entgegen?</i>																		
<i>erschwert die Orientierung, durch eine uneinheitliche Gestaltung</i>	<table border="1" style="margin: auto;"> <caption>Data for 'erschwert die Orientierung, durch eine uneinheitliche Gestaltung'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>0</td></tr> <tr><td>-/+</td><td>0</td></tr> <tr><td>+</td><td>0</td></tr> <tr><td>++</td><td>2</td></tr> <tr><td>+++</td><td>5</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	0	-/+	0	+	0	++	2	+++	5	<i>erleichtert die Orientierung durch eine einheitliche Gestaltung</i>
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	0																	
-/+	0																	
+	0																	
++	2																	
+++	5																	
<i>lässt einem im Unklaren darüber, ob eine Eingabe erfolgreich war oder nicht</i>	<table border="1" style="margin: auto;"> <caption>Data for 'lässt einem im Unklaren darüber, ob eine Eingabe erfolgreich war oder nicht'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>2</td></tr> <tr><td>-/+</td><td>0</td></tr> <tr><td>+</td><td>0</td></tr> <tr><td>++</td><td>4</td></tr> <tr><td>+++</td><td>1</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	2	-/+	0	+	0	++	4	+++	1	<i>lässt einem nicht im Unklaren darüber, ob eine Eingabe erfolgreich war oder nicht</i>
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	2																	
-/+	0																	
+	0																	
++	4																	
+++	1																	
<i>informiert in unzureichendem Maße über das, was sie gerade macht</i>	<table border="1" style="margin: auto;"> <caption>Data for 'informiert in unzureichendem Maße über das, was sie gerade macht'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>2</td></tr> <tr><td>-/+</td><td>1</td></tr> <tr><td>+</td><td>1</td></tr> <tr><td>++</td><td>1</td></tr> <tr><td>+++</td><td>2</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	2	-/+	1	+	1	++	1	+++	2	<i>informiert in ausreichendem Maße über das was sie gerade macht</i>
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	2																	
-/+	1																	
+	1																	
++	1																	
+++	2																	
<i>reagiert mit schwer vorhersehbaren Bearbeitungszeiten</i>	<table border="1" style="margin: auto;"> <caption>Data for 'reagiert mit schwer vorhersehbaren Bearbeitungszeiten'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>2</td></tr> <tr><td>-/+</td><td>1</td></tr> <tr><td>+</td><td>1</td></tr> <tr><td>++</td><td>3</td></tr> <tr><td>+++</td><td>1</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	2	-/+	1	+	1	++	3	+++	1	<i>reagiert mit gut vorhersehbaren Bearbeitungszeiten</i>
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	2																	
-/+	1																	
+	1																	
++	3																	
+++	1																	
<i>lässt sich nicht durchgehend nach einem einheitlichen Prinzip bedienen</i>	<table border="1" style="margin: auto;"> <caption>Data for 'lässt sich nicht durchgehend nach einem einheitlichen Prinzip bedienen'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>0</td></tr> <tr><td>-/+</td><td>0</td></tr> <tr><td>+</td><td>0</td></tr> <tr><td>++</td><td>1</td></tr> <tr><td>+++</td><td>6</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	0	-/+	0	+	0	++	1	+++	6	<i>lässt sich durchgehend nach einem einheitlichen Prinzip bedienen</i>
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	0																	
-/+	0																	
+	0																	
++	1																	
+++	6																	

B Auswertung der Fragebögen

Fehlertoleranz Bietet Ihnen die Software die Möglichkeit, trotz fehlerhafter Eingaben das beabsichtigte Arbeitsergebnis ohne oder mit geringem Korrekturaufwand zu erreichen?																		
ist so gestaltet, dass kleine Fehler schwerwiegende Folgen haben können	<table border="1" style="display: none;"> <caption>Data for 'ist so gestaltet, dass kleine Fehler schwerwiegende Folgen haben können'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>0</td></tr> <tr><td>-*</td><td>3</td></tr> <tr><td>+</td><td>1</td></tr> <tr><td>++</td><td>3</td></tr> <tr><td>+++</td><td>0</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	0	-*	3	+	1	++	3	+++	0	ist so gestaltet, dass kleine Fehler keine schwerwiegenden Folgen haben können
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	0																	
-*	3																	
+	1																	
++	3																	
+++	0																	
informiert zu spät über fehlerhafte Eingaben	<table border="1" style="display: none;"> <caption>Data for 'informiert zu spät über fehlerhafte Eingaben'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>0</td></tr> <tr><td>-*</td><td>2</td></tr> <tr><td>+</td><td>2</td></tr> <tr><td>++</td><td>1</td></tr> <tr><td>+++</td><td>2</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	0	-*	2	+	2	++	1	+++	2	informiert sofort über <u>fehlerhafte</u> Eingaben
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	0																	
-*	2																	
+	2																	
++	1																	
+++	2																	
liefert schlecht verständliche Fehlermeldungen	<table border="1" style="display: none;"> <caption>Data for 'liefert schlecht verständliche Fehlermeldungen'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>0</td></tr> <tr><td>-*</td><td>3</td></tr> <tr><td>+</td><td>0</td></tr> <tr><td>++</td><td>2</td></tr> <tr><td>+++</td><td>2</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	0	-*	3	+	0	++	2	+++	2	liefert gut verständliche Fehlermeldungen
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	0																	
-*	3																	
+	0																	
++	2																	
+++	2																	
erfordert bei Fehlern im großen und ganzen einen hohen Korrekturaufwand	<table border="1" style="display: none;"> <caption>Data for 'erfordert bei Fehlern im großen und ganzen einen hohen Korrekturaufwand'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>0</td></tr> <tr><td>-*</td><td>1</td></tr> <tr><td>+</td><td>0</td></tr> <tr><td>++</td><td>3</td></tr> <tr><td>+++</td><td>3</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	0	-*	1	+	0	++	3	+++	3	erfordert bei Fehlern im großen und ganzen einen geringen Korrekturaufwand
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	0																	
-*	1																	
+	0																	
++	3																	
+++	3																	
gibt keine konkreten Hinweise zur Fehlerbehebung	<table border="1" style="display: none;"> <caption>Data for 'gibt keine konkreten Hinweise zur Fehlerbehebung'</caption> <thead> <tr><th>Kategorie</th><th>Häufigkeit</th></tr> </thead> <tbody> <tr><td>---</td><td>0</td></tr> <tr><td>--</td><td>0</td></tr> <tr><td>-</td><td>0</td></tr> <tr><td>-*</td><td>4</td></tr> <tr><td>+</td><td>0</td></tr> <tr><td>++</td><td>1</td></tr> <tr><td>+++</td><td>2</td></tr> </tbody> </table>	Kategorie	Häufigkeit	---	0	--	0	-	0	-*	4	+	0	++	1	+++	2	gibt konkrete Hinweise zur Fehlerbehebung
Kategorie	Häufigkeit																	
---	0																	
--	0																	
-	0																	
-*	4																	
+	0																	
++	1																	
+++	2																	

B Auswertung der Fragebögen

Lernförderlichkeit <i>Ist die Software so gestaltet, dass Sie sich ohne großen Aufwand in sie einarbeiten konnten und bietet sie auch dann Unterstützung, wenn Sie neue Funktionen lernen möchten?</i>																		
<i>erfordert viel Zeit zum Erlernen</i>	<p>Detailed description: A bar chart with a y-axis from 0 to 5 and an x-axis with categories ---, --, -, -/+, +, ++, +++. The bars are at ++ (4) and +++ (3).</p> <table border="1"> <tr><th>Kategorie</th><td>---</td><td>--</td><td>-</td><td>-/+</td><td>+</td><td>++</td><td>+++</td></tr> <tr><th>Häufigkeit</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>4</td><td>3</td></tr> </table>	Kategorie	---	--	-	-/+	+	++	+++	Häufigkeit	0	0	0	0	0	4	3	<i>erfordert wenig Zeit zum Erlernen</i>
Kategorie	---	--	-	-/+	+	++	+++											
Häufigkeit	0	0	0	0	0	4	3											
<i>ermutigt nicht dazu, auch neue Funktionen auszuprobieren</i>	<p>Detailed description: A bar chart with a y-axis from 0 to 5 and an x-axis with categories ---, --, -, -/+, +, ++, +++. The bars are at -/+ (1), + (1), ++ (4), and +++ (1).</p> <table border="1"> <tr><th>Kategorie</th><td>---</td><td>--</td><td>-</td><td>-/+</td><td>+</td><td>++</td><td>+++</td></tr> <tr><th>Häufigkeit</th><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>4</td><td>1</td></tr> </table>	Kategorie	---	--	-	-/+	+	++	+++	Häufigkeit	0	0	0	1	1	4	1	<i>ermutigt dazu, auch neue Funktionen auszuprobieren</i>
Kategorie	---	--	-	-/+	+	++	+++											
Häufigkeit	0	0	0	1	1	4	1											
<i>erfordert, dass man sich viele Details merken muss</i>	<p>Detailed description: A bar chart with a y-axis from 0 to 5 and an x-axis with categories ---, --, -, -/+, +, ++, +++. The bars are at + (2), ++ (4), and +++ (1).</p> <table border="1"> <tr><th>Kategorie</th><td>---</td><td>--</td><td>-</td><td>-/+</td><td>+</td><td>++</td><td>+++</td></tr> <tr><th>Häufigkeit</th><td>0</td><td>0</td><td>0</td><td>0</td><td>2</td><td>4</td><td>1</td></tr> </table>	Kategorie	---	--	-	-/+	+	++	+++	Häufigkeit	0	0	0	0	2	4	1	<i>erfordert nicht, dass man sich viele Details merken muss</i>
Kategorie	---	--	-	-/+	+	++	+++											
Häufigkeit	0	0	0	0	2	4	1											
<i>ist so gestaltet, dass sich einmal Gelerntes schlecht einprägt</i>	<p>Detailed description: A bar chart with a y-axis from 0 to 5 and an x-axis with categories ---, --, -, -/+, +, ++, +++. The bars are at ++ (3) and +++ (4).</p> <table border="1"> <tr><th>Kategorie</th><td>---</td><td>--</td><td>-</td><td>-/+</td><td>+</td><td>++</td><td>+++</td></tr> <tr><th>Häufigkeit</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3</td><td>4</td></tr> </table>	Kategorie	---	--	-	-/+	+	++	+++	Häufigkeit	0	0	0	0	0	3	4	<i>erfordert wenig Zeit zum Erlernen</i>
Kategorie	---	--	-	-/+	+	++	+++											
Häufigkeit	0	0	0	0	0	3	4											
<i>ist schlecht ohne fremde Hilfe oder Handbuch erlernbar</i>	<p>Detailed description: A bar chart with a y-axis from 0 to 4 and an x-axis with categories ---, --, -, -/+, +, ++, +++. The bars are at -/+ (2), + (1), ++ (3), and +++ (1).</p> <table border="1"> <tr><th>Kategorie</th><td>---</td><td>--</td><td>-</td><td>-/+</td><td>+</td><td>++</td><td>+++</td></tr> <tr><th>Häufigkeit</th><td>0</td><td>0</td><td>0</td><td>2</td><td>1</td><td>3</td><td>1</td></tr> </table>	Kategorie	---	--	-	-/+	+	++	+++	Häufigkeit	0	0	0	2	1	3	1	<i>ermutigt dazu, auch neue Funktionen auszuprobieren</i>
Kategorie	---	--	-	-/+	+	++	+++											
Häufigkeit	0	0	0	2	1	3	1											

Glossar

Account

(engl.) in einem Vertriebssystem bezeichnet Account ein Unternehmen, das mit dem Lieferanten in irgendeiner Beziehung steht.

Account Type

(engl.) Einordnung eines Unternehmens aus der Sicht des Lieferanten (Anwender des Vertriebssystems). Es stehen in diesem Vertriebssystem folgende Einordnungsmöglichkeiten zur Verfügung:

- Competitor
- Customer
- Customer from SP
- End Cus Competitor
- Potential CUS
- Potential SP
- Potential Technology Partner
- Press
- SP Competitor
- Solution Provider
- Supplier
- Technology Partner
- not active

Activity

(engl.) Frei definierbare Art einer Aktivität innerhalb eines Vertriebssystems.

Applicationserver

(engl.) Im Allgemeinen ist ein Applicationserver ein Server in einem Computer-Netzwerk auf dem Anwendungen laufen. Im engeren Sinne bezeichnet ein Applicationserver eine Software, die als Ablaufumgebung für Enterprise-Anwendungen dient. Er stellt verschiedene Dienste, z.B. für die Authentifizierung zur Verfügung.

Buying Center

(engl.) Einkaufsstelle eines Unternehmens. Gruppe von Personen, die an Entscheidungen beteiligt ist. Den Personen eines Buying Centers werden Rollen zugewiesen. So werden folgende Rollen in einem Buying Center unterschieden:

- Buyer
- Decider
- Gatekeeper
- Influencer
- User

Cluster

(engl.) Ein Cluster gruppiert zusammengehörige oder ähnliche Daten.

Contact

(engl.) Bezeichnung für eine Kontaktperson, die einem Unternehmen (Account) innerhalb eines Vertriebssystems, zugeordnet ist.

Hibernate

Hibernate ist ein Open Source-Persistenz-Framework für Java. Damit wird es möglich den Zustand eines Objekts dauerhaft in einer relationalen Datenbank zu speichern, ohne die Zugriffe explizit per SQL programmieren zu müssen. Diese Funktionalität nennt man auch Objektrelationales Mapping. Da das Framework sich selbst um die Übertragen von und zur Datenbank kümmert, ist die Anwendung unabhängig vom SQL-Dialekt der verwendeten Datenbank.

Industry

(engl.) Industriezweig eines Unternehmens. Im vorliegenden Vertriebssystem stehen folgende Industriezweige zur Auswahl zur Verfügung:

- Advertising Agency
- Automotive

- Call Center
- Finance
- Government
- IT
- Industry
- Insurance
- Media
- Other
- Press
- Telecommunication
- Tourism
- Wholesale

J2EE

J2EE/JEE Java Enterprise Edition (früher J2EE) ist die Spezifikation einer Softwarearchitektur.

JAAS

(engl.) Java Authentication and Authorization Service ist eine Java-API, die es ermöglicht Dienste zur Authentifizierung zur Verfügung zu stellen.

JBoss

Implementierung eines Applicationsservers. Er dient hier auch als Laufzeitumgebung für die EJBs und kümmert sich mit seinem Hibernate Dienst um das Objektrelationale Mapping.

JPA

(engl.) Die Java Persistence API ist eine Schnittstelle für Java-Anwendungen, die die Zuordnung und die Übertragung von Objekten zu Datenbankeinträgen vereinfacht. Sie vereinfacht die Lösung des Problems, Laufzeit-Objekte einer Java-Anwendung über eine einzelne Sitzung hinaus zu speichern (Persistenz), wobei relationale Datenbanken eingesetzt werden können, die ursprünglich nicht für objektorientierte Datenstrukturen vorgesehen sind.

Model View Controller

(engl.) MVC bezeichnet ein Architekturmuster zur Strukturierung von Software-Entwicklung in die drei Einheiten Datenmodell (engl. Model), Präsentation (engl. View) und Programmsteuerung (engl. Controller). Ziel des Musters ist es, einen flexiblen Programmentwurf zu erstellen, der u. e. eine spätere Änderung oder Erweiterung erleichtert und eine Wiederverwendbarkeit der einzelnen Komponenten ermöglicht.

Opportunity

(engl.) Geschäftsgelegenheit bzw. potentieller Umsatz im Rahmen eines Vertriebssystems.

Relationship

(engl.) im Rahmen dieses Vertriebssystems gerichtete Beziehung zwischen zwei Accounts. In diesem Vertriebssystem stehen folgende Beziehungen zur Verfügung:

- ist Mutter von
- gehört zu
- arbeitet für
- beauftragt

Rich Client

(engl. auch Fat Client) Bezeichnung für einen Client, bei dem die eigentliche Verarbeitung der Daten vor Ort auf dem Client vollzogen wird, er stellt auch meistens die grafische Benutzeroberfläche zur Verfügung.

Struts

Struts ist ein Open Source Framework für die Präsentations- und Steuerungsschicht von Java-Webanwendungen. Es enthält viele applikationsrelevante Funktionen die bereits implementiert und einsatzbereit sind. Struts wird bereits in sehr vielen Webanwendungen eingesetzt und gemeinhin als solides Framework angesehen. Struts wurde von Craig McClanahan im Jahr 2000 entwickelt. Seitdem arbeitet eine ständig wachsende Entwicklergemeinschaft an der Verbesserung des Frameworks.

Abkürzungsverzeichnis

AJAX	Asynchronous JavaScript and XML
CRM	Customer Relationship Management
EJB	Enterprise Java Beans
JPQL	Java Persistence Query Language
JSF	JavaServer Faces
LDAP	Lightweight Directory Access Protocol
LOC	Lines of Code
POJO	Plain Old Java Object
WYSIWYG	What You See Is What You Get

Literaturverzeichnis

- [Aleaf 2008] Aleaf, Fahim: *Entwicklung eines Reportmoduls für Ad-hoc Abfragen als Komponente eines webbasierten Open Source Vertriebssystems*. Hamburg, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit in Vorbereitung, 2008
- [DIN 1987–07] DIN: *DIN 55350 - Begriffe der Qualitätssicherung und Statistik - Begriffe zu Bescheinigungen über die Ergebnisse von Qualitätsprüfungen; Qualitätsprüf-Zertifikate*. 1987-07
- [Geary und Horstmann 2004] Geary, David ; Horstmann, Cay: *core JavaServer Faces*. Prentice Hall, 2004. – ISBN 0-13-146305-5
- [IEEE 2000] IEEE: *IEEE Standard 1471*. 2000. – URL <http://www.ieee.org>. – Abrufdatum: 14.08.2008
- [Ihns u. a. 2007] Ihns, O. ; Harbeck, D. ; Heldt, S. ; Koschek, H.: *EJB 3 professionell*. dpunkt.verlag, 2007. – ISBN 978-3-89864-431-0
- [Ludewig und Lichter 2007] Ludewig, Jochen ; Lichter, Horst: *Software Engineering - Grundlagen, Menschen, Prozesse, Techniken*. dpunkt.verlag, 2007. – ISBN 3-89864-268-2
- [McCabe 1976] McCabe, Thomas J.: *A Complexity Measure*. 1976. – URL <http://www.literateprogramming.com/mccabe.pdf>. – Abrufdatum: 24.08.2008
- [Pomberger und Weinreich 1994] Pomberger, G. ; Weinreich, R.: *The Role of Prototyping in Software Development, Tutorial Paper, Conference on the Technology of Object-Oriented Languages and Systems (TOOLS Europe '94), Versailles, France*. Prentice Hall, 1994
- [Pomberger und Pree 2004] Pomberger, Gustav ; Pree, Wolfgang: *Software Engineering, Architektur-Design und Prozessorientierung*. Hanser, 2004. – ISBN 3-446-22429-7
- [Red Hat Inc.] Red Hat Inc.: *JBoss Application Server*. – URL <http://www.jboss.org/>. – Abrufdatum: 04.08.2008

- [Sadowski 2008] Sadowski, Martin: *Konzeption und prototypische Entwicklung eines Open-Source Vertriebssystems*. Hamburg, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2008
- [Schweibenz und Thissen 2003] Schweibenz, Werner ; Thissen, Frank: *Qualität im Web - Benutzerfreundliche Webseiten durch Usability Evaluation*. Springer-Verlag Berlin Heidelberg, 2003. – ISBN 3-540-41371-5
- [Sotograph 2008] Sotograph: *Software Tomography*. 2008. – URL <http://www.software-tomography.de/html/sotograph.html>. – Abrufdatum: 04.08.2008
- [Sun Microsystems Inc. 2008a] Sun Microsystems Inc.: *Core J2EE Patterns - Service Locator*. 2008. – URL <http://java.sun.com/blueprints/corej2eepatterns/Patterns/ServiceLocator.html>. – Abrufdatum: 20.08.2008
- [Sun Microsystems Inc. 2008b] Sun Microsystems Inc.: *Java 2 Platform, Enterprise Edition (J2EE) Overview*. 2008. – URL <http://java.sun.com/j2ee/overview.html>. – Abrufdatum: 04.08.2008
- [Tanenbaum und van Steen 2008] Tanenbaum, Andrew S. ; Steen, Maarten van: *Verteilte Systeme - Prinzipien und Paradigmen*. Pearson Studium, 2008. – ISBN 978-3-8273-7293-2
- [Winkelmann 2003] Winkelmann, Peter: *Vertriebskonzeption und Vertriebssteuerung*. Vahlen, 2003. – ISBN 3-8006-3234-9
- [Züllighoven] Züllighoven, Heinz: *Anwendungsorientierte Dokumenttypen, Arbeitskreis Softwaretechnik, Universität Hamburg*. – URL <http://www.informatik.uni-hamburg.de/SWT/attachments/LVTermine/00SE3.pdf>. – Abrufdatum: 26.07.2008

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 26. August 2008

Ort, Datum

Unterschrift