

Kooperative Abwesenheitsplanung in einem ERP-System

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Wirtschaftsingenieurwesen Informatik

eingereicht von

Petra Eibl

Matrikelnummer 0126096

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuerin: Assoc. Prof. Dipl.-Ing. Dr. Hilda Tellioglu

Wien, 21.11.2011

(Unterschrift Verfasserin)

(Unterschrift Betreuerin)

Erklärung zur Verfassung der Arbeit

Petra Eibl
Teichbach 7
3620 Spitz

‘Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.’

Wien, 21.11.2011

(Unterschrift Verfasserin)

Kurzfassung

Personal und deren Verfügbarkeit ist ein wichtiger Faktor jedes Unternehmens. Diese Arbeit hat sich zum Ziel gesetzt, sich mit den Grundlagen der Abwesenheitsplanung und deren Umsetzung zu befassen. Am Beispiel eines Unternehmens wurde deren aktuelle Situation untersucht. Durch Interviews wurden die Nachteile des aktuellen Zustands mit SAP und Wünsche für die Zukunft eruiert. Mit den daraus gewonnenen Kenntnissen wurde ein Programm zur Abwesenheitsplanung mit Verbindung zu einem ERP-System entwickelt.

Im ersten Teil wird das theoretische Rahmengeniet näher behandelt. Da der Informationsaustausch über Abwesenheiten durch eine Applikation erfolgen soll und die Personen dabei zusammenarbeiten, wurde auf den Bereich CSCW und die damit verbundene Groupware näher eingegangen. Im Bereich Arbeitsrecht wird erklärt, aus welchen Gründen Mitarbeiterinnen und Mitarbeiter abwesend sein dürfen und welche Zeiträume diese Abwesenheit umfassen darf. Inwieweit Personen von den Abwesenheiten untereinander wissen dürfen, wurde im Datenschutzgesetz erläutert. Die Benutzung der Applikation soll für die Anwenderinnen und Anwender so einfach wie möglich sein. Daher wurde auf die Themen Visualisierung und Usability näher eingegangen. Bei Abwesenheitsapplikation handelt es sich um ein verteiltes System, bei dem den Benutzerinnen und Benutzern verborgen bleiben soll, dass die Daten sich auf einem anderen Rechner befinden.

Der zweite Teil der Arbeit beschäftigt sich mit der praktischen Realisierung der Applikation. Des Weiteren wurden nach schon existierenden Programmen für die Abwesenheitsplanung recherchiert. Neben der aktuellen Situation des Beispielunternehmens wurde auf die Entwicklung näher eingegangen. Nach der Realisierung wurde die praktische Implementierung getestet und Verbesserungen erläutert.

Ergebnis war eine Applikation, durch welche es Mitarbeiterinnen und Mitarbeiter einfach ermöglicht wird. Abwesenheiten zu planen und aufeinander abzustimmen. Bei dieser Arbeit hat sich herausgestellt, dass die Benutzerinnen und Benutzer vor allem auf die grundlegenden Funktionen einer Abwesenheitsapplikation Wert legen. Diese Funktionen sollten so einfach wie möglich gestaltet werden und dadurch eine Anwendung, auch wenn sie nur sporadisch ist, leicht durchzuführen sein. Durch die Einbindung der Anwenderinnen und Anwender in den Entwicklungsprozess wurden Fehler schon früh beseitigt und unnötige Aufwendungen von vornherein vermieden. Das SAP-System konnte als Datenbank dienen. Dadurch wurde es möglich, die Systemanforderungen gering zu halten und dennoch stets aktuelle Daten bereitzustellen.

Abstract

An important factor of every business company is the staff and the availability. The goal of this thesis is to examine possibilities for absence planning and its implementation. For example the current situation of a business company was investigated. By interviews, the disadvantages of the current status and needs were exposed. Based on this knowledge, a program was developed for absence planning with connection to an ERP system. As ERP system SAP is in use.

In the first part the theoretical background is layed out. The information sharing about absences in this application is used by more than one person, so CSCW (computer supported cooperative work) and groupware are discussed. The area of labor law clarifies the different reasons for absences and which periods this absence may enclose. How much the users may know about absences of their colleagues is explained in the data protection act. The use of the application should be as simple as possible for the users, so visualization and usability were discussed. The absence application is a distributed system, where the distribution of the data on more than one computer should be hidden for the user.

The second part of the work deals with the practical realization of the application. Furthermore, existing programs for the absence planning were investigated. The implementation will also be presented. In the end the prototype is tested by users and improvements are explained.

The result was a simple application for absence planning. This work shows that for the users the basic functions of an absence planning application is important. These functions should be formed as simple as possible, so that users who use the application only sporadically can perform these functions easily. Mistakes were removed by the integration of the users in the development process and therefore unnecessary effort was avoided from the start. The SAP system could be used as database. This made it possible to minimize the system requirements and to always have the current data without redundant storage of the data.

Inhaltsverzeichnis

Inhaltsverzeichnis	vi
1 Einleitung	1
1.1 Aktuelle Situation	1
1.2 Entwicklung	2
1.3 Ergebnis	3
2 Theoretischer Rahmen für kooperative Abwesenheitsplanung	5
2.1 Computer Supported Cooperative Work (CSCW)	5
2.1.1 Groupware	6
2.1.2 Awareness	8
2.2 User-Centered Design	10
2.3 User Interface	18
2.4 Layout	19
2.4.1 Mapping	20
2.4.2 Visualisierung	21
2.4.3 Usability	22
2.5 Abwesenheitsplanung	25
2.6 Arbeitsrecht	26
2.7 Datenschutz	29
2.8 SAP	30
2.8.1 SAP ERP Human Capital Management	31
2.8.2 SAP und der Datenschutz	32
2.9 Verteilte Systeme	33
2.9.1 Entfernter Prozeduraufruf	34
2.9.2 Verteiltes System mit SAP	36
3 Praktische Realisierung	41
3.1 Aktuelle Situation	41
3.2 Existierende Programme	48
3.2.1 GeCOSOFIT	48
3.2.2 HR-Starter®ASP	49
3.2.3 Softsimple	49

3.3	Use Cases und Aktivitätsdiagramme	51
3.3.1	Use Cases	51
3.3.2	Aktivitätsdiagramme	54
3.3.3	Unified Modeling Language	56
3.4	Layout der Abwesenheitsplanung	59
3.4.1	Layout der Mitarbeiterinnen und Mitarbeiter	59
3.4.2	Layout der Vorgesetzten	62
3.5	Implementierung	64
3.6	Programm	72
3.7	Usability-Test	80
4	Conclusio	87
4.1	Ausblick	89
A	SAP-Web Services in Silverlight/C#	91
	Abbildungsverzeichnis	94
	Literaturverzeichnis	97

Kapitel 1

Einleitung

Die Verfügbarkeit von Personal ist ein wichtiges Instrument eines Unternehmens. Insbesondere bei größeren Unternehmen ist diese Aufgabenstellung nicht mehr mit einem einfachen Kalender, in dem die Abwesenheitszeiten, wie zum Beispiel Urlaub, Schulungen usw. eingetragen werden, zu bewältigen.

1.1 Aktuelle Situation

Der Bereich der Personalwirtschaft setzt sich aus mehreren Bereichen zusammen, zum Beispiel Personalbeschaffung, Personalplanung, Personalentwicklung, Personalstammdatenverwaltung und natürlich Personalabrechnung. Die meisten dieser Bereiche werden von Expertinnen und Experten gewartet. Seit den 1990er Jahren haben Firmen auf der ganzen Welt Enterprise Resource Planning (ERP)-Systeme als einheitliche Informationssysteme in ihren jeweiligen Organisationen eingeführt [Rajagopal, 2002, S. 87]. Ein sehr gefragtes ERP-System ist SAP. 1999 haben sich fünf Hauptanbieter von ERP-Software-Systemen, mit SAP AG als Führer, mit mehr als 30% Marktanteil herausgebildet [Brown, 1999, S. 411].

Dass jede Mitarbeiterin und jeder Mitarbeiter seine Abwesenheitszeiten selbst einreichen kann, ist in der Standardversion von SAP nicht vorgesehen. Dazu wird zusätzlich unter anderem ein Enterprise Portal benötigt, in dem der Bereich Abwesenheitsplanung eingebettet ist. Das Portal kann als visuelle Schnittstelle zwischen einer Benutzerin oder einem Benutzer und allen grundlegenden Anwendungen und Informationen eines Unternehmens gesehen werden [Gomis, 2007]. SAP bietet dabei eine Vielzahl von Applikationen an, die durch das Enterprise Portal für den Anwender leichter zu bedienen sind. Das SAP Portal mit seinen Design-Tools ist ein sehr umfassendes und mächtiges Tool, allerdings auch an einigen Stellen etwas unübersichtlich [Weißenberg und Stemmer, 2009, S. 11]. Falls nur die Abwesenheitsplanung benötigt wird, wären zusätzliche Funktionen für die Benutzerinnen und Benutzer mehr verwirrend als hilfreich. Renate Kristiansen hat verschiedene Userinnen und User in 5 Firmen in Norwegen interviewt, die mit verschiedenen Versionen von SAP Client und SAP Enterprise Portal arbeiten [Kristiansen, 2005, S. 5]. Dabei kam heraus, dass die komplexe Bildschirmkonfiguration den Benutzerinnen und Benutzern keine Unterstützung für den Prozess zum Lösen einer bestimmten

Aufgabe bietet. Einige der interviewten Probandinnen und Probanden sagten, dass sie nur etwa 20 Prozent von dem, was auf dem Bildschirm erscheint, auch wirklich verstehen. Zudem bietet das Enterprise Portal auch in technischer Hinsicht Hindernisse, so ist es laut Till Jeske nach SAPs eigenen Schulungsunterlagen nur mit einer ganz speziellen Version eines bestimmten Betriebssystems nutzbar [Jeske, 2005, S. 259].

Zwar gibt es auf dem Markt einige Systeme, mit denen Benutzerinnen und Benutzer Abwesenheiten planen und die Daten mit einem SAP-System abgleichen können. Allerdings passiert das nur in regelmäßigen Abständen, daher sind die Daten nicht immer aktuell. Zudem verfügen sie über zusätzliche Anforderungen, wie zum Beispiel eine eigene Datenbank.

In dieser Arbeit wird untersucht, wie die Abwesenheitsplanung konkret in einem Unternehmen durchgeführt wird, wie zufriedenstellend die Abläufe sind und wie diese verbessert werden können. Zu guter Letzt soll ein Prototyp erstellt werden, der die Erkenntnisse vereint.

1.2 Entwicklung

Das in der Arbeit untersuchte Unternehmen ist mit der aktuellen Situation unzufrieden. Als betriebswirtschaftliche Software wird SAP verwendet. Für die Abwesenheitsplanung wird allerdings auf ein externes Programm zugegriffen, das keinerlei Verbindung zur SAP-Applikation zur Verfügung stellt. Daher werden die von den Mitarbeiterinnen und Mitarbeitern in der Abwesenheitsapplikation gesammelten Daten „händisch“ in das SAP-System übertragen. Dadurch kommt es immer wieder zu Fehlern, deren Resultat die Ungleichheit der Daten beider Systeme (Abwesenheitsapplikation und SAP-System) ist. Bei den Mitarbeiterinnen und Mitarbeitern führt diese Situation zu Verwirrungen, da sie sich über die angegebene Anzahl der Resturlaubstage und Überstundenanzahl, die noch abgebaut werden muss, nicht sicher sein können. Zudem müssen Ungleichheiten in mühevoller Kleinarbeit korrigiert werden.

Abgesehen von der fehlenden SAP-Verbindung hatten die Anwenderinnen und Anwender einiges am Layout zu bekritteln, wobei vieles davon dem individuellen Geschmack entspricht und es schwierig ist, all diese entsprechend zu berücksichtigen. Dem Administrator waren die Einstellmöglichkeiten zu umfangreich. Er wollte sich von vornherein mit der Software nicht intensiv beschäftigen und fand sich unter den vielen Varianten und Möglichkeiten nur schlecht zurecht. Wollte er nur einfache Dinge ändern, war ihm die Suche, wo er dies könne, einfach zu langwierig. Aus den Interviews ist hervorgegangen, dass sich die Benutzerinnen und Benutzer ein minimalistisches Programm wünschen, das die Grundfunktionen beherrscht, einfach in der Anwendung ist und die SAP-Übertragung übernimmt.

Für die Entwicklung der Applikation wurde User-Centered Design verwendet, bei dem die späteren Anwenderinnen und Anwender während des Entwicklungsprozesses integriert wurden. Durch das regelmäßige Feedback wurde verhindert, dass die fertige Applikation nicht deren Bedürfnissen entspricht. Somit kann schon rechtzeitig auf deren Wünsche eingegangen und die Entwicklung in die richtige Richtung gelenkt werden.

Den Anwenderinnen und Anwendern soll ermöglicht werden, ihre Abwesenheiten aufeinander abzustimmen und gemeinsam einen Abwesenheitsplan zu erstellen, beziehungsweise bei Bedarf zu ändern und anzupassen. Da alle diese Vorgänge nicht durch ein Face-To-Face-Meeting,

sondern mittels eines Computers ausgeführt werden, handelt es sich dabei um Computer Supported Cooperative Work (CSCW). Eine Gruppe von Personen, das heißt mehr als zwei Mitarbeiterinnen und Mitarbeiter, greifen daher auf gemeinsame Daten der Abwesenheiten zu. Daher muss die Applikation die Anforderungen eines Groupware-Systems besitzen. Es muss den Benutzerinnen und Benutzern möglich sein, Informationen auszutauschen. Der Ablauf der Beantragung, Verschiebung, Genehmigung oder Ablehnung muss koordiniert werden. Es dürfen zum Beispiel nicht zwei Anwenderinnen und Anwender gleichzeitig zur selben Zeit Abwesenheit beantragen, wenn sie einander vertreten. Die einzelnen Benutzerinnen und Benutzer sollen sich dabei im Klaren sein, dass sie mit anderen Personen ihre Abwesenheiten abstimmen müssen. Ihre Abwesenheiten haben Auswirkungen auf ihre Kolleginnen und Kollegen und umgekehrt (Awareness).

Im Rahmen dieser Arbeit mussten auch rechtliche Dinge geklärt werden. Zu welchen Abwesenheiten es in einem Unternehmen kommen kann, wird im Arbeitsrecht näher behandelt. Zudem wurde versucht zu eruieren, welche Daten die Mitarbeiterinnen und Mitarbeiter voneinander erfahren dürfen. Daher wurde auf den Bereich des Datenschutzes näher eingegangen.

Vor der eigentlichen Programmierung wurde auf Papier das Layout entworfen, da das ein gutes Werkzeug darstellt [Cooper et al., 2007, S. 133]. Es fördert die Diskussion und ist leicht zu ändern. Dieses Layout wurde mit den Anwenderinnen und Anwendern besprochen, wobei es zu leichten Änderungen kam.

Als Programmiersprache wurden Silverlight und C# verwendet. Als Grundlage für die Abwesenheitsapplikation wurde der Silverlight Scheduler von CodePlex (Open Source Project Community) verwendet, dessen Code frei verfügbar ist [Codeplex, 2010]. Da er für die Planung von Events gedacht ist, mussten einige Änderungen vorgenommen werden. Zu den schwierigsten Herausforderungen zählten die Verbindungen der Applikation mit SAP, wobei das vor allem auf die Unerfahrenheit mit dem System zurückzuführen ist. Bei dieser Konstellation handelt es sich um ein verteiltes System, da das SAP-System auf einen Server läuft, während die Abwesenheitsapplikation auf den Rechnern der einzelnen Mitarbeiterinnen und Mitarbeiter. Die in Silverlight eingebauten SAP-Web Services waren nicht funktionstüchtig, da die domänenübergreifenden Richtlinien fehlten. Zwar stellt Silverlight einen Lösungsweg für dieses Problem bereit, der leider in SAP nicht umsetzbar ist. Erst lange Internetrecherche, die zu Tage brachte, dass die Kombination von Silverlight und SAP nicht alltäglich ist, brachte die Lösung. Es wurde daraufhin ein eigenes Programm in SAP erstellt.

1.3 Ergebnis

Das Resultat war ein Prototyp für die Planung von Abwesenheiten. Mit Hilfe dieses Programms können Mitarbeiterinnen und Mitarbeiter Abwesenheiten beantragen, verschieben und auf Wunsch auch wieder löschen. Vorgesetzte haben zusätzlich die Möglichkeit, Beantragungen zu genehmigen und zu löschen. Die Applikation verfügt über keine eigene Datenbank, alle Daten werden direkt von SAP-Datenbanktabellen gelesen und auch dort wieder hineingeschrieben.

Während des gesamten Entwicklungsprozesses von den Anforderungen der Applikation, über das Layout, bis hin zum fertigen Prototyp waren die Anwenderinnen und Anwender involviert. Durch das regelmäßige Feedback kam es schon früh zu Korrekturen und Fehlentwicklungen.

gen wurden verhindert.

Bei der Entwicklung wurde versucht, ein möglichst einfaches Layout zu gestalten. Vorgänge, die sich ähnlich sind, wie zum Beispiel eine Abwesenheit zu beantragen oder zu editieren, wurden, so weit es geht, gleich gestaltet. Es wurde versucht zu vermeiden, dass Benutzerinnen und Benutzer Fehler machen können. Falls dies jedoch geschah, wurden eindeutige Fehlermeldungen vergeben.

Nach der Fertigstellung des Prototyps wurde er mit einer Gruppe von Personen getestet. Das Ergebnis war, dass die Testerinnen und Tester im Großen und Ganzen zufrieden waren. Einige Programmfehler müssen allerdings noch korrigiert werden. Zudem sollte das fertige Programm über ein paar zusätzliche Funktionen verfügen, beziehungsweise müssen bestehende noch geändert werden.

Kapitel 2

Theoretischer Rahmen für kooperative Abwesenheitsplanung

Wenn in einem Unternehmen das Personal zusammenarbeiten soll, muss dessen An- und Abwesenheit geplant werden. Die Planung der Abwesenheiten, zu denen unter anderem Urlaub und Zeitausgleich gehören, kann unterschiedlich erfolgen. Man kann zum Beispiel Besprechungen abhalten. Soll eine größere Anzahl von Mitarbeiterinnen und Mitarbeiter ihre Abwesenheiten aufeinander abstimmen, ist die computerunterstützte Planung sinnvoll, da nicht alle Personen zur selben Zeit anwesend sein müssen. Wie die letzte Variante umgesetzt wird und welche Dinge beachtet werden müssen, soll in diesem Kapitel geklärt werden.

2.1 Computer Supported Cooperative Work (CSCW)

Bei der Abwesenheitsplanung sollen mehrere Personen ihre Abwesenheit planen und aufeinander abstimmen. Alle diese Vorgänge sollen mit Hilfe eines Computers abgewickelt werden. Daher muss auf das Forschungsgebiet CSCW näher eingegangen werden.

CSCW ist ein Forschungsgebiet für rechnerische Unterstützung einer Gruppe oder eines Teams in Erfüllung eines gemeinsamen Ziels [Liang et al., 1996, S. 156]. Das Forschungsgebiet umfasst dabei mehrere Disziplinen [Schwabe et al., 2001, S. 2]. Es besteht aus Informatik, Wirtschaftsinformatik, Psychologie, Betriebswirtschaftslehre, Arbeitswissenschaften, Soziologie, Organisations- und Wirtschaftswissenschaften. Diese verschiedenen Disziplinen geben unterschiedliche Sichten auf CSCW. Zum Beispiel sehen Informatikerinnen und Informatiker eher die technischen Aspekte, wie das Netzwerk. Während für Psychologinnen und Psychologen zum Beispiel das Gruppenverhalten der Benutzerinnen und Benutzer im Vordergrund steht. Durch die verschiedenen Disziplinen kommt es zu Konflikten zwischen den einzelnen Definitionen, da Personen aus verschiedenen Forschungsgebieten andere Schwerpunkte setzen [Lubich, 1995, S. 28].

Die grundlegende Aufgabe der CSCW-Technologie besteht darin, dass Computer Tätigkeitsmuster von zusammenarbeitenden Personen imitieren, um zusammenhängende Computeraufgabe zu erledigen [Lin et al., 2006]. Neben diesen Computeraufgaben soll dabei noch koopera-

tive Arbeit unterstützen werden [Shangguang und Meilin, 1996, S. 187]. CSCW ist dabei ein Oberbegriff für die Art und Weise wie Personen in Gruppen mit Technologien von Computer-Netzwerken arbeiten und wie diese mit der dazugehörigen Hardware, Software und Technik kombiniert ist [Wilson, 1991, S. 1].

Wenn mehrere Personen in Gruppen arbeiten, kann es bei der traditionellen Face-To-Face-Gruppeninteraktion zu Problemen kommen [Kamel und Davison, 1998, S. 210]. Dabei gibt es Kategorien von Problemen, die zu Ineffizienzen führen:

1. **Zeit-ähnliche Probleme:** Oft ist es schwierig, einen Termin für eine Besprechung zu finden, zu dem alle Beteiligten ausreichend Zeit haben. Zudem benötigen Besprechungen viel Zeit, die eventuell besser genutzt werden könnte. Bei einer Befragung berichteten 18%, dass sie mehr als 20% ihrer Arbeitszeit in Meetings verbringen [Kamel und Davison, 1998, S. 210]. Nur 28% der Befragten meinten, dass mehr als 60% der Zeit in eine ernsthafte Diskussion verwendet worden sei und nur 33% waren der Ansicht, dass die Zeit effizient genutzt wurde.
2. **Distanz:** Kolleginnen und Kollegen arbeiten nicht immer nebeneinander. Besprechungen, bei denen sie sich gegenüber sitzen, können oft wegen der räumlichen Distanz schwierig zu realisieren sein. Müssen lange Anreisewege angetreten werden, kostet das Zeit, die stattdessen in sinnvollere Tätigkeiten investiert werden könnte.
3. **Kostenprobleme:** Müssen Personen zu den Besprechungen extra anreisen, ist das mit Zeit und dadurch auch mit Geld verbunden. Aber nicht nur die Anreisezeit kostet Geld, da die Mitarbeiterin oder der Mitarbeiter „untätig“ ist, sondern auch die Reise selbst ist mit Kosten verbunden, da die Person mit Auto, Bahn oder Flug anreist und eventuell noch eine Unterkunft benötigt.
4. **Verhaltensweise:** Vom Verhalten der Besprechungsteilnehmerinnen und Teilnehmer hängt das Ergebnis ab und kann es kann dazu kommen, dass kein Ergebnis zustande kommt. So kann zum Beispiel jede Teilnehmerin oder jeder Teilnehmer eine feste Meinung haben, von der sie oder er nicht bereit ist abzuweichen. Dadurch kommt es zu einem Problem, wenn eine einstimmige Entscheidung getroffen werden soll.
5. **Dominieren der Diskussion und Nachgeben gegen höheren Status:** Verschiedene Personen haben verschiedene Persönlichkeiten. Es gibt dominante Personen, die starken Einfluss auf die Besprechung und deren Ergebnisse haben. Andere Personen wiederum lassen sich leicht beeinflussen.

Während CSCW das Forschungsgebiet darstellt, definiert Groupware die Technologie dahinter [Grudin, 2011].

2.1.1 Groupware

Groupware ist ein Hilfsmittel zur Unterstützung der Kooperation innerhalb von Gruppen [Burger, 1997, S. 7].

Mit einer Gruppe ist, eine Gruppe von Personen gemeint, die aus mindestens zwei Personen besteht, die ein gemeinsames Ziel verfolgen [Burger, 1997, S. 28]. Die Größe einer Gruppe ist nach oben hin unbeschränkt, irgendwann spricht man allerdings nicht mehr von einer Gruppe. Wie dieses gemeinsame Ziel erreicht wird, ist für die einzelnen Gruppenmitglieder unerheblich, auch ob die einzelnen Personen diese Ziele selbst entwickelt haben oder sie von einer anderen Person (Vorgesetzte oder Vorgesetzter) vorgeben bekommen haben, ist gleichgültig. Es kann zwischen zwei unterschiedliche Arten von Gruppen unterschieden werden. Sie können statisch oder dynamisch sein [Crowcroft und Markus, 2001, S. 178]. Statische Gruppen bestehen aus den gleichen Personen, während die Mitglieder dynamischer Gruppen wechseln können. Gruppen können aber nicht nur in dynamische und statische eingeteilt werden, sondern auch, ob sie heterogen oder homogen sind [Balmer et al., 2000, S. 23]. Heterogene Gruppen sind im Gegensatz zu homogenen aufgrund unterschiedlicher Eigenschaften, wie zum Beispiel Ausbildung oder Alter durchmischt.

Innerhalb von Gruppen finden Kontakte mit wechselseitiger Beeinflussung statt [Hartung, 2006, S. 102]. Die Kommunikation dabei kann synchron oder asynchron stattfinden [Shirani et al., 1999, S. 140]. Bei einer synchronen Kommunikation treffen sich die Gruppenmitglieder zur selben Zeit am selben Ort. Um Abwesenheit zu beantragen, ist man sowohl zeit-, als auch ortsunabhängig von der oder dem Vorgesetzten, die oder der diese Abwesenheit genehmigt. Daher spricht man hier von einer asynchroner Interaktion.

Um Abwesenheiten aufeinander abzustimmen, müssen Informationen daraus auch für die jeweils anderen Mitarbeiterinnen und Mitarbeiter einsehbar sein. Mittels Groupware haben Mitarbeiterinnen und Mitarbeiter die Möglichkeit, auf gemeinsam benötigte Informationen zugreifen. Das verbreitetste Merkmal von Groupware, das in der Literatur gefunden wurde, ist die zeitliche Interaktion zwischen Gruppenmitgliedern [Fouss und Chang, 2000, S. 118]. Damit ist gemeint, wie die Software die Kommunikationen zwischen den Mitgliedern abwickelt oder/und wie die Mitglieder am Projekt arbeiten.

Groupware kann auch als computergestütztes System bezeichnet werden, das eine Gruppe von Personen in einer allgemeinen Aufgabe unterstützt und eine Schnittstelle zu einer geteilten Umgebung zur Verfügung stellt [Ellis et al., 1991, S. 40].

Zudem kann man Groupware unterscheiden nach der Anzahl der Benutzerinnen und Benutzer, die gleichzeitig damit arbeiten können. Bei der asynchronen Groupware kann nur ein Mitglied an dem Projekt arbeiten, während bei der synchronen mehrere Benutzerinnen und Benutzer gleichzeitig aktiv sein können [Klinke, 2008, S. 18]. Auch können die Mitglieder im selben Raum oder an verschiedenen Standorten sein, ohne direkten Sichtkontakt. Die meisten Groupwaresysteme unterstützen allerdings nur eine von diesen Arten [Ellis et al., 1991, S. 46]. Beispiele für Groupware sind Gruppendeditoren, Konferenzsysteme oder Nachrichtensysteme.

Um die Eigenschaften von Groupware besser zu verstehen, ist die Gruppeninteraktion an sich näher zu behandeln. Folgende Formen der Interaktion von Individuen in Gruppen werden häufig genannt [Gross et al., 2007, S. 8]:

- Mit *Koexistenz* ist das gleichzeitige Vorhandensein und Bereitstellen von gegenseitigen Informationen von Benutzerinnen und Benutzern gemeint, um diese nicht voneinander zu isolieren. Koexistenz bildet somit die Grundlage für gemeinsames Arbeiten [Far, 2010, S.

63]. Für die Abwesenheitsplanung bedeutet das, dass die Abwesenheiten nicht aufeinander abgestimmt werden können, wenn die einzelnen Personen nicht auf die Abwesenheiten der Kolleginnen und Kollegen Zugriff haben.

- Mittels *Kommunikation* ist es den Benutzerinnen und Benutzern möglich, Informationen auszutauschen. So müssen Vorgesetzte in Kenntnis gesetzt werden, dass einer ihrer Mitarbeiterinnen und Mitarbeiter Abwesenheiten beantragt haben. Diese müssen wiederum erfahren, ob ihr Antrag stattgegeben oder abgelehnt worden ist.
- *Koordination* ist notwendig, um Abwesenheiten aufeinander abzustimmen. Die Abläufe der Abwesenheitsapplikation müssen koordiniert werden. Es dürfen nicht zwei Benutzerinnen und Benutzer zur selben Zeit Abwesenheit beantragen, wenn sie einander vertreten.
- *Kooperation* lässt sich von Kommunikation nicht exakt trennen, denn durch die Kommunikation entsteht zumindest teilweise gemeinsames Material [Schiestl und Schelle, 2010, S. 14]. Durch die Arbeit mit diesem Material findet Kooperation statt.
- Die Gruppe muss über eine bestimmte Thematik übereinstimmen. Bei der Abwesenheitsplanung müssen sie sich über bestimmte Abwesenheitstermine absprechen.

Bei Groupware-Systemen ist es schwierig, für Personen einander wahrzunehmen [Gutwin, 1997, S. 21]. Awareness ist allerdings ein wichtiger Bestandteil jeder Zusammenarbeit und bei alltäglichen Face-To-Face-Kommunikation eine Selbstverständlichkeit. Groupwaresysteme beinhalten im Gegensatz zu herkömmlicher Kommunikation nur einen Teil der Informationen von anderen Personen. Verhaltensweisen und Interaktionen, die bei einer face-to-face Kommunikation mühelos waren, können mit einem Groupwaresystem gespreizt und formal sein.

2.1.2 Awareness

Das Konzept von Awareness spielt eine zentrale Rolle in CSCW und von Anfang an haben CSCW-Wissenschaftlerinnen und Wissenschaftler erforscht, wie Computer-basierten Technologien eine Art "Bewusstsein" zwischen den kooperierenden Akteuren erleichtern können [Schmidt, 2002, S. 285]. Awareness ist die Wahrnehmung von Aktivitäten von anderen Personen, welche wiederum Auswirkungen auf die eigenen Aktivitäten haben [Dourish und Bellotti, 1992, S. 107]. Awareness bedeutet einfach zu wissen, wer anwesend ist, welche Aktivitäten praktiziert werden und wer mit wem spricht [Yong, 2009, S. 48]. Bei einer Zusammenarbeit, bei der sich Personen treffen, beinhaltet Awareness, dass Mitarbeiterinnen und Mitarbeiter sich ihrer Kolleginnen und Kollegen bewusst werden [Yong, 2009, S. 48]. Sie nehmen die Tätigkeiten, Aufgabenverteilung, Diskussionen und gemeinsame Instrumente wahr.

Durch ein Groupwaresystem wird das visuelle Feld von einer Person auf einem Computerbildschirm minimiert [Gutwin et al., 1996b, S. 258]. Charakteristische Bewegungen und Geräusche, die mit diesen Dingen in Verbindung gebracht werden, können von der Person nicht wahrgenommen werden. Alle diese Veränderungen bewirken, dass Personen nur mehr einen Teil der Informationen im Gegensatz zu einem Face-to-face-Meeting empfangen und Awareness schwierig wird.

Es gibt verschiedene Arten von Gruppenarawareness [Jacko, 2009, S. 96]. Die Beziehungen werden anhand der Abbildung 2.1 verdeutlicht.

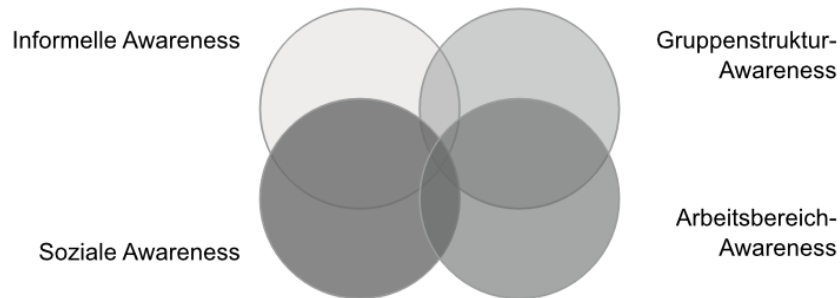


Abbildung 2.1: Verschiedene Arten von Awareness [Böhringer, 2009, S. 27].

- Informelles Bewusstsein (informal awareness) befasst sich mit dem grundlegenden Wissen über den Aufenthaltsort von anderen Personen: Wer physisch im selben Raum ist, und wo die Menschen sich im Bezug auf die anderen Personen befinden [Dasgupta, 2006, S. 282]. Informal-Awareness wird auch als „Presence-Awareness“ bezeichnet und in Software über einen Präsenzstatus angezeigt [Riemer et al., 2005, S. 32]. In gängigen Instant-Messaging-Systemen ist diese Form der Statusanzeige üblich.
- Mit sozialem Bewusstsein (social awareness) ist das Bewusstsein über die soziale Situation der Mitglieder gemeint, das Bewusstsein über das, was sie tun, wenn sie mit jemandem sprechen, usw. [Tollmar et al., 1996, S. 289]. In unserer täglichen Arbeit ist soziales Bewusstsein ein Schlüsselement. In dieser Zeit sammeln wir ständig Informationen über unsere Kolleginnen und Kollegen und handeln dementsprechend. Wenn wir sprechen, hören sie zu, wenn sie nicht hier sind, können wir sie telefonisch erreichen oder ihnen Nachrichten hinterlassen. Falls unsere Kolleginnen und Kollegen in der richtigen Stimmung sind, können wir eine Diskussion beginnen, wenn nicht, haben wir diese zu verschieben.
- Gruppenstruktur-Awareness (group-structural awareness) beinhaltet Wissen über die Rollen von Personen und ihre Verantwortlichkeiten, ihre Positionen zu einem Thema, ihren Status und gruppendynamische Prozesse [Gutwin et al., 1996a, S. 6].
- Arbeitsbereich-Awareness (Workspace Awareness) beinhaltet das Bewusstsein von Personen, wie sie mit dem Arbeitsplatz interagieren und das Geschehen an ihm abläuft [Otjacques et al., 2006, S. 94]. Arbeitsbereich-Awareness beinhaltet Wissen, welche Personen in dem Arbeitsbereich arbeiten, was sie tun oder tun werden, wie und wann sie ihre Arbeiten ausführen, und was ihre Motivation ist, es zu tun und warum [Ferscha, 2001, S. 327]. Das Ziel dabei ist, den Aufwand der Koordination von Aufgaben zu reduzieren. Der Teilnehmerinnen- und Teilnehmerwechsel zwischen individueller Arbeit und Arbeit in Gruppen soll erleichtert werden [Schmitz, 2007, S. 187]. Zusätzlich soll ein Kontext

geschaffen werden, in dem Aktivitäten anderer Personen vorausgeahnt und Äußerungen interpretiert werden können.

Bestehende CSCW-Systeme unterscheiden sich in zwei Mechanismen, um Awarenessinformationen zu unterstützen [Dourish und Bellotti, 1992, S. 109]. Einer dieser Mechanismen ist der informelle. Mit ihm wird den Benutzerinnen und Benutzern explizit die Möglichkeit gegeben, sich über die Aktivitäten zu informieren. Der zweite Mechanismus bietet explizite Unterstützung für Rollen (rollenbasiert). Eine Rolle beschreibt die Beziehungen zu den gemeinsamen Arbeitsobjekten und zu anderen Teilnehmerinnen und Teilnehmern und ist in der Regel mit einer Reihe von Operationen verknüpft. Die beiden Mechanismen haben auch Nachteile. Beim informellen Mechanismus bestimmt eine Person, wie viele Informationen sie preisgeben will, ohne selbst direkt davon zu profitieren oder etwas über die Bedürfnisse der anderen Personen zu wissen. Bei dem rollenbasierten Mechanismus sollte beachtet werden, dass Rollen gewechselt werden können und dieser Wechsel soll leicht ermöglicht werden.

Neben der Gruppenawareness gibt es noch individuelle Awareness [Hosni und Khalil, 2004, S. 172]. Diese kann durch die zwei Charakteristiken Nimbus und Focus beschrieben werden. Mit Nimbus und Focus von Benutzerinnen oder Benutzern lässt sich die Stärke der gegenseitigen Awareness beschreiben [Bürger, 1999, S.85]. Mit Nimbus ist der Einfluss gemeint, den eine Benutzerin oder ein Benutzer auf andere Benutzerinnen, Benutzer oder Objekte ausübt [Universität der Bundeswehr München, 2011]. Fokus ist die Menge von Objekten, die im Interessenfeld der Betrachterin oder des Betrachters liegen. Das bedeutet, dass Einzelpersonen verschiedene Nimbus aber den gleichen Focus haben können und umgekehrt [Hosni und Khalil, 2004, S. 172].

Das Design von Groupware ist eine komplexe Tätigkeit [van der Veer und van Welie, 2000, S. 326]. Methoden für die Gestaltung solcher komplexen Systeme müssen viele relevante Aspekte eines Groupware-Systems beachten, einschließlich der Nutzerinnen und Nutzer, ihrer Aufgaben und der Software. Sie müssen zusätzlich die physische und soziale Umwelt des Systems berücksichtigen. Eine dieser Methoden ist User-Centered Design.

2.2 User-Centered Design

Entwicklerinnen und Entwickler von Software denken traditionell technikgetrieben [Grechenig et al., 2010, S. 505]. Es besteht die Gefahr, dass die Bedürfnisse und Wünsche der Benutzerinnen und Benutzer zu wenig beachtet werden. Es kann vorkommen, dass die Entwicklerinnen und Entwickler nicht die späteren Benutzerinnen und Benutzer sind und daher nicht über die Kenntnisse verfügen, welche Dinge der spätere Benutzerpersonkreis wirklich möchte. Zudem kann es passieren, dass Entwicklerinnen und Entwickler sich mehr auf die Entwicklung selbst konzentrieren. Eine einfachere Entwicklung muss nicht automatisch zu einer einfacheren Handhabung führen.

Wählt man bei der Entwicklung und Gestaltung einen menschenzentrierten Ansatz, dann bietet das einen erheblichen sozialen und ökonomischen Vorteil für Anwenderinnen, Anwender, Arbeitgeberin, Arbeitgeber, Anbieterin und Anbieter [DIN EN ISO 9241-210, 2011, S. 8]. User-Centered Design hebt daher die Gefühle und Bedürfnisse der Benutzerinnen und Benutzer hervor

[Wei und Xing, 2010]. Ist das Programm zum Beispiel selbsterklärend und einfach zu bedienen, reduziert sich die Dauer der Schulung, des Erlernens und der Betreuung. Wird die Produktivität der Benutzerinnen und Benutzer gesteigert, wirkt sich das positiv auf das Unternehmen aus. Alle diese Dinge können bewirken, dass Produkte höhere Preise erzielen, sich öfter verkaufen und damit erfolgreicher sind. Daher ist es sinnvoll, die Benutzerinnen und Benutzer bei der Software-Entwicklung von Anfang an mit einzubeziehen. Wenn deren Anforderungen und Bedürfnisse in das zu entwickelnde Produkt einfließen, ist die Wahrscheinlichkeit höher, dass genau diese auch später durch das Produkt erfüllt sind.

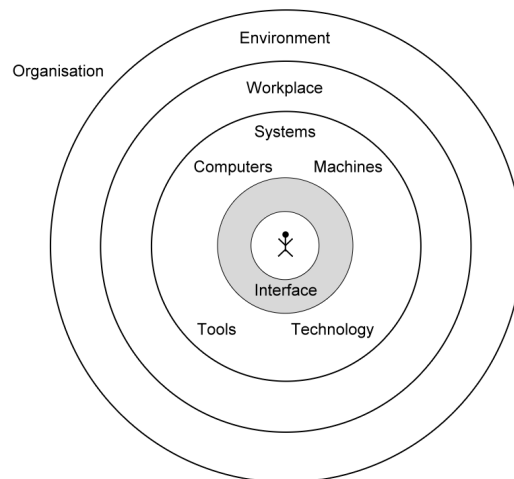


Abbildung 2.2: Darstellung des Designs mit der Benutzerin oder des Benutzers im Mittelpunkt [Noyes und Baber, 1999, S. xi].

User-Centered Design wird graphisch wie in Abbildung 2.2 oft mit Hilfe von immer größer werdenden Kreisen, die zum Beispiel das Interface, die Technologie, der Arbeitsplatz und die Umgebung mit dem Mensch im Zentrum dargestellt. Der internationale Standard ISO 13407 (Human-Centred Design Processes for Interactive Systems) wurde 1999 zugelassen und ist das allgemeine Referenzmodell für User-Centered Design Prinzipien und Prozesse [Jokela, 2002, S. 20]. Die Grundsätze menschenzentrierter Gestaltung sind [DIN EN ISO 9241-210, 2011, S. 9fff]:

1. Benutzerinnen und Benutzer, Arbeitsaufgaben und Arbeitsumgebung umfassend verstehen

Um ein Produkt für Menschen zu gestalten, ist es wichtig, die Personen bzw. Personengruppen zu kennen, die später dieses Produkt verwenden [DIN EN ISO 9241-210, 2011, S. 9]. Eine Hauptursache für den Misserfolg von Produkten ist es, dass die Erfordernisse der zukünftigen Benutzerinnen und Benutzer unvollständig recherchiert wurden. Die Benutzerinnen und Benutzer eines Abwesenheitsplanungsprogramms kommen aus verschiedenen sozialen Schichten. Es kann davon ausgegangen werden, dass die wenigsten Anwenderinnen und Anwender Computerexperten sind, sondern Personen, die eventuell noch nicht sehr viel Erfahrung im Umgang mit Computern haben. Daher will man

die Benutzeroberfläche so einfach wie möglich gestalten, damit ein breites Spektrum von Personen damit zurechtkommt.

In der Literatur findet man mehrere Arten von Nutzerinnen und Nutzern:

- **Unerfahrene Benutzerinnen und Benutzer [Herczeg, 2005, S. 67] / Absolute Anfängerinnen und Anfänger [Thaller, 2002, S. 42]:** Die absolute Anfängerin oder der absolute Anfänger verfügt über keinerlei Erfahrung. Sie oder er kann Anweisungen befolgen und die Reaktion darauf beobachten. Sie treten meistens in Arbeitssituationen auf, wenn sie oder er an einem neuen Arbeitsplatz gerade erst begonnen hat oder mit einem neuen Anwendungssystem zu arbeiten beginnt [Herczeg, 2005, S. 68].
- **Anfängerinnen und Anfänger [Cooper et al., 2007, Thaller, 2002, S. 42]:** Anfängerinnen und Anfänger benötigen viele Erklärungen und Beispiele, um mit dem Programm umgehen zu können. Die Benutzerin oder der Benutzer versteht zwar die Funktion des Gerätes, kann es aber nicht auf eine andere Situation übertragen und dort anwenden. Jede Person verbringt eine Zeit als Anfängerin oder Anfänger, bleibt aber niemals lange in diesem Zustand [Cooper et al., 2007, S. 42]. Sie gelten als inkompetent und daher haben Anfängerinnen und Anfänger das Bedürfnis schnell über dieses Stadium hinauszukommen oder sie geben das Produkt ganz auf.
- **Benutzerinnen und Benutzer mit Vorkenntnissen [Thaller, 2002, S. 42]:** Benutzerinnen und Benutzer verfügen über geringe Erfahrungen mit dem Programm, sie beginnen die Logik zu verstehen, kann daraus aber keine weiteren Schlussfolgerungen ziehen.
- **Fortgeschrittene [Cooper et al., 2007, S. 43]:** Die meisten der Anfängerinnen und Anfänger geben entweder das Produkt auf oder sie kommen irgendwann in das Stadium der Fortgeschrittenen. Die meisten Personen bleiben in diesem Stadium und werden selten zu Experten oder Power Usern.
- **Routinebenutzer [Herczeg, 2005, S. 67]:** Routinebenutzer verwenden das Produkt regelmäßig und nutzen es intensiv.
- **Power User [Thaller, 2002, S. 42]/Expertinnen und Experten [Cooper et al., 2007, S. 43]:** Power User verfügen über systematisches Verständnis des Programms und seiner Funktionen [Thaller, 2002, S. 42].
- **Gelegentliche Benutzerinnen und Benutzer [Siedersleben und Krug, 2003, S.67]/ Außenseiterinnen und Außenseiter [Thaller, 2002, S 42]:** Benutzerinnen und Benutzer verwenden das Programm nur selten, daher wollen sie sich auch nicht näher damit auseinandersetzen. Es macht für sie keinen Sinn, sich näher damit zu befassen, da sie glauben, es bis zum nächsten Gebrauch wieder vergessen zu haben. Sie beschäftigen sich nur mit der vorliegenden Aufgabe.

2. Während der Gestaltung und Entwicklung sind Benutzerinnen und Benutzer mit einzubeziehen

Das Wissen der Anwenderinnen und Anwender ist eine wertvolle Quelle über die zukünftige Benutzung des Produkts. Sie sollen aktiv in die Entwicklung involviert sein. Beim Design eines Programms sind mehrere Interessen zu vereinen. Die Benutzerinnen und Benutzer müssen miteinbezogen werden, da sie bestimmte Vorstellungen haben, wie dieses Programm auszusehen hat. Die Designerin oder der Designer beschäftigt sich mit dem, was die Anwenderinnen und Anwender benötigen, und bildet ihre oder seine eigenen Vorstellungen. Je mehr diese beiden übereinstimmen, umso besser gelingt das Design.

3. Gestaltungslösungen verfeinern und anpassen

Bei der Erstellung von Gestaltungslösungen sollen fortlaufend Anwenderinnen und Anwender involviert werden. Deren Rückmeldungen sollen zum Verfeinern und Anpassen miteinfließen.

4. Iterationen

Bis das gewünschte Ergebnis erreicht ist, wird eine Folge von Schritten immer wieder wiederholt. Diese Abfolge wird Iteration genannt. Mit Hilfe von Iterationen sollen bestehende Unsicherheiten schrittweise entfernt werden. Eine Möglichkeit, um Risiken zu minimieren, bietet die Iteration von Lösungsvorschlägen unter Berücksichtigung von Rückmeldungen aus einer Benutzerperspektive [DIN EN ISO 9241-210, 2011, S. 11]. Für Designer ist es schwierig, alle für Benutzerinnen und Benutzer wichtigen Usability-Kriterien zu kennen oder sich vorzustellen. Das ist nur durch gesammeltes Feedback in einem interaktiven iterativen Prozess durch Einbeziehung von Anwenderinnen und Anwendern möglich [Abrams et al., 2004, S. 5]. Der Prozess von User-Centered Design ist iterativ. Dadurch, dass ähnliche Schritte mehrfach ausgeführt werden, wird das Resultat immer mehr verfeinert [Felix und Rosenbusch, 2011, S. 155]. Die Überprüfung des Konzepts erfolgt während der Entwicklung mit Hilfe von geeigneten Methoden, wie zum Beispiel mit Usability-Tests mit möglichen Anwenderinnen und Anwendern.

5. User Experience berücksichtigen

Im Alltag kommen Personen mit verschiedenen Produkten von verschiedenen Marken in Berührung [Giordano et al., 2005, S. 98]. Alle diese gesammelten Erfahrungen nennt man User Experience. Dabei geht es um den Gesamteindruck einer Applikation und die Art und Weise, wie Benutzerinnen und Benutzer eine Oberfläche eines Programms wahrnehmen [Liebhart et al., 2007, S. 182].

Die aktive Gestaltung der User Experience nennt sich User Experience Design, dessen grundlegende Bedeutung das Sammeln von Daten über die späteren Benutzerinnen und Benutzer und deren Reaktion, ist [Joonhwan et al., 2006, S. 190]. User Experience umfasst wiederum die Gebiete User Research, strategische Planung, Design und Technologie [Giordano et al., 2005, S.98].

User Research kann dabei viele Formen annehmen, von quantitativen Daten über Benutzerinnen- und Benutzerentwicklung, Ergonomie, Bedürfnisse, Verhaltensweisen bis

zu qualitativen Daten über den Benutzerkontext [Macaulay und Busse, 2009, S. 495]. Welche Art von User Research durchgeführt wird, ist je nach Zusammenhang und Projekt unterschiedlich.

Qualitative Research hilft uns im Gegensatz zu quantitativer Research, den Zusammenhang und die Einschränkungen eines Produkts auf verschiedene, nützliche Weisen zu verstehen [Cooper et al., 2007, S. 50]. In der Praxis haben sich Arten von Qualitative Research als nützlich herausgestellt [Cooper et al., 2007, S. 52ff]:

Ein Stakeholder ist eine Person, die für das Design eines Produkts mitverantwortlich oder verantwortlich ist, in der Praxis eine Person, die das Design in Auftrag gibt. Interviews mit Stakeholdern sollte vor dem User Research stattfinden, da die Ergebnisse die Themen und den Verlauf des User Research beeinflussen können. Bei Interviews mit Stakeholdern sind Themen wie Zeitplan, Budget, Geschäftsinteressen und technische Einschränkungen im Mittelpunkt. Stakeholder werden auch als Individuen oder Gruppierungen beschrieben, die auf Unternehmungen einwirken, einwirken können oder umgekehrt [Helm, 2007, S. 168]. Dadurch werden aktiv Ziele und Handlungen beeinflusst. Hierzu gehören auch Kundinnen, Kunden, Userinnen und User, die ebenfalls befragt werden müssen. Bei ihnen handelt es sich oft nicht um dieselben Personen. Kundinnen und Kunden sollten beim Interview nach dem Ziel und dem Entscheidungsprozess beim Kauf des Produktes gefragt werden, warum ihnen die gegenwärtige Situation nicht gefällt, ihre Rolle bei der Installation, Verwaltung und Wartung des Produktes und Probleme des Fachbereichs. Im Mittelpunkt des Designs sollten die aktuellen und die zukünftigen Userinnen und User sein. Interviews mit Fachleuten, für die das Produkt bestimmt ist, sind insbesondere in der Anfangsphase eines Designprojekts oft sehr nützlich [Cooper et al., 2007, S. 54]. Man sollte aber auch während des Prozesses auf sie zurückgreifen. Bei diesen Interviews sollte beachtet werden, dass Fachleute Power User sind. Sie sind eventuell durch lange Erfahrung an gegenwärtige Interaktion gewöhnt. Fachleute sind, auch wenn sie das Produkt kennen, keine Designer und arbeiten in komplexen speziellen Bereichen.

Die Befragung und Beobachtung sollte in deren normaler Arbeitsumgebung stattfinden, da es hier Artefakte geben kann, die ihre Verhaltensweise beeinflussen. Beobachtung und die Befragungen sollten sich dabei abwechseln. Die Designerin oder der Designer muss zwischen den Zeilen lesen und die Fakten für das Design herausarbeiten. Dabei soll sie oder er nicht nur vorbereitete Fragen stellen, sondern solche Daten erfragen, die für die Designprobleme relevant sind.

6. Fachübergreifende Kenntnisse und Gesichtspunkte werden vom Gestaltungsteam, das für den Gestaltungsprozess verantwortlich ist, vereint [DIN EN ISO 9241-210, 2011, S. 12]:

Teams sollten zumindest so groß sein, dass sie folgende Kompetenzbereiche abdecken:

- Usability, Mensch-Computer-Interaktion, Ergonomie, Arbeitswissenschaften
- Benutzerinnen und Benutzer
- Inhaltliche Fachkenntnisse

- Marketing, Verkauf, Wartung, Sicherheit
- Benutzerschnittstellen, Gestaltung (Produkt)
- Schulung, Dokumentation, Benutzerbetreuung
- Geschäftsanalyse, Systemanalyse
- Hardware- und Softwareentwicklung, Programmierung

Die Usability eines Produkts ist abhängig von der jeweiligen Benutzergruppe und deren Kontext [Beier et al., 2002, S. 62]. Folgt man dieser Einsicht, gibt es verschiedene Methoden um Programme dementsprechend zu gestalten, genannt „User-Centred Design“ (UCD).

Methoden

In User-Centered Design gibt es eine Vielzahl von unterschiedlichen Methoden in den verschiedenen Phasen (Anforderung, Design, Prototyping und Evaluierung) [Grechenig et al., 2010, S. 529]:

- Der erste Schritt ist das Herausfinden der Anforderungen, die an das Programm gestellt werden. Nicht nur die Anforderungen an die Funktionalität sollen eruiert werden, sondern auch die an die Usability. Letztere wird nicht durch die Entwicklerinnen und Entwickler festgelegt, sondern durch die zukünftigen Anwenderinnen und Anwender. Dies kann durch Interviews erfolgen, die unterschiedlich umgesetzt werden können. Bei Contextual Inquiry werden die Benutzerinnen und Benutzer nicht nur befragt, sondern auch beobachtet [Holtzblatt et al., 2005, S. 80]. Mit den Interviewpartnerinnen und Partnern soll dabei während des Beobachtens über das aktuelle Geschehen gesprochen werden.

Eine schriftliche Befragung mittels Fragebogen eignet sich insbesondere zum Befragen von großen homogenen Gruppen und ist dabei relativ kostengünstig [Raab-Steiner und Benesch, 2008, S. 44]. Diese Form hat allerdings den Nachteil, dass der Befragungsinhalt schon im Vorfeld feststehen muss und man nicht in die Befragung steuernd eingreifen kann.

Bei Card Sorting werden die Strukturelemente auf Karten geschrieben und die Respondentinnen und Respondenten werden gebeten, diese in Gruppen zusammenzufassen [Kittl, 2009, S. 116]. Die Durchführung in Gruppen hat den Vorteil, dass die Gruppe beobachtet und die Kommunikation dabei zur Nachvollziehung der Vorgangsweise bei der Lösungsfindung hilft. Die Teilnehmerinnen und Teilnehmer bekommen die bereits beschriebenen Karten und leere für neue Elemente, die sie eventuell hinzufügen wollen und die sie sortieren sollen. Auch diese Variante ist kostengünstig, allerdings werden die Funktionen hinter den Elementen nicht berücksichtigt.

- Design: Der User-Centered-Designprozess hängt von den Rahmenbedingungen des einzelnen Projekts ab und wird iterativ durchlaufen [Grechenig et al., 2010, S. 537]. Das Design ändert sich in der Regel im Laufe der Entwicklung. Es gibt allerdings Patterns, die bei der Lösung von Problemen des Designs helfen. Ein Pattern ist eine Lösung für ein

spezielles Designproblem, das immer wieder auftritt und in einem bestimmten Kontext vorkommt. Pattern haben den Vorteil, dass schon bewährte Lösungen verwendet werden, sie sind allerdings keine gesamte Designanleitung, dass sie nur auf Problemlösungen beruhen.

- **Prototyping:** Prototypen helfen Designideen zu untermauern, Annahmen zu testen und man bekommt Feedback von Anwenderinnen und Anwendern [Warfel, 2009]. Mit Hilfe von Prototypen ist es möglich, den späteren Benutzerinnen und Benutzern Ideen zu zeigen und deren Meinungen einzuholen, bevor viel Zeit und Geld in die eigentliche Entwicklung gesteckt wird [Verhoeven und van Germert-Pijnen, 2010, S. 106].
- **Evaluierung:** Bei der Evaluierung wird die Usability eines Produkts überprüft. Bei diesen Usability-Tests wird eine Applikation auf der Gebrauchstauglichkeit geprüft. Die genaue Definition von Gebrauchstauglichkeit bzw. Usability findet man im Abschnitt 2.4.3. Der Test durch Userinnen und User kann erst am bestehenden User Interface oder Prototyp angewandt werden, was ein Nachteil ist, da zuvor schon einiges an Arbeit investiert werden muss [Ivory und Hearst, 2001, S. 471]. Dagegen kann heuristische Evaluierung bereits in früheren Entwicklungsphasen eingesetzt werden.

Wie umfangreich ein Usabilitytest sein soll und wie viele Personen getestet werden sollen, hängt vom Umfang des Projekts ab [Shneiderman, 2002, S. 159]. So kann ein nationales Flugsystem mit verschiedenen Phasen über zwei Jahre getestet werden, während für ein internes, kleines Buchführungssystem ein dreitägiger Test mit 6 Testpersonen reicht.

Laut Jakob Nielsen hat ein typischer Usabilitytest vier Stufen [Nielsen, 1993, S. 187]:

1. **Preparation:** In der Aufbereitungsphase ist von der Experimentatorin oder dem Experimentator darauf zu achten, dass der Raum, in dem das Experiment stattfindet, verfügbar ist, das System gestartet und bereit ist. Der Testarbeitsplatz sollte ergonomisch gestaltet und für die Testpersonen komfortabel sein. Alle Materialien, Instruktionen und Fragestellungen sollen verfügbar sein. Es soll alles für den Test vorbereitet sein, bevor die Userinnen und User eintreffen. Programme, die den Test stören können, zum Beispiel Chat- oder E-Mail-Programme sollen geschlossen sein.
2. **Introduction:** Gibt es mehrere Experimentatorinnen und Experimentatoren, dann soll eine offizielle Experimentperson bestimmt werden, die die Einführung hält, die anderen Personen sollen sich eher im Hintergrund halten. Die Experimentatorin oder der Experimentator gibt den Testerinnen und Testern eine Einführung. Sie oder er erklärt den Personen, worum es sich bei dem Test handelt, und wie er ablaufen soll. Damit die Experimentatorin oder der Experimentator keine wichtigen Dinge vergisst, kann sie oder er eine Checkliste zur Hilfe nehmen. Dennoch soll frei gesprochen und nicht mechanisch von der Checkliste heruntergelesen werden.

Typische Elemente für die Testeinführung wären:

- Die Absicht des Tests ist die Software zu testen und nicht die Testerinnen und den Tester.

- Sollte die Experimentatorin oder der Experimentator auch gleichzeitig die Entwicklerin oder Entwickler der Software sein, sollte dies verschwiegen werden. Die Testerin oder Tester sollte keine Hemmungen haben, Kritik zu äußern, aus Angst vor der Reaktion der Experimentatorin oder des Experimentators.
- Das Resultat des Test kann sich auf das User Interface auswirken, daher kann sich das entgeltliche Interface von dem getesteten unterscheiden.
- Falls das Testergebnis bei der Applikation nicht geheimzuhalten ist, ist es eventuell von Vorteil, wenn eine Diskussion zwischen den Testuserinnen und Usern und ihren Kolleginnen und Kollegen stattfindet.
- Die Mitarbeit der Testerinnen und Testern ist freiwillig, und sie können ihr Mitwirken jederzeit stoppen.
- Die Testresultat werden nicht an Dritte weitergegeben.
- Videos, die bei den Test aufgenommen werden, zeigen nicht die Gesichter der Testpersonen, sondern fokussieren die Tastatur, den Bildschirm und die Rückansicht der Testerinnen und Tester. Eventuell kann so die Angst der Testpersonen beruhigt werden, dass sie aufgenommen werden.
- Den Testerinnen und Testern soll erklärt werden, dass sie während des Tests gerne Fragen stellen sollen. Diese werden allerdings erst später beantwortet, da man herausfinden will, wie gut die Personen mit dem System ohne fremde Hilfe zurechtkommen.
- Spezifische Instruktionen sollen den Testpersonen mitgeteilt werden, wie zum Beispiel, dass sie so schnell wie möglich arbeiten sollen und/oder laut bei ihren Tätigkeiten mitsprechen sollen.
- Es soll nachgefragt werden, ob die Testpersonen noch irgendwelche Fragen zum Testablauf haben.

Nach dieser Einführungsphase bekommen die Testpersonen eine, wenn vorhandene schriftliche Einführung, die sie lesen dürfen und anschließend werden Fragen beantwortet.

3. Während des Testes soll sich die Experimentatorin oder der Experimentator aus dem Test heraushalten. Sie oder er soll keine Tätigkeiten der Testpersonen kommentieren, auch keinerlei Laute, ob etwas gut, schlecht etc. ist, äußern. Erst wenn die Testerin oder Tester unzufrieden mit der Situation ist oder aus früheren Test schon größere Probleme an einer Stelle bekannt sind, ist ein Eingriff möglich.
4. Nach dem Test sollen die Testpersonen nach ihren persönlichen Empfindungen und ihrer Zufriedenheit gefragt werden, wo sie Probleme hatten und was sie verbessern würden.

Bevor man allerdings eine Applikation auf deren Usability überprüfen kann, muss zuerst das User Interface einmal gestaltet werden.

2.3 User Interface

Eine gute Gestaltung von Benutzerschnittstellen ist wichtig, da sie den Kommunikationskanal zwischen Anwenderin oder Anwender und Maschine herstellt [Konrad, 2003, S. 4]. In vielen Themenbereichen gibt es unzählige Programme. Benutzerinnen und Benutzer haben die freie Auswahl. Programme, bei denen die Benutzeroberflächen optisch attraktiv, verständlich und leicht bedienbar und erlernbar sind, haben einen Vorteil.

Mit User Interface werden mehrere Bezeichnungen verbunden [Chlebek, 2006, S. 27]. Manche bezeichnen es auch als Bedieneroberfläche, Dialogmaske, Anwenderoberfläche, Benutzerschnittstelle, Dialogschnittstelle, Softwareoberfläche, Human Computer Interface (HCI), Benutzeroberfläche Graphical User Interface (GUI) oder Mensch-Maschine-Schnittstelle (MMS). Mit all dem ist die Oberfläche gemeint, die Benutzerinnen und Benutzer der Software sehen. Für diese ist sie die eigentliche Software. Heutzutage ist ein User Interface durch eine ereignisorientierten Arbeitsweise gekennzeichnet [Chlebek, 2006, S. 30]. Das heißt, dass die Anwenderin oder der Anwender die Bedienelemente ohne eine bestimmten Reihenfolge beachten zu müssen, bedienen können.

Interaction Design befasst sich, wie der Name schon sagt, mit der Interaktion zwischen einem System mit dessen Benutzerinnen und Benutzern [Richter und Markus, 2010, S. 129]. Es definiert dabei, welche Möglichkeiten die Anwenderinnen und Anwender zur Steuerung haben, sowie das Verhalten, das System und dessen Rückmeldungen.

Bevor man sich allerdings genauer mit dem User Interface befasst, ist es wichtig zu eruieren, wer die späteren Benutzerinnen und Benutzer sind und über welche Funktionen das Programm verfügen soll [Hansen, 1971, S. 527]. Gibt es nur eine Benutzergruppe oder sollen mehrere das Programm später bedienen? Der Nutzungskontext von Benutzergruppen und Anwenderinnen und Anwendern kann sich unterscheiden, ja sogar widersprechen. Es macht einen Unterschied, ob die Anwenderinnen und Anwender mit dem Programm bereits vertraut sind, oder ob es ihnen komplett neu ist. Es muss eruiert werden, welche Daten die Benutzerinnen und Benutzern ansehen, eingeben und manipulieren können und auf welcher Art und Weise das bewerkstelligt werden soll. Ob diese Daten weiterverwendet werden und ob es Optionen und Wahlmöglichkeiten für die Anwenderinnen und Anwendern gibt, ist ebenfalls eine wichtige Frage, die es zu klären gilt. Die Anforderungen, die in der Anfangsphase im Projekt erfasst werden, sind wahrscheinlich nicht ausreichend, denn einige Anforderungen zeigen sich erst dann, wenn ein Lösungsvorschlag vorliegt [DIN EN ISO 9241-210, 2011, S. 14].

Als Entwicklerin oder Entwickler ist es wichtig, sich in die Rolle der Anwenderinnen und Anwender hineinzusetzen [Thissen, 2003, S. 32]. Die Aktionen sollen möglichst mit der realen Welt übereinstimmen [Norman, 1993, S. 72]. Zum Beispiel sollen bei den Daten nicht eine Telefonnummer und eine Mobilfunknummer verlangt sein, da es Personen gibt, die nicht beides besitzen. Begriffe, die für die Funktionen benutzt werden, sollen dem Vokabular der Benutzerinnen und Benutzer entsprechen und nicht dem Vokabular der Entwickler. Mit Wörtern wie Klasse oder Objekt können Laien nur schwer etwas anfangen. Aufgaben, die von den Benutzerinnen und Benutzern am häufigsten verwendet werden (Abwesenheit beantragen, Abwesenheiten genehmigen), sollen einfach aufzufinden sein und so einfach wie möglich gestaltet werden [Shneiderman, 2002, S. 95]. Funktionen, die weniger oft benutzt werden, zum Beispiel die Ein-

stellung der Farbauswahl im Kalender für die Abwesenheiten, können dagegen etwas in den Hintergrund treten und durchaus auch ausgedehnt sein.

Im Computerbereich werden im Großen und Ganzen folgende Varianten an User Interfaces unterschieden [ten Hompel et al., 2008, S. 241]:

- **CLI (Command Line Interpreter)**

Bei CLI findet die Eingabe in einem Textmodus statt. Die Befehle teilweise Abkürzungen werden mit Hilfe einer Tastatur eingegeben. Für Laien stellt diese Variante eine schwierige Form der Bedienung dar.

- **GUI (Graphical User Interface)**

Bei GUI findet die Bedienung meistens mit der Maus statt, kann aber auch durch andere Eingabegeräte erfolgen. Wie der Name schon sagt, wird eine graphische Benutzeroberfläche verwendet. Diese Art der Anwendung ist leichter als bei CLI, da die Benutzerinnen und Benutzer keine Art von „Code“ beherrschen müssen.

- **TUI (Text User Interface)**

TUI sind ähnlich wie CLI allerdings müssen die Anwenderinnen und Anwender nicht die Befehle eingeben. Die Eingabe erfolgt meistens über Masken durch die Tastatur.

- **TUI (Tangible User Interface)**

Bei Tangible User Interfaces ist das grundlegende Gestaltungsprinzip die körperliche Interaktion [Hellige, 2008, S. 235]. Man agiert in seiner materiellen Umgebung und berührt das Interface selber. Solche Systeme stellen eine Herausforderung dar, da Computer und Software mit unsere Umwelt verschmolzen werden.

- **VUI (Voice User Interface)**

Bei VUI reagiert das System auf Sprache. Das gesprochene Wort wird vom System erkannt und interpretiert. Meisten werden zu dieser Eingabemöglichkeiten auch andere angeboten.

2.4 Layout

Eine Applikation, wie Abwesenheitsplanung muss auf verschiedenen Systemen mit unterschiedlicher Hardware funktionieren. Dabei soll das User Interface noch leicht verständlich, erlernbar, schnell ausführbar und optisch ansprechend sein.

Joanna McGrenere und Gale Moore haben eine Studie mit 53 Microsoft Word, Office 97 Usern durchgeführt, bei der die Frage nach der Wahrnehmung der Komplexität im Vordergrund stand [McGrenere und Moore, 2000, S. 4]. Dabei kam unter anderem heraus, dass von den 265 grundlegenden Funktionen 15% (42) überhaupt nicht genutzt wurden, nur 21.5% wurden von mehr als der Hälfte der Teilnehmerinnen und Teilnehmer und nur 3.3% wurden von 3/4 der Teilnehmerinnen und Teilnehmer regelmäßig benutzt. Bei der Entwicklung des Layouts muss daher beachtet werden, dass die Benutzerinnen und Benutzer anscheinend gar nicht so viele

Funktionen wünschen. Bei vielen Funktionen kann es leicht passieren, dass man das Wesentliche übersieht oder es eine längere Suche benötigt, um das Gewünschte zu finden. Man sollte daher die Funktionen auf das Minimum reduzieren. Sie sollten dabei allerdings auch nicht so stark simplifiziert werden, dass sie von Benutzerinnen und Benutzern gar nicht mehr gefunden werden, weil sie die Funktion also solche gar nicht mehr erkennen.

2.4.1 Mapping

Bei der Gestaltung und Funktion von Steuerelementen ist auf Mapping zu achten. Mapping gibt die Beziehung zwischen Aufbereitung der Darstellung und dem aktuellen Ding, das abgebildet wird, an [Norman, 1993, S. 72]. Es beschreibt somit die Beziehung zwischen der Steuerung, der Sache, die es betrifft, und dem gewünschten Ergebnis [Cooper et al., 2007, S. 242].

Menschen stellen zwischen Abstraktionen und realen Dingen Verbindungen her. Wenn man im User Interface Symbole verwendet, sollte man darauf achten. So gibt es bei vielen Programmen einen Button, auf dem ein Drucker abgebildet ist. Den Benutzerinnen und Benutzern ist dadurch schnell klar, dass es sich hierbei um die Druckfunktion handelt. Das Symbol „Papierkorb“ dient zum Löschen. Dabei ist es wichtig, dass die Funktionen prinzipiell die gleichen Eigenschaften haben wie ihre realen Synonyme. Wenn man etwas in den Papierkorb wirft, will man sich im Normalfall davon trennen. Sollte man jedoch zu einem späteren Zeitpunkt bemerken, dass man es dennoch benötigt, kann man es auch wieder aus dem Papierkorb herausfischen. Erst wenn man den Müll wirklich wegbringt (aus der Wohnung/Haus) und dieser von der Müllabfuhr abgeholt wird, dann ist das Weggeworfene erst endgültig verloren. Daher sind die meisten Papierkorb-Funktionen in Software genau so ausgelegt. Man kann etwas in den Papierkorb schieben, wo es noch länger verweilt. Erst wenn man die Datei im Papierkorb löscht, vergleichbar mit der Müllabfuhr, ist es für immer weg.

Vorteile dieser Symbole sind, dass sie von Personen schnell zu erfassen und zu verstehen sind [Cooper et al., 2007, S. 303]. Auch sind sie sprachunabhängig. Menschen in Frankreich werden es genauso verstehen, wie Personen in Österreich oder Griechenland. Ein Problem gibt es, wenn es um komplexere Vorgänge geht. Diese sind schwierig mit einem einfachen Zeichen darzustellen. Diese Vorgänge kann man zum Beispiel durch eine einfache nichtmodale Anzeige in einer Toolbar vorsehen [Cooper et al., 2004, S. 216]. Man sollte aber Befehle und Einrichtungen trennen. Ein Befehl wird im Allgemeinen zehn Mal häufiger aufgerufen als konfiguriert. Der Druckbefehl und die Druckeinrichtung ist zum Beispiel eine solche Situation. Die meisten Personen wollen die Datei nur einmal ausdrucken und verwenden dafür auch meistens den selben Drucker. Diese Funktion lässt sich einfach durch ein Symbol „Drucker“ realisieren. Die Druckeinstellung dagegen, sollte die Einstellungsoptionen bieten, aber auch den Druckbefehl inkludieren. Da die Einstellungen aber seltener durchgeführt werden, können diese in die Toolbar verbannt werden.

Neben dem Bezug von Steuerelementen zwischen der realen Welt und der Funktion, kommt es beim User Interface Design auch auf die Visualisierung generell an.

2.4.2 Visualisierung

Um Mitarbeiterinnen und Mitarbeitern zu ermöglichen, ihre Abwesenheiten mit Hilfe eines Programms selbst zu koordinieren, müssen Daten/Informationen möglichst effizient dargestellt werden. Ben Shneiderman [Shneiderman, 2002, S. 99f] hat 8 Goldene Regeln des Schnittstellendesigns definiert. Diese Prinzipien können in den meisten interaktiven Systemen Anwendung finden.

- **Konsistenz** : Layout, Farb- und Schriftauswahl, Menüs, Hilfestellungen, Menüs, Eingabeaufforderungen etc. sollten durchgehend konsistent angewandt werden.
- **Shortcuts** : Wenn Benutzerinnen und Benutzer ein Programm oft verwenden, wünschen sie Shortcuts, um die Anzahl der Interaktionen zu reduzieren und schneller voranzukommen. Diese Tastenkombinationen werden von erfahrenen Anwenderinnen und Anwendern gerne benutzt.
- **Informatives Feedback**: Bei jeder Handlung sollen Benutzerinnen und Benutzer ein Feedback erhalten. Werden diese Handlungen öfter vollzogen, können die Antworten kürzer ausfallen, als bei Handlungen, die nur selten ausgeführt werden.
- **Geschlossene Dialoge**: Benutzerinnen und Benutzern soll eine geschlossene Handlungssequenz ermöglicht werden. Nach deren Beendigung sollen sie eine Art Zufriedenheit, Erleichterung spüren. Sie sollen wieder bereit sein, eine neue Handlung zu beginnen.
- **Fehlervermeidung, einfaches Umgehen mit Fehlern**: Fehler sollten, wenn möglich, vermieden werden. Wenn es dennoch zu Fehlern kommt, soll die Fehlermeldung den Grund des Fehlers enthalten, sie soll konstruktiv gestaltet werden und nicht mehr technische Details enthalten, als unbedingt notwendig [Shneiderman et al., 1997, S. 8]. Änderungen, um den Fehler zu korrigieren, sollen für Anwenderinnen und Anwender leicht durchzuführen sein.
- **Umkehr von Aktionen**: Wenn es die Aktion erlaubt, sollten diese rückgängig gemacht werden können. Damit wird den Anwenderinnen und Anwendern die Angst genommen, Aktionen bei denen sie sich unsicher sind, auszuführen. Die Gewissheit der Umkehr, ermöglicht den Personen die Aktion einfach auszuprobieren.
- **Interne Kontrollbedürfnisse unterstützen**: Anwenderinnen und Anwendern soll das Gefühl gegeben werden, dass sie die vollständige Kontrolle über das System haben und dass das System ihren Anweisungen folgt. Ansonsten können die Anwendungen ermüdend wirken, Angst und Unzufriedenheit auslösen.
- **Kurzzeitgedächtnis**: „Nach dem sogenannten 7 ± 2 Prinzip können sich Menschen im Schnitt fünf bis neuen Aspekte gleichzeitig merken“ [Friedman, 2009, S. 302]. Nach diesem Prinzip sollen Anwendungen einfach gehalten und die Fenster-Bewegungen auf das Nötigste reduziert werden.

Ein optimales User Interface zu designen, ohne das Fachgebiet Usability Engineering zu berücksichtigen, ist unmöglich [Nielson, 1993, S. 10].

2.4.3 Usability

Unter Usability von Software versteht man die Gebrauchstauglichkeit von Software, damit ist die Benutzerfreundlichkeit gemeint [Richter und Markus, 2010, S. 3]. Sie hängt eng mit User Interface zusammen. Ein gutes User Interface schließt immer auch eine gute Usability ein. Die beiden Thematiken sind daher miteinander verwurzelt und lassen sich schwer trennen.

Gebrauchstauglichkeit ist das Ausmaß, in dem ein Produkt von Benutzerinnen und Benutzer in einem Kontext benutzt wird, um ein bestimmtes Ziel effizient, effektiv und zufriedenstellend zu erreichen [DIN EN ISO 9241-11, 199, S. 4]. Effizienz ist der eingesetzte Aufwand, im Verhältnis zur Genauigkeit und Vollständigkeit, mit dem Anwenderinnen und Anwender ein bestimmtes Ziel erreichen. Unter Effektivität wird die Vollständigkeit und Genauigkeit mit der Benutzerinnen und Benutzer ein bestimmtes Ziel erreichen, verstanden. Zufriedenstellung ist als die Freiheit von Beeinträchtigungen und positiven Einstellungen gegenüber der Nutzung des Produkts definiert.

Damit ein System von Benutzerinnen und Benutzern akzeptiert wird und alle Anforderungen und Bedürfnisse erfüllt sind, muss nicht nur Usability gewährleistet sein. Usability, wie Abbildung 2.3, ist ein Teil von „System acceptability“ [Nielson, 1993, S. 25]. Die Akzeptanz eines Systems besteht neben der „Social acceptability“ aus der „Practical acceptability“. Zu der praktischen Akzeptanz gehören neben den Kosten, der Reliabilität und der Kompatibilität auch, dass das System überhaupt brauchbar („Usefulness“) ist. Um die benötigten Funktionen zu gewährleisten, ist diese Brauchbarkeit in „Utility“ und in „Usability“ unterteilt. Mit „Utility“ ist gemeint, ob das System die Funktionen prinzipiell beherrscht, während er unter „Usability“ die Art und Weise, wie Benutzerinnen und Benutzer diese Funktionen umsetzen, versteht.

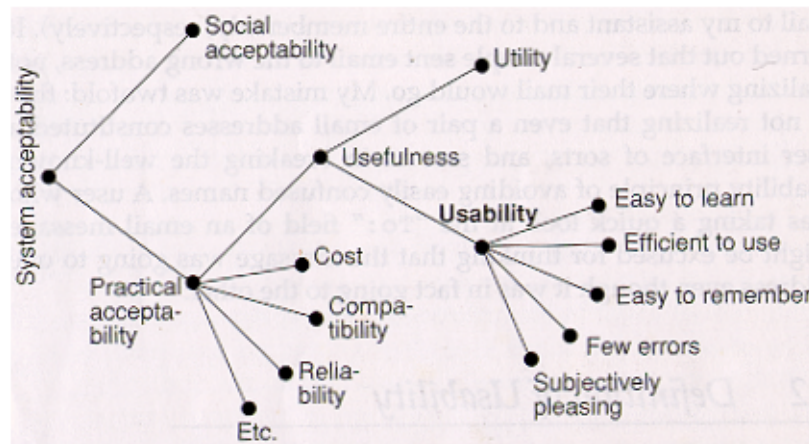


Abbildung 2.3: Ein Modell der Eigenschaften von „system acceptability“ [Nielson, 1993, S. 25].

Es gibt ein paar Regeln, um Usability zu gewährleisten:

- Lernbarkeit [Nielson, 1993, S. 26]: Die Bedienung eines Programms soll von Anfang an keine Schwierigkeiten machen, es soll leicht erlernbar sein, möglichst sofort verstanden werden und bedienbar sein. Instruktionen und Anleitungen sollten, wenn möglich, nicht

benötigt werden. Es ist allerdings auch zu beachten, dass auch fortgeschrittene Anwenderinnen und Anwender das Programm benutzen. Diese sollen nicht für ihr Vorwissen und Können bestraft werden.

- Einprägsamkeit [Nielson, 1993, S. 26]: Auch wenn Anwenderinnen und Anwender das Programm einige Zeit nicht benutzen, sollen sie sich wieder leicht zurechtzufinden und die Bedienbarkeit sollte nicht wieder erlernt werden müssen.
- Insbesondere bei komplexeren Systemen wird eine Hilfefunktion und Bedienungsanleitung benötigt [Niegemann et al., 2008, S. 435]. Diese sollte so verständlich wie möglich formuliert sein.
- Bei Software Produkten werden durchwegs einheitliche Begriffe verwendet, die können aber von Benutzerinnen und Benutzern missverstanden werden [Johnson, 2000a, S. 198]. Funktionen sollen eindeutige Namen besitzen, und es sollte die Möglichkeit ausgeschlossen werden, mit anderen Funktionen verwechselt zu werden. Zum Beispiel kann mit „enter“ gemeint sein, dass man an dieser Stelle etwas eintippen kann/muss oder dass es eine Art „Weiter“ oder „Eintreten“-Button ist, mit dem man in einen anderen Bereich kommt. Als Vokabular sollte das der Benutzerinnen und der Benutzer verwendet werden, auf Programmiererjargon sollte verzichtet werden.
- Viele Software-Produkte verwenden mehrere Begriffe für einheitliche Konzepte, was bei den Benutzerinnen und Benutzern für Verwirrung sorgt [Johnson, 2000b, S. 30]. Daher ist es wichtig, dass die Art und Weise, wie eine Funktion umgesetzt wurde, im ganzen Programm beibehalten wird. Zum Beispiel soll es nicht einmal „bearbeiten“, dann „ändern“ oder „umändern“ heißen. Die Bezeichnungen sollten mit denen in der Bedienungsanleitung übereinstimmen. Nachträgliche Umbenennungen von Funktionen sollten auch hier geändert werden, ansonsten kommt es bei den Benutzerinnen und Benutzern zu Verwirrung wenn zum Beispiel eine Button in der Anleitung „beenden“ heißt und in der Applikation „schließen“. Anwenderinnen und Anwender sind zu sehr mit der Arbeit an sich beschäftigt und wollen sich nicht weiter mit der Bedienung befassen. Personen, die sich voll und ganz auf eine Tätigkeit konzentrieren, verlieren das Bewusstsein für periphere Probleme und Ablenkung [Cooper et al., 2007, S. 201].
- Ein und dieselbe Bezeichnung für verschiedene Funktionen schafft ebenfalls Verwirrung.
- Programmfenster müssen einen eindeutigen Titel haben [Johnson, 2000a, S. 236]. Zwei oder mehrere Fenster dürfen nicht denselben Titel besitzen. Dieses Szenario kann leicht entstehen, wenn der Programmierer den Code kopiert, dabei aber vergisst den Titel zu ändern.
- Fehler [Nielson, 1993, S. 26]: Programme sollten so gestaltet werden, dass die Benutzerinnen und Benutzer schon im Vorhinein wenige Fehler machen können. Sollten Fehler dennoch vorkommen, sollen die Meldungen verständlich und klar formuliert werden, damit diese korrigiert werden können. Zum Beispiel mit „Error: 123“ können Anwenderinnen und Anwender wenig anfangen.

