

Hochschule für Technik und Wirtschaft Dresden (FH)
Fachbereich Informatik/Mathematik

Diplomarbeit

im Studiengang Medieninformatik

Konzeption, Entwicklung und Usability Evaluation einer Webanwendung
für die Verwaltung von Webhosting Leistungen

Melanie Reichel

Betreuer:

Prof. Dr. phil. Teresa Merino

Prof. Dr. rer. nat. Ralph Großmann

28. November 2006

Vorwort

Ich möchte mich bei allen Personen bedanken, die mich bei der Bearbeitung meines Diplomthemas unterstützt haben. Dazu zählen in erster Linie die Kunden von Netfarm, Bekannten und Mitstudenten, die mir bei den Usability Evaluationen als Gutachter oder Testnutzer zur Verfügung standen. Außerdem möchte ich mich bei meinen Freunden und meiner Familie dafür danken, dass sie mich während meines gesamten Studiums und besonders während der Diplomzeit motiviert haben. Letztlich danke ich meinen Betreuern Frau Prof. Merino und Herrn Prof. Großmann für ihre hilfreichen Ratschläge.

Inhaltsverzeichnis

Vorwort	i
Inhaltsverzeichnis	ii
Abkürzungsverzeichnis	v
Abbildungsverzeichnis	vi
Tabellenverzeichnis	vii
Listenverzeichnis	vii
Einleitung	1
1 Software-Ergonomie – theoretische Grundlagen und praktischer Einsatz	2
1.1 Begriffe	2
1.1.1 Ergonomie	2
1.1.2 Software-Ergonomie	2
1.2 Gegenstand und Ziele der Software-Ergonomie	3
1.2.1 Gegenstand der Software-Ergonomie	3
1.2.2 Ziele der Software-Ergonomie	5
1.3 Menschliche Informationsverarbeitung als Grundlage der Software-Ergonomie	6
1.3.1 Die Architektur der menschlichen Kognition	7
1.3.2 Gedächtnis und mentale Modelle	9
1.4 Benutzerzentrierter Entwicklungsprozess – Usability Engineering	10
1.4.1 Usability Engineering Lifecycle nach Mayhew und Nielsen	11
1.4.2 Planung und Integration des Usability Engineering Prozesses	15
2 Web Usability	16
2.1 Begriffe im Kontext des WWW	16
2.1.1 Usability	16
2.1.2 Effektivität	16
2.1.3 Effizienz	16
2.1.4 Zufriedenheit	17
2.2 Gründe für Web Usability	18
2.2.1 Vielfalt des Webangebots	18
2.2.2 Das Web als Allzweckwerkzeug	18
2.2.3 Webseiten sind gratis	18
2.2.4 Wohlwollen und Vertrauen	19
2.3 Verhalten der Nutzer im Web	19
2.3.1 Information Foraging	19
2.3.2 Drei Fakten über das Nutzerverhalten im Web nach Krug	21
2.4 Allgemeine Web Usability Richtlinien	22
2.4.1 Informationsarchitektur und Navigation	22
2.4.2 Information	23
2.4.3 Faktoren, die Usability Richtlinien beeinflussen	24
3 Methoden der Usability Evaluation	26
3.1 Begriffe und Klassifizierung	26
3.1.1 Usability Evaluation	26
3.1.2 Kriterien der empirischen Forschung	27
3.1.3 Klassifikation der Methoden	28

3.2	Expertenorientierte Methoden	28
3.2.1	Allgemeine Aspekte	28
3.2.2	Heuristische Evaluation	30
3.2.3	Cognitive Walkthrough	33
3.3	Nutzerorientierte Methoden	38
3.3.1	User Testing (Nutzertest)	38
3.3.2	Focus Group (Fokusgruppe)	47
3.4	Diskussion der vorgestellten Methoden	51
3.4.1	Expertenorientierte Methoden gegen nutzerorientierte Methoden	51
3.4.2	Traditionelles Usability Engineering gegen Discount Usability Engineering	52
3.5	Auswertung einer Usability Evaluation	53
3.5.1	Bewertung der Ernsthaftigkeit der Probleme (Severity Ratings)	53
3.5.2	Präsentation der Ergebnisse	54
4	Konzeption, Gestaltung und Heuristische Evaluation der Webhosting Webanwendung	56
4.1	Konzeption	56
4.1.1	Funktionsumfang Kundenbereich	56
4.1.2	Funktionsumfang Administratorbereich	57
4.1.3	Informationsarchitektur und Navigationsstruktur	57
4.1.4	Systemvoraussetzungen	59
4.1.5	Datenbankkonzeption	59
4.2	Gestaltung der Benutzeroberfläche	60
4.2.1	Bildschirmerteilung	60
4.2.2	Navigation	61
4.2.3	Interaktionselemente	62
4.2.4	Schriftformat	63
4.2.5	Farben	63
4.3	Heuristische Evaluation	64
4.3.1	Planung der Untersuchung	65
4.3.2	Durchführung der Untersuchung	67
4.3.3	Auswertung der Ergebnisse	69
4.3.4	Beurteilung der Untersuchungsmethode	70
5	Implementierung eines Prototyps der Webhosting Webanwendung	72
5.1	Konzepte und Techniken	73
5.1.1	MVC (Model-View-Controller)	73
5.1.2	ORM (Object-Relational Mapping)	73
5.1.3	YAML (YAML Ain't Markup Language)	74
5.1.4	YAML (Yet Another Multicolumn Layout)	74
5.1.5	AJAX (Asynchronous JavaScript and XML)	74
5.2	Logische Struktur der Webanwendung als Symfony Projekt	75
5.3	Datenmodell und Datenzugriff (Model)	77
5.4	Aktionen und Komponenten (Controller)	79
5.4.1	Aktionen	79
5.4.2	Komponenten	81
5.5	Layout, Templates und AJAX (View)	82
5.5.1	Layout	82
5.5.2	Templates und AJAX	83
5.6	Konfiguration des Projekts	84

5.6.1	Projektweite Konfiguration – Datenbankverbindung	84
5.6.2	Applikationsweite Konfiguration – Menü und Routing	84
5.6.3	Modulweite Konfiguration – Validierung und Sicherheit	85
5.6.4	Sicherheit	86
6	Nutzertests am Prototyp der Webhosting Webanwendung	87
6.1	Planung der Nutzertests	87
6.1.1	Testnutzer und Testbedingungen	87
6.1.2	Aufgaben	88
6.1.3	Interviewfragen	90
6.2	Durchführung der Nutzertests	91
6.2.1	Einführung und Eingangsinterview	91
6.2.2	Bearbeitung der Aufgaben und Abschlussinterview	92
6.3	Auswertung der Ergebnisse	93
6.3.1	Eingangsinterview – Erfahrungen der Nutzergruppen	93
6.3.2	Ergebnisse Kundenbereich	94
6.3.3	Ergebnisse Administratorenbereich	96
6.3.4	Maßnahmen zur Verbesserung der Usability	98
6.3.5	Bedeutung der Ergebnisse	100
	Schlusswort	101
	Literaturverzeichnis	103
	Glossar	106
	Anhang	109
	Thesen	123
	Selbstständigkeitserklärung	124

Abkürzungsverzeichnis

<i>Abkürzung</i>	<i>Bedeutung</i>
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface (Schnittstelle zur Anwendungsprogrammierung)
CIF	Common Industry Format
CSS	Cascading Style Sheets
DIN	Deutsches Institut für Normung e. V.
DOM	Document Object Model
dpi	dots per Inch (Punkte pro Zoll)
DSL	Digital Subscriber Line
GUI	Graphical User Interface (grafische Benutzeroberfläche)
HCI	Human-Computer-Interaction
HTML	Hypertext Markup Language
HTTPS	Hypertext Transfer Protocol Secure
IEA	International Ergonomics Association
ISO	Internationale Organisation für Normung
LAMP	Linux-Apache-MySQL-PHP (kombinierten Einsatz von Programmen und Betriebssystemen, um einen Webserver zur Verfügung zu stellen)
MCI	Mensch-Computer-Interaktion
MVC	Model-View-Controller
ORM	Object-Relational Mapping
PHP	PHP: Hypertext Preprocessor (Programmiersprache)
pt	typographische Punkt
px	Pixel (Bildpunkt)
SQL	Structured Query Language (deklarative Datenbanksprache)
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language (erweiterbare Auszeichnungs-Sprache)
YAML	1. YAML Ain't Markup Language 2. Yet Another Multicolumn Layout

Abbildungsverzeichnis

Abbildung 1: Die Architektur der menschlichen Kognition nach [Glaser 1994, S. 8]	8
Abbildung 2: Das Star-Life-Cycle-Modell nach Hix und Hartson [vgl. Schweibenz & Thissen 2003, S. 51]	11
Abbildung 3: Kategorien von Design Problemen, die Nutzer an ihren Aufgaben im Web scheitern lassen nach [Nielsen & Loranger 2006, S. 133]	22
Abbildung 4: Zusammenhang zwischen der Anzahl der Prüfer und der Menge an gefundenen Usability Problemen (in Prozent) (Quelle: [Nielsen ca. 1994])	30
Abbildung 5: Zusammenhang zwischen der Anzahl der Nutzer und der Menge an gefundenen Usability Problemen (in Prozent) (Quelle: [Nielsen 2000])	40
Abbildung 6: Triangulation von Usability Problemen nach Dumas & Redish 1994 [vgl. Schweibenz & Thissen 2003, S. 151]	46
Abbildung 7: Zweite Navigationsebene im Kundenbereich für den Menüpunkt <i>E-Mail</i>	58
Abbildung 8: Zweite Navigationsebene im Administratorbereich für den Menüpunkt <i>Mailboxen</i>	58
Abbildung 9: Schema der Tabellen <i>dbmail_users</i> und <i>dbmail_aliases</i> der Maildatenbank	59
Abbildung 10: Entwurf der Startseite des Kundenbereichs mit Bereichsaufteilung	60
Abbildung 11: Entwurf der Startseite des Administratorbereichs mit Bereichsaufteilung	61
Abbildung 12: Entwurf des Menüs im Kundenbereich (aktiver Menüpunkt E-Mail Übersicht)	61
Abbildung 13: Entwurf des Menüs im Administratorbereich (erste und zweite Navigationsebene)	62
Abbildung 14: Suchbutton im Entwurf (links) und in der HTML-Umsetzung (rechts)	62
Abbildung 15: Die beiden charakteristischen Blautöne der Webanwendung und die Hintergrundfarbe	64
Abbildung 16: Statusfarbe Grün und Fehlerfarbe Rot als Schriftfarbe verwendet	64
Abbildung 17: Entwurf aus dem Kundenbereich mit Fehlermeldungen	65
Abbildung 18: Ebenen des Object-Relational Mappings	73
Abbildung 19: Synchrone Interaktion einer traditionellen Webanwendung im Vergleich mit dem asynchronen Muster einer AJAX Applikation (Quelle: [Garrett 2005])	75
Abbildung 20: Projektstruktur der Webanwendung in der Entwicklungsumgebung	76
Abbildung 21: Vereinfachte Darstellung des Zusammenhangs der Elemente des Views für den Kundenbereich	82
Abbildung 22: Schematische Darstellung der Template Organisation für die Seite zum Bearbeiten eines E-Mail-Kontos im Kundenbereich (<code>updateindexSuccess.php</code>)	83
Abbildung 23: Screenshot zum Usability Problem Leistungen (Kundenbereich)	94
Abbildung 24: Screenshot zum Usability Problem E-Mail-Adresse anlegen (Kundenbereich)	94
Abbildung 25: Screenshot zum Usability Problem E-Mail-Konten bearbeiten (Kundenbereich)	95
Abbildung 26: Screenshot zum Usability Problem Statusmeldung (Kundenbereich)	95
Abbildung 27: Screenshot zum Usability Problem Löschen von E-Mail-Adressen (Kundenbereich)	95
Abbildung 28: Screenshot zum Usability Problem Alle anzeigen (Administratorbereich)	97
Abbildung 29: Screenshot zum Usability Problem Löschen von Aliases (Administratorbereich)	97

Tabellenverzeichnis

Tabelle 1 Vergleich von empirischen und Inspektionsmethoden nach [Heuer 2003, S. 117]	51
Tabelle 2: Informationskategorien und Menüpunkte im Kundenbereich	58
Tabelle 3: Informationskategorien und Menüpunkte im Administratorbereich	58
Tabelle 4: Auszug aus der Auflistung der gefundenen Usability Probleme durch die Experten	68
Tabelle 5: Stufen der Ernsthaftigkeit eines Usability Problems (nach [Schweibenz & Thissen 2003, S. 114]) ..	69
Tabelle 6: Schwerwiegende Usability Probleme mit Lösungsvorschlag der Experten	69
Tabelle 7: Umgesetzte Funktionalität im Prototyp der Webanwendung	72
Tabelle 8: Aufgaben des Nutzertests mit der Testgruppe Kunden und dessen Zweck	88
Tabelle 9: Aufgaben des Nutzertests mit der Testgruppe Administratoren und dessen Zweck	89
Tabelle 10: Fragen nach dem Aufgabenteil im Nutzertest	91
Tabelle 11: Schwerwiegende und katastrophale Usability Probleme im Kundenbereich	94
Tabelle 12: Schwerwiegende und katastrophale Usability Probleme im Administratorenbereich	96
Tabelle 13: Usability Probleme mit Lösungsvorschlägen der Experten und Bewertungsstufe der Ernsthaftigkeit (gerundeter Mittelwert)	112
Tabelle 14: Ergebnisse, die als Vorschläge der Experten eingeordnet wurden, mit Bewertungsstufe (gerundeter Mittelwert)	116
Tabelle 15: Lösungen, die im Prototyp umgesetzt wurden, um Usability Probleme (Problem Nr.) zu beheben.	116
Tabelle 16: Usability Probleme im Nutzertest Kunden mit Bewertung durch den Versuchsleiter	118
Tabelle 17: Usability Probleme im Nutzertest Administratoren mit Bewertung durch den Versuchsleiter ____	120

Listenverzeichnis

Liste 1: Die Phasen des Usability Engineering Lifecycle nach [Mayhew 1999]	12
Liste 2: Die Phasen des Usability Engineering Lifecycle nach [Nielsen 1993, S. 72]	14
Liste 3: Überblick über den Cognitive Walkthrough Prozess nach [Wharton et al 1994, S. 106]	35
Liste 4: Aufgaben des Moderators in den Phasen der Fokusgruppensitzung	50
Liste 5: Kategorien der Heuristiken mit inhaltlicher Kurzcharakteristik und Beispiel	66
Liste 6: Beispiel einer Problembeschreibung durch einen Gutachter	68
Liste 7: Verzeichnisstruktur, die nach dem Ausführen der init-Befehle generiert wurde	77
Liste 8: Fragen im Kunden- und Administratorentest vor den Aufgaben	90

Einleitung

Das Web als Werkzeug bietet dem Benutzer nicht nur eine Vielzahl von Webseiten zum Finden von Informationen, sondern immer mehr Verwaltungsaufgaben können online erledigt werden. Überzeugen kann eine solche Anwendung, wenn der Nutzer sich schnell zurechtfindet und einfach seine Ziele erreichen kann. Usability (Gebrauchstauglichkeit) ist im Web ein entscheidendes Merkmal für Webseiten, um sich in der Vielfalt behaupten zu können. Auch Webanwendungen müssen dem Nutzer hohe Usability bieten, um als brauchbares Werkzeug bestehen zu können.

Usability bildet ein Maß dafür, wie effektiv, effizient und zufrieden der Nutzer seine Ziele mit dem Produkt erreichen kann. Ein hohes Maß an Usability kann nur erreicht werden, wenn Richtlinien, die durch die Software-Ergonomie entwickelt werden, im Entwicklungsprozess beachtet werden und regelmäßig der Erfolg der Maßnahmen bewertet wird. Die Usability Evaluation ist der Schwerpunkt jedes benutzerzentrierten Entwicklungsprozesses (Usability Engineering). Aufgrund der Bedeutung der Usability Evaluation stellen ihre Methoden einen Schwerpunkt dieser Diplomarbeit dar.

Um das Thema Usability in Webanwendungen zu verdeutlichen, wird ein Prototyp für eine Webanwendung entwickelt, mit der die Verwaltung von Webhosting Leistungen möglich sein soll. Dabei werden für die beiden Zielgruppen Kunden und Administratoren getrennte Bereiche konzipiert, die den unterschiedlichen Bedürfnissen gerecht werden sollen. Diese Anwendung soll nach dessen Fertigstellung den Webhosting Anbieter Netfarm zur Verfügung stehen. Die Entwürfe wurden einer Heuristischen Evaluation unterzogen. Der Prototyp wurde für einen Teilbereich unter Zuhilfenahme eines Frameworks in PHP erstellt. Anschließend wurde sowohl mit Kunden als auch mit Administratoren ein Nutzertest durchgeführt.

Das erste Kapitel beschäftigt sich mit der Software-Ergonomie. Die menschliche Informationsverarbeitung wird darin als eine Grundlage dargestellt. Außerdem wird der Usability Engineering Lifecycle als Anwendung der Erkenntnisse der Software-Ergonomie vorgestellt. Das zweite Kapitel stellt Usability im Kontext des Webs vor. Erkenntnisse über das Verhalten der Webnutzer und allgemeine Web Usability Richtlinien sind Teile dieses Kapitels. Das dritte Kapitel beschäftigt sich mit den Methoden der Usability Evaluation. Es werden darin ausgewählte Methoden detailliert vorgestellt. Abschließend werden die vorgestellten Methoden diskutiert und Aspekte der Auswertung von Usability Evaluationen erläutert. Im vierten Kapitel geht es um die Konzeption und den Entwurf der Webanwendung. Außerdem werden darin die Ergebnisse der Heuristischen Evaluation vorgestellt. Im fünften Kapitel wird die Implementierung des Prototyps mit dem Framework erörtert. Das sechste Kapitel erläutert die Planung, Durchführung und Ergebnisse der Nutzertests.

1 Software-Ergonomie – theoretische Grundlagen und praktischer Einsatz

1.1 Begriffe

1.1.1 Ergonomie

„Ergonomie ist ein wissenschaftlicher Ansatz, damit wir aus diesem Leben die besten Früchte bei der geringsten Anstrengung und mit der höchsten Befriedigung für das eigene und das allgemeine Wohl ernten.“

[Jastrzebowski 1857]

Diese älteste bekannte Definition für Ergonomie war bis Ende der sechziger Jahre in Vergessenheit geraten. Deshalb schlug Murrell 1949 in England *ergonomics* als Begriff für eine neue wissenschaftliche Disziplin vor. Für dieses Kunstwort führte er die beiden altgriechischen Wörter *ergon* – *Arbeit, Werk, Tat* und *nomos* – *Gesetz, Brauch, Übereinkunft* zusammen. *Human Factors* wird häufig als Synonym für Ergonomie verwendet.

Die IEA (International Ergonomics Association) gab 2000 die offizielle Definition von Ergonomie heraus:

„Ergonomics (or human factors) is the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data and methods to design in order to optimize human well-being and overall system performance.“ [IEA 2006]

So allgemeingültig diese Definition auch ist, sie berücksichtigt die Tatsache, dass die Ergonomie nicht mehr nur in der Arbeitswelt beachtet werden muss. Auch in der Freizeit gilt es Interaktionen zwischen Menschen und anderen Elementen eines Systems (hierzu zählen nicht nur technische Geräte, sondern auch Alltagsgegenstände) so zu gestalten, dass sie an die physischen Eigenschaften und psychischen Fähigkeiten des Menschen angepasst sind.

1.1.2 Software-Ergonomie

„Software-Ergonomie befasst sich disziplinübergreifend mit der benutzergerechten Gestaltung der Mensch-Maschine-Interaktion sowie der Berücksichtigung der Aufgaben- und Organisationserfordernisse und der Benutzerbelange.“ [GI-Fachgruppe Software Ergonomie 2005, S. 7]

Die Mensch-Maschine-Interaktion kann für die Software-Ergonomie auf die Mensch-Computer-Interaktion (MCI) oder Human-Computer-Interaction (HCI) eingegrenzt werden. Im englischsprachigen Raum wird letztgenannter Begriff synonym zum deutschen Begriff Software-Ergonomie verwendet, obwohl die HCI der Gegenstand der Software-Ergonomie als auch der Hardware-Ergonomie ist.

Die Software-Ergonomie grenzt sich von der *Hardware-Ergonomie* dadurch ab, dass sie sich mit der Anpassung an die kognitiven Fähigkeiten (Möglichkeiten zur Verarbeitung von Informationen) des Menschen beschäftigt, während die Hardware-Ergonomie Werkzeuge zur Mensch-Computer-Interaktion (Ein- und Ausgabegeräte) an die physiologischen Eigenschaften des Menschen anpassen will.

Beide Disziplinen befassen sich mit der *Benutzungsschnittstelle* (User Interface), welche nach Herzeg

- die Bedienoberfläche mit den Eingabemöglichkeiten des Benutzers und den Ausgabemöglichkeiten des Computersystems,
- die Regeln der Ein- und Ausgabevorgänge an der Bedienoberfläche sowie
- Systeme zur Unterstützung der Mensch-Computer-Kommunikation

enthält. [vgl. Herzeg 2005]

Mit Ein- und Ausgabemöglichkeiten sind nicht nur die technischen Geräte wie Maus oder Monitor gemeint (Hardware-Ergonomie), sondern auch die softwareseitigen Konstrukte der Dialoggestaltung wie Kommandodialoge, Menüs oder die direkte Manipulation (Software-Ergonomie). Dadurch findet eine wechselseitige Beeinflussung zwischen Mensch und Computer statt, die *Interaktion*.

1.2 Gegenstand und Ziele der Software-Ergonomie

Computerbasierte Werkzeuge haben gegenüber traditionellen Werkzeugen eine enorme Vielseitigkeit zu bieten. In ihrer Bedienbarkeit erleiden sie dadurch aber meist einen Rückschritt. Ursache dafür ist neben der Komplexität selbst das Fehlen langjähriger und evolutionärer Entwicklungsprozesse und die damit verbundene Anpassung an den Menschen aufgrund ihrer vergleichsweise kurzen Geschichte.

Die Software-Ergonomie hat die Motivation, den Mangel an Bedienbarkeit für die programmtechnische Benutzungsoberfläche interaktiver Computersysteme zu beseitigen. Denn versteht ein Benutzer die Art und Weise der Bedienung der Software nicht, wird er auch nicht fähig sein, ihre Vielseitigkeit zu nutzen.

1.2.1 Gegenstand der Software-Ergonomie

Aus der Definition von Software-Ergonomie geht hervor, dass sie sich *mit der benutzer-gerechten Gestaltung der Mensch-Maschine-Interaktion* beschäftigt. Diese Interaktion findet über die Benutzungsschnittstelle statt. Damit stellt sie den eigentlichen Gegenstand der Software-Ergonomie dar. Sichtbar und wirksam wird die Gestaltung schließlich an der Benutzungsoberfläche. Die Oberflächengestaltung wird in der Software-Ergonomie aber immer im Zusammenhang mit der Funktionalität betrachtet. Die Vielfalt in der Funktionalität heutiger Software muss sich in dem Maße in der Oberfläche abbilden, wie der Benutzer damit umgehen kann.

Es sind vier Faktoren zu berücksichtigen, wenn es darum geht, die Benutzungsschnittstelle benutzergerecht zu gestalten:

Menschengerechte Gestaltung

Im Gegensatz zu rein technischen Schnittstellen (zwischen zwei Maschinen) hat die Benutzungsschnittstelle dem Menschen Rechnung zu tragen. Der Benutzer kann mit seinen Stärken, Schwächen, Bedürfnissen und Unterschieden in der Entwicklung nur schwer formal beschrieben werden. Dennoch muss eine Anpassung der Software an den Menschen (nicht umgekehrt) erfolgen, um benutzergerecht zu sein. Dazu werden „Kenntnisse über menschliche Wahrnehmung, Denken und Problemlösen sowie über Lernen, Kommunikation und Kooperation benötigt“ [Eberleh, Oberquelle & Oppermann 1994, S. 4]. Im Abschnitt 1.3 werden einige dieser psychologischen Grundlagen vorgestellt.

Aufgabenangemessene Gestaltung

Software wird vom Benutzer für einen bestimmten Zweck verwendet. Meist möchte der Benutzer eine konkrete Aufgabe, insbesondere im Arbeitsleben, damit lösen. Aufgabenangemessen kann nur gestaltet werden, wenn Kenntnisse über die Aufgaben der Benutzer und ihre Vorgehensweise beim Erledigen der Aufgaben bekannt sind. Doch diese Aufgaben müssen zuerst menschengerecht formuliert werden bevor eine aufgabenangemessene Funktionalität der Software bei der Erledigung Unterstützung bieten kann. Hierzu werden Kenntnisse der Arbeitswissenschaften in der Software-Ergonomie angewandt.

Technikbewusste Gestaltung

Bei der Vielfalt an technischen Optionen, die heutzutage angeboten werden, fällt es nicht leicht, diese auch benutzergerecht einzusetzen. Häufig werden neue Technologien nur zum Selbstzweck eingesetzt. Für die benutzergerechte Gestaltung ist es notwendig, die verfügbaren technischen Möglichkeiten zum Wohle des Benutzers einzusetzen. Das kann bedeuten, dass der Schritt zu einer innovativen Technologie gewagt wird, oder aber etablierte, traditionelle Techniken verwendet werden. Letztlich müssen hier ökonomische und im Einzelfall auch ökologische Rahmenbedingungen bei der Entscheidung berücksichtigt werden.

Organisationsgerechte Gestaltung

Benutzer handeln sowohl im Arbeitsleben als auch im Privatleben meist in einem Netzwerk zusammen. Daneben findet durch die Verbreitung des Internets und der Computer auch eine technische Vernetzung statt. Diese beiden Aspekte der sozialen und technischen Vernetzung müssen durch eine organisationsgerechte Gestaltung berücksichtigt werden. Dabei muss die jeweils vorhandene organisatorische Einbindung des Softwarebenutzers beachtet werden, wenn Kenntnisse über den Menschen und seine Aufgaben gewonnen werden.

Über menschengerechte und aufgabenangemessene Gestaltung existiert ein weites Spektrum fundierten Wissens durch die Forschungsarbeit der Software-Ergonomie. Die letzteren Aspekte, technikbewusste und organisationsgerechte Gestaltung, haben erst Mitte der neunziger Jahre an Bedeutung gewonnen. Als Grund dafür können die Verbreitung des Internets sowie

die Entwicklung von „haushaltstauglichen“ PCs angesehen werden. Heutzutage haben alle vier Aspekte eine große Bedeutung, wenn es um die Gestaltung neuer Software geht.

Beispielsweise ermöglichen neue Technologien den Einsatz von bekannter Software in neuen Formaten. PocketPCs und die Multifunktionalität von Mobiltelefonen seien als Beispiele für Geräte genannt, die eine Anpassung der Benutzerschnittstelle an ein vergleichsweise kleines Ausgabegerät (im Gegensatz zum Desktop-Monitor) und das damit verbundene Benutzerverhalten erfordern. Technikbewusst muss auch der Einsatz von Computern als eingebettete Systeme (Embedded Systems) in modernen technischen Geräten betrachtet werden. Hinzu kommt, dass Software damit nicht mehr als grafische Oberfläche in Erscheinung tritt. Bei der Entwicklung ergonomischer Richtlinien für eingebettete Systeme verwischt die Grenze zwischen Software- und Hardware-Ergonomie zunehmend.

Es sollte auch nicht vergessen werden, dass Computersysteme nicht mehr nur zur Arbeit genutzt werden, sondern auch im Freizeitbereich. Gerade hier kann die Nutzungsdauer sogar länger als am Arbeitsplatz sein, da die Software meist zum Vergnügen genutzt wird. In diesem Zusammenhang ist auch wieder die Hardware-Ergonomie gefragt, welche die körperlichen Belastungen durch lange Nutzung des Computersystems minimieren soll.

Auch die veränderte Alterstruktur der Benutzer muss in der Software-Ergonomie berücksichtigt werden. Sowohl Kinder und Jugendliche als auch Senioren nutzen Computer in einem höheren Maße als noch vor einigen Jahren und stellen spezielle Anforderungen an die Hardware und Software durch ihre körperlichen und geistigen Fähigkeiten.

1.2.2 Ziele der Software-Ergonomie

Wenn die Grundsätze der Software-Ergonomie ignoriert werden, dann trifft den Benutzer eine Reihe von negativen Auswirkungen wie psychische und physische Belastungen, geringere Flexibilität, ungewohnte Einteilung der Arbeitsabläufe und damit verbunden ein höherer Einarbeitungsaufwand. Mit diesen Beeinträchtigungen beim Benutzer gehen Einbußen in der Effektivität und Effizienz der Arbeit mit dem Computersystem einher. Als Folge dessen sinkt die Zufriedenheit des Benutzers und die Kosten für den Arbeitgeber steigen, da Aufgaben langsamer erledigt werden.

Die Eigenschaften eines Softwaresystems müssen an die psychischen Eigenschaften der damit arbeitenden Menschen angepasst werden, damit folgende Ziele erreicht werden können:

Effektivität

„die Genauigkeit und Vollständigkeit, mit der Benutzer ein bestimmtes Ziel erreichen.“ [DIN EN ISO 9241 Teil 11]

Effizienz

„der im Verhältnis zur Genauigkeit und Vollständigkeit eingesetzte Aufwand, mit dem Benutzer ein bestimmtes Ziel erreichen.“ [DIN EN ISO 9241 Teil 11]

Zufriedenheit

„Freiheit von Beeinträchtigungen und positiven Einstellungen gegenüber der Nutzung des Produkts“ [DIN EN ISO 9241 Teil 11]

In der Praxis gilt es zu bestimmen, in wie weit diese Ziele erreicht wurden und welches Maß an Usability (Gebrauchstauglichkeit) erreicht wurde. Damit stellt sich die Software-Ergonomie ein weiteres Ziel:

Usability Evaluation

Um das Ausmaß festzustellen, in dem ein Produkt von einem Benutzer effizient, effektiv und mit Zufriedenheit genutzt werden kann, damit er in einem bestimmten Kontext seine Ziele erreicht, entwickelt die Software-Ergonomie Methoden für die Evaluation des Produkts.

Neben der Entwicklung von Methoden zur software-ergonomischen Evaluation bemüht sich die Software-Ergonomie außerdem darum weitere Richtlinien zu schaffen, die im Software Engineering Prozess genutzt werden sollten, um das Produkt besser an den Menschen anzupassen. Da Computersysteme als Werkzeuge erst seit relativ kurzer Zeit im Einsatz sind, ist für die Entwicklung solcher Richtlinien besonders das Sammeln von Erfahrungen der Nutzer und das Aufbereiten der Erkenntnisse in genormter Form wichtig, damit ein einheitlicher Fortschritt zu Gunsten der Benutzer möglich ist. Ein weiteres Ziel lautet somit:

Normierung

Durch Benutzer-, Tätigkeits- und Kontextanalysen will die Software-Ergonomie Erfahrungen sammeln, die in Richtlinien zur Modellierung, Gestaltung und Evaluation benutzergerechter Softwaresysteme umgesetzt werden.

Um diese Ziele zu erreichen, bedient sich die Software-Ergonomie Erkenntnissen und Methoden anderer Disziplinen.

1.3 Menschliche Informationsverarbeitung als Grundlage der Software-Ergonomie

Computersysteme werden von Benutzern im Kontext einer sozialen Organisation, um Aufgaben mit dem technischen System zu erfüllen. Deshalb kann Software-Ergonomie nur auf der Grundlage anderer Fachbereiche entwickelt werden. Herczeg gibt einen umfassenden Überblick, indem er schreibt:

„Die beteiligten Wissenschaftsgebiete und Disziplinen sind vor allem

- Arbeitswissenschaften (Arbeitsschutz und Arbeitssicherheit, Arbeitsorganisation, Arbeitssoziologie, Arbeitswirtschaft),
- Human- und Geisteswissenschaften (Arbeits- und Organisationspsychologie, Wahrnehmungs- und Kognitionspsychologie, Physiologie, Medizin, Linguistik, Soziologie, Medien- und Kommunikationswissenschaften),

- Computerwissenschaften (Informatik, Informationswissenschaften, Telematik, Elektrotechnik, Nachrichtentechnik) sowie
- Gestaltungswissenschaften (Design, Kunst).“ [Herczeg 2005, Hervorhebungen im Original]

Obwohl sich einzelne Fachbereiche überlappen, zeigt der Überblick eine Komplexität, die hier nicht tiefer erläutert werden soll. Vielmehr wird mit der menschlichen Informationsverarbeitung eine wichtige psychologische Grundlage der Software-Ergonomie vorgestellt. Sie stellt notwendiges Wissen über die menschliche Kognition bereit, welche die Voraussetzung dafür ist, dass der Benutzer die dargestellten Informationen einer Benutzeroberfläche wahrnehmen und verarbeiten kann.

1.3.1 Die Architektur der menschlichen Kognition

In den sechziger Jahren wurde Verhalten und Erleben des Menschen erstmals als Informationsverarbeitung in der Psychologie betrachtet. In diesem Zusammenhang wurde die Computer-Metapher entwickelt. Als Analogie zur Hardware wird die Physiologie des Menschen gesehen. Die Software wird durch das geistige und seelische Geschehen im Menschen repräsentiert.

Wie im Computer sich Hardware und Software gegenseitig beeinflussen, so konnten in der Psychologie kausale Abhängigkeiten zwischen körperlichen und seelischen Geschehen besser durch die Computer-Metapher beschrieben werden. Beispielsweise beeinflussen Emotionen die Herzrate oder umgekehrt kann die Veränderung der Hardware des Körpers zu softwareseitigen (seelischen) Reaktionen führen. Beim Phantomschmerz z. B. verursachen die Synapsen, die für den verlorenen Körperteil (Hardware) zuständig waren, ein Schmerzgefühl (Software) in den nicht mehr vorhandenen Gliedmaßen, da sie immer noch Signale (Erregung) zu den Nervenzellen aussenden [vgl. Glaser 1994, S. 10].

Auf dieser Basis wurde in der Informationsverarbeitungspsychologie die Architektur der menschlichen Kognition entworfen (Abbildung 1). Der Mensch steht dabei der Umwelt gegenüber. Der informationelle Einfluss der Umwelt auf das Individuum wird unter dem Begriff Wahrnehmung¹ zusammengefasst. Mittels der Motorik kann der Mensch auf seine Umwelt einwirken. Wahrnehmung und Motorik bilden damit die beiden Schnittstellen des Exekutivsystems mit der Außenwelt. Wie der Mensch seine Umwelt wahrnimmt, spielt in der Software-Ergonomie besonders bei der Gestaltung der Benutzungsschnittstelle eine Rolle.

¹ Wahrnehmung wird in dieser Arbeit nicht näher behandelt. Deshalb wird zum Einstieg auf [Glaser 1994] und [Zimbardo & Gerrig 1999] verwiesen.

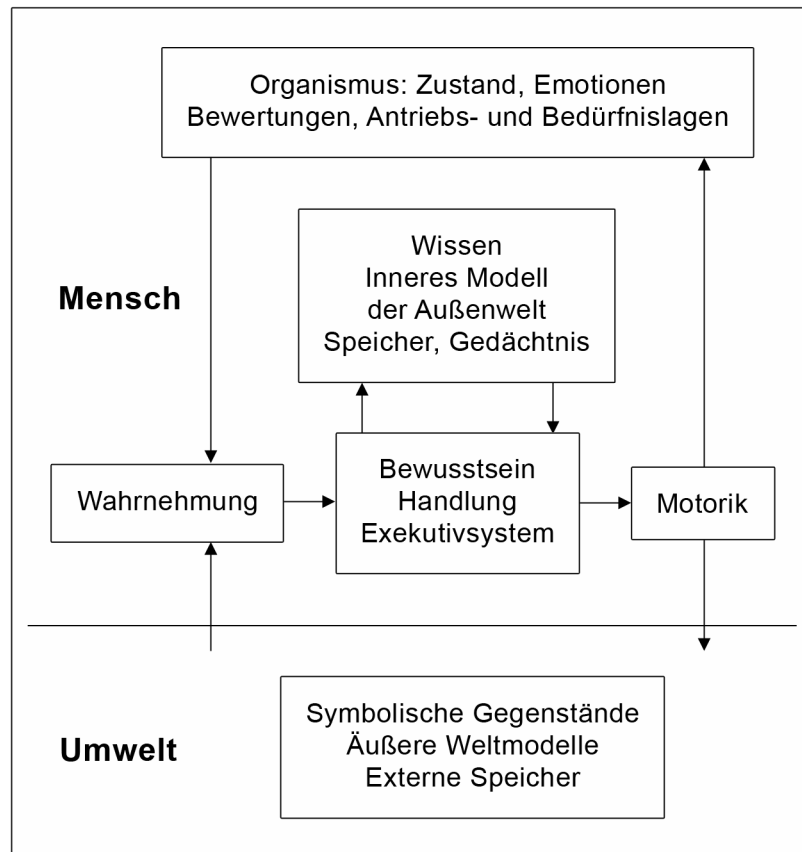


Abbildung 1: Die Architektur der menschlichen Kognition nach [Glaser 1994, S. 8]

Dem Bewusstsein wird in der kognitiven Architektur die Rolle des Exekutivsystems zugeteilt. Zwar fehlt bis heute eine „befriedigende, erklärende Theorie des Bewusstseins“ [Glaser 1994, S. 9], aber informationell kann zumindest festgehalten werden, dass hier der Output des Wahrnehmungssystems verarbeitet wird, indem bewusste Handlungen geplant, gesteuert und überwacht werden. Der Output dieses Subsystems wird dann über die Motorik auf die Umwelt angewendet.

Daneben verfügt die menschliche Kognition über ein Gedächtnissystem, in den Erfahrungen als Wissensbasis kurz- oder langfristig für kognitive Leistungen gespeichert werden (siehe Abschnitt 1.3.2). Ähnlich den Schnittstellen zwischen Wahrnehmung, Motorik und Bewusstsein, laufen auch zwischen Gedächtnis und Exekutivsystem Input und Output Mechanismen ab (Speichern von Informationen, Abrufen von Erinnerungen).

Emotionen stellen im Informationsverarbeitungsmodell die Art und Weise dar, wie Bewertungen, Antriebs- und Bedürfnislagen wahrgenommen werden. Ihre Ursachen liegen sowohl auf physiologischer Ebene (Hunger, Durst, Schmerz etc.) als auch auf kognitiver Ebene (Unlust, Spannung, Beruhigung etc.).

1.3.2 Gedächtnis und mentale Modelle

Das Gedächtnis kann in Arbeitsgedächtnis und Langzeitgedächtnis eingeteilt werden. Das *Arbeitsgedächtnis* hat eine begrenzte Kapazität und speichert Informationen vorübergehend zum Lösen von komplexen Aufgaben. Es unterstützt beim Problemlösen und dem Formulieren von Zielen. Außerdem hilft es dabei, die Umwelt zu verstehen und mentale Modelle dafür zu entwickeln.

Das *Langzeitgedächtnis* ist für dauerhaftes Speichern von Informationen zuständig. Mit ihm stehen Prozesse wie Lernen, Behalten (durch regelmäßigen Abruf), Erinnern und Vergessen in Verbindung. Die zu speichernde Information spielt eine Rolle dabei, in welchen Teilen des Langzeitgedächtnis sie gespeichert wird. Im *deklarativen Gedächtnis* werden Tatsachen und Ereignisse gespeichert, die bewusst wiedergegeben werden können, z. B. Dresden ist die Landeshauptstadt von Sachsen. Das *prozedurale Gedächtnis* speichert Fertigkeiten, Erwartungen und Verhaltensweisen. Die dort gespeicherten Inhalte können ohne Einsatz des Bewusstseins das Verhalten beeinflussen, z. B. Gehen, Radfahren oder Klavierspielen.