Auch wenn die Anwenderin oder der Anwender einen Fehler begangen hat, ist es wichtig, dass sie oder er ein Feedback bekommt. Würde das Programm einfach nichts machen und weiter auf Anweisungen „warten“, würde das die Benutzerin oder den Benutzer verwirren, denn sie oder er wüsste nicht, ob das Programm abgestürzt ist, oder ob etwas falsch gemacht wurde. Wenn eine Fehlermeldung ausgegeben wird, wäre es sinnvoll, dass sie oder er den Grund dafür erhalten würde. Als Anwenderin oder Anwender kann man oft viele Fehler begehen, wenn die Fehlermeldung dies einschließt, kann es helfen zu erkennen, was man falsch gemacht hat und den Fehler zu korrigieren.

Fehlermeldungen sollen keine negative Wirkung auf Anwenderinnen und Anwender haben und eine verletzende Terminologie besitzen [Shneiderman, 2002, S. 436]. Dies könnte Anwenderinnen und Anwender vor den Kopf stoßen. Stattdessen sollen Informationen angeboten werden, warum dieser Fehler aufgetreten ist und eine Hilfestellung bieten. Wörter wie „Fehler“, „Illegal“, „Schlecht“ oder „Ungültig“ sollen vermieden werden und nur, wenn es keine Alternativen gibt, verwendet werden.

Rund ein Drittel der Fehler bei Computersystemen ist auf menschliche Faktoren zurückzuführen [Thaller, 2002, S. 63]. Entwicklerinnen und Entwickler müssen sich bewusst sein, dass nicht sie selbst, sondern andere Personen das Programm bedienen. Andere Personen denken anders, haben andere Erfahrungen, anderes Wissen und einen anderen Hintergrund. In der direkten menschlichen Kommunikation geben Personen durch ihre Körpersprache ihrem Gegenüber zu verstehen, dass sie an der Kommunikation teilnehmen. Bei der Kommunikation durch technische Medien muss diese Teilnahme mit anderen Mitteln ausgeglichen werden.

- Geringe Fehlerrate: Es muss auf eine geringe Fehlerrate geachtet werden [Holzinger, 2005, S. 72]. Die Anwenderinnen und Anwender sollen von vornherein keine Fehler machen können. Lassen sich Fehler nicht vermeiden, dann soll es für die Personen so einfach wie möglich sein, ihre gemachten Fehler zu korrigieren.
- Effizienz [Nielson, 1993, S. 26]: Das Programm soll effizient arbeiten. Wenn die Bedienung einmal erlernt wurde, sollen Anforderungen der Anwenderinnen und Anwender erfüllt werden.
- Wie im Kapitel User Interface beschrieben, soll das Programm die reale Welt darstellen.
- Die geringe Wartezeit ist ebenfalls ein wichtiges Kriterium. Dabei muss nicht unbedingt das Ergebnis sofort angezeigt werden, das ist auch nicht immer möglich, sondern eine einfache Meldung, dass das System die Eingabe entgegengenommen hat und nun verarbeitet [Johnson, 2000a, S. 384]. Damit wird den Anwenderinnen und Anwendern geholfen das Programm zu verstehen.

Menschen sind auf Feedback angewiesen und dieses Feedback sollte schnell erfolgen, noch innerhalb von 0,1 Sekunden, ansonsten hat die Anwenderin oder der Anwender das Gefühl, das System funktioniert nicht oder ihre Aktion ist gar nicht vom System empfangen worden und sie versuchen es noch einmal [Johnson, 2000a, S. 389]. Bei Software

erwartet die Benutzerin oder der Benutzer in der Regel, dass Aktionen sofort durchgeführt werden. Leider kann es vorkommen, dass etwas ein paar Sekunden oder auch Minuten dauert. Damit die Anwenderin oder Anwender nicht glaubt, das Programm wäre abgestürzt, ist ein Feedback von Nöten. Das kann auf unterschiedliche Art und Weise erfolgen. Wie sie konkret aussehen, liegt im Ermessen der Designerinnen und Designer, beziehungsweise der Entwicklerinnen und Entwickler. Es kann zum Beispiel ein Balken mit einer Prozentangabe sein oder eine Sanduhr. Damit weiß die Benutzerin oder der Benutzer, dass das Programm reagiert hat und einfach etwas länger für den Ablauf benötigt.

Untersuchungen haben zum Verhalten von Menschen im Umgang mit dem Computer „Toleranzgrenzen“ ermittelt, die eine Erwartungshaltung widerspiegeln [Thissen, 2003, S. 261].

- Die Grenze für das Gefühl der Benutzerin oder des Benutzers, dass ein System schnell reagiert und fehlerfrei ist, beträgt 1/10 Sekunde.
- Bei einer Wartezeit von 1 Sekunde hat die Anwenderin oder der Anwender noch das Gefühl, dass das System ordnungsgemäß arbeitet. Wenn der Prozess länger dauert, sollte das System ein Feedback geben, dass es die Eingabe empfangen hat, aber es zum Verarbeiten länger benötigt. Hilfreich ist hierbei, der Benutzerin oder dem Benutzer eine Abbruchmöglichkeit zu bieten.
- Nach 10 Sekunden schwindet die Aufmerksamkeit der Anwenderin oder des Anwenders. Sie oder er bekommt das Gefühl, dass das System nicht richtig funktioniert. Eventuell versucht sie bzw. er die Eingabe ein zweites Mal oder sie bzw. er bricht die Aktion ganz ab.

Neben den Forschungsgebiet CSCW und User Interface Design müssen rechtliche Fragen noch geklärt werden, wie prinzipiell Abwesenheitsplanung aussieht und welche Rechte damit verbunden sind.

2.5 Abwesenheitsplanung

Die grundlegende Aufgabe der Abwesenheitsapplikation ist die Koordination der Abwesenheitstermine. Zur Lösung von Terminproblemen müssen zunächst Fragen formal formuliert werden [Burger, 1997, S. 167]:

1. Wer muss - soll - darf in welcher Rolle teilnehmen?

Im Unternehmen dürfen alle Mitarbeiterinnen und Mitarbeiter Abwesenheiten beantragen. Es ist genau geregelt, welche Personen Vorgesetzte sind und wie viele Personen und welche genau unter ihnen stehen. Vorgesetzte dürfen Abwesenheiten im Gegensatz zu den anderen Personen nicht nur beantragen, sondern diese auch ihren Mitarbeiterinnen und Mitarbeitern genehmigen. Wie viel Abwesenheit zu genehmigen ist, und inwieweit Vorgesetzte die Abwesenheit ihrer Mitarbeiterinnen und Mitarbeiter bestimmen dürfen, wird im nächsten Kapitel Arbeitsrecht 2.6 näher behandelt.

2. Wer kann in welcher Rolle teilnehmen?

Jede Mitarbeiterin und jeder Mitarbeiter kann, muss aber nicht, eine oder mehrere Vertretungen haben. Diese Rollen legen fest, ob eine Person vom Unternehmen abwesend sein darf oder nicht. Wenn jemand eine Vertretung hat, muss davon ausgegangen werden, dass dessen Arbeit nicht liegen bleiben kann, sondern von der Vertretung ganz bzw. zum Teil mitübernommen werden muss. Daher muss in dem Fall, dass die Person Abwesenheit konsumieren will, deren Vertretung im Unternehmen anwesend sein.

3. Wann und über welchen Zeitraum hinweg sind die einzelnen Teilnehmerinnen und Teilnehmer mit den Aktivitäten beschäftigt?

Die Aktivität im Falle der Abwesenheitsplanung ist die Abwesenheit selbst. Wie lange diese Zeiträume prinzipiell dauern dürfen, hängt mit dem Arbeitsrecht zusammen, das im gleichnamigen Kapitel 2.6 näher behandelt wird.

4. Wie wichtig sind die Aktivitäten?

Wie wichtig eine Abwesenheit ist, kommt auf die jeweilige Art an. Ein Krankenhausaufenthalt hat höhere Priorität als ein Urlaub. Über wieviel Urlaub eine Mitarbeiterin und ein Mitarbeiter selbst bestimmen kann und wie weit das der Arbeitgeber vorschreiben kann, wird im folgenden Kapitel näher behandelt.

2.6 Arbeitsrecht

Abwesenheiten von Mitarbeiterinnen und Mitarbeitern sind im Arbeitsrecht geregelt. Unterschieden wird dabei das Arbeitsrecht in der Privatwirtschaft und im öffentlichen Dienst. In Österreich gibt es über 20 öffentliche Dienstrechtsordnungen. Da der Prototyp die Abwesenheitsplanung für privatwirtschaftliche Betriebe umfassen soll, wird nur dieses näher behandelt [Gärtner et al., 2007, S. 3].

Laut Arbeitsrecht gibt es verschiedene Abwesenheiten:

1. Urlaub

Jeder Arbeitnehmerin und jedem Arbeitnehmer steht ein bezahlter Urlaub zu. Beträgt die Dienstzeit weniger als 25 Dienstjahre sind es 30 Werktage. Nach Vollendung des 25. Dienstjahres bekommt man 6 Werktage mehr, daher 36 Werktage [WKO, 2010a] .

Ein Werktag ist ein Wochentag von Montag bis einschließlich Samstag. Arbeiten die Arbeitnehmerin oder der Arbeitnehmer nur 5 oder weniger Tage in der Woche, dann ist der Urlaubsanspruch dementsprechend umzurechnen. Bei 5 Tagen in der Woche, wären es zum Beispiel 25 Arbeitstage.

Bei den Dienstzeiten sind alle Dienstjahre bei derselben Arbeitgeberin oder demselben Arbeitgeber zusammenzurechnen, auch wenn es unterschiedliche Dienstverhältnisse (zum Beispiel Lehrverhältnis) waren. Die Unterbrechung der Dienstzeiten darf aber nicht mehr als drei Monate betragen.

Zu welchem Zeitpunkt und wie lange eine Arbeitnehmerin oder ein Arbeitnehmer Urlaub beanspruchen kann, ist immer mit der Arbeitgeberin oder dem Arbeitgeber zu vereinbaren [AK2, 2011]. Dabei sollte sowohl auf die Erholungsmöglichkeit der Arbeitnehmerin oder des Arbeitnehmers als auch auf das Interesse des Unternehmens Rücksicht genommen werden.

Die Arbeitnehmerin oder der Arbeitnehmer kann Urlaub ohne die Zustimmung der Arbeitgeberin oder des Arbeitgebers beanspruchen [Arb, 2010]. Es müssen allerdings mehrere Voraussetzungen erfüllt sein. So muss im Betrieb ein zuständiger Betriebsrat vorhanden sein, die Arbeitnehmerin oder der Arbeitnehmer den von ihm gewünschten Urlaubszeitpunkt mindestens drei Monate im Voraus bekannt gegeben haben und mindestens zwei Wochen auf einmal verbrauchen. Gibt es im Betrieb keinen Betriebsrat, muss zwischen der Arbeitnehmerin oder dem Arbeitnehmer und der Arbeitgeberin oder dem Arbeitgeber eine Vereinbarung bezüglich des Urlaubes getroffen werden. Jugendlichen (bis zur Vollendung des 18. Lebensjahres) ist zwischen 15. Juni und 15. September auf Verlangen Urlaub von mindestens zwölf Werktagen zu genehmigen.

2. Krankenstand [AKK, 2011a]

Ist eine Arbeitnehmerin oder ein Arbeitnehmer erkrankt, kann sie bzw. er nicht den Krankenstand im Voraus genehmigen lassen. Es wäre aber dennoch von Vorteil, wenn die oder der Vorgesetzte die Abwesenheit in der Applikation einträgt, da die Arbeitskolleginnen und Kollegen von der Abwesenheit der oder des Erkrankten wissen und ihr Zeitmanagement danach ausrichten können.

Kann der Krankenstand im Voraus beantragt werden, muss das von der Arbeitnehmerin oder dem Arbeitnehmer in der Applikation gemacht werden. Beispiele dafür wären notwendige Arztbesuche oder ein Kuraufenthalt. Ist letzterer von einer Krankenkasse oder Pensionsversicherung genehmigt, muss dieser von der Arbeitgeberin oder dem Arbeitgeber akzeptiert werden [AKK, 2011b].

3. Krankenpflegefreistellung / Betreuungsfreistellung [WKO, 2010b]

Freistellung von der Arbeitsleistung, kann aus zwei Gründen erfolgen:

- Krankenpflegefreistellung: Pflege eines erkrankten, im gemeinsamen Haushalt lebenden nahen Angehörigen oder
- Betreuungsfreistellung zur notwendigen Betreuung eines Kindes bei Ausfall der ständigen Betreuungsperson [WKO, 2010c].

Die Krankenpflegefreistellung kann auch nur für Stunden erfolgen. Das maximale Ausmaß ist die Dauer einer regelmäßigen wöchentlichen Arbeitszeit pro Arbeitsjahr. Dieser kann sich bei Leistung von regelmäßigen Überstunden erhöhen. Ist ein Kind (unter 12 Jahre) der Arbeitnehmerin oder des Arbeitnehmers erkrankt, kann der Pflegeurlaubsanspruch um eine Arbeitswoche erhöht werden, falls der Grundanspruch schon verbraucht ist.

Nahe Angehörigen

Eine nahe Angehörige oder ein naher Angehöriger ist eine Person, mit der die Arbeitnehmerin oder der Arbeitnehmer verheiratet ist. Auch eine Lebensgefährtin oder Lebensgefährte zählt als solche, sofern die Beziehung aus wirtschaftlicher Sicht einer Ehe gleicht. Eltern, Großeltern, Kinder, Enkelkinder, auch Pflege- und Wahlkinder sind als Verwandte gerader Linie als nahe Angehörige zu bezeichnen.

Pflegebedürftigkeit

Eine Pflegebedürftigkeit liegt dann vor, wenn eine nahe Angehörige oder ein naher Angehöriger nicht sich selbst überlassen werden kann. Gründe können eine Erkrankung der Person sein oder ein Kind unter 12 Jahren, dessen Betreuung nicht gegeben ist. Die Entscheidung, wann eine Pflegebedürftigkeit besteht und wie lange der Zeitraum dafür ist, unterliegt der Beurteilung einer Ärztin oder eines Arztes.

4. Persönliche Dienstverhinderungsgründe

Bei bestimmten persönlichen Dienstverhinderungsgründen muss die Arbeitnehmerin oder der Arbeitnehmer nicht an seinem Arbeitsplatz erscheinen. Das Entgelt muss von der Arbeitgeberin oder dem Arbeitgeber weitergezahlt werden. Persönlichen Dienstverhinderungsgründe sind zum Beispiel Todesfälle, Hochzeiten, wenn es nahe Angehörige betrifft, Vorladung von Ämtern oder auch Niederkunft der Ehegattin.

Eine Arbeitnehmerin oder ein Arbeitnehmer behält den Anspruch auf das Entgelt, wenn er

- ohne sein Verschulden
- durch wichtige Gründe, die ihre oder seine Person betreffen
- während einer verhältnismäßig kurzen Zeit

an der Leistung seiner Dienste verhindert ist [WKO, 2010c].

5. Mutterschutz

Die Schutzfrist beginnt 8 Wochen vor den Entbindungstermin [AK3, 2011]. Es besteht die Möglichkeit, dass diese Frist aus gesundheitlichen Gründen der werdenden Mutter und/oder des Kindes verlängert wird. Tritt die Geburt vor oder nach dem Entbindungstermin ein, wird die Schutzfrist verkürzt oder verlängert.

6. Karenz

Nach dem Ende der Mutterschutzfrist beginnt die Karenz [AK, 2010]. Diese kann maximal bis zum zweiten Geburtstag des Kindes dauern. Wie lange die Mutter tatsächlich in Karenz gehen will, muss sie bis spätestens 8 Wochen nach der Geburt bekannt geben. Zwischen den Eltern kann die Karenz bis zu zwei Mal geteilt werden.

7. Bildungskarenz

Zwischen Arbeitgeberin oder Arbeitgeber und Arbeitnehmerin oder Arbeitnehmer kann eine Bildungskarenz vereinbart werden [Bil, 2011]. Voraussetzung ist, dass die Arbeitnehmerin oder der Arbeitnehmer sich im oder nach dem siebten Arbeitsmonat befindet.

Die Bildungskarenz muss mindestens zwei Monaten und darf maximal ein Jahr dauern. Sie kann in Teile aufgeteilt werden, wobei ein Teil mindestens zwei Monate dauern muss und die Gesamtdauer der einzelnen Teile ein Jahr nicht überschreiten darf. Zwischen Bildungskarenzen müssen mindestens vier Jahre liegen.

8. Weiterbildung

Durch die ständige Veränderung der Arbeitswelt ist es wichtig, die Arbeitnehmerinnen und Arbeitnehmer aus- und weiterzubilden [Wei, 2011].

9. Zeitausgleich

Es liegen Überstunden vor, wenn entweder

- die wöchentlich zulässige Normalarbeitszeit überschritten oder
- die tägliche Normalarbeitszeit überschritten wird [AZG, 2010, §6.(1)].

Wird bei Gleitzeitarbeit das Zeitguthaben in die nächste Gleizeitperiode übertragen, gilt es nicht als Überstunden. Für Überstunden gebührt

- 50% Zuschlag oder
- Zeitausgleich. Bei der Bemessung des Zeitausgleichs ist der Überstundenzuschlag zu berücksichtigen oder gesondert auszuzahlen [AZG, 2010, §10.(1)].

10. Ersatzruhe

Die Arbeitnehmerin oder der Arbeitnehmer hat, wenn sie oder er während ihrer oder seiner wöchentlichen Ruhezeit (§ 2 Abs. 1 Z 3) beschäftigt wird, in der folgenden Arbeitswoche Anspruch auf Ersatzruhe, die auf seine Wochenarbeitszeit anzurechnen ist [ARG, 2010, §6.(1)].

Auch wenn Arbeitnehmerinnen und Arbeitnehmer alle diese Abwesenheiten in Anspruch nehmen dürfen bzw. müssen, stellt sich noch die Frage, inwieweit Kollegen davon wissen dürfen. Diese Frage wurde im nächsten Kapitel über Datenschutz zu klären versucht.

2.7 Datenschutz

Da Mitarbeiterinnen und Mitarbeiter ihre Abwesenheiten mit ihren Kolleginnen und Kollegen koordinieren müssen, ist die Frage zu klären, inwieweit sie dabei erfahren dürfen, um welche Art von Abwesenheit es sich handelt und welche Auswertungen darüber gemacht werden dürfen. Dabei ist das Datenschutzgesetz näher zu beachten.

Im Datenschutzgesetz 2000 sind die Regeln zur Verwendung von personenbezogenen Daten geregelt [RIS, 2011]. Personenbezogene Daten sind darin alle Daten, über die die Identität des Betroffenen bestimmt oder bestimmbar ist.

Die Betroffenen können dabei sowohl eine natürliche als auch eine juristische Person sein, aber auch eine Personengemeinschaft ist möglich. Da es sich in diesem Fall um Abwesenheitsplanung in einem Unternehmen handelt, ist davon auszugehen, dass alle Betroffenen natürliche

Personen sind. Prinzipiell sind personenbezogene Daten geheim zu halten, außer es geht um lebenswichtige Interessen des Betroffenen, die betroffene Person hat zugestimmt oder durch Eingriff einer staatliche Behörde auf Grund von Gesetzen. Diese Gesetze dürfen die Verwendung von besonders schutzwürdigen Daten nur vorsehen, wenn es zur Wahrung wichtiger Interessen dient und diese müssen dabei ausreichend geschützt werden, damit das Geheimhaltungsinteresse der Betroffenen gewahrt bleibt [RIS, 2011, §1.(2)] .

Unter besonders schutzwürdige Daten (sensible Daten) sind Daten von natürlichen Personen über ihre Religionszugehörigkeit, ethnische und rassische Herkunft, philosophische Überzeugung, Sexualleben, politische Meinung, Gewerkschaftszugehörigkeit und ihrer Gesundheit gemeint.

Es gibt Gründe, wann das schutzwürdige Geheimhaltungsinteresse von sensiblen Daten verletzt werden darf. Das sind unter anderem:

- Die oder der Betroffene veröffentlicht selbst die Daten.
- Die Daten sind nur indirekt personenbezogen.
- Die oder der Betroffene hat seine ausdrückliche Zustimmung zur Verwendung seiner sensiblen Daten geben. Er hat allerdings die Möglichkeit diese Zustimmung zu widerrufen, dann ist eine weitere Verwendung der Daten untersagt.

Arbeitnehmerinnen und Arbeitnehmer achten darauf, welche Informationen sie der Arbeitgeberin oder dem Arbeitgeber preisgeben und welche archiviert werden. Immerhin können diese Daten für die zukünftige Karriere entscheidend sein. Sollte es in einem Unternehmen in Österreich Systeme geben, die personenbezogene Daten verarbeiten, dann muss dies durch eine Betriebsvereinbarung geregelt werden [Formann und Kalliauer, 2010].

Um den Datenschutz bei der Abwesenheitsplanung zu berücksichtigen, werden nur die Vorgesetzten den Grund der Abwesenheit erfahren. Kolleginnen und Kollegen müssen nicht den Grund der Abwesenheit erfahren, da es für sie keinen Unterschied macht, ob sie die Kollegin oder den Kollegen vertreten müssen, wenn er auf Urlaub, Pflegeurlaub oder im Krankenstand ist. Die Person ist für sie einfach abwesend und muss daher vertreten werden.

2.8 SAP

Für die diese Arbeit wurde die Abwesenheitsplanung eines Unternehmens näher untersucht. Zwar werden die Abwesenheit im UrlaubsManger der Firma AddWare Solutions geregelt, die Daten werden dann aber händisch in SAP eingegeben, da es für die Personalwirtschaft verwendet wird.

SAP ist ein Enterprise Resource Planning (ERP). ERP gehört zum Bereich der Business Applikationen. Darunter ist ein Softwaresystem zu verstehen, bei dem mehrere betriebswirtschaftliche Applikationen durch eine gemeinsame Datenbasis integriert sind [Gadatsch, 2001, S. 71]. ERP wird vor allem in mittelständischen und größeren Unternehmen verwendet. Enterprise Resource Planing wurden seit den 1990er Jahren rund um die Welt implementiert, um ein

einheitliches Informationssystem im Unternehmen zu haben und die Geschäftsprozesse umzuarbeiten [Rajagopal, 2002, S. 87]. Ein sehr gefragtes ERP-System ist SAP. Bereits 1999 haben sich fünf Hauptanbieter von ERP-Software-Systemen herausgebildet, mit der SAP AG als Führer mit mehr als 30% Marktanteil [Brown, 1999, S. 411].

SAP R/3 gliedert sich in folgende vier Hauptbereiche: Rechnungswesen, Logistik, Personalwirtschaft und Basissystem/Web Application Server (Web AS) [Krämer et al., 2003, S. 21].

2.8.1 SAP ERP Human Capital Management

Die Abwesenheitsplanung ist in SAP HCM (Human Capital Management) enthalten. SAP HCM ist ein Modul für Personalwirtschaft und laut der Webseite von SAP [SAP, 2010c] sind die wichtigsten Einsatzmöglichkeiten:

- **Mitarbeiter- und Manager-Self-Services:** Die Mitarbeiterinnen und Mitarbeiter können über die gesamte Organisation auf Funktionen, ERP-Services und Business Content zugreifen.
- **Workforce Analytics:** In Echtzeit kann die aktuelle Situation der Belegschaft überblickt werden.
- **Talent Management:** Mitarbeiterinnen und Mitarbeiter können bei der Fort- und Weiterbildung bis hin zur Karriereplanung unterstützt werden.
- **Workforce Process Management:** Daten und Prozesse, die für die Personalarbeit wichtig sind, werden standardisiert und in einer Plattform zusammengefügt. Personalarbeit kann zum Beispiel Organisationsmanagement, Personalverwaltung, Vergütungssysteme, Zeitmanagement und Gehalts- und Lohnabrechnung sein.
- **Workforce Deployment:** Wird unter anderem zum Überwachen von Projekten eingesetzt.

SAP NetWeaver PORTAL

Dass jede Mitarbeiterin und jeder Mitarbeiter ihre oder seine Abwesenheitszeiten selbst einreichen kann, ist in der Standardversion von SAP nicht vorgesehen. Dazu werden zusätzlich unter anderem ein NetWeaver Portal (früher: Enterprise Portal) benötigt, in dem der Bereich Abwesenheitsplanung eingebettet ist. Ein Portal kann man sich als visuelles Interface vorstellen, das eine Verbindung zwischen der Anwenderin oder dem Anwender und der Applikation herstellt [Gomis, 2007].

SAP bietet dabei eine Vielzahl von Applikationen an, die durch das NetWeaver Portal für den Anwender leichter zu bedienen sind. Das SAP Portal ist mit seinen Design-Tools sehr umfassend und mächtig, allerdings auch an einigen Stellen etwas unübersichtlich [Weißberg und Stemmer, 2009, S. 11]. Falls nur die Abwesenheitsplanung benötigt wird, wären zusätzliche Funktionen für die Benutzerinnen und Benutzer mehr verwirrend als hilfreich. Renate Kristiansen [Kristiansen, 2005, S. 5] hat verschiedene Userinnen und User in 5 Firmen in Norwegen interviewt, die mit verschiedenen Versionen von SAP Client und SAP Enterprise Portal arbeiten. Dabei kam heraus, dass die komplexe Bildschirmkonfiguration keine Hilfe

leistet. Einige von der Interviewpartner erzählten, dass sie nur etwa 20% des am Bildschirm Dargestellten, verstanden hätten.

Zudem biete das Enterprise Portal auch in technischer Hinsicht Hindernisse, so ist es nach SAPs eigenen Schulungsunterlagen nur mit einer ganz speziellen Version eines bestimmten Betriebssystems nutzbar [Jeske, 2005, S. 259].

Das in dieser Arbeit untersuchte Unternehmen verfügt über kein SAP NetWeaver Portal. Es hat sich aus finanziellen und technischen Gründen dagegen entschieden.

SAP und Abwesenheitsplanung

Abwesenheiten in einem SAP-System einzutragen, erfolgt entweder direkt im System oder über ein Mitarbeiterportal auf Basis von SAP Enterprise/NetWeaver Portal, wobei NetWeaver den aktuellen Namen darstellt [Frevert, 2011, S. 55]. Das Portal kann als visuelle Schnittstelle zwischen einem Benutzer und allen grundlegenden Anwendungen und Informationen eines Unternehmens gesehen werden [Gomis, 2007].

Mit Hilfe dieses Portals können Mitarbeiterinnen und Mitarbeiter Zugriff auf eine Reihe von Services haben, zum Beispiel Informationen, Personensuche, Handbücher oder können ihre eigenen Personaldaten pflegen. Die Integration von SAP NetWeaver Portal und ERP-System erleichtert auch die Genehmigung von Urlaub. Falls eine Mitarbeiterin oder ein Mitarbeiter einen Urlaubsantrag ausgefüllt hat, erhält die oder der zuständige Vorgesetzte einen Service aus SAP ERP die fällige Genehmigung als Aufgabe zugewiesen. Im Falle der Freigabe wird der Urlaub im ERP-System verbucht und die Mitarbeiterin oder der Mitarbeiter bekommt eine entsprechende Rückmeldung.

Wurde in einem früheren Kapitel der Datenschutz näher behandelt, stellt sich nun die Frage, wie dieser in einem SAP-System umgesetzt wurde.

2.8.2 SAP und der Datenschutz

Die Komplexität eines SAP-Systems hat zu einer Diskussion bezüglich des Datenschutzes geführt [SAP, 2011b]. Früher bezog sich diese Diskussion vermehrt auf den Arbeitnehmerinnen- und Arbeitnehmer-Datenschutz. Man hatte die Befürchtung, dass Daten von einem Bereich im Unternehmen, in einem anderen Bereich auftauchen und dort ausgewertet werden könnten.

Heute bezieht sich die Diskussion mehr auf das SAP-Berechtigungskonzept. Darin werden die Funktionen der einzelnen Anwenderinnen und Anwender und auf welche Funktionen sie oder er zugreifen darf festgelegt. Dieses Berechtigungskonzept ist sehr komplex. Das R/3-Release 3.1i umfasst 543 Berechtigungsobjekte. Bei 4.6X spricht man von ca. 900 Berechtigungsobjekten. Daher ist es nicht verwunderlich, dass in der Praxis zahlreiche Benutzer die Berechtigung SAP_All besitzen.

Die Komplexität der Benutzerverwaltung führt dazu, dass Systemverwalter die Standardprofile überprüfen, ob diese für die Benutzerinnen und Benutzer ausreichend sind. Wenn deren Rechte nicht ausreichen, werden sie erweitert, bis der Zugriff auf die gewünschten Ressourcen gegeben ist. Diese Vorgangsweise hat den Nachteil, dass dadurch nicht überprüft wird, ob durch

die Vergabe der Rechte auch Zugriff auf Daten gewährt werden, die die Anwenderinnen und Anwender eigentlich nicht benötigen.

2.9 Verteilte Systeme

Da die Abwesenheitsplanung lokal auf einem Rechner laufen soll, das SAP-System allerdings auf einen Server ausgelagert ist, handelt es sich dabei um ein verteiltes System mit bestimmten Eigenschaften.

Ein verteiltes System wirkt für die Benutzerinnen und Benutzer wie ein einfaches, zentralisiertes System, aber es läuft auf mehreren unabhängigen CPUs (central processing units) [Tanenbaum und van Renesse, 1986, S. 419]. Vorteile von verteilten Systemen sind, dass sie im Prinzip einfach zu erweitern sind und immer zur Verfügung stehen, da sie auch weiter verwendet werden können, wenn einzelne Komponenten ausfallen. Auch kann man eine höhere Leistung erzielen, da rechenintensive Prozesse auf verschiedene Rechner verteilt werden oder auf einem anderen Rechner ablaufen können. Geräte, zum Beispiel Drucker, können von mehreren Rechnern gemeinsam benutzt werden. Dabei ist ein wichtiges Ziel von verteilten Systemen, dass die Anwenderinnen und Anwender so leicht wie möglich auf entfernte Ressourcen zugreifen können. Die Ressourcen sind im Falle der Abwesenheitsplanung die personenbezogenen Daten, zum Beispiel, wie viel Urlaub zu Verfügung steht. Die Anwenderinnen und Anwender sollen dabei nicht merken, dass die Ressourcen nicht von ihrem Rechner stammen, sondern von einem anderen aufgerufen werden. Diese Eigenschaft bezeichnet man auch als Transparenz.

Es gibt verschiedene Arten von Transparenz [Tanenbaum und van Steen, 2003, S. 21]:

- **Zugriffstransparenz:**

Bei der Zugriffstransparenz sollen Unterschiede in der Datendarstellung der Benutzerin oder dem Benutzer verborgen bleiben. So können die Rechner, die ein verteiltes System bilden, auf unterschiedlichen Betriebssystemen laufen.

- **Positionstransparenz:** Den Anwenderinnen und Anwender soll verborgen bleiben, dass die Daten sich eigentlich auf einem anderen Rechner befinden.

- **Migrationstransparenz:** Daten können verschoben werden, gleichgültig, wie der Zugriff darauf erfolgt.

- **Relokationstransparenz:** Daten können auch verschoben werden, während der Zugriff darauf erfolgt.

- **Nebenläufigkeitstransparenz**

- **Replikationstransparenz:** Die Existenz von mehreren Kopien einer Ressource wird vor den Benutzerinnen und Benutzern verborgen.

- **Persistenztransparenz:** Es wird verborgen, wenn zwischen dem primären und sekundären Speicher gewechselt wird.

- **Fehlertransparenz:** Die Anwenderin und der Anwender erkennen nicht, wenn ein Teil des verteilten Systems nicht mehr korrekt funktioniert.

Ein verteiltes System kann aber auch Nachteile besitzen. Der Aufbau ist komplexer. Es wird mehr Angriffsfläche für Eindringlinge geboten, was ein erhöhtes Sicherheitsrisiko bedeutet. Die Zuverlässigkeit ist vom Netzwerk abhängig und die Geschwindigkeit hängt von der Bandbreite ab. Da die Rechner untereinander kommunizieren, wird ein komplexeres Betriebssystem benötigt. Wenn ein Teil des verteilten Systems nicht mehr funktioniert, kann die Benutzerin oder der Benutzer den Fehler nicht so einfach beheben, da der Rechner auf einem ganz anderen Standort sein kann.

Es gibt mehrere Möglichkeiten, um Komponenten in einem verteilten System miteinander zu verbinden.

2.9.1 Entfernter Prozeduraufruf

Bei einem entfernten Prozeduraufruf (Remote Procedure Call, RPC) ist es einem Programm möglich, Prozeduren aufzurufen, die sich auf anderen Rechnern befinden [Tanenbaum und van Steen, 2003, S. 90ff]. Ruft ein Programm auf Rechner A eine Prozedur auf Rechner B auf, wird der Prozess auf A unterbrochen und die aufgerufene Prozedur findet auf Rechner B statt. Der Ablauf soll weitgehend transparent ablaufen, die Anwenderinnen oder Anwender sollen nicht bemerken, dass die Prozedur auf einer anderen Maschine abläuft.

Synchroner RPC [Tanenbaum und van Steen, 2003, S. 93]

Sollen zum Beispiel Daten aus einer Datei gelesen werden, dann verpackt der Client-Stub Parameter diese in eine Nachricht. Dieses Verpacken wird auch Marshalling genannt. Dabei werden zusätzlich zu den Angaben der Datei auch die Nummer und der Name der Prozedur in die Nachricht verpackt, damit der Server weiß, welche Prozedur er genau aufrufen muss. Danach fordert er, dass diese Nachricht an den Server gesendet und blockiert wird bis er eine Antwort erhält. Das Betriebssystem des Clients sendet die Nachricht an das Betriebssystem des Servers.

Empfängt das Betriebssystem des Servers eine Nachricht, gibt es sie an den Server-Stub weiter. Er packt die Nachricht aus und ruft das Gewünschte auf. Der Server führt die Prozedur aus und gibt das Ergebnis aus. Der Server-Stub verpackt es und gibt es zum Senden frei. Das Betriebssystem des Server sendet die Nachricht an des Betriebssystem des Clients. Der Client-Stub wiederum packt die Nachricht aus. Der Ablauf eines synchronen Remote Procedure Call kann man in Abbildung 2.4 sehen.

Asynchroner RPC [Tanenbaum und van Steen, 2003, S. 100]

Bei dem synchronen, entfernten Prozeduraufruf, wie in Abbildung 2.5 dargestellt, wartet der Client bis er eine Antwort erhält und blockiert so lange. Es wäre sinnvoll, wenn er in dieser Zeit statt zu warten, weiterarbeiten könnte. Bei asynchroner RPC sendet der Client eine Nachricht an den Server. Der Server schickt eine Antwort, dass er die Nachricht bekommen hat,

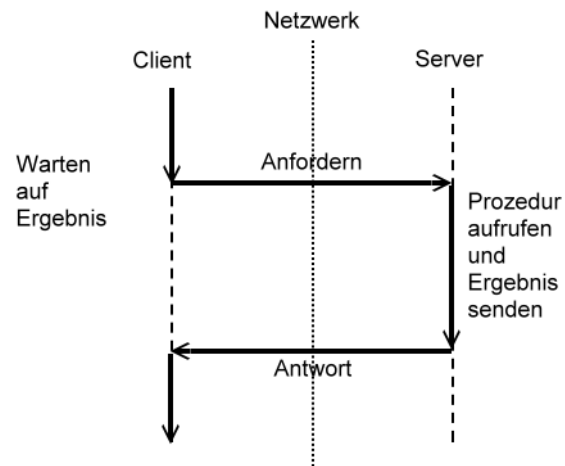


Abbildung 2.4: Synchroner Remote Procedure Call [Tanenbaum und van Steen, 2003, S. 101]

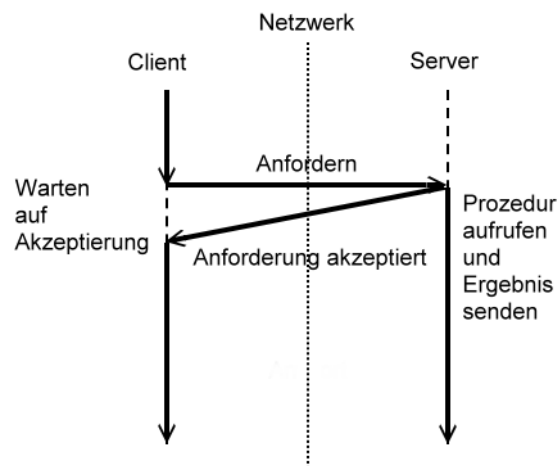


Abbildung 2.5: Asynchroner Remote Procedure Call [Tanenbaum und van Steen, 2003, S. 101]

an dem Client zurück. Daraufhin arbeitet der Client weiter. Es gibt auch Varianten asynchroner, entfernter Prozeduraufrufe, bei denen der Client direkt nach dem Absenden der Nachricht weiterarbeitet und nicht darauf wartet, ob der Server die Nachricht auch wirklich empfangen hat [Tanenbaum und van Steen, 2003, S. 101]. Diese Art der asynchronen RPC wird auch als Einweg-RPC bezeichnet.

2.9.2 Verteiltes System mit SAP

SAP-Systeme können untereinander verbunden werden, es besteht aber auch die Möglichkeit SAP mit Nicht-SAP-Systemen zu verbinden. Die letzte Variante ist für diese Arbeit interessant, da die Abwesenheitsplanungsapplikation ein Nicht-SAP-System darstellt, das eine Verbindung zu einem SAP-System benötigt.

Remote Function Call

Remote Function Call (RFC) ist die Standardschnittstelle, um ein Nicht-SAP-System mit einem SAP-System, aber auch SAP-Systeme untereinander zu verbinden [SAP, 2011e]. Dabei gibt es zwei verschiedene Arten:

- Synchroner RFC: Der Synchroner RFC (sRFC) war die erste Version. Er basiert auf synchroner Kommunikation, das heißt, dass das System während des Aufrufs erreichbar sein muss [SAP, 2011f]. Der Empfänger kann also die Nachricht sofort empfangen und wenn erwünscht auch weiterverarbeiten. Da die beiden benötigten Systeme gleichzeitig ansprechbar sein müssen, kann es zu Problemen kommen, wenn diese Voraussetzung einmal nicht gegeben ist. Der Prozessablauf kann dann massiv gestört sein.

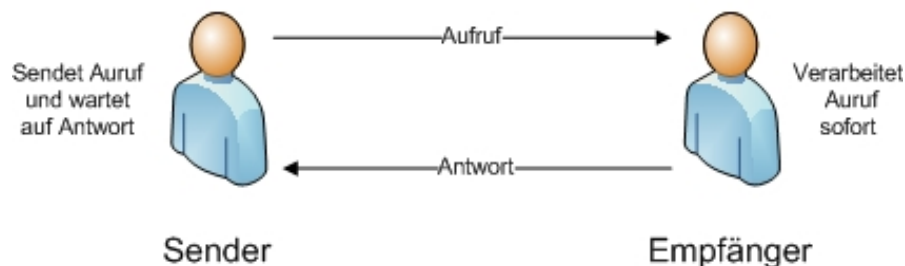


Abbildung 2.6: Synchroner Kommunikation

- Transaktionaler RFC: Der Transaktionale RFC (tRFC) wurde ursprünglich asynchroner RFC genannt [SAP, 2011n]. Er basiert auf asynchroner Kommunikation. Im Unterschied zur synchronen Kommunikation kann der Empfänger aktiv sein, muss aber nicht. Ist der Empfänger nicht bereit, steht der Aufruf weiter in der lokalen Warteschlange und wird in regelmäßigen zeitlichen Abständen gesendet. Wird innerhalb eines bestimmten Zeitraums die Nachricht nicht entgegengenommen, dann wird der Aufruf als Hintergrundjob eingeplant. Der Vorteil ist, dass der Empfänger nicht bereit sein muss, zum Beispiel kann

er gewartet werden, ohne dass der Betrieb gestört wird und der Client kann parallel weiterarbeiten. Der Nachteil ist natürlich, dass die Rückantwort nicht sofort zurückkommt, sondern dass das erst viel später passieren kann.

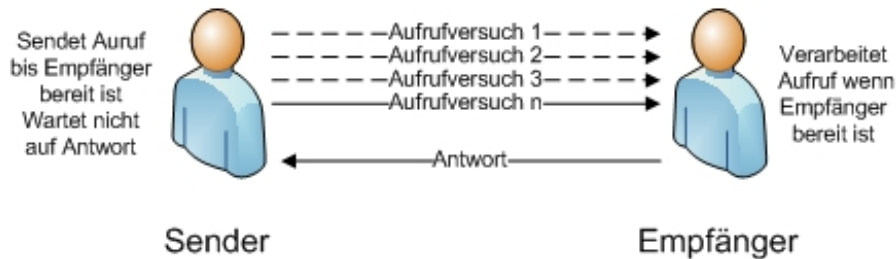


Abbildung 2.7: Asynchrone Kommunikation

ALE/EDI

Application Link Enabling (ALE) und Electronic Data Interchanging (EDI) dienen zum Austausch von Geschäftsdaten zwischen unterschiedlichen Systemen [SAP, 2011a]. Für den Datenaustausch wird eine IDoc-Schnittstelle benötigt.

- Durch Application Link Enabling (ALE) wird die Integration von Geschäftsprozessen von mehreren SAP-Systemen oder Nicht-SAP-Systemen ermöglicht. Es wird verwendet, wenn man von einem, mit einem oder mehreren anderen Systemen kommunizieren will. Diese Systeme sind meist interne Systeme. „Die IDoc-Schnittstelle besteht aus der Definition einer Datenstruktur und einer Verarbeitungslogik für diese Datenstruktur“ [SAP, 2011g]. Das IDoc-Format wird zur Datenübermittlung verwendet und die Übertragung erfolgt mittels tRFC.
- Sollen Geschäftsbelege mit einem anderen System (Nicht-SAP-System) ausgetauscht werden, wird Electronic Data Interchanging (EDI) verwendet. Dabei werden die Nachrichten in IDoc-Format an ein EDI-Subsystem gesendet und dort in ein Format umgewandelt, damit diese auch von Nicht-SAP-Systemen verstanden werden.

IDoc-Schnittstelle

Mit Intermediate Document (IDoc) ist die Verbindung von unterschiedlichen Anwendungssystemen über eine nachrichtenbasierte Schnittstelle möglich und ist ein Standard SAP-Dokumentenformat [SAP, 2011c].

Laut SAP-Bibliothek [SAP, 2010a] werden dabei drei Ziele verfolgt:

- Geschäftsbelege sollen automatisch verbucht und strukturiert ausgetauscht werden.
- Die komplexen Strukturen unterschiedlicher Anwendungssysteme auf eine möglichst einfache Struktur zu reduzieren.

- Vor der Verbuchung der Daten ausführliche Fehlerbehandlung.

Business Application Programming Interface

Mit Business Application Programming Interface (BAPI), die spezielle Methoden sind, kann man bestimmte Geschäftsaufgaben durchführen [SAP, 2010b]. Sie sind im Function Builder der ABAP Workbench als RFC-fähige Funktionsbausteine abgelegt.

Kommunikation zwischen ABAP und anderen Technologien

Da für die spätere Programmierung C# gewählt wurde und diese mit SAP-ERP-Anwendungen kommunizieren muss, werden hier verschiedene Möglichkeitsarten vorgestellt, wobei die ABAP-Programmiersprache verwendet wird, um SAP-ERP-Anwendungen zu schreiben [Kogent Learning Solutions, 2010, S. 15].

- SAP Business Connector

Mit SAP BC (SAP Business Connector), einer eigenständigen vom SAP-System unabhängige Softwarekomponente, können Geschäftsprozesse über das Internet ausgeweitet und Nicht-SAP Produkte integriert werden [SAP, 2011i].

Er umfasst synchrone, asynchrone und bidirektionale Kommunikation mit einem SAP Server. SAP BC besteht aus einem RFC Server und einem Client. Das RFC-Format wird in HTML oder XML umgewandelt, das auch von einem Nicht-SAP-System gelesen werden kann.

- SAP Java Connector

Die Middlewarekomponente SAP Java Connector (SAP JCo) ermöglicht SAP-fähige Anwendungen und Komponenten in Java zu entwickeln, sowohl für Server-Anwendungen als auch für Desktop-Anwendungen [SAP, 2011j].

- SAP Java Resource Adapter

Mit Hilfe von SAP Java Resource Adapter (SAP JRA) können SAP-Systeme mit J2EE Anwendungsservern integriert werden [SAP, 2011k].

- SAP .NET Connector

Der SAP .NET Connector ermöglicht die Kommunikation zwischen SAP-Systemen und Microsoft.Net-Plattformen [SAP, 2011l]. Er unterstützt Web Services und RFCs und ermöglicht es, verschiedene Applikationen zu erstellen, zum Beispiel Web-Formular oder Konsolanwendungen, in Microsoft Visual Studio.Net.

- Internet Communication Framework

Internet Communication Framework (ICF) ermöglicht das Kommunizieren von Internet Standardprotokollen mit einem SAP-System [SAP, 2011d]. Es werden dabei keine zusätzlichen Programmbibliotheken von SAP benötigt, die Systemplattform muss lediglich Internet-tauglich sein.

- SOAP Framework

Durch ein SOAP Framework wird die Erzeugung und Verwendung von Web Services, die auf einem SOAP-Protokoll basieren, ermöglicht [SAP, 2011m].

- Web Service Technologie

Heutzutage müssen nicht mehr alle Funktionen einer Software mit der selben Technologie auf dem selben Rechner vorhanden sein. Mit Hilfe von Web Services kann man all diese Funktionen in einem System integrieren. Web Services werden netzwerkweit publiziert, gesucht, aufgerufen und sind dabei ausführbare, eigenständige Entitäten [SAP, 2011h].

Drei Kernstandards für Web Service sind WSDL, UDDI und SOAP [MSW, 2011].

- WSDL: Web Services Description Language (WSDL) ist eine Beschreibungssprache für technische Schnittstellen, die das Format der Nachrichtenströme (Anforderung und Antwort) bezeichnet, um Funktionen von anderen Programmen aufzurufen. [MSW, 2011].
- UDDI: UDDI steht für Universal Description, Discovery und Integration, wird als Web Service bereitgestellt und ist ein Industrievorschlag für verteilte webbasierende „Dienstekataloge“ [MSW, 2011]. UDDI stellt Anwendungsschnittstellen für verschiedene Web Services-Anbieter zur Verfügung und basiert auf XML und SOAP.
- SOAP: In dezentraler, verteilter Umgebung ist SOAP ein Protokoll für Datenaustausch, das auf XML basiert [SOA, 2011]. Es besteht aus drei Teilen: Eine Verpackung, die ein Framework für die Beschreibung der Nachricht und des Ablaufs definiert, eine Reihe von Regeln zur Verschlüsselung und einen Grundsatz für die Darstellung von Remote Procedure Calls und Antworten.

Kapitel 3

Praktische Realisierung

Bei der praktischen Realisierung wurde auf die aktuelle Situation in einem Unternehmen, das untersucht wurde, näher eingegangen. Die Frage nach alternativen Programmen wurde geklärt. Nach dem theoretischen Teil folgt die Entwicklung der eigentlichen Applikation. Zuerst wurden die Use Cases und das Layout entworfen. Die anschließende Implementierung sorgte für einige Probleme, insbesondere im Zusammenhang mit SAP. Nach Abschluss des Programmierens wurde die Applikation an Benutzerinnen und Benutzern getestet und Verbesserungen ausgearbeitet.

3.1 Aktuelle Situation

Das als Beispiel behandelte Unternehmen hat sich aus Kosten- und Hardwaregründen gegen ein SAP Enterprise Portal und stattdessen für den UrlaubsManager der Firma EDV-Support Hartlaub GmbH entschieden [Add, 2010]. In Interviews wurden mehrere Personen zur aktuellen Situation befragt und hier die Ergebnisse erläutert.

Im Rahmen dieser Arbeiten wurden sechs Mitarbeiterinnen und Mitarbeiter, davon zwei Vorgesetzte und eine Administratorin bzw. ein Administrator zur aktuellen Situation befragt. Bei diesen Interviews wurde gefragt, was sie mit der Applikation machen wollen. Da sowohl das Unternehmen als auch die Personen anonym bleiben wollen, wurden die Namen anonymisiert. Bei den Screenshots handelt es sich lediglich um Beispieldaten der Firma Hartlaub GmbH und nicht um reale Personen.

- **Abwesenheit beantragen**

Mitarbeiterinnen und Mitarbeiter wollen Abwesenheit beantragen. Dazu wollen sie einen gut dargestellten Kalender, bei dem sie die Abwesenheiten ihrer Kolleginnen und Kollegen klar erkennen können.

- **Kalenderansicht**

Die Kalenderansicht soll klar verständlich sein und die verschiedenen Abwesenheiten mit unterschiedlich Farben gekennzeichnet werden. Die Mitarbeiterinnen und Mitarbeiter bevorzugen mehrere Ansichten des Kalenders. Sie hätten gerne die Möglichkeit, sich in einer Monats- oder Jahresansicht einen Überblick zu verschaffen. Für die genauere Bestimmung der Abwesenheiten bevorzugen sie eine Möglichkeit des Zooms oder eine Detailansicht einer Woche oder/und eines Tages.

- **Abwesenheit genehmigen**

Die oder der Vorgesetzte möchte über einen Abwesenheitsantrag per E-Mail informiert werden und darin über den Grund und den Zeitraum (Datum) informiert werden. Zusätzlich ist einen Link zur Kalenderübersicht erwünscht, damit sie oder er sieht, inwieweit die Abwesenheit der Mitarbeiterinnen oder Mitarbeiter mit deren Kolleginnen und Kollegen zu vereinbaren ist.

- **Mitarbeiterliste**

Die Anwenderinnen und Anwender wünschen sich eine Liste ihrer Kolleginnen und Kollegen, bzw. ihrer Mitarbeiterinnen und Mitarbeiter. Im aktuellen Zustand kann diese nicht verändert werden. Diese Situation stört insbesondere, wenn Personen längere Zeit nicht anwesend sind, zum Beispiel weil sie in Karenz oder auf Kur sind, und sie dennoch aufgelistet sind. Wobei da weniger die Tatsache stört, dass der Name der Person in der Liste vorkommt, sondern mehr, dass dazu auch die dazugehörige Abwesenheit im Kalender dargestellt wird. Es soll daher möglich sein, einzelne Mitarbeiterinnen und Mitarbeiter auszublenden.

- **Administrator**

Daten der Mitarbeiterinnen und Mitarbeiter sollen leicht zu editieren sein. Im Falle eines Abteilungswechsels soll es ohne Probleme möglich sein, einer Person einer anderen Vorgesetzten oder eines anderen Vorgesetzten und anderen Kolleginnen und Kollegen zuzuordnen. In der aktuellen Situation war nur eine Löschung und ein neues Einfügen der Person möglich, wobei alle Daten verloren gingen.

Es zeigte sich, dass die Personen im Beispielunternehmen eine heterogene Gruppe bilden. Die Arten von Gruppen werden näher in Abschnitt 2.1.1 beschrieben. Die Personen, die der Gruppe angehören, sind aufgrund unterschiedlicher Eigenschaften durchmischt. Die Personen sind nicht nur im Bereich des Geschlechts, Alters, der Ausbildung und Kompetenzen verschieden, sie haben auch unterschiedliche Rollen in der Gruppe. Mit Gruppe ist dabei der Personenkreis gemeint, der so weit miteinander zusammenarbeitet, dass deren Abwesenheiten aufeinander abgestimmt werden müssen oder wenn sie nicht direkt zusammenarbeiten, dass sie einander vertreten können. Im Unternehmen handelt es sich um eine Mischung aus statischen und dynamischen Gruppen. Gruppen können über Jahre aus den selben Personen bestehen. Es kann aber auch Bereiche geben, insbesondere größere, in denen die Mitarbeiterinnen und Mitarbeiter öfter wechseln.

Bei der Befragung wurde versucht herauszufinden, um welche Art von Benutzerinnen und Benutzer es sich im Unternehmen handelt. Welche Arten es prinzipiell geben kann wird in Abschnitt 2.2 erläutert. Im Unternehmen wird man mit mehreren Personen dieser Gruppen zu tun haben, was das Design schwieriger macht, als wenn nur eine Gruppe betroffen ist. Aus den Interviews ist hervorgegangen, dass vor allem „Außenseiter“ die Applikation verwenden. Urlaub, Kur etc. zu beantragen, sind keine Dinge, die täglich, sondern sporadisch zu erledigen sind. Ausnahme bilden die Vorgesetzten, insbesondere die mit mehreren Mitarbeiterinnen und Mitarbeitern, die täglich oder zumindest alle paar Tage überprüfen müssen, ob es Anträge gegeben hat.

Zudem wurde eruiert, wie im Moment die Abwesenheitsplanung abgewickelt wird. Zur Zeit wird der UrlaubsManager im Unternehmen benutzt, um Abwesenheiten zu beantragen und zu genehmigen. Abwesenheiten sind in dem Unternehmen Urlaub, Zeitausgleich, Schulungen, Bildungskarenz, Mutterschutz, Karenz, Pflegeurlaub, persönliche Verhinderungsgründe (Todesfall, Hochzeit, ...), Kur und Krankheiten, falls diese schon vorzeitig bekannt sind, wie zum Beispiel geplante Krankenhausaufenthalte.

Das Ergebnis der Befragung ist, dass die Anwenderinnen und Anwender mit dem Programm prinzipiell zufrieden sind. Ein Hauptproblem für die Personen war es, dass der UrlaubsManager keine Verbindung zu einem SAP-System hat. Daher werden die Daten von ihm händisch in SAP übertragen. Diese Variante ist nicht nur zeitaufwändig, es kommt auch immer wieder zu Fehlern, die dann in mühsamer Kleinarbeit gesucht und ausgebessert werden müssen. Das ist auch der Grund dafür, dass der Resturlaub, der im UrlaubsManager eingetragen ist, nicht immer mit dem im SAP-System oder dem tatsächlichen Resturlaub übereinstimmt. Die Anzahl der Urlaubstage wird einmal im UrlaubsManager angegeben und nur mit dem Urlaub, der hier eingetragen wurde, abgeglichen. Abwesenheiten, die nur im SAP-System eingegeben werden, werden dabei nicht berücksichtigt.

Den befragten Personen fällt insbesondere die graphische Benutzeroberfläche positiv auf. Eine Einschulung, Erklärungen oder Hilfe wurde von dem Mitarbeiterinnen und Mitarbeiter noch nicht benötigt, die Applikation wurde als selbsterklärend beschrieben.

In Abbildungen 3.1 kann man die Abwesenheitsverwaltung des UrlaubsMangers sehen. Im oberen Bereich kann man zwischen drei Ansichten wechseln, „Aktuelle Anträge“, „Mitarbeiter“ und „Kalender“. Beim Kalender steht der Benutzerin oder dem Benutzer eine Ansicht des aktuellen Monats und der zukünftigen Monate zur Verfügung. Darunter findet man eine Detailansicht der Wochen und die eingetragenen Abwesenheiten.

Bestimmte Abwesenheiten können mit verschiedenen Farben gekennzeichnet werden. Einige Personen haben die, für ihren Geschmack, zu kräftige Farbauswahl kritisiert. Das kräftige Gelb und Rot sei ihnen zu schreiend. Da die Liste der Mitarbeiterinnen und Mitarbeiter sehr lang ist und verschiedene Abwesenheiten eingetragen werden, sei es ihnen zu farbenfroh. Die Farben können allerdings vom Administrator angegeben werden, wobei diese dann für alle Benutzerinnen und Benutzer gleich sind. Individuell ist die Farbuweisung leider nicht möglich. Auch die graue Schriftfarbe wurde kritisiert, die auf dem etwas anderen grautonfarbigen Hintergrund schwer lesbar sei. Dieses Problem ist ebenfalls durch den Administrator änderbar, da auch die Schriftfarbe frei wählbar ist.

Ein weiterer Kritikpunkt ist, dass man die Firma/Abteilung rechts oben anhaken muss, wie

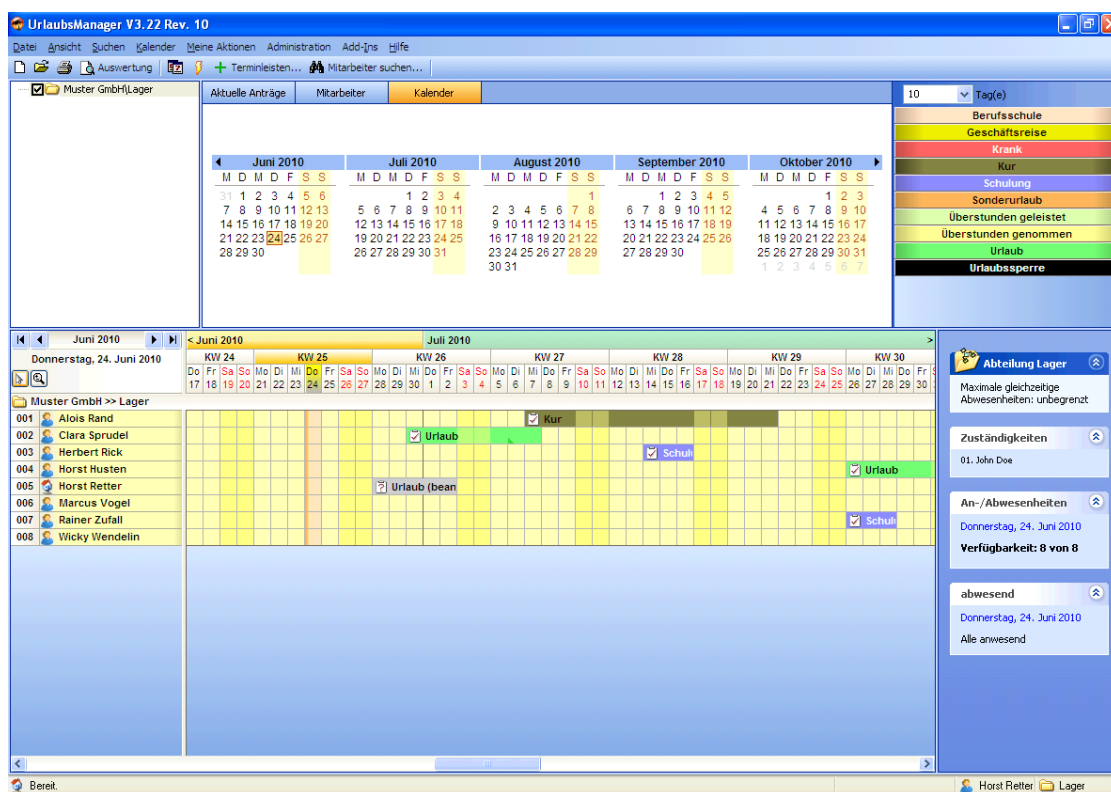


Abbildung 3.1: UrlaubsManager - Abwesenheitsverwaltung

in Abbildung 3.2 dargestellt, damit man selbst und die Arbeitskolleginnen und Kollegen aufgelistet sind. Im Laufe der Benutzung ist bei allen sechs Personen das Problem aufgetreten, dass die Firma/Abteilung nicht markiert waren und sie ihre Daten vermisst haben. Das Programm habe zwar eine Hilfestellung geleistet, diese brachte aber keine Hilfe und die Lösung wurde durch Suchen und Probieren gefunden. Da die Benutzerinnen und Benutzer über das Programm Abwesenheiten beantragen wollen und dafür eigentlich immer Einsicht in die Abwesenheiten ihrer Kolleginnen und Kollegen wollen, würden sie sich wünschen, dass sie immer automatisch Zugriff auf ihre Kolleginnen und Kollegen haben und diese Ansicht von vornherein nicht ausschaltbar ist. Damit würde auch gar nicht die Möglichkeit bestehen, dass dieser Fehler auftritt.

Vorgesetzte kritisierten zudem das hohe E-Mailaufkommen. Wenn eine Mitarbeiterin oder ein Mitarbeiter zum Beispiel drei Abwesenheiten direkt hintereinander beantragt, werden ihm drei E-Mail-Benachrichtigungen zugesendet. In Abbildung 3.3 sieht man den Posteingang des UrlaubsManagers. Herbert Rick hat hintereinander drei Abwesenheiten beantragt und für jede Beantragung ist eine Nachricht eingegangen. Die Vorgesetzten würden sich wünschen, dass es ihren Mitarbeiterinnen und Mitarbeitern ermöglicht wird, mehrere Abwesenheiten zu beantragen und dass diese in eine Nachricht zusammengefasst werden.

Darüber hinaus seien darin enthaltene Informationen wie in Abbildung 3.4 sehr dürftig. Sie würden nur erfahren, um welche Person es sich handelt, welche Art der Abwesenheit erwünscht

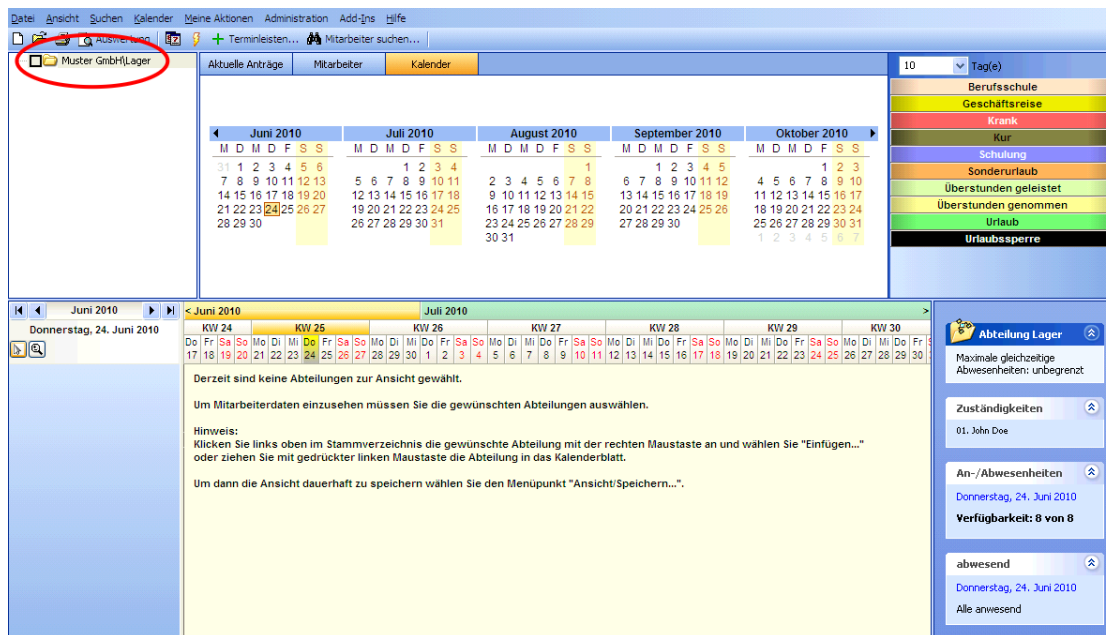


Abbildung 3.2: UrlaubsManager -Keine Firma/Abteilung ausgewählt

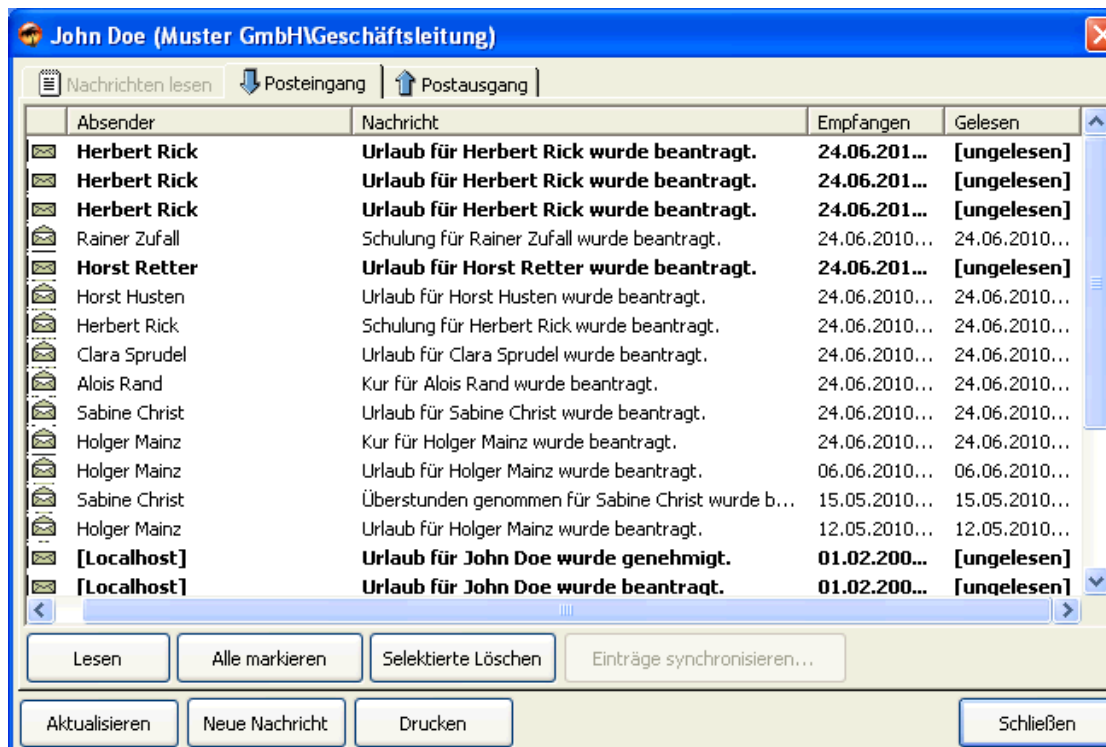


Abbildung 3.3: UrlaubsManager - Posteingang

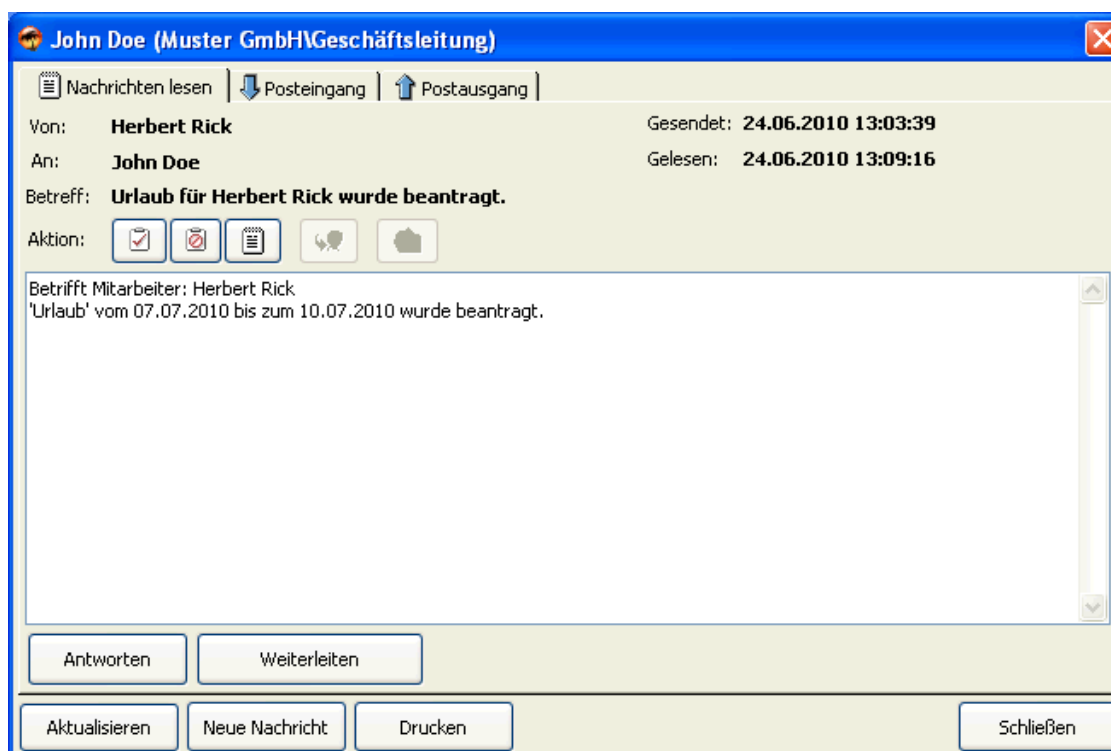


Abbildung 3.4: UrlaubsManger - Nachricht

ist und das Datum des Zeitraumes. Um Näheres zum Termin oder der Person zu erfahren, muss erst das Programm geöffnet und die jeweilige Person gesucht werden. Vorgesetzte würden sich wünschen, dass zumindest der Wochentag als zusätzliche Angabe vorhanden wäre oder noch besser ein direkter Link zu den Daten der Person in der Abwesenheitsverwaltung.

Einige Kritikpunkte könnte die Administratorin oder der Administrator beseitigen, diese oder dieser bezeichnet das Programm allerdings als umständlich. Es würde zu viele Einstellmöglichkeiten besitzen, die sie oder er als zu komplex betrachtet. Sie oder er hätte zudem ein Problem mit Personen, die die Abteilung wechseln, da es laut ihr oder ihm keine Möglichkeit dazu im UrlaubsManager geben würde. Man müsste die Person löschen und danach wieder neu eintragen. Alle bis dahin gesammelten Daten würden dabei verloren gehen.

Ein weiterer Kritikpunkt ist, dass es kompliziert wäre, zwei Kalender aufeinander abzustimmen. Die Mitarbeiterinnen und Mitarbeiter würden Microsoft Outlook verwenden, um ihre Termine zu koordinieren. Neben beruflichen Terminen, wie zum Beispiel Besprechungen, Kundenterminen, Seminaren usw., tragen die Mitarbeiter auch private Termine ein, wie zum Beispiel Geburtstage und Konzerttermine. All diese Dinge würden sie gerne mit der Abwesenheitsplanung verbinden und dabei aber gerne private Termine von der Öffentlichkeit ausschließen.

Ein Problem für die Benutzerinnen und Benutzer ist zudem, dass im UrlaubsManger nur Abwesenheiten von der Firma eingetragen werden. Für interne Termine verwendet jede Person für sich den Kalender von Microsoft Outlook, bei dem Kolleginnen und Kollegen keine Zugriff

haben. Dadurch kann es zu Problemen kommen, da eine Person Abwesenheit genehmigt bekommt, die vertretenen Kolleginnen und Kollegen auch prinzipiell im Unternehmen sind, aber da sie zum Beispiel eine Besprechung haben, die Arbeit nicht übernehmen können. Diese Problematik wird im UrlaubsManager nicht behandelt.

The screenshot shows a software window titled "Abwesenheit(en) erstellen / bearbeiten...". It contains several input fields and buttons. The "Wer" field is filled with "Clara Sprudel;". The "Stellvertretung" field is empty. The "Was" field is a dropdown menu set to "Urlaub". The "Status" field is a dropdown menu set to "Beantragt". The "Beginn" field is a date picker set to "24.07.2010" and a time picker set to "Ganztags". The "Ende" field is a date picker set to "24.07.2010". There is an "Info:" label above a large empty text area. At the bottom, there is a "Vorschau:" section showing "Urlaub (beantragt)" and three buttons: "Prüfen", "Senden", and "Abbrechen". On the right side, there is a "Resturlaub" section showing the number "18" and a "Details..." button. Above the "Resturlaub" section, there is a "Serie..." button and a small image of a hand holding a document.

Abbildung 3.5: UrlaubsManager - Abwesenheitsbeantragung

Von den Benutzerinnen und Benutzern wurde hervorgehoben, dass der Antrag auf Abwesenheit unkompliziert sei. Mit einem Klick wäre das Formular, wie in Abbildung 3.5 ersichtlich, geöffnet und selbsterklärend. Nur die Terminauswahl wurde als dürftig empfunden. Man hat zwar einen Kalender, wie man ihn in Abbildung 3.6 sehen kann, zur Auswahl, bei dem man aber keinerlei Referenzen zu den Terminen der Arbeitskolleginnen und Kollegen oder zu den eigenen Terminen besitzt. Falls man sich nicht genau das Datum gemerkt hat oder noch bei diesem Schritt Überlegungen über eine Terminänderung vornimmt, muss man ihn abbrechen und wieder einen Schritt zurück zur Abwesenheitsverwaltung gehen.

Aus den Interviews wurde versucht, die Anforderungen, wie in Abschnitt 2.2 beschrieben, zu ermitteln. Es ergab sich, dass ein einfaches Programm erwünscht wird, das einfach zu bedienen und erlernbar ist. Die Hauptanforderung ist, dass die Applikation eine Verbindung mit SAP besitzen soll, in dem die Abwesenheit direkt geschrieben wird, und aus dem die noch zur Verfügung stehenden Urlaubstage angezeigt werden. Es soll ein optisch ansprechender Kalender dargestellt werden. Der Vorgang der Beantragung soll einfach zu handhaben sein und die Kalenderansicht und die Einträge der Kolleginnen und Kollegen beinhalten. Falls eine Person mehrere Abwesenheiten hintereinander beantragt, wäre es wünschenswert, wenn nur eine einzi-



Abbildung 3.6: UrlaubsMangager - Kalender

ge Nachricht verschickt wird.

Auf dem Markt existieren neben dem UrlaubsManager auch Konkurrenzprodukte. Diese sollen im nächsten Kapitel näher behandelt werden.

3.2 Existierende Programme

Vor der praktischen Realisierung des Prototyps wurde eruiert, welche Software es in diesem Bereich gibt, und welchen Funktionsumfang diese umfassen. Die Recherche zeigte, dass alle Programm über zusätzliche Anforderungen, ob eine eigene Datenbank oder gleich ein ganzes SAP-System, verfügen.

Für die Realisierung der Abwesenheitsplanung existieren auf dem Markt einige Produkte. Manche von ihnen besitzen auch eine SAP-Schnittstelle. In diesem Kapitel wird auf einige Beispiele aus dem deutschsprachigen Markt näher eingegangen. Diese Produkte wurden allerdings nicht auf die praktische Realisierung überprüft.

3.2.1 GeCOSOF T

Ein Programm zur Abwesenheitsplanung ist GeCOSOF T, wie in Abbildung 3.7 gezeigt, von der Firma Gesellschaft für Computertechnik Organisation und Software GmbH, wobei es aus mehreren Teilen besteht [Gec, 2011]. Neben der Abwesenheitsplanung (gecosoft-workflow) verfügt GeCOSOF T über eine Reihe von anderen Funktionen wie Zutrittskontrolle, Betriebsdatenerfassung und Personaleinsatzplanung. Mit gecotime-workflow kann man über ein Web-Terminal Abwesenheitsanträge übermitteln. Vorgesetzte werden automatisch per E-Mail über Anträge informiert. Beantragte und verrechnete Abwesenheiten sind farblich gekennzeichnet. Nachteil des Programms ist, dass Abwesenheitsanträge nur über ganze oder halbe Tage gestellt werden können. GeCOSOF T verfügt über eine eigene Datenbank, die laut der Webseite der Firma periodisch mit SAP HR abgeglichen wird [Gec, 2011].

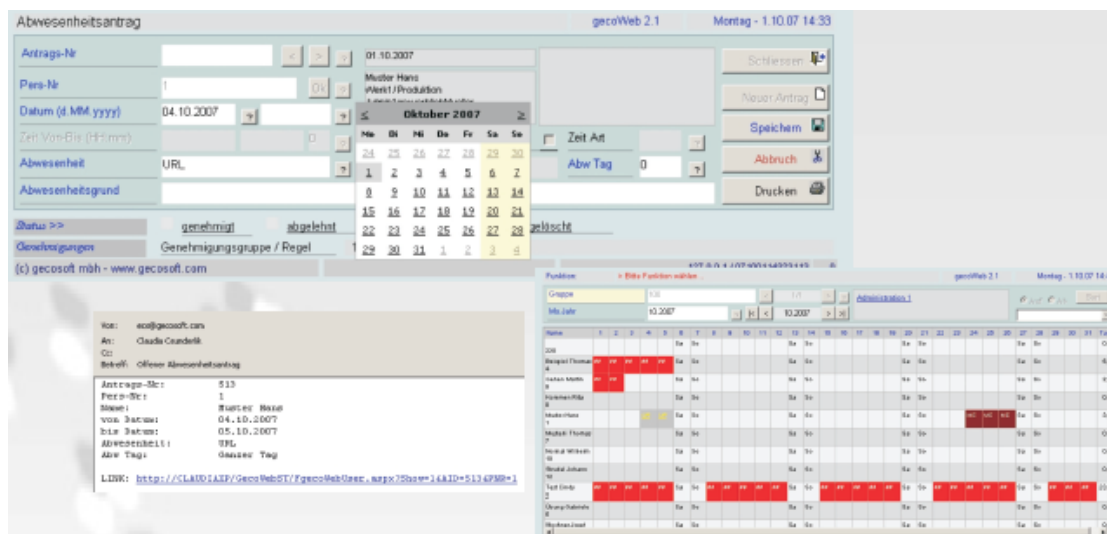


Abbildung 3.7: gecotime - workflow [Gec, 2011]

3.2.2 HR-Starter®ASP

HR-Starter®ASP ist ein Programm zum Mieten von der Firma inside GmbH [ins, 2011]. Es besteht aus mehreren Modulen (zum Beispiel Personalabrechnung, Zeitwirtschaft, Organisationsmanagement, Auswertungen, elektronische Personalakte, Mitarbeiter-Qualifikationen,...), die einzeln genutzt werden können. In einem Portal können Mitarbeiterinnen und Mitarbeiter in ihr aktuelles Urlaubskontingent einsehen und einen Eintrag erstellen. Ein elektronischer Genehmigungs-Workflow wird automatisch an den Vorgesetzten gesendet. Wie in Abbildung 3.8 dargestellt, wird durch einen Hosting Partner ein SAP-System zur Verfügung gestellt, zu dem Kunden sich mittels VPN (Virtual Private Network) verbinden können. Laut Hersteller-Webseite ist eine ALE-Anbindung zu einem beim Kunden bestehenden SAP-System möglich [ins, 2011]. Genaueres zu ALE-Anbindung findet man im Abschnitt 2.9.2. Ein Hindernis für Unternehmen bei dieser Lösung könnte sein, dass sensible Daten über ihr Personal ausgelagert werden.

3.2.3 Softsimple

Mit dem Produkt „Out of Office“ der Firma CAD-CAM-KI Software Solutions Simple können Unternehmen An- und Abwesenheiten managen [Software Solutions Simple, 2010]. Einen Screenshot kann man in Abbildung 3.9 sehen. Das Programm ist sehr umfangreich, so ist eine direkte Kopplung an ein Adressbuch möglich, jede Mitarbeiterin und jeder Mitarbeiter kann über ein eigenes Arbeitszeitmodell verfügen. Sonderregelungen, zum Beispiel für die Nachtstunden und Pausenregelungen, können getroffen werden. Das System kann ebenfalls zur Zeiterfassung und Arbeitszeitkorrektur verwendet werden. Laut Leistungskatalog ist Lotus Notes/Domino 6.5 oder eine höhere Version Systemvoraussetzung. Zur Übergabe der Daten an ein HR-System wie SAP steht ein Export zur Verfügung.

ZUGRIFF ÜBER GESICHERTE NETZWERKVERBINDUNGEN:

VPN - VIRTUAL PRIVATE NETWORK

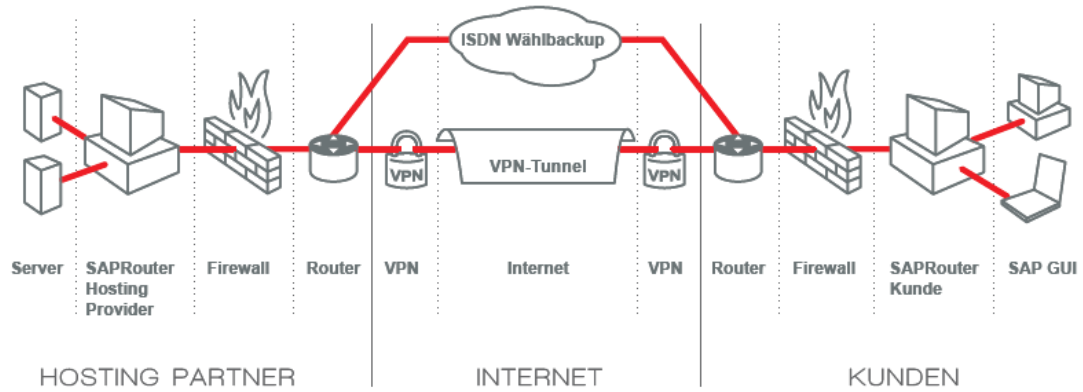


Abbildung 3.8: HR-Starter®ASP - Zugriff über gesicherte Netzwerkverbindung [ins, 2011]

Über/Unterstunde
Urlaub/Ausgleich
Abwesenheit

Organisation und Vorgaben

Abteilungsname: **GL**

Für mich gilt die Region: **DE-Bayern**

Meine Anträge entscheiden: **Christian Huber**

Meine Urlaubsvertretung **kein Vertreter bestimmt**

Anträge - in Bearbeitung

von - bis	Status	gestellt am
▼ 2008		
▼ Eigene Arbeitszeitkorrekturanträge		
Zeitraum : 2008-01	Nicht beantragt	01/15/2008
▼ Abwesenheitsanträge		
- 01./03/2008 - 01./03/2008	Nicht beantragt	01./03/2008
▼ 2007		
▼ Entscheidung erwartet von Callcenter, Barbara		
12/11/2007 - 12/12/2007 (2 Tage)	Beantragt	12/11/2007
Betreff (Christian Huber, 12/11/2007)		
▼ Eigene Urlaubsanträge		
11/02/2007 - 11/02/2007 (1 Tage)	Beantragt	11/14/2007

Meine aktuellen Daten

Für das Jahr 2007 kann ich noch Urlaub beantragen für **3 Tage**

Mein Gesamturlaubsanspruch für das Jahr 2007 beträgt **27 Tage**

Mein aktueller Zeitkontostand für das Jahr 2007 beträgt **14 Stunden**

Für das Jahr 2008 kann ich noch Urlaub beantragen für **16 Tage**

Mein Gesamturlaubsanspruch für das Jahr 2008 beträgt **30 Tage**

Mein aktueller Zeitkontostand für das Jahr 2008 beträgt **0 Stunden**

Übersicht

Bezeichnung	Von	Bis	Anspruch Urlaub (Tage)	Davon aktuell	Ausgleich aktuell	Aktuell Krank (Tage)
▶ Übersicht Jahr 2006						
			31	0	0	0
▼ Übersicht Jahr 2007						
▼ A-Anspruch Urlaub			6	26	4	4
Mein Kontingent			4,00			
Sonderkontingent (Umzug)			04.09.2007 - 04.09.2007	1,00		
Sonderkontingent (Umzug)			04.09.2007 - 04.09.2007	1,00		
Sonderkontingent (Umzug)			05.09.2007 - 05.09.2007	1,00		
Sonderkontingent (Umzug)			19.12.2007 - 19.12.2007	-1,00		
▼ B-Anspruch Ausgleich			0	0	0	0
Arbeitszeitkorrektur 2007-12			12.12.2007 - 12.12.2007			
Sonderkontingent			01.01.2007 - 31.12.2007			
▼ C-Urlaube			0	26	0	0
Urlaub genommen			20.08.2007 - 29.08.2007			
Urlaub genommen			31.08.2007 - 31.08.2007			

Abbildung 3.9: Out of Office [Software Solutions Simple, 2010]

Fazit

Diese Produkte haben alle den Nachteil, dass sie bestimmte Systemanforderungen mitbringen. Sie benötigen alle eine eigene Datenbank, die in regelmäßigen Abständen mit den Daten des SAP-Systems abgeglichen werden muss. Manche benötigen zusätzlich ein Programm bzw. verfügen überhaupt über ein eigenes SAP-System, das dann mit einem bestehenden verbunden wird. Alle diese Dinge machen die Implementierung kompliziert. Daher soll ein Programm entwickelt werden, das als Datenbank direkt die SAP-Datenbank verwendet, damit Redundanz vermieden wird.

Um eine Software zu entwickeln, ist es wichtig deren grundlegenden Funktionen näher zu behandeln und die Hauptaufgaben und Zusammenhänge zu definieren.

3.3 Use Cases und Aktivitätsdiagramme

Um ein Softwareprogramm neu zu entwickeln, müssen die Hauptaufgaben beschrieben werden. Use Case-Diagramme dienen zur Beschreibung solcher Aufgaben [Stephan, 2009, S. 343]. Im Falle der Abwesenheitsplanung gibt es mehrere Hauptaufgaben. Abwesenheiten müssen beantragt werden können und danach änderbar sein. Auch ein Löschen von Abwesenheiten soll ermöglicht werden. Vorgesetzte müssen beantragte Abwesenheiten genehmigen oder ablehnen. Wie diese Hauptaufgaben konkret ablaufen, soll dieses Kapitel zeigen.

3.3.1 Use Cases

Eine Use Case ist ein Anwendungsfall, der ein bestimmtes Verhalten beschreibt, das von einem zu entwickelnden System erwartet wird [Hitz und Kappel, 2003, S. 105]. Mit ihm kann man schriftlich fixierte Geschichten beschreiben [Larman, 2005, S. 98]. Insbesondere bei größeren Projekten ist es schwierig, alle Anwendungsfälle im Vorfeld zu bestimmen. Alle diese Anwendungsfälle zusammen machen die Funktionalität des gesamten Systems aus. Im Anwendungsfalldiagramm (Use Case Diagramm) sind alle Anwendungsfälle aufgeführt. In UML wird ein Anwendungsfall durch eine Ellipse gekennzeichnet. Das Rechteck, das mehrere Anwendungsfälle umschließt, stellt das System dar, in dem diese Anwendungsfälle zugeordnet sind. Die Anwenderinnen und Anwender (Akteurinnen und Akteure) befinden sich außerhalb dieses Rechtecks. Kommuniziert eine Akteurin oder ein Akteur mit einem Anwendungsfall, wird diese Beziehung mit einer ungerichteten Linie dargestellt, dabei kann die Kommunikation in beiden Richtungen stattfinden. Jede Akteurin und jeder Akteur muss mindestens mit einem Anwendungsfall kommunizieren. Anwendungsfälle können auch untereinander in Beziehung stehen. Man unterscheidet drei Beziehungsarten: die include-, die extend-, und die Generalisierungsbeziehung [Hitz und Kappel, 2003, S.106]. Sowohl die include- als auch die extend-Beziehung sind gerichtete Beziehung zwischen zwei Anwendungsfällen. Mit einer include-Beziehung von Anwendungsfall A nach Anwendungsfall B ist gemeint, dass das Verhalten von B in A eingefügt wird. Bei einer extend-Beziehung kann der Anwendungsfall A, der die Basisfunktion beschreibt, auch ohne den Anwendungsfall B ausgeführt werden, der den Anwendungsfall A nur erweitert.

Das System der Abwesenheitsplanung ist durch dieses Anwendungsfalldiagramm in Abbildung 3.10 gekennzeichnet. Diese Fälle decken die Anwendung einer Vorgesetzten oder eines

Vorgesetzten ab. Eine Mitarbeiterin oder ein Mitarbeiter ohne die Stellung einer Vorgesetzten oder eines Vorgesetzten haben weniger Anwendungsfälle. In ihrem Fall steht ihnen die Funktion der Genehmigung der Abwesenheit nicht zur Verfügung.

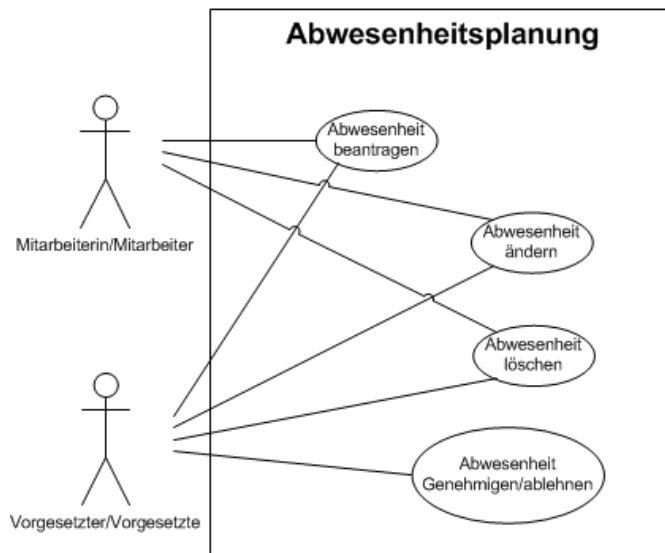


Abbildung 3.10: Anwendungsfalldiagramm der Abwesenheitsplanung

Folgende Anwendungsfälle, die zuvor in Interviews erfragt wurden, müssen das System abdecken:

- Der grundlegende Anwendungsfall ist die Beantragung einer Abwesenheit. Sie wird sowohl von Mitarbeiterinnen und Mitarbeitern als auch von Vorgesetzten durchgeführt. In Abbildung 3.11 wird als Person nur Mitarbeiterin bzw. Mitarbeiter angegeben, da die oder der Vorgesetzte nur zusätzliche Funktionen besitzt und im Falle der Abwesenheitsbeantragung wie eine Mitarbeiterin oder Mitarbeiter handeln. Um Abwesenheit zu beantragen, muss die Person einen Abwesenheitszeitpunkt und eine Abwesenheitsart auswählen. Wenn sie oder er möchte, kann auch eine Bemerkung verfasst werden.
- Aus unterschiedlichen Gründen kann eine Benutzerin oder ein Benutzer eine bereits beantragte und eventuell schon genehmigte Abwesenheit zu einem späteren Zeitpunkt ändern wollen. Dieser Vorgang wird in Abbildung 3.12 gezeigt. Dazu muss eine bereits existierende Abwesenheit ausgewählt werden. Die Benutzerin oder der Benutzer kann die Abwesenheitsart und/oder einen neuen Zeitraum festlegen. Wird weder die Abwesenheitsart noch der Zeitraum neu gewählt, bleibt die Abwesenheit unverändert.
- Es kann vorkommen, dass eine Benutzerin oder ein Benutzer eine bereits beantragte und eventuell schon genehmigte Abwesenheit zu einem späteren Zeitpunkt löschen will. Der Vorgang der Löschung wird in Abbildung 3.13 gezeigt.

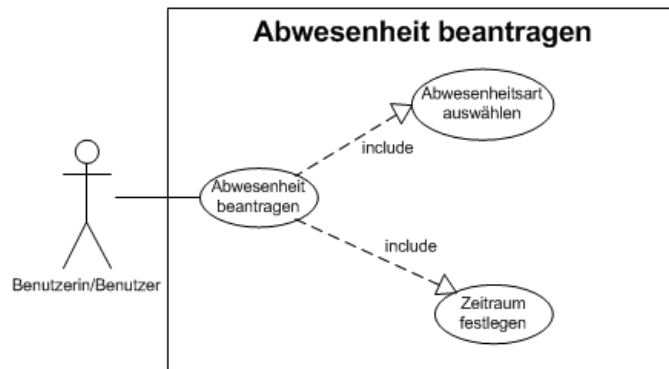


Abbildung 3.11: Use Case der Beantragung einer Abwesenheit

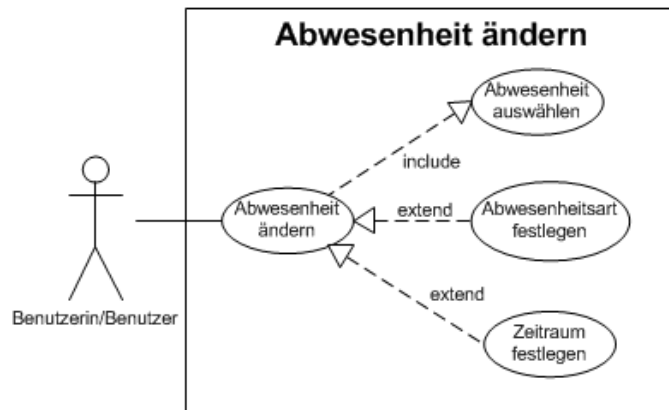


Abbildung 3.12: Use Case für die Änderung einer Abwesenheit

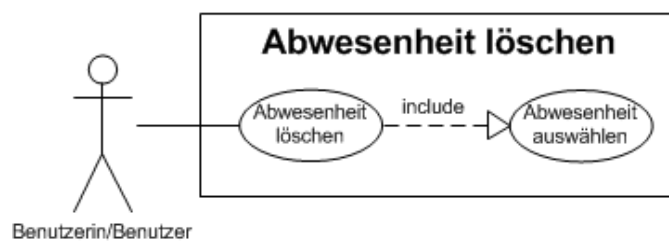


Abbildung 3.13: Use Case für die Löschung einer Abwesenheit

- Eine Vorgesetzte oder ein Vorgesetzter kann neben der Beantragung von Abwesenheiten, diese von seinen Mitarbeiterinnen und Mitarbeitern auch genehmigen bzw. ablehnen. In Abbildung 3.14 wird gezeigt, dass sie oder er dafür erst alle offenen Anträge prüfen muss, diese auf eventuelle Ablehnungsgründe (zum Beispiel Kollisionen) überprüfen und sie danach ablehnen oder genehmigen kann. Das System überprüft selbst, ob es zu eventuellen Kollisionen unter den Mitarbeiterinnen und Mitarbeitern kommt. Zum Beispiel, ob eine bestimmte Personenanzahl zur selben Zeit Abwesenheiten beantragt bzw. genehmigt hat. Allerdings kann es auch zu anderen Ablehnungsgründen kommen, zum Beispiel durch Projektabgabeterminen oder Stoßzeiten, bei denen mehr Arbeitskräfte als gewöhnlich anwesend sein müssen. Diese Gründe kann das System nicht erkennen.

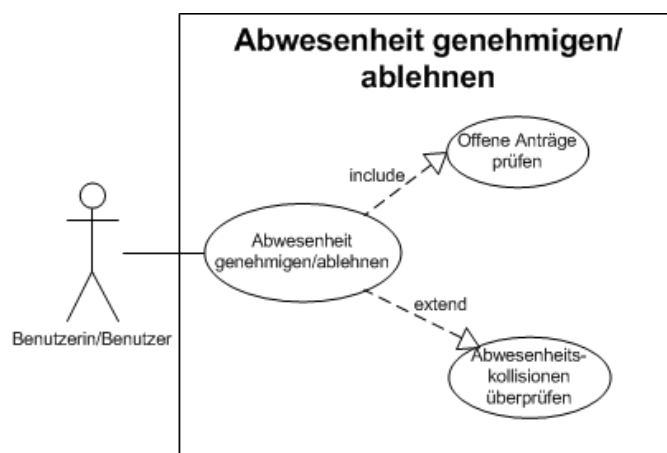


Abbildung 3.14: Use Case für die Genehmigung bzw. Ablehnung einer Abwesenheit

3.3.2 Aktivitätsdiagramme

Ein Aktivitätsdiagramm beschreibt einen Ablauf, wobei spezifiziert werden kann,

- was die einzelnen Schritte des Ablaufs tun,
- in welcher Reihenfolge diese Schritte ausgeführt werden und
- optional, wer für sie verantwortlich ist [Hitz und Kappel, 2003, S. 160].

Eine wichtige Aktivität ist die Beantragung von Abwesenheiten. Das Aktivitätsdiagramm wird in Abbildung 3.15 näher beschrieben. Dafür muss die Benutzerin oder der Benutzer ein Formular öffnen und dieses ausfüllen. Beim Ausfüllen ist insbesondere die Abwesenheitsart, zum Beispiel Urlaub, Zeitausgleich etc. und der Zeitraum, in dem die Abwesenheit stattfinden soll, wichtig. Optional kann noch eine Bemerkung erfasst werden. Das System überprüft das Formular auf richtige Eingabe, zum Beispiel, ob das Datum des Abwesenheitsendes nicht vor dem Startzeitpunkt liegt. Ist das Formular richtig ausgefüllt worden, wird es vom System auf

eventuelle Kollisionen überprüft. Kollisionen kommen vor, wenn zum Beispiel zum selben Zeitpunkt bereits von derselben Benutzerin oder demselben Benutzer eine Abwesenheit beantragt bzw. ihr oder ihm genehmigt wurde. Falls Kollisionen vorhanden sind, wird die Anwenderin oder der Anwender über diese informiert und kann/muss das Formular noch einmal ausfüllen. Kommen keine Kollisionen vor, wird die Abwesenheit als „vorgemerkt“ ins System eingetragen und die oder der zuständige Vorgesetzte wird per E-Mail über den Antrag informiert.

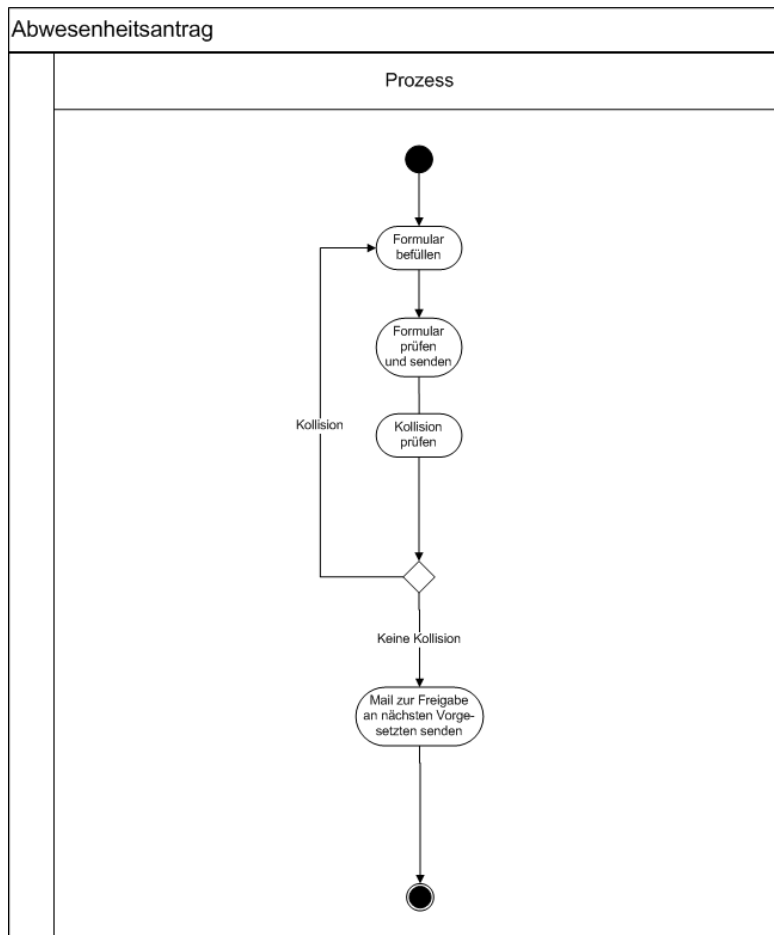


Abbildung 3.15: Abwesenheit beantragen

In Abbildung 3.16 wird der Ablauf der Änderung einer Abwesenheit beschrieben. Im ersten Schritt muss die Abwesenheit, die die Benutzerin oder der Benutzer ändern möchte, ausgewählt werden. Nach dem Öffnen eines Formulars kann man die jeweiligen Daten ändern bzw. neu angeben. Danach ist der Vorgang der Abwesenheitsbeantragung sehr ähnlich. Die Daten werden auf Richtigkeit der Eingabe und anschließend auf Kollisionsfreiheit mit anderen Abwesenheiten geprüft. Falls alles ihrer Richtigkeit entspricht, wird die oder der zuständige Vorgesetzte per E-Mail informiert.

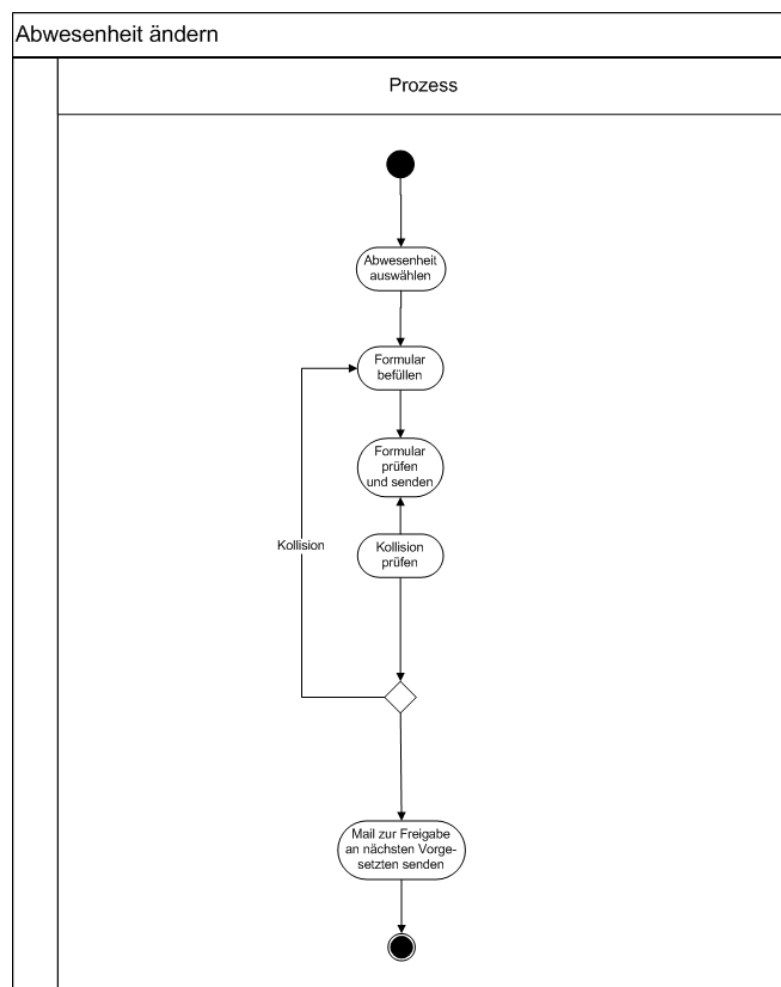


Abbildung 3.16: Abwesenheitsänderung

In Abbildung 3.17 wird der genaue Ablauf der Genehmigung beschrieben. Die oder der Vorgesetzte muss zuerst das System auf offene Anträge überprüfen. Falls eine Abwesenheit von einer Mitarbeiterin oder einem Mitarbeiter beantragt wurde, kann diese von der oder dem Vorgesetzten auf eventuelle Ablehnungsgründe überprüft werden. Falls die Abwesenheit nicht genehmigt werden kann, wird sie abgelehnt und die Person, die diese Abwesenheit beantragt hat, über die Ablehnung informiert. Spricht nichts für eine Ablehnung, kann sie genehmigt werden. Die Abwesenheit wird als solche vermerkt und der Antragsteller bekommt über die Genehmigung ein E-Mail zugesandt.

3.3.3 Unified Modeling Language

Die Applikation besteht aus mehreren Datenbanktabellen, die zusammen in Beziehung stehen. Diese Zusammenhänge sollen in diesem Unterkapitel erläutert werden.

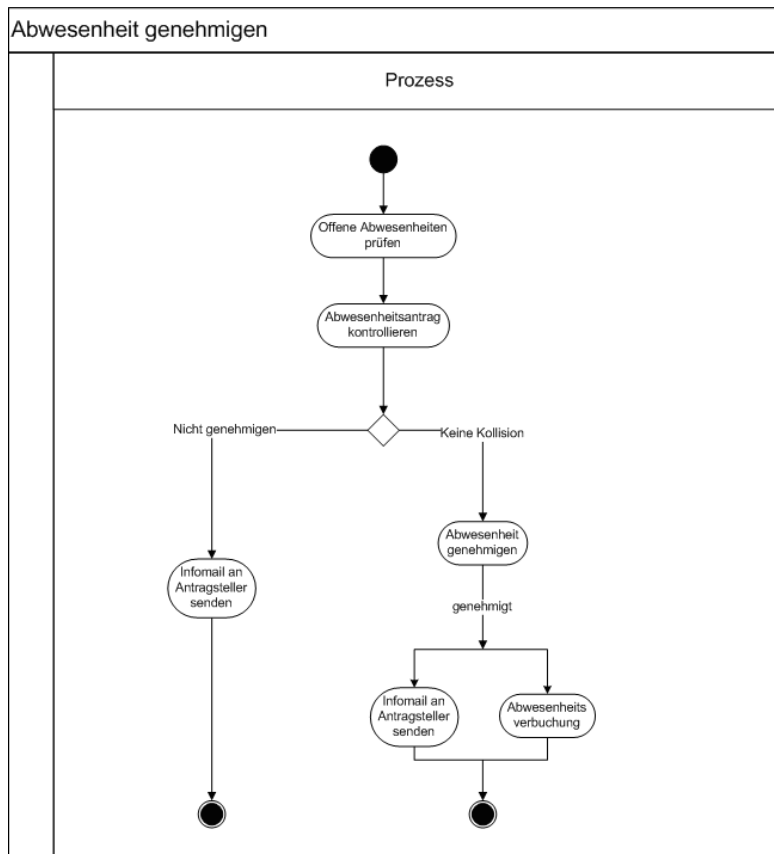


Abbildung 3.17: Abwesenheitsgenehmigung

Vereinfacht ist Unified Modeling Language (UML) eine Familie grafischer Notationen, die bei der Entwicklung und Beschreibung von Softwaresystemen helfen [Fowler, 2004, S. 19]. Mit Hilfe dieser Sprache stellt man Entitätstypen, Attribute, Beziehungstypen und die Multiplizitäten leicht verständlich und übersichtlich dar [Moos, 2004, S. 48].

Im Abbildung 3.18 erkennt man die Zusammenhänge der Datenbanktabellen. Personaldaten werden in der Tabelle PA0001 in SAP gespeichert. In ihr sind sowohl die eindeutige Personalnummer (PERNR), der Name (ENAME) wie auch das Kürzel der Organisationseinheit (ORGTEX) der Mitarbeiterin oder des Mitarbeiters gespeichert. Welche Abteilung hinter der jeweiligen Organisationseinheit steht, erfährt man in der Tabelle T527X, in der man den Organisationseinheitenkurztext (ORGTX) findet. Jede Person gehört einer Abteilung an, diese kann wiederum über mehrere Personen als Personal verfügen. Mitarbeiterinnen, Mitarbeiter und Vorgesetzte gehören zum Personal, wobei jede Mitarbeiterin und jeder Mitarbeiter genau eine Vorgesetzte oder einen Vorgesetzten hat. Eine Vorgesetzte oder ein Vorgesetzter kann über mehrere Mitarbeiterinnen und Mitarbeiter verfügen.

Personal kann Abwesenheiten beantragen. Diese werden in der Tabelle (ZPAbwesenheiten) zwischengespeichert. Diese Tabelle ist in SAP nicht standardmäßig vorgegeben, sondern wurde

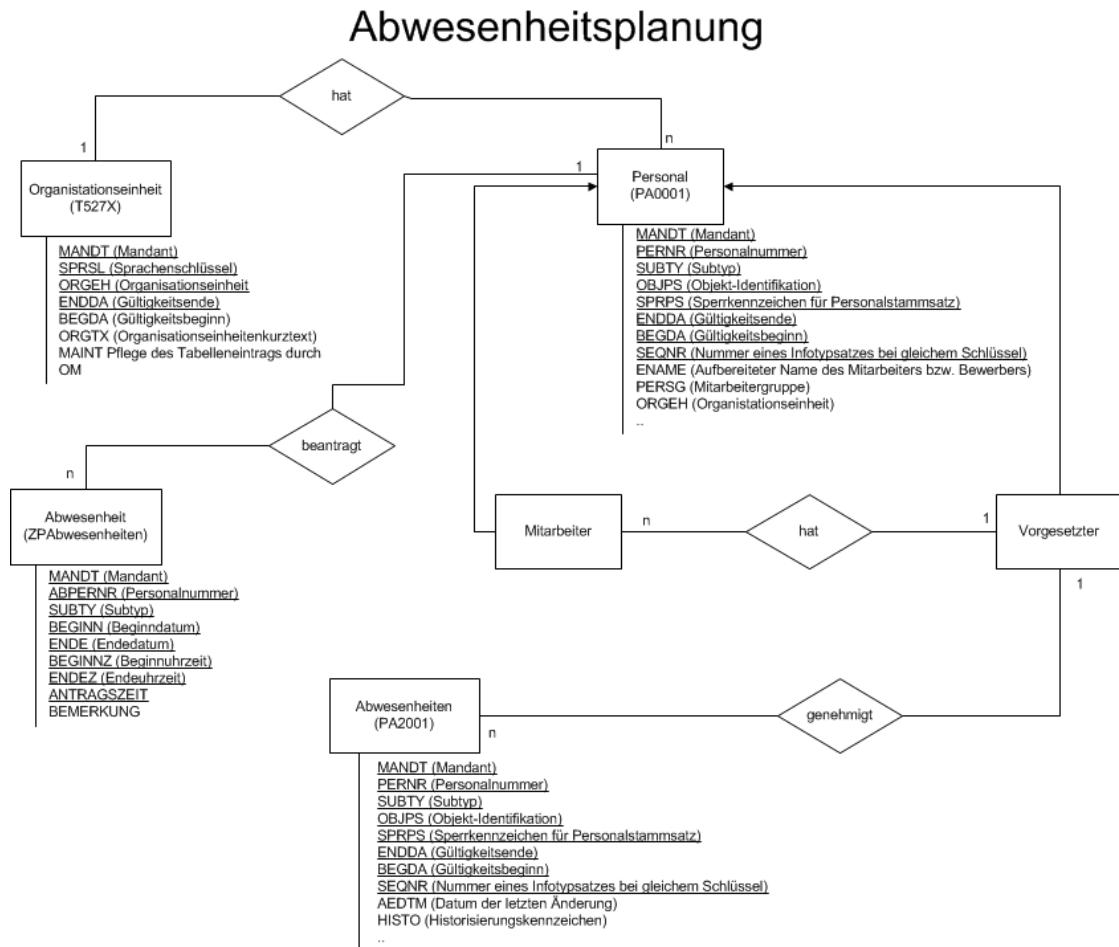


Abbildung 3.18: UML-Diagramm

für die Abwesenheitsplanungapplikation angelegt. Für jeden Eintrag werden die Personalnummer (ABPERNR) des Antragstellers, die Nummer der Abwesenheitsart (SUBTY), der Antrags-, Beginn- und der Endzeitpunkt und eventuelle Bemerkungen abgespeichert. Wird eine Abwesenheit genehmigt, wird diese in der Tabelle PA2001 eingetragen und in der „Zwischentabelle“ ZPAbwesenheiten gelöscht. Die Tabelle PA2001 ist eine „Standardtabelle“ von SAP, die nicht erweiterbar ist. Wenn die Daten direkt dort hineingeschrieben werden, sind diese Abwesenheiten für die Abwesenheitsapplikation bereits genehmigt, da es nicht möglich ist, sie dort als „vorläufig“ (bis sie genehmigt sind) zu kennzeichnen. Daher ist eine „Zwischentabelle“ nötig, in der die beantragten, aber noch nicht genehmigten Abwesenheiten vermerkt werden.

Wurden bis jetzt die Anforderungen und der grundlegende Aufbau der Abwesenheitsplanungapplikation geplant, wurde nun versucht das Layout zu gestalten.

3.4 Layout der Abwesenheitsplanung

In diesem Kapitel wurde versucht, die aus der Theorie in den Abschnitten 2.3 und 2.4 gewonnenen Kenntnisse in die Praxis umzusetzen. Das zukünftige Abwesenheitsprogramm soll über ein Graphical User Interface verfügen. Für die Gestaltung des dazugehörigen Layouts müssen einige Dinge beachtet werden. Auf Papier wurden Skizzen des Layouts entwickelt, die anschließend mit Benutzerinnen und Benutzern besprochen wurden.

Zu Beginn wurde überlegt, wie das Layout gestaltet werden soll. Dafür wurden mit der Hand Skizzen auf weißes Papier gezeichnet, da sie ein gutes Werkzeug darstellen, die Diskussion fördern und leicht zu ändern sind. [Cooper et al., 2007, S. 133]. Bei dem Layout wurde versucht die 8 goldenen Regeln des Schnittstellendesigns von Ben Shneiderman, wie in 2.4.2 beschrieben, in die Praxis umzusetzen. So wurde zum Beispiel auf Konsistenz geachtet. Dialoge sind durchwegs gleich oder ähnlich gestaltet. Farbe, Schrift, Menüs etc. werden in der Applikation nicht verändert.

Danach wurden diese Layout-Skizzen mit drei Mitarbeiterinnen und Mitarbeitern besprochen. Diese Vorgehensweise orientiert sich an den User-Centered Design-Ansatzes, der in Abschnitt 2.2 erläutert wurde, welche die Einbeziehung der späteren Benutzerinnen und Benutzer fordert. Dabei kam es zu einigen Punkten, die abgeändert bzw. ergänzt worden sind. Das Ergebnis waren folgende Layouts:

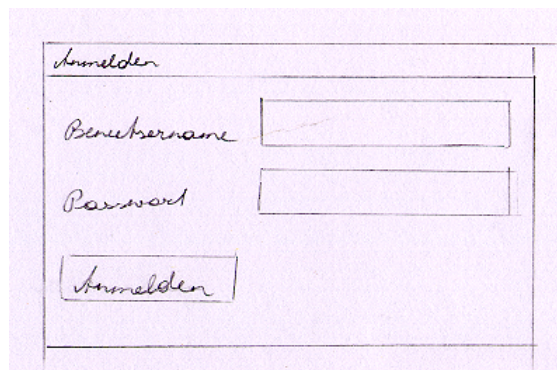


Abbildung 3.19: Anmeldefenster

In Abbildung 3.19 wird das Fenster für die Anmeldung gezeigt. Alle Benutzerinnen und Benutzer haben über dieses die Möglichkeit sich anzumelden. Es ist sehr einfach gehalten. Die Anwenderinnen und Anwender müssen nur ihren Benutzernamen und ihr Passwort eintippen. Durch Klicken auf den „Anmelden“-Button werden sie zum eigentlichen Programm weitergeleitet.

3.4.1 Layout der Mitarbeiterinnen und Mitarbeiter

Die Anwenderin und der Anwender haben in der Standardansicht, in Abbildung 3.20 dargestellt, einen Kalender zur Verfügung. Der aktuelle Tag ist sowohl farblich markiert (in der Abbildung grün) als auch die Zahl selbst gekennzeichnet. Links oben befindet sich ein kleiner Kalender mit



Abbildung 3.20: Standardansicht der Mitarbeiterinnen und Mitarbeiter

einer Monatsansicht. Mit einem Klick auf ein bestimmtes Datum wird der jeweilige Zeitraum um dieses Datum rechts im Hauptbereich angezeigt. Durch einen Doppelklick auf einen bestimmten Tag gelangt man direkt zur Tagesansicht. Die Abwesenheitsanträge der Person selbst sind, neben dem Grund der Abwesenheit, ebenfalls mit einer Farbe gekennzeichnet (in der Abbildung rot). Damit die Anwenderinnen und Anwender auf einen Blick sehen, wann ihre engsten Kolleginnen und Kollegen, also die, deren Vertretung sie darstellen, abwesend sind, sind diese Tage ebenfalls farblich hervorgehoben (in der Abbildung blau). Der Grund dieser Abwesenheiten wird aber verschwiegen, da es nicht von Bedeutung ist. Auch vorgemerkte Abwesenheiten sind farblich (in der Abbildung orange) gekennzeichnet.

Die Jahres-, Wochen und Tagesansicht, die in Abbildung 3.21, 3.22 und 3.23 gezeigt wird, wird immer im jeweiligen Ausschnitt ausgetauscht. Ist bei der Standardansicht noch die des Monats vorhanden, ändert sich diese auf die Tagesansicht, falls das erwünscht ist. Die Jahresansicht besteht aus allen Monaten des Jahres. Der aktuelle Tag ist, falls das aktuelle Jahr dargestellt wird, gekennzeichnet. Mit der Monatsansicht verhält es sich ähnlich. Auch hier ist der aktuelle Tag, falls er enthalten ist, markiert.

Links neben dem Hauptbereich befindet sich ein Fenster, in dem alle Kolleginnen und Kollegen der Anwenderinnen und Anwender aufgelistet sind, mit denen diese ihre Abwesenheiten abstimmen müssen.

Hat eine Anwenderin oder ein Anwender den Wunsch eine Abwesenheit zu beantragen, macht sie oder er das, indem sie oder er auf den Button „neue Abwesenheit“ klickt. Daraufhin öffnet sich ein Formularfenster, wie in Abbildung 3.24 dargestellt, in dem man sowohl den Anfangszeitpunkt der Abwesenheit als auch den Endzeitpunkt bekannt geben muss. Falls der Abwesenheitszeitraum nur über ein paar Stunden gehen soll, können auch Stunden ausgewählt

Abbildung 3.21: Jahresansicht

	MO 02.08.2011	DI 03.08.2011	MI 04.08.2011	DO 05.08.2011	FR 06.08.2011	SA 09.08.2011	SO 08.08.2011
07:00					Zeitraumgleich		A
08:00							
09:00							
10:00							
11:00							
12:00							
13:00							
14:00							
15:00							
16:00							
17:00							
18:00							
19:00							
20:00							B

Abbildung 3.22: Wochenansicht

MONTAG, 2 August 2011	
07:00	
08:00	
09:00	
10:00	
11:00	11
12:00	
13:00	
14:00	
15:00	
16:00	
17:00	
18:00	

Abbildung 3.23: Tagesansicht

werden. Ein Abwesenheitstyp muss ebenfalls ausgewählt werden. Es kann nur eine Abwesenheit beantragt werden oder mit Hilfe des „weiter beantragen“-Buttons auch mehrere Abwesenheiten. Mit den „vor“- und „zurück“- Buttons kann man zwischen den Abwesenheiten, die man beantragen möchte, hin und her wechseln. Diese Funktion soll eingebaut werden, da man eventuell noch etwas vergessen hat, oder die zuvor eingetragene Abwesenheit noch einmal ändern möchte. Zudem soll es eine Funktion der Umkehr, wie in Abschnitt 2.4.2 beschrieben, geben, damit den Anwenderinnen und Anwendern die Angst genommen wird, Aktionen, bei denen sie sich unsicher sind, auszuführen. Durch die Betätigung des „senden“-Buttons werden zuvor eingetragene Abwesenheiten beantragt. Die oder der zuständige Vorgesetzte wird per E-Mail über die Abwesenheitswünsche informiert und können die Abwesenheiten genehmigen oder ablehnen. Über diese Entscheidung wird die Benutzerin und der Benutzer per E-Mail informiert, wie die Abbildung 3.25 zeigt. In dieser Nachricht sind die Abwesenheitsart, der Zeitraum dafür, das Entscheidungsergebnis und der Name der oder des Vorgesetzten enthalten.

3.4.2 Layout der Vorgesetzten

Das Standardlayout der oder des Vorgesetzten, wie es in Abbildung 3.26 zu sehen ist, ist dem der normalen Mitarbeiterinnen und Mitarbeiter sehr ähnlich. In der Liste (links) sind neben den direkten Kolleginnen und Kollegen (falls vorhanden) auch alle Mitarbeiterinnen und Mitarbeiter aufgelistet, von denen die Anwenderin oder der Anwender die Funktion der oder des Vorgesetz-

Abwesenheit beantragen

Zurück Vor

Start:

Ende:

Typ: ▾

Bemerkung:

Abbildung 3.24: Abwesenheit beantragen

Von:
Abteilung:
Art:
Zeitraum:
Bemerkung:

/

Abbildung 3.25: Nachricht über Entscheidung der Abwesenheit

ten übernimmt. Um die beiden Gruppen voneinander zu unterscheiden, sind diese mit anderen Farben hinterlegt. Bei den Kalenderansichten werden neben den eigenen Abwesenheiten, auch die der Mitarbeiterinnen und Mitarbeiter angezeigt. Da sie genauso wie ihre Mitarbeiterinnen und Mitarbeiter auch Vorgesetzte haben können, können sie ebenfalls Abwesenheiten beantragen. Sollte das nicht der Fall sein, ist der Button grau und hat keine Funktion.

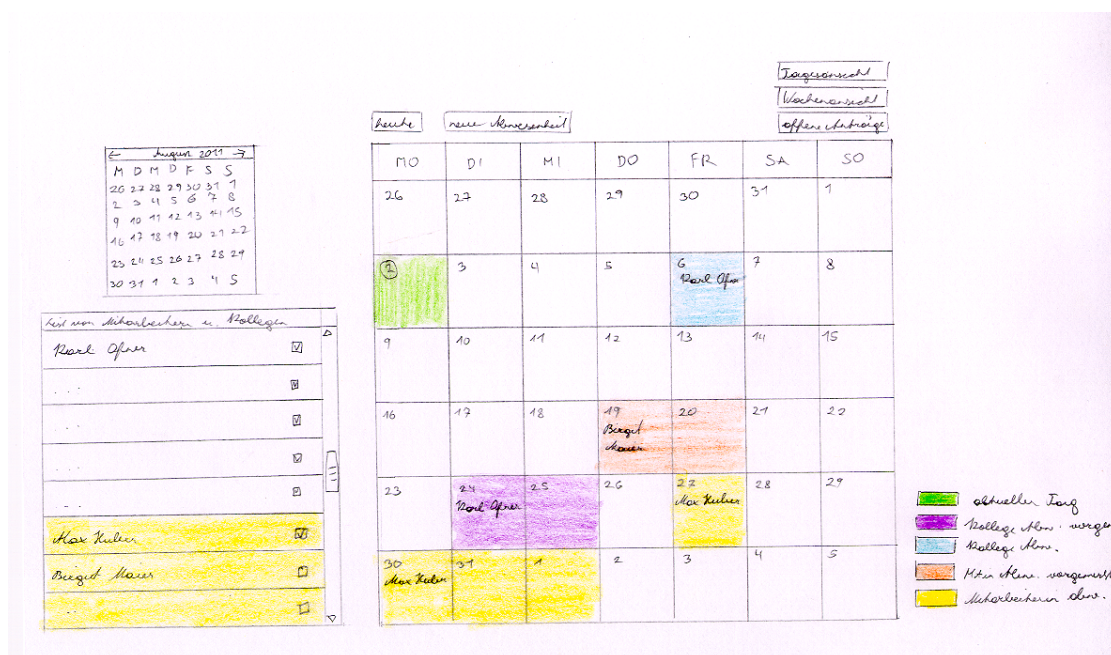


Abbildung 3.26: Standardansicht der oder des Vorgesetzten

Hat eine Mitarbeiterin oder ein Mitarbeiter Abwesenheit beantragt, bekommt die oder der Vorgesetzte ein E-Mail, wie in Abbildung 3.27 an seine herkömmliche E-Mail-Adresse gesandt. Sie oder er muss daher nicht in das Programm einsteigen, nur um nachzusehen, ob die Mitarbeiterinnen und Mitarbeiter Abwesenheiten beantragt haben. Darin ist neben dem Namen der Antragstellerin oder des Antragstellers auch die Art der gewünschten Abwesenheit und eventuelle Bemerkungen enthalten. Über den Button „offene Anträge“ im Programm selbst gelangt die Benutzerin oder der Benutzer zu einer Liste, in der alle offenen Anträge mit den jeweiligen Links aufgeführt werden. Diese Liste ist in Abbildung 3.28 dargestellt. Nach der Gestaltung des Layouts kann mit der eigentlichen Implementierung begonnen werden.

3.5 Implementierung

Wurde die bisherige Entwicklung ausschließlich auf Papier und theoretisch abgehandelt, befasst sich dieses Kapitel mit der eigentlichen Implementierung. Nach der Auswahl der Programmiersprache, wurde ein Programm gefunden, dessen Code frei verfügbar ist und zur Planung von Events verwendet werden kann. Dieses Programm diente als Grundlage und wurde auf für die

Name: Max Mustermann
 Abteilung: XY
 Art: Urlaub
 Zeitraum: MO 02.08.2011 - DI 03.08.2011
 Bemerkung:

genehmigen ablehnen

Abbildung 3.27: Antrag einer Mitarbeiterin oder eines Mitarbeiters

Offene Anträge	
Beleg Max 19.08.2011 - 20.08.2011	A
...	
...	
...	
...	

Abbildung 3.28: Liste alle offenen Anträge

Abwesenheitsplanung umgeändert. Die Programmierung verlief nicht ohne Probleme. Um welche es sich handelt und wie die gelöst wurden, beschreibt dieses Kapitel.

Als Programmiersprachen für die Abwesenheitsplanungsapplikation wurden Microsoft Silverlight und C# ausgewählt. Silverlight ist eine Entwicklungsplattform, mit deren Hilfe man Applikationen fürs Web, Desktop- und auch mobile Anwendungen online oder offline erstellen kann. Silverlight ist ein freies plug-in und kompatibel mit verschiedenen Browsern und Systemen [Sil, 2011]. Das Design und die damit verbundene Logik sind getrennt. Das Design wird in Silverlight in Extensible Application Markup Language (XAML) erstellt. XAML initialisiert Objekte und deren Eigenschaften [xam, 2011].

Als Grundlage für die Abwesenheitsplanung wurde der Silverlight Scheduler von CodePlex (Open Source Project Community) verwendet, dessen Code freiverfügbar ist [Codeplex, 2010].

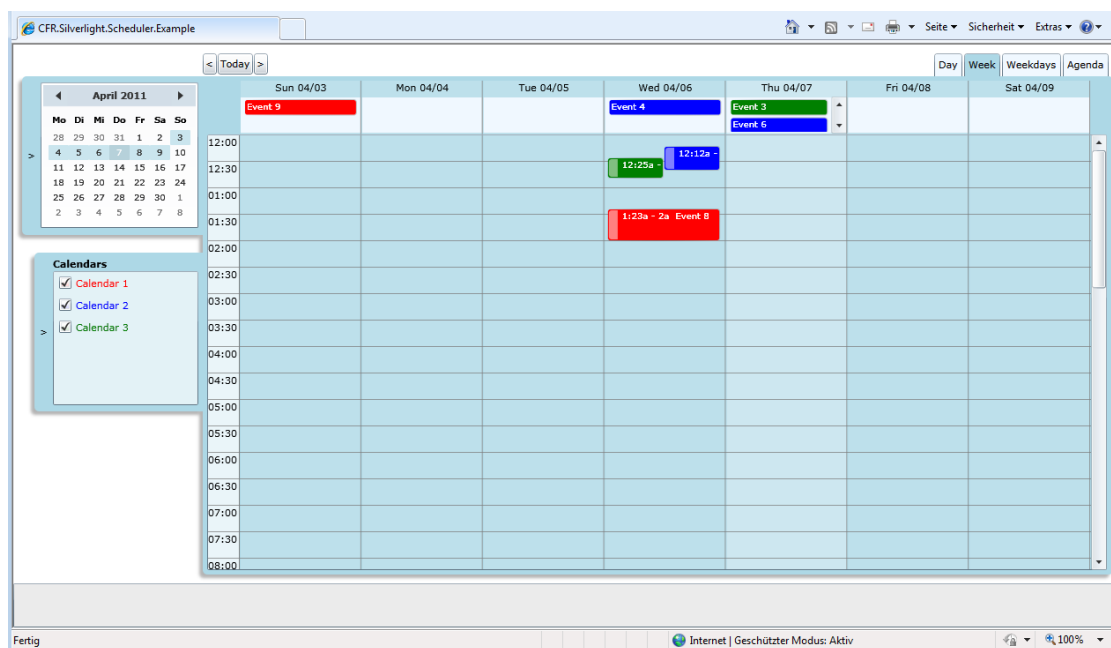


Abbildung 3.29: Silverlight Scheduler

In Abbildung 3.29 wird der Silverlight Scheduler gezeigt, er ist dazu bestimmt um Events zu verwalten. Links neben dem eigentlichen Kalender mit den eingetragenen Terminen befindet sich eine kleine Monatsansicht, auf der man das jeweilige Datum auswählen kann, das dann im Hauptfenster angezeigt wird. Darunter sind die einzelnen „Calendar“ aufgelistet, im Beispiel sind es drei. Jeder dieser Kalender hat eine eigene Farbe und alle Termine zu diesem „Calendar“ sind in derselben Farbe gekennzeichnet. Sowohl der kleine Kalender zur Datumsauswahl als auch die Calendar-Liste lassen sich wegklappen, wie man in Abbildung 3.30 sehen kann. Dort sieht man ebenfalls die Agenda, bei der alle Einträge der aktuellen Woche aufgeführt sind. Events, die über den ganzen Tag andauern, sind im Kalender oberhalb der Zeitleiste angeführt. Neue Events erzeugt man, indem man im Kalender auf eine bestimmte Zeit klickt, dann wird für diese Zeit, ein halbstündiger Event, einfach eingetragen. Termine über einen längeren Zeitraum

sind in der Grundversion nicht vorgesehen. Im Silverlight Scheduler kann zwischen mehreren Kalenderansichten gewechselt werden. Es gibt eine Tagesansicht, wie in Abbildung 3.31 gezeigt, eine Wochenansicht, und eine Wochentageansicht, wie man in Abbildung 3.32 erkennen kann.

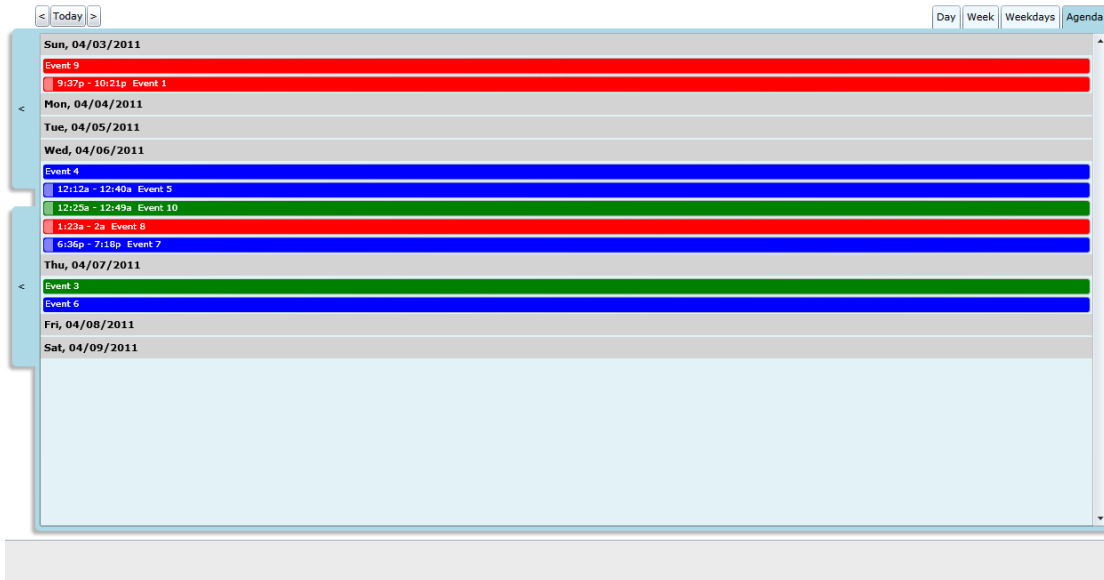


Abbildung 3.30: Silverlight Scheduler - Agenda

Für die Abwesenheitsplanungsapplikation mussten folgende Dinge im Layout geändert werden:

- Ein Login-Fenster muss dem eigentlichen Programm vorgelagert werden. Benutzername und Passwort müssen mit dem SAP-System übereinstimmen, nur so gelangt man zum Hauptprogramm.
- Die Sprache der Applikation soll Deutsch sein, daher sind die Namen der Wochentage, das Datums- und das Uhrzeitformat zu ändern.
- Die Liste mit den „Calendars“ muss gegen eine Liste der Kolleginnen, Kollegen, Mitarbeiterinnen und Mitarbeiter der angemeldeten Benutzerin oder des Benutzers ersetzt werden. Diese müssen aus dem SAP-System ausgelesen werden. Um die Kolleginnen und Kollegen von den Mitarbeiterinnen und Mitarbeitern zu unterscheiden, sollen diese mit einer anderen Farbe hinterlegt werden.
- Es soll möglich sein, Abwesenheiten von einer halben Stunde bis über mehrere Tage zu beantragen. Wochenenden sollen dabei ausgespart bleiben. Gibt die Anwenderin oder der Anwender zum Beispiel eine Woche (7 Tage) Urlaub ein, dann sollen nur 5 Tage von Montag bis Freitag als Urlaub eingetragen werden.

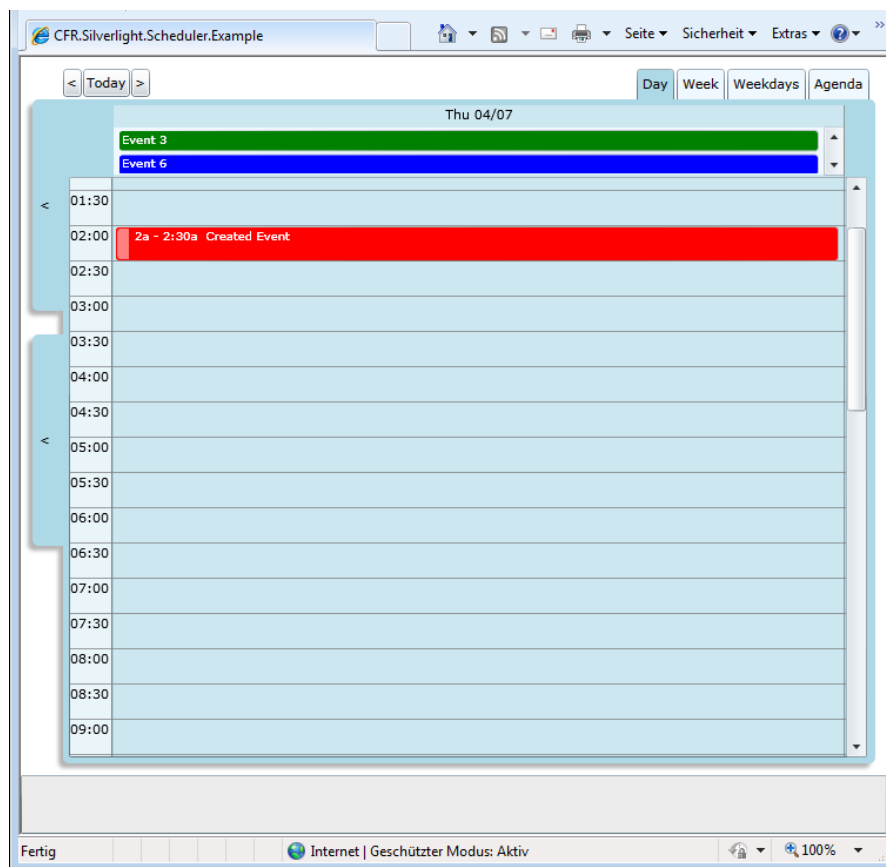


Abbildung 3.31: Silverlight Scheduler - Tagesansicht

- Abwesenheiten sollen mit unterschiedlichen Farben dargestellt werden. Abwesenheiten von Kolleginnen und Kollegen sollen eine einheitliche Farbe besitzen, es macht keinen Unterschied, welcher Art sie sind, noch ob sie genehmigt oder nur beantragt sind. Von Mitarbeiterinnen und Mitarbeitern sollen sie, falls sie erst beantragt sind, einheitlich gekennzeichnet werden. Sind sie bereits genehmigt, werden sie je nach der Abwesenheit unterschiedlich eingefärbt.
- Zu den Abwesenheiten sollen folgende Informationen angegeben werden:
 - Abwesenheitsart
 - Person, die die Abwesenheit konsumieren möchte bzw. konsumiert hat.
 - Zeitraum (Datum mit Uhrzeit)
 - Beantragt oder genehmigt
- Funktionen zum Löschen und Editieren von Abwesenheit sollen hinzugefügt werden.

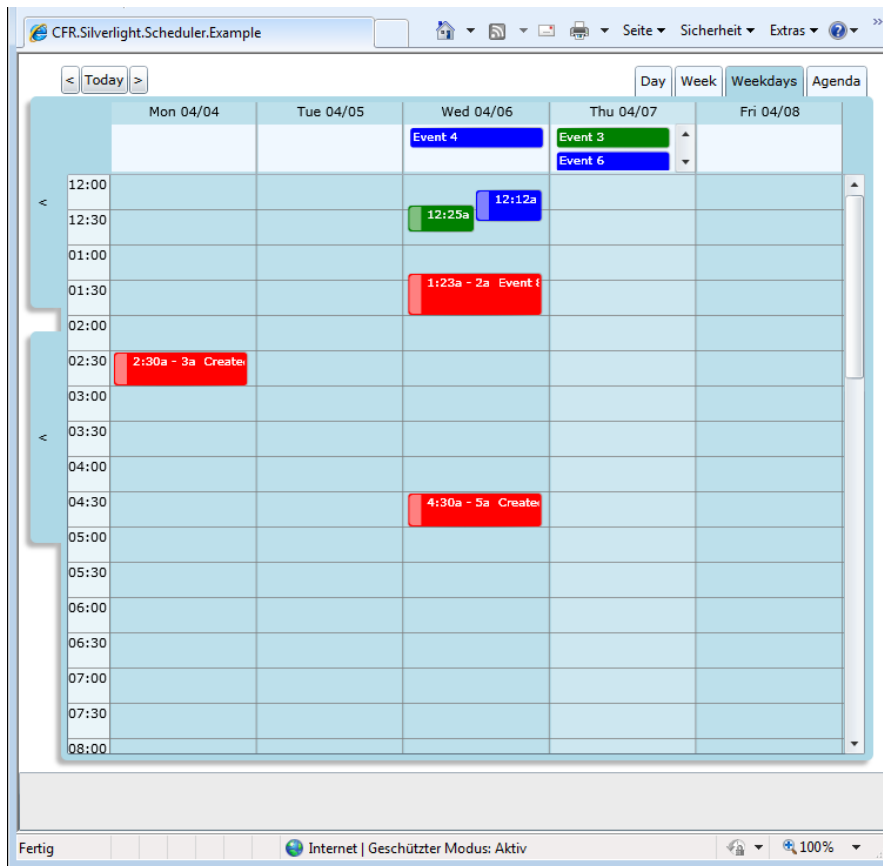


Abbildung 3.32: Silverlight Scheduler - Wochentageansicht

- Neben „Day“, „Week“, „Weekdays“ und „Agenda“ soll es einen neuen Bereich „Offene Anträge“ geben, bei dem nur die noch nicht genehmigten Abwesenheiten für eine oder einen Vorgesetzten aufgelistet werden.
- Die Applikation soll über eine Hilfe verfügen.

Die größte Herausforderung bei der Programmierung selbst war die Verbindung des Silverlight-Programms mit SAP. Es handelt sich dabei um SAP der Version SAP ECC 6.0, wobei die SAP_HR Release 600 Level 0011 und SAP_ABA Release 700 Level 0011 ist. Für die Verbindung gibt es mehrere Möglichkeiten, die genauer in Abschnitt 2.9.2 erklärt werden. Für die Applikation wurden Web Services verwendet, da diese Variante eine einfache Lösung des Problems darstellt.

In Funktionsbausteine werden dabei die eigentlichen Aufgaben erfüllt, die man von dem SAP-System benötigt. Im Fall der Abwesenheitsplanung ist das zum Beispiel das Abrufen der verschiedenen Abwesenheiten oder das Schreiben in einer Tabelle von SAP. Technisch gesehen ist ein Funktionsbaustein wie ein Unterprogramm einer Prozedur eines ABAP-Programms

[Keller und Krüger, 2001, S. 255]. Dabei stellte es eine Schwierigkeit dar, in SAP die Tabellen zu finden, in denen die jeweiligen Daten vorhanden waren.

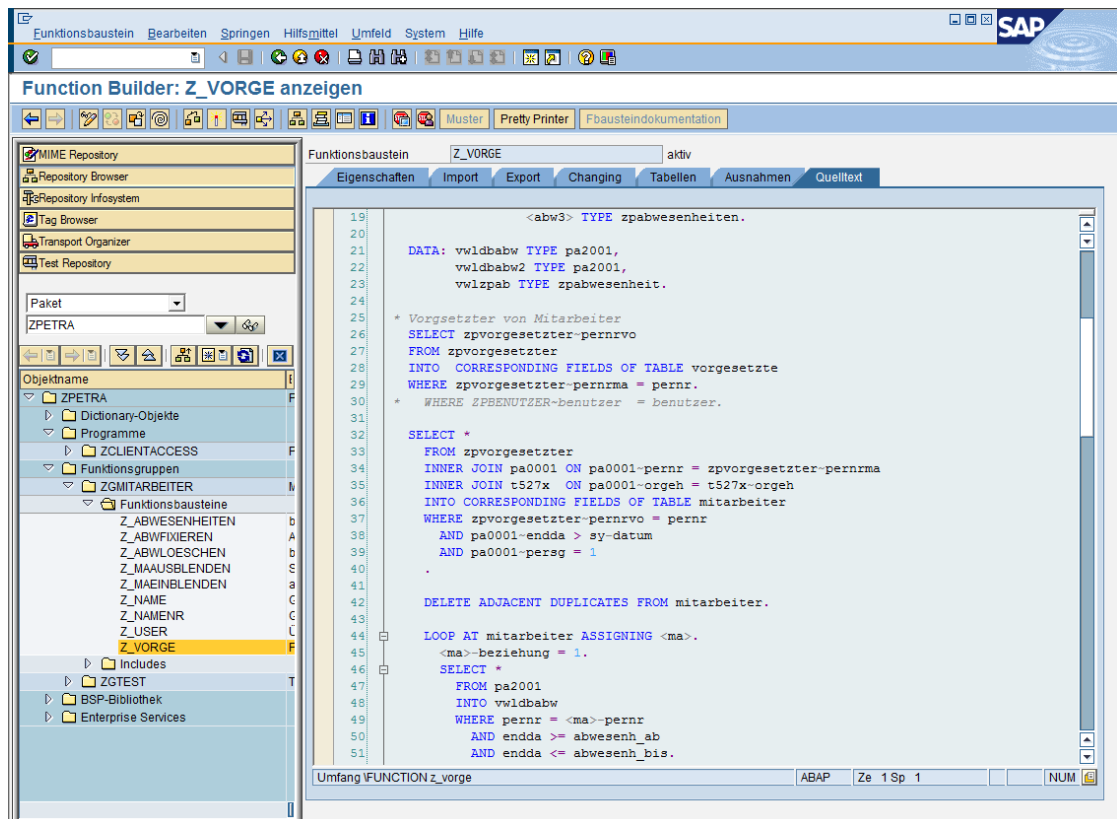


Abbildung 3.33: Beispiel eines Funktionsbausteins in SAP

Mehrere Funktionsbausteine werden in Funktionsgruppen zusammengefasst, die dabei so eine Art Behälter sind. Damit Funktionsbausteine nicht umsonst in den Speicher geladen werden, sollen nur solche zusammen in einer Funktionsgruppe angelegt werden, die auch gemeinsame Komponenten der Funktionsgruppe nutzen [Keller und Krüger, 2001, S. 256f]. Aus Funktionsbausteinen können Web Services angelegt werden. Näheres zu Web Service erfährt man im Abschnitt 2.9.2. Wenn aus einer Funktionsgruppe ein Web Service generiert wird, dann enthält es alle in der Funktionsgruppe enthaltenen Funktionsbausteine.

Über den SAP-Bereich WSADMIN - Administration Web Service gelangt man zum dazugehörigen Pfad des XML-Files.

Zwar lässt sich dieses Web Service in den Silverlight/C#-Code einbinden, es ist allerdings nicht funktionsfähig, da die domänenübergreifenden Richtlinien fehlen. Laut MSDN Library muss der Dienst, der in einer anderen Domain liegt, explizit ausgewählt werden, damit der domänenübergreifende Zugriff aktiviert wird. Durch dieses Auswählen bestätigt ein Dienst, dass die Vorgänge, die er verfügbar macht, ohne Bedenken der Sicherheit von einem Silverlight 4-Steuerungselement aufgerufen werden kann und dadurch keine Daten beschädigt werden

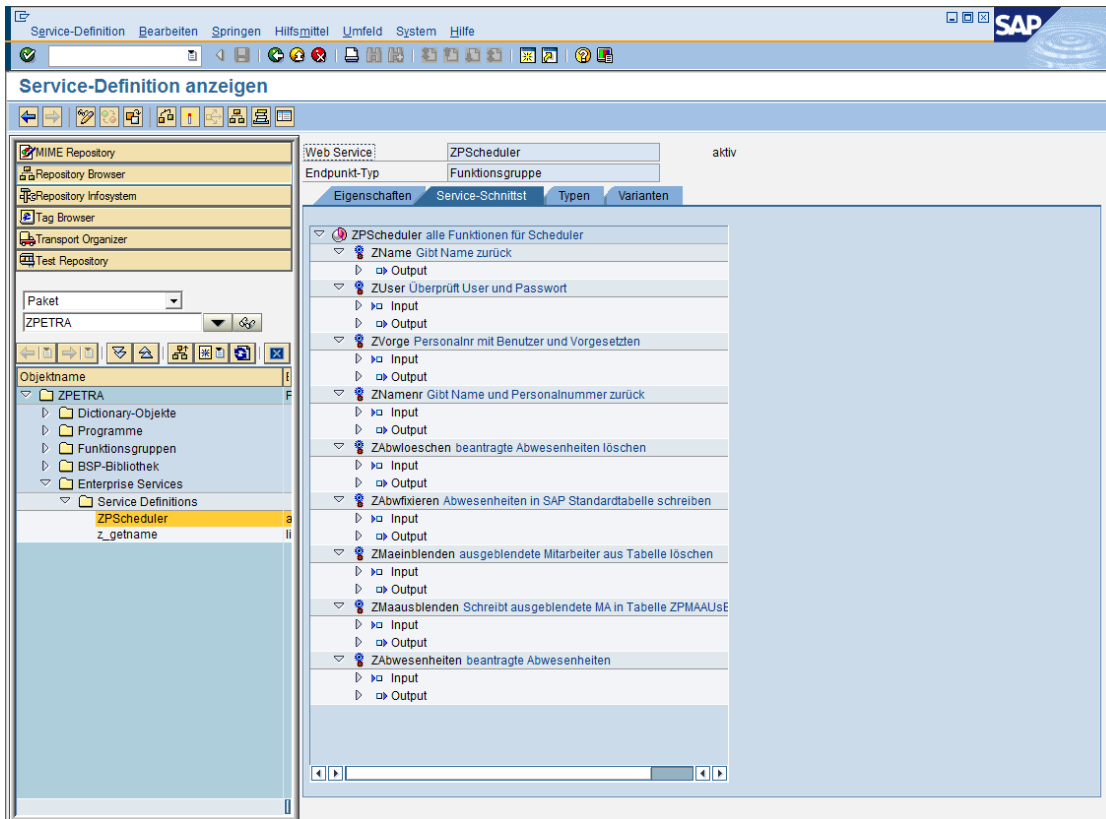


Abbildung 3.34: Web Service einer Funktionsgruppe

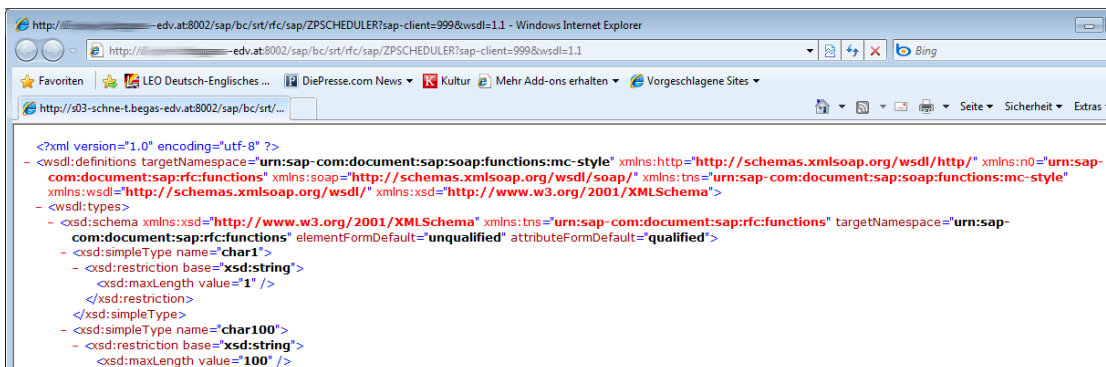


Abbildung 3.35: Pfad des XML-Files des Web Service

[Ric, 2011]. Dazu soll man ein `clientaccesspolicy.xml`-File im Stammverzeichnis der Domäne stellen. Leider ist das bei einem SAP-System nicht möglich. Durch lange Internetrecherche und viele Versuche wurde im Bereich SE80 (Object Navigator) ein eigenes Programm, genannt „crossdomain“, erstellt, welches im Anhang A bereitgestellt wird.

Die Applikation verfügt über keine eigene Datenbank. Alle Tabellen werden aus dem SAP-System verwendet. Tabellen, die das Programm benötigt, aber im SAP-System für den benötigten Zweck nicht vorhanden waren, wurden dort neu erstellt. Zum Beispiel eine Tabelle, die die beantragten, aber noch nicht genehmigten Abwesenheiten enthält.

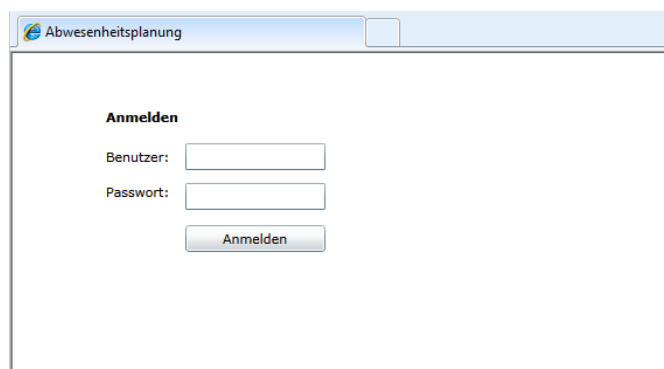
Eine weitere Herausforderung war, wie man durch die Silverlight-Applikation Daten in bereits im SAP-System bestehende Tabellen schreibt, zum Beispiel, wenn Abwesenheiten genehmigt wurden. Dabei wurden die entsprechenden Daten von der Tabelle, die noch nicht genehmigte Abwesenheiten enthält, in die Tabelle geschrieben, in der das SAP-System diese Daten vorsieht. Leider war diese Funktion nicht durch einfaches Web Service umzusetzen. SAP, dessen Code frei zugänglich ist, beinhaltet allerdings schon einige Funktionsbausteine. Nach längerer Suche wurde auch einer solcher gefunden (`BAPI_ABSENCE_CREATE`), der diese Funktion erfüllt.

Trotz einiger Probleme konnte die Abwesenheitsapplikation fertig gestellt werden und das Ergebnis soll nun erläutert werden.

3.6 Programm

Zwar hat sich die Programmierung nicht als problemlos herausgestellt, es konnten allerdings dennoch Lösungen gefunden werden. Das Ergebnis des Prototypen, wie die Funktionen, die gewählt wurden und das Layout, sollen in diesem Kapitel gezeigt werden.

Für den Prototyp wurde ein einfaches Login-Fenster, wie man in Abbildung 3.36 erkennen kann, erstellt. Eigentlich war geplant, dass man den „Anmelden“-Button auch mittels Drücken der Enter-Taste betätigen kann. Internetrecherche brachte allerdings das Ergebnis, dass ein Default-Button in Silverlight nicht Standard ist und nur umständlich umgesetzt werden könnte. Da es sich um einen Prototyp handelt, wurde auf diese Funktion verzichtet.



The image shows a screenshot of a web application window titled "Abwesenheitsplanung". The window has a light blue header bar with the title and a small icon on the left. Below the header, the main content area is white and contains a login form. The form is titled "Anmelden" in bold. It consists of two text input fields: the first is labeled "Benutzer:" and the second is labeled "Passwort:". Below these fields is a button labeled "Anmelden".

Abbildung 3.36: Anmelden

Nach der Anmeldung mit einem Benutzernamen und einem dazugehörigen Passwort, die mit den Benutzernamen und Passwörter in einer SAP-Tabelle verglichen werden, kommt man in das eigentliche Abwesenheitsprogramm. Zwar wurde versucht, wie in 2.4.2 näher beschrieben, Fehler möglichst von vornherein auszuschließen, dennoch gibt es Situationen, bei denen es zu Fehlern kommen kann. Einen dieser Fehler kann die Benutzerin oder der Benutzer begehen, indem sie oder er einen falschen Benutzernamen und/oder Passwort eingibt. Daraufhin wird die Fehlermeldung „Benutzername oder Passwort existiert nicht oder stimmen nicht überein!“ ausgegeben und damit der Person der Grund ihres Fehlers bekannt gegeben.

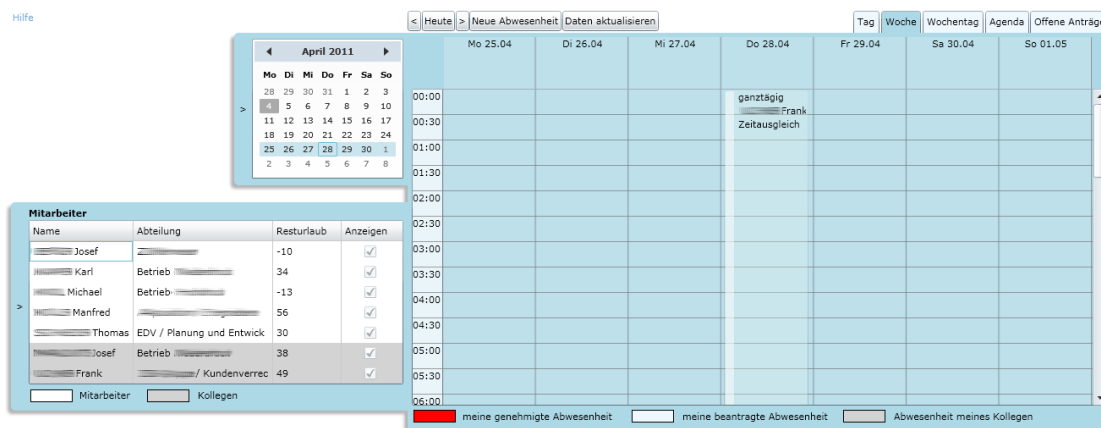


Abbildung 3.37: Hauptbereich

Der Hauptbereich besteht aus einem Kalender. Horizontal sieht man die Tage Montag bis Sonntag einer jeweiligen Woche. Horizontal die Uhrzeit. Mittels Tabs (Karteikarten), lassen sich die Kalenderansichten in einen einzelnen Tag, die ganze Woche oder in Wochentage (Montag bis Freitag) wechseln. Die Monatsansicht, die prinzipiell vorgesehen war, wurde im Prototypen nicht realisiert. Im Kalender sind die einzelnen Abwesenheiten in vertikalen Balken eingetragen. Abwesenheiten, der Kolleginnen und Kollegen sind in der Farbe Grau gestaltet worden. Man erfährt nicht, um welche Art von Abwesenheit es sich handelt, da diese Information für Kolleginnen und Kollegen nicht von Bedeutung ist. Entscheidend ist nur, dass sie oder er nicht anwesend ist. Mittels Tooltip lassen sich zu Abwesenheiten mehr Details erfahren, unter anderem, ob die Abwesenheit schon genehmigt oder nur beantragt ist. Es wurde diese Variante gewählt, da die Information für Anwenderinnen und Anwender nicht essentiell ist und die Information, die direkt bei den Abwesenheiten selbst steht, möglichst gering gehalten werden wollte, um die Benutzerinnen und Benutzer nicht mit zu viel Information zu überfordern.

Die im Unternehmen arbeitenden Personen müssen ihre Abwesenheiten aufeinander abstimmen. Um das zu bewerkstelligen, müssen sie von einander Informationen erhalten. Welche Arten von Awareness es geben kann, wird im Abschnitt 2.1.2 näher behandelt. In der Abwesenheitsplanungapplikation wurde das folgendermaßen umgesetzt: Abwesenheiten von Mitarbeiterinnen und Mitarbeitern können unterschiedliche Farben annehmen. Ist die Abwesenheit nur beantragt, aber noch nicht genehmigt, ist sie grau. Genehmigte Abwesenheiten von Mitar-

beiterinnen und Mitarbeitern nehmen je nach Abwesenheitsart unterschiedliche Farben an, zum Beispiel für Urlaub ist es ein zartes Grün. Eigene genehmigte Abwesenheiten sind mit einem kräftigen Rot gekennzeichnet, damit sie gegenüber anderen Abwesenheiten herausstechen. Ist die Abwesenheit noch nicht genehmigt, nimmt sie einen weißen bis durchsichtigen Farbton an. Eine Legende wurde im unteren Bereich der Kalenderansicht hinzugefügt. Sie beinhaltet allerdings nur die Erklärungen für die eigenen Abwesenheiten und die der Kolleginnen und Kollegen. Auf die einzelnen Farben der Abwesenheiten wurde nicht näher eingegangen, da diese sehr vielfältig sind und dies für die Benutzerinnen und Benutzer zu unübersichtlich geworden wäre. Da bei den Abwesenheiten von Mitarbeiterinnen und Mitarbeitern der Abwesenheitstyp dabei steht, sollte die Farbe nicht mehr einen so großen Erklärungsbedarf haben. In der Hilfefunktion der Applikation sind allerdings alle Farbvarianten aufgeführt. Somit nehmen Mitarbeiterinnen und Mitarbeiter Abwesenheiten ihrer Kollegen wahr, um diese mit ihren eigenen abzustimmen. Die Art der Abwesenheit wird ihnen, im Gegensatz zu ihren Vorgesetzten, allerdings verschwiegen.

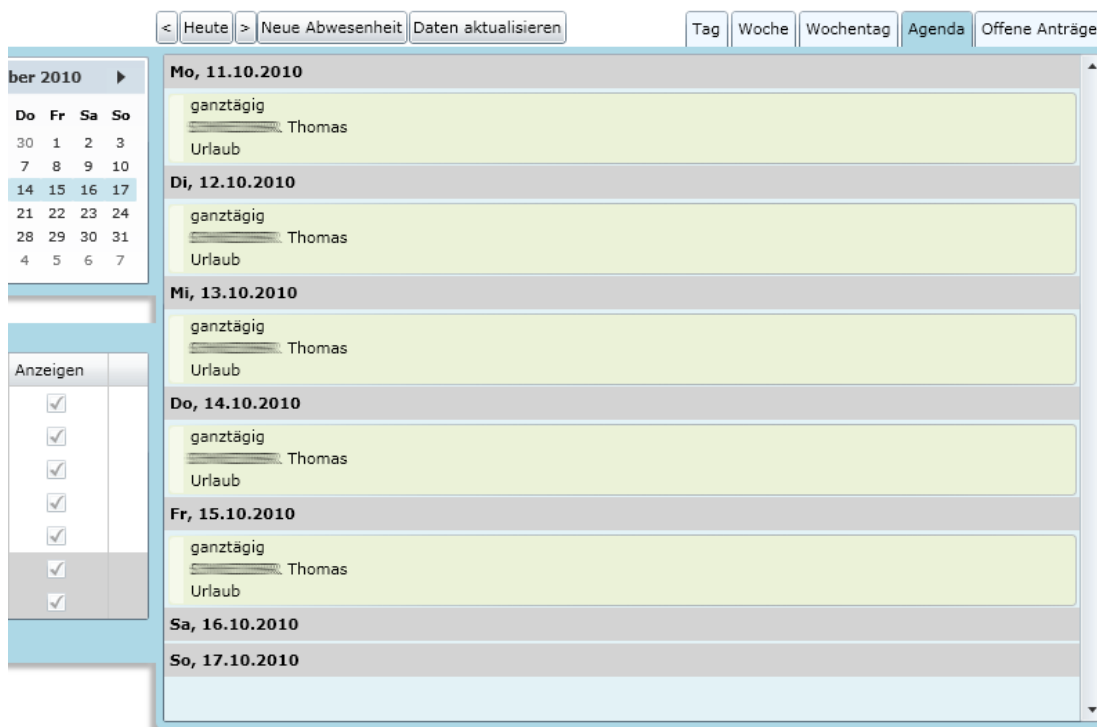


Abbildung 3.38: Agenda

Im Bereich „Agenda“, wie man in Abbildung 3.38 sehen kann, lassen sich die Abwesenheiten für den zuvor ausgewählten Bereich zusammengefasst (mit Datumsangabe, aber ohne Zeitangaben) sehen.

In „Offene Anträge“, wie in Abbildung 3.39 ersichtlich, sehen Vorgesetzte offene Anträge, also Anträge, die erst beantragt wurden, allerdings noch nicht genehmigt sind, aufgelistet. Um diese Anträge zu genehmigen, gibt es zwei Möglichkeiten. Entweder man klickt im Kalender

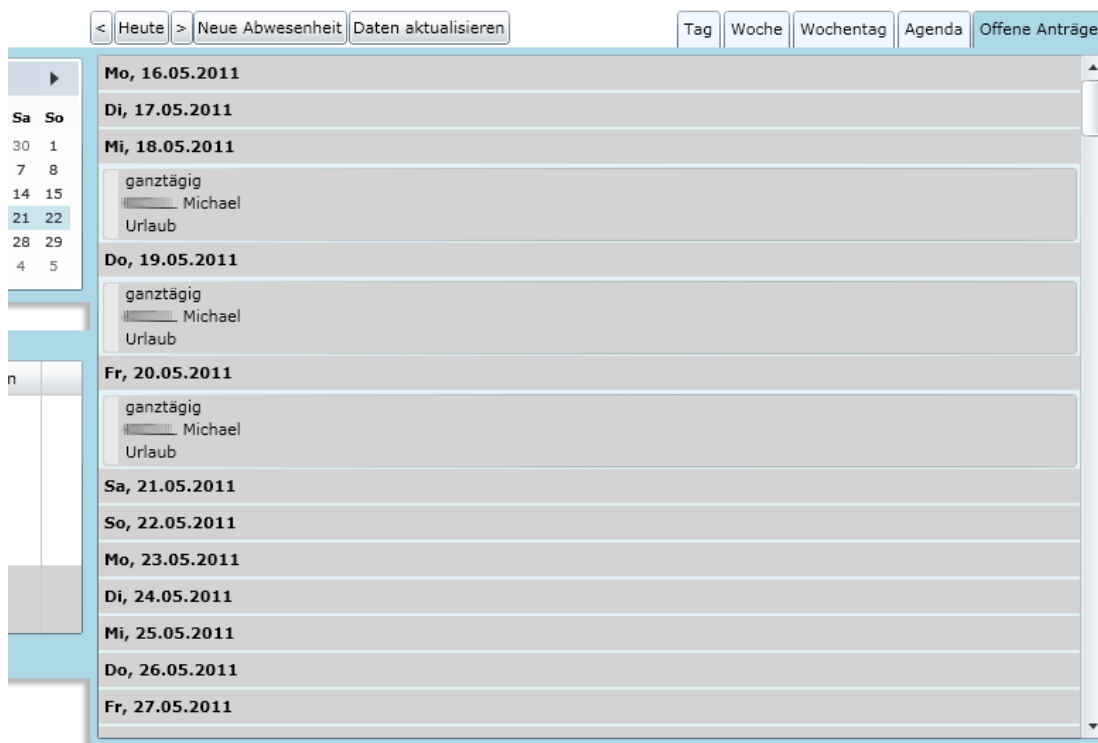


Abbildung 3.39: Offene Anträge

auf einen dieser noch nicht genehmigten Anträge oder man klickt im Bereich „Offenen Anträge“ auf den jeweiligen Antrag. Daraufhin öffnet sich ein Fenster, wie in Abbildung 3.40, in dem die Daten der Abwesenheit und eventuell von der Antragstellerin oder dem Antragsteller vermerkte Bemerkungen aufgelistet sind. Die oder der Vorgesetzte kann die Abwesenheit genehmigen oder nicht. Bei der Genehmigung wird die Abwesenheit von einer „Zwischentabelle“ von SAP, in der die beantragten Abwesenheiten gespeichert werden, in eine SAP-„Standardtabelle“ verschoben, in die Tabelle in der SAP Abwesenheiten speichert. Im Abwesenheitsplanungsapplikation ist diese Veränderung nur erkennbar, da die Abwesenheit nun nicht mehr als beantragt, sondern als genehmigt gekennzeichnet ist. Wird eine Abwesenheit abgelehnt, dann wird sie aus der Datenbank gelöscht. In beiden Fällen bekommt, die Mitarbeiterin oder der Mitarbeiter, der die jeweilige Abwesenheit beantragt hat, eine Nachricht per E-Mail über die Entscheidung der oder des Vorgesetzten.

Links werden alle Kolleginnen und Kollegen, mit denen die Anwenderin oder der Anwender seine Abwesenheiten koordinieren muss, aufgelistet. Im Falle, dass sie oder er eine Vorgesetzte bzw. Vorgesetzter ist, werden auch alle Mitarbeiterinnen oder Mitarbeiter aufgelistet, deren Abwesenheiten genehmigt oder abgelehnt werden müssen. Um Kolleginnen und Kollegen von Mitarbeiterinnen und Mitarbeitern unterschieden werden können, sind sie in unterschiedlichen Farben hinterlegt worden. In den Befragungen zu den Wünschen der Anwenderinnen und An-

Genehmigung Mi. 18.05.2011

Von: Michael

Art: Urlaub

Zeitraum: 2011-05-18 00:00:00 bis 2011-05-20 23:59:59

Bemerkung:

genehmigen ablehnen abbrechen

So, 29.05.2011

Abbildung 3.40: Abwesenheiten genehmigen

Mitarbeiter

Name	Abteilung	Resturlaub	Anzeigen
Josef		-10	<input checked="" type="checkbox"/>
Karl	Betrieb	34	<input checked="" type="checkbox"/>
Michael	Betrieb	-13	<input checked="" type="checkbox"/>
Manfred		56	<input checked="" type="checkbox"/>
Thomas	EDV / Planung und Entwick	30	<input checked="" type="checkbox"/>
Josef	Betrieb	38	<input checked="" type="checkbox"/>
Frank	/ Kundenverrec	49	<input checked="" type="checkbox"/>

Mitarbeiter Kollegen

Abbildung 3.41: Mitarbeiterliste

wender, wurde oft die Möglichkeit der Ausblendung von sowohl Kolleginnen und Kollegen als auch Mitarbeiterinnen und Mitarbeitern genannt. Dem Wunsch wurde entsprochen. Neben dem Namen und der Abteilung kann man mittels eines Hakens die jeweilige Person ausblenden. Die Funktion macht dann Sinn, wenn die Person längere Zeit, zum Beispiel bei Bildungsurlaub oder Karenz im Unternehmen fehlt. Wie in Abbildung 3.41 ersichtlich, wird ebenfalls an dieser Stelle der Resturlaub der jeweiligen Personen angezeigt.

Abbildung 3.42: Abwesenheit beantragen

Um Abwesenheiten zu beantragen, haben Anwenderinnen und Anwender zwei Möglichkeiten. Entweder sie klicken links oben auf den Button „Neue Abwesenheit“, oder man klickt auf den jeweiligen Tag und dabei noch die jeweilige Uhrzeit. Bei beiden Varianten öffnet sich daraufhin ein Fenster, wie in Abbildung 3.42 ersichtlich. Bei der letzteren Möglichkeit steht allerdings das Datum und die jeweilige Uhrzeit als Startzeitpunkt bereits im Formular. Selbstverständlich kann man diese im Nachhinein noch ändern. Abwesenheiten können einen Zeitraum von ein paar Stunden annehmen oder mehrere Tage. Urlaube, Kur usw. sind nur ganztägig möglich. Sollen auch andere Abwesenheiten ganztägig sein, kann die Benutzerin oder der Benutzer einfach einen Haken bei „ganztägig“ machen, damit entfällt die Eingabe bei der Uhrzeit. Liegt der Endzeitpunkt der Abwesenheit vor dem Startzeitpunkt, bekommt die Anwenderin oder der Anwender eine entsprechende Fehlermeldung, dass dieser Fall eingetreten ist, ebenfalls bei Kollisionen. Allgemein wurde bei der Applikation darauf geachtet, dass Benutzerinnen und Benutzer von vornherein möglichst wenig Fehler machen können, wie breites im Abschnitt 2.4.3 erläutert wurde. Ganz konnte es allerdings nicht vermieden werden. Sollten Benutzerinnen

und Benutzer dennoch Fehler machen, werden sie in einfachen, aussagekräftigen Meldung auf ihren Fehler hingewiesen. Durch Wegklicken der Fehlermeldung können die zuvor fehlerhaften Angaben ausgebessert werden. Im Formular besteht die Möglichkeit durch Betätigung des Buttons „Weitere beantragen“ mehrere Abwesenheiten direkt hintereinander zu beantragen. Hat eine Anwenderin oder ein Anwender mehrere Abwesenheitsanträge ausgefüllt, ist es ihr oder ihm möglich durch Klicken auf den Vor- bzw. Zurück-Button zwischen den einzelnen Anträgen zu wechseln. Dadurch kann sie oder er frühere Anträge noch editieren oder ganz löschen. Bei dieser Funktion wurde versucht der 8 Schnittstellendesignregel „Umkehr von Aktionen“, wie in Abschnitt 2.4.2 erklärt, umzusetzen. Werden alle Anträge bestätigt, bekommt die oder der Vorgesetzte ein E-Mail über alle Anträge.

Die Applikation kommuniziert direkt mit SAP. Das heißt, es hat selbst keine eigene Datenbank. Alle Daten werden von SAP geholt und dort wiederum hineingeschrieben. Falls nun eine neue Abwesenheit beantragt wurde, erscheint diese nicht sofort in der Applikation, da die Datenbank nicht neu geladen wurde. Ein neues Laden der Datenbank kann die Benutzerin oder der Benutzer mit einem Klick auf den „Daten aktualisieren“-Button selbst steuern. Wenn die Benutzerin oder der Benutzer Abwesenheiten beantragt oder genehmigt bzw. nicht genehmigt bekommt, werden die Daten nach Abschluss automatisch aktualisiert. Ansonsten bleiben die Daten unverändert.

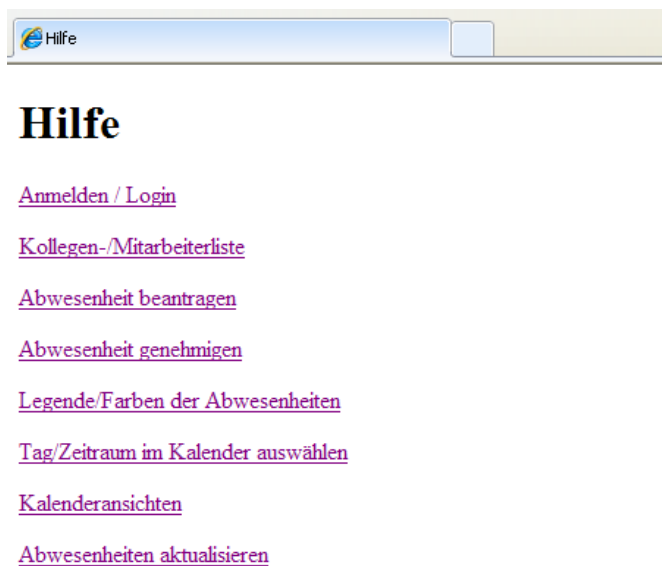
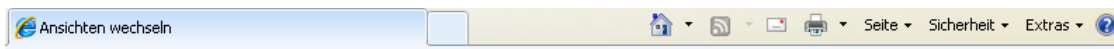


Abbildung 3.43: Hilfe - Übersicht

Zwar wurde versucht alle Funktionen der Applikation so klar darzustellen wie möglich, falls dies nicht oder nur schlecht gelungen ist und den Benutzerinnen und den Benutzern die Arbeit zu erleichtern, verfügt die Applikation über eine einfache Hilfe, wie in Abschnitt 2.4.3 beschrieben und wie in den Abbildungen 3.43 und 3.44 dargestellt, in der die Funktionen erklärt werden. Sie ist über den Link links oben erreichbar.



[zurück](#)

Ansichten wechseln

Sie können zwischen mehreren Kalenderansicht wechseln:



Durch eine Art "Karteikarten"-System rechts oben lassen sich zwischen Tagesansicht, Wochenansicht (Montag - Sonntag) und Wochentagsansicht (Montag - Freitag) wechseln. Zusätzlich werden in dem Bereich "Agenda" alle Abwesenheiten in dem zuvor gewählten Zeitraum angezeigt. Für Vorgesetzte wird unter "Offene Anträge" alle nicht genehmigte Abwesenheitsanträge ihrer Mitarbeiterinnen und Mitarbeiter angezeigt.

Abbildung 3.44: Hilfe - Ansicht wechseln

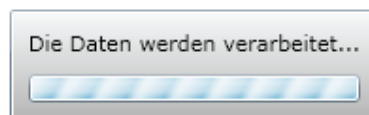


Abbildung 3.45: Balken um Wartezeit zu überbrücken

Bei allen Funktionen wurde ein Balken, wie in Abbildung 3.45, eingefügt, der die Wartezeit bis zum Ergebnis verkürzen soll, bzw. der Anwenderin und dem Anwender signalisieren soll, dass ihre oder sein Befehle verarbeitet werden. Damit bekommen Benutzerinnen und Benutzer ein Feedback von der Applikation, was in Abschnitt 2.4.2 erklärt wurde.

Nach der Fertigstellung des Protoyps wurde er von mehreren Probanden getestet, um zu erfahren, wie gut die Theorie in die Praxis umgesetzt wurde und die Funktionalität gelungen ist.

3.7 Usability-Test

Nach Abschluss der Programmierarbeiten wurde die Abwesenheitsplanungsapplikation getestet. Das Ziel war herauszufinden, inwieweit Benutzerinnen und Benutzer das Programm annehmen, wie sie mit den Funktionalitäten zurechtkommen und warum sie die Anweisungen der Abwesenheitsplanungsapplikation nicht Folge leisten können bzw. umgekehrt [Beier et al., 2002, S. 99].

Um die Abwesenheitsplanungsapplikation zu testen, wurden fünf Mitarbeiterinnen und Mitarbeiter des Unternehmens einzeln gebeten, eine Reihe von Aufgaben zu bewältigen. Die fünf Testperson mit unterschiedlichen Qualifikationen stammen aus verschiedenen Bereichen des Unternehmens. Drei der fünf Testpersonen waren Vorgesetzte. Drei Personen waren jüngerer Generationen, davon zwei Mitarbeiterinnen und Mitarbeiter und eine oder ein Vorgesetzter. Diese drei Personen, waren mit Rechnern allgemein besser vertraut und würden diese auch privat gebrauchen. Die anderen zwei Mitarbeiter waren Vorgesetzte älterer Generation und würden nur für berufliche Zwecke mit einem Rechner arbeiten. Sie würden für ihre Arbeit vertrauten Programme beherrschen, hatten aber außerhalb wenig Erfahrung mit Computer, Internet und dergleichen.

Zwar sollte, wie in Abschnitt 2.2 erklärt, verschwiegen werden, dass der Experimentator und der Entwickler dieselbe Person sind, da der Umfang dieser Arbeit eher gering ist, war das in der Praxis nicht durchzuführen. Alle Probanden wurden allerdings darauf hingewiesen, dass Kritik nicht persönlich genommen werden würde, da man aus Fehlern lernen wolle. Zudem wurde den Personen erklärt, dass sie auf Wunsch den Test jederzeit abbrechen könnten. Zwar werde das Testresultat im Rahmen dieser Arbeit veröffentlicht, allerdings die Namen der Probanden und des Unternehmens vertraulich behandelt. Da man testen möchte, wie die Testpersonen ohne fremde Hilfe mit dem System umgehen, wurde den Probanden erklärt, dass sie gerne Fragen stellen dürfen, die allerdings erst, soweit es geht, nach dem Test erläutert werden würden. Zudem wurden die Testperson gebeten ihre Gedanken während des Test auszudrücken.

Während des Tests selbst wurde eine Audioaufzeichnung mit Hilfe eines Diktiergeräts gemacht. Zusätzlich wurden händisch Notizen gemacht. Mit CamStudio, einer Open Source Software, wurden die Bewegungen am Bildschirm aufgezeichnet [CamStudio, 2011]. Zwar wurde auch geplant, die Testpersonen mittels einer Videokamera zu filmen, was allerdings alle Testpersonen, trotz der Versicherung dabei nicht die Gesichter aufzunehmen, ablehnten. Die Personen und das Unternehmen wollten anonym bleiben und je weniger Aufzeichnung es über ihre Person geben würde, umso angenehmer wurde es von ihnen empfunden.

Den Testpersonen wurde ein Laptop zur Verfügung gestellt, bei dem die Applikation mit

dem Login-Fenster geöffnet war. Als Testdaten wurden zwar reale Personen aus dem Unternehmen verwendet, allerdings nicht die Kolleginnen und Kollegen, Mitarbeiterinnen und Mitarbeiter, mit denen sie in der Wirklichkeit ihre Abwesenheitsplanung abstimmen müssen. Diese Konstellation wurde bewusst gewählt, denn man wollte die Funktionalität des Programm testen und vermeiden, dass die Testpersonen den Unterschied zwischen Kolleginnen und Kollegen und Mitarbeiterinnen und Mitarbeiter, schon anhand des Namens kennen und nicht aufgrund der unterschiedlichen Darstellung durch die Abwesenheitsapplikation. Es sollte ein neutrales Umfeld geschaffen werden.

Folgende Aufgaben hatten die Personen zu bewältigen:

1. Die Testpersonen mussten sich einloggen.

Drei der fünf Personen klickten nicht auf den Button „Anmelden“, sondern versuchten den Benutzernamen und das Passwort mit der Entertaste zu bestätigen. Diese Funktion wurde aus Gründen, die bereits weiter vorne im Abschnitt 3.6 erläutert wurden, nicht eingebaut. Sie zeigte sich allerdings bei dem Test wichtiger als ursprünglich gedacht. Offenbar war diese Funktion für die Testpersonen sehr geläufig und sie empfanden sie als selbstverständlich. Als die Betätigung der Entertaste nicht den gewünschten Erfolg versprach, wurde auf den „Anmelden“-Button geklickt. Für eine der Testpersonen war diese fehlende Funktion nach eigenen Aussagen regelrecht demotivierend für die folgenden Aufgabenstellungen. Die anderen Personen nahmen das Fehlen der Funktion der Entertaste einfach zur Kenntnis und probierten danach einfach mit dem „Anmelden“-Button zum gewünschten Ergebnis zu kommen. Viel mehr beschäftigte sie das Problem nicht.

Allerdings hat sich gezeigt, dass es sehr von Vorteil wäre, diese Funktion noch zu ergänzen. Zu erwähnen wäre, dass es sich bei den drei Testpersonen um Personen handelte, die mit dem Umgang eines Rechners sehr vertraut waren. Den beiden anderen Personen war das Fehler dieser Funktion von vornherein nicht aufgefallen.

2. Die Testpersonen sollten verschiedene Fragen beantworten

Nach dem erfolgreichen Einloggen, wurde es den Testpersonen ermöglicht sich umzusehen. Sie klickten sich durch den Kalender, begutachteten die verschiedenen Kalenderansichten und waren damit beschäftigt zu erkunden, wie die Applikation aufgebaut ist.

Danach wurden ihnen folgende Fragen gestellt:

- a) Mit wie vielen Personen müssen Sie Ihre Abwesenheiten abstimmen?
- b) Von wie vielen Mitarbeiterinnen und Mitarbeitern müssen Sie die Abwesenheiten genehmigen bzw. ablehnen?

Mit diesen Frage wollte man herausfinden, ob die Darstellung der Mitarbeiterinnen und Mitarbeiter verständlich ist. Im Falle, dass die Testperson eine Vorgesetzte oder ein Vorgesetzter ist, wurde zusätzlich die zweite Frage gestellt, um deren Verständnis für die Kennzeichnungen, die die Kolleginnen und Kollegen von den Mitarbeiterinnen und Mitarbeitern unterschieden, zu überprüfen.

Für die Testpersonen ohne Vorgesetztenfunktion stellt diese Frage keinerlei Probleme dar. Die Vorgesetzten benötigen allerdings einen Augenblick (wenige Sekunden)

des Nachdenkens/Sehens, um zu verstehen, welche Kennzeichnung welche Mitarbeiterin und welche Mitarbeiter darstellt.

Danach wurden mehrere Fragen über bereits existierende Abwesenheiten gestellt.

- c) Welche Mitarbeiterin, Kollegin oder welcher Mitarbeiter, Kollege hat zu bestimmten Zeitpunkten Abwesenheiten eingetragen und sind diese Abwesenheiten genehmigt oder nicht?

Die Probanden haben sehr schnell die jeweilige Abwesenheit im Kalender gefunden. Auch der Name der Person stellte überhaupt kein Problem dar. Probleme ergab die Beantwortung der Frage, ob die Abwesenheit beantragt oder auch schon genehmigt ist. Dies für die Testerinnen und Tester nicht auf Anhieb klar. Dass sie den Unterschied im Falle von Mitarbeiterinnen und Mitarbeiter an der Farbe erkennen oder in beiden Fällen (auch bei Kolleginnen und Kollegen) in Form eines Tooltips ersichtlich ist, benötigte einige Zeit. Die Probanden suchten mit der Maus über den Bildschirm und verweilten auf der jeweiligen Abwesenheit, sodass sich der Tooltip öffnete. Von vier Testpersonen wurde diese Art der Darstellung nicht kritisiert. Sie haben ihre Schwierigkeiten darauf zurückgeführt, dass sie dieses Problem das erste Mal lösen und sich erst umsehen mussten. Für die spätere Anwendung wäre ihnen die Vorgehensweise klar und sie würden sich auf Anhieb zurechtfinden.

Eine der Testpersonen, die keine Vorgesetzte bzw. Vorgesetzter ist, daher auch keine Unterscheidungsmöglichkeit durch Farben besitzt, kritisierte, dass die Möglichkeit nur über den Tooltip bestand. Daher sollte eventuell diese Information auch direkt bei der jeweiligen Abwesenheit gezeigt werden. Wobei allerdings zu beachten ist, dass das Wissen, ob eine Abwesenheit genehmigt ist oder nicht, für die Planung nicht sehr entscheidend ist, und wenn die Darstellung direkt bei der Abwesenheit erfolgt, dort eventuell zu viele Informationen vorhanden sind.

Nach diesen Verständnisfragen, sollten Aufgaben gelöst werden.

3. Die Testpersonen sollten einen mehrtägigen Urlaub beantragen.

Drei Personen klickten auf den „Neue Abwesenheit“-Button. Die Möglichkeit direkt im Bereich des jeweiligen Tages (zum Beispiel Anfangszeitpunkt) zu klicken, fanden sie nicht heraus. Die zwei anderen Personen klickten direkt in den jeweiligen Tag. Allerdings aus dem Grund, dass sie länger benötigten um sich zu orientieren, auch etwas den Eindruck der Überforderung machten und einfach im Programm „herumklickten“. Dadurch waren schon bei einer früheren Aufgabe auf dieses Formular unabsichtlich gestoßen und wussten bereits, dass man so eine neue Abwesenheit beantragen kann. Der „Neue Abwesenheit“-Button wurde von diesen Testpersonen gar nicht mehr beachtet.

Nachdem die anderen Testpersonen von dieser Möglichkeit der Abwesenheitsbeantragung erfahren hatten, wurde diese positiv aufgenommen, da bereits das Datum und die Uhrzeit eingetragen waren und die Probanden meinten, dass sie diesen auch in Zukunft benutzen würden, nachdem sie davon erfahren hätten.

Beim Ausfüllen hatten vier Testpersonen keine gravierenden Fehler gemacht. Zwei von ihnen beklagten, dass man das Datum nicht direkt eingeben konnte. Erst wenn man durch

den Button rechts vom Datumseingabefeld ein Datum ausgewählt hat, dann kann man dieses auch per Tastatureingabe ändern. Wenn man die Abwesenheit durch Klicken auf einen jeweiligen Tag beantragt, ist von vornherein bereits ein Datum ausgewählt. Dadurch besteht dieses Problem bei dieser Variante von vornherein nicht. Es tritt nur bei der Beantragung durch Betätigen des „Neue Abwesenheit“-Buttons auf. Es kann umgangen werden, wenn auch hier von vornherein ein Datum ausgewählt wird, zum Beispiel das des ersten Tages der zu dieser Zeit angezeigten Woche.

4. Eine beantragte Abwesenheit zu ändern, war ebenfalls eine Aufgabe.

Bei einer Testperson stellte sich diese Aufgabe als schwierig heraus. Hatten die anderen damit keine Probleme, da sie einfach auf gut Glück darauf klickten und sich dadurch das „Editier“-Formular öffnete, wusste die eine Testperson nicht wirklich, wie sie diese Aufgabe bewerkstelligen sollte. Sie suchte einen Button, leider erfolglos. Erst spät und eher durch einen zufälligen Klick auf die jeweilige Abwesenheit, war auch sie erfolgreich.

Eine Person hatte größere Probleme, das Datum auszuwählen beziehungsweise einzutippen. Sie wollte die Abwesenheit durch einen Klick auf das neue Datum ändern und wollte in diesem Formular das Datum ändern. Da öffnete sich allerdings das „Abwesenheit beantragen“-Formular. Als auch sie das Änderungsformular fand, funktionierte die Eingabe des End-Datums auf Anhieb. Beim Eintippen des Anfangsdatums stürzte das Programm ab. Das passierte insgesamt drei Mal nach erneutem Wiederholen. Nach Beendigung des Tests, wurde anhand von Notizen, des aufgezeichneten Videos und Erinnerungen versucht, diese Fehler nachzuvollziehen, leider ohne Erfolg. Auch durch Eintippen des gleichen Inhaltes funktionierte das Programm und der Fehler trat nicht mehr auf.

5. Die Testpersonen, die auch Vorgesetzte sind, mussten eine Abwesenheit genehmigen.

Alle fünf Personen suchten am Anfang diese Funktion. Vier von ihnen versuchten relativ schnell einfach auf die jeweilige Abwesenheit zu klicken und kamen dadurch auf die gewünschte Funktion. Gestört hat dabei, dass man keinen Kommentar schreiben konnte, man sah nur den Kommentar der Antragstellerin oder des Antragstellers, konnte aber selbst keinen verfassen.

Eine Person suchte sehr lange nach einem Button und fand die Funktion durch Klicken auf die jeweilige Abwesenheit für nicht zufriedenstellend und unlogisch. Er wollte einen Button für diese Funktion. Durch so einen Button stellte sich aber die Frage, wie die Anwenderin oder der Anwender eine Beziehung zwischen der jeweiligen Abwesenheit und der Genehmigung herstellen sollte. Das könnte damit gelöst werden, dass in einer ComboBox alle zu genehmigenden Abwesenheiten aufgelistet werden und die Benutzerin oder der Benutzer diese auswählen kann.

6. Nach den Aufgaben wurden die Probanden nach ihren Eindrücken, Wünschen, Kritiken gefragt.

- a) Eine Testperson kritisierte im Bereich „Agenda“, dass bei der Agenda nur die Abwesenheiten der im Kalender ausgewählten Woche angezeigt würden. Die Person würde sich einen größeren Bereich wünschen.

- b) Eine Testperson wünschte sich, dass die in der kleinen Monatsübersicht rechts, bei der man die Tage auswählen kann, alle Tage, in denen man selbst eine Abwesenheit beantragt oder auch schon genehmigt bekommen hat, diese in dieser Übersicht fett oder in irgendeiner anderen Form gekennzeichnet werden.
- c) Der Wunsch einer Person war es, dass ganztägige Abwesenheiten nicht direkt im Kalender eingetragen werden, sondern eventuell oben in einer anderen Form. Damit wäre es übersichtlicher.
- d) Der Zeitbalken soll der Firma, bzw. dem Arbeitnehmerin oder dem Arbeitnehmer, der seine Abwesenheiten einträgt, angepasst werden können. Man soll zum Beispiel in einen Bereich von 20 Uhr bis 7 Uhr nicht einblicken können.
- e) Hilfsfunktion: Während allen Aufgaben stand die Hilfsfunktion des Programmes zur Verfügung, allerdings wurde sie von keiner Person in Anspruch genommen.

Resultat

Im Großen und Ganzen waren die Testpersonen zufrieden. Einige der Probleme, die im Laufe des Tests aufgetreten sind, waren darauf zurückzuführen, dass sie das Programm zum ersten Mal verwendetet haben und daher die Suche einige Zeit in Anspruch genommen hat. So war ihnen der Tooltip erst nach längerer Suche aufgefallen. Die Testpersonen meinten, dass sie sich bei weiteren Anwendungen der Abwesenheitsapplikation schneller zurechtfinden würden:

Dennoch sollten folgende Funktionen noch ergänzt werden:

- Eine Jahresansicht fehlt beim Prototypen noch und muss hinzugefügt werden.
- Im Loginbereich soll der Anmeldebutton auch durch Drücken der Entertaste betätigt werden können, damit Anwenderinnen und Anwender mit entsprechenden Kenntnissen nicht bestraft werden, wie schon im Abschnitt 2.4.3 aufgeführt.
- Funktionen sollen auch mittels Shortcuts ausgeübt werden können. Gerade wenn Programme öfter benutzt werden, stellen diese, wie im Abschnitt 2.4.2 erklärt, eine Erleichterung für die Benutzerinnen und Benutzer dar.
- In der Kalenderübersicht, bei der die einzelnen Tage ausgewählt werden können, sollen die Tage, in denen die Benutzerin oder der Benutzer Abwesenheit eingetragen hat, fett gekennzeichnet sind. Damit stechen diese mehr hervor und die Anwenderin und der Anwender weiß auf einen Blick, an welchen Tagen sie oder er Abwesenheiten beantragt hat.
- Genehmigung der Abwesenheiten soll durch einen eigenen Button ermöglicht werden und nicht nur die durch Klicken auf die jeweilige beantragte Abwesenheit.
- Bei Genehmigung und Ablehnungen von Abwesenheiten sollen Vorgesetzte Kommentare verfassen können.
- Der Bereich für die „Agenda“ soll nicht nur die Abwesenheiten des ausgewählten Bereichs beinhalten, sondern einen längeren Ausschnitt umfassen. Wobei dabei zu hinterfragen ist, wie groß der Ausschnitt gewählt werden soll, damit er nicht zu unübersichtlich wird.

- Bei der Beantragung einer neuen Abwesenheit durch den Button „Neue Abwesenheit“ soll der Abwesenheitsanfangszeitpunkt automatisch in den Endzeitpunkt übernommen werden. Dadurch wird das Auswählen des Endzeitpunkts erleichtert, da dieser meistens in der Nähe des Beginnzeitpunktes liegt.
- Vorgesetzte sollten auch für ihre Mitarbeiterinnen und Mitarbeiter Abwesenheiten eintragen können, zum Beispiel in Falle eines Krankenstandes. Beim Prototypen fehlt diese Funktion noch.
- Der Zeitbalken soll der Arbeitszeit der Anwenderin oder des Anwenders entsprechen. Damit wäre der Kalender übersichtlicher, da unnötige Daten, die die Benutzerin oder den Benutzer gar nicht betreffen, wegfallen würden.

Kapitel 4

Conclusio

In dieser Arbeit wurde eine Applikation für Abwesenheitsplanung entwickelt, die eine Verbindung zu SAP integriert hat. Die Suche nach vergleichbaren Produkten zeigte, dass es solche Programme zwar auf dem Markt gibt, diese allerdings zusätzliche Anforderungen benötigen. Entweder besitzen sie eine eigene Datenbank, die in regelmäßigen zeitlichen Abständen mit dem SAP-System abgeglichen wird oder sie verfügen über ein eigenes SAP-System und der Datenaustausch erfolgt zwischen den SAP-Systemen. Für manche Applikationen werden auch zusätzliche Programme benötigt, bzw. die Abwesenheitsplanung ist nur ein Teil des Funktionsumfangs. Eine einfache Abwesenheitsplanungsapplikation, bei der Personen ihre Abwesenheiten beantragen, ändern, verschieben, löschen, genehmigen und ablehnen können und dabei direkt die Datenbank von SAP verwenden, wurde bei der Recherche nicht gefunden.

Zu Beginn wurden in einem Unternehmen Interviews durchgeführt. Die Personen wurden über die aktuelle Situation, über Probleme und Wünsche befragt. Bei dieser Befragung zeigte sich, dass die befragten Personen mit der aktuellen Situation unzufrieden sind. Sie würden sich ein einfaches Programm wünschen, mit dem Abwesenheiten beantragt, genehmigt bzw. abgelehnt werden können. Zusätzlich sollen die Daten in ihr SAP-System übertragen werden.

Bei der Softwareentwicklung besteht die Gefahr, dass die Bedürfnisse und Wünsche der Benutzerinnen und Benutzer zu wenig beachtet werden [Grechenig et al., 2010, S. 505]. Daher wurde für die Entwicklung User Centered Design verwendet. User Centered Design hebt die Gefühle und Bedürfnisse der Benutzerinnen und Benutzer hervor [Wei und Xing, 2010]. Es ist wichtig, die Personen bzw. Personengruppen zu kennen, die später das Abwesenheitsplanungsprogramm benutzen [DIN EN ISO 9241-210, 2011, S. 9fff]. Aus den Interviews ist hervorgegangen, dass das Programm zwar regelmäßig, aber mit größeren Zeitabständen verwendet wird. Mitarbeiterinnen und Mitarbeiter, die keine Führungsaufgabe besitzen, verwenden das Programm nur selten, da sie nur Abwesenheiten beantragen. Eine Ausnahme bilden die Vorgesetzten, insbesondere die mit mehreren Mitarbeiterinnen und Mitarbeitern, die täglich oder zumindest alle paar Tage überprüfen müssen, ob es Anträge gibt. Das Wissen der Anwenderinnen und Anwender ist eine wichtige Quelle für die zukünftige Benutzung des Produktes. Daher wurden sie aktiv in der Entwicklung miteinbezogen. Deren Rückmeldungen sind zum Verfeinern und Anpassen miteingeflossen. Bis das gewünschte Ergebnis erreicht war, wurde eine Folge von Schritten immer wiederholt. Damit wurden bestehende Unsicherheiten schrittweise entfernt.

Für die Arbeit wurde Literaturrecherche zu den Themenbereichen, die mit Abwesenheitsplanung zusammenhängen, durchgeführt. Einer dieser Themenbereiche war CSCW. Da mit Hilfe der Abwesenheitsplanungapplikation mehrere Personen ihre Abwesenheit planen und alle diese Vorgänge mit Hilfe eines Computers abgewickelt werden sollen, wurde dieses Forschungsgebiet näher behandelt. CSCW ist ein Oberbegriff, für die Art und Weise, wie Personen in Gruppen mit den Technologien von Computer-Netzwerken arbeiten und wie die dazugehörige Hardware, Software und Technik kombiniert wird [Wilson, 1991, S. 1]. Auch der Bereich der Groupware wurde behandelt, da er die Technologie zum Forschungsgebiet CSCW darstellt [Grudin, 2011] und die Abwesenheitsplanungapplikation dessen Eigenschaften besitzen soll.

Für die Abwesenheitplanung waren nicht nur die Anforderungen an die Applikation wichtig und wie die Personen mittels eines Computers zusammenarbeiten, sondern die Planung selbst. Zu Beginn wurde überlegt, wie das Layout gestaltet werden soll. Dazu wurde der Ansatz von Alan Cooper verwendet und Skizzen auf Papier angefertigt, da das ein gutes Werkzeug darstellt, da es die Diskussion fördert und leicht zu ändern ist [Cooper et al., 2007, S. 133]. Danach wurden diese Layout-Skizzen mit drei Mitarbeiterinnen und Mitarbeitern besprochen. Dabei kam es zu ein paar Veränderungen, die geändert bzw. ergänzt worden sind.

Bei der Visualisierung hat man sich an die Goldenen Regeln des Schnittstellendesigns orientiert [Shneiderman, 2002, S. 99f]. Es wurde darauf geachtet, dass Layout, Farb- und Schriftauswahl, Hilfestellungen, Menüs, Eingabeaufforderungen etc. möglichst durchgehend konsistent gestaltet sind. Bei jeder Handlung bekommen Benutzerinnen und Benutzer ein Feedback. Falls es zu einem Fehler kommt, enthält die Fehlermeldung den Grund des Fehlers, und bei der Beantragung von mehreren Abwesenheiten hintereinander, besteht die Möglichkeit der Umkehr.

Im Rahmen dieser Arbeit wurde ebenfalls die rechtliche Perspektive der Abwesenheiten behandelt. So sind die Abwesenheiten von Mitarbeiterinnen und Mitarbeitern im Arbeitsrecht geregelt. Im Beispielunternehmen werden zwischen folgenden Abwesenheiten unterschieden: Urlaub, Krankenstand, Krankenpflegefreistellung, Betreuungsfreistellung, Mutterschutz, Karenz, Bildungskarenz, Schulung und Zeitausgleich. Da diese Daten vertraulich sein können, wurde auf den Datenschutz näher eingegangen. Man erlangte die Kenntnis, dass Mitarbeiterinnen und Mitarbeitern Abwesenheitsgründe ihrer Kolleginnen und Kollegen verschwiegen werden sollten.

Neben dem rechtlichen Bereich wurde auf technische Aspekte der Abwesenheitsplanungapplikation näher eingegangen. Da das Programm auf einzelnen Rechnern laufen soll, während die Datenbank vom SAP-System auf einem ausgelagerten Server zur Verfügung gestellt wird, handelt es sich dabei um ein verteiltes System. Um von diesem Rechner auf das SAP-System zuzugreifen, gibt es mehrere Möglichkeiten, unter anderem Web Service, die bei der Implementierung benutzt wurden.

Im praktischen Teil wurden Programme mit ähnlicher Funktion vorgestellt und näher auf die eigentliche Entwicklung eingegangen. Bei der Implementierung wurden als Programmiersprache Microsoft Silverlight und C# ausgewählt. Bei Silverlight sind das Design und die damit verbundene Logik getrennt. Das Design wird in Silverlight in Extensible Application Markup Language (XAML) erstellt. XAML initialisiert Objekte und deren Eigenschaften [xam, 2011]. Als Grundlage für die Abwesenheitsplanung wurde der Silverlight Scheduler von CodePlex (Open Source Project Community) verwendet, dessen Code freiverfügbar ist [Codeplex, 2010]. Seine Funktion umfasst das Verwalten von Events. Für die Abwesenheitsplanungapplikation mussten

im Layout einige Dinge geändert werden. Die Verbindung des Silverlight-Programms mit SAP stellte allerdings die größte Herausforderung dar. In Funktionsbausteinen werden die eigentlichen Aufgaben erfüllt, die man von dem SAP-System benötigt, die sich über Web Service in die Abwesenheitsapplikation einbauen lassen, wobei die domainenübergreifenden Richtlinien ein Problem darstellten. Durch lange Internetrecherche und viele Versuche wurde dieses Problem gelöst und in dieser Arbeit vorgestellt.

Dadurch, dass die Daten direkt von einem ERP-System gelesen bzw. wieder hineingeschrieben werden, sind sowohl in der Abwesenheitsplanungsapplikation als auch im ERP-System immer die aktuellen Daten vorhanden. Eine eigene Datenbank wird daher nicht benötigt. Die Systemvoraussetzungen wurden auf das Minimalste beschränkt. Tabellen, die das Programm benötigt, aber im SAP-System für den benötigten Zweck nicht vorhanden waren, wurden dort neu erstellt.

Nach der Fertigstellung wurden an mehreren Personen der Prototyp getestet. Dabei stellte sich heraus, dass die Anforderungen im Großen und Ganzen erfüllt waren. Im Bereich der Usability kann es aber noch zu Verbesserungen kommen. Zudem sollten noch einige zusätzliche Funktionen implementiert werden.

Bei dieser Arbeit hat sich herausgestellt, dass die Anwenderinnen und Anwender vor allem auf die grundlegenden Funktionen einer Abwesenheitsapplikation Wert legen. Je mehr Funktionen eine Applikation besitzt, umso unübersichtlicher erscheint es den Benutzerinnen und Benutzern. Werden die Funktionen auf das Notwendigste beschränkt und ist mit diesen die Aufgabenstellung der Applikation effizient lösbar, sind sie zufriedener und verzichten gerne auf zusätzlichen Funktionsumfang.

Mit dem Programm wurde ermöglicht Abwesenheiten zu beantragen, zu verschieben, zu löschen, zu genehmigen oder abzulehnen. Diese Funktionen wurden so einfach wie möglich gestaltet um den Benutzerinnen und Benutzern auch eine nur sporadischer Anwendung leicht zu ermöglichen. Die Abwesenheitsplanung kann dadurch effizienter umgesetzt werden und führt zu einer Zeitersparnis. Durch die regelmäßige Einbindung der späteren Anwenderinnen und Anwender in den Entwicklungsprozess wurden Fehler schon früh beseitigt und Aufwendungen, die sich später als Fehlentscheidungen herausstellten, vermieden. Das SAP-System konnte als Datenbank dienen. Dadurch wurde es möglich, redundanten Daten zu verhindern und dennoch stets aktuelle Daten bereitzustellen.

4.1 Ausblick

Nach der Fertigstellung des Programms wurde es mit Userinnen und Usern getestet. Die Abwesenheitsplanungsapplikation sollte noch in einigen Bereichen verbessert werden. Im Großen und Ganzen waren die Testpersonen mit dem Programm zufrieden und meinten, dass wenn sie einmal herausgefunden hätten, wie die einzelnen Funktionen aufzurufen bzw. zu finden sind, die Benutzung für weitere Verwendungen kein Problem wäre. So hatten sie Schwierigkeiten herauszufinden, ob eine Abwesenheit beantragt war oder nicht. Da sie sich erst kurz im Programm umgesehen hatten, war ihnen die unterschiedliche Kennzeichnung durch Farben nicht aufgefallen und der Tooltip ist ihnen nach etwas längerer Suche aufgefallen. Nachdem sie die Funktionen

allerdings herausgefunden hätten, würde es für weitere Verwendungen keine Probleme mehr darstellen. Folgende Funktionen sollen aber noch ergänzt werden

- Eine Jahresansicht muss hinzugefügt werden.
- Der Anmeldebutton im Loginbereich soll auch durch Drücken der Entertaste betätigt werden.
- Funktionen sollen auch mittels Shortcuts ausgeübt werden können.
- In der Kalenderübersicht, der Übersicht bei dem der einzelnen Tage ausgewählt werden können, sollen die Tage, an denen die Benutzerin oder der Benutzer Abwesenheit eingetragen hat, fett gekennzeichnet sein.
- Genehmigung der Abwesenheiten soll nicht nur durch Klicken auf die jeweilige Abwesenheit durchgeführt werden, sondern auch durch einen eigenen Button.
- Bei Genehmigung und Ablehnungen sollen Kommentare verfasst werden können.
- Der Bereich für die „Agenda“ soll nicht nur die Abwesenheiten des ausgewählten Bereichs beinhalten, sondern einen längeren Ausschnitt umfassen.
- Wird durch den Button „Neue Abwesenheit“ eine neue Abwesenheit beantragt und dabei der Anfangszeitpunkt der Abwesenheit gewählt, soll dieser automatisch in den Abwesenheitsendzeitpunkt übernommen werden. Dadurch wird das Auswählen des Endzeitpunktes erleichtert, da dieser meistens in der Nähe des Beginnzeitpunktes liegt.
- Vorgesetzte sollten auch für ihre Mitarbeiterinnen und Mitarbeiter Abwesenheiten eintragen können, zum Beispiel in Falle eines Krankenstands.
- Der Zeitbalken soll der Arbeitszeit der Anwenderin oder des Anwenders entsprechen.

Anhang A

SAP-Web Services in Silverlight/C#

Will man SAP-Web Services in ein Silverlight/C# einbinden, sind diese nicht sofort funktionsfähig, da die domänenübergreifenden Richtlinien fehlen. Wie in MSDN Library beschrieben, muss der Dienst, der in einer anderen Domain liegt, explizit ausgewählt werden, damit der domainübergreifende Zugriff aktiviert wird. Ein Dienst bestätigt dabei, dass die Vorgänge, die er verfügbar macht, ohne Bedenken der Sicherheit von einem Silverlight 4-Steuerungselement aufgerufen werden kann und dadurch keine Daten beschädigt werden [Ric, 2011]. Dafür soll man ein `clientaccesspolicy.xml`-File im Stammverzeichnis der Domäne stellen. Das ist leider bei einem SAP-System nicht möglich. Durch lange Internetrecherche und viele Versuche wurde im Bereich SE80 (Object Navigator) ein eigenes Programm entwickelt, dessen Code hier veröffentlicht wird.

```
REPORT ZCROSSDOMAIN.
"-----
"Create the Crossdomain.xml
"-----
DATA crossdomainxml_t type string.
DATA crossdomainxml   type xstring.
CONCATENATE
'<?xml version="1.0" encoding="UTF-8" ?>'
'<cross-domain-policy xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance"
'xsi:noNamespaceSchemaLocation="http://
www.adobe.com/xml/schemas/PolicyFile.xsd">'
'<allow-access-from domain="*" /> '
'<site-control permitted-cross-domain-policies="
master-only" />'
'<allow-http-request-headers-from domain="*" headers="*"
secure="true" />'
'</cross-domain-policy>'
into crossdomainxml_t SEPARATED BY SPACE.
```

```

"convert to binary

CONCATENATE'<!DOCTYPE cross-domain-policy SYSTEM "http://
www.macromedia.com/xml/dtds/cross-domain-policy.dtd">'
'<cross-domain-policy>'
'  <allow-http-request-headers-from domain="*"
headers="*" />'
'</cross-domain-policy>'

      '<access-policy>'
      '<cross-domain-access>'
      '<policy>'
      '<allow-from http-request-headers="*">'
      '<domain uri="*" />'
      '</allow-from>'
      '<grant-to>'
      '  <resource path="/" include-subpaths=
      "true" />'
      '</grant-to>'
      '</policy>'
    '</cross-domain-access>'
  '</access-policy>' into crossdomainxml_t SEPARATED BY SPACE.

CONCATENATE
'<?xml version="1.0"?>'
'<!DOCTYPE cross-domain-policy SYSTEM "http://www.adobe.com/xml/
dtds/cross-domain-policy.dtd">'
'<cross-domain-policy>'
'<site-control permitted-cross-domain-policies="all" />'
'<allow-access-from domain="*" secure="false" />'
'<allow-http-request-headers-from domain="*" headers="*"
secure="false" />'
'</cross-domain-policy>'
INTO crossdomainxml_t SEPARATED BY SPACE.

CALL FUNCTION 'SCMS_STRING_TO_XSTRING'
  EXPORTING
    TEXT          = crossdomainxml_t
  IMPORTING
    BUFFER        = crossdomainxml
.
"-----
" Create Response object with the crossdomain data

```

```
"-----  
data lr_cached_response type ref to if_http_response.  
  
create object lr_cached_response type cl_http_response  
exporting add_c_msg = 1.  
lr_cached_response->set_data( crossdomainxml ).  
lr_cached_response->set_content_type( 'text/xml' ).  
lr_cached_response->set_status( code = 200 reason = 'OK' ).  
"set the validity for more than a day(86400)  
lr_cached_response->server_cache_expire_rel( expires_rel  
= 90000 ).  
  
"-----  
"Upload to the ICM root directory  
"-----  
CALL METHOD CL_HTTP_SERVER=>SERVER_CACHE_UPLOAD  
EXPORTING  
    URL      = '/crossdomain.xml'  
*    URL      = '/ClientAccessPolicy.xml'  
    RESPONSE = lr_cached_response  
    SCOPE    = IHTTP_INV_GLOBAL
```

Abbildungsverzeichnis

2.1	Verschiedene Arten von Awareness [Böhringer, 2009, S. 27].	9
2.2	Darstellung des Designs mit der Benutzerin oder des Benutzers im Mittelpunkt [Noyes und Baber, 1999, S. xi].	11
2.3	Ein Modell der Eigenschaften von „system acceptability“ [Nielson, 1993, S. 25].	22
2.4	Synchroner Remote Procedure Call [Tanenbaum und van Steen, 2003, S. 101]	35
2.5	Asynchroner Remote Procedure Call [Tanenbaum und van Steen, 2003, S. 101]	35
2.6	Synchrone Kommunikation	36
2.7	Asynchrone Kommunikation	37
3.1	UrlaubsManager - Abwesenheitsverwaltung	44
3.2	UrlaubsManager -Keine Firma/Abteilung ausgewählt	45
3.3	UrlaubsManager - Posteingang	45
3.4	UrlaubsManger - Nachricht	46
3.5	UrlaubsManager - Abwesenheitsbeantragung	47
3.6	UrlaubsMangager - Kalender	48
3.7	gecotime - workflow [Gec, 2011]	49
3.8	HR-Starter®ASP - Zugriff über gesicherte Netzwerkverbindung [ins, 2011]	50
3.9	Out of Office [Software Solutions Simple, 2010]	50
3.10	Anwendungsfalldiagramm der Abwesenheitsplanung	52
3.11	Use Case der Beantragung einer Abwesenheit	53
3.12	Use Case für die Änderung einer Abwesenheit	53
3.13	Use Case für die Löschung einer Abwesenheit	53
3.14	Use Case für die Genehmigung bzw. Ablehnung einer Abwesenheit	54
3.15	Abwesenheit beantragen	55
3.16	Abwesenheitsänderung	56
3.17	Abwesenheitsgenehmigung	57
3.18	UML-Diagramm	58
3.19	Anmeldefenster	59
3.20	Standardansicht der Mitarbeiterinnen und Mitarbeiter	60
3.21	Jahresansicht	61
3.22	Wochenansicht	61
3.23	Tagesansicht	62
3.24	Abwesenheit beantragen	63

3.25	Nachricht über Entscheidung der Abwesenheit	63
3.26	Standardansicht der oder des Vorgesetzten	64
3.27	Antrag einer Mitarbeiterin oder eines Mitarbeiters	65
3.28	Liste alle offenen Anträge	65
3.29	Silverlight Scheduler	66
3.30	Silverlight Scheduler - Agenda	67
3.31	Silverlight Scheduler - Tagesansicht	68
3.32	Silverlight Scheduler - Wochentageansicht	69
3.33	Beispiel eines Funktionsbausteins in SAP	70
3.34	Web Service einer Funktionsgruppe	71
3.35	Pfad des XML-Files des Web Service	71
3.36	Anmelden	72
3.37	Hauptbereich	73
3.38	Agenda	74
3.39	Offene Anträge	75
3.40	Abwesenheiten genehmigen	76
3.41	Mitarbeiterliste	76
3.42	Abwesenheit beantragen	77
3.43	Hilfe - Übersicht	78
3.44	Hilfe - Ansicht wechseln	79
3.45	Balken um Wartezeit zu überbrücken	79

Literaturverzeichnis

- [ins, 2011] (01.02.2011). inside GmbH, Unternehmensberatung Oldenburg - Personalwirtschaft, Personalzeitwirtschaft, SAP. Website. <http://www.inside-gmbh.de/Losungen/HR-Starter/ASP.aspx>.
- [Gec, 2011] (01.02.2011). Zeiterfassung und Zutrittskontrolle in Österreich. Website. <http://www.gecosoft.at>.
- [AKK, 2011a] (02.05.2011a). Krankenstand - AK - Portal. Website. <http://www.arbeiterkammer.at/online/krankenstand-45257.html>.
- [AKK, 2011b] (02.05.2011b). Kuraufenthalt - AK - Portal. Website. <http://www.arbeiterkammer.at/online/kuraufenthalt-928.html>.
- [Add, 2010] (02.07.2010). AddWare Software. Website. <http://www.addware.de>.
- [WKO, 2010a] (02.07.2010a). Wirtschaftskammer Österreich. Website. http://portal.wko.at/wk/format_detail.wk?AngID=1\&StID=421055\&DstID=0\&titel=Urlaubsanspruch.
- [xam, 2011] (05.04.2011). Xaml overview. Website. [http://msdn.microsoft.com/en-us/library/cc189036\(v=vs.95\).aspx](http://msdn.microsoft.com/en-us/library/cc189036(v=vs.95).aspx).
- [Arb, 2010] (05.07.2010). Arbeiterkammer. Website. <http://www.arbeiterkammer.at/online/page.php?P=92\&IP=45415\&AD=917\&REFP=0\#faq45415>.
- [WKO, 2010b] (05.07.2010b). Wirtschaftskammer Österreich. Website. http://portal.wko.at/wk/format_detail.wk?AngID=1\&StID=424183\&DstID=7228\&titel=Pflegefreistellung.
- [WKO, 2010c] (05.07.2010c). Wirtschaftskammer Österreich. Website. http://portal.wko.at/wk/format_detail.wk?AngID=1\&StID=423314\&DstID=\&titel=Persönliche,Dienstverhinderungsgründe.
- [SAP, 2011d] (07.02.2011d). Internet Communication Framework (SAP-Bibliothek - Komponenten der SAP-Kommunikationstechnologie). Website. http://help.sap.com/saphelp_nw04/helpdata/de/36/020d3a0154b909e10000000a114084/content.htm.

- [SAP, 2011f] (07.02.2011f). RFC (SAP Library - Components of SAP Communication Technology). Website. http://help.sap.com/saphelp_nw04/helpdata/en/6f/1bd5b6a85b11d6b28500508b5d5211/content.htm.
- [SAP, 2011g] (07.02.2011g). SAP-Bibliothek – ALE/EDI-Geschäftsprozesse. Website. http://help.sap.com/saphelp_sm32/helpdata/de/c9/d97749ae0311d1a5120060087832f8/frameset.htm.
- [SAP, 2011i] (07.02.2011i). SAP Business Connector (SAP-Bibliothek - Komponenten der SAP-Kommunikationstechnologie). Website. http://help.sap.com/saphelp_nw04/helpdata/de/6f/1bd5c2a85b11d6b28500508b5d5211/content.htm.
- [SAP, 2011j] (07.02.2011j). SAP .NET (SAP-Bibliothek - SAP .NET Connector). Website. http://help.sap.com/saphelp_nw04/helpdata/de/e9/23c80d66d08c4c8c044a3ea11ca90f/content.htm.
- [SAP, 2011m] (07.02.2011m). SOAP Framework (SAP-Bibliothek - Komponenten der SAP-Kommunikationstechnologie). Website. http://help.sap.com/saphelp_nw04/Helpdata/de/bb/ddb33d2ae46b3be10000000a114084/content.htm.
- [Sil, 2011] (08.03.2011). Get Started: The Official Microsoft Silverlight Site. Website. <http://www.silverlight.net/getstarted/>.
- [Ric, 2011] (08.03.2011). Warnung: Domänenübergreifende Richtlinien Unter '<URL>' nicht gefunden. Website. [http://msdn.microsoft.com/de-de/library/cc838225\(v=vs.95\).aspx](http://msdn.microsoft.com/de-de/library/cc838225(v=vs.95).aspx).
- [Wei, 2011] (13.06.2011). Csr - arbeitsrechtliche aspekte - wko.at. Website. http://portal.wko.at/wk/format_detail.wk?angid=1&stid=434295&dstid=8683&titel=CSR%2C-%2Carbeitsrechtliche%2CAspekte.
- [SAP, 2011h] (14.02.2011h). SAP-Bibliothek - ABAP Workbench: Werkzeuge. Website. http://help.sap.com/saphelp_nw04/Helpdata/de/9b/dad1ae3908ee44a5caf57e10918be9/frameset.htm.
- [SOA, 2011] (14.02.2011). Simple Object Access Protocol (SOAP) 1.1. Website. <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>.
- [MSW, 2011] (14.02.2011). Web Services Standards: SOAP, UDDI and WSDL. Website. <http://msdn.microsoft.com/de-de/library/bb979447.aspx>.
- [SAP, 2011b] (14.06.2011b). Der landesbeauftragte für datenschutz und informationsfreiheit bremen. Website. http://www.datenschutz-bremen.de/technik/datenschutz_sap.php.
- [RIS, 2011] (16.02.2011). RIS - Gesamte Rechtsvorschrift für Datenschutzgesetz 2000. Website. <http://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=bundesnormen&Gesetzesnummer=10001597&ShowPrintPreview=True>.

- [AK, 2010] (24.08.2010). Ak.portal: Karenz-regelung. Website.
<http://www.arbeiterkammer.at/online/page.php?P=28&IP=2086>.
- [ARG, 2010] (24.08.2010). Ris - gesamte rechtsvorschrift für arbeitsruhe-gesetz. Website. [http://www.ris.bka.gv.at/GeltendeFassung.wxe? Abfrage=Bundesnormen&Gesetzesnummer=10008541](http://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10008541).
- [AZG, 2010] (24.08.2010). Ris - gesamte rechtsvorschrift für arbeitszeit-gesetz. Website. <http://www.ris.bka.gv.at/GeltendeFassung.wxe? Abfrage=Bundesnormen&Gesetzesnummer=10008238>.
- [SAP, 2010a] (25.07.2010a). SAP-Bibliothek: Komponenten der SAP-Kommunikationstechnologie. Website. http://help.sap.com/saphelp_nw2004s/helpdata/de/22/0424fe488911d189490000e829fbbd/frameset.htm.
- [SAP, 2010b] (25.07.2010b). SAP-Bibliothek: Komponenten der SAP-Kommunikationstechnologie. Website. http://help.sap.com/saphelp_nw2004s/helpdata/de/22/0424fe488911d189490000e829fbbd/frameset.htm.
- [AK2, 2011] (27.04.2011). Ak - urlaubsrecht. Website.
http://www.arbeiterkammer.at/bilder/d36/Urlaubsrecht_Sept05.pdf.
- [SAP, 2011a] (27.04.2011a). ALE/EDI (SAP-Bibliothek). Website.
http://help.sap.com/saphelp_nw04/helpdata/de/6f/1bd5bea85b11d6b28500508b5d5211/content.htm.
- [Bil, 2011] (27.04.2011). Help.gv.at - arbeit - bildungskarenz. Website.
<http://www.help.gv.at/Content.Node/k17/Seite.171800.html>.
- [SAP, 2011c] (27.04.2011c). IDoc-Schnittstelle (SAP-Bibliothek). Website.
http://help.sap.com/saphelp_nw04/helpdata/de/6f/1bd5baa85b11d6b28500508b5d5211/content.htm.
- [AK3, 2011] (27.04.2011). Mutterschutz - regelung. Website.
<http://www.arbeiterkammer.at/online/mutterschutz-regelung-2009.html>.
- [SAP, 2011e] (27.04.2011e). Remote Function Call (RFC) (SAP-Bibliothek - SAP-Kommunikation: CPI-C-Programmierung (BC-CST-GW)). Website.
http://help.sap.com/saphelp_wp/helpdata/de/b7/2eb39c54f411d188850000e83539c3/content.htm.
- [SAP, 2011j] (27.04.2011j). SAP Java Connector - SAP-Dokumentation. Website.
http://help.sap.com/saphelp_nw73/helpdata/de/48/70792c872c1b5ae10000000a42189c/content.htm.

- [SAP, 2011k] (27.04.2011k). SAP Java Resource Adapter (SAP-Bibliothek - Komponenten der SAP-Kommunikationstechnologie. Website. http://help.sap.com/saphelp/_nwmobile71/helpdata/de/6f/1bd5caa85b11d6b28500508b5d5211/content.htm.
- [SAP, 2011n] (27.04.2011n). Transaktionale RFC-Aufrufe (SAP Bibliothek - RFC-Programmierung in ABAP. Website. http://help.sap.com/saphelp/_46c/helpdata/de/22/042578488911d189490000e829fbbd/content.htm.
- [SAP, 2010c] (30.06.2010c). SAP Deutschland - SAP ERP HCM. Website. <http://www.sap.com/germany/solutions/business-suite/erp/hcm/featuresfunctions/index.epx>.
- [Abrams et al., 2004] Abrams, C., Maloney-Krichmar, D., und Preece, J. (2004). User-centered design. W. *Encyclopedia of Human-Computer Interaction*, pages 1–14.
- [Balmer et al., 2000] Balmer, R., Inversini, S., von Planta Annina, und Semmer, N. (2000). *Innovation im Unternehmen - Leitfaden zur Selbstbewertung für KMU*. vdf Hochschulverlag AG an der ETH Zürich.
- [Beier et al., 2002] Beier, M., von Gizycki, V., Fröhlich, M., und Mühlig, J. (2002). *Usability - Nutzerfreundliches Web-Design*. Springer Berlin Heidelberg.
- [Böhringer, 2009] Böhringer, M. (2009). *Enterprise Microblogging - Grundlagen, Einsatzpotenziale und Konzept zur Unterstützung der wissensbasierten Projektkommunikation*. Edition Wissensmanagement.
- [Brown, 1999] Brown, Carol und Vessey, I. (1999). ERP Implementation Approaches : Toward A Contingency Framework. *Framework*, pages 411–416.
- [Burger, 1997] Burger, C. (1997). *Groupware Kooperationsunterstützung für verteilte Anwendungen*. dpunkt.verlag.
- [Bürger, 1999] Bürger, M. (1999). *Unterstützung von Awareness bei der Gruppenarbeit mit gemeinsamen Arbeitsbereichen*. Herbert Uzt Verlag GmbH.
- [CamStudio, 2011] CamStudio (18.06.2011). CamStudio - Free Screen Recording Software. Website. <http://camstudio.org/>.
- [Chlebek, 2006] Chlebek, P. (2006). *User Interface-orientierte Softwarearchitektur*. Friedr.Vieweg & Sohn Verlag.
- [Codeplex, 2010] Codeplex (10.08.2010). Silverlight Scheduler. Website. <http://silverlightscheduler.codeplex.com/>.
- [Cooper et al., 2007] Cooper, A., Reimann, R., und Corin, D. (2007). *About Face 3 The Essentials of Interaction Design*. Wiley Publishing, Inc.

- [Cooper et al., 2004] Cooper, A., Reimann, R., und Cronin, D. (2004). *About Face: Interface und Interaction Design*. mitp.
- [Crowcroft und Markus, 2001] Crowcroft, J. und Markus, H. (2001). *Networked Group Communication - Third International COST264 Workshop, NGC 2001 London, UK, November 2001, Proceedings*. Springer-Verlag Berlin Heidelberg New York.
- [Dasgupta, 2006] Dasgupta, S. (2006). *Encyclopedia of Virtual Communities and Technologies*. Idea Group Reference.
- [DIN EN ISO 9241-11, 199] DIN EN ISO 9241-11 (199). Ergonomische anforderungen für bürotätigkeiten – teil 11: Anforderungen an die gebrauchstauglichkeit-leitsätze (iso 9241-11:1999).
- [DIN EN ISO 9241-210, 2011] DIN EN ISO 9241-210 (2011). Ergonomie der mensch-system-interaktion – teil 210: Prozess zur gestaltung gebrauchstauglicher interaktiver systeme (iso 9241-210:2010).
- [Dourish und Bellotti, 1992] Dourish, P. und Bellotti, V. (1992). Awareness and coordination in shared workspaces. *Proceedings of the 1992 ACM conference on Computer-supported cooperative work - CSCW 92*, pages 107–114.
- [Ellis et al., 1991] Ellis, C., Gibbs, S., und G.L., R. (1991). Groupware – Some Issues and Experiences. *Communications of the ACM*, 34(january):38–58.
- [Far, 2010] Far, S. M. (2010). *Social Software in Unternehmen - Nutzenpotentiale und Adoption in der innerbetrieblichen Zusammenarbeit*. Josef Eul Verlag GmbH.
- [Felix und Rosenbusch, 2011] Felix, D. und Rosenbusch, A. (2011). User-centered design in sieben punkten kurz erklärt. *ICT-Jahrbuch 2011*, pages 155–156.
- [Ferscha, 2001] Ferscha, A. (2001). Integrating Pervasive Information Acquisition to Enhance Workspace Awareness. *Proceedings Ninth Euromicro Workshop on Parallel and Distributed Processing*, pages 327–336.
- [Formann und Kalliauer, 2010] Formann, D. und Kalliauer, J. (2010). *Beschäftigte fühlen sich überwacht : AK fordert besseren Datenschutz in Betrieben*. Pressekonferenz Mittwoch, 12. Mai 2010, um 11 Uhr in der Arbeiterkammer Linz.
- [Fouss und Chang, 2000] Fouss, J. D. und Chang, K. H. (2000). Classifying groupware. *Proceedings of the 38th annual on Southeast regional conference - ACM-SE 38*, pages 117 – 124.
- [Fowler, 2004] Fowler, M. (2004). *UML konzentriert - Eine kompakte Einführung in die Standard-Objektmodellierungssprache*. Addison-Wesley.
- [Frevert, 2011] Frevert, J. (2011). Nächste halt: Ihr protal! *SAP Spectrum*, pages 54–58.

- [Friedman, 2009] Friedman, V. (2009). *Praxisbuch Web 2.0*. Galileo Press.
- [Gadatsch, 2001] Gadatsch, A. (2001). *Management von Geschäftsprozessen (Methoden und Werkzeuge für die IT-Praxis: Eine Einführung für Studenten und Praktiker)*. Vieweg.
- [Gärtner et al., 2007] Gärtner, J., Klein, C., und Lutz, D. (2007). *Arbeitszeitmodelle*. LexisNexis Verlag ARD Orac GmbH & CO KG, Wien.
- [Giordano et al., 2005] Giordano, M., Hummel, J., und Wiener, C. (2005). *Mobile Buisness Vom Geschäftsmodell zum Geschäftserfolg - Mit Fallbeispielen zu Mobile Marketing, mobilen Protalen und Content-Anbieter*. Betriebswirtschaftlicher Verlag Dr.Th.Gabler/GWV Fachverlage GmbH.
- [Gomis, 2007] Gomis, M.-J. (2007). Web-based ERP Systems : The new generation Case Study : mySAP ERP. *Master Procject 2007, Computer Engineering*.
- [Grechenig et al., 2010] Grechenig, T., Bernhart, M., Breiteneder, R., und Kappel, K. (2010). *Softwaretechnik - Mit Fallbeispielen aus realen Entwicklungsprojekten*. Pearson Studium.
- [Gross et al., 2007] Gross, T., Koch, M., und Herczeg, M. (2007). *Computer-Supported Cooperative Work*. Oldenbourg Wissenschaftsverlag GmbH.
- [Grudin, 2011] Grudin, J. (11.06.2011). CSCW: History and Focus. Website. <http://research.microsoft.com/en-us/um/people/jgrudin/past/papers/ieee94/ieeecomplastsub.html>.
- [Gutwin, 1997] Gutwin, C. (1997). *Workspace Awareness in Real-Time Distributed Groupware*. PhD thesis, University of Calgary.
- [Gutwin et al., 1996a] Gutwin, C., Greenberg, S., und Roseman, M. (1996a). Workspace awareness in real-time distributed grouware: Framework, widgets, and evaluation. pages 1–24.
- [Gutwin et al., 1996b] Gutwin, C., Roseman, M., und Greenberg, S. (1996b). A usability study of awareness widgets in a shared workspace groupware system. In *Proceedings of the 1996 ACM conference on Computer supported cooperative work, CSCW '96*, pages 258–267, New York, NY, USA. ACM.
- [Hansen, 1971] Hansen, W. J. (1971). User engineering principles for interactive systems. *Fall Joint Computer Conference, 1971*, pages 523–532.
- [Hartung, 2006] Hartung, J. (2006). *Sozialpsychologie*. W. Kohlhammer GmbH Stuttgart.
- [Hellige, 2008] Hellige, H. D. (2008). *Mensch-Computer-Interface - Zur Geschichte und Zukunft*. Gabler Edition Wissenschaft.
- [Helm, 2007] Helm, S. (2007). *Unternehmensreputation und Stakeholder-Loyalität*. Deutscher Universitäts-Verlag.

- [Herczeg, 2005] Herczeg, M. (2005). *Softwareergonomie*. Oldenbourg Wissenschaftsverlag GmbH.
- [Hitz und Kappel, 2003] Hitz, M. und Kappel, G. (2003). *UML@Work*. dpunkt.verlag GmbH.
- [Holtzblatt et al., 2005] Holtzblatt, K., Wendell, J. B., und Shelley, W. (2005). *Rapid Contextual Design - A How-To Guide to Key Techniques für User-Centered Design*. Oldenbourg Wissenschaftsverlag GmbH.
- [Holzinger, 2005] Holzinger, A. (2005). Usability Engineering Methods for Software Developers. *Communications of the ACM*, 48(1):71–74.
- [Hosni und Khalil, 2004] Hosni, Y. A. und Khalil, T. (2004). *Management of Technology: Internet Economy: Opportunities and Challenges for Developed and Developing Regions of the World*. Elsevier Ltd.
- [Ivory und Hearst, 2001] Ivory, M. Y. und Hearst, M. A. (2001). The State of the Art in Automating Usability Evaluation of User Interfaces. *ACM Computing Surveys*, 33(4):470–516.
- [Jacko, 2009] Jacko, J. A. (2009). *Human-Computer Interaction - Ambient, Ubiquitous and Intelligent Interaction*. Springer-Verlag.
- [Jeske, 2005] Jeske, T. (2005). *SAP für Java-Entwickler - Konzepte, Schnittstellen, Technologien*. Springer Berlin Heidelberg.
- [Johnson, 2000a] Johnson, J. (2000a). *GUI Bloopers - Don'ts and Do's for Software Developers and Web Designers*. Morgan Kaufmann Publishers.
- [Johnson, 2000b] Johnson, J. (2000b). Textual Bloopers An excerpt form GUI Bloopers. *The Stock Market*, pages 28 – 48.
- [Jokela, 2002] Jokela, T. (2002). Making User-Centred Design Common Sense: Striving for An Unambiguous and Communicative UCD Process Model. *ACM*, pages 19–26.
- [Joonhwan et al., 2006] Joonhwan, K., Sanghee, L., und SungWoo, K. (2006). Understanding Users in Consumer Electronics Experience Design. *ACM 1-59593-298-4/06/0004 - CHI 2006 - Experience Report - Creative User Experience Methods*, pages 189–194.
- [Kamel und Davison, 1998] Kamel, N. N. und Davison, R. M. (1998). Applying csw technology to overcome traditional barriers in group interactions. *Inf. Manage.*, 34:209–219.
- [Keller und Krüger, 2001] Keller, H. und Krüger, S. (2001). *ABAP Objects - Einführung in die SAP-Programmierung*. Galileo Press.
- [Kittl, 2009] Kittl, C. (2009). *Kundenakzeptanz und Geschäftsrelevanz: Erfolgsfaktoren für Geschäftsmodelle in der digitalen Wirtschaft*. Gabler Edition Wissenschaft.
- [Klinke, 2008] Klinke, M. (2008). *Groupware in Kleinunternehmen - Eine empirische Untersuchung*. Diplomica Verlag GmbH.

- [Kogent Learning Solutions, 2010] Kogent Learning Solutions, I. (2010). *SAP ABAP - Questions and Answers*. Jones and Bartlett Publishers, LLC.
- [Konrad, 2003] Konrad, W. (2003). *Grundlagen eines guten User Interface Designs*. GRIN Verlag.
- [Krämer et al., 2003] Krämer, C., Lübke, C., und Ringling, S. (2003). *mySAP HR Personalwirtschaft*. SAP PRESS.
- [Kristiansen, 2005] Kristiansen, R. (2005). *Tailoring of ERP user interfaces using an model-based approach*. PhD thesis, Norwegian University of Science and Technology.
- [Larman, 2005] Larman, C. (2005). *UML 2 und Patterns angewendet - Objektorientierte Softwareentwicklung*. mitp-verlag.
- [Liang et al., 1996] Liang, H., Zhou, C., Ma, F., und Wu, Q. (1996). A CSCW-Based Organizational Decision Support System. *Proceedings of the IEEE International Conference on Industrial Technology (ICIT'96)*, pages 152–156.
- [Liebhart et al., 2007] Liebhart, D., Schmutz, G., Lattmann, M., Heinisch, M., Könings, M., Kölliker, M., Pakull, P., und Welkenbach, P. (2007). *Architecture Blueprints - Ein Leitfaden zur Konstruktion von Softwaresystemen mit Java Spring, .Net, ADF, Forms und SOA*. Calr Hansen Verlag.
- [Lin et al., 2006] Lin, J., Song, C., und Wang, H. (2006). Application of Business Rule in CSCW System. *2006 10th International Conference on Computer Supported Cooperative Work in Design*, pages 1–5.
- [Lubich, 1995] Lubich, H. P. (1995). *Towards a CSCW Framework for Scientific Cooperation in Europe*. Springer-Verlag.
- [Macaulay und Busse, 2009] Macaulay, C. und Busse, D. (2009). Using User Research in Creativity - Informing Systems , Service and Product Experience Design. *ACM 978-1-60558-403-4/09/10, C&C'09 October 26-30,2009, Berkeley, Clifornia, USA*, pages 495–496.
- [Mcgrenerere und Moore, 2000] Mcgrenerere, J. und Moore, G. (2000). Are we all in the same „bloat“? *Distribution*, pages 1–10.
- [Moos, 2004] Moos, A. (2004). *Datenbank- Engineering - Analyse, Entwurf und Implementierung objektrelationaler Datenbanken*. Friedr. Vieweg&Sohn Verlag.
- [Niegemann et al., 2008] Niegemann, H. M., Domagk, S., Hessel, S., Hein, A., Hupfer, M., und Zobel, A. (2008). *Kompendium mulitmediales Lernen*. Springer-Verlag Berlin Heidelberg.
- [Nielson, 1993] Nielson, J. (1993). *Usability Engineering*. Academic Press.
- [Norman, 1993] Norman, D. A. (1993). *Things That Make Us Smart*. Basic Books.

- [Noyes und Baber, 1999] Noyes, J. und Baber, C. (1999). *User-Centred Design of Systems*. Springer-Verlag Berlin Heidelberg New York.
- [Otjacques et al., 2006] Otjacques, B., Noirhomme, M., Gobert, X., und Feltz, F. (2006). Cooperation indexes to support workspace awareness. *Y.A. Dimitriadis et al.: CRIWG 2006, LNCS 4254, pp.*, pages 94–101.
- [Raab-Steiner und Benesch, 2008] Raab-Steiner, E. und Benesch, M. (2008). *Der Fragebogen - Von der Forschungsidee zur SPSS-Auswertung*. Facultas Verlags- und Buchhandels AG.
- [Rajagopal, 2002] Rajagopal, P. (2002). An innovation - diffusion view of implementation of enterprise resource planning (ERP) systems and development of a research model. *Information & Management*, 40:87–114.
- [Richter und Markus, 2010] Richter, M. und Markus, F. (2010). *Usability Engineering kompakt - Benutzbare Software gezielt entwickeln*. Spektrum Akademischer Verlag Heidelberg.
- [Riemer et al., 2005] Riemer, K., Arendt, P., und Wulf, A. (2005). *Marktstudie Kooperations-systeme - Von E-Mail über Groupware zur Echtzeitkooperation*. Cuvillier Verlag.
- [Schiestl und Schelle, 2010] Schiestl, J. und Schelle, H. (2010). *Groupware - Software für die Teamarbeit der Zukunft. Grundlegende Konzepte und Fallstudien*. Tectum Verlag.
- [Schmidt, 2002] Schmidt, K. (2002). The problem with 'awareness': Introductory remarks on 'awareness in cscw'. *Comput. Supported Coop. Work*, 11:285–298.
- [Schmitz, 2007] Schmitz, R. (2007). *Kompodium Medieninformatik: Medienpraxis, Band 2*. Springer-Verlag.
- [Schwabe et al., 2001] Schwabe, G., Streitz, N., und Unland, R. (2001). *CSCW - Kompodium*. Springer-Verlag.
- [Shangguang und Meilin, 1996] Shangguang, W. und Meilin, S. (1996). Support Environment For CSCW Research -Design And Implementation Of A Desktop Computer Conferencing System. *Communication Technology Proceedings, 1996. ICCT'96.*, 1:187–190.
- [Shirani et al., 1999] Shirani, A. I., H.A., T. M., und Affisco, J. F. (1999). Task and technology fit: a comparison of two technologies for synchronous and asynchronous group communication. *Information & Management*, 36(3):139–150.
- [Shneiderman, 2002] Shneiderman, B. (2002). *User Interface Design*. mitp-Verlag, Bonn.
- [Shneiderman et al., 1997] Shneiderman, B., Byrd, D., und Croft, W. B. (1997). Clarifying search - a user-interface framework for text searches. *D-Lib Magazine*, pages 1–17.
- [Siedersleben und Krug, 2003] Siedersleben, J. und Krug, W. (2003). *Softwaretechnik - Praxiswissen für Softwareingenieure*. Carl Hansen Verlag.

- [Software Solutions Simple, 2010] Software Solutions Simple (12.03.2010). Zeitmanagement par excellence. Website. http://www.softsimple.de/pageID_3082656.html.
- [Stephan, 2009] Stephan, K. (2009). *Grundkurs Software-Engineering mit UML - Der pragmatische Weg zur erfolgreichen Softwareprojekten*. Vieweg+Teubner.
- [Tanenbaum und van Steen, 2003] Tanenbaum, A. und van Steen, M. (2003). *Verteilte Systeme – Grundlagen und Pradigmen*. Pearson Studium.
- [Tanenbaum und van Renesse, 1986] Tanenbaum, A. S. und van Renesse, R. (1986). Distributed Operating Systems. *Computing Survey*, 17(4):419–470.
- [ten Hompel et al., 2008] ten Hompel, M., Büchter, H., und Franzke, U. (2008). *Identifikationssysteme und Automatisierung*. Springer-Verlag Berlin Heidelberg.
- [Thaller, 2002] Thaller, G. E. (2002). *Interface Design Die Mensch-Maschine-Schnittstelle gestalten*. Software & Support Verlag GmbH.
- [Thissen, 2003] Thissen, F. (2003). *Kompendium Screen-Design Effektiv informieren und kommunizieren mit Multimedia*. Springer-Verlag.
- [Tollmar et al., 1996] Tollmar, K., Sandor, O., und Schömer, A. (1996). Supporting social awareness @ work design and experience. *Proceedings of the 1996 ACM conference on Computer supported cooperative work - CSCW '96*, pages 298–307.
- [Universität der Bundeswehr München, 2011] Universität der Bundeswehr München (18.07.2011). FocusNimbusModel. Website. <http://twiki.informatik.unibw-muenchen.de/Main/FocusNimbusModel>.
- [van der Veer und van Welie, 2000] van der Veer, G. und van Welie, M. (2000). Task based groupware design: putting theory into practice. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, DIS '00, pages 326–337, New York, NY, USA. ACM.
- [Verhoeven und van Germert-Pijnen, 2010] Verhoeven, F. und van Germert-Pijnen, J. (2010). Discount user-centered e-health deisgn: A quick-but-not-dirty method. *HCI in Work and Learning, Life and Leisure - 6th Symposium of the Workgroup, Human-Computer Interaction and Usability Engineering, USAB 2010, Klagenfurt, Austria, November 2010, Proceedings*, pages 101–111.
- [Warfel, 2009] Warfel, T. Z. (2009). *Prototyping - A Practitioner's Guide*. Rosenfeld Media, LLC.
- [Wei und Xing, 2010] Wei, C. und Xing, F. (2010). The comparison of user-centered design and goal-directed design. *2010 IEEE 11th International Conference on*, 1:359 – 360.

- [Weißenberg und Stemmer, 2009] Weißenberg, N. und Stemmer, M. (2009). Moderne IT-Plattformen für Geschäftsprozess-Management und Portale Vergleichende Bewertung der Toolsuiten von IBM, IDS Scheer & SAP, sowie Intalio & Liferay Kurzfassung. http://www.isst.fraunhofer.de/Images/IBM-Studie-Kurzfassung_tcm81-23024.pdf.
- [Wilson, 1991] Wilson, P. (1991). *Computer Supported Cooperative Work*. British Library Cataloguing in Publication Data.
- [Yong, 2009] Yong, L. T. (2009). Collaborative awareness for translation groupware. 2009 *International Conference on Information and Multimedia Technology*, pages 47–51.