Für die Software-Ergonomie sind Erkenntnisse über die Belastung des Gedächtnis' der Benutzer bei der Erfüllung ihrer Aufgaben relevant. Da das Arbeitsgedächtnis sehr begrenzt ist, gilt es den Informationsoutput der Benutzeroberfläche so zu gestalten, dass er in diesem Rahmen verarbeitet werden kann. Andernfalls muss das Gedächtnis durch Lern- oder Behaltensprozesse die Informationen längerfristig speichern. Dieser Mehraufwand beeinträchtigt jedoch meist die Effizienz, mit welcher der Benutzer seine Aufgaben erfüllt.

Die Leistungsfähigkeit des Gedächtnis' wird „durch die Bildung mentaler Modelle zur internen Repräsentation durch Erfahrung gewonnenen Wissens über Sachverhalte und Funktionen der Außenwelt“ [Glaser 1994, S. 13] bestimmt. *Mentale Modelle* sind die Abbildung von Prozessen der Wirklichkeit durch die menschliche Wahrnehmung. Aber auch über die eigenen Fähigkeiten bestehen mentale Modelle. Mit ihrer Hilfe versucht der Mensch Vorgänge zu erklären und Probleme zu lösen.

Beim Erstellen eines mentalen Modells fasst der Mensch die Teilschritte eines Prozesses als Einheit mit bestimmten In- und Output zusammen. Der Output wird durch das mentale Modell berechnet und entspricht dem vorhergesagten Ergebnis des realen Prozesses. Für die Software-Ergonomie sind mentale Modelle von Bedeutung, da Menschen von ihren Aufgaben solche Modelle erstellen. Um ein Softwaresystem nutzen zu können erstellt der Benutzer auch davon ein mentales Modell. Durch das Erfassen und Verstehen der Software wird dieses Modell kontinuierlich erweitert. Wichtig aus ergonomischer Sicht ist dabei, dass das mentale Modell und die reale Anwendung in möglichst vielen Punkten übereinstimmen, d. h. dass der vorhergesagte Output des mentalen Modells dem tatsächlichen Output der Anwendung entspricht.

Schwierig wird dies insbesondere, wenn eine Software bestehende Prozesse, von denen Benutzer schon mentale Modelle haben, unterstützen soll. Hier kann es sehr lange dauern, bis der Benutzer die Software versteht, wenn diese dem bestehenden mentalen Modell entgegensteht. Deshalb sollte in diesem Fall versucht werden, die Software dem mentalen Modell der Benutzer anzupassen. Eine vollständige Übereinstimmung zwischen mentalen Modell und

Anwendung wird nicht erreichbar sein. Dafür sollten Hilfssysteme bereitgestellt werden, die dem Benutzer beim Erlernen der Software und somit Abgleichen seines mentalen Modells mit der Realität behilflich sind.

Diese Grundlagen sind ein Bruchstück dessen, was die Interdisziplinarität der Software-Ergonomie ausmacht, sollen aber im Rahmen dieser Diplomarbeit genügen, um den Wert der Zusammenarbeit mit anderen Fachbereichen zu betonen.

1.4 Benutzerzentrierter Entwicklungsprozess – Usability Engineering

Die Erkenntnisse aus der Software-Ergonomie müssen praktisch angewendet werden, um nutzerfreundliche Software zu schaffen. Mit dem *Usability Engineering* wurde ein ingenieurmäßiger Prozess geschaffen, der die allgemeinen Komponenten des Software Engineering wie das Definieren von Anwendungsanforderungen, Setzen von Zielen und das iterative Design und Testen adaptiert und mit Methoden der software-ergonomischen Evaluation (Usability Evaluation) verbindet. Ziel soll es sein, Usability als Produkteigenschaft zu objektivieren und messbar zu machen.

Nielsen definiert in seinem gleichnamigen Buch den Begriff folgendermaßen:

„Usability engineering is not a one-shot affair where the user interface is fixed up before the release of a product. Rather, usability engineering is a set of activities that ideally take place throughout the lifecycle of the product, with significant activities happening at the early stages before the user interface has even been designed.“ [Nielsen 1993, S. 71]

Meist werden Usability-Experten am Ende des Projekts hinzugezogen, um einen Usability Test durchzuführen. Dabei stellt sich heraus, dass die Ergebnisse des Tests nicht mehr ins Produkt umgesetzt werden können, da nicht mehr genügend Zeit bis zur Veröffentlichung bleibt oder das Produkt bereits auf dem Markt ist und umfangreiche Änderungen somit schwer zu bewerkstelligen. Neben dem Zeitmangel am Ende der Entwicklung sind Änderungen zu diesem Zeitpunkt auch wesentlich kostenintensiver. Diese Vorgehensweise endet eher in Unzufriedenheit über das Thema Usability, beweist aber wie wichtig die Usability Evaluation schon in frühen Phasen des Produkt Lebenszyklus’ und das Zusammenspiel der Maßnahmen des Usability Engineering mit dem Software-Entwicklungsprozess sind. Umfangreiche Ausführungen über die Chancen durch den Einsatz von Usability Engineering finden sich sowohl in [Mayhew 1999] Kapitel 18 als auch in [Kalbach 2003]. Hier soll nicht näher darauf eingegangen werden.

Der *Usability Engineering Lifecycle* legt Aufgaben in einer bestimmten Reihenfolge fest, welche in den bestehenden Produkt Entwicklungszyklus integriert werden, um Usability zu erreichen. Nach Mayhew sind es Folgende:

- Strukturierte Aufgaben zur Analyse der Usability Anforderungen
- Eine explizite Aufgabe zum Setzen der Usability Ziele, die sich aus der Anforderungsanalyse direkt ergeben

- Aufgaben zur Unterstützung eines strukturierten top-down Ansatzes für das Design der Benutzungsschnittstelle, die sich direkt aus den Usability Zielen und anderen Anforderungen ergeben
- Aufgaben zur objektiven Usability Evaluation des iterativen Designs zum Erreichen der Usability Ziele

[vgl. Mayhew 1999, S. 5f.]

Für jede Aufgabe stehen Techniken zur Verfügung, aus welchen je nach Art und Größe des Projekts gewählt werden sollte, um das Ziel der jeweiligen Aufgabe zu erreichen. Die Evaluation soll dabei in allen Phasen des Designprozesses im Mittelpunkt stehen und nicht erst am Ende. Das Star-Life-Cycle-Modell nach Hix und Hartson (Abbildung 2) verdeutlicht diesen Prozess und nennt gleichzeitig die grundsätzlichen Phasen des Usability Engineering Lifecycle.

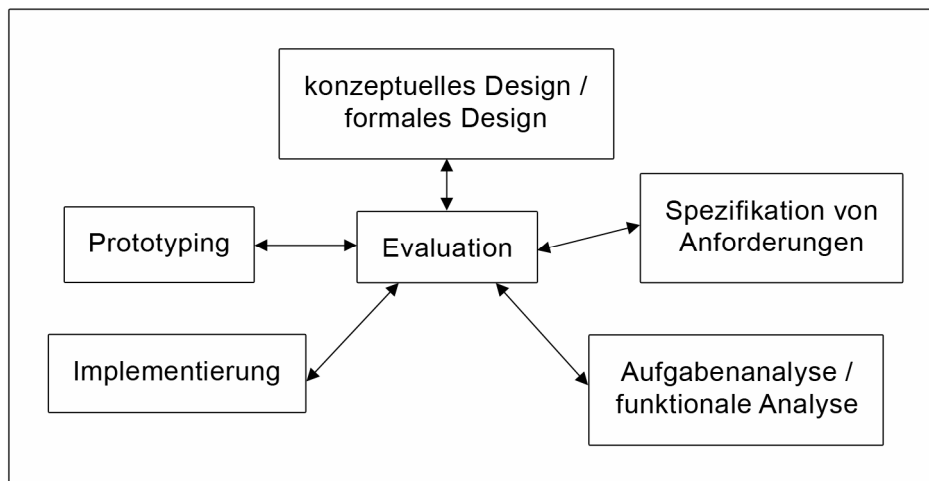


Abbildung 2: Das Star-Life-Cycle-Modell nach Hix und Hartson [vgl. Schweibenz & Thissen 2003, S. 51]

1.4.1 Usability Engineering Lifecycle nach Mayhew und Nielsen

Mayhew teilt ihr Modell in drei große Phasen ein, in denen jeweils spezifische Aufgaben zusammengefasst sind (Liste 1). Nachfolgend werden die Aufgaben kurz beschrieben.

Nutzerprofil

Es wird eine Beschreibung der für das User Interface Design relevanten Nutzer Charakteristik (z. B. Computerkenntnisse, Erfahrung im Beruf) erstellt. Diese identifiziert auch die Hauptnutzerkategorien für die kontextbezogene Aufgabenanalyse.

Kontextbezogene Aufgabenanalyse

Hierbei wird eine Studie über die aktuellen Aufgaben der Nutzer und deren Arbeitsabläufe erstellt, die ein Verständnis der spezifischen zugrunde liegenden Nutzer Ziele bieten soll.

Phase 1: Anforderungsanalyse

- 1) Nutzerprofil
 - 2) Kontextbezogene Aufgabenanalyse
 - 3) Usability Ziele setzen
 - 4) Möglichkeiten und Beschränkungen der Plattform
 - 5) Allgemeine Design Prinzipien
-

Phase 2: Entwurf/Test/Entwicklung

- | | |
|---|---------|
| 6) Überarbeitung der Aufgaben | Level 1 |
| 7) Entwurf konzeptioneller Modelle | |
| 8) Konzeptionelle Mock-ups | |
| 9) Iterative Evaluation der konzeptionellen Modelle | |
| 10) Style Guide Entwicklung | |
| <hr style="border-top: 1px dashed #000;"/> | |
| 11) Screen Design Standards | Level 2 |
| 12) Prototyping nach Screen Design Standards | |
| 13) Iterative Evaluation der Screen Design Standards | |
| 14) Style Guide Entwicklung | |
| <hr style="border-top: 1px dashed #000;"/> | |
| 15) Detailliertes User Interface Design | Level 3 |
| 16) Iterative Evaluation des detaillierten User Interface Designs | |
-

Phase 3: Installation

- 17) Feedback der Nutzer
-

Liste 1: Die Phasen des Usability Engineering Lifecycle nach [Mayhew 1999]

Usability Ziele setzen

Es werden qualitative Ziele gesetzt, die sich aus den Usability Anforderungen zum Nutzer Profil (User Profile) und der kontextbezogenen Aufgabenanalyse ergeben. Außerdem werden quantitative Ziele festgelegt, die Kriterien für die minimal akzeptable Nutzer Leistung und Zufriedenheit festhalten. Diese Ziele stellen später den Fokus für das Design dar und bilden die Basis für die iterative Usability Evaluation.

Möglichkeiten und Beschränkungen der Plattform

Die Möglichkeiten und Auflagen der für das Produkt gewählten Plattform (z. B. Macintosh, Windows, Unix) werden dokumentiert. Sie grenzen die Möglichkeiten für das User Interface Design ein.

Allgemeine Design Prinzipien

Zusätzlich zu den Informationen der produktspezifischen Analyse werden relevante Prinzipien und Richtlinien des allgemeine User Interface Design gesammelt und geprüft. Diese fließen neben dem aus den ersten vier Aufgaben entwickelten Style Guide in den Design Prozess ein.

Überarbeitung der Aufgaben

Das Modell der Nutzeraufgaben Organisation und Aufgaben Szenarios, welche aus der Aufgaben Analyse gewonnen wurden, wird überarbeitet, um die Stärken der Automatisierung auszunutzen und effektiver Unternehmensziele zu unterstützen.

Entwurf konzeptioneller Modelle

Basierend auf allen bisherigen Aufgaben werden erste konzeptuelle Design Alternativen generiert. Besonders wichtig sind hierbei Navigationswege, Hauptansichten und Regeln für die konsistente Präsentation von Arbeitsprozessen. Das Screen Design spielt dabei noch keine Rolle.

Konzeptionelle Mock-ups

Skizzen und prototypische Modelle (Mock-ups) der high-level Design Ideen aus der vorigen Aufgabe werden bereitet, um die funktionale Organisation zu repräsentieren.

Iterative Evaluation der konzeptionellen Modelle

Die Modelle werden evaluiert und bearbeitet durch iterative Evaluation Techniken (z. B. formale Usability Tests mit repräsentativen Endnutzern, bei denen Aufgaben an den Modellen ausgeführt werden als ob sie die wirkliche Benutzeroberfläche wären). Diese und die beiden vorigen Aufgaben werden iterativ durchlaufen bis alle größeren Usability Probleme gefunden und beseitigt wurden.

Screen Design Standards

Auf der Basis von Industrie- und Firmenstandards, den Daten aus der Anforderungsanalyse und dem produktspezifischen konzeptuellen Design Modell werden produktspezifische Standards und Konventionen für alle Aspekte des detaillierten Screen Designs entwickelt, um Konsistenz in der Benutzungsschnittstelle zu erzielen.

Prototyping nach Screen Design Standards

Die Screen Design Standards (und das konzeptuelle Design Modell) werden auf ein detailliertes Design des User Interface für ausgewählte Produktfunktionen angewendet. Dieses Design wird in Prototypen realisiert.

Iterative Evaluation der Screen Design Standards

Evaluationstechniken wie formale Usability Tests werden an den Screen Design Prototypen angewendet. Danach wird iterativ der Design/Evaluations-Prozess wiederholt, um die Screen Design Standards zu verfeinern und zu überprüfen bis alle größeren Usability Probleme beseitigt wurden.

Style Guide Entwicklung

Sowohl nach der Iteration des konzeptuellen Design als auch des Screen Design werden die Ergebnisse für das detaillierte Screen Design in dem Produkt Style Guide dokumentiert. Die-

ses Dokument enthält schon die Ergebnisse aus der Anforderungsanalyse. Während der folgenden Aufgabe wird den Aufzeichnungen im Style Guide gefolgt, um Qualität, Konsistenz und Stimmigkeit zu gewährleisten.

Detailliertes User Interface Design

In dieser Aufgabe wird das User Interface Design für das komplette Produkt detailliert erarbeitet. Diese Phase des Designs geht direkt in die Produktentwicklung über.

Iterative Evaluation des detaillierten User Interface Designs

Während der Produktentwicklung werden kontinuierlich Evaluationstechniken angewendet, um die Überprüfung auf bisher unbewertete Teile der Funktionalität und Nutzergruppen auszuweiten, das User Interface zu verfeinern und bezüglich der Usability Ziele zu überprüfen.

Feedback der Nutzer

Nachdem das Produkt veröffentlicht wurde, wird Feedback der Nutzer gesammelt, um es in Erweiterungen des Designs oder für verwandte Produkte zu verwenden.

Nielsen schlägt ein Modell aus 11 Schritten (Liste 2), welches hier im Überblick als Vergleich zum Modell von Mayhew dienen soll. Besondere Beachtung findet auch bei ihm die Aufgabe, den Nutzer kennen zu lernen, bevor das Design überhaupt konzipiert wird. Denn weder Designer, Entwickler oder Führungskräfte sind Nutzer des Produkts und sollten deshalb nicht auf ihre Intuition oder Vorlieben bezüglich des Produkts Design-Entscheidungen treffen [vgl. Nielsen 1993, S. 13f.].

-
- 1) Den Nutzer kennen
 - a) Individuelle Nutzercharakteristik
 - b) Aktuelle und gewünschte Aufgaben des Nutzers
 - c) Funktionsanalyse
 - d) Die Entwicklung des Nutzers und des Jobs
 - 2) Konkurrenzanalyse
 - 3) Setzen von Usability Zielen
 - a) Analyse der finanziellen Beeinträchtigungen
 - 4) Paralleles Design
 - 5) Design mit Teilnahme
 - 6) Abgestimmtes Design der gesamten Benutzeroberfläche
 - 7) Anwenden von Richtlinien und heuristische Analyse
 - 8) Prototyping
 - 9) Empirische Untersuchungen
 - 10) Iteratives Design
 - a) Erfassen des Design Grundprinzips
 - 11) Sammeln von Feedback aus Feldstudien
-

Liste 2: Die Phasen des Usability Engineering Lifecycle nach [Nielsen 1993, S. 72]

Dennoch schreibt Nielsen auf seiner Webseite provokativ „First Rule of Usability? Don't Listen to Users“ [Nielsen 2001]. Dies ist keine Aufforderung, sich vom Nutzer abzuwenden; vielmehr soll die Aufgabe den Nutzer kennen zu lernen richtig verstanden werden. Nutzer sollten nicht zu einem Design befragt werden, da hierbei Kommentare zur Oberfläche nicht aber zum tatsächlichen Nutzen erhalten werden. Vielmehr sollten Nutzer dabei beobachtet werden, wenn sie konkrete Aufgaben mit dem User Interface versuchen zu erledigen. Nach solchen formalen Usability Tests kann eine Umfrage mit den Nutzern durchgeführt werden, wobei auch hier Probleme mit der Objektivität der Daten auftreten können.

In der Praxis ist es meist weniger wichtig welchem Usability Engineering Lifecycle gefolgt wird. Hauptsache ist, dass zu Beginn eine Analyse des Nutzers und seiner Aufgaben durchgeführt wird, sich daraus ergebende Usability Ziele klar definiert werden und in iterativen Schritten das Design und die Implementierung nach Kriterien zum Erreichen dieser Ziele evaluiert werden. Die größere Hürde ist eher die Integration des Usability Engineering in bestehende Projektabläufe.

1.4.2 Planung und Integration des Usability Engineering Prozesses

Usability Engineering Aufgaben laufen sowohl parallel als auch überschneidend mit den Software Engineering Aufgaben. Der Software-Entwicklungsprozess ist dabei meist nicht nutzerorientiert ausgerichtet. Grundlegende Aufgabe des Usability-Experten muss es daher sein, im Projektteam ein Bewusstsein für das Thema Benutzerfreundlichkeit zu schaffen, damit er die Ergebnisse seiner Arbeit mit den Schritten der Software-Entwicklung abgleichen kann. Besonders wichtig ist dabei, dass die Projektleitung sowie andere Entscheidungsträger Usability klar als Ziel des Projekts definieren.

Nach Mayhew können Usability Aufgaben am besten durch „a whole cross-functional project team in which usability experts represent one of many important skill sets and perspectives“ [Mayhew 1999, S. 21] ausgeführt werden. Traditionell werden im Software Engineering verschiedene Schritte von unterschiedlichen Spezialisten getrennt voneinander durchgeführt. In einem *cross-functional team* arbeiten Spezialisten gemeinsam an einem Schritt, wodurch jeder seine entsprechende Perspektive und Fähigkeit in die Entscheidungen einbringen kann. Damit wird die Kommunikation vereinfacht und ein besseres Verständnis für die Arbeit der Teammitglieder gefördert.

Ebenfalls wichtig für die Integration ist die Planung der Usability Maßnahme. Für das Usability Engineering sollte ein eigener Projektplan erstellt werden, welcher die verschiedenen Aktivitäten, Termine, Beteiligten und Meilensteine enthält. Bei der Erstellung sollten der gesamte Entwicklungsprozess berücksichtigt werden sowie der Projektplan für das gesamte Projekt. Denn es muss definiert sein, wann ein Austausch mit anderen Abteilungen notwendig ist, z. B. wenn der Usability Experte auf Prototypen aus der Entwicklung angewiesen ist.

Um die Erfüllung des erstellten Plans zu gewährleisten, sollte dieser von der Projektleitung überwacht werden. Außerdem sollten die Usability Aktivitäten dokumentiert und anderen Teammitgliedern (z. B. Entwicklern, Designern) zugänglich gemacht werden. Dies erhöht die Akzeptanz des Themas im Projektteam und verringert das Risiko, dass der benutzerzentrierte Entwicklungsprozess scheitert.

2 Web Usability

2.1 Begriffe im Kontext des WWW

2.1.1 Usability

„Usability is a quality attribute relating to how easy something is to use. More specifically, it refers to how quickly people can learn to use something, how efficient they are while using it, how memorable it is, how error-prone it is, and how much users like using it.“

[Nielsen 2006]

Diese Definition von Nielsen zeigt wie viele Attribute mit dem Begriff Usability verbunden sind. Ähnlich vielfältig sind auch die Übersetzungen ins Deutsche: Nutzbarkeit, Gebrauchstauglichkeit oder Benutzerfreundlichkeit sind einige Synonyme, welche in dieser Diplomarbeit auch verwendet werden. Klarheit über den Begriff wurde durch eine allgemein anerkannte Definition in der DIN EN ISO 9241 geschaffen, welche Usability abstrakt definiert.

„Usability bezeichnet das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und mit Zufriedenheit zu erreichen.“ [DIN EN ISO 9241 Teil 11]

Web Usability beschäftigt sich mit dem Produkt Webseite in seinen Dimensionen Inhalt (Content-Design), Gestaltung (Page-Design) und Struktur (Site-Design). Nachfolgend wird die Bedeutung der Usability Kriterien Effektivität, Effizienz und Zufriedenheit, welche schon im Abschnitt 1.2.2 als Ziele der Software-Ergonomie definiert wurden, in den einzelnen Dimensionen einer Webseite erläutert.

2.1.2 Effektivität

Ein Benutzer verfolgt eine bestimmte Absicht, wenn er sich im Internet aufhält. Eine Webseite ist dann effektiv, wenn er seine Ziele erreichen kann. Der Inhalt der Seite ist effektiv, wenn er die vom Nutzer benötigten Informationen enthält und diese von ihm auch auffindbar sind. Effektivität in der Gestaltung einer Seite liegt vor, wenn Elemente sich durch die Gestaltung in ihrer Funktion unterscheiden lassen und somit eine Orientierung für den Nutzer überhaupt möglich ist. Wenn Links zwischen den einzelnen Seiten eines Webauftritts vorhanden und als solche erkennbar sind, dann trägt das zur Effektivität der Seitenstruktur bei.

2.1.3 Effizienz

Eine Webseite ist effizient, wenn der Benutzer wenige Ressourcen (physisch, psychisch, technisch, finanziell) einsetzen muss, um sein Ziel zu erreichen. „Der Aufwand muss für den Nutzer geringer als der durch die Website erreichte Ertrag sein oder ihm entsprechen [...]“

[von Gizycki 2002, S. 3], sonst neigt der Nutzer dazu, seine Absichten auf einer anderen Webseite umzusetzen. Dabei sollten die Inhalte so aufbereitet sein, dass sie leicht vom Nutzer erfassbar sind (siehe Abschnitt 2.3). Die Aufteilung in Blöcke und Spalten, eine abwechslungsreiche Präsentation durch einen Mix aus Text und Bild sowie dem Inhalt angemessene Ladezeiten sorgen für einen effizienten Inhalt der Webseite.

Für die Gestaltung bedeutet Effizienz, dass der Nutzer eine optische Hierarchie erkennen kann und Inhalte in Module einordnen kann. Dies kann zum Beispiel durch Metaphern erreicht werden, wenn sie im Zusammenhang gut geeignet sind. Effiziente Gestaltung bedeutet auch, dem Nutzer gelernte Prinzipien darzubieten. Beispielsweise hat sich die Konvention gebildet, dass Primär- und Sekundärnavigationen links oder oben auf einer Seite angeordnet werden.

Sind die Erwartungen des Nutzers an die Navigation erfüllt, d. h. sind Navigationshierarchien erkennbar und leicht nutzbar, ist es für den Nutzer erkennbar, wo in der Struktur er sich befindet und sind Rückschritte (besonders zur Startseite) möglich, dann kann von einer effizienten Struktur der Seite gesprochen werden.

2.1.4 Zufriedenheit

Beim Erreichen des Ziels soll der Benutzer zufrieden sein. Zufriedenheit wird durch die Erwartungen des Nutzers an die Webseite beeinflusst. Diese sollten für jede Webseite bei der Entwicklung bekannt sein und überprüft werden.

Effektivität und Effizienz tragen zur Zufriedenheit bei, wobei auch hierfür Erwartungen gebildet werden, die der Benutzer erfüllt sehen will. Faktoren wie Ladezeiten und visuelle Gestaltung, die Effizienz und Effektivität beeinflussen, werden dabei subjektiv bewertet. Nach von Gizycki kann sich Zufriedenheit sowohl aus dem Erfüllen von Erwartungen an die Größen Effektivität und Effizienz ergeben (im klassischen Sinn) als auch aus dem Übererfüllen von Erwartungen als zusätzliche Zufriedenheit (im Sinne der ISO-Norm) [vgl. von Gizycki 2002, S. 4].

Der Benutzer ist bei Letztgenannter positiv überrascht vom dargestellten Inhalt, von der visuellen Gestaltung oder der Struktur der Webseite. Beispielsweise können Individualisierung von Inhalten oder nicht erwartete Zusatzinformationen zur Zufriedenheit bezüglich des Inhalts beitragen. Zielgruppenspezifisch kann eine progressive ungewöhnliche Darstellung die Zufriedenheit des Nutzers mit der Gestaltung steigern. Zufriedenheit bezüglich der Struktur kann durch verschiedene Wege der Orientierung und Navigation sowie durch eine lernfähige Navigation (es werden sich häufig genutzte Wege gemerkt und als sog. „Hotlinks“ angeboten) erhöht werden.

Mit den Erwartungen ist auch ein Anspruchsniveau verbunden, welches steigt, sobald Erwartungen kontinuierlich erfüllt werden. Eine Anpassung der Webseite an die Ansprüche des Benutzers ist notwendig, um Zufriedenheit zu erhalten und im Falle einer Unternehmensseite den Kunden nicht an einen Mitbewerber zu verlieren.

Es existieren zwar empirische Erkenntnisse über grundlegende Anforderungen an Webseiten, letztlich müssen immer zweck- und zielgruppenspezifische Kriterien für Effektivität, Effizienz und Zufriedenheit beachtet werden. Deshalb sollte auch bei der Entwicklung von Webprojekten Usability Engineering (vgl. Abschnitt 1.4) angewendet werden. In der Analysephase gilt es hier eine viel heterogenere Zielgruppe zu untersuchen als bei herkömmlicher Software. Außerdem sollte auch das eigene Unternehmen daraufhin untersucht werden, in wie weit es bereit für einen Webauftritt ist. Dieser muss schließlich immer aktuell gehalten werden, was meist Veränderungen in der Unternehmensstruktur (z. B. der Aufgabengebiete von Abteilungen, neues Personal) notwendig macht [vgl. Manhartsberger & Musil 2001].

2.2 Gründe für Web Usability

„Ten years ago, the Web was exciting to people. Today it's routine. It's a tool. If it's convenient, they will use it; if not, they won't." [Nielsen & Loranger 2006, S. xv]

Nielsen bringt hier die wesentlichen Gründe für Usability im Web auf den Punkt. Sie werden in diesem Abschnitt genauer betrachtet.

2.2.1 Vielfalt des Webangebots

Mitte der neunziger Jahre gab es weniger als 10 Millionen Webseiten. Heutzutage gibt es zehnmal so viele Seiten im Internet [vgl. Nielsen & Loranger 2006, S. xx]. Mit der Anzahl der Webseiten veränderte sich auch die Einstellung der Nutzer gegenüber diesen. Der Benutzer ist weniger tolerant gegenüber schwierig zu bedienenden Seiten, da es immer eine Alternative gibt, auf der das gewünschte Ziel einfacher erreicht werden kann. War der Benutzer vor 10 Jahren noch geduldig und wartete bis zu 30 Sekunden auf den Inhalt einer Webseite, werden heute schnelle Ladezeiten, auch aufgrund der Fortschritte der Internetanbindung, erwartet. Geht dem Benutzer etwas nicht schnell genug, versucht er es auf einer anderen Webseite.

2.2.2 Das Web als Allzweckwerkzeug

Aufgrund der Vielfalt des Webangebots wird das Web für die unterschiedlichsten Bedürfnisse benutzt. Alles scheint heutzutage online möglich zu sein. Durch diese Annahme machen Benutzer das Web zu einem Werkzeug, das im täglichen Gebrauch leicht zu bedienen sein muss, um die vielfältigen Wünsche damit schnell und einfach zu befriedigen.

Nielsen vergleicht das Web heutzutage mit dem Telefon. Der Nutzer macht sich im Allgemeinen keine Gedanken über die Funktionsweise des Telefons, wenn er jemanden anrufen will. Genauso sind Webseiten heute nur Mittel zum Zweck, über dessen Funktion der Nutzer sich keine Gedanken machen will.

2.2.3 Webseiten sind gratis

Ein Webauftritt kann gleichsam einem realen Geschäft kostenlos betreten und verlassen werden. Will ein Unternehmen etwas online verkaufen, muss er den Benutzer animieren, die Webseite auch zu bedienen. Wer den Weg zur Kasse nicht benutzerfreundlich gestaltet, hat

schnell Umsatz verloren, denn die Vielfalt lässt den Benutzer genügend Alternativen offen. Aufgrund der Anonymität im Internet sind Benutzer beim Online Einkauf wesentlich kritischer und scheuen sich nicht, die Verkaufsseite unverrichteter Dinge zu verlassen.

2.2.4 Wohlwollen und Vertrauen

Nutzer betreten eine Webseite mit einem bestimmten Maß an positiver Einstellung gegenüber dieser. Dieses Maß ist sowohl benutzerabhängig (einige sind grundsätzlich skeptischer als andere) als auch abhängig von der jeweiligen Stimmung beim Besuch der Seite. Dieses Wohlwollen kann durch eine Reihe von Usability Problemen sinken, im schlimmsten Fall bis der Nutzer verärgert die Webseite verlässt. Doch nicht nur die Webseite wird der Nutzer dann meiden, auch das Image des Unternehmens, das hinter der Webseite steht, kann Schaden dadurch nehmen.

Dem gegenüber gibt es eine Reihe vertrauensbildender Maßnahmen, wodurch Benutzer die Seite gerne und auch wiederholt besuchen werden. Dazu gehören die Erwartungen der Benutzer an die Seite zu erfüllen, intuitiv bedienbar zu sein, die Geschäftsbedingungen offen zu legen (bei Online-Shops) und aktuelle, hoch qualitative Inhalte zu liefern. Umfangreichere Beschreibungen zur Vertrauensbildung sind in [Manhartsberger & Musil 2001] zu finden.

Diese webspezifischen Gründe lassen sich durch allgemeine Vorteile hoher Usability wie Kostenersparnis bis hin zu Umsatzsteigerung oder auch Zeitersparnis während der Entwicklung ergänzen [vgl. Kalbach 2003].

2.3 Verhalten der Nutzer im Web

Im vorigen Abschnitt wurde schon beschrieben, dass Benutzer das Web als einen Ort für verschiedene Aktivitäten ansehen. Hauptanliegen der Nutzer ist es, Informationen zu finden. Dabei wird das Web im Ganzen als Ressource gesehen, und die einzelnen Webseiten haben für sich meist eine geringere Bedeutung. Entsprechend hat sich das Suchverhalten von einer Suche nach der passenden Seite (Ressource) zu einer Suche nach der passenden Antwort entwickelt.

Dazu kommt, dass Web Nutzer extrem ungeduldig sind, denn sie verbringen nach einer Studie von Nielsen im Durchschnitt nur 27 Sekunden auf einer Webseite. [vgl. Nielsen & Lorange 2006, S. 22] Der Grund dafür liegt in der immensen Vielfalt im Web, inklusive überflüssigen Inhalts. Der Benutzer ist deshalb ständig bemüht schnell relevante Informationen zu filtern, um sich nicht unnötig auf „schlechten Webseiten“ aufzuhalten.

2.3.1 Information Foraging

Pirolli und Card entwickelten die *Information Foraging* Theorie über das menschliche Verhalten bei der Informationssuche. Diese basiert auf Theorien aus der Evolutionsbiologie über das Verhalten von Tieren beim Sammeln von Nahrung. Grundprinzip ist es, bei minimalem Aufwand maximalen Nutzen zu erreichen.

Dabei folgen die Nutzer auf ihrer „Jagd“ dem *Informationsduft* (Information Scent) zur Identifikation der Informationsquelle (z. B. eine Webseite oder ein Online-Dokument). Dieser Informationsduft ist eine unvollständige Wahrnehmung des Wertes, der Kosten oder des Zugangspfades zur Informationsquelle, welche aus Hinweisen wie Links, Kategorienamen oder Icons bezogen wird [vgl. Pirolli & Card 1999, S. 10]. Ist der Informationsduft stark genug, d. h. sind die Hinweise eindeutig, dann kann der Nutzer die beste Informationsquelle mit minimalen Aufwand identifizieren. Der Informationsduft wird also dafür genutzt, herauszufinden wie wertvoll die Informationsquelle ist (Rentabilität) und wie leicht sie zu erreichen ist (Verbreitung).

Das Konzept der *Informationsnahrung* (Information Diet) sagt aus, dass der Nutzer auf Basis der Wahrnehmung des Informationsdufts, eine Entscheidung über die Auswahl der Informationen trifft. Dabei entscheidet sich der Nutzer nicht für die wertvollste Informationsquelle, wenn sie schwer zu erreichen ist. Bei der Navigation im Web wägt der Nutzer zwischen Kosten und Nutzen der Information ab.

In einer weiteren Analogie zur Jagd wird jede Webseite mit einem Jagdgrund verglichen und die Informationen der Webseite als Beute. In der Natur wechseln Tiere ihren Jagdgrund, wenn der Weg zum nächsten einfach ist und es schwer wird im Aktuellen die nächste Beute zu finden. Im Web war es zwar immer schon einfach zwischen Webseiten zu wechseln, aber in den Anfängen war es aus Sicht des Information Foraging besser, länger auf einer gefundenen Webseite zu bleiben und deren Informationen zu nutzen, da die Wahrscheinlichkeit auf eine bessere Alternative relativ gering war.

Verbesserungen der Suchmaschinen machen es dem Nutzer heutzutage einfacher gute Webseiten zu finden. Nach dem Information Foraging verlässt der Nutzer umso schneller einer Webseite, desto einfacher er gute Alternativen finden kann. Ergebnisse einer aktuellen Studie von Nielsen² bestätigen diese Hypothese. Im Durchschnitt verbrachten die Testnutzer 1 Minute 49 Sekunden auf einer Webseite, bevor sie entschieden, dass die Seite unbrauchbar war. Die Nutzer besuchten im Durchschnitt 3,2 Webseiten zusätzlich zur Suchmaschinenseite, wobei sie aber nur 0,4 Mal eine Seite wiederbesuchten. Auf der Seite, die sie als beste Informationsquelle einschätzten und ihre Aufgabe erledigten, verbrachten sie durchschnittlich 3 Minuten 49 Sekunden.

Laut Nielsen wird dieser Trend zu kurzen Besuchen von Webseiten durch die Verbreitung von Breitbandverbindungen, die den Nutzer immer online sein lassen, unterstützt. „Information snacking“ nennt Nielsen die Suche nach schnellen Antworten. Dadurch werden Nutzer häufiger das Web benutzen, somit auch erfahrener im Umgang damit werden. Die Besuche einzelner Webseiten werden dadurch noch kürzer, da erfahrene Nutzer weniger Zeit für die Entscheidung für oder gegen die Information der Seite benötigen (*Information Diet Selection*) [vgl. Nielsen & Loranger 2006, S. 54].

² [vgl. Nielsen & Loranger 2006, S. 5ff.]

2.3.2 Drei Fakten über das Nutzerverhalten im Web nach Krug

Krug hat in drei Aussagen wesentliche Verhaltensmuster von Webnutzern zusammengefasst, welche die Theorie des Information Foraging ergänzen.

We don't read pages. We scan them.

In durchschnittlich weniger als 30 Sekunden, die Nutzer auf einer Seite (sowohl Homepage als auch interne Seite) verbringen, scannen sie im Allgemeinen den Text nach Informationen ab. Der Fokus liegt dabei auf Wörtern und Sätzen, die entweder relevant für die aktuelle Aufgabe sind oder einem persönlichen Interesse entsprechen. Zusätzlich werden bestimmte Wörter wie *gratis*, *Ausverkauf*, *Sex* oder der *eigene Name* vorrangig wahrgenommen. Alles andere (insbesondere was nach Werbung aussieht) wird ignoriert. Nutzer verhalten sich so, weil sie beim Besuch einer Seite nicht wissen können, ob sie hier ihre Aufgaben erledigen können, und keine Zeit verschwenden wollen, wenn es nicht so ist.

We don't make optimal choices. We satisfice.

Meist wählen Nutzer nicht die beste Möglichkeit aus, sondern nehmen die erste vernünftige Option. Diese Strategie genannt *Satisficing*³ wenden Nutzer im Web an, weil für eine optimale Entscheidung mehr Zeit und Aufwand notwendig wäre. Zudem kann eine falsche Entscheidung durch ein oder zwei Klicks auf den Zurück-Button wieder rückgängig gemacht werden. Im Allgemeinen sind solche spontanen Entscheidungen effizienter, auch wenn vielleicht mehrere Schätzungen notwendig sind. Dennoch gibt es auch Situationen, in denen Nutzer ihre Entscheidungen abwägen bevor sie klicken (z. B. wenn sie der Webseite vertrauen).

We don't figure out how things work. We muddle through.

Designer und Entwickler machen häufig den Fehler, anzunehmen, andere Webnutzer würden die Webseite genauso nutzen wie sie. Nutzertests haben aber schon oft das Gegenteil belegt, da die Teilnehmer die Webseite auf völlig unerwartete Weise nutzten. Dazu kommt, dass sie dies meist tun, ohne ein Verständnis davon zu haben, wie sie funktioniert oder sie haben ein völlig falsches Verständnis über die Funktionsweise (siehe auch Abschnitt 1.3.2).

Dennoch erledigen Nutzer in 60 % der Fälle⁴ ihre Aufgaben erfolgreich im Web (wenn sie bekannte Seiten verwenden ist die Erfolgsrate noch höher). Menschen kreieren ihre eigenen Vorstellungen darüber, warum Technologie (in diesem Fall Webseiten) funktioniert, so wie sie diese benutzen. Die wenigstens lesen vorher ein Handbuch darüber. Dies betrifft sowohl Anfänger im Internet als auch technisch versierte Webnutzer.

Ein prägendes Beispiel für dieses Verhalten ist die Nutzung des Suchfeldes auf Webportalen wie Yahoo oder AOL. Einige Nutzer geben in dieses Suchfeld die komplette URL der Seite ein, zu der sie wollen, und das nicht nur beim ersten Mal, sondern jedes Mal. Sie haben das Verständnis, dass dies der Weg ist, wie das Internet genutzt wird oder sie glauben Yahoo/AOL seien das Internet. [vgl. Krug 2006, S. 27]

³ Eine Zusammensetzung aus *satisfying* (befriedigend) und *sufficing* (genügend), geprägt durch Herbert A. Simon.

⁴ Die Rate lag bei Vorgabe einer konkreten Webseite sogar bei 66 %. [vgl. Nielsen & Loranger 2006, S. 24]

2.4 Allgemeine Web Usability Richtlinien

Per Definition ist Usability ein recht abstrakter Begriff. Für die Bewertung von Usability müssen messbare Kriterien definiert werden. Auf dem Nutzerverhalten im Web (siehe voriger Abschnitt) basieren eine Reihe von allgemeinen Richtlinien. Dazu gibt es welche für spezielle Arten von Seiten (z. B. E-Commerce) oder für spezielle Altersgruppen. Letztlich entstehen aus Nutzeranalysen und Nutzertests für einzelne Webseiten zusätzlich spezifische Richtlinien, die nur für diese Seiten gelten.

Es kann an dieser Stelle nur eine Auswahl von allgemeinen Usability Richtlinien präsentiert werden. Die Auswahl wurde nach den häufigsten Ursachen für das Scheitern von Nutzern im Web (Abbildung 3) und nach Relevanz für die in Kapitel 4 und 5 vorgestellte Webanwendung getroffen.

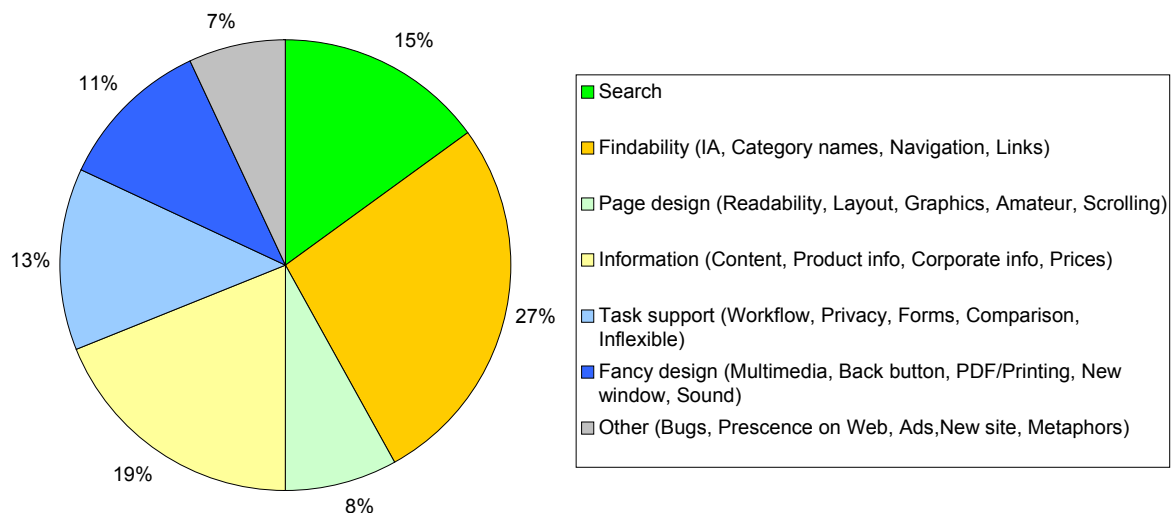


Abbildung 3: Kategorien von Design Problemen, die Nutzer an ihren Aufgaben im Web scheitern lassen nach [Nielsen & Loranger 2006, S. 133]

2.4.1 Informationsarchitektur⁵ und Navigation

Die Struktur einer Webseite sollte den Erwartungen der Benutzer entsprechen.

Benutzer haben beim Besuch einer Webseite aufgrund bisheriger Erfahrungen im Web und ihren Wünschen an die betreffende Seite bestimmte Erwartungen an die Struktur der Seite, d. h. wo bestimmte Informationen zu finden sind und wie sie dorthin gelangen. Eine nutzerge-rechte Kategorisierung deckt sich meist nicht mit der, die den Designer oder Firmenchef vertraut ist. So sollten Produkte beispielsweise nicht nach Herstellernamen organisiert werden, sondern nach Attributen, die für den Nutzer entscheidend sind (Größe, Preis etc.).

⁵ Die Organisation von Information, um diese auffindbar zu machen. Ausführliche Informationen zu diesem Thema in [Rosenfeld & Morville 1998]

Die Navigation sollte konsistent sein und den Konventionen entsprechen.

Damit hilft sie Benutzern ihren aktuellen Standort auf der Webseite klar zu lokalisieren und die nächstmöglichen Schritte zu erkennen. Schnell auffindbar ist die Navigation, wenn sie nach der Konvention im Web links oder oben platziert ist. Zudem sollten die Navigationseinträge nicht ändern innerhalb einer Webseite und stets an derselben Position zu finden sein. Die Hauptnavigation sollte dabei immer komplett sichtbar sein. Somit ist sie für den Benutzer vorhersehbar und Optionen können schneller wahrgenommen werden, da sie sofort sichtbar sind.

Redundanz in der Navigation sollte vermieden werden.

Mehrere Navigationsbereiche mit ähnlichen oder denselben Links machen es dem Nutzer unnötig schwer, die Struktur der Seite zu erkennen. Weniger Elemente auf einer Seite sind schneller vom Benutzer erfassbar und auch merkbar.

Links und Kategorienamen sollten konkrete Aussagen treffen.

Die Beschriftung von Links und Kategorien sollte genau beschreiben, was der Benutzer nach dem Klicken darauf erwarten kann. Daher sollten allgemeine Begriffe wie *weiter* oder *hier klicken* aber auch clever wirkende Redewendungen oder Slogans nicht verwendet werden. Besser sind dagegen kurze mit einem Schlüsselwort beginnende Links. In einer Liste ist es für den Nutzer schneller wahrnehmbar, wenn die Links nicht alle mit dem gleichen Wort beginnen.

Links sollten sich klar vom Inhalt unterscheiden.

Ein Link sollte nicht nur darauf hinweisen, wohin er führt, sondern auch mitteilen, wenn die betreffende Seite gerade besucht wird und dann nicht mehr aktiv sein. In allen anderen Fällen muss ein Link klar als solcher erkennbar sein. Dies bedeutet heutzutage nicht mehr, dass er zwingend blau und unterstrichen sein muss, aber umgekehrt sollte normaler Text nicht blau gestaltet werden, weil viele Benutzer eine Verlinkung damit assoziieren.

2.4.2 Information

Information sollte so prägnant wie möglich geboten werden.

Die meisten Erwachsenen können 200 bis 300 Wörter in der Minute lesen. Das würde für eine Homepage einen Text von etwa 100 Wörtern bedeuten, da sie durchschnittlich 30 Sekunden betrachtet wird. Realistisch sind eher 10 bis 20 Wörter für die Botschaft, was der Zweck der Seite ist, da auch noch Zeit zur Orientierung und Navigation benötigt wird. Interne Seiten, die 45 bis 60 Sekunden (beim ersten Besuch) betrachtet werden, könnte der Nutzer theoretisch 200 Wörter lesen, praktisch sind es aber auch nicht mehr als 100. Auch hier benötigt der Nutzer Zeit für die Entscheidung, wohin er als nächstes navigieren möchte.

Nutzergerechte, einfache Sprache sollte verwendet werden.

Der Inhalt sollte an das Leseniveau und die Lesegewohnheiten der Nutzer angepasst werden. Dabei mögen die meisten Nutzer eher einen weniger formalen Tonfall, da er direkter und persönlicher wirkt. Umgangssprache, ironische oder klischeehafte Ausdrücke, Fachausdrücke (die der Nutzer nicht versteht) und Marketing-Slogans sind zu vermeiden. Stattdessen sollten die Fakten klar dargelegt werden, die, wenn sie gut sind, auch ohne Beiwerk überzeugen.

Der Schreibstil sollte dem umgekehrten Pyramidenstil entsprechen.

Dabei wird mit einer Zusammenfassung begonnen und danach werden die entsprechenden Fakten dazu dargelegt. Dies erleichtert den Nutzer den Inhalt zu scannen und nur bei Bedarf in die Details einzutauchen.

Formatierung des Textes sollte die Überschaubarkeit des Inhalts verbessern.

Einige Formatierungstechniken erleichtern die Lesbarkeit des Inhalts und ermöglichen es dem Nutzer relevante Bereiche schneller zu erfassen.

- Überschriften sollten größer und fetter als der Textbereich dargestellt werden und max. 60 Zeichen umfassen.
- Schlüsselwörter im Text können hervorgehoben besser wahrgenommen werden. Ganze Sätze oder Abschnitte hervorzuheben verlangsamt eher die Lesegeschwindigkeit.
- Nummerierte oder Aufzählungslisten helfen Schrittfolgen und Anweisungen prägnanten darzustellen. Dabei sollte der Stil der Elemente konsistent gehalten werden (z. B. alle beginnen mit einem Verb).
- Inhalt sollte in kurze Absätze, die etwa fünf Sätze umfassen, zu einem Stichwort aufgeteilt werden.

Einen Überblick zum Schreiben für das Web liefert ein Artikel von [Morkes & Nielsen 1997].

2.4.3 Faktoren, die Usability Richtlinien beeinflussen

Obwohl Usability Richtlinien meist längerfristig gültig sind, gibt es einige Faktoren, die ihre Bedeutung beeinflussen können.

Technologische Veränderungen

Die drastischen Veränderungen in der Browsertechnologie, den Auszeichnungs- und Skriptsprachen zur Webseitenerstellung (HTML, JavaScript, CSS) und die Verbreitung der Breitband-Anschlüsse (DSL, Kabelmodem) in den letzten 10 Jahren scheinen auch radikalen Veränderungen zum Thema Usability zu erfordern. Dies ist nicht der Fall. Usability begründet sich auf menschlichen Verhalten und weniger auf technologischen Erkenntnissen. Die Charakteristik der Web Nutzer hat sich in den letzten Jahren nicht geändert. Dennoch gibt es einige Usability Probleme, die durch den technischen Fortschritt verringert wurden, wie beispielsweise Ladezeiten, Frames (moderne Webbrowser verarbeiten diese nutzerfreundlicher), Multimedia und Videos (durch höhere Bandbreite akzeptable Downloadzeiten).

Anpassung der Nutzer

Die Charakteristik der Web Nutzer hat sich seit Beginn des Internets nicht verändert. Aber sie lernten Webseiten besser zu bedienen. Beispielsweise haben sie sich an die Konventionen für Links (farbig und unterstrichen oder 3D Effekt bei Grafiken) gewöhnt. Durch diese Anpassung sind früher wichtige Usability Richtlinien wie „Links müssen blau und unterstrichen sein“ abgeschwächt worden, da jetzt auch andere Farben dafür funktionieren.

Erfahrung der Designer

Auch Designer haben in den letzten Jahren durch ihre Erfahrungen im Web-Design dazu beigetragen, dass einige Usability Probleme gemildert worden. Sie haben Zurückhaltung bezüglich ausgefallener Technologie und individuellen Anpassungen von GUI-Elementen (z. B. Scrollbar) bewiesen und mehr auf konventionelle Techniken, die dem Benutzer vertraut sind, gesetzt.

Richtlinien können hilfreich beim Erreichen von mehr Usability sein, sind aber keine Faustregeln für alle Webseiten. Die Wirkung der allgemeinen Richtlinien muss für das individuelle Projekt überprüft werden (Usability Tests) und mit eigenen Regeln auf Grundlage von Nutzeranalysen ergänzt werden.

3 Methoden der Usability Evaluation

3.1 Begriffe und Klassifizierung

3.1.1 Usability Evaluation

„The most basic advice with respect to interface evaluation is simply to *do it*, and especially to conduct some user testing.“
[Jakob Nielsen 1993]

Die klassischen Methoden der Usability Evaluation haben ihre Wurzeln in psychologischen Forschungsmethoden, die aufgrund ihrer Komplexität nicht den erwünschten Einzug in die Praxis der Produktentwicklung gehalten haben, weil sie im Sinne des obigen Zitats von Nielsen „nicht einfach gemacht“ werden können. Dabei bietet das Konzept des Usability Engineering Lifecycle mehrere Möglichkeiten, Untersuchungen durchzuführen und die Auswirkungen von Veränderungen in der Gestaltung aufgrund der Evaluationsergebnisse zu kontrollieren.

- Analyse von Konkurrenzprodukten
- Analyse von Design Entwürfen im iterativen Design Prozess
- Analyse von Prototypen des Produkts
- Analyse des Endprodukts

Bevor eine Evaluation durchgeführt wird, sollten klar die Ziele der Evaluation definiert werden, da sie einen Einfluss auf die Auswahl der Methode haben. Nielsen unterscheidet dabei zwischen *formative evaluation* und *summative evaluation*. [vgl. Nielsen 1993, S. 170] Die formative Evaluation wird verwendet, um das Interface im Rahmen eines iterativen Design Prozesses zu verbessern. Dabei wird Wert darauf gelegt, welche einzelnen Aspekte des Interface gut oder schlecht sind und wie es verbessert werden kann. Die summative Evaluation zielt auf die Qualität der Benutzeroberfläche insgesamt, um sich beispielsweise zwischen alternativen Entwürfen zu entscheiden oder Konkurrenzprodukte zu vergleichen. Weitere Ziele könnte die Gewinnung und empirische Bestätigung neuer Usability Richtlinien sein (im Rahmen einer Reihe von Usability Evaluationen). Ein wünschenswerter Nebeneffekt von Usability Evaluationen ist die Schulung des beteiligten Entwicklerteams in Usability Richtlinien und die dauerhafte Integration von Usability Maßnahmen im Entwicklungsprozess.

In diesem Kapitel werden die Methoden der Usability kategorisiert und Ausgewählte detailliert vorgestellt (siehe Abschnitte 3.2 und 3.3). Außerdem wird eine Diskussion über die Effektivität und Tauglichkeit der Methoden bezüglich der Ziele der Usability Evaluation durchgeführt (siehe Abschnitt 3.4). Als Einstieg werden Kriterien, die bei Evaluationen beachtet werden sollten, erläutert.

3.1.2 Kriterien der empirischen Forschung

Die Kriterien der empirischen Forschung gelten für Usability Evaluationen, da auch ihre Ergebnisse im wissenschaftlichen Sinne zur Gewinnung und zum Beleg von Theorien und Hypothesen herangezogen werden. Aber auch wenn Untersuchungen nur projektintern durchgeführt werden, sollten dennoch diese Kriterien beachtet werden, da es zur Anerkennung der Maßnahmen des Usability Engineering beiträgt.

Objektivität

„Die Grundforderung an die wissenschaftliche Forschung ist die Forderung nach Objektivität.“ [Zimbardo & Gerrig 1999, S. 18] Dieser Anspruch ist erreicht, wenn die Daten und Ergebnisse durch andere Forscher nachgeprüft werden können. In einer objektiven Untersuchung sollte der Forscher Verzerrungen durch den Einfluss persönlicher Vorlieben gering halten oder ausschalten. Die Datenerhebungen und -analysen sollten umfassend dokumentiert werden, so dass das Vorgehen von anderen verstanden werden kann und es möglich ist, die Untersuchung zu einem anderen Zeitpunkt, an einem anderen Ort und mit anderen Teilnehmern zu wiederholen. Objektivität ist die Voraussetzung für Reliabilität und Validität.

Reliabilität (Zuverlässigkeit)

„Reliabilität bezeichnet die Genauigkeit der Messung.“ [Zimbardo & Gerrig 1999, S. 22] Während zur Objektivität gehört, dass die Dokumentation so durchgeführt wird, dass andere Forscher die Untersuchung wiederholen können, bedeutet Reliabilität, dass die Wiederholung unter gleichen Bedingungen identische Resultate erzielt. Wenn die Untersuchungsbedingungen standardisiert sind und Quellen unerwünschter Variation kontrolliert werden, erhöht sich die Zuverlässigkeit. Außerdem erhöht eine hohe Anzahl an Beobachtungen die Reliabilität, da einzelne Untypische das Gesamtbild nicht verzerren.

Validität (Gültigkeit)

Eine Evaluation ist dann valide, wenn sie tatsächlich das Merkmal untersucht, welche für die Messung durch den Forscher festgelegt wurde. Eine Usability Evaluation ist somit valide, wenn sie tatsächlich die Usability des Produkts untersucht und beispielsweise nicht nur die Bedienbarkeit des Produkts. Um einer Untersuchung Validität zu geben, muss das zu untersuchende Merkmal vorher sorgfältig analysiert werden und es sollte ein theoretisches Verständnis für das Merkmal vorhanden sein.

Validität setzt Reliabilität voraus. Es ist aber keine Garantie für hohe Validität, wenn die Untersuchung zuverlässig ist. Dann kann immer noch nicht das festgelegte Merkmal Gegenstand der Untersuchung gewesen sein.

Diese Kriterien sind mehr oder weniger ausgeprägte Eigenschaften der Evaluationsmethoden. Probleme bei der Gewährleistung von Objektivität, Reliabilität und Validität werden bei den vorgestellten Methoden kurz diskutiert. Es sollte klar werden, dass es keine *perfekte* Evaluation geben kann, da die Rahmenbedingungen immer neu durch die Teilnehmer definiert werden und diese menschliche Komponente nicht hundertprozentig kontrollierbar ist.

3.1.3 Klassifikation der Methoden

Die Methoden der Usability Evaluation lassen sich in zwei Kategorien einteilen. Bei den *expertenorientierte (analytische) Methoden* begutachten Experten (Abschnitt 3.2.1) die Benutzerschnittstelle (User Interface) eines Produkts und bewerten dessen Benutzertauglichkeit. Dabei untersuchen sie entweder einzelne Funktionen des Produkts oder überprüfen das Design insgesamt auf Übereinstimmung mit anerkannten Usability Richtlinien. Im Abschnitt 3.2 wird ein Überblick über allgemeine Merkmale der expertenorientierten Methoden gegeben. Mit der Heuristischen Evaluation wird im Abschnitt 3.2.2 die bekannteste expertenorientierte Methode vorgestellt. Der Cognitive Walkthrough ist eine Methode dieser Kategorie für den Schwerpunkt Erlernbarkeit und der Abschnitt 3.2.3 beschreibt dessen Besonderheiten.

In den *nutzerorientierte (empirische) Methoden* beurteilen Mitglieder der Zielgruppe (Endnutzer) des Produkts dessen Usability. Diese Methoden werden auch unter dem Begriff *User Testing (Nutzertest)* zusammengefasst, obwohl er nur ungenau die Bandbreite der nutzerorientierten Methoden widerspiegelt. Unter User Testing sollen hier die Methoden verstanden werden, bei denen die Benutzer mit dem Produkt interagieren und dabei deren Verhalten bezüglich auftretender Probleme untersucht wird (simultane Methoden). Als wichtige und effiziente simultane Methode wird der Nutzertest ausführlich in Abschnitt 3.3.1 mit dem häufig verwendeten Verfahren des Lauten Denkens (Thinking Aloud) vorgestellt. Daneben gibt es aber auch sog. retrospektive Methoden, die nach der Interaktion mit dem Produkt veranstaltet werden. Dazu gehören Umfragen, Interviews und Fokusgruppen. Letztere wird beispielhaft für diese Art der Methoden vorgestellt (siehe Abschnitt 3.3.2).⁶

3.2 Expertenorientierte Methoden

3.2.1 Allgemeine Aspekte

Bei der Auswahl der geeigneten Methode, ist es wichtig vorher Ziele zu definieren, die mit der Evaluation erreicht werden sollen. Es muss festgelegt, welche Aspekte des Produkts die Usability beeinflussen, da diese Gegenstand der Untersuchung sein sollten. Bei der Planung werden *Richtlinien* festgelegt, *Experten* rekrutiert und der *Untersuchungsrahmen* festgelegt.

Experten

Welche Experten in der Evaluation eingesetzt werden hängt einerseits von der Verfügbarkeit ab, andererseits spielen auch Zielsetzungen für die entsprechende Untersuchung eine Rolle. Ein ursprüngliches Ziel war es, Usability Spezialisten die Möglichkeit zu geben, selbst die Evaluation durchzuführen. [vgl. Nielsen & Mack 1994, S. 10]

⁶ Auf eine Übersicht weiterer Methoden wird in dieser Arbeit verzichtet. Stattdessen sei auf die Fachliteratur im Literaturverzeichnis verwiesen.

Experten können in folgende Gruppen eingeteilt werden:

- *Usability-Experten* sind auf dem Gebiet der Usability, insbesondere der Usability Evaluation, spezialisiert;
- *Fach-Experten* sind spezialisiert auf dem Gebiet, in dem das Produkt eingesetzt wird (z. B. Finanzen);
- *System-Experten* sind Entwickler oder Designer, die Erfahrungen und Kenntnisse auf dem Gebiet der Produktentwicklung und -gestaltung haben (z. B. Software-Entwickler);
- *Doppel-Experten* sind Spezialisten auf dem Gebiet der Usability und einem weiteren Fachgebiet.

Vorteil der Usability-Experten ist, dass sie aufgrund ihrer Erfahrung meist effektiver an einer Evaluation arbeiten. Aber sie sollten auch die Folgen ihrer Vorschläge für die Entwicklung einschätzen können. Fach- oder System-Experten werden dann eingesetzt, wenn die Evaluation nicht nur Probleme aufdecken soll, sondern gleichzeitig den Gutachtern die Idee des benutzerzentrierten Interface näher zu bringen soll. Ist ein Produkt sehr fachspezifisch kann es sinnvoll sein, Experten auf diesem Gebiet einzusetzen. Ein gemischtes Team der oben genannten Experten kann durch die unterschiedlichen Perspektiven der Spezialisten auf Probleme des Produkts die Evaluation erfolgreicher werden lassen. Letztlich muss die Auswahl der Evaluatoren für jedes Produkt neu überdacht werden. Denn ein Experte, der einmal gute Ergebnisse lieferte, muss bei der nächsten Evaluation nicht genauso gut geeignet sein.

Das zusammengestellte Team aus Experten kann entweder gemeinsam das Produkt evaluieren oder es untersucht jeder einzeln die Benutzeroberfläche. Dabei gilt es die Vor- und Nachteile beider Varianten abzuwägen. Arbeiten die Prüfer alleine, so wird ihr Gutachten nicht durch die Meinung der anderen beeinflusst. Im Team können sich die Experten gegenseitig beeinflussen und die Evaluation kann dadurch in eine bestimmte Richtung gelenkt werden. Die Arbeit im Team hat aber den Vorteil, dass durch den Meinungsaustausch mehr Probleme entdeckt und kreative Lösungsvorschläge gefunden werden. Außerdem muss der Prüfer im Team seine Beurteilung begründen, wodurch objektivere Aussagen zu den gefundenen Problemen gemacht werden. Letztlich nimmt die Arbeit im Team aber auch mehr Zeit wegen der Diskussion der Probleme in Anspruch.

Untersuchungsrahmen – Szenarios

Die Gutachter untersuchen das Interface entweder unter Vorgabe von Szenarios oder können sich frei durch das Interface bewegen. Szenarios sind Vorgaben, wie typische Aufgaben von Endnutzern durchgeführt werden. Sie repräsentieren also für den Experten die Art wie Endnutzer das System verwenden werden. Dabei werden Experten auf bestimmte Aspekte des Interface hingewiesen und es wird ihnen eine aufgabenorientierte Sicht ermöglicht. Diese ist besonders im Cognitive Walkthrough von Bedeutung, wie der Abschnitt 3.2.3 zeigen wird. In sehr fachspezifischen und komplexen Systemen können Szenarios die Fachkenntnisse bieten, die benötigt werden, um das System zu bedienen und die der Experte vielleicht nicht besitzt.

Es gibt aber auch Argumente, die für eine freie Erkundung des Interface durch den Experten sprechen. Endnutzer folgen auch keinen Szenarios, sie setzen sich Ziele und versuchen diese mit den Möglichkeiten des Interface zu erreichen. Außerdem schränken Szenarios die Sicht des Experten auf das Interface ein, indem er nur vorgegebene Aufgaben ausprobiert und andere Teile des Systems nicht entdeckt. Wenn keine Szenarios gegeben sind, erforscht der Experte meist trotzdem das Interface in Bezug auf sinnvolle Aufgaben für den Endnutzer, ohne dabei darauf begrenzt zu sein.

3.2.2 Heuristische Evaluation

Die Heuristische Evaluation ist eine Methode, bei der wenige Prüfer (Experten) anhand von Usability Richtlinien (Heuristiken) das User Interface Design untersuchen, um Probleme zu finden, die im interaktiven Design Prozess des Usability Engineering behoben werden sollten. [vgl. Nielsen 1994, S. 25f.] Im Gegensatz zu anderen Methoden (z. B. Cognitive Walkthrough) wird das gesamte Erscheinungsbild des Interface begutachtet.

Gutachter

Für die Heuristische Evaluation wird ein kleines Team von Experten als Gutachter zusammengestellt. Studien⁷ haben gezeigt, dass einzelne Gutachter nur etwa 35 % der Usability Probleme eines Produkts finden. Mehrere Prüfer finden hingegen nicht nur mehr Probleme, sondern ihre Ergebnisse sind auch unterschiedlich, so dass die Evaluation dadurch effektiver wird. Einfache, offensichtliche Probleme werden dabei von den meisten Evaluatoren gefunden; schwerwiegende Probleme werden nur noch von wenigen Prüfern gefunden. Letztgenannte finden aber nicht zwangsläufig die meisten Probleme.

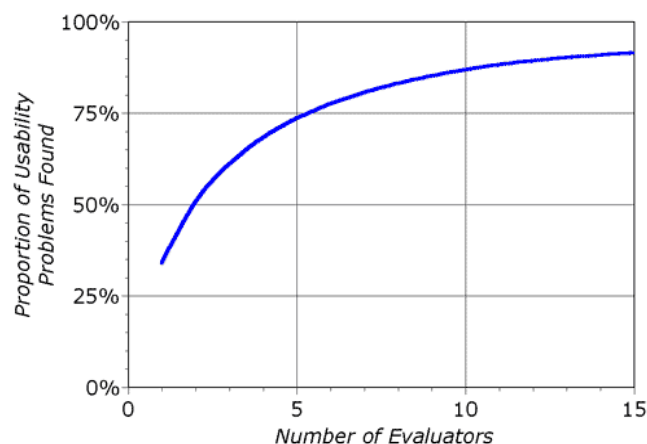


Abbildung 4: Zusammenhang zwischen der Anzahl der Prüfer und der Menge an gefundenen Usability Problemen (in Prozent) (Quelle: [Nielsen ca. 1994])

Schon zwei Experten finden mehr als 50 % der Probleme (siehe Abbildung 4). Nielsen empfiehlt in den meisten Projekten zwischen drei und fünf Prüfer einzusetzen. Der weitere Einsatz von Evaluatoren erhöht nur minimal die Ergebnismenge der Probleme, kann aber zu unver-

⁷ Nielsen begründet seine Aussagen durch sechs Projekte, in denen die Heuristische Evaluation angewendet wurde [vgl. Nielsen 1994, S. 32]. Ähnliche Aussagen trifft [Schweibenz 2003, S. 103] aufgrund von Ergebnissen von L. Kantner und S. Rosenbaum von 1997.

hältnismäßig höheren Kosten führen, da den Gutachtern in den meisten Fällen ein Honorar bezahlt wird. Eine Kosten-Nutzen Rechnung für die Heuristische Evaluation sollte bei der Wahl der optimalen Anzahl von Evaluatoren helfen. Eine größere Anzahl an Prüfern kann sich für Projekte lohnen, bei denen Usability eine kritische Eigenschaft des Produkts ist (z. B. medizinische Apparate) oder erhebliche Mehrkosten (im Vergleich zu den Kosten für die Prüfer) durch schlechte Usability entstehen.

Heuristiken

Nielsen prägte den Begriff *heuristics* für Richtlinien, die bei der Gestaltung eines benutzertauglichen Interface beachtet werden sollten. Die Heuristiken basieren auf empirischen Untersuchungen der Usability und allgemeinen psychologischer Grundlagen und wurden von Usability-Experten (siehe Abschnitt 1.3) zusammengestellt. Es sind derzeit vielfältige Listen von Richtlinien im Umlauf und es muss für jede Heuristische Evaluation eine individuelle Auswahl dieser getroffen werden.

Dabei gilt es besonders allgemeine Prinzipien der Mensch-Computer-Interaktion in konkrete Richtlinien umzuwandeln, die für die Untersuchung des entsprechenden Produkts geeignet sind. Prinzipien sind meist zu allgemein formuliert und stellen nur abstrakte Zielvorgaben dar. Das Prinzip *eine benutzertaugliches Produkt muss vom Nutzer effizient nutzbar sein* sollte sich beispielsweise für das Produkt Webseite in einer Richtlinie wie *die Navigation sollte konsistent sein und Redundanz sollte vermieden werden* widerspiegeln.

Richtlinien können selbst noch unterschiedlich konkret sein. Dabei gibt es allgemeine Richtlinien, die auf Kenntnissen über die menschliche Informationsverarbeitung beruhen und für ein breites Spektrum an Produkten gelten (z. B. Richtlinien über die Lesbarkeit auf Bildschirmen). Categoriespezifische oder nutzerspezifische Heuristiken gelten nur für eine bestimmte Produktgruppe. Solche Richtlinien lassen sich durch Mitbewerber-Analysen oder Nutzertests an bestehenden Produkten gleicher Art gewinnen, indem abstrakte Regeln gebildet werden, die die gefundenen Usability Probleme erklären. Dennoch kann es wichtig sein, für die eigene Heuristische Evaluation auch produktspezifische Richtlinien zu entwickeln, die sich aus Nutzer- und Aufgabenanalysen ergeben und nur für dieses Produkt gültig sind.

Für jede Heuristische Evaluation muss nicht nur die Auswahl der Heuristiken stimmen, sondern auch der Umfang an Richtlinien, auf welche die Prüfer das Produkt untersuchen sollen. Sind die Heuristiken sehr detailliert und umfangreich, kann sich das negativ auf den Erfolg der Evaluation auswirken, da die Prüfer zu viele Aspekte zu berücksichtigen haben. Im Allgemeinen werden Heuristiken für die Evaluation als Anweisungen oder Fragen formuliert, die durch den Prüfer mit Ja oder Nein beantwortet werden können. Es kann aber auch eine Bewertungsskala vorgegeben werden, mit welcher der Prüfer angeben kann, in wie weit die Regel verletzt wurde.

Durchführung

Bei der Heuristischen Evaluation untersucht jeder Prüfer allein die Benutzeroberfläche. Erst nachdem alle Prüfer ihre Untersuchung abgeschlossen haben, sollte es ihnen erlaubt sein, miteinander zu sprechen, um ihre Ergebnisse auszutauschen. Eine Heuristische Evaluation im

Team wird selten praktiziert. Stattdessen wird eine Nachbesprechung empfohlen, wie weiter unten näher erläutert wird.

Während der Evaluation begutachtet der Prüfer die einzelnen Dialogelemente der Benutzeroberfläche, indem er mehrmals systematisch das User Interface durchläuft. Dabei beurteilt er die Usability der Elemente und des Gesamtsystems anhand der ihm vorliegenden Heuristiken. Der Prüfer kann zusätzlich auch andere ihm bekannte Usability Richtlinien berücksichtigen, die relevant für das entsprechende Dialogelement sein könnten. Obwohl die Prüfer selbst entscheiden können, wie sie bei der Untersuchung vorgehen, hat es sich bewährt, mindestens zweimal das Interface zu durchlaufen. Die erste Phase dient dabei zum Kennen lernen der Interaktionsmöglichkeit und des gesamten Systems. In der zweiten Phase kann sich der Prüfer auf einzelne Elemente konzentrieren mit dem Wissen, wie sie in das Gesamtsystem einzuordnen sind.

Die Ergebnisse können entweder durch den Prüfer selbstständig aufgezeichnet werden oder ein Beobachter notiert die Kommentare des Prüfers. Bei eigenen Aufzeichnungen des Prüfers muss dieser einen gewissen Mehraufwand betreiben und die Berichte aller Prüfer müssen von einem Untersuchungsleiter gelesen und zusammengefasst werden. Ein Beobachter während der Untersuchung ermöglicht eine schnelle Zusammenfassung der Ergebnisse, da er nur seine eigenen Notizen zu sichten hat. Im Gegensatz zum Nutzertest (vgl. Abschnitt 3.3.) muss der Beobachter nicht das Verhalten der Prüfer beurteilen und mit Usability ins Verhältnis setzen, sondern lediglich die Kommentare des Prüfers aufzeichnen. Denn es ist die Aufgabe des Prüfers das Interface zu analysieren. Außerdem kann der Beobachter den Prüfern helfen, wenn es Probleme gibt, beispielsweise aufgrund geringer Kenntnisse auf dem jeweiligen Einsatzgebiet des Produkts. Auch hier besteht ein grundsätzlicher Unterschied zum Nutzertest, da dort die Hilfestellung des Beobachters (Untersuchungsleiters) nur minimal ausfallen darf, da die Nutzer durch den Gebrauch des Produkts Antworten auf ihre Fragen finden sollen. Den Prüfern hingegen sollte in fachspezifischen Fragen und Problemen im Umgang mit dem Produkt immer geholfen werden, da es die Effizienz der Untersuchung steigert, wenn der Prüfer die vorgegebene Zeit nicht damit verschwendet, die Bedienung des Produkts kennen zu lernen.

Besonders wenn das System für ein bestimmtes Fachgebiet bestimmt ist und die Prüfer keine Experten auf diesem Gebiet sind, ist es notwendig, sie in das System einzuweisen, damit sie es überhaupt bedienen können. Eine Möglichkeit für das Kennen lernen des Systems ist, den Prüfer ein typisches Nutzungsszenario zu liefern, indem die einzelnen Schritte aufgelistet werden, die ein Nutzer ausführen würde, um realistisch Aufgaben damit zu erledigen. Solch ein Szenario kann auf Basis von Aufgabenanalysen zu Beginn des Usability Engineering Lifecycle erstellt werden (siehe Abschnitt 1.4).

Die Dauer der Evaluation ist abhängig vom Umfang der zu untersuchenden Benutzeroberfläche. In der Regel kann aber mit einer Untersuchungsdauer von ein bis zwei Stunden pro Prüfer gerechnet werden [vgl. Nielsen 1994]. Falls das User Interface eine längere Evaluation notwendig macht, sollten lieber mehrere kürzere Untersuchungen für einzelne Bereiche der Benutzeroberfläche durchgeführt werden.

Auswertung

Für die Auswertung ist es wichtig, dass die gefundenen Usability Probleme zu den Prinzipien notiert werden, die dabei verletzt wurden. Bei der Formulierung der Problembeschreibung sollte der Evaluator genau erläutern, warum das Element seiner Meinung nach gegen diese Richtlinie verstößt. Dabei gilt es zu beachten, dass ein Dialogelement Probleme bezüglich mehrerer Richtlinien verursachen kann und deshalb sollte jedes einzelne Problem entsprechend notiert werden. Diese Aufspaltung der Probleme eines Elements hat zwei Vorteile für die Auswertung der Evaluation und daraus folgend für die Verbesserung des Produkts. Wenn nicht jedes Problem extra vermerkt wird, besteht die Gefahr, dass in der Weiterentwicklung des Produkts Probleme (obwohl sie erkannt wurden) verschleppt werden und weiterhin die Usability schmälern. Zwar ist es meist nicht möglich alle Probleme im nächsten Design zu eliminieren, aber eine detaillierte Auflistung macht es einfacher, Prioritäten für die Beseitigung zu setzen und zumindest das betreffende Element teilweise benutzertauglicher zu machen.

Zur Auswertung einer Heuristischen Evaluation gehört auch die Bewertung der Ernsthaftigkeit der gefundenen Usability Probleme. Dazu sollten alle Gutachter nach der getrennten Untersuchung eine Liste der gesamten Probleme vorgelegt werden, um sie zu bewerten. (siehe Abschnitt 3.5.1).

Beurteilung der Methode

Die Heuristische Evaluation bietet zwar eine Begründung für ein Usability Problem anhand der entsprechenden Heuristik, kann aber keinen systematischen Weg aufzeigen, um dieses zu beseitigen oder bei zukünftigen Design Entwürfen das Maß an Usability zu erhöhen. Einige Lösungen können dennoch aus den verletzten Richtlinien generiert werden oder die Lösung ist offensichtlich sobald das Problem festgestellt wurde. Beispielsweise kann das Problem der inkonsistenten Nutzung von Schriftarten behoben werden, indem nur noch eine Schriftart im gesamten User Interface verwendet wird. Dabei bietet dieser Lösungsansatz dem Designer keine konkrete Hilfe für Veränderungen der Benutzeroberfläche. Denn es bleibt offen, welche Schriftart als einzige weiterhin eingesetzt werden soll.

Abhilfe kann hier eine Erweiterung der Heuristischen Evaluation schaffen, indem eine Nachbesprechung nach der letzten Untersuchung eines Prüfers stattfindet. Daran sollten sich sowohl die Prüfer und, falls eingesetzt, der Beobachter als auch einige Mitglieder des Design-teams des Produkts beteiligen. In einer Art Brainstorming können sich die Teilnehmer über mögliche Design Änderungen austauschen und vorrangige Usability Probleme diskutieren. Eine Nachbesprechung kann auch ein Zeitpunkt sein, indem positive Aspekte des Produkts besprochen werden, da die Heuristische Evaluation sonst nicht darauf eingeht.

3.2.3 Cognitive Walkthrough

Die Methode des Cognitive Walkthrough zielt auf die Evaluation eines Produkts in Bezug auf ein Usability Attribut – leichtes Erlernen durch Erkunden. Dabei wird das Problemlösungsverhalten der Benutzer beim Umgang mit dem Produkt simuliert. Die Gutachter gehen dafür die Benutzeroberfläche nach Vorgabe bestimmter Aufgaben aus Sicht von Nutzern

durch. Der Ablauf wird durch die vorgegebenen Handlungsschritte der Produktdesigner bestimmt.

Zugrunde liegende Theorie

Der Cognitive Walkthrough basiert auf einer Theorie des Lernens durch Erkunden und auf Forschungen im Bereich Problemlösung. [vgl. Wharton et al 1994, S. 125] Beobachtungen, haben gezeigt, dass Nutzer es bevorzugen, Software durch Erkunden zu erlernen. Sie wollen lieber Funktionen während der Arbeit an ihren alltäglichen Aufgaben mit der Software lernen, als vorher darin geschult zu werden. Dieser Ansatz des Lernens stellt sicher, dass die Kosten für das Erlernen durch den gleichzeitigen Nutzen der neu erlernten Funktion für den Nutzer relativiert werden.

„Modern theories of skill acquisition [...] assume that problem-solving processes are used to discover correct actions and that the learning mechanisms store representations of correct actions with the users' current goals and task contexts.“ [Wharton et al 1994, S. 125]

Der Cognitive Walkthrough simuliert das Problemlösungsverhalten der Nutzer. Es wird ein Ziel formuliert, die nächste Aktion ausgewählt und das Ziel aufgrund der Konsequenzen der Handlung angepasst. Die Grundidee ist dabei, dass die korrekte Handlung auf Basis ihrer vermuteten Ähnlichkeit oder Relevanz mit dem Ziel des Nutzers ausgewählt wird. [vgl. Wharton et al 1994, S. 125] Die Auswahl der nächsten Aktion ist außerdem abhängig von der Vorstellungen des Nutzers über sein Ziel und den möglichen alternativen Handlungen.

Nutzer mit wenig Hintergrundwissen auf dem Fachgebiet des Produkts haben meist klare Vorstellungen von ihren Aufgaben und Zielen, aber wenig Wissen über die Konsequenzen von möglichen Aktionen. Sie wählen deshalb beispielsweise Dialogelemente für die Interaktion, deren Beschriftung am meisten ihren Zielen entsprechen. Erfahrene Nutzer zerlegen Aufgaben eher in Unteraufgaben, die es ermöglichen, das System effektiver zu nutzen. Außerdem haben sie mehr Wissen über Aktionen und ihre Konsequenzen. Dennoch wählen auch sie die Aktionen, die relevant oder ähnlich ihrem aktuellen Ziel sind, wenn auch mit mehr Hintergrundwissen.

Der kognitive Aspekt der Cognitive Walkthrough Methode wird darauf begründet, dass Annahmen über das Hintergrundwissen der Nutzer, ihre Gedanken und Ziele während des Ausführens der Aufgabe und über die Folge der Aktion, die zum Erledigen der Aufgabe notwendig ist, getroffen werden. Es wird simuliert, „welche konkreten Handlungen die Benutzer an einem bestimmten Punkt der Interaktion vollziehen und wie ihre Aktionen mit der Nutzung des Interface verbunden sind.“ [Schweibenz & Thissen 2003, S. 90]

Überblick über den Ablauf

Bevor die einzelnen Phasen des Cognitive Walkthrough erläutert werden, werden hier allgemeine Bemerkungen zur Durchführung der Methode gemacht. Der Ablauf folgt den Handlungsschritten, welche der Produktdesigner für eine konkrete Aufgabe geplant und für den Walkthrough vorgegeben haben. Dabei führt er durch den Arbeitsablauf und präsentiert die Aktionen, die von zukünftigen Benutzern erwartet werden. Die Guthaber haben die Aufgabe, das Design zu diskutieren und Probleme im geplanten Ablauf aufzudecken. Wenn solche auf-

treten, werden die kritischen Punkte notiert, aber der Ablauf beibehalten. Das Protokoll sollte die Aktionen nach einzelnen Handlungsschritten und möglichen Fehlbedienungen inklusive deren Gründe kategorisieren.

Grundsätzlich besteht der Cognitive Walkthrough aus zwei Phasen (siehe Liste 3): einer *Vorbereitungsphase* und einer *Analysephase*. [vgl. Wharton et al 1994, S. 108] In der Vorbereitungsphase treffen sich die Gutachter und die Eingangsbedingungen für den Durchlauf werden festgelegt: die Aufgaben, die Handlungsabläufe für jede Aufgabe, die möglichen Nutzergruppen und das Interface, das analysiert werden soll. In der Analysephase gehen die Gutachter mit der Erklärung der Produktdesigner die einzelnen Schritte für jede Aufgabe im Interface Design durch, analysieren sie und protokollieren mögliche Probleme.

1) Definieren der Eingangsbedingungen für den Durchlauf	<i>Vorbereitungsphase</i>
Identifikation der Nutzer	
Probeaufgaben für die Evaluation	
Handlungsabläufe für die Erfüllung der Aufgaben	
Beschreibung oder Implementierung der Benutzeroberfläche	
2) Rekrutieren der Gutachter	
3) Durchlaufen der Aktionsfolgen für jede Aufgabe	<i>Analysephase</i>
Erzählen einer glaubwürdigen Geschichte unter Rücksicht auf...	
Wird der Nutzer versuchen die erwünschten Effekte zu erreichen?	
Wird der Nutzer feststellen, dass die richtige Aktion verfügbar ist?	
Wird der Nutzer die richtige Aktion mit dem gewünschten Effekt, den der Nutzer erreichen will, in Verbindung setzen?	
Falls die richtige Aktion ausgeführt wurde, wird der Nutzer sehen, dass ein Fortschritt in Richtung Erfüllen der Aufgabe eingetreten ist?	
4) Protokollieren kritischer Informationen	
Voraussetzungen im Wissen des Nutzers	
Vermutungen über die Zusammensetzung der Nutzergruppe	
Notizen über Alternativen und Design Änderungen	
Die glaubwürdige Erfolgsgeschichte	
5) Überarbeiten der Benutzeroberfläche, um die Probleme zu lösen	

Liste 3: Überblick über den Cognitive Walkthrough Prozess nach [Wharton et al 1994, S. 106]

1. Schritt: Definition der Eingangsbedingungen für den Durchlauf

Die Beschreibung der möglichen Nutzer des Systems sollte so spezifisch wie möglich sein. Eine einfache Beschreibung wie *Leute, die Webseiten erstellen* wäre zu wenig aussagekräftig. Besser ist es, wenn Erfahrungen und technisches Hintergrundwissen der Nutzer bekannt sind, da solche Eigenschaften Einfluss darauf haben, wie die Benutzer mit dem System umgehen können (z. B. Leute, die berufsmäßig Webseiten gestalten und schon mit professioneller Webentwicklungssoftware gearbeitet haben). Außerdem sollten die Kenntnisse über die Aufgabe und das zu prüfende Interface mit berücksichtigt werden.

Die Aufgaben, die für den Walkthrough verwendet werden, sollten repräsentativ sein oder sich auf einen kritischen Ablauf (evtl. bekannt aus früheren Versionen) beziehen. Repräsentative Aufgaben können aus den Ergebnissen von Marketinganalysen, Bedarfs- und Anforderungsanalysen bezogen werden. Des Weiteren können Aufgaben sich auf die Kernfunktionen des Systems beziehen. Für jede Aufgabe sollte ein Durchlauf geplant werden. Die Aufgabenbeschreibung beinhaltet den Kontext der Aufgaben, eine Aufgliederung der einzelnen Aktionen, welche die Handlungsschritte im Detail schildern, und Angaben zur Sicht der Nutzer auf die Aufgabe bevor sie das System kennen lernen.

Die Sequenz der Aktionen sollte so beschrieben werden, dass die Aufgabe mit dem System erledigt werden kann. Wie detailliert die einzelnen Anweisungen beschrieben werden, hängt von der Erfahrung der zukünftigen Nutzer ab (Einzelanweisungen oder zusammengefasste Anweisungen). Außerdem ist eine detaillierte Beschreibung des Designs notwendig. Je nach Entwicklungsstand können es Entwürfe auf dem Papier sein oder funktionstüchtige Prototypen. Existieren nur Entwürfe müssen notwendige Reaktionen des Systems auf Aktionen des Nutzers beschrieben werden.

2. Schritt: Rekrutieren der Gutachter

Die Gutachter setzen sich aus Experten zusammen, die im Sinne von Systemexperten Designer, und Entwickler sein können oder Mitarbeiter aus Bereichen wie Marketing, technische Dokumentation oder Schulung (Fachexperten). Bewährt hat sich hier eine Mischung aus System-, Fach- und Usability-Experten. Meist wird diese Methode in der Gruppe (keine bestimmte Größe der Gruppe empfohlen) ausgeführt. Dafür ist zusätzlich ein Usability-Experte als Moderator für die Leitung der Evaluation empfehlenswert. Der Moderator muss einerseits dafür sorgen, dass die Diskussion auf die Aufgabe konzentriert bleibt. Andererseits muss er darauf achten, dass das Protokoll präzise geführt wird. Die Designer des zu untersuchenden Produkts stellen für den Walkthrough ausgewählte Arbeitsschritte vor und erläutern der Gruppe der Gutachter, wie sie den Arbeitslauf geplant haben. Jedes Mitglied der Gruppe der Prüfer hat eine spezifische Aufgabe während der Durchführung: Eine Person sollte als Protokollant bestimmt werden, eine andere als Vermittler. Die Übrigen tragen mit ihren unterschiedlichen Kenntnissen zur Evaluation bei. Die Gutachter müssen sich genau an die formalen Bedingungen der Methode halten. Es darf immer nur eine Aufgabe durchgegangen und bewertet werden. Ansonsten ufer die Diskussion in eine allgemeine Debatte über das Design aus und die Methode verfehlt ihr Ziel.

3. Schritt: Durchlaufen der Aktionsfolgen für jede Aufgabe

Beim Untersuchen jeder einzelnen Aktion wird versucht eine glaubhafte Geschichte („credible story“ [Watson et al 1994, S. 114]) zu erzählen, warum der zukünftige Nutzer die Aktion gewählt hätte (Erfolgsgeschichte) oder nicht gewählt hätte (Misserfolgsgeschichte). Diese Geschichten basieren auf Vermutungen über das Hintergrundwissen und die Ziele der Nutzer und dem Verständnis über den Problemlösungsprozess, der den Nutzer zur Auswahl der korrekten Aktion befähigt hat (siehe auch zugrunde liegende Theorie).

Als Richtlinien für die Bewertung der Geschichten können folgende vier Fragen, welche nach der Theorie des Problemlösungsprozesses formuliert wurden, verwendet werden:

- Wird der Nutzer versuchen die erwünschten Effekte zu erreichen?
- Wird der Nutzer feststellen, dass die richtige Aktion verfügbar ist?
- Wird der Nutzer die richtige Aktion mit dem gewünschten Effekt, den der Nutzer erreichen will, in Verbindung setzen?
- Falls die richtige Aktion ausgeführt wurde, wird der Nutzer sehen, dass ein Fortschritt in Richtung Erfüllen der Aufgabe eingetreten ist?

Bei der Beantwortung der Fragen spielen allgemeine psychologische Erkenntnisse und Erfahrungswerte aus empirischen Untersuchungen eine Rolle, die sich in Vermutungen über das Verhalten der Nutzer äußern. Eine Misserfolgsgeschichte scheitert an mindestens einer dieser Fragen. Um zu verdeutlichen, wie eine solche Geschichte aussehen kann, eine beispielhafte Erfolgsgeschichte nach [Watson et al 1994, S. 114]:

Ein erfahrener Macintosh Nutzer beginnt eine Aufgabe indem er auf das Icon der Anwendung doppelt klickt, um diese zu starten.

Begründung:

- Der Nutzer versucht die Anwendung zu starten, weil der Nutzer weiß, dass eine Anwendung gestartet werden muss, um sie zu benutzen.
- Der Nutzer weiß aus seiner Erfahrung, dass Doppelklicken möglich ist.
- Der Nutzer weiß aus seiner Erfahrung, dass Doppelklicken die Aktion ist, welche genutzt werden sollte.
- Veränderungen am Bildschirm und in der Menüleiste signalisieren den Start der Anwendung.

Hat die Analyse ergeben, dass der Nutzer nicht die korrekte Aktion gewählt hätte, wird im nächsten Schritt dennoch davon ausgegangen, dass er dies getan hätte, da nicht vom vorgegebenen Handlungsablauf abgewichen werden soll.

4. Schritt: Protokollieren kritischer Informationen

Das Aufzeichnen von Informationen sollte so geschehen, dass die Evaluation effektiv und effizient ablaufen kann. Mögliche Verfahren sind das Aufzeichnen auf Video oder die Nutzung von Flipcharts oder Overhead-Projektoren, da diese für alle Gruppenmitglieder sichtbar sind. Wenn Informationen über die Nutzergruppen aufgezeichnet werden, sollten folgende Fragen beantwortet werden:

- Was müssen die Nutzer wissen bevor sie die Aufgabe durchführen?
- Was sollten die Nutzer lernen während sie die Aufgabe durchführen?

Bei der Aufzeichnung der (Miss-)Erfolgsgeschichte sollten die im dritten Schritt genannten Fragen beantwortet werden und Begründungen dafür aus den Vorkenntnissen der Nutzer und allgemeinen Usability Richtlinien notiert werden. Wichtig bei der Protokollierung ist auch, dass nicht nur die Probleme und ihre möglichen Ursachen identifiziert werden, sondern „mögliche Alternativen zum Design vorgeschlagen werden und veränderte Abläufe diskutiert werden.“ [Schweibenz & Thissen 2003, S. 94] Die Vorschläge werden dann im nächsten Schritt nochmals diskutiert und konkrete Lösungen durch die Designer entwickelt.

5. Schritt: Überarbeiten der Benutzeroberfläche, um die Probleme zu lösen

Die Problemlösung kann sich auch nach den vier Fragen aus dem dritten Schritt richten, da eine Misserfolgsgeschichte mindestens eine dieser Fragen verneinte. Die Lösungsvorschläge ergeben sich dabei aus allgemeinen Richtlinien der Software-Ergonomie und Regeln für das konkrete Produkt. Wo Probleme wenig Übereinstimmung zwischen der Vorstellung des Nutzers von der Aufgabe und dem Konzept des Designers verursachen, sollte eine „globale Reorganisation“ [vgl. Watson et al 1994, S. 125] vorgenommen werden, anstatt einzelne Elemente zu verbessern, um das Konzept anzupassen.

Beurteilung der Methode

Der Cognitive Walkthrough kann in alle Phasen des Entwicklungsprozesses angewendet werden; als expertenorientierte Methode eignet er sich aber auch besonders für frühe Phasen wo noch kein nutzbarer Prototyp vorliegt. Die Methode kann insbesondere bei aufgabenorientierte Anwendungen verwendet werden, da mit ihr festgestellt werden kann, welche Merkmale des Designs einzelne Arbeitsschritte der Aufgabe erschweren.

Wichtig für den Erfolg der Methode ist die Auswahl der Aufgaben, die durchgeführt werden. Deshalb ist eine detaillierte Aufgabenanalyse eine notwendige Voraussetzung für den Cognitive Walkthrough. Insgesamt wird durch die Auswahl der Aufgaben nur ein geringer Teil der Benutzeroberfläche evaluiert. Im Vergleich dazu findet bei der Heuristischen Evaluation eine systematische Untersuchung des kompletten Systems statt. Das bedeutet auch, dass der Cognitive Walkthrough bezüglich der Anzahl gefundener Probleme weniger effektiv ist als die Heuristische Evaluation. Dafür bietet sie Lösungsvorschläge für aufgedeckte Mängel, die direkt aus der Evaluation hervorgehen.

3.3 Nutzerorientierte Methoden

3.3.1 User Testing (Nutzertest)

„User testing with real users is the most fundamental usability method and is in some sense irreplaceable, since it provides direct information about how people use computers and what their exact problems are with the concrete interface being tested.“ [Nielsen 1993, S. 165]

Der Nutzertest ist nicht nur die fundamentalste Methode, sondern auch eine sehr aufwendige und komplexe Methode, wenn sie in der klassischen wissenschaftlichen Form durchgeführt wird. Diese Methode kann in verschiedenen Phasen des Entwicklungsprozesses angewendet werden, sobald ein Prototyp vorliegt, der von den Nutzern bedient werden kann. Nutzertest sind keine Methoden, um herauszufinden, was der Durchschnittsnutzer, den es in diesem Sinne gar nicht gibt [vgl. Krug 2006, S. 128], möchte oder ob ein Bedienelement grundsätzlich gut oder schlecht ist.

Es gibt eine Vielzahl von Verfahren, wie ein Nutzertest durchgeführt werden kann, doch grundsätzlich hat er folgende Kennzeichen:

- Die Usability des Produkts und dessen Entwicklungsprozess sollen verbessert werden.
- Die Teilnehmer sind Personen aus der Zielgruppe.
- Die Teilnehmer führen realistische Aufgaben aus.
- Das individuelle Verhalten der Teilnehmer beim Umgang mit dem Produkt wird beobachtet und aufgezeichnet.
- Die Ergebnisse werden analysiert und Änderungen zur Beseitigung der Probleme vorgeschlagen.

Bis auf wenige Ausnahmen, gilt es zielgruppentypische Nutzer zu beobachten, wie sie das konkrete Interface bedienen und herauszufinden, wie dessen Bedienelemente in dem Systemzusammenhang funktionieren. Die Auswahl der Teilnehmer und Gestaltung der Aufgaben sind für den Nutzertest daher wichtige Aspekte, auf die nachfolgend näher eingegangen wird. Außerdem werden die Phasen eines Nutzertests behandelt und das Verfahren des Lauten Denkens (Thinking Aloud) vorgestellt. Abgeschlossen wird dieser Abschnitt mit Möglichkeiten, diese Methode zu vereinfachen (Discount Usability Engineering).

Auswahl der Nutzer

Als Fastregel nennt Nielsen für die Auswahl der Nutzer, dass die Teilnehmer so repräsentativ wie möglich für die Zielgruppe sein sollten. [vgl. Nielsen 1993, S. 175] Das können im Idealfall Personen sein, die das System schon nutzen oder die es in Zukunft nutzen werden. Ansonsten sollte Wert darauf gelegt werden, dass die Teilnehmer persönliche Eigenschaften und Nutzungsverhalten abbilden, die laut Zielgruppenanalyse denen der Endnutzer entsprechen. Deshalb kommen in den meisten Fällen weder Mitglieder des Entwicklerteams oder des Managements als Testnutzer in Frage, weil sie nicht die Zielgruppe repräsentieren (außer es soll ein Produkt für Entwickler oder Manager getestet werden). Vielmehr soll diesen Gruppen durch Nutzertests mit *echten* Nutzern bewusst gemacht werden, dass die Endnutzer andere Ansprüche an ein Produkt haben.

Bezüglich der Anzahl der Nutzer wird von Nielsen eine ideale Gruppengröße von vier bis fünf Personen empfohlen, da diese etwa 75 % der Usability Probleme finden (Abbildung 5). Durch weitere Teilnehmer würden nur wenige Prozentpunkte gutgemacht werden, so dass sich Kosten und Nutzen dafür nicht decken. In der Praxis reicht eine solche Größe aus, da es sich nicht um eine wissenschaftliche Studie handelt (um statistische Aussagen zu treffen ist eine größere Stichprobe notwendig), sondern beeinträchtigende Produktmängel aufgedeckt werden sollen. Kleinere Gruppen haben außerdem den Vorteil, dass sie schneller ausgewertet werden können und Probleme damit eher beseitigt werden. Wenn es möglich ist, mehrere Tests (mit kleineren Gruppen) durchzuführen, kann die Anzahl der gefundenen Probleme insgesamt erhöht werden [vgl. Krug 2006, S. 138f.].

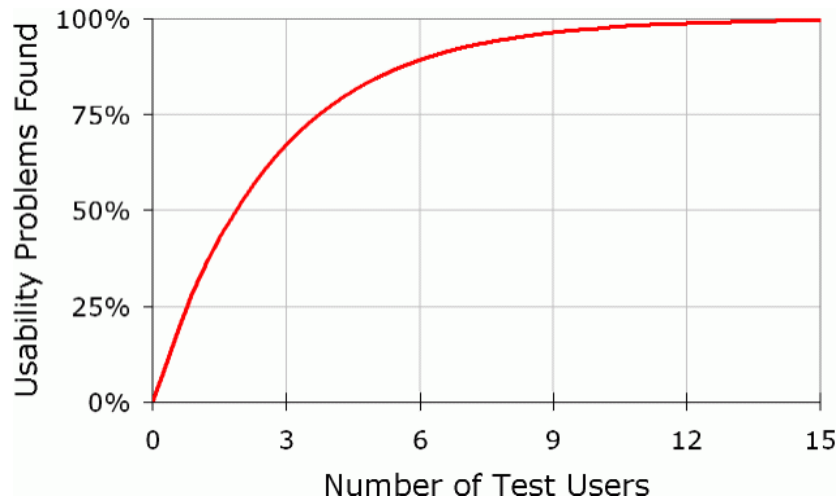


Abbildung 5: Zusammenhang zwischen der Anzahl der Nutzer und der Menge an gefundenen Usability Problemen (in Prozent) (Quelle: [Nielsen 2000])

Bei der Zusammenstellung der Teilnehmer werden häufig zwei bis drei Untergruppen bezüglich unterschiedlicher Zielgruppen (z. B. nach Alter oder Kenntnisstand) gebildet. Besonders das Alter bedingt unterschiedliche Anforderungen an das Produkt. Beim Kenntnisstand kann vereinfacht davon ausgegangen werden, dass in den meisten Fällen erfahrene Nutzer auch dann mit dem System zurechtkommen, sobald unerfahrene Nutzer damit klar kommen. Dennoch können auch hier gruppenspezifische Test Klarheit über die unterschiedlichen Bedürfnisse bringen.

Auf das Vorgehen bei der Rekrutierung der Teilnehmer soll in dieser Diplomarbeit nicht weiter eingegangen werden. Es bleibt festzuhalten, dass der Aufwand für die Rekrutierung repräsentativer Nutzer nicht dazu führen darf, dass weniger Tests durchgeführt werden. Sind die Teilnehmer für einen Test gefunden, gibt es einige Regeln im Umgang mit ihnen zu beachten.

Ethische Hinweise

Bei der Durchführung von Nutzertests darf eines nicht vergessen werden: Nutzer sind auch Menschen [vgl. Nielsen 1993, S 181] und sollten entsprechend respektvoll behandelt werden. Auch wenn keine körperliche Gefahr vom Usability Test ausgeht, so stellt er für Nutzer eine Stresssituation dar, da sie sich getestet fühlen, auch wenn ihnen versichert wird, dass das Produkt bewertet wird. Der Umstand, dass die Nutzer beobachtet oder gefilmt werden, erhöht das Unbehagen insbesondere, wenn sie glauben, etwas falsch gemacht zu haben. Die ungewohnte Umgebung eines Usability Labor kann auch einschüchternd auf die Nutzer wirken. Deshalb sollten sie möglichst in einem anderen Raum empfangen werden und das einführende Gespräch dort in entspannter Atmosphäre stattfinden. Dabei sollten dem Nutzer der Zweck des Tests mitgeteilt und das zu testende Produkt vorgestellt werden.

Der Untersuchungsleiter hat während des Tests die Aufgabe, den Nutzern die Testsituation so angenehm wie möglich zu machen. Um den Nutzern klar zu machen, dass nicht sie das Testobjekt sind, sollten sie auch nicht mit Begriffen wie *Subjekt* oder *Versuchskaninchen* bezeichnet werden, sondern eher als *Testperson*, *Gutachter* oder *Teilnehmer*. Außerdem muss der Versuchsleiter über ein gewisses Maß an Selbstkontrolle verfügen, z. B. darf er niemals über

einen Nutzer lachen. Vielmehr muss er geduldig sein und beruhigend auf den Nutzer einwirken, wenn dieser sich durch ein aufgetretenes Problem unsicher fühlt oder sich selbst die Schuld für das Problem gibt.

Die Ergebnisse sollten vertraulich behandelt werden. Dazu gehört, dass die Nutzer im Bericht nicht einzeln identifiziert werden können, indem z. B. Nummern statt Namen verwendet werden. Außerdem sollte die Anzahl der Beobachter möglichst gering ausfallen, da Nutzer verunsichert werden, aus Angst vor mehreren Personen Fehler zu machen. Video- oder Audioaufnahmen des Tests müssen ebenso vertraulich behandelt werden und nur mit Erlaubnis der Nutzer außerhalb der Untersuchung vorgeführt werden. Falls der Nutzer Mitarbeiter des Unternehmens ist, für das ein System getestet wird, sollten keine Informationen über die Leistung des Nutzers an die Vorgesetzten herausgegeben werden.

Gelegentlich stellt sich während des Tests heraus, dass ein Teilnehmer ungeeignet ist, weil er der Aufgabe nicht gewachsen ist oder sich nicht an die Testaufgaben halten will, weil er meint, es besser zu wissen. In solchen Fällen sollte der Versuchsleiter den Test unter einem Vorwand versuchen abubrechen ohne dass dem Nutzer die Schuld daran gegeben wird.

In [Nielsen 1993] kann eine zusammenfassende Liste ethischer Hinweise nachgelesen werden. Darüber hinaus kann es eine gute Erfahrung für einen Untersuchungsleiter sein, selber einmal Testteilnehmer gewesen zu sein, um die Situation nachempfinden zu können. [vgl. Nielsen 1993, S. 185]

Gestaltung der Aufgaben

Die Aufgaben für einen Nutzertest müssen folgenden Anforderungen genügen:

- *Realitätsnähe:* Die Aufgaben sollten einer typischen Nutzungssituation in der Praxis entsprechen und entsprechend anspruchsvoll sein.
- *Rücksicht auf Testsituation:* Der Umfang einer Aufgabe muss so gestaltet sein, dass sie in angemessener Zeit gelöst werden kann, aber auch nicht trivial erscheint.
- *Abdeckung wesentlicher Funktionen:* Die Aufgaben sollten möglichst die wichtigsten Bereiche der Benutzerschnittstelle verwenden. Bei komplexen Produkten können nicht alle nötigen Funktionen getestet werden, um es muss abgewogen werden, welche Aspekte berücksichtigt werden sollen.

Für den Zeitrahmen eines Tests muss immer eine Auswahl aus allen den Anforderungen entsprechenden Aufgaben erstellt werden. Aufgaben sollten in erster Linie danach ausgewählt werden, in wie weit sie vermutete Mängel eines Produkts aufdecken können. Die Vermutungen können sich auf eigene Bedenken und Erfahrungen während der Produktentwicklung stützen, z. B. bei Alternativen eines Designs. Passende Aufgaben können den Nutzertest zur Entscheidungshilfe werden lassen. Weitere Aufgaben lassen sich aus anderen Kriterien wie Schwierigkeiten bei der Fehlerbehebung ableiten. Letztlich sollten auch Aufgaben berücksichtigt werden, die aus Aufgabenanalysen oder Feldstudien bekannte tatsächliche Arbeitsaufgaben der Benutzer sind. [vgl. Schweibenz & Thissen 2003, S. 140]

Ist die Auswahl abgeschlossen, müssen die Aufgaben so formuliert werden, dass eindeutig erkennbar ist, was der Testnutzer für ein Ergebnis erzielen soll. Dabei kann dem Teilnehmer

dennoch eine gewisse Freiheit zur individuellen Ausführung der Aufgabe gelassen werden. Beispielsweise kann eine Aufgabe lauten *Sie möchten 2 Wochen Urlaub machen. Buchen Sie ihre Reise auf der Seite des Anbieters ...* anstatt vorzugeben *Buchen Sie ein 14tägiges Pauschalangebot für 2 Personen in Spanien*. Außerdem dürfen Aufgaben nicht anstößig, humorvoll oder aggressiv sein [vgl. Nielsen 1993, S. 186], um keinen Teilnehmer zu beleidigen und die Seriosität des Tests zu wahren. Stattdessen kann die künstliche Testsituation realistischer wirken, wenn die Aufgaben in ein Szenario, welches eine Rahmenhandlung vorgibt, eingebettet werden.

Die Aufgaben sollten eine logische Reihenfolge haben und an Anfang und Ende sollte jeweils eine Aufgaben stehen, die einfach zu lösen sind, um den Testnutzer den Einstieg zu erleichtern und das Gefühl vermitteln, dass sie etwas erreicht haben. [vgl. Nielsen 1993, S. 187] Nachdem die Aufgaben erstellt und geordnet wurden, sollten sie einem Pilottest mit zwei bis drei Testnutzern unterzogen werden, um ihre Verständlichkeit zu überprüfen und festzustellen, wie viel Zeit für die Lösung der Aufgaben benötigt wird, um die Dauer des Nutzertests genauer planen zu können.

Während des Tests werden die Aufgaben den Nutzern üblicherweise in schriftlicher Form vorgelegt, so dass jeder Teilnehmer die gleiche Formulierung erhält und er sich während der Durchführung darauf beziehen kann. Der Versuchleiter sollte nach Ausgabe jeder Aufgabe dem Nutzer die Möglichkeit geben, Fragen zur Aufgabenstellung zu stellen, um Missverständnisse zu vermeiden.

Planung und Vorbereitung des Tests

Ein Nutzertest muss genau geplant⁸ und vorbereitet werden, da er aufgrund des Aufwands nicht so einfach wiederholt werden kann und jede Sitzung des Tests gleich ablaufen sollte, damit die Ergebnisse vergleichbar sind. Deshalb ist es sinnvoll, einen Ablaufplan zu erstellen, der die Schritte einzeln auflistet und vom Versuchleiter eingehalten wird. Jede einzelne Testsitzung hat dabei vier typische Phasen [nach Nielsen 1993, S. 187]:

- Vorbereitung
- Einführung
- Der eigentliche Test
- Nachbesprechung

Die Vorbereitung des Testraum und der Materialien sollte bevor der Teilnehmer eintrifft abgeschlossen sein, um diesen nicht zu verunsichern. Dazu zählt, dass die technischen Geräte einsatzbereit sind, alle Unterlagen wie Aufgabenzettel und Fragebögen bereit liegen, alle Daten der vergangenen Testsitzung gesichert wurden und das System wieder in Startposition versetzt wurde (ursprüngliche Daten hergestellt, Zwischenspeicher gelöscht etc.). Funktionen, die während der Sitzung stören könnten wie Bildschirmschoner, Benachrichtigungssysteme oder Telefone sollten abgestellt werden.

⁸ Eine Liste von Themen, die bei der Planung des Tests abgearbeitet werden sollten, wird beispielsweise bei [Nielsen 1993, S. 170] vorgegeben.

Einführung in den Test

Am Anfang steht die Begrüßung des Teilnehmers durch den Versuchleiter und eine kurze Erklärung zur Zielsetzung des Tests. Die nachfolgenden einleitenden Hinweise können anhand einer Checkliste abgearbeitet werden ohne dabei mechanisch zu wirken. Die Einführung sollte folgende Inhalte [nach Nielsen 1993, S. 188f.] haben:

- Es sollte betont werden, dass es Ziel des Tests ist, das Produkt zu bewerten und nicht den Nutzer.
- Der Versuchleiter sollte versichern, dass er selbst kein persönliches Interesse an dem Produkt hat, das evaluiert wird, außer er ist gleichzeitig der Produktdesigner. Dann sollte er diese Punkt unerwähnt lassen. Dieser Hinweis hilft dem Testnutzer sich freier zu äußern, da er keine Rücksicht auf die Gefühle des Versuchleiters nehmen muss.
- Es sollte klar werden, dass die Testergebnisse benutzt werden, um das Produkt zu verbessern, so dass das endgültige Produkt sich von dem getesteten unterscheiden wird.
- Der Testnutzer muss zur Verschwiegenheit über das getestete System verpflichtet werden. Das beinhaltet auch, dass er sich nicht mit anderen Teilnehmern darüber austauscht, um diese nicht zu beeinflussen.
- Der Teilnehmer sollte eine Erklärung abgeben, dass er freiwillig daran teilnimmt und ihm bewusst ist, dass er jederzeit den Test abbrechen kann.
- Der Versuchsleiter sollte dem Teilnehmer versichern, dass die Ergebnisse vertraulich und anonym behandelt werden.
- Der Versuchsleiter muss alle verwendeten Video- und Audioaufnahmen und deren spätere Verwendung dem Teilnehmer erklären.
- Dem Teilnehmer sollte erklärt werden, dass er jederzeit Fragen zum Test stellen darf, der Versuchsleiter aber einige davon evtl. nicht beantworten kann, da getestet werden soll, wie gut das Produkt ohne Hilfe von außen bedient werden kann.
- Anweisungen zur verwendeten Methode im Test müssen dem Teilnehmer erklärt werden, z. B. das Verfahren des lauten Denkens.
- Bevor der Test beginnt, sollte der Teilnehmer noch Fragen stellen dürfen, falls ihm nach der Einleitung etwas unklar ist.

Es wird empfohlen ein kurzes, einfaches Formular, welches die oben genannten Inhalte auflistet, dem Teilnehmer zum Einverständnis vorzulegen. Auf dem gleichen Formular kann sich auch die Einverständniserklärung über die öffentliche Verwendung der Aufzeichnungen (z. B. gegenüber den Auftraggebern oder auf wissenschaftlichen Konferenzen) befinden. Die Erklärungen sollten durch zwei getrennte Unterschriften eingeholt werden, da sich einige Nutzer zwar gerne am Test beteiligen würden, aber gegen eine Veröffentlichung des Videomaterials sind. Auch wenn Nielsen meint, dass solche Formulare das „Angstlevel der Nutzer“ [vgl. Nielsen 1993, S. 189] erhöhen, stellen sie doch eine Formalität dar, die dem Teilnehmer seine

Pflichten, aber auch Rechte, deutlich macht und aus rechtlichen Gründen teilweise notwendig ist.

Bevor der eigentliche Test beginnt, sollte noch überprüft werden, ob die Testgeräte an den Teilnehmer optimal angepasst sind (Kamera- und Mikrofonausrichtung, Maus für Linkshänder etc.). Dies kann im Zusammenhang mit einem Probelauf zum Kennen lernen der verwendeten Methode geschehen. Beispielsweise kann zum Üben des Lauten Denkens eine kleine Aufgabe vor dem eigentlichen Test vom Nutzer erledigt werden.

Durchführung des Tests

Bei der Durchführung können neben dem Testnutzer und dem Versuchsleiter auch ein Assistent und Beobachter anwesend sein (im gleichen Raum oder getrennten Beobachtungsraum). Nur der Versuchsleiter sollte während des Tests mit dem Nutzer interagieren, wobei sich die Frage stellt, in wie weit jener mit dem Testnutzer kommunizieren darf. Ob die Interaktion auf ein Minimum beschränkt wird oder eine normale Kommunikation stattfindet (als der Versuchsleiter auch Zwischenfragen stellen darf), hängt vom Ziel des Tests ab. In beiden Fällen darf der Untersuchungsleiter keine persönlichen Kommentare zum Verhalten des Nutzers machen oder ihn durch seine Fragen beeinflussen. Die Fragen sollten deshalb möglichst neutral gestellt werden (offene Fragen), z. B. *Wie könnte diese Prozedur verbessert werden?*.

Andererseits suchen Teilnehmer den Kontakt zum Versuchsleiter durch Kommentare oder Fragen, und erwarten eine Reaktion des Versuchsleiters darauf. Dieser muss darauf achten durch seine Reaktion den Nutzer nicht zu beeinflussen. Auf Fragen des Teilnehmers, die er nicht direkt beantworten darf wegen dem Test, kann der Versuchsleiter mit Gegenfragen reagieren. So kann er beispielsweise auf die Frage *Soll ich jetzt auf den Link klicken?* mit der Gegenfrage *Was möchten Sie denn erreichen?* antworten. Auch Hilfestellungen durch den Versuchsleiter müssen gut abgewogen werden und die eingeschlagene Strategie konsequent bei allen Teilnehmern angewendet werden. Nach Nielsen sollte dem Nutzer erst geholfen werden, wenn er von sich aus nicht weiter kommt und unglücklich mit der Situation ist oder bei einem Problem ist, dass aus vergangenen Tests bereits eindeutig bekannt ist. [vgl. Nielsen 1993, S. 190]

Eine weitere Frage ist, wann der Testnutzer eine Aufgabe abgeschlossen hat. Teilt der Nutzer dies nicht von selbst mit, weil er sich nicht sicher ist, sollte der Versuchsleiter eingreifen ohne den Nutzer weiter zu verunsichern und zu frustrieren. Wichtig ist auch dabei, dass die einmal eingeschlagene Strategie diesbezüglich bei allen Teilnehmern angewendet wird. Nach Abschluss jeder Aufgabe oder erst am Ende des Tests kann ein Fragebogen oder Interview folgen, in dem der Nutzer nach seiner Meinung und Verbesserungsvorschlägen gefragt wird. Fragebögen nach den Aufgaben sind hilfreich bei Fragen zu Details, an die sich der Nutzer nach dem Test vielleicht nicht mehr erinnert, haben aber den Nachteil, dass sie den Ablauf unterbrechen und monoton wirken, wenn sie zu oft eingesetzt werden.

Auch wenn der Test aufgezeichnet wird, sollte sich der Versuchsleiter (und sein Mitarbeiter) Notizen zu aufgetretenen Problemen und interessanten Aussagen machen. Nach einer Reihe von Tests geraten Details, die für die Auswertung interessant sind, sonst in Vergessenheit.

„Am Ende eines Testtages sollten die Notizen in eine Problemliste übertragen werden, um die spätere Auswertung zu vereinfachen.“ [Schweibenz & Thissen 2003, S. 150]

Verfahren: Thinking Aloud (Lautes Denken)

„Thinking aloud may be the single most valuable usability engineering method.“ [Nielsen 1993, S. 195] Denn sie hilft, die Vorstellungen der Testnutzer über das System zu verstehen und Abweichungen zwischen dem Nutzerkonzept und dem Produktkonzept aufzudecken. Die Teilnehmer eines Tests werden dabei aufgefordert, während dem Erledigen der Aufgaben laut zu äußern was sie denken und was sie tun. Durch die Parallelität von Aussage und Handlung kann der Versuchleiter genau feststellen, welches Dialogelement ein Problem verursacht hat. Gleichzeitig kann bei Überbewertung der Meinungen des Testnutzers ein falscher Eindruck über die Ursache des Usability Problems entstehen.

Lautes Denken fällt den meisten Nutzern schwer, da es ein ungewohntes Verhalten ist. Deshalb sollte vor dem eigentlichen Test eine Einführung in die Methode gegeben werden (z. B. in Form eines Videoausschnitts aus einem vergangenen Nutzertest) und die Methode an einer Beispielaufgabe, die nicht im Zusammenhang mit dem eigentlichen Testprodukt steht, vom Nutzer geübt werden. Während des Tests sollte der Versuchleiter den Nutzer zum lauten Denken animieren, indem er, ohne zu beeinflussen, Rückmeldung gibt oder Fragen stellt (z. B. Was denken Sie gerade?).

Die *aktive Intervention (active intervention)*, eine Variante des lauten Denkens, erlaubt einem Mitglied des Untersuchungsteams den Teilnehmer gezielt zu fragen, ob und wie er etwas verstanden hat. Ähnlich unterstützend arbeitet auch die *„coaching method“* [Nielsen 1993, S. 199], bei der ein Experte oder der Versuchleiter dem Testnutzer Fragen zum System beantwortet. Letztere ist zudem hilfreich bei der Analyse der benötigten Hilfestellung besonders von unerfahrenen Nutzern. Eine weitere Variante, die das laute Denken natürlicher macht, ist die *konstruktive Interaktion (constructive interaction)* [Nielsen 1993, S. 198], bei der zwei Testnutzer zusammen den Test durchführen und sich gegenseitig unterstützen.

Nachteile des lauten Denkens sind, dass die Performance des Nutzers eingeschränkt sein kann und deshalb Messungen der Dauer zum Erledigen einer Aufgabe weniger repräsentativ ist. Außerdem kann der Problemlösungsprozess des Nutzers beeinflusst werden, da er sich auf die Äußern seiner Gedanken konzentrieren muss. Trotzdem kann das laute Denken dem Nutzer helfen eine Aufgabe effizienter zu lösen oder weniger Fehler zu machen als Teilnehmer, die nicht laut gedacht haben. [vgl. Nielsen 1993, S. 196] Dies kann wiederum die Reliabilität der Ergebnisse verändern.

Nachbereitung

Nach Ende des Tests wird der Teilnehmer in Form eines Fragebogens oder Interviews zum Produkt befragt. Außerdem kann der Testnutzer an dieser Stelle eigene Anmerkungen und Verbesserungsvorschläge machen. Der Versuchleiter sollte jetzt auch die Fragen beantworten, die er vor oder während des Tests nicht beantworten durfte. Danach sollte der Teilnehmer nochmals an seine Verschwiegenheitspflicht erinnert werden, er sollte das vereinbarte Honorar erhalten und verabschiedet werden.

Im zweiten Teil der Nachbereitung erfolgt die Archivierung der Testdaten und -unterlagen unter einer Teilnehmernummer. Das Produkt wird in den Startzustand zurückgesetzt, damit der nächste Test beginnen kann. Checklisten können an dieser Stelle wieder hilfreich sein, um gewissenhaft alle notwendigen Schritte durchzuführen.

Auswertung

Zur Identifizierung von Usability Problemen müssen die gewonnenen Daten in ein angemessenes Verhältnis gesetzt werden. Dumas und Redish schlagen dafür die Methode der Triangulation (Abbildung 6) vor, bei der im Zentrum eines Dreiecks sich das Usability Problem aus den an den Spitzen angeordneten Datenquellen (sortiert nach der Art von Daten) ergibt.

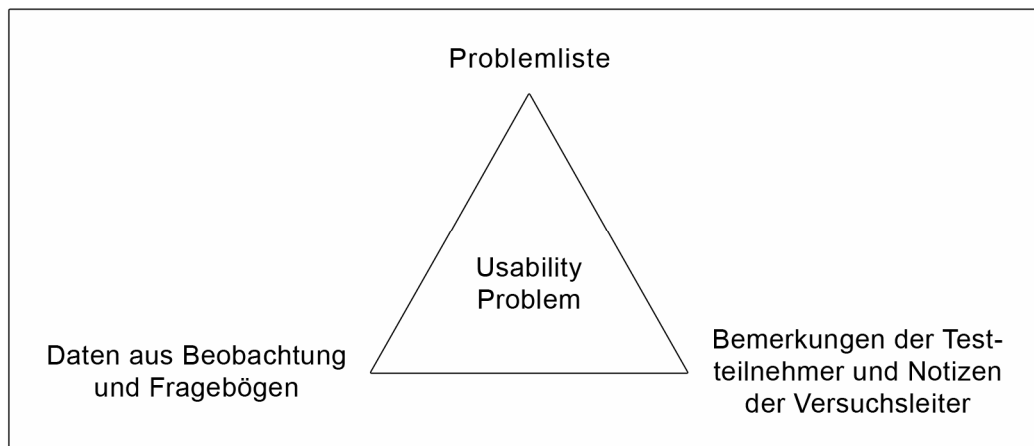


Abbildung 6: Triangulation von Usability Problemen nach Dumas & Redish 1994 [vgl. Schweibenz & Thissen 2003, S. 151]

Dabei kann die Auswertung der Daten leistungsorientiert oder nach subjektiven Maßgaben erfolgen. Bei der leistungsorientierten Auswertung werden Maßzahlen wie benötigte Zeit zum Erledigen einer Aufgabe oder Anzahl der Fehler berücksichtigt. Diese Zahlen können nur durch genaue Beobachtung erhoben werden und machen eine Videoaufzeichnung unerlässlich. Dafür sind sie genau messbar und quantitativ auswertbar. Subjektive Maßgaben sind Angaben über den Ausdruck von Zufriedenheit oder Frustration der Testnutzer. Sie können entweder quantitativ (z. B. Anzahl der Kommentare) oder qualitativ (z. B. Inhalt der Kommentare) ausgewertet werden. Tabellen eignen sich für die Darstellung der quantitativen Daten und können dadurch besser miteinander verglichen werden. Qualitative Daten werden meist in Form von Zitaten der Teilnehmer dargestellt und können nach Aufgaben und nach Ähnlichkeit des Problems geordnet werden.

Bei der Auswertung eines Produkttests kommt es darauf an, die Daten richtig zu interpretieren und somit die tatsächlichen, den Beobachtungen zugrunde liegenden Usability Probleme zu identifizieren. Zur Verarbeitung der Ergebnisse im weiteren Entwicklungsprozess sollten die gefundenen Usability Probleme nach ihrer Ernsthaftigkeit bewertet werden (siehe Abschnitt 3.5.1) bevor ein abschließender Bericht die Ergebnisse zusammenfasst. Für diesen Bericht müssen die Videoaufzeichnungen transkribiert werden, d. h. die Handlungen und Aussagen der Nutzer werden niedergeschrieben (meist in Form einer Tabelle).

Wege zum Discount Usability Engineering

Obwohl der Nutzertest als unverzichtbare Methode der Usability Evaluation gilt, hat sich gezeigt, dass die Komplexität der Methode ihrem Einsatz in der Praxis im Wege stehen kann. Deshalb empfiehlt Nielsen die Methode zu vereinfachen, dass sie zwar keine „perfekten“ Ergebnisse liefert, aber noch viele Usability Probleme aufdeckt. Ein Schritt zur Vereinfachung, der gleichzeitig Kosten spart, ist der Verzicht auf ein Usability Labor. In der Praxis lohnt sich die Ausstattung eines kompletten Usability Labors (und damit verbunden auch die Anstellung eines Usability-Experten Teams) nur wenn Usability Tests Hauptaufgabe des Unternehmens sind. Ansonsten reicht im einfachsten Fall ein Raum, der für den Test hergerichtet wurde.

Ein Verzicht auf Videoaufzeichnung reduziert ebenso den Aufwand und die Kosten. Stattdessen kann die Beobachtung verstärkt werden, indem ein oder zwei zusätzliche Beobachter Notizen während des Tests machen. Alternativ können andere Aufzeichnungsmethoden verwendet werden. Ausrüstung zur Aufzeichnung des Blicks des Nutzers und der Mausbewegung sind an dieser Stelle zu nennen (Eye-Tracking⁹), obwohl auch hier die Kosten für professionelle Ausrüstung nicht unerheblich sind. Wird eine Videoaufzeichnung gemacht, so kann der Aufwand bei der Auswertung reduziert werden, wenn das Material nicht vollständig transkribiert wird, sondern nur wichtige Zitate schriftlich festgehalten.

Nielsen schlägt zwei weitere Änderungen des Nutzertests vor. Zum einen können als Testmaterial sog. *Scenarios* verwendet werden und zum anderen kann das Verfahren des lauten Denkens vereinfacht werden. Scenarios sind Prototypen, die sowohl in der Anzahl der Features als auch im Umfang der Funktionalität reduziert wurden. Diese können schnell und billig hergestellt werden und bieten daher die Möglichkeit mehr Tests durchzuführen. Die Methode des lauten Denkens wird dadurch vereinfacht, dass anstatt von Usability-Experten oder Psychologen die Entwickler die Methode durchführen und es keine Videoaufzeichnung gibt. Nielsen stellt fest, dass diese mit minimalem Training durchaus in der Lage sind, die Thinking Aloud Methode effektiv anzuwenden. [vgl. Nielsen 1993, S. 19]

Eine Gegenüberstellung zwischen traditionellen und Discount Methoden wird in Abschnitt 3.4.2 dargestellt. Außerdem sei auf Kapitel 6 verwiesen, in dem die im Rahmen dieser Diplomarbeit durchgeführten Nutzertests beschrieben werden.

3.3.2 Focus Group (Fokusgruppe)

Bei Fokusgruppen handelt es sich um moderierte Gruppendiskussionen über ein vorgegebenes Thema, um Meinungen und Bedürfnisse der Teilnehmer zu sammeln. Synonym dazu werden auch die Begriffe Gruppendiskussion und Gruppeninterview verwendet. Fokusgruppen werden zu einem Zeitpunkt der Entwicklung veranstaltet, wo es noch keine oder erste Designentwürfe gibt oder nachdem ein Produkt eine Zeitlang im Einsatz war (evtl. als Vorbereitung für das Nachfolgeprodukt oder ein anderes der Produktreihe). Ziel ist es, Anregungen für die Produktentwicklung zu geben oder Entscheidungen zur Gestaltung zu erleichtern.

⁹ Eye-Tracking (Blickmessung) ist ein Verfahren, mit dem gemessen wird, an welchen Punkten der Blick des Nutzers stehen bleibt, da nur dann Informationen wahrgenommen werden.

Auswahl, Anzahl und Rekrutierung der Teilnehmer

Die Auswahl der Teilnehmer sollte die Zielgruppe repräsentativ vertreten. Es werden meist mehrere Fokusgruppen mit unterschiedlichen Teilnehmern zu einem Thema durchgeführt, da es die Objektivität der Ergebnisse erhöht. Es sollte darauf geachtet werden, dass die Teilnehmer sich untereinander nicht kennen, da Bekannte oder Freunde meist eine gemeinsame Grundlage für ihre Meinung haben und dadurch weniger Bereitschaft zur Diskussion besteht. Außerdem kann unterschieden werden zwischen einer heterogenen (Teilnehmer mit unterschiedlichen Hintergrund) und einer homogenen (ähnlicher Hintergrund) Zusammensetzung der Gruppe. In homogenen Gruppen wird meist schnell ein Konsens erreicht, während heterogene Gruppe mehr Konfliktpotential besitzen. Grundsätzlich sind letztgenannte reizvoller für die Fokusgruppe, da sie das Thema aus sehr unterschiedlichen Blickpunkten betrachtet; sie ist aber auch schwieriger zu moderieren.

Insgesamt sollte eine Gruppe aus 6 bis 10 Personen bestehen, da diese Größe einen angemessenen Rahmen für eine Diskussion schafft. Bei der Rekrutierung der Teilnehmer müssen die Personen alle notwendigen Informationen über den Inhalt und Ablauf der Fokusgruppensitzung erhalten, damit diese sich für oder gegen die Teilnahme entscheiden können. Per Anmeldeformular wird die Teilnahme bestätigt. Grundsätzlich wird empfohlen, ein bis zwei Personen zusätzlich zu rekrutieren, um Ausfälle anderer Teilnehmer ausgleichen zu können.

Gestaltung der Themen

Fokusgruppen haben im Allgemeinen ein oder zwei Themen. Diese können strukturiert oder unstrukturiert diskutiert werden. Wenn eine Diskussion frei erfolgen soll, reicht es, das Thema (evtl. als Frage) vorzugeben, um die Diskussion anzustoßen. Die unstrukturierte Form bietet die Möglichkeit, dass die Teilnehmer Fragen zum Thema aufwerfen, die bei der Vorgabe durch den Moderator nicht diskutiert worden wären. Andererseits kann es passieren, dass bestimmte Aspekte des Themas nicht diskutiert werden, da die Teilnehmer nicht auf diese Punkte hingewiesen worden.

Für eine strukturierte Diskussion gibt der Moderator präzise Fragen meist in Form eines Gesprächsleitfadens, nach dem der Verlauf der Diskussion geplant wird, vor. Bei den Fragen sollte es sich um offene Fragen (also solche, die nicht mit Ja, Nein oder einer quantitativen Angabe beantwortet werden können) handeln, welche die Meinungen der Teilnehmer nicht im Vorhinein beeinflusst. Eine mögliche Frage wäre *Was könnte aus Ihrer Sicht verbessert werden?* anstatt *Sollte die Schriftgröße lieber größer sein?*.

Die Diskussion wird belebt, indem anfangs allgemeine Fragen gestellt werden, die alle Teilnehmer beantworten können, und gegen Ende spezielle Fragen. Außerdem sollten positiv formulierte Fragen vor negativ formulierten Fragen gestellt werden, da es den Teilnehmern leichter fällt, sich positiv zu äußern, wenn sie vorher keine negativen Aspekte des Themas besprochen haben. Alle Fragen sollten ausformuliert sein, damit die Reliabilität der Fokusgruppensitzung gewährleistet ist. Zur Strukturierung des Ablaufs der Sitzung bietet sich ein

Gesprächsleitfaden an, der auch Anweisungen für den Moderator oder Angaben zu verwendeten Materialien (Video, Flipchart etc.) enthält.¹⁰

Gesprächsregeln

Folgende Regeln sind im Rahmen einer Fokusgruppe einzuhalten, gelten aber auch (teilweise) bei anderen nutzerorientierten Methoden.

- *Gute Umgangsformen:* Besonders innerhalb einer Gruppe sollten die Teilnehmer daran erinnert werden, dass auch hier allgemeine Regeln wie *Ausreden lassen*, *alle Meinungen sind richtig und gleichberechtigt* oder *keine persönlichen Beleidigungen* gelten.
- *Eigene Erfahrungen wiedergeben:* Teilnehmer neigen dazu, über Bedürfnisse und Wünsche anderer Nutzer zu spekulieren oder Verallgemeinerungen vorzunehmen. Bei der Interaktion des Nutzers mit dem zu testenden Produkt sollte der Untersuchungsleiter den Nutzer motivieren, nur über seine eigenen Erfahrungen zu sprechen (z. B. was stört ihn an der Bedienung). In Gruppendiskussionen ist der Moderator dafür verantwortlich, dass die individuellen Hintergründe der Teilnehmer zur Geltung kommen und sich die Teilnehmer nicht gegenseitig beeinflussen oder vorschnell einander zustimmen.
- *Keine Rücksicht auf die technische Machbarkeit:* Der Nutzer sollte dazu motiviert werden, frei seine Bedürfnisse und Vorschläge zu äußern, die seiner Meinung nach das Produkt tauglicher für ihn machen würden, ohne dabei auf die technische Machbarkeit Rücksicht zu nehmen. Nicht die Teilnehmer sollen Lösungen finden, sondern die Produktentwickler auf Grundlage der Ergebnisse aus der Arbeit mit den Nutzern.

Moderation

Die Moderation einer Fokusgruppe ist ein entscheidendes Kriterium für den Erfolg. Der Moderator sollte geschult sein in Moderationstechniken und Erfahrung in Umgang mit Gruppenprozessen haben. Wertvoll sind auch Eigenschaften wie Einfühlungsvermögen und Selbstkontrolle. Der Moderator muss eine Beziehung zur Gruppe aufbauen, dabei aber gleichzeitig Objektivität gegenüber den Teilnehmern und dem Auftraggeber wahren. Welche Aufgaben der Moderator im Verlauf der Fokusgruppensitzung hat zeigt Liste 4. Dazu gehören die Begrüßung der Teilnehmer, das Erklären der Gesprächsregeln und die Beschreibung des Ablaufs. Bevor die Diskussion beginnt, sollte er noch die Einverständniserklärung der Nutzer einholen, falls die Sitzung per Video oder Audio aufgenommen wird. Falls vom Auftraggeber gewünscht, werden die Teilnehmer auch zum Stillschweigen über gezeigte Inhalte verpflichtet. Zum Ende der Einleitungsphase sollten die Teilnehmer sich noch untereinander vorstellen, damit sich die Gruppe und auch der Moderator die Gruppe besser kennenlernt.

¹⁰ Ein Beispiel für einen Gesprächsleitfaden dieser Art kann bei [Schweibenz & Thissen 2003] gefunden werden.

Einleitung

Ziel der Fokusgruppe vorstellen

Begrüßung der Teilnehmer

Gesprächsregeln definieren

Um Einverständniserklärung zur Video- oder Audioaufnahme bitten (bei Bedarf)

Teilnehmer zum Stillschweigen über Inhalt und Gestaltung des gezeigten Produkts bitten (bei Bedarf)

Vorstellung der Teilnehmer untereinander

Aktivitäten

Einführung in das Thema (Stimulus)

Fragen zum Thema stellen

Zwischenergebnisse zusammenfassen

Abschluss

Zusammenfassung der Ergebnisse

Diskussion über abschließende Bemerkungen der Teilnehmer verhindern

Danke an die Teilnehmer

Liste 4: Aufgaben des Moderators in den Phasen der Fokusgruppensitzung

Eine weitere Aufgabe des Moderators ist es durch Fragen zum Thema die Diskussion in Gang zu bringen (siehe Gestaltung der Themen). Während der Diskussion sollte der Moderator gelegentlich die Ergebnisse zusammenfassen, um die Teilnehmer auf den gleichen Stand zu bringen und die Diskussion wieder auf das Ziel ausrichten, falls sie davon abgekommen ist. Durch vertiefende Fragen kann der Moderator Oberflächlichkeit in der Diskussion vermeiden, da es das Ziel ist, die Beweggründe der Teilnehmer zu erforschen.

Zum Abschluss fasst der Moderator die Ergebnisse der Gruppe zusammen. Danach sollte jeder Teilnehmer die Möglichkeit erhalten, ein Fazit zu ziehen. Dieses wird nicht mehr diskutiert. Danach dankt der Moderator den Teilnehmer und beendet formal die Fokusgruppe.

Auswertung

Während der Sitzung steht dem Moderator ein Protokollant zur Seite, der die Ergebnisse festhält. Zusätzlich oder anstelle können Video- oder Audioaufnahmen genutzt werden. Direkt nach der Sitzung sollten sich Moderator, Protokollant und Auftraggeber (falls er bei der Sitzung anwesend war) zu einer Nachbesprechung treffen, um ihr persönliches Verständnis der Ergebnisse zu diskutieren. Dadurch werden Missverständnisse ausgeräumt, bevor der Abschlussbericht angefertigt wird.

Eine praktische Form der Auswertung ist ein Bericht, der auf Basis des Sitzungsprotokolls angefertigt wird und die Einstellungen der Teilnehmer wiedergibt. Er kann mit Zitaten der Teilnehmer angereichert werden, um die Aussagen zu illustrieren. In einer wissenschaftlichen Form der Auswertung (Inhaltsanalyse) werden die Video- oder Audioaufnahmen transkribiert, indem Abschriften der gesamten Sitzung (oder wichtiger Teile) angefertigt werden. „Aus den Transkriptionen werden dann Aussagen oder Passagen ausgewählt, die mit Codes versehen und ausgeschnitten und mit verwandten Aussagen oder Passagen zusammengeführt werden.“

[Schweibenz & Thissen 2003, S. 128] Danach wird der Verlauf der Diskussion protokolliert. Aus den Verlaufsprotokoll und Zitaten wird der Bericht erstellt, wobei die Balance zwischen Diskussionsverlauf und ausgewählten Aussagen erhalten bleiben muss, damit der Bericht zwar anschaulich, aber nicht unübersichtlich wird (siehe auch Abschnitt 3.5.2 zur Präsentation von Ergebnissen).

Beurteilung der Methode

Fokusgruppen werden kritisch bezüglich ihres Nutzens zur Usability Evaluation gesehen, da hierbei die Nutzer lediglich befragt werden, aber nicht ihr Verhalten mit dem Produkt beobachtet wird. Es besteht immer eine gewisse Differenz zwischen dem, was Nutzer sich für ein Produkt wünschen oder angeben damit tun zu wollen und ihrer tatsächlichen Nutzung des Produkts. Außerdem können die Teilnehmer sich gegenseitig beeinflussen, was die Unabhängigkeit der Meinungen mindern kann. Eine Diskussion in der Gruppe liefert aber auch Chancen unterschiedlichen Ansichten direkt miteinander zu diskutieren. Ziel sollte es aber nicht sein, die Widersprüche zu lösen, sondern die zugrunde liegenden Ursachen bei den Nutzern zu erforschen.

Fokusgruppen sollten nie als einzige Methode zur Evaluation eingesetzt werden, sondern in Kombination mit Nutzertests, da hierbei eine Beobachtung des Verhaltens der Nutzer bei der Interaktion mit dem Produkt stattfindet.

3.4 Diskussion der vorgestellten Methoden

3.4.1 Expertenorientierte Methoden gegen nutzerorientierte Methoden

Beide Arten von Methoden haben ihre Vor- und Nachteile, wie der Vergleich in Tabelle 1 zeigt. Welche Methode verwendet wird, ist situationsabhängig und sollte nicht vom voraussichtlichen Aufwand abhängig gemacht werden. Auch wenn expertenorientierte Methoden oftmals weniger Aufwand bedeuten und die Heuristische Evaluation besonders einfach zu erlernen ist, sollten sie nicht ausschließlich verwendet werden.

Tabelle 1 Vergleich von empirischen und Inspektionsmethoden nach [Heuer 2003, S. 117]

<i>Inspektionsmethoden z. B. Heuristische Evaluation</i>	<i>Empirische Verfahren z. B. Usability Tests</i>
Begutachter beurteilen ein Interface basierend auf ihrer Erfahrung und Richtlinien	Repräsentative Nutzer bearbeiten Aufgaben
Nutzergruppen und Aufgaben werden oft nicht berücksichtigt	Kenntnisse über Nutzergruppen und Aufgaben erforderlich
Verfahren sind z. T. sehr schnell und effizient einsetzbar (z. B. Checklisten)	Umfangreiche Tests können sehr aufwändig sein
Auch gut für die Beurteilung nicht-funktionaler Prototypen geeignet	Ideal für Untersuchungen funktional vollständiger Systeme
Keine Erhebung quantitativer Maße	Geeignet zu Erhebung von quantitativen Maßen (Fehlerrate, Zeitdauer, subjektive Zufriedenheit)

Nachteil der expertenorientierten Methoden ist, dass sie nicht aus Sicht echter Benutzer durchgeführt werden. Dies kann die Überzeugungskraft der Ergebnisse schmälern, da sie „als bloße Meinungsäußerung diskreditiert werden“ [Heuer 2003, S. 117]. Außerdem gibt es einen Unterschied in der Art der Probleme, die gefunden werden. Inspektionsmethoden finden im Allgemeinen eher weniger ernsthafte Probleme, die in ihrer Gesamtheit auch den Nutzer beeinträchtigen können. Nutzerorientierte Methoden finden dafür eher kritische Usability Probleme, die den Nutzer bei der Erledigung der Aufgaben erheblich behindern und frustrieren oder ganz davon abhalten. Außerdem wurde festgestellt, dass expertenorientierte Methoden im direkten Vergleich mit empirischen Methoden weniger effektiv sind, d. h. es werden weniger Probleme gefunden.

Deshalb hat sich in der Praxis eine Kombination aus experten- und nutzerorientierten Methoden bewährt, da sie sich gegenseitig ergänzen. Aufgrund ihrer Unterschiede eignen sich die Methodenarten für den Einsatz zu unterschiedlichen Zeitpunkten im Entwicklungsprozess. Experten können oftmals schon in der Entwurfsphase Untersuchungen durchführen. Nutzer hingegen sind effizienter, wenn sie einem funktionstüchtigen Prototypen oder ein Endsystem überprüfen. Im iterativen Designprozess werden auch lieber expertenorientierte Methoden eingesetzt, da für die mehreren aufeinander folgenden Untersuchungen keine Benutzer „verschwendet“ werden (Nutzer können nur eine begrenzte Zeit für die Evaluation desselben Projekt eingesetzt werden, da sonst ihre Objektivität durch vorhandene Vorkenntnisse beeinträchtigt wird.).

3.4.2 Traditionelles Usability Engineering gegen Discount Usability Engineering

Der Usability Engineering Prozess (siehe Abschnitt 1.4) sieht die Entwicklung eines Produkts begleitende Usability Evaluationen vor. Ziel ist es dabei möglichst nah am Zeitplan der Produktentwicklung Ergebnisse über die aktuelle Benutzertauglichkeit des Interface zu liefern und die Beseitigung bekannter Probleme in den weiteren Verlauf einzuplanen. Die traditionellen Methoden, die auf Forschungsmethoden der Psychologie basieren, werden aufgrund ihrer Komplexität (und ihrem wissenschaftlichen Anspruch) meist nicht eingesetzt und es mussten anderen Methoden entwickelt werden, die besser geeignet sind für den praktischen Einsatz.

Die hohen formalen Ansprüche der traditionellen Methoden (z. B. Nutzertests), um zuverlässige Ergebnisse zu erzielen, tragen dazu bei, dass sie eher für die Forschung als für die Praxis geeignet sind. Für einen klassischen Nutzertest werden beispielsweise ein Usability Labor mit entsprechender Ausstattung benötigt, eine der Zielgruppe entsprechende Gruppe von Testpersonen (acht oder mehr, um statistisch relevant zu sein) und ein Usability-Experte, der die Untersuchung plant und leitet. Außerdem ist für die Vorbereitung und die eigentliche Testphase ein Zeitraum von mehreren Wochen einzuplanen. Aufgrund dieser Bedingungen sind Nutzertests auch sehr kostenintensiv. Auch wenn sie die effektivsten Methoden zur Bestimmung der Usability sind, da sie direkt auf den Reaktionen der Nutzer auf das System beruhen, werden sie in der klassischen Form in der Praxis kaum eingesetzt.

Anstatt in der Praxis keine Usability Evaluation durchzuführen, weil die beste Methode zu teuer, zu zeitintensiv oder zu komplex ist, sollten Techniken gewählt werden, die gute aber

nicht „perfekte“ Resultate erzielen. *Discount Usability Engineering* wurde von Nielsen für diese Vorgehensweise geprägt. Die Vereinfachung der Evaluation war durch expertenorientierte Methoden (besonders wird hier von Nielsen die Heuristische Evaluation genannt) als auch durch Anpassung traditioneller Methoden durch das vereinfachte laute Denken und Scenarios möglich. Kritisiert werden am Discount Usability Engineering häufig die geringe Anzahl der Experten sowie der Einsatz von Endnutzern als Gutachter bei Inspektionen. Aufgrund der geringen Anzahl an Nutzer fehlt auch die statistische Aussagekraft der nutzerorientierten Methoden, die aber in der Praxis eine untergeordnete Rolle spielt.

Discount Usability Engineering scheint sich trotz Kritik wegen der geringen Kosten und relativ einfacher Anwendbarkeit durchzusetzen. [vgl. Schweibenz & Thissen 2003, S. 82] Besonders für die Evaluation von Webseiten wird das Discount Usability Engineering von Nielsen empfohlen, da es die einzige Alternative zur Unterlassung der Usability Evaluation darstellt. Aufgrund der Vielzahl von Webseiten kann der Bedarf an Usability Evaluation durch traditionelle Methoden nicht gedeckt werden.

3.5 Auswertung einer Usability Evaluation

3.5.1 Bewertung der Ernsthaftigkeit der Probleme (Severity Ratings)

Nach Abschluss der Untersuchungen liegt eine ungeordnete Liste der gefundenen Usability Probleme vor. Die Bewertung der Probleme kann helfen, die Beseitigung der einzelnen Probleme mit Rücksicht auf die vorhandenen Ressourcen zu priorisieren. Die Ernsthaftigkeit eines Usability Problems wird durch drei Faktoren bestimmt:

- *Häufigkeit*: Wie viele Nutzer haben das Problem wahrgenommen oder wie oft wurde es von einem Experten angegeben? Die Ernsthaftigkeit des Problems steigt mit zunehmender Häufigkeit.
- *Einfluss*: Wie viel Ärger bereitet das Problem dem Nutzer bei der Arbeit oder wie schätzt der Experte die Möglichkeit ein, dass ein Nutzer das Problem meistern kann? Je mehr das Problem den Nutzer beim Erreichen seiner Ziele beeinflusst (oder ihn sogar davon abhält), desto schwerwiegender ist das Problem.
- *Hartnäckigkeit*: Kann das Problem durch den Nutzer gemeistert werden, wenn es ihm einmal bekannt ist oder wird es ihn wiederholt beeinträchtigen? Auch wenn es nur wenige Probleme betrifft [vgl. Nielsen & Loranger 2006], sind wiederholt störende Probleme mit hoher Ernsthaftigkeit einzuschätzen.

Die Aspekte der Ernsthaftigkeit sollten in einer gemeinsamen Bewertung des Problems zusammengefasst werden. Im einfachsten Fall können Bewertungsstufen wie *niedrig*, *mittel* und *hoch* ausreichen, wobei jedes mit hoher Ernsthaftigkeit bewertete Problem unbedingt behoben werden muss und mittelschwere Probleme in gewissem Maße behoben werden sollten. Die übrigen Probleme können auch zu einem späteren Zeitpunkt bearbeitet werden, da sie weniger störend für den Nutzer sind (solange sie nicht in solch erhöhter Anzahl auftreten, dass ihre Gesamtheit eine frühere Bearbeitung rechtfertigt).

Numerische Bewertungsstufen bieten den Vorteil, dass sie sich besser für statistische Auswertungen eignen, wobei die Skalierung der Stufen je nach Verwendungszweck von detailliert bis relativ grob reichen kann. In den Evaluationen dieser Diplomarbeit wurde die bekannte 5-Stufen Bewertungsskala nach Nielsen verwendet. Auf dieser Skala von 0 bis 4 bewegt sich die Ernsthaftigkeit eines Problems zwischen *kein Usability Problem (0)* und *Usability Katastrophe (4)* (siehe Abschnitte 4.3.3 und 6.3)

Da bei expertenorientierten Methoden die Gutachter die Bewertung vornehmen, wird empfohlen die Bewertung nicht während der Untersuchung der Benutzeroberfläche durchzuführen. Die Experten können dabei erstens nur ihre eigenen Probleme bewerten und zweitens sollte der Prüfer sich lieber ausschließlich auf das Finden von Usability Problemen konzentrieren. Besser ist eine anschließende Bewertung aller gefundenen Probleme, die der Beobachter aus den einzelnen Untersuchungen zusammenfasst, durch jeden Prüfer einzeln (auch hier sollte die Unabhängigkeit gewährleistet sein) in Form einer Befragung. Da jeder Prüfer nur einen Teil der Probleme selbst gefunden hat, kann es notwendig sein, sie dem Prüfer im gewissen Maße zu beschreiben, damit er eine Bewertung abgeben kann.

Nach einer nutzerorientierten Methode, bei der eine Interaktion mit dem Produkt stattfindet, wertet der Untersuchungsleiter (Usability-Experte) die einzelnen Sitzungen aus und gewinnt daraus die Usability Probleme. Er kann dabei auch durch die oben genannten Faktoren die Bewertung der Probleme vornehmen. Die Nutzer selber können auch in einer abschließenden Befragung nach Problemen befragt werden, obwohl solche Aussagen immer im Zusammenhang mit den gemachten Beobachtungen ausgewertet werden sollten (Objektivität und Reliabilität der Evaluation muss gewährleistet sein).

3.5.2 Präsentation der Ergebnisse

Obwohl bei der Vorstellung der Methoden schon vereinzelt auf die Präsentation der Ergebnisse eingegangen wurden, sollen hier einige Kriterien für die Darstellung der Ergebnisse zusammengefasst werden. Denn schließlich ist eine Usability Evaluation erst wirklich erfolgreich, wenn die gefundenen Probleme aus dem Weg geräumt wurden (praktisches Ziel). Dem Usability-Experten muss es gelingen, klar die Probleme zu definieren und Vorschläge zu machen, wie sie behoben werden können. Die Ergebnisse sollten deshalb

- kurz und prägnant dargestellt sein,
- die nötige Detailtiefe haben,
- so positiv wie möglich dargestellt werden,
- ansprechend aufbereitet sein und
- logisch strukturiert dargestellt werden.

[vgl. Vogt, Heuer & Heinsen 2003, S. 222]

Die Präsentation der Ergebnisse muss der Zielgruppe des Ergebnisberichts entsprechen. So sollte beispielsweise ein Bericht für das Management prägnanter sein als für die Entwickler, da jene sich mehr für die Gesamteinschätzung und mögliche Folgen interessieren. Entwickler hingegen bevorzugen detaillierte Darstellungen mit Lösungsvorschlägen für die einzelnen

Elemente. Reports als Darstellung der Ergebnisse werden meist von verschiedenen Zielgruppen gelesen. Deshalb bietet sich hier eine Darstellung auf mehreren Ebenen an (von Allgemeinen zum Detail).

Die Präsentation der Ergebnisse darf nicht nur aus den gefundenen Problemen bestehen, sondern es werden auch Empfehlungen des Usability-Experten zu Lösungen und nächsten Schritten erwartet. Wie genau dies erfolgen muss, ist wieder abhängig von der Zielgruppe (von allgemeine Schritten eines benutzerzentrierten Prozesses bis zu konkreten Designvorschlägen). Außerdem sollte die Kritik möglichst positiv verpackt sein, da sonst die verwendeten Usability Methoden als negativ wahrgenommen werden. Eine Bereicherung der Darstellung durch interessante Ausschnitte aus der Untersuchung (Zitate, Videosequenzen, Fotos) trägt auch zur positiven Aufnahme der Probleme bei. Grundsätzlich ist die interaktive Präsentationsform dem schriftlichen Report in der Praxis vorzuziehen. Mögliche Darstellungsformen sind Briefings, Präsentationen, Workshops und schriftliche Reports.

Briefings sind die einfachste und schnellste Form, Ergebnisse zu präsentieren und sollten aufgrund ihrer informellen Form nur projektintern eingesetzt werden (Auftrag für die Evaluation nicht von außerhalb). Meist werden Ergebnisse der expertenorientierten Methoden auf diese Weise dargestellt (siehe auch Abschnitt 3.2.2). Dabei treffen sich Vertreter des Entwickler-teams und die Usability-Experten (die Gutachter bei der Evaluation), um die Ergebnisse (und deren Bewertung) zu besprechen und gemeinsam Verbesserungsvorschläge zu diskutieren.

Die Präsentation eignet sich für den Einsatz in einer größeren Runde und bei Evaluationen im Auftrag. Hier muss die Detailtiefe der Präsentation der Zuhörerschaft angepasst werden. So sollte z. B. die verwendete Methode kurz vorgestellt werden, wenn die Zielgruppe nicht vertraut mit dem Thema Usability ist. Wird eine Präsentation mit einer anschließenden Diskussion der Ergebnisse verbunden, handelt es sich um einen Workshop, der bis zu einem ganzen Tag dauern kann. Auch hier sollte zu Beginn die verwendete Methode den Teilnehmern vorgestellt werden. Danach sollten die Ergebnisse kurz zusammengefasst werden und schließlich die erste Diskussion erfolgen. Dem Usability-Experten kommt im Workshop die Rolle des Moderators zu, der präsentiert, zwischen den Teilnehmern vermittelt und die Diskussionen lenkt. Der Workshop ist eine sehr anspruchsvolle Methode, da eine gute Vorbereitung der Präsentationen als auch der Diskussionen notwendig ist.

Der schriftliche Report sollte die Präsentation ergänzen, besonders wenn diese nicht sehr detailliert war. Auch hier empfiehlt es sich, zu Beginn die angewandte Methode vorzustellen und danach die Ergebnisse auf verschiedenen Detailebenen zu beschreiben. Die unterschiedliche Ernsthaftigkeit der Probleme muss dabei betont werden. Je nach verwendeter Methode empfiehlt sich eine unterschiedliche Strukturierung für den Report. Bei expertenorientierten Methoden hat sich die Einteilung nach Komponenten des Produkts (z. B. Druck-Dialog) bewährt. [vgl. Vogt, Heuer & Heinsen 2003, S. 228] Für die Dokumentation von Usability Tests bietet sich eher die Darstellung nach gefundenen Problemen oder anhand der Aufgaben an. Für letztere gibt es eine Reportvorlage namens *Common Industry Format (CIF)*¹¹, die ein gemeinsames Format vorgibt, um die Qualität der Usability Arbeit zu sichern.

¹¹ Als *ISO/IEC 25062:2006 – Software product Quality Requirements and Evaluation (SQuaRE) – Common Industry Format (CIF) for usability test* herausgegeben. Weiterführende Informationen dazu unter [NIST 2006]

4 Konzeption, Gestaltung und Heuristische Evaluation der Webhosting Webanwendung

Der Auftrag ist es, eine Webanwendung zu entwickeln, die den Kunden des Webhosting Anbieters Netfarm (nachfolgend Auftraggeber genannt) eine Verwaltungsoberfläche bietet, mit der sie ihre Leistungen weitgehend autonom anpassen können. Zudem soll ein Administrationsbereich für den Betreiber erstellt werden, welcher zum Schluss eine umfassende Online-Verwaltung des Webhosting Servers bietet.

Die zu entwickelnde Verwaltungsoberfläche tendiert nach den Kriterien für die Unterscheidung zwischen Webseite und Webapplikation [vgl. Fowler & Stanwick 2004, S. 14f.] eindeutig zu den Webapplikationen, da sie einen Login erfordert, bevor der Benutzer etwas in der Anwendung bewirken kann. Dann können Daten durch die Interaktion mit Formularen erstellt, geändert (gespeichert) und gelöscht werden. Die gesamte Interaktion unterliegt dabei einer gewissen Kontrolle, damit sensible Bearbeitungsvorgänge ausreichend vor Datenverlust abgesichert sind. Letztlich ist zum Beenden der Anwendung wieder ein Abmelden erforderlich.

In diesem Kapitel wird die Entwurfsphase für die Webhosting Webanwendung beschrieben. Die entstandenen Design-Entwürfe wurden, wie es dem Usability Engineering entspricht, einer Überprüfung mittels Heuristischer Evaluation unterzogen. Über deren Durchführung und Ergebnisse wird in diesem Kapitel auch berichtet.

4.1 Konzeption

Der Auftraggeber hat bis jetzt keine solche Verwaltungsoberfläche. Deshalb wurde die Anwendung sowohl inhaltlich als auch technisch konzipiert. In Zusammenarbeit mit dem Auftraggeber wurde der Funktionsumfang der Anwendung festgelegt. Es wird einen Bereich für Kunden und einen für Administratoren (Betreiber) geben, die den beiden Benutzergruppen sowohl Informationen als auch die Möglichkeit zum Erledigen von Aufgaben bezüglich des Webhosting bieten sollen. Damit soll der bisherige Verwaltungsaufwand für beide Seiten verringert werden, da Kunden einige Funktionen selbst verwalten können (anstatt Aufträge an den Betreiber zu erteilen) und der Betreiber eine grafische Oberfläche zur besseren Übersichtlichkeit seiner Aufgaben geliefert bekommt (anstatt in einer nichtgrafischen Konsolenumgebung zu arbeiten).

4.1.1 Funktionsumfang Kundenbereich

Im Kundenbereich sollen dem Nutzer folgende Funktionen zur Verfügung stehen:

- Bearbeitung seiner persönlichen Daten
- Änderung oder bei Verlust Neuanforderung der Passwörter der diversen Zugangsdaten
- Bestellen und Kündigen von Webhosting Leistungen

- Anlegen und Bearbeiten von E-Mail-Konten sowie Hinzufügen, Bearbeiten und Löschen von E-Mail-Adressen innerhalb dieser Konten
- Hinzufügen und Entfernen von Domains zu/aus seinem Webhosting Konto sowie Anlegen und Löschen von Subdomains
- Zugriffsmöglichkeiten auf andere Verwaltungsoberflächen (phpMyAdmin, Webmail)

Darüber hinaus soll die Webanwendung dem Kunden jederzeit Informationen über den aktuellen Stand seiner Webhosting Leistungen bieten. Dazu gehören:

- Informationen zum Webhosting Konto des Kunden und möglichen Erweiterungen
- Informationen zum Server (Status, Wartungsarbeiten etc.)
- Informationen zur verbrauchten Menge des E-Mail-Speichers und Webspace

Vervollständigt soll die Anwendung mit einer Hilfe und einem Kontaktformular werden. Für weitere Versionen der Webanwendung sind zusätzlichen Funktionen wie ein Online Rechnungssystem und HTTPS Unterstützung, die aber lediglich in der Datenbankkonzeption bisher berücksichtigt wurden.

4.1.2 Funktionsumfang Administratorbereich

Im Administratorbereich sollen dem Nutzer folgende Funktionen zur Verfügung stehen:

- Anlegen, Bearbeiten und Löschen von Kunden und ihren Leistungen, Domains, E-Mail-Konten und Datenbanken (insbesondere Aufgaben, die der Kunde aus Sicherheitsgründen nicht selbst erledigen darf wie Änderung der Webspace Größe)
- Bearbeitung der Preise für Domains und Leistungen sowie Erstellen von Angebotspaketen
- Bearbeitung des Status von Kunden, Domains und Leistungen in einem Auftragssystem
- Zugriffsmöglichkeiten auf andere Verwaltungsoberflächen (phpMyAdmin, Webmail)

Darüber hinaus werden dem Administrator Suchfunktionen für die einzelnen Bereiche (Kunden, Domains, E-Mail-Konten) geliefert, damit ein schnelles Finden der Daten möglich ist. Erweiterungen der Funktionalität im Kundenbereich bedingen auch eine Anpassung der Funktionen des Administratorbereichs (z. B. Zahlungen eintragen).

4.1.3 Informationsarchitektur und Navigationsstruktur

Die genannten Funktionen wurden in jedem Bereich in sechs Kategorien aufgeteilt, die im weiteren Verlauf des Entwurfs zu Menüpunkten in der Navigation wurden. Ziel bei der Aufteilung war es, eine benutzergerechte Zusammenfassung von Funktionen zu bieten (Kategorisierung), damit sie schnell gefunden werden können. Hierbei mussten die unterschiedlichen Erfahrungsniveaus der Kunden und der Administratoren bezüglich Internetnutzung und Webhosting Termini beachtet werden.

Tabelle 2: Informationskategorien und Menüpunkte im Kundenbereich

Informationskategorie	Menüpunkt
Status und Information	Start
Persönliche Daten	Meine Daten
Webhosting Leistungen	Mein Konto
Domains	Domain
E-Mail-Konten	E-Mail
Datenbanken	Datenbank

Tabelle 3: Informationskategorien und Menüpunkte im Administratorbereich

Informationskategorie	Menüpunkt
Status und Aufgabenübersicht	Start
Verwaltung der Angebote und Preise	Angebot
Kunden (Persönliche Daten und Leistungen)	Kunden
Domains	Domains
E-Mail-Konten	Mailboxen
Spezielle Funktionen	Extras

Für jeden Informationsbereich ist eine detaillierte Struktur der darin enthalten Informationen und Funktionen notwendig, welche auch konzipiert wurde. Für die Navigationsstruktur ergibt sich daraus eine weitere Navigationsebene (außer für den Menüpunkt *Start*). Im Rahmen dieser Diplomarbeit wird diese Struktur nur für die Kategorien *E-Mail-Konten* (unvollständig) und *Status* mit Inhalt gefüllt. Den anderen Menüpunkten wird lediglich die nächste Navigationsebene hinzugefügt.



Abbildung 7: Zweite Navigationsebene im Kundenbereich für den Menüpunkt *E-Mail*



Abbildung 8: Zweite Navigationsebene im Administratorbereich für den Menüpunkt *Mailboxen*

Die Kategorien werden nicht nur für den Nutzer von Bedeutung sein, sondern auch bei der Implementierung genutzt werden. Es werden Klassen zu den einzelnen Menüpunkten erstellt werden, die die Aktionen in den Bereichen steuern sollen (Geschäftslogik der Anwendung).

4.1.4 Systemvoraussetzungen

Das System, auf dem die Webanwendung letztlich laufen soll, wird eine LAMP (Linux-Apache-MySQL-PHP5) Konfiguration besitzen. Um der Komplexität der Anwendung gerecht zu werden, wurde entschieden ein PHP-Framework für die Entwicklung einzusetzen. Aus der Auswahl von Open Source Frameworks in den unterschiedlichsten Entwicklungsphasen wurde das relativ neue Framework *Symfony* für die Entwicklung gewählt, da es eine aktive Weiterentwicklung, umfangreiche Dokumentation und Unterstützung bietet. Eine Einführung in das Framework sowie die technische Realisierung damit wird im Kapitel 5 beschrieben. Für den Endanwender werden für die Nutzung der Anwendung lediglich ein Internetzugang und ein beliebiger grafischer Browser notwendig sein.

4.1.5 Datenbankkonzeption

Im System, für das entwickelt wird, wird für die E-Mail Verwaltung dbmail mit MySQL Datenbank verwendet. Deshalb war für die Speicherung der E-Mail relevanten Daten (Benutzerdaten, Mailbox Größe, E-Mail-Adressen) die Datenbankstruktur vorgegeben (Anhang A). Relevant für die geplante Webanwendung waren die Tabellen *dbmail_users*, welche die E-Mail-Konten verwaltet und *dbmail_aliases*, welche die E-Mail-Adressen zu den Konten verwaltet (Abbildung 9).

dbmail_users	dbmail_aliases
user_idnr: BIGINT	alias_idnr: BIGINT
userid: VARCHAR(100)	alias: VARCHAR(100)
passwd: VARCHAR(34)	deliver_to: BIGINT
client_idnr: BIGINT	client_idnr: BIGINT
maxmail_size: BIGINT	
curmail_size: BIGINT	
encryption_type: VARCHAR(20)	
last_login: DATETIME	
UNIQUE userid	

Abbildung 9: Schema der Tabellen *dbmail_users* und *dbmail_aliases* der Maildatenbank

Zusätzlich wurde eine Datenbank entworfen, welche die Webhosting relevanten Daten speichern wird (Anhang A). Die Verbindung zwischen den beiden Datenbanken wurde über den Kunden hergestellt. In der Anwendungsdatenbank befindet sich für das Objekt Kunde die Tabelle *customer*, die als Primärschlüssel die *customer_id* besitzt. In der Maildatenbank existierten dazu entsprechende Fremdschlüssel namens *client_idnr* in den Tabellen *dbmail_users* und *dbmail_aliases*.

Bei der Konzeption der Tabellen wurde von den Hauptobjekten (Kunden, Domains, Angebote, Rechnungen etc.) der Anwendung ausgegangen, da das Datenbankmodell Grundlage für einen Teil der Klassen der Anwendung sein wird (siehe Abschnitt 5.3).

4.2 Gestaltung der Benutzeroberfläche

Bei der Gestaltung wurden Usability Richtlinien und allgemeinen Kriterien zur Gestaltung von Bildschirmanwendungen berücksichtigt, damit die Benutzeroberfläche möglichst effizient bedient werden kann. Außerdem soll durch das Design die Sachlichkeit der Anwendung unterstrichen werden. Im folgenden Abschnitt werden Aspekte der Gestaltung erläutert und dabei auf berücksichtigte Usability Richtlinien hingewiesen.

4.2.1 Bildschirmeinteilung

Bei einer Webanwendung beeinflussen die Bildschirmauflösung und die Browsergröße die Darstellung der Seite. Die Variabilität dieser Faktoren macht es notwendig, einen geeigneten Kompromiss bei der Gestaltung zu wählen, so dass möglichst viele Nutzer den Großteil der Seite auf den ersten Blick erfassen können. Dabei ist außerdem zu beachten, dass die Aufmerksamkeit der Nutzer besonders auf Elemente links und oben gerichtet ist und in Richtung rechte untere Ecke abnimmt.

Es wurden für den Kunden- und den Administratorbereich zwei unterschiedliche Bildschirmaufteilungen entworfen (Abbildung 10 und Abbildung 11). Beide wurden für eine variable Breite entwickelt, die auch auf einer Auflösung von 800 x 600 px noch ohne horizontales Scrollen darstellbar sein wird (aktuell wird eine Auflösung von 1024 x 768 px als die meistverwendete Auflösung angesehen [vgl. Nielsen & Loranger 2006]).

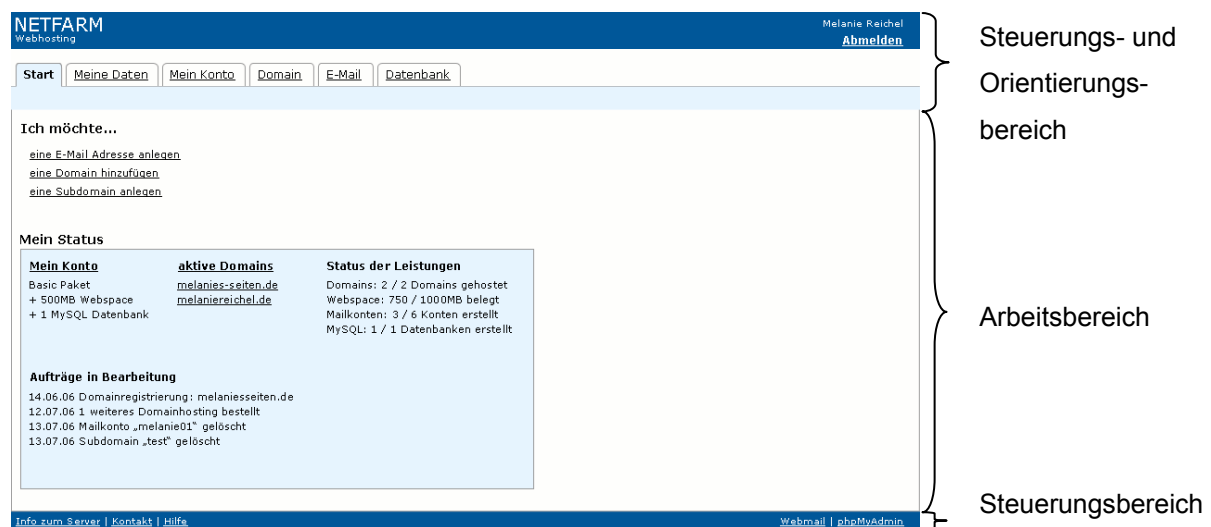


Abbildung 10: Entwurf der Startseite des Kundenbereichs mit Bereichsaufteilung

Die Kundenoberfläche wurde in drei waagerechte Bereiche eingeteilt. Der Reihe nach gibt es einen Steuerungs- und Orientierungsbereich, in dem Logo, Anmeldeinformationen und das Menü untergebracht wurden, den Arbeitsbereich und einen weiteren Steuerungsbereich für Links zu Nebenfunktionen, die nicht direkt mit dem Webhosting zu tun haben.

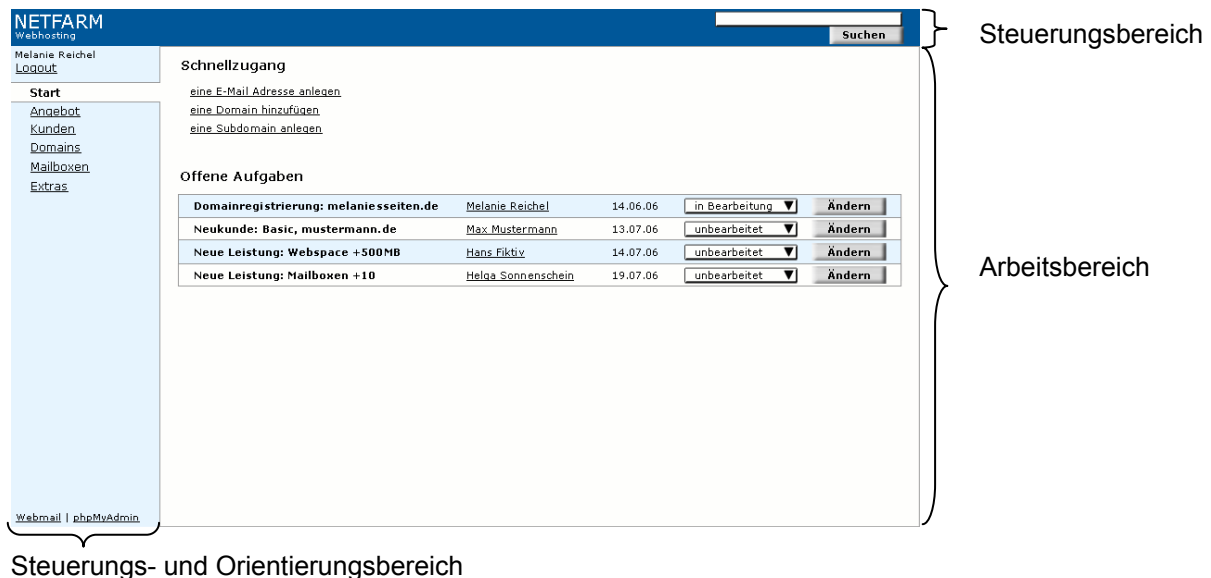


Abbildung 11: Entwurf der Startseite des Administratorbereichs mit Bereichsaufteilung

Im der Administratoroberfläche wurde ein Steuerungs- und Orientierungsbereich am linken Seitenrand senkrecht platziert und beinhaltet das Menü, Anmeldeinformationen und weitere Links. Am oberen Rand gibt es einen weiteren Steuerungsbereich, der eine Suchfunktion und das Logo (auch für Navigationszwecke genutzt) enthält. Den Rest der Seite wird vom Arbeitsbereich belegt.

4.2.2 Navigation

Navigationselemente sind Menüs, die Links in Kategorien zusammenfassen, und Verlinkungen, die im Textzusammenhang stehen. Da sich Links eindeutig vom restlichen Text abheben und dem Nutzer bekannt formatiert werden sollten, wurden alle Links unterstrichen. Beim Berühren eines Links mit der Maus, verschwindet die Unterstreichung und der Mauszeiger verändert sich in die typische Handform, wodurch erkennbar ist, dass es sich um ein interaktives Element handelt, da es auf Nutzereingaben (Bewegung der Maus über den Link) reagiert. Befindet sich der Nutzer auf der Seite, zu welcher der Link führte, wird er fett angezeigt und reagiert nicht mehr auf Nutzereingaben. Dies entspricht einer Usability Richtlinie, die besagt, dass nicht auf die aktuelle Seite verlinkt werden soll.

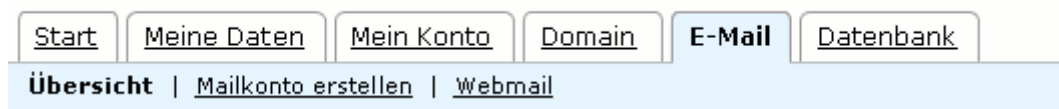


Abbildung 12: Entwurf des Menüs im Kundenbereich (aktiver Menüpunkt E-Mail Übersicht)

Das Menü im Kundenbereich (Abbildung 12) wurde über die Metapher der Karteikartenreiter umgesetzt und waagrecht im oberen Bereich der Seite angeordnet, in welchem Nutzer bevorzugt nach Navigationselementen suchen. Die Metapher wird von den meisten Nutzern intuitiv verstanden. Um dies zu gewährleisten, muss durch Hervorhebung der Unterschied zwischen aktiver Karte und inaktiven Karteikarten klar gemacht werden, wie hier durch die blaue Farbe und die abgrenzende Linie um die aktive Karte.

Die Untermenüpunkte bilden auf dem durchgängigen blauen Untergrund eine Einheit mit der Hauptkategorie (z. B. E-Mail in Abbildung 12).



Abbildung 13: Entwurf des Menüs im Administratorbereich (erste und zweite Navigationsebene)

Im Administratorbereich wurde das Menü an den linken Seitenrand gesetzt, was auch eine konventionelle Position dafür darstellt. Hier wurden der aktive Menüpunkt und seine Unterpunkte durch eine Verbindung mit dem Arbeitsbereich aufgrund derselben Hintergrundfarbe vom restlichen hellblauen Menü abgehoben. Die abgrenzende Linie wurde für den aktiven Menübereich unterbrochen.

Auch Suchfelder können als eine Form von Navigationselement angesehen werden, da auch sie von Nutzer zum bewegen durch die Seite verwendet werden. Suchfunktionen machen allgemein auf großen Webseiten (mehr als 100 Seiten) oder auf datenintensiven Webseiten Sinn. Für diese Webanwendung wurde im Administratorbereich eine allgemeine Suchfunktion am rechten oberen Rand platziert. In den einzelnen Menükategorien sind spezielle Suchfunktionen geplant.

Im Kundenbereich wurde auf eine Suchfunktion verzichtet, um die Navigation zu vereinfachen, da besonders weniger erfahrene Nutzer (Zielgruppe des Kundenbereichs) eher Menüs zur Steuerung verwenden.

4.2.3 Interaktionselemente

Für Interaktionselemente wie Eingabefelder, Buttons und Links ist es wichtig, dass sie ihre Funktion klar visualisieren. Beschriftungen sollten klar machen, was die Interaktion mit diesem Element bewirken wird. Deshalb wurde besonders im Kundenbereich darauf geachtet, dass die Beschriftung der Eingabefelder die korrekte Eingabe unterstützt. Außerdem wurde für die Eingabefelder eine angemessene Breite gewählt, die auch längere Eingaben komplett anzeigen. Nach einer Usability Richtlinie von Nielsen sollte beispielsweise ein Suchfeld mindestens 27 Zeichen breit sein.



Abbildung 14: Suchbutton im Entwurf (links) und in der HTML-Umsetzung (rechts)

Buttons wurden in den Entwürfen ähnlich den Standard HTML Buttons gezeichnet, die von Nutzer intuitiv verstanden werden. Während der späteren Umsetzung in HTML wurden diese

jedoch an das grafische Konzept der Webanwendung angepasst. Um den Button einen 3D-Effekt zu geben, wurde mit unterschiedlicher Rahmendicke rechts/unten und links/oben gearbeitet (Abbildung 14). Diese Änderung konnte in der Entwicklung durchaus vorgenommen werden, da der Prototyp im Nutzertest auf seine Bedienbarkeit untersucht wird.

Im Kundenbereich wurde des Weiteren besonders auf eine verständliche Beschriftung der Buttons und Links geachtet. So wurde im Menü z. B. der Link *Mailkonto erstellen* eingesetzt anstatt eines allgemeinen *Anlegen* Links, wie es im Administratorbereich durchaus sinnvoll ist, da es sich um erfahrene Nutzer handeln wird, die solche generellen Beschriftungen aufgrund ihrer Erfahrung schnell erfassen.

4.2.4 Schriftformat

Allgemeine Richtlinien zur Bildschirmgestaltung sagen aus, dass serifenlose Schriftarten am Monitor besser lesbar sind als Serifenschriften. Die Serifen erscheinen aufgrund der geringeren Punktauflösung der Bildschirme (meist 96 dpi) als vergleichsweise im Druck (300 dpi) in kleinen und mittelgroßen Schriftgraden (kleiner 14 pt) verschwommen und erschweren damit die Lesbarkeit der Buchstaben.

Deshalb wurde in der Webanwendung durchgängig die Schriftart Verdana, die speziell für das Web entwickelt wurde, eingesetzt. Die Schriftgröße des Textes wurde auf 11 pt im Entwurf gesetzt. Für die HTML Umsetzung wurde sich für die Schriftgröße 12 px entschieden, da sie in vielen Browsern der Normalgröße (100 %) entspricht und noch besser lesbar ist. Einträge des Menüs und Überschriften wurden durch einen höheren Schriftgrad (14 px im Menü, 16 px bzw. 14 px für Überschriften) in ihrer Bedeutung betont. Umgesetzt wurden die Schriftgrößen durch Prozentangaben (100 % entspricht 12 px) in den CSS Dateien, um den Nutzer die Möglichkeit zu geben, die Schriftgröße über die Browserfunktionen (vergrößern/verkleinern) anpassen zu können.

4.2.5 Farben

Durch Farben und Kontraste sollte vor allem die Lesbarkeit gefördert und die Orientierung auf der Seite unterstützt werden (siehe auch Navigation). Ebenso können Farben eine gewisse Stimmung auf der Webseite vermitteln. Für einen guten Kontrast wurde meistens als Hintergrund eine nahezu weiße Farbe eingesetzt. Es hat sich als angenehmer für das Lesen am Bildschirm erwiesen, wenn kein reines Weiß als Hintergrundfarbe eingesetzt wird, da dies zu grell wirkt und das Auge deshalb schneller ermüdet.

Um bestimmte Bereiche hervorzuheben wurde mit zwei verschiedenen Blautönen gearbeitet. Beide Farben vermitteln Glaubwürdigkeit, Seriosität und Frische im westlichen Kulturkreis, was zum Thema der Anwendung passt. Das dunkle Blau wurde als Rahmung am oberen und unteren Ende der Webseite eingesetzt, auf der Schrift in der sonst als Hintergrund verwendete Farbe gesetzt wurde. Die aktive Kategorie des Menüs (Karteikarte) und einige Felder im Arbeitsbereich wurden durch das helle Blau abgegrenzt. Die ansonsten verwendete schwarze Schrift sowie die Fehler- und Statusmeldungen in Rot und Grün ergeben mit dem Hellblau

auch einen guten Kontrast. Für Rahmenlinien wurde ein neutrales Grau gewählt, um die Sachlichkeit der Webanwendung zu betonen.



Abbildung 15: Die beiden charakteristischen Blautöne der Webanwendung und die Hintergrundfarbe

Die Farben wurden aber auch aufgrund der Tatsache ausgewählt, dass ein gewisser Prozentsatz der Nutzer eine Form von Farbenblindheit hat und sich dann vorwiegend auf den Kontrast und andere Merkmale verlassen muss. Obwohl Rot und Grün als Schriftfarbe für Fehler- und Statusmeldungen benutzt wird, wurde am Anfang jeder Meldung als weiteres Merkmal *Fehler:* oder *Status:*, um Rot-Grün-Blinden als auch Nutzern von Screenreadern o. ä. ein Form der Hervorhebung zu liefern¹².



Abbildung 16: Statusfarbe Grün und Fehlerfarbe Rot als Schriftfarbe verwendet

Werte für Farben, Schriftarten und -größen sowie andere Formatierungen wurden in einem Style Guide (Anhang A) festgelegt, um während der Entwicklung der HTML-Elemente konsistent zu bleiben.

4.3 Heuristische Evaluation

Wie in Kapitel 3 beschrieben eignet sich die Heuristische Evaluation auch für die Inspektion in sehr früher Entwicklungsphasen eines Programms. Es waren zwar keine Skizzen auf Papier, die es zu untersuchen galt, aber die vorhandenen Designentwürfe konnten noch nicht wirklich als Prototyp eingesetzt werden, um einen Nutzertest durchzuführen.

Die durchgeführte Heuristische Evaluation wurde nicht wie üblich an einem bestimmten Ort durchgeführt, sondern fand als *Fernuntersuchung* statt. Dieses Vorgehen sollte vor allem zum Vorteil für die Organisation sein, da die Prüfer nicht alle vor Ort in Dresden waren. Am Ende dieses Abschnitts werden Probleme diskutiert, die durch die Art der Durchführung aufgetreten sind.

¹² Auf das Thema Accessibility (Barrierefreiheit) wird in dieser Diplomarbeit nicht weiter eingegangen. Informationen zu Accessibility im Web können auf [World Wide Web Consortium 2006] abgerufen werden.

4.3.1 Planung der Untersuchung

Untersuchungsgegenstand

Ziel der Evaluation war es, schon in den Design-Entwürfen erste Usability Probleme zu lokalisieren und diese bereits in der Entwurfsphase zu beseitigen. Dazu wurden den Gutachtern die in Photoshop erstellten Screen Designs für einzelne Seiten des späteren Prototyps vorgelegt. Die Seiten wurden als Flashfilm so miteinander verbunden, dass in dem begrenzten Rahmen der vorhandenen Entwürfe auch die Navigation veranschaulicht werden konnte. Einige der Entwürfe waren mit Fehler- oder Statusmeldungen versehen, die beispielhaft das Konzept der Fehler- und Statuswiedergabe darstellen sollten.

The screenshot shows the 'NETFARM Webhosting' customer interface. At the top, there's a navigation bar with links like 'Start', 'Meine Daten', 'Mein Konto', 'Domain', 'E-Mail', and 'Datenbank'. The 'E-Mail' section is active, showing 'E-Mail Übersicht' and 'Schnell: E-Mail Adresse anlegen'. There are two input fields: 'Neue E-Mail Adresse' with the value 'meineueemail' and 'im Mailkonto' with a dropdown menu. Both fields have red error messages: 'Fehler: Keine Domain ausgewählt. Bitte wählen Sie eine Domain aus.' and 'Fehler: Kein Mailkonto ausgewählt. Bitte wählen Sie ein Mailkonto aus.' respectively. Below these fields is an 'Anlegen' button. At the bottom, there's a section 'Meine Mailkonten' listing three accounts: 'melanie01', 'melanie02', and 'melanie03', each with links for 'Passwort ändern' and 'E-Mail Adressen einblenden'.

Abbildung 17: Entwurf aus dem Kundenbereich mit Fehlermeldungen

Während einer Untersuchung, in der sich der Untersuchungsleiter mit dem Prüfer getroffen hätte, wären das Material dem Prüfer verbal erklärt worden. Da dies nicht möglich war, wurde ein Dokument mit Beschreibungen beigelegt (siehe Anhang B). Darin wurden die allgemeine Seitenstruktur der geplanten Anwendung, die möglichen Navigationspunkte im Flashfilm und Funktionen von im Flashfilm unbenutzbaren Links erläutert. Außerdem enthielt das Dokument eine ausführliche Auflistung möglicher Fehler- und Statusmeldungen zu den verfügbaren Seiten, welche bei der Evaluation mit berücksichtigt werden sollten (einige Heuristiken bezogen sich auf die Fehler- und Statusdarstellung).

Gutachter

Die Gruppe der Gutachter setzte sich aus vier ehemaligen Mitstudenten zusammen. Sie wurden ausgewählt, da sie alle über ein Grundwissen zum Thema Software-Ergonomie und Usa-

bility verfügen und auch die technischen Aspekte der zu untersuchenden Anwendung (wenn auch nur deren Entwürfe bis dahin vorhanden waren) durch ihr Studium einschätzen können. Im Sinne von Nielsen können sie damit als Usability-Experten bezeichnet werden. Die Rekrutierung der Prüfer aus Mitstudenten hat den Nachteil, dass durch das persönliche Verhältnis zum Entwickler die Objektivität der Prüfer gemindert wird. Die Ergebnisse beweisen aber, dass alle kritisch ihre Aufgabe erfüllt haben und die Qualität der Untersuchung damit nicht negativ beeinflusst wurde.

Die Inspektion sollten die Gutachter zu hause durchführen können. Dafür war es notwendig die Unterlagen so aufzubereiten, dass sie per Mail zugeschickt und ohne persönliche Einweisung verstanden werden konnten. Ich habe als Untersuchungsleiterin die Unterlagen entsprechend aufbereitet, verschickt und die Ergebnisse ausgewertet.

Heuristiken

Die verwendeten Heuristiken wurden auf der Basis der 10 Heuristiken von Nielsen erstellt [vgl. Nielsen 1994]. Da diese allgemein formuliert sind, wurden sie verfeinert, indem mehrere konkrete Kriterien erstellt wurden, wobei sich auch an den Grundsätzen des Interaktion Designs von Bruce Tognazzini orientiert wurde [vgl. Tognazzini 2003]. Insgesamt wurden 22 Heuristiken aufgestellt, die zur besser Übersicht in vier Kategorien eingeteilt wurden.

Kategorie: Fehler und Status

- Verständlichkeit und Qualität von Fehler- und Statusmeldungen sollten bewertet werden.
- Qualität der Fehlermeldungen
Beschreiben die Fehlermeldungen das Problem präzise und geben eine konstruktive Lösung vor?

Kategorie: Interaktion

- Die Interaktionselemente sollten auf Konsistenz, Einfachheit und Effizienz für den Nutzer untersucht werden.
- Leichtes Auffinden von Funktionen
Ist die Oberfläche so gestaltet, dass sie beim Auffinden von Inhalten und Funktionen dem Benutzer behilflich ist?

Kategorie: Erkennen und Orientieren

- Erkennbarkeit der Funktionalität, klare Standortbestimmung und Verständlichkeit der Texte waren in dieser Kategorie zu bewerten.
- Eindeutige Standortbestimmung
Ist immer klar für den Benutzer, an welcher Position in der Navigation er sich befindet?

Kategorie: Gestaltung

- Die Gestaltung sollte auf einfache und klare Darstellung, passende Größenverhältnisse und Lesbarkeit untersucht werden.
- Klare räumliche Anordnung
Sind zusammengehörige bzw. zusammenverwendbare Interaktionselemente durch räumliche Nähe und/oder farbliche Identität als solche erkennbar?

Liste 5: Kategorien der Heuristiken mit inhaltlicher Kurzcharakteristik und Beispiel

Zu jedem Kriterium wurde eine Entscheidungsfrage gestellt, da sie verständlicher und konkreter erschien als der bloße Stichpunkt. Die komplette Liste der Heuristiken ist im Anhang C zu finden. Die Gefahr einer Entscheidungsfrage war, dass die Prüfer keine genaue Problembeschreibung liefern, sondern nur die Frage mit Ja oder Nein beantworten. Deshalb wurde in den Hinweisen zur Evaluation eine deutliche Anweisung gegeben. Die Ergebnisse weiter unten im Abschnitt sprechen für einen Erfolg dieser Methode.

4.3.2 Durchführung der Untersuchung

Die Evaluation wurde in drei Phasen durchgeführt. Jede Phase begann mit einer Mail der Untersuchungsleiterin an die Gutachter mit den entsprechenden Unterlagen und jeweils einem Dokument mit Hinweisen, wie mit diesen Unterlagen umzugehen sei.

Verstehen der Heuristiken

Den Prüfern wurden die erstellten Heuristiken vorgelegt, damit sie diese vorab lesen und Fragen dazu stellen konnten. Begleitend bekamen die Gutachter in dieser ersten Mail einen Zeitplan für den Verlauf der Untersuchung mitgeteilt, der sehr großzügig gestaltet war und angepasst werden konnte. Den Prüfern wurde aufgetragen, Fragen zu den Heuristiken und Anmerkungen zum Zeitplan zurückzuschicken. Nachdem alle Prüfer geantwortet hatten, wurden die Kriterien mit Antworten zu den Fragen der Prüfer kommentiert. Zum Zeitplan wurden keine Änderungswünsche mitgeteilt.

Durchführung der Evaluation

Die Gutachter bekamen die kommentierten Heuristiken zurück. Dazu erhielten sie den Untersuchungsgegenstand wie oben beschrieben. In den Hinweisen zur Untersuchung bekamen die Prüfer die Anweisung die Untersuchung ohne Unterbrechung und alleine durchzuführen. Außerdem wurde darauf hingewiesen, sich erst nach Abschluss der dritten Phase untereinander auszutauschen, da die Qualität der Bewertung negativ beeinflusst werden könnte.

Die Durchführung selbst wurde in eine Einarbeitungsphase und eine Untersuchungsphase eingeteilt. Während ersterer sollten die Gutachter sich einen Überblick verschaffen, indem sie den Flashfilm gemeinsam mit den Beschreibungen durchsehen sollten. Die Heuristiken wurden dabei noch nicht betrachtet. In der Untersuchungsphase sollte der Flashfilm auf jede Heuristik hin untersucht werden. Die erkannten Probleme waren zu den entsprechenden Kriterien präzise und mit Begründung zu notieren. Traten bei einem Element Probleme zu mehreren Kriterien auf, sollten diese auch mehrmals notiert werden. Verbesserungsvorschläge, die in kein Kriterium eingeordnet werden konnten oder kein direktes Problem darstellten, konnten die Prüfer als *Eigene Vorschläge* notieren. Die kompletten Dokumente zur Evaluation befinden sich im Anhang B.

Bewertung der Probleme

Die nach der Evaluation zurückgeschickten Notizen der Gutachter zu den Problemen waren recht unterschiedlich in Umfang und Präzision. Daraus wurde eine einheitliche Tabelle der gefundenen Usability Probleme erstellt. Dabei wurden Probleme, die von mehreren Prüfer

gleich oder ähnlich beschrieben wurden, zusammengefasst. Jedes Problem wurde entsprechend seiner Zuordnung zu einer Heuristik aufgelistet, also auch mehrmals, wenn es bei der Evaluation so eingeordnet wurde.

Qualität der Statusmeldungen

Sind Statusmeldungen auf den ersten Blick sichtbar und als solche erkennbar?

- Die Unterscheidung zwischen Status- und Fehlermeldung ist nicht gegeben. Siehe 10/20 und 19/20 Ändern der Mailboxgröße: Falsches Zeichen ist ein Fehler, Systemfehler ist wiederum eine Statusmeldung wert?
- Positionierung offenbar immer an nächstgelegener Überschrift, jedoch nicht im Bild 20/20 Bereich Aliases (hinter den Buttons)

Liste 6: Beispiel einer Problembeschreibung durch einen Gutachter

Dazu haben die Prüfer zu einigen Problemen Lösungen vorgeschlagen oder aber unabhängig Vorschläge gegeben. Die Vorschläge wurden auch in der erstellten Tabelle erfasst, da sie auch bewertet werden sollten, um einschätzen zu können, in wie weit diese die Usability der Webanwendung steigern können.

Es ergab sich eine Liste mit 49 Einträgen (inkl. vier Vorschlägen und drei mehrmals eingeordneten Problemen), die bewertet werden sollten. Die komplette Liste befindet sich im Anhang D; hier wird ein beispielhafter Ausschnitt gegeben.

Tabelle 4: Auszug aus der Auflistung der gefundenen Usability Probleme durch die Experten

<i>Problembeschreibung</i>	<i>Lösungsvorschlag</i>	<i>Heuristik</i>	<i>Prioritätsstufe</i>
Der Admin kann beim Ändern des Passworts die Funktionen <i>Generieren</i> und <i>Ändern</i> nicht eindeutig unterscheiden.	Hier wäre eine konsistente Nutzung von Buttons besser.	Zweckgebundene Interaktionselemente	
Der Weg zurück zu den Suchergebnissen ist nicht klar genug. Besonders nachdem eine Aktion auf einer Seite nach den Ergebnissen (Mailbox bearbeiten) ausgeführt wurde, kann der Zurück-Button des Browsers problematisch werden.	Evtl. Link <i>Suchergebnisse</i> in Menü einbauen oder Link auf den Nachfolge-seite anbieten.	Benutzerfreiheit	
Begriffe wie Mailkonto, Hosting, Domain werden im Kundenbereich nicht erklärt.	Die Hinweise wäre eine gute Möglichkeit für Erklärungen.	Textverständnis	

Die Gutachter sollten die Ernsthaftigkeit jedes Problems anhand der Faktoren *Häufigkeit*, *Einfluss* und *Hartnäckigkeit* bestimmen und eine Priorität nach folgender Tabelle vergeben:

Tabelle 5: Stufen der Ernsthaftigkeit eines Usability Problems (nach [Schweibenz & Thissen 2003, S. 114])

Prioritätsstufe	Beschreibung des Problems
0	Dieses Problem stellt aus meiner Sicht kein Problem für die Benutzerfreundlichkeit dar.
1	Dieses Problem ist lediglich kosmetisch. Es muss nicht unbedingt behoben werden, außer wenn genügend Zeit dazu bleibt.
2	Dieses Problem ist nicht schwerwiegend. Es sollte behoben werden, hat aber eine niedrige Priorität, was die Beseitigung betrifft.
3	Dieses Problem ist schwerwiegend. Es sollte dringend behoben werden und hat eine hohe Priorität, was die Beseitigung betrifft.
4	Dieses Problem ist katastrophal. Es muss unbedingt behoben werden, bevor die Webseite öffentlich zugänglich gemacht wird. Es hat deshalb höchste Priorität, was die Beseitigung betrifft.

4.3.3 Auswertung der Ergebnisse

Ziel der Auswertung war es festzustellen, welche Probleme im weiteren Verlauf der Entwicklung in welcher Reihenfolge behoben werden müssen und welche vernachlässigt werden können, die sie geringe Priorität haben. Die Bewertungen der vier Gutachter wurden gemittelt und gerundet. Dabei ergab sich, dass kein Problem mit Stufe 0 bewertet und keines mit Stufe 4 bewertet wurde. 12 Probleme wurde die Stufe 1 gegeben, 29 Probleme erhielten die Stufe 2 und acht Probleme wurden mit der Stufe 3 bewertet (Tabelle 6).

Tabelle 6: Schwerwiegende Usability Probleme mit Lösungsvorschlag der Experten

Problembeschreibung	Lösungsvorschlag
<i>Fehler und Status</i>	
Einige Formulierungen der Fehlermeldungen sind missverständlich.	<i>vorderen Teil</i> der gewünschten Adresse; hier sollte besser ein anderer Ausdruck gefunden werden oder der entsprechende String angezeigt werden. Fehler: <u>Altes</u> Passwort fehlt. Bitte geben Sie ihr <u>aktuelles</u> Passwort ein. Altes fehlt, aktuelles eingeben? Hier sollten die Begriffe gleich sein. Fehler: <u>Neue E-Mail Adresse</u> fehlt. Bitte geben Sie den <i>vorderen Teil</i> der gewünschten Adresse ein. Sollte im Zusammenhang mit der Berichtigung der „vorderen Teil“ Formulierung korrigiert werden. Fehler: Größenangabe fehlt. Bitte geben Sie <u>eine Zahl ein</u>. Besser: ...die neue Größe der Mailbox an.
Unzureichender Lösungsvorschlag bei dem Fehler: Kein Mailkonto ausgewählt. Bitte wählen Sie ein Mailkonto aus.	Es sollte verständlich gemacht werden, dass E-Mail Adressen immer einem Mailkonto hinzugefügt werden müssen.

Beim Umbenennen von Aliases ist es möglich falsche Domains einzugeben.	Dies kann wie andersorts schon gemacht, durch eine Auswahl möglicher Domains verhindert werden.
<i>Interaktion</i>	
Der Admin kann beim Ändern des Passworts die Funktionen <i>Generieren</i> und <i>Ändern</i> nicht eindeutig unterscheiden.	Hier wäre eine konsistente Nutzung von Buttons besser.
Der Weg zurück zu den Suchergebnissen ist nicht klar genug. Besonders nachdem eine Aktion auf einer Seite nach den Ergebnissen (Mailbox bearbeiten) ausgeführt wurde, kann der Zurück-Button des Browsers problematisch werden.	Evtl. Link <i>Suchergebnisse</i> in Menü einbauen oder Link auf den Nachfolgesseite anbieten.
<i>Erkennen und Orientieren</i>	
Die Abgrenzung der funktionellen Bereiche klappt optisch nicht vollständig. Besonders auf der <i>Mailbox bearbeiten</i> Seite ist nicht klar, ob der Button „Ändern“ bei den Nutzerdaten auch die Mailboxgröße ändern, da beides auf demselben Hintergrund liegen.	Hier sollte eine Rahmung Klarheit schaffen.
Begriffe wie Mailkonto, Hosting, Domain werden im Kundenbereich nicht erklärt.	Die Hinweise wäre eine gute Möglichkeit für Erklärungen.
Der Weg zur Bearbeitungsseite eines Mailkontos/einer Mailbox ist durch die ID nicht sofort erkennbar.	Hier könnte alternativ ein <i>Bearbeiten</i> Link angeboten werden.

Das ist eine typische Verteilung für Heuristische Evaluationen, da hierbei meist kleinere Usability Probleme gefunden werden. Dabei ergaben sich aber aufgrund der Rundung Mittelwerte, die die kontroverse Bewertung der einzelnen Gutachter nicht mehr widerspiegeln. So wurden neun der letztlich mit Stufe 2 eingeschätzten Probleme von den Gutachtern sowohl als kosmetische als auch als schwerwiegende (bis katastrophale) Probleme bewertet. Für einige dieser Probleme wurde zwar auch eine Lösung erarbeitet, dennoch sollten diese im Nutzertest nochmals beachtet werden.

Im Anhang D werden alle Änderungen aufgeführt, die bei der Umsetzung des Designs im Prototyp vorgenommen wurden. Es wurde versucht, etwa 70 % der Probleme zu lösen. Dabei wurden alle Probleme, die von den Experten als schwerwiegend bewertet wurden, bearbeitet sowie eine Reihe der leichten bis kosmetischen Probleme, je nach dem wie viel Aufwand die Lösung bedeutet hat.

4.3.4 Beurteilung der Untersuchungsmethode

Wie am Anfang des Abschnitts schon erwähnt, wurde die Evaluation über E-Mail Kontakt durchgeführt. Vorteil davon ist, dass die Prüfer einen zeitlichen Spielraum für die Bearbeitung der Untersuchung haben und nicht an einem bestimmten Termin und Ort gebunden sind. Der Untersuchungszeitraum erstreckte sich über 10 Tage (24.07.2006 – 02.08.2006), obwohl es auch in einem kürzeren Zeitraum machbar gewesen wäre, wenn die Evaluation an einem bestimmten Ort stattgefunden hätte. Der Zeitraum war großzügig gesetzt, da die Gut-

achter die Untersuchung in ihrer Freizeit vorgenommen haben. Der zeitliche Rahmen kann noch weiter eingegrenzt werden, wenn die Kommunikation zwischen Prüfern und Untersuchungsleiter reibungslos klappt und es die Zeit der Prüfer erlaubt.

Weiters bedeutet es für den Untersuchungsleiter einen Mehraufwand bei der Erstellung der Unterlagen, da auch der Teil schriftlich festgehalten werden muss, der sonst mündlich den Gutachtern erklärt wird. Dies führt zum nächsten Problem, dass Fragen zu Beschreibungen oder Anweisungen während der Untersuchung nicht unmittelbar beantwortet werden können, sondern eine E-Mail zum Kontakt notwendig ist. In diesem Fall wurde auch der Kontakt über einen Instant Messenger hergestellt, da dort zumindest eine Echtzeit-Kommunikation möglich ist.

Die Beschreibungen zu den Entwürfen waren, laut Aussagen der Prüfer, an einigen Stellen missverständlich und der Umgang während der Evaluation mit drei Dokumenten (Flashfilm, Beschreibungen und Heuristiken) am Bildschirm erschwerte die Konzentration auf die eigentliche Untersuchung. Dadurch konnte die geplante Bearbeitungsdauer von den Gutachtern nicht eingehalten werden, weil eine längere Einarbeitungsphase notwendig war und auch während des eigentlichen Findens der Probleme ein Mehraufwand (Wechsel zwischen den Dokumenten, Notieren der Probleme) eingeplant hätte werden müssen.

Abschließend ist diese Form der Evaluation als problematisch einzuschätzen und die herkömmliche Vorgehensweise ist dieser vorzuziehen, wenn es die Umstände erlauben. In diesem Fall war es nicht anders möglich gewesen. Besonders die Planung und Erstellung der Dokumente einer Fernevaluation bedarf einer erhöhten Aufmerksamkeit und muss für einen reibungsloseren Ablauf der Untersuchung noch stärker an diese Form angepasst.

5 Implementierung eines Prototyps der Webhosting Webanwendung

„Symfony is an open-source web framework written in PHP5. Based on the best practices of web development [...] symfony aims to speed up the creation and maintenance of web applications, and to replace the repetitive coding tasks by power, control and pleasure.“
[The Symfony Project 2006]

Wie schon zur Konzeption der Anwendung beschrieben, wird ein PHP Framework namens Symfony (in der Version 0.6.3) für die Entwicklung verwendet. Dieses Framework gibt eine Struktur für den Quellcode vor und setzt Mechanismen auf die Programmiersprache PHP auf, um die Entwicklung zu beschleunigen. Es ist im Rahmen dieser Diplomarbeit nicht möglich, umfangreich die Realisierung der Funktionalität des Prototyps (Tabelle 7) zu beschreiben, da dafür eine ausführlichere Beschreibung des Frameworks notwendig wäre. In diesem Kapitel wird deshalb versucht, anhand von Strukturen und Mitteln, die das Framework vorgibt, beispielhaft zu zeigen wie diese bei der Implementierung genutzt wurden.

Tabelle 7: Umgesetzte Funktionalität im Prototyp der Webanwendung

<i>Funktionalität Kundenbereich</i>	<i>Funktionalität Administratorbereich</i>
Authentifizierungssystem mittels Login und Rechtemanagement	
Menügenerierung	
Validierung der Nutzereingaben (inkl. Fehlerausgabe) und Statusrückgabe nach erfolgreicher Aktion	
Anzeigen von E-Mail-Konten mit zugehörigen E-Mail-Adressen	
Anlegen, Umbenennen und Löschen von E-Mail-Adressen	
Hinweissystem	
Statussystem über Kontoinformationen des Kunden	Aufgabensystem für den Teilbereich Domain
Ändern des Passworts für E-Mail-Konten	Ändern der Benutzerdaten für E-Mail-Konten
	Ändern der Größe von E-Mail-Konten
	Suche nach E-Mail-Konten über Benutzername und Konto-ID

5.1 Konzepte und Techniken

5.1.1 MVC (Model-View-Controller)

Symfony basiert auf der MVC Architektur, welche folgende Ebenen enthält:

- *Model* – repräsentiert die Informationen, mit denen die Anwendung arbeitet; seine Business Ebene, welche den Rohdaten Bedeutung gibt (Abschnitt 5.3).
- *View* – wandelt das Model in eine Darstellung passend für die Interaktion mit dem Benutzer um (Abschnitt 0).
- *Controller* – verarbeitet Ereignisse (Aktionen des Benutzers) und ruft Änderungen des Models oder Views hervor, wenn sie nötig sind (Abschnitt 5.4).

5.1.2 ORM (Object-Relational Mapping)

ORM ist eine Programmier Technik, welche die relationale Logik (Tabellen, Spalten, SQL-Abfragen) einer Datenbank auf die objektorientierte Logik (Klassen, Attribute, Methoden) der Anwendung abbildet und umgekehrt. Die Daten (Objekte) können somit persistent mit ihren Relationen in der Datenbank gespeichert werden, aber über objektorientierte Methoden darauf zugegriffen werden.

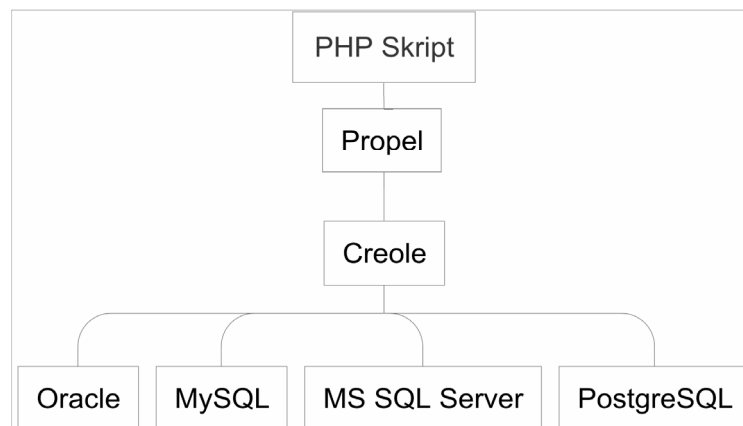


Abbildung 18: Ebenen des Object-Relational Mappings

In Symfony wurde die Open Source Lösung *Propel*¹³ zur Umsetzung des ORM integriert und stellt die *Model* Ebene der MVC Architektur dar. Propel generiert auf Grundlage einer projektspezifischen XML Datei die Objekte mit entsprechenden Klassen und SQL Datei. Durch die Integration von *Creole*¹⁴ besitzt diese Lösung eine herstellerunabhängige Datenbankschnittstelle. Creole abstrahiert die PHP eigene datenbankspezifische API, um einen portablen Code zu erzeugen. Abschnitt 5.3 beschreibt das Datenmodell der Webanwendung.

¹³ Nähere Informationen unter [The Propel Project 2006]

¹⁴ Nähere Informationen unter [The Creole Project 2006]

5.1.3 YAML (YAML Ain't Markup Language)

YAML ist eine Sprache zur Datenserialisierung, die sowohl für den Menschen lesbar als auch technisch leistungsfähig sein soll [vgl. YAML.org 2006]. In YAML wird die Annahme umgesetzt, dass jede beliebige Datenstruktur sich nur mit assoziativen Arrays, einfachen Arrays und Skalaren darstellen lässt und nutzt dabei die ursprünglichen Datenstrukturen der Sprache, in der es implementiert wird.

YAML wurde in Symfony integriert und das Format wird standardmäßig für alle notwendigen Konfigurationsdateien genutzt. Alternativ können aber auch XML oder INI Dateien verwendet werden. Wie für eine Konfiguration der Webanwendung dieses Format verwendet wurde, zeigt Abschnitt 5.6.

5.1.4 YAML (Yet Another Multicolumn Layout)

Das Akronym YAML bezeichnet in einer anderen Definition ein CSS Framework zur Erstellung moderner Layouts für Webseiten auf Grundlage von *float*-Umgebungen. Dabei werden DIV-Tags zur Erstellung mehrspaltiger Seiten anstatt von Tabellenstrukturen eingesetzt. Mit der *float*-Eigenschaft formatierte Elemente werden dabei von anderen HTML Elementen umflossen.

Ein Basislayout dieses Frameworks wurde für die Gestaltung der HTML Templates bei der Entwicklung der Webanwendung in angepasster Form verwendet (siehe Abschnitt 0).

5.1.5 AJAX (Asynchronous JavaScript and XML)

AJAX ist ein Konzept zur Verbesserung der Interaktivität von Webanwendungen, in dem Technologien wie XHTML/CSS, DOM/JavaScript, XML und XMLHttpRequest miteinander kombiniert werden. Durch AJAX können Webanwendungen nutzerfreundlicher gemacht werden, da Funktionalitäten aus Desktop Applikationen (z. B. Drag & Drop) auch im Web umgesetzt werden können und sich die Wartezeiten minimieren lassen.

Der Browser (Client) lädt die AJAX Engine (meist in JavaScript geschrieben), welche für die Generierung des User Interface und die Kommunikation mit dem Server verantwortlich ist. Anfragen des Nutzers werden als JavaScript Aufruf an die Engine geschickt und diese erzeugt die Antwort selbstständig, wenn keine Daten vom Server notwendig sind. Ansonsten findet eine asynchrone Kommunikation mit dem Server mittels XMLHttpRequest (HTTP-Anfragen werden vom Server mit XML Daten beantwortet) statt.

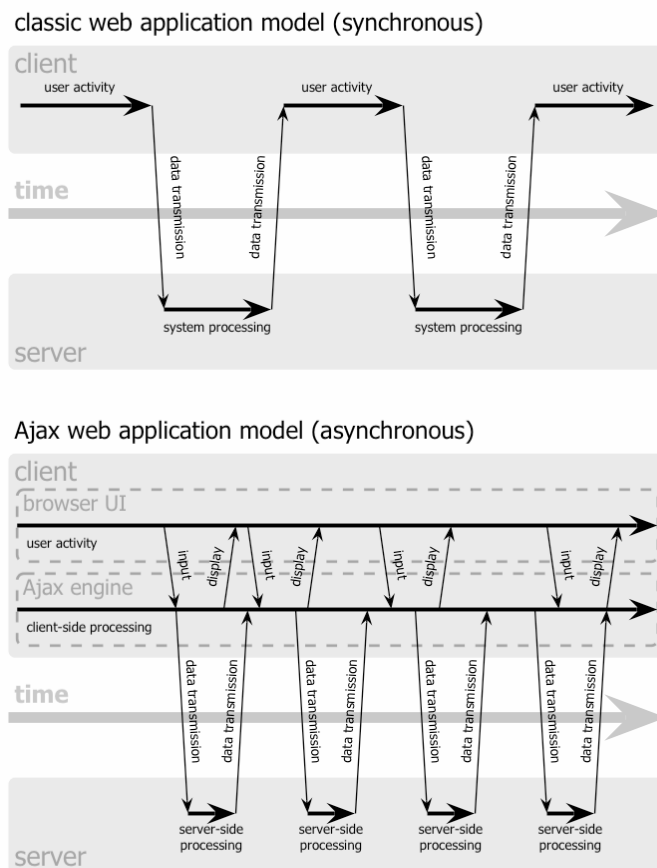


Abbildung 19: Synchrone Interaktion einer traditionellen Webanwendung im Vergleich mit dem asynchronen Muster einer AJAX Applikation (Quelle: [Garrett 2005])

In Symfony sind verschiedene Hilfsbibliotheken (Helpers) verfügbar, welche die Implementierung von AJAX Interaktionen in die eigene Webanwendung erleichtern (siehe Abschnitt 0).

5.2 Logische Struktur der Webanwendung als Symfony Projekt

Für die Verwendung von Symfony muss für die Webanwendung ein Projektverzeichnis angelegt werden, in dem die frameworkspezifischen Kommandos ausgeführt werden und die logische Struktur angelegt wird. Innerhalb eines Projekts werden Operationen logisch in Applikationen gruppiert. In diesem Fall wurden die Applikationen *customer* für den Kundenbereich und *backend* für den Administratorbereich angelegt, welche unabhängig voneinander lauffähig sind.

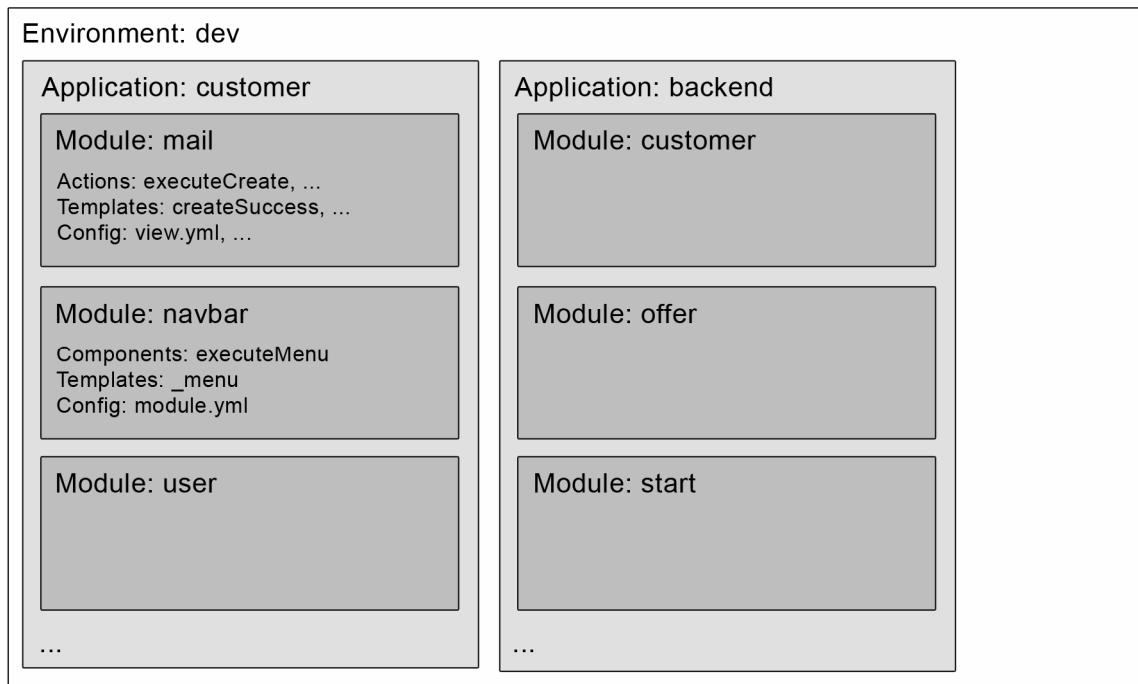


Abbildung 20: Projektstruktur der Webanwendung in der Entwicklungsumgebung

Jede Anwendung ist eine Menge aus ein oder mehreren Modulen, welche jeweils eine Funktion der Applikation umsetzen. Dazu gehören u. a. Konfigurationsdateien für die Sicherheit und die Anzeigeeigenschaften des Moduls, Konfigurationsdateien für die Validierung von Formulardaten aus Templates des Moduls und Klassen für Aktionen (s. u.). In der Abbildung 20 wurden exemplarisch drei Module für jede Applikation aufgeführt. Das Modul *mail* kapselt alle Funktionen, die mit den E-Mail-Konten zu tun haben (z. B. Anlegen einer E-Mail-Adresse).

Module enthalten Aktionen, welche die Geschäftslogik der Applikation enthalten. Im Quellcode sind Aktionen Methoden einer von der frameworkinternen Klasse *sfActions* abgeleiteten Klasse.

```
class mailActions extends sfActions
{
    public function executeCreate() // Aktion
    {
        // Anweisungen zum Erstellen einer E-Mail-Adresse wie
        // Verarbeitung der Nutzereingaben und Zugriff auf das Datenmodell zur
        // Speicherung der Eingaben und Setzen von Variablen für die
        // Präsentationsebene
    }
}
```

Quellcode 1: Klassendeklaration des Moduls *mail* mit der Methode für die Aktion *create*

Jede Applikation kann in verschiedenen Umgebungen laufen. Standardmäßig werden die Umgebungen Entwicklung (dev), Test (test) und Produktion (prod) angelegt. Diese können konfiguriert werden und es können weitere angelegt werden.

Um Projekte, Applikationen und Module anzulegen, wird der Befehl *symfony* mit einer entsprechenden Option auf der Kommandozeile im Projektverzeichnis ausgeführt. Dadurch werden entsprechend der Konfigurationsdatei *package.php* einige Verwaltungsaufgaben (z. B. Anlegen von Standarddateien) automatisiert.

```
$ symfony init-project wica
$ symfony init-app customer
$ symfony init-module customer mail
```

Quellcode 2: Anlegen des Projekts (wica), der Applikation für den Kundenbereich (customer) und des Moduls für die E-Mail-Konten (mail)

Dadurch wurde eine Verzeichnisstruktur angelegt, welche in Liste 7 in ihrem wichtigsten Punkten dargestellt ist.

```
wica/
  apps/
    customer/
      config/ (applikationsspezifische Konfigurationsdateien)
      [...] (weitere Ordner)
      lib/ (applikationsspezifische Klassen und Bibliotheken)
      modules/
        mail/
          [...]
      templates/ (globale Templates, in welche die Module hineingeladen
werden)
      [...]
      config/ (allgemeine YAML Konfigurationsdateien)
      data/ (Dateien wie XML für Datenbank-Schema oder SQL-Dateien)
      [...]
      lib/ (externe Klassen und Bibliotheken, die sich alle Applikationen
teilen)
      model/ (Objekt-Modell des Projekts)
      [...]
      web/ (Root-Verzeichnis des Projekts für den Webserver)
      [...]
```

Liste 7: Verzeichnisstruktur, die nach dem Ausführen der init-Befehle generiert wurde

Um die Applikation über den Browser erreichen zu können, muss das Verzeichnis *web* zugänglich gemacht werden, ebenso ein Datenverzeichnis des Symfony Frameworks. Dies wurde durch Änderungen in der Konfigurationsdatei *httpd.conf* des Apache Webserver bewirkt, indem dort ein Virtual Host für jenes Verzeichnis angelegt wurde und entsprechende Directory-Rechte vergeben wurden.

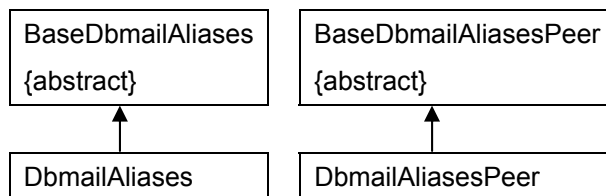
5.3 Datenmodell und Datenzugriff (Model)

Um das relationale Datenmodell in ein objektorientiertes Modell umzuwandeln, ist eine XML Datei erstellt worden, welche Tabellen, Relationen und Datentypen der verwendeten Datenbank definiert. Genauer gesagt, wurden sowohl für die bestehende Maildatenbank als auch für die Datenbank der Webanwendung solche Dateien erstellt (vgl. Abschnitt 4.1.5).

```
<?xml version="1.0" encoding="utf-8"?>
<database name="dbmail" defaultIdMethod="native">
  <table name="dbmail_aliases" idMethod="native">
    [...]
    <column name="alias_idnr" type="BIGINT" required="true"
      autoIncrement="true" primaryKey="true">
    </column>
    <column name="alias" type="VARCHAR" size="100"
      required="true" default="">
    </column>
    <column name="deliver_to" type="VARCHAR" size="250"
      required="true" default="">
    </column>
    <column name="client_idnr" type="BIGINT" required="true" default="0">
    </column>
    <index name="alias_index">
      <index-column name="alias"/>
    </index>
    <index name="client_idnr_index">
      <index-column name="client_idnr"/>
    </index>
  </table>
  [...]
</database>
```

Quellcode 3: Auszug aus der XML Schemadatei für die Maildatenbank (dbmail.schema.xml)

Für jede Tabelle werden vier Klassen mittels Propel generiert. Für die Tabelle *dbmail_aliases* aus dem obigen Beispiel ergeben sich folgende Klassen:



Die Base-Klassen werden in der Entwicklung nicht bearbeitet, da sie bei Veränderung des Datenbankschemas und Regenerierung der Klassen überschrieben werden. Erweiterungen der Methoden wurden in den abgeleiteten Klassen (*DbmailAliases*, *DbmailAliasesPeer*) definiert.

Beim Anlegen einer neuen E-Mail-Adresse in der Datenbank wird in der entsprechenden Controller Klasse (entspricht einer Aktion in der Frameworksemantik) beispielsweise ein *DbmailAliases* Objekt verwendet.

```
$this->alias = new DbmailAliases();
$this->alias->setAlias($name);
$this->alias->setDeliverTo($this->getRequestParameter('mailbox'));
$this->alias->setClientIdnr($this->getUser()->getSubscriberId());
$this->alias->save();
```

Quellcode 4: Anlegen einer neuen E-Mail-Adresse (Klasse *mailActions*; Methode *executeCreate()*)

Über die Methode *save* wird die Verbindung zur Datenbank hergestellt und die Insert-Anweisung als Transaktion gestartet.

Zum Finden von Feldern (Records) aus der Datenbank, werden Methoden der Peer-Klassen verwendet. Beispielsweise wurde für die Abfrage aller E-Mail-Adressen eines E-Mail-Kontos die *DbmailAliasesPeer* Klasse um folgende Methode erweitert:

```
public static function getAliasesByUser($userid)
{
    $c = new Criteria();
    $c->add(self::DELIVER_TO, $userid);

    return self::doSelect($c);
}
```

Quellcode 5: Methode zum Selektieren aller E-Mail-Adressen eines E-Mail-Kontos (Parameter *userid*)

Über Criteria-Objekte kann die Select-Anweisung spezifiziert werden. In diesem Fall entspricht es einer Where-Klausel. Die Rückgabe ist ein Array von *DbmailAliases* Objekten.

```
$this->aliases = DbmailAliasesPeer::getAliasesByUser($this->mailbox_id);
```

Quellcode 6: Verwendung der Methode zum Selektieren der E-Mail-Adressen per Klassenaufruf (Peer-Klasse)

Somit konnte die Datenlogik entsprechend dem MVC Konzept auf einer Ebene (Model) zusammengefasst werden und Änderungen daran können unabhängig vorgenommen werden.

5.4 Aktionen und Komponenten (Controller)

5.4.1 Aktionen

Aktionen bilden Klassen, welche die Geschäftslogik der Webanwendung umsetzen. Das Modul *mail* beispielsweise vereint die Funktionen, die zur Ansicht und Bearbeitung der E-Mail-Konten und E-Mail-Adressen notwendig sind. Dafür ergibt sich folgende Klasse (in der Applikation *customer* für den Kundenbereich):

mailActions	
+ executeIndex ()	
+ executeList ()	// Auflistung aller E-Mail-Konten eines Kunden
+ executeShow ()	// Anzeige aller E-Mail-Adressen eines E-Mail-Kontos
+ executeCreate ()	// Anlegen einer E-Mail-Adresse
+ handleErrorCreate ()	// Fehlerbehandlung beim Anlegen einer E-Mail-Adresse
+ executeUpdateindex ()	// Anzeige der Eigenschaften eines E-Mail-Kontos zur Bearbeitung
+ executeUpdate ()	// Ändern einer E-Mail-Adresse
+ handleErrorUpdate ()	// Fehlerbehandlung beim Ändern einer E-Mail-Adresse
+ executeDelete ()	// Löschen einer E-Mail-Adresse
+ executePwupdate ()	// Ändern des Passworts eines E-Mail-Kontos
+ handleErrorPwupdate ()	// Fehlerbehandlung beim Ändern des Passworts

In einem weiteren Schritt des Refactoring des Quellcodes könnten die Funktionen, welche nur E-Mail-Adressen betreffen in ein eigenes Modul ausgelagert werden. Während der Prototyp-erstellung war dies aber noch nicht erforderlich.

Aktionen entsprechen meist bestimmten Seiten der Webanwendung. Zum Beispiel werden in der Methode *executeUpdateindex* Daten zusammengestellt, die – im entsprechenden Template – zur Anzeige der Bearbeitungsseite verwendet werden. Andererseits können Aktionen auch nur zur Verarbeitung von Formulardaten dienen in dem Sinne nur Teilbereiche einer Seite betreffen. Die Methode *executeCreate* wertet z. B. die Eingaben des Nutzers zum Erstellen einer E-Mail-Adresse aus, ruft das entsprechende Datenmodell-Objekt zur Speicherung der E-Mail-Adresse auf und stellt notwendige Daten zur erneuten Anzeige des Formulars zusammen. Für die Formularverarbeitung gibt es meist noch eine zweite Methode, die in Kraft tritt, wenn über die Konfiguration die Validierung des entsprechenden Formulars festgelegt wurde und Eingabefehler aufgetreten sind (*handleErrorCreate*). Inhaltlich ähnelt die Error-Handle Methode dem Original, nur das keine Veränderung in den Daten (Datenbankzugriff) vorgenommen wird.

Der Rückgabewert jeder Methode ist der gewünschte View (Attribut der frameworkinternen *sfView* Klasse), welcher einem Template (Datei) entspricht. Meist wurde hier der Standardwert *SUCCESS* verwendet. So ergibt sich für die Aktion *create* (*executeCreate*) der Aufruf des Templates *createSuccess.php*. In anderen Fällen wurde ein spezielles Template über die Aktion angefordert wie das nachfolgende Beispiel zeigt.

```
public function executeList()
{
    [...]
    $this->setTemplate('index');
    return sfView::SUCCESS;
}
```

Quellcode 7: Die Aktion *list* verwendet das Template *indexSuccess.php* auf der Präsentationsebene

Aktionen werden ausgeführt, wenn sie

- direkt über die URL aufgerufen werden
Beispiel: <http://wica.netfarm.at/mail/index> ruft die Aktion *index* im Modul *mail* auf
- oder als Parameter einer Funktion zur Erzeugung von HTML-Elementen dienen

```
echo form_remote_tag(array(
    'update' => array('success' => 'whitebox'),
    'url'     => 'mail/pwupdate',
))
```

Quellcode 8: Funktion zum Erzeugen eines AJAX-gestützten Formulars, dessen Auswertung die Aktion *pwupdate* des Moduls *mail* übernimmt

5.4.2 Komponenten

Das Konzept der Komponenten (Component Slots) des Frameworks wurde für Anzeigebereiche eingesetzt, die auf allen Seiten der Webanwendung eingesetzt werden, aber inhaltlich seitenspezifisch geändert werden. Das Menü allgemein und das Hinweissystem im Kundenbereich sind Beispiele dafür. Komponenten sind wie Aktionen Teil eines Moduls und bilden eine Klasse. Methoden in Komponenten können im Gegensatz zu Aktionen keine unterschiedlichen Views aufrufen; eine Validierung mittels Komponenten ist auch nicht möglich. Für das Menü und die Hinweise waren diese fehlenden Möglichkeiten aber nicht von Nachteil.

```
class navbarComponents extends sfComponents
{
    public function executeMenu()
    {
        // Generierung des Hauptmenüs
    }
}
```

Quellcode 9: Klasse für die Komponente *navbar* mit Methode *executeMenu* zur Menügenerierung

Um das Menü variable zu gestalten wurden die Einträge sowie die Struktur aus einer Konfigurationsdatei bezogen, welche im Abschnitt 5.6. näher erläutert wird. Eine Komponente notwendig machte die Tatsache, dass für die aktuelle Seite der Menüpunkt anders dargestellt werden sollte (fett und ohne Verlinkung). Während es eine Aufgabe der View-Ebene ist, wie das Menü dargestellt wird, musste auf der Controller-Ebene in der Komponentenklasse das Auslesen der Konfigurationsdatei und die Feststellung der aktuellen Seite (welche Aktion welches Moduls) passieren.

Verwendet werden Komponenten, indem sie in der Konfiguration der Applikation oder der einzelnen Module festgelegt werden. Die Menü-Komponente wurde applikationsweit konfiguriert, da nur eine Methode (*executeMenu*) benötigt wird, um den Inhalt dynamisch anzupassen.

Das Hinweissystem wurde auch als Komponente umgesetzt. Da nur statische Templates (in diesem Prototyp zumindest noch) passend zum Kontext aufgerufen werden, sind die Methoden der Komponentenklasse *adviceComponents* leer, bewirken aber den Aufruf des dazugehörigen Templates. Eingesetzt wurde die Komponente entsprechend der Konfiguration in den Modulen, da hier eine applikationsweite Konfiguration nicht sinnvoll war.

```
default: # gilt für alle Templates des Moduls
  components:
    advice: [advice, mailindex] # Methode executeMailindex verwendet

updateindexSuccess: # überschreibt default für Template updateindexSuccess
  components:
    advice: [advice, mailupdate] # Methode executeMailupdate verwendet
```

Quellcode 10: Konfiguration der Hinweiskomponente im Modul *mail* (view.yml)

5.5 Layout, Templates und AJAX (View)

Als Ergebnis einer Aktion (oder Komponente) wird ein Template zur Präsentation und Interaktion mit dem Nutzer aufgerufen (View-Ebene). Ein Template ist eine Datei, die HTML Code und Variablen aus den Aktionen verwendet. Templates werden in ein applikationsweites (Applikation im Sinne des Frameworks) Layout eingebunden, welches unter Verwendung von Konfigurationsdateien für die Einbindung von Meta-Tags, Stylesheets, Javascripts und Komponenten zur HTML Seite generiert wird, die im Browser angezeigt wird.

Die Abbildung 21 zeigt schematisch den Aufbau eines Layouts aus der Applikation des Kundenbereichs und die Einbindung von Konfiguration und Templates.

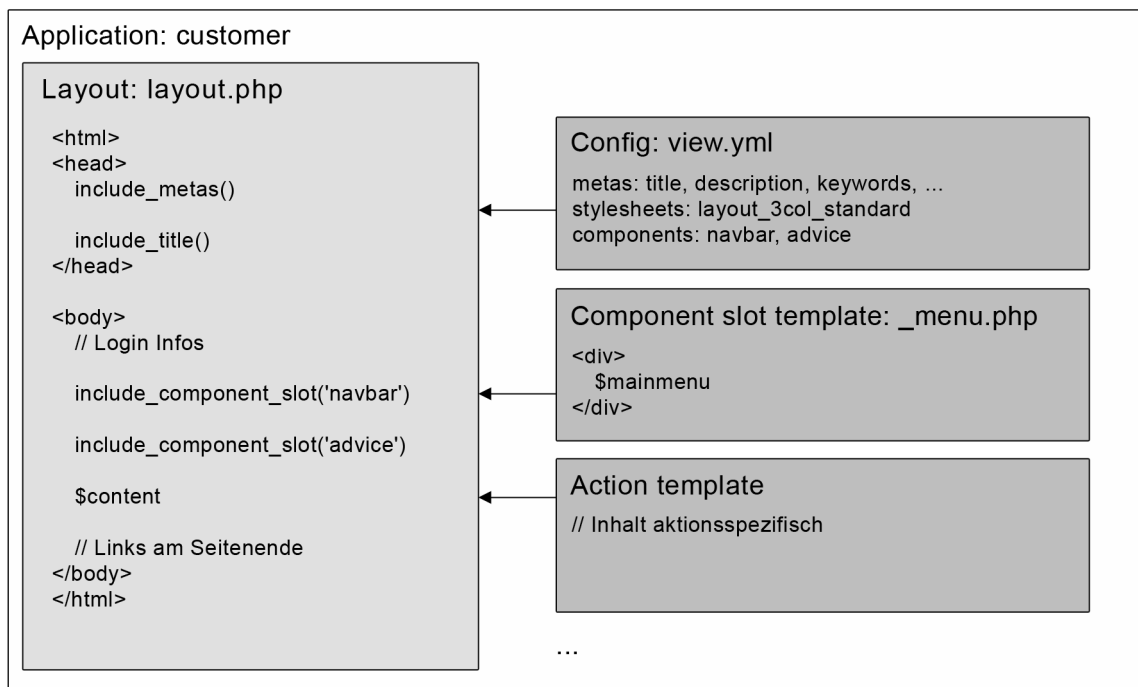


Abbildung 21: Vereinfachte Darstellung des Zusammenhangs der Elemente des Views für den Kundenbereich

5.5.1 Layout

Für jede Applikation wurden zwei Layouts (*layout.php* und *layout_login.php*) erstellt, da für die Loginseite zur Webanwendung weniger Struktur (keine Spalten) notwendig war und Elemente wie Menü und Hinweise erst nach der Anmeldung dem Nutzer zur Verfügung stehen dürfen. Die HTML Struktur des Hauptlayouts (*layout.php*) wurde in Anlehnung an ein Basislayout des YAML Frameworks entwickelt. Dabei wird eine verschachtelte Struktur aus DIV-Tags aufgebaut, welche die Teilbereiche der Seite logisch (nicht visuell) gliedert. Die Anordnung der DIV-Bereiche wird ausschließlich über CSS Eigenschaften umgesetzt, indem jedes DIV-Tag eine eindeutige ID erhält. So wurden die vier Hauptbereiche *header* (Logo und Logininformationen), *nav* (Hauptnavigation), *main* (Inhalt und Hinweise) und *footer* (Nebenavigation) geschaffen. Da der *main* Bereich aus zwei Spalten bestehen soll, wurde er nochmals in DIV-Tags für den Inhalt (linke Spalte) und die Hinweise (rechte Spalte) unterteilt.

Die Gestaltung der Präsentation wurde ausschließlich über CSS Dateien vorgenommen. Eine Trennung von Struktur und Design ermöglicht eine einfache Veränderung des Aussehens der Seite indem wenige Dateien (Stylesheets) geändert werden. Die CSS Anweisungen wurden nach Detailgrad in verschiedene Dateien aufgeteilt (z. B. gibt es eine CSS Datei, in der alle Anweisungen für die Darstellung des Hauptmenüs stehen). Zur Einbindung in das Layout wird schließlich nur eine CSS Datei benötigt, in die alle anderen importiert werden.

5.5.2 Templates und AJAX

Templates bilden die Präsentationsebene der Webanwendung, also die Oberfläche, die Nutzer sehen und bedienen. Wie diese organisiert wurden, soll das Beispiel der Seite zum Bearbeiten eines E-Mail-Kontos im Kundenbereich zeigen (Abbildung 22).

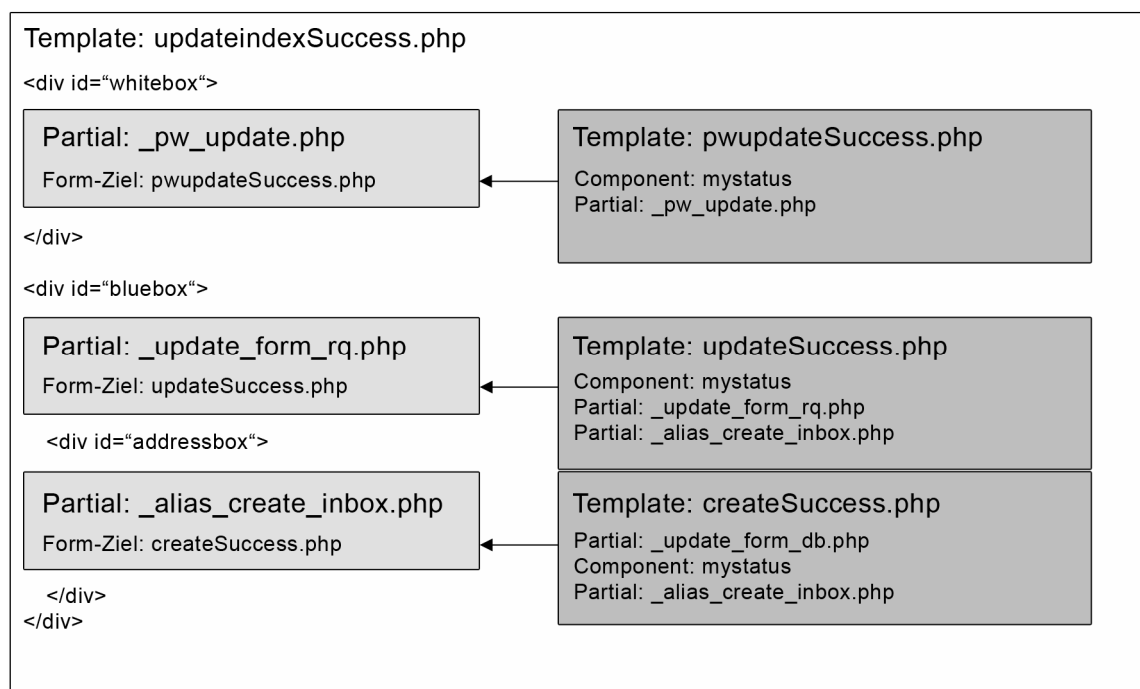


Abbildung 22: Schematische Darstellung der Template Organisation für die Seite zum Bearbeiten eines E-Mail-Kontos im Kundenbereich (`updateindexSuccess.php`)

Das Template *updateindexSuccess* bildet die Grundstruktur der Seite, indem es DIV-Tags definiert, in welche Partials (Templates ohne Bezug zur einer Aktion) geladen werden. Diese kapseln meist ein Formular, welches mittels AJAX einen asynchronen Request auslöst, damit nur der Inhalt des entsprechenden DIV-Tags aktualisiert wird. Da die Formulare durch Aktionen ausgewertet werden, liefern die dazugehörigen Partials den neuen Inhalt der DIV-Tags (siehe Abbildung 22 rechte Seite).

Nach ähnlichem Muster sind auch andere Seiten aufgebaut. Ausnahme bildet die Suchseite für E-Mail-Konten im Administratorbereich, welche mittels eines sog. Admingenerators des Frameworks erstellt wurde. Mit dem Admingenerator können Suchmasken (Filter) schnell erstellt und die Suchergebnisse leicht aufbereitet werden (Blätterfunktion, Sortierfunktion). Danach wurden die generierten Templates und Aktionen an die Bedürfnisse der Webanwendung angepasst.

5.6 Konfiguration des Projekts

Konfiguration kann in Symfony auf der Ebene des Projekts, der einzelnen Applikationen und der Module vorgenommen werden. Außerdem kann sie für die einzelnen Umgebungen unterschiedlich gestaltet werden. Die Dateien sind in YAML Syntax formatiert.

5.6.1 Projektweite Konfiguration – Datenbankverbindung

Die Konfigurationsdatei *databases.yml* enthält die Informationen zum Herstellen der Datenbankverbindung (Datenbanktyp, Host, Datenbankname, Login) zur Maildatenbank *dbmail* und der Webanwendungsdatenbank *wica*. Diese werden bei der Generierung der Datenmodell Klassen und deren Objekte für Abfragen an die Datenbank verwendet.

```
all:
  dbmail:
    class:      sfPropelDatabase
    param:
      phptype:  mysql
      host:     localhost
      database: dbmail-2-0-10
      username: dbmail
      password: dbmail
```

Quellcode 11: Auszug aus der Konfigurationsdatei *databases.yml*

5.6.2 Applikationsweite Konfiguration – Menü und Routing

Die Klasse, welche das Menü generiert, greift auf die Informationen der Konfigurationsdatei *app.yml* zurück. Die Methode ist in der Kunden- und Administratoranwendung dieselbe, nur die Konfigurationsdateien mussten die unterschiedlichen Menüpunkte enthalten.

```
all:
  .settings:
    categories:
      'Start':      '@start'
      'Meine Daten': '@daten'
      'Mein Konto': '@konto'
      'Domain':     '@domain'
      'E-Mail':     '@mail'
      'Datenbank':  '@datenbank'
    [...]
  mail:
    'Übersicht':      '@mail'
    'E-Mail-Konto erstellen': '@mailkonto_erstellen'
    'E-Mail-Konto bearbeiten': '@mailkonto_bearbeiten'
    'Webmail':        '@webmail'
```

Quellcode 12: Auszug aus der Konfigurationsdatei mit den Menüeinträgen des Kundenbereichs (*app.yml*)

Im Ausschnitt der Konfigurationsdatei (Quellcode 12) ist zu sehen, dass unter *categories* die Haupteinträge des Menüs zusammengefasst wurden. Dabei wird der Name vor dem Doppel-

punkt (z. B. E-Mail) zur Bezeichnung des Links verwendet; der Wert (nach dem Doppelpunkt) ist der Name der Route, die angibt, welche Aktion in welchen Modul damit verbunden ist (z. B. @mail). Für das Untermenü wurden für jede Kategorie weitere Parameter angelegt (*mail* in Quellcode 12).

In einer anderen Konfigurationsdatei (*routing.yml*) wird die Zuordnung der Routen verwaltet.

```
mail:
  url:    /mail/index
  param: { module: mail, action: index }
```

Quellcode 13: Auszug aus der Routing Konfiguration des Kundenbereichs (*routing.yml*)

Das Routingsystem wird hier etwas abseits seiner eigentlichen Funktion zur Vereinfachung von URLs (SmartURLs) gebraucht, da die Route lediglich notwendig ist, um das Ziel des Links im Menü festzulegen. Für die Kategorie *E-Mail* im Kundenbereich bedeutet dies, dass der Link die Aktion *index* des Moduls *mail* aufruft. Allgemein wird die Routing Konfiguration bei jedem Request ausgewertet und die angeforderte URL mit der des Parameters *url* verglichen.

5.6.3 Modulweite Konfiguration – Validierung und Sicherheit

Die Überprüfung von Nutzereingaben wird in jedem Modul durch aktionsspezifische Konfigurationsdateien ausgelöst. Als Beispiel für eine solche Konfiguration soll hier die Überprüfung der Passwortänderung für ein E-Mail-Konto im Kundenbereich dienen (Quellcode 14).

```
[...]
names:
  old_pw:
    required:      Yes
    required_msg:  [...]
    validators:    existValidator
[...]

existValidator:
  class:          myUniqueValidator
  param:
    subject:      mailpw
    mailbox_id:   mailbox_id

pw2Validator:
  class:          sfCompareValidator
  param:
    check:        new_pw
    compare_error: [...]
```

Quellcode 14: Auszug aus der Konfiguration zur Validierung der Nutzereingaben zur Passwortänderung im Kundenbereich (*pwupdate.yml*)

Zuerst wird überprüft, ob Pflicht-Eingabefeldern (*required: Yes*) ausgefüllt wurden (im Beispiel das Feld *old_pw* für das aktuelle Passwort). Falls dies nicht der Fall ist, erscheint die

Fehlermeldung unter *required_msg*. Zusätzlich können weitere *Validators* angegeben werden, die Eingaben überprüfen. So wurde die globale Klasse *myUniqueValidator* geschrieben, die aufgrund verschiedener Parameter Eingaben mit Daten aus der Datenbank vergleicht und sie auf Eindeutigkeit überprüft. In diesem Fall wurde die Klasse genutzt, um das eingegebene aktuelle Passwort mit dem in der Datenbank gespeicherten Passwort zu vergleichen. Nützlich ist diese Klasse auch, wenn gewährleistet werden muss, dass E-Mail-Adressen eindeutig angelegt oder umbenannt werden.

Das Framework selbst bietet für häufig verwendete Vergleiche interne Klassen, die parametrisiert genutzt wurden wie in diesem Beispiel der *sfCompareValidator*, der zwei Strings auf Gleichheit überprüft (neues Passwort und Wiederholung des neuen Passworts).

Die Validierung wurde bei allen Eingabemöglichkeiten des Prototyps im Kunden- und Administratorbereich eingesetzt, um Fehlermeldungen auszugeben, wie sie in der Konzeption festgelegt wurden.

5.6.4 Sicherheit

Eine weitere Funktion des Prototyps ist ein Authentifizierungssystem. Nutzer der einzelnen Bereiche müssen sich mit einem Login (Benutzername, Passwort) anmelden, bevor sie die Funktionen der Webanwendung nutzen können. Dennoch ist es möglich, dass geschützte Seiten mittels direkter URL aufgerufen werden. Das kann zu Fehlermeldungen führen, da nutzerspezifische Daten fehlen um z. B. Datenbankabfragen erfolgreich durchzuführen.

Es wurde ein Rechtemanagement konzipiert, welches in der Implementierung in Form von *Credentials* genutzt wird. Im Prototyp haben Kunden alle das Credential *subscriber*, das Administratoren Credential ist *admin*. In der weiteren Entwicklung werden sich die Kunden Credentials je nach bestellten Leistungen unterscheiden, weil damit verbunden auch unterschiedliche Funktionen zugänglich sein werden (z. B. Datenbankzugang).

Durch eine modulspezifische Konfiguration ist es möglich zu überprüfen, ob der angemeldete Nutzer das entsprechende Credential besitzt (als Session Attribut gespeichert), um spezifische Aktionen des Moduls auszuführen (z. B. Anlegen von E-Mail-Adressen). Da im Prototyp noch jeder Kunde dieselbe Berechtigung hat, ist keine Unterteilung für einzelne Aktionen notwendig.

```
all: # alle Aktionen des Moduls sichern
  is_secure: on
  credentials: subscriber
```

Quellcode 15: Konfiguration zur Sicherheit eines Moduls im Kundenbereich vor unberechtigtem Zugriff (security.yml)

Die in Quellcode 15 dargestellte Konfiguration bewirkt, dass z. B. die Seite zum Anzeigen der E-Mail-Konten eines Kunden bei unbefugtem Aufruf nur einen Hinweis ausgibt, dass ein Login notwendig ist, um diese Seite anzuzeigen.

6 Nutzertests am Prototyp der Webhosting Webanwendung

6.1 Planung der Nutzertests

Im Anhang D sind die Ergebnisse der Heuristischen Evaluation aufgelistet und die tatsächlich bei der Erstellung des Prototyps gemachten Änderungen zusammengestellt. Ziel der Nutzertests war es, einerseits diese Änderungen auf ihre Wirksamkeit bezüglich Usability zu überprüfen, andererseits bisher unbekannte Usability Probleme aufzudecken. Dabei wurden die Tests als Discount Maßnahmen geplant, da aufgrund der geringen Teilnehmerzahl und der zur Verfügung stehenden Mittel keine rein wissenschaftliche Vorgehensweise möglich war. Gegenstand der Untersuchung war der in Kapitel 4 und 5 vorgestellte Prototyp (vgl. Tabelle 7, S. 72 und Anhang E).

Als Methode für die Tests wurde das *laute Denken* (vgl. Abschnitt 3.3.1) in der Variation der *aktiven Intervention* verwendet. Dabei ist es dem Untersuchungsleiter gestattet die geplanten Aufgaben durch geeignete Zwischenfragen zu ergänzen. Grund für diese Wahl war, dass der Prototyp nur begrenzte Aufgabenstellungen zulässt. Selbst die gestellten Aufgaben konnte der Nutzer versuchen in Bereichen zu lösen, die nicht funktionstüchtig waren. Außerdem sollte es den Teilnehmer unterstützen, da das laute Denken kein intuitives Verhalten ist und keine Vorübung dafür geplant wurde.

6.1.1 Testnutzer und Testbedingungen

Aufgrund der unterschiedlichen Charakteristik der Zielgruppen für beide Anwendungsbereiche, wurden zwei getrennte Nutzertests durchgeführt. Für jeden Test wurde eine repräsentative Gruppe aus vier Personen rekrutiert. Die Teilnehmer für den Test des Kundenbereichs sind bestehende Kunden des Anbieters, denen in Zukunft diese Webanwendung zur Verfügung gestellt werden soll. Die Gruppe der Testnutzer für den Administratorbereich setzte sich aus derzeitigen Administratoren des Anbieters und befreundeten Kollegen, die ähnliche Services anbieten, zusammen. Nachfolgend werden die Testnutzer des Kundenbereichs als *Kunden* bezeichnet; die Teilnehmer am Test des Administratorbereichs als *Administratoren*, damit keine Verwechslung zwischen den beiden Testgruppen auftritt.

Die Tests mit den Kunden wurden bei den Kunden vor Ort auf einem Laptop durchgeführt. Der Laptop wurde jeweils mitgebracht und war mit einem 13 Zoll Bildschirm, dem Betriebssystem Windows XP und dem Browser Firefox (Version 1.5 zum Zeitpunkt des Tests) ausgestattet. Die Administratoren testeten ihren Bereich in einem Arbeitszimmer des Anbieters an einem PC mit einem 19 Zoll Monitor, dem Betriebssystem Windows XP und ebenfalls im Browser Firefox. In beiden Fällen gab es einen Arbeitsplatz für den Teilnehmer und daneben einen Sitzplatz für den Untersuchungsleiter, damit dieser den Bildschirm beobachten konnte. Die Sitzung wurde ab der Bearbeitung der Aufgaben auf ein Diktiergerät aufgenommen, um sich besser auf die Beobachtung und Zwischenfragen konzentrieren zu können. Dazu wurde eine Einverständniserklärung der Teilnehmer zu Beginn des Tests eingeholt. Als Dauer des gesamten Tests wurden 30 Minuten eingeplant.

6.1.2 Aufgaben

Da es sich um einen Prototyp mit sehr eingeschränkter Funktionalität handelt, beziehen sich nur einige der Aufgaben auf die konkrete Nutzung des Prototyps zum Erreichen eines Ziels. Dieser Teil der Aufgaben wurde möglichst realistisch geplant. Dabei sollten die Aufgaben nicht zu trivial werden, so dass hier nur eine begrenzte Anzahl solcher aufgestellt wurde. Darüber hinaus wurden Fragen nach Erwartungen und voraussichtlichen Handlungen des Nutzers formuliert. Es wurde sich für diese Mischung aus Beobachtung und Gespräch mit dem Kunden entschieden, da keine konkrete Nutzer- und Aufgabenanalyse im Vorfeld möglich war und sich die Konzeption der Anwendung nur auf die Aussagen des Auftraggebers stützen konnte. Außerdem konnte selbst die Durchführung der Aufgaben die Nutzer in Bereiche führen, die in diesem Prototyp noch nicht funktionstüchtig waren. Dann musste der Versuchsleiter eingreifen und den Nutzer nach seinen Absichten befragen. Die Tabelle 8 listet die Aufgaben für die Kunden mit dem dazugehörigen Zweck auf, den die Aufgabe im Test erfüllen sollte. In Tabelle 9 sind die Aufgaben für die Administratoren zu finden.

Tabelle 8: Aufgaben des Nutzertests mit der Testgruppe Kunden und dessen Zweck

Nr.	Formulierung der Aufgabe	Zweck im Test
1	Melden Sie sich mit dem Benutzernamen <i>[kundenspezifisch]</i> und dem Passwort <i>NeuPW!</i> an.	Leichte Einstiegsaufgabe, die dem Kunden das Gefühl geben sollte, dass er den Test bewältigen kann. Sie sollte helfen aufzudecken, ob es Probleme mit dem Verständnis des Anmeldedialogs gibt und ob im Fehlerfall die Fehlermeldung hilfreich ist.
2	Schauen Sie sich alles an und identifizieren sie folgende Bereiche: - das Logo - den Namen der Seite - die Kategorien der Seite - ein Merkmal, das Ihnen zeigt, auf welcher Seite Sie sich gerade befinden	Es sollte damit festgestellt werden, ob die Struktur der Webanwendung erfasst wird und ob sich der Nutzer leicht orientieren kann. Wird die Seitenüberschrift wahrgenommen? Ist das System der Karteikartenreiter verständlich umgesetzt?
3	Was erwarten Sie von den Kategorien?	Es sollte geprüft werden, ob die Beschriftung der Kategorien (Verständnis der Begriffe) ihren Inhalt passend widerspiegelt und ob die Einteilung mit dem Nutzererwartungen übereinstimmt.
4	Was wäre die erste Aufgabe oder regelmäßige Aufgaben, die Sie über diesen Verwaltungsbereich erledigen möchten? Wo würden Sie klicken, um diese Aufgabe zu erledigen? Was erwarten Sie nach dem Klick vorzufinden? (Falls diese Aufgabe schon real erledigt werden konnte im Prototyp, wurde der Kunde gebeten, die Aufgabe tatsächlich durchzuführen.)	Die Aufgabe sollte testen, ob die Direktlink Funktionen vom Kunden erkannt werden bzw. welcher Navigationsweg typischerweise vom Kunden gewählt wird. Die Erwartungen des Kunden sollen helfen bei der Weiterentwicklung der Anwendung noch benutzerge-rechter zu arbeiten.

5	Legen Sie eine E-Mail-Adresse Ihrer Wahl an!	Eine realistische Aufgabe. Wird das Hinweissystem genutzt, wenn Probleme auftreten? Sind die Eingabedialoge verständlich? Unterstützen Status- und Fehlermeldungen den Kunden beim Erreichen des Ziels?
6	Löschen Sie die eben angelegte E-Mail-Adresse!	Hier ging es darum, ob der Kunde die Links zum Einblenden von E-Mail-Adressen nutzt oder auf welchen Weg er nach einer E-Mail-Adresse sucht. Ist die Statusmeldung ausreichend?
7	Sie möchten eine weitere Domain bei Netfarm hosten. Wo würden Sie klicken, um dies zu erledigen? Was erwarten Sie nach dem Klick vorzufinden?	Test, ob der Kunde die Begriffe Domain und Hosting versteht und welche Vorstellungen er über den Vorgang hat. Findet er sich in der Navigationsstruktur zurecht?
8	Sie möchten weitere Leistungen bei Netfarm bestellen. Wo würden Sie dafür klicken? Was erwarten Sie nach dem Klick vorzufinden?	Test, welche Vorstellungen der Kunde von dem Bestellvorgang von weiteren Leistungen hat.
9	Melden Sie sich aus dem System ab!	Leichte Abschlussaufgabe, um dem Kunden das Gefühl zu geben, dass er etwas erreicht hat. Testen, ob Abmeldefunktion gefunden wird.

Tabelle 9: Aufgaben des Nutzertests mit der Testgruppe Administratoren und dessen Zweck

Nr.	Formulierung der Aufgabe	Zweck im Test
1	Melden Sie sich mit dem Benutzernamen <i>admin</i> und dem Passwort <i>D4sg3ht!</i> an. Verschaffen Sie sich einen Überblick, aber klicken Sie noch nichts an! Was sehen Sie alles?	Das Anmelden sollte eine einfache Einstiegsaufgabe sein. Danach sollte getestet werden, ob die Struktur der Seite erkannt wird und der Inhalt der Startseite verständlich ist. Erkennt der Administrator, was auf der Startseite getan werden kann?
2	Was wäre die erste Aufgabe oder regelmäßige Aufgaben, die Sie über diesen Verwaltungsbereich erledigen möchten? Wo würden Sie klicken, um diese Aufgabe zu erledigen?	Die Aufgabe sollte testen, ob die Direktlink Funktionen vom Administrator erkannt werden bzw. welcher Navigationsweg typischerweise vom Administrator gewählt wird.
3	Der Kunde Max Mustermann möchte einen anderen Loginnamen für die Mailbox <i>maxmust</i> . Ändern Sie den Loginnamen der Mailbox zu <i>maximilian</i> !	Es soll die Suchfunktion des Mailboxbereichs getestet werden. Sind die Eingabedialoge verständlich? Sind die Suchergebnisse verständlich aufbereitet? Helfen die Fehlermeldungen das Ziel zu erreichen? Außerdem sollte der Dialog zum Bearbeiten des Loginnamens getestet werden. Hier waren in der Heuristischen Evaluation Probleme genannt worden aufgrund dessen das Design verändert wurde.
4	Der Kunde Max Mustermann bittet Sie, die E-Mail-Adresse <i>gut@meine-seite.at</i> zu löschen, da er derzeit diese Aufgabe nicht selbst erledigen kann. Erfüllen Sie ihm seinen Wunsch!	Die Rückkehr zu den Suchergebnissen sollte getestet werden. Wie kehrt er zu den Suchergebnissen zurück? Wird eine neue Suche gestartet? Außerdem sollte der Dialog zum Löschen der E-Mail-Adresse getestet werden. Geben die Statusmeldungen eine ausreichende Systemantwort?

5	Die Domain <i>kovac.at</i> wurde erfolgreich registriert. Verwenden Sie das System, um dem Kunden sichtbar zu machen, dass die Registrierung abgeschlossen wurde.	Hier ging es darum, das Aufgabenverwaltungssystem der Startseite zu testen. Würde der Administrator sich an die Funktion auf der Startseite erinnern und diese Nutzen oder wie würde er ansonsten das Ziel erreichen wollen? Auch hier waren auch wieder die Statusmeldungen zu beurteilen.
6	Loggen Sie sich aus dem Administrationsbereich aus!	Einfache Abschlussaufgabe, um den Administrator mit einem guten Gefühl den Test abschließen zu lassen. Ist der Logout Link leicht zu finden?

6.1.3 Interviewfragen

Vor und nach den beiden Tests wurde ein kurzes Interview geplant. Das vorherige Interview (Liste 8) diente dazu herauszufinden, wie viel Erfahrung die Nutzer im Umgang mit dem Internet haben (Fragen 1 bis 4) und ob sie im Internet schon andere Anwendungen zum Verwalten nutzen oder genutzt haben (Fragen 6 und 7). Die Frage nach der Lieblingswebseite war zur Auflockerung des Gesprächs gedacht.

- 1) Was ist Ihr derzeitiger Beruf / Job?
- 2) Seit wann etwa nutzen Sie das Internet?
- 3) Wie viele Stunden verbringen Sie etwa die Woche im Internet (einschließlich E-Mail Nutzung)?
- 4) Wie verbringen Sie die Zeit im Internet typischerweise (bei der Arbeit und zu Hause)?
- 5) Haben Sie eine Lieblingswebseite / Lieblingswebseiten?
- 6) Nutzen oder nutzten Sie schon einmal einen Webservice, bei dem Sie selbstständig etwas verwalten können? Wie heißt der Webservice, den Sie nutzen / nutzten? (Kunden)
- 6) Haben Sie Erfahrung mit Online Verwaltungsanwendungen? Welchen haben Sie bisher genutzt oder nutzen Sie zurzeit? (Administratoren)
- 7) Wie oft oder regelmäßig nutzen Sie die Verwaltungsfunktionen des Webservice (falls genannt)? (Kunden)
- 7) Wie oft oder regelmäßig nutzen Sie diese Online Verwaltungen (falls genannt)? (Administratoren)

Liste 8: Fragen im Kunden- und Administratorentest vor den Aufgaben

Die Fragen nach dem Aufgabenteil (Tabelle 10) sollte den Nutzern die Möglichkeit geben, ihre Meinung zu der Anwendung zu äußern, nachdem sie mit ihr gearbeitet haben. Hier könnte interessant sein, in wie weit die Meinung von der tatsächlichen Handlung des Nutzers abweichen würde. Diese Einschätzung darf zwar bei der Weiterentwicklung nicht überbewertet werden (getreu dem Motto von Nielsen „Don't Listen to Users“ [vgl. Nielsen 2001]). aber sie kann doch als Anregung genutzt werden. Für die Verbesserung der Usability wird vorrangig auf die Ergebnisse der Arbeit mit den Aufgaben wert gelegt.

Tabelle 10: Fragen nach dem Aufgabenteil im Nutzertest

Frage	Zweck
Haben Sie irgendwelche Funktionen vermisst? Oder gibt es eine Aufgabe, wo Sie nicht wüssten, wo Sie klicken würden?	Der Nutzer sollte die Möglichkeit haben, seinen Eindruck von der Anwendung zu reflektieren und Wünsche zu Funktionen zu äußern, die ihm fehlen würden. Würde der Nutzer Funktionen nennen, die ihm auch während des Tests schon gefehlt haben?
Fehlten Ihnen bestimmte Informationen? Waren Begriffe unklar?	Da die Anwendung sehr fachspezifisch ist, war es wichtig zu wissen, ob die Fachbegriffe verständlich waren (besonders beim Kundentest) und ob Informationen zum passenden Zeitpunkt präsentiert wurden.
Was hat Sie während der Arbeit mit dem System am meisten gestört / behindert?	Hier wäre besonders der Vergleich mit beobachteten Problemen interessant.

6.2 Durchführung der Nutzertests

Die Nutzertests wurden im Zeitraum zwischen 04.10.2006 und 12.10.2006 durchgeführt. Dazu wurden die Administratoren ins Büro des Anbieters eingeladen. Den Kunden wurde entgegengekommen, indem die Tests bei ihnen zu Hause stattfanden. Ich habe die Rolle des Untersuchungsleiters übernommen. Dabei habe ich die Teilnehmer durch den Test begleitet, Fragen gestellt und ihr Verhalten beobachtet. Die Tests fanden als Einzelsitzungen statt. Dabei war in den meisten Fällen der Anbieter als Beobachter anwesend. Um den Kunden durch die Vorbereitungsmaßnahmen (Aufbau des Laptops, Bereitlegen der Unterlagen) nicht zu verunsichern, unterhielt sich der Anbieter mit dem Kunden bis die Vorbereitungen abgeschlossen waren. Bei den Tests mit den Administratoren waren die Vorbereitungen des Arbeitsplatzes abgeschlossen bevor der Teilnehmer im Büro eintraf.

6.2.1 Einführung und Eingangsinterview

Jede Sitzung begann damit, dass der Teilnehmer begrüßt wurde und wir uns zum Testplatz setzen. Danach wurde dem Teilnehmer das Ziel des Tests und seine Aufgabe dabei erklärt. Es wurde betont, dass nicht er getestet wird, sondern die Anwendung. Dem Teilnehmer wurde mitgeteilt, dass er jederzeit Fragen stellen kann, diese aber evtl. nicht während des Tests beantwortet werden, da es wichtig ist, zu sehen, wie der Nutzer ohne Hilfe mit der Anwendung klar kommt. Dann wurde die Methode des lauten Denkens kurz erläutert. Schließlich wurde dem Teilnehmer mitgeteilt, dass das Gespräch zu Zwecken der Auswertung auf ein Diktiergerät aufgezeichnet wird und er wurde um seine Einverständniserklärung gebeten. Diese Erklärungen bekam der Teilnehmer in Form eines Hinweisblatts, auf dem er auch die Einverständniserklärung abgeben sollte, ausgehändigt (Anhang F).

Nachdem die formelle Einführung abgeschlossen war, konnte der Teilnehmer noch Fragen zum Verlauf stellen. Diese Gelegenheit nutzte aber kein Teilnehmer. Anschließend begann der Test mit dem einleitenden Interview, um Internetkenntnisse und Erfahrungen mit Webanwendungen des Nutzers zu erfahren. Die Ergebnisse des Interviews werden in Abschnitt 6.3.1 präsentiert.

6.2.2 Bearbeitung der Aufgaben und Abschlussinterview

Dem Teilnehmer wurde keine Einarbeitungsphase in die Anwendung gegeben, da es möglich sein soll in der Praxis diese Webanwendung bei Bedarf ohne Einarbeiten zu nutzen. Die Aufgaben wurden dem Teilnehmer einzeln auf Zetteln vorgelegt, damit er nicht die gesamte Anzahl sehen konnte. Dabei wurde der Testnutzer noch mal darauf hingewiesen, Fragen zu stellen, falls ihm bei der Aufgabenstellung etwas unklar ist, und beim Erledigen der Aufgabe laut zu denken.

Da das Verhalten des Nutzers nicht auf Video festgehalten wurde, wurden während der Bearbeitung Notizen gemacht. Außerdem wurde an Stellen, in denen die Beobachtung nicht eindeutig war oder der Nutzer nicht weiter wusste, Zwischenfragen gestellt, die den Teilnehmer möglichst nicht beeinflussten. Dennoch gab es Situationen, in denen eine Unterstützung darüber hinaus notwendig war. Da es sich um eine sehr fachspezifische Anwendung handelt, mussten z. B. zwei Kunden Begriffe erklärt werden, damit sie die Arbeit fortsetzen konnten, da die Beschreibung in der Anwendung anscheinend nicht ausreichend war. Einige Aufgaben wollten sowohl Kunden als auch Administratoren über Bereiche beginnen, die noch nicht funktionstüchtig waren. Dies wurde registriert, dennoch sollten die Nutzer auch die Aufgaben erledigen dürfen und deshalb wurde Ihnen ein entsprechender Einstiegspunkt gezeigt. So wollte beispielsweise ein Administrator zuerst nach dem Kunden anstatt nach dessen Mailbox suchen, was aber noch nicht implementiert war.

Um einen Einblick in die Gesprächsführung bei der Bearbeitung der Aufgaben zu geben, wird hier ein kleiner Teil einer Sitzung transkribiert, d. h. es wurde aus der Audioaufnahme und den gemachten Notizen der Verlauf schriftlich festgehalten. In der Nachbereitung der einzelnen Sitzungen wurden die Audioaufnahmen teilweise transkribiert, um Zitate für die Auswertung verfügbar zu haben.

Beispielhafter Ausschnitt aus dem Nutzertest mit einem Kunden:

(Kunde hat die Aufgabe 5 vorgelegt bekommen und durchgelesen)

Kunde:

„Da würd ich jetzt auf E-Mail.“ (klickt auf den Menüpunkt E-Mail)

(liest laut vor) „Neue E-Mail-Adresse. Da würd ich erst mal schauen, was hab ich schon.“

(klickt auf die Links zum Einblenden der E-Mail-Adressen bei seinen Mailboxen und liest sich die Adressen durch)

„Dann würd ich mal jetzt gleich das Sternderl eingeben und auf [sein 1. E-Mail-Konto], weil webmaster bin ich.“ (in dem Konto befindet sich die Adresse webmaster@...)

(schreibt *, bedient die Auswahl für die Domain)

„Wenn ich jetzt mehrere Domains hätte, dann könnt ich hier noch auswählen.“

(bedient die Auswahl für das Mailkonto und klickt auf Anlegen)

„Also ist auf jeden Fall deppeneinfach muss man sagen.“

(schaut sich weiter auf der Seite um und bleibt beim Untermenü stehen)

„Ja, da könnt ich jetzt noch E-Mail-Konto anlegen. Weil durch das unterlegte seh ich ja, wo ich grad bin, durch das blasse.“ (zeigt mit der Maus auf das grau geschriebene *E-Mail-Konto bearbeiten*) „Eigentlich logisch aufgebaut.“

Versuchsleiter:

„Sie meinen, dass grau unterlegt ist das, wo Sie sich grade befinden?“

Kunde: „Ja, einfach deswegen weil es sich von den anderen so tragisch unterscheidet.“

(Nachdem der Kunde die nächste Aufgabe bearbeitet hat, befindet er sich erst auf der E-Mail-Konto bearbeiten Seite.)

Versuchsleiter:

„Was würden Sie jetzt noch mal zu der Navigation oben sagen. Wo befinden Sie sich gerade?“

Kunde:

„Auch wieder bei E-Mail-Konto bearbeiten, weil es jetzt wieder stärker unterlegt ist als die anderen. Also es unterscheidet sich für mich deutlich.“ (Anmerkung: Jetzt ist dieser Link fett schwarz geschrieben.)

Nachdem alle Aufgaben erledigt waren, wurden dem Nutzer noch drei abschließende Fragen gestellt. Dabei wurde auf einen Fragebogen verzichtet sondern nochmals ein kurzes Interview geführt. Nach Abschluss der Untersuchung wurde sich bei dem Teilnehmer für seine Mitarbeit gedankt. Nachdem dieser verabschiedet worden war, wurden die Datenbankeinträge zurückgesetzt (besonders wichtig bei den Administratoren, weil diese mit demselben Benutzerdaten arbeiteten) und der Browsercache gelöscht, um keine Vorgaben zu bewahren. Da die einzelnen Tests nicht direkt hintereinander abliefen, blieb genug Zeit die Notizen aufzuarbeiten und die Antworten des Eingangsinterviews in eine Tabelle zu übernehmen.

6.3 Auswertung der Ergebnisse

6.3.1 Eingangsinterview – Erfahrungen der Nutzergruppen

Da dem Interview zu Beginn bei beiden Tests annähernd dieselben Fragen zu Grunde gelegt wurden, wird hier eine gemeinsame Auswertung gemacht. Die Antworten bestätigten die Vermutung, dass die Kunden eher geringe oder durchschnittliche Erfahrungen mit dem Internet haben und die Administratoren als Nutzer mit hohem Erfahrungsschatz angesehen werden können. Diese Aussagen lassen sich dadurch begründen, dass die Administratoren einen Beruf ausüben, in dem sie mehr mit dem Medium umgehen als die Kunden. Außerdem nutzen jene das Internet im Mittelwert sechs Jahre länger und fast viermal so lange pro Woche.

Der typische Einsatz des Internets unterscheidet sich dagegen kaum in den beiden Zielgruppen. Fünf der acht Teilnehmer gaben an, das Internet zur Suche nach Informationen zu nutzen. Weitere Zwecke waren Unterhaltung (Surfen und Spielen), E-Mail und berufsspezifische Aufgaben wie Wartung und Programmierung.

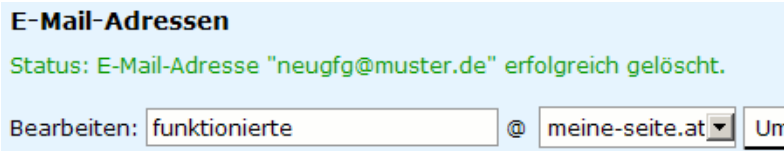
Bezüglich anderen Online Verwaltungsanwendungen hatten die Administratoren wieder mehr Erfahrung vorzuweisen. Während nur ein Administrator angab, solche Services noch nicht genutzt zu haben, gab lediglich ein Kunde an, Erfahrung damit zu haben. Es konnte erwartet werden, dass den Administratoren ihre Erfahrung bei der Nutzung der Anwendung zu Gute kommen wird.

6.3.2 Ergebnisse Kundenbereich

Die folgenden Usability Probleme wurden aus den Aussagen der Nutzer und den Beobachtungen zusammengefasst. Eine Bewertung wurde aufgrund der schon bei der Heuristischen Evaluation verwendeten Kriterien (Häufigkeit, Einfluss und Hartnäckigkeit) und Prioritätsstufen (vgl. Tabelle 5, S. 69) vorgenommen. Im Anhang G befindet sich die komplette Liste der 14 gefundenen Usability Probleme; hier werden nur die sieben schwerwiegenden bis katastrophalen Probleme erörtert (Tabelle 11).

Tabelle 11: Schwerwiegende und katastrophale Usability Probleme im Kundenbereich

Problembe- schreibung	Anmerkung zur Bewertung, Kundenzitat/-verhalten und Screenshot
Leistungen: Der Status der Leistungen wird nicht oder nicht seiner Bedeutung entsprechend verstanden.	Schwerwiegendes Problem: Dieses Problem mindert die Zufriedenheit und Wohlwollen des Kunden mit und gegenüber der Anwendung, weil Unklarheit über die Vertragsleistungen besteht. Drei der Testkunden waren sichtlich irritiert über die Angaben. Kundenzitate: „Was heißt Domains 1 2, ein halbes? Damit fang ich nichts an.“ „10 MB Webspace...“ (stutzt) „Achso, na das (zeigt auf die Zahl vor dem Schrägstrich) ist mein Status den ich jetzt hab und nach dem Schrägstrich ist das, auf was man aufrüsten kann.“ Status der Leistungen Domains: 2 / unbegrenzt Webpace: 10 MB / 2000 MB E-Mail-Konten: 2 / 35 Datenbanken: 0 / 1 Abbildung 23: Screenshot zum Usability Problem Leistungen (Kundenbereich)
E-Mail-Adresse anlegen: Die Zerteilung der Eingaben beim Anlegen einer E-Mail-Adresse ist nicht selbsterklärend.	Schwerwiegendes Problem: Es hat zwar nur zwei Kunden beeinflusst, diese wurden aber erheblich in ihrer Arbeit beeinträchtigt. Kundenverhalten: Zwei Kunden haben in das Eingabefeld für den lokalen Teil bereits die komplette E-Mail-Adresse eingegeben. Das Domainfeld wurde dabei nicht beachtet. Neue E-Mail-Adresse: @ muster.de Abbildung 24: Screenshot zum Usability Problem E-Mail-Adresse anlegen (Kundenbereich)

E-Mail-Konto bearbeiten: Der Weg von der Übersicht ins E-Mail-Konto ist schwer zu finden.	Katastrophales Problem: Die betroffenen Kunden konnten die Aufgabe ohne Hilfe durch den Versuchsleiter nicht beenden. Auch nach der Hilfe war ihnen der Link noch nicht klar. Das Problem hat damit großen Einfluss auf die Effektivität des Anwenders. Kundenverhalten: Die Hälfte der Kunden fand den Weg ohne Hilfe gar nicht, die andere mehr oder weniger problemlos. Die betroffenen Kunden waren recht verzweifelt bei diesem Schritt.  Abbildung 25: Screenshot zum Usability Problem E-Mail-Konten bearbeiten (Kundenbereich)
Statusmeldung: Statusmeldungen beim Anlegen und Löschen von E-Mail-Adressen werden nicht als Abschluss des Vorgangs wahrgenommen.	Schwerwiegendes Problem: Wenn der Abschluss einer Aufgabe nicht klar erkannt wird, führt das zu Unsicherheit und Unzufriedenheit beim Nutzer. Nur einer der Kunden hatte keine Probleme, das Ende einer Aufgabe klar zu definieren, alle anderen waren sich unsicher oder erkannten erst auf Nachfrage die Statusmeldung. Kundenzitate: „Jetzt müsst ich wahrscheinlich noch sagen <i>Speichern</i>. Aber speichern kann ich Passwort.“ „Denken Sie, die Sache ist jetzt abgeschlossen? – „Nein. (sieht die Statusmeldung) Oh ja, doch. Das heißt, ich könnte jetzt ne zweite anlegen.“  Abbildung 26: Screenshot zum Usability Problem Statusmeldung (Kundenbereich)
Löschen von E-Mail-Adressen: Der Link zum Löschen von E-Mail-Adressen wird nur schwer gefunden.	Schwerwiegendes Problem: Diesen Link haben alle Kunden nicht intuitiv gefunden, da er an unerwarteter Stelle war. Kundenzitat: (Schaut sich um, findet aber anscheinend das Löschen nicht.) „Wo ist das Löschen?“ (schaut weiter) „Ach da isses.“  Abbildung 27: Screenshot zum Usability Problem Löschen von E-Mail-Adressen (Kundenbereich)

Löschen Sicherheit: Es fehlt eine Sicherheitsabfrage, bevor das Löschen wirksam wird.	Schwerwiegendes Problem: Obwohl es nur für zwei Kunden offensichtlich unerwartet war, war es allen nicht bewusst, dass das Löschen endgültig war und bedeutet bei ungewollter Nutzung einen relativ hohen Aufwand zur Wiederherstellung. Dies verunsichert den Kunden über die Auswirkungen seiner Handlung und kann zu Unzufriedenheit führen. Kundenzitat: „Jetzt müsst ich wahrscheinlich noch sagen ‚Speichern‘. Aber speichern kann ich Passwort. Gibt’s sonst noch was zu speichern?“
Hinweise: Die Hinweise sind teilweise nicht kontextbezogen.	Schwerwiegendes Problem: Es wird eine hohe Erinnerungsleistung erwartet, besonders wenn die Informationen wirklich neu für den Nutzer sind. Dadurch ist der Nutzer weniger effektiv. Kundenverhalten: In einer anderen Situation hatte sich ein Kunde zwar die Erklärungen zu E-Mail-Konto und E-Mail-Adresse durchgelesen, konnte sie auf der nächsten Seite aber schon nicht mehr anwenden.

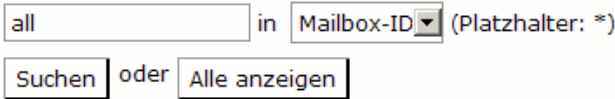
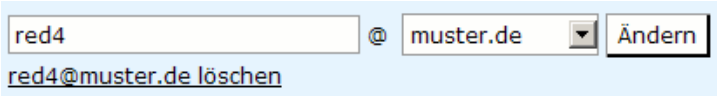
Im Abschlussinterview hat die letzte Frage eine überraschend gute Übereinstimmungen zwischen der subjektiven Meinung des Kunden und dem tatsächlich für ihn am schwerwiegendsten Problem gezeigt. Aufgrund der Aussagen allein wäre aber keine zufrieden stellende Beurteilung der Usability möglich gewesen. Die beiden anderen Fragen haben lediglich zu Anregungen und Vorschlägen der Kunden geführt, die in der weiteren Entwicklung als Inspiration dienen können.

6.3.3 Ergebnisse Administratorenbereich

Die Ergebnisse und Bewertung wurden wie beim Test des Kundenbereichs gewonnen. Es gab insgesamt 11 Usability Probleme, von denen fünf als schwerwiegend bis katastrophal eingestuft wurden (Tabelle 12). Es sind somit weniger Probleme als bei den Kunden und es wurden auch weniger Probleme als schwerwiegend eingeschätzt. Zwar können die beiden Tests nicht direkt miteinander verglichen werden, da der Untersuchungsgegenstand unterschiedlich war, aber insgesamt lässt sich doch festhalten, dass die Administratoren besser mit ihrem Bereich zu Recht kamen. Dies begründet sich auf ihre Erfahrungen mit ähnlichen Webanwendungen und ihrer allgemein höheren Erfahrung im Internet.

Tabelle 12: Schwerwiegende und katastrophale Usability Probleme im Administratorenbereich

<i>Problembeschreibung</i>	<i>Anmerkung zur Bewertung, Administratorzitat/-verhalten und Screenshot</i>
Offene Aufgaben: Die offenen Aufgaben wurden nach dem Verlassen der Startseite nicht für die Erfüllung weiterer Aufgaben in Betracht gezogen. Diese Funktion schien nicht selbsterklärend genug.	Schwerwiegendes Problem: Grundsätzlich kann ein Nutzer natürlich den von ihm bevorzugten Weg nehmen. In diesem Fall beeinflusst es die Effizienz während der Arbeit aber enorm. Drei Administratoren hätten die Statusänderung über das Menü durchgeführt, welches erheblich länger gedauert hätte (Suche notwendig). Administratorzitate: „Domains würd ich aber unter ‚Domains‘ genauso suchen, wenn ich schon einmal im System unterwegs bin.“ „Ja, wenn ich mich neu einloggen würde, und ich würd es auf der Startseite sehen, dann würde ich es auf der Startseite machen.“

Suchkriterien: Die Suchkriterien waren nicht ausreichend, um die Suche einzuschränken.	Katastrophales Problem: Jeder Administrator war davon betroffen und das Finden einer E-Mail-Adresse wurde ihnen dadurch sehr erschwert, was ihre Effektivität und Effizienz beeinflusst hat. Administratorzitat: „Hm, nach E-Mail-Adresse kann man nicht suchen.“
Alle anzeigen: Der Link zum Anzeigen aller Mailboxen wird an der aktuellen Position neben dem Suchbutton nicht wahrgenommen.	Schwerwiegendes Problem: Administratoren, die diese Funktion nutzen wollten, haben sie nicht gefunden und mussten sich alle Ergebnisse auf ineffizientere Weise anzeigen lassen. Da die Suchfunktion ein zentrales Element der Administratorbereichs ist, wurde dieses Problem als schwerwiegend eingestuft. Administratorzitat: (Administrator befindet sich auf der Mailboxen Suchseite) „Mir fehlt eine Gesamtliste.“ (nachgefragt, warum er den Link dort nicht gesehen hat) „... in der Regel sind Suchformulare so, dass man einen Suchknopf und einen Zurücksetzenknopf hat. Deshalb hab ich den Button automatisch als Zurücksetzen wahrgenommen.“  Abbildung 28: Screenshot zum Usability Problem Alle anzeigen (Administratorbereich)
Wechsel zwischen den Mailboxen: Es ist auf einer Mailbox bearbeiten Seite nicht ersichtlich welche weiteren Mailboxen der Kunde noch hat. Ein Wechsel dorthin ist nicht direkt möglich.	Schwerwiegendes Problem: Es führte bei den betroffenen Administratoren zu Einbußen in der Effizienz. Aufgrund der Aussagen der Administratoren bezüglich regelmäßiger Aufgaben kann davon ausgegangen werden, dass oft auf Mailboxen zugegriffen wird. Das Problem würde dabei dauerhaft stören. Administratorzitat: „Also in dieser Mailbox ist die E-Mail-Adresse nicht. Wenn der Kunde mehrere Mailboxen hat, wüsst ich jetzt adhoc nicht, wie ich danach suchen sollte.“
Löschen von Aliases: Der Link zum Löschen von Aliases wird nur schwer gefunden.	Schwerwiegendes Problem: Es war für alle Administratoren nicht intuitiv, weil sie den Link immer neben den Ändern-Button erwartet hatten und hat sie deshalb Zeit gekostet beim Erledigen der Aufgabe. Es ist sinnvoller die Darstellung an die Erwartungen des Nutzers anzupassen, da dies zu mehr Zufriedenheit mit der Anwendung führt. Administratorzitat: „Was mir ein wenig abgeht ist die Löschfunktion. Einfach leer machen (deutet auf das Eingabefeld für den Alias) und ändern ist eine problematische Variante. (schaut noch mal hin) AH! Da drunter ist das.“  Abbildung 29: Screenshot zum Usability Problem Löschen von Aliases (Administratorbereich)

Zentrale Aussage beim abschließenden Interview mit den Administratoren war, dass sie die Suche als besonders störend empfunden haben, da sie eine wichtige Rolle beim Erfüllen der Aufgaben gespielt hat. Auf die Frage bezüglich fehlender Informationen oder unklarer Begriffe wurde einmal das Verständnisproblem mit der Kategorie *Extras* genannt, ein anderes Mal die unklare Auswahl des Status bei den offenen Aufgaben. Beide Probleme wurden auch während der Nutzung der Webanwendung deutlich und als Usability Probleme aufgelistet. Vermisste Funktionen waren teilweise durch den Stand des Prototyps begründet und sind für weitere Entwicklungsschritte geplant. Ansonsten wurden Funktionen genannt, die zwar vorhanden sind, aber die der Administrator nicht wie erwartet nutzen konnte (z. B. Suche).

6.3.4 Maßnahmen zur Verbesserung der Usability

Die folgenden Änderungsvorschläge wurden im Rahmen dieser Diplomarbeit nicht mehr umgesetzt, sondern werden in die geplante Weiterentwicklung der Webanwendung einfließen. Ziel ist es, in jedem Fall die katastrophalen und schwerwiegenden Probleme bis zu einem nächsten Nutzertest behoben zu haben. Die Vorschläge beruhen dabei auf Usability Richtlinien und den Beobachtungen des Nutzerverhaltens in den Tests.

Lösungsvorschläge für den Kundenbereich

Leistungen:

Die Überschrift wird von *Status der Leistungen* in *Ihre aktuellen Leistungen* geändert. Die Leistungen werden in einer Tabelle mit den Spalten *genutzter Anteil* und *noch möglicher Anteil* dargestellt.

E-Mail-Adresse anlegen:

Das @-Zeichen zwischen den Eingabe- und Auswahlfeld muss stärker betont werden (z. B. durch Größe oder Abstand). Die Felder sollten mehr als Einheit wahrgenommen werden (z. B. durch eine Rahmung).

E-Mail-Konto bearbeiten:

Auf der E-Mail Übersichtseite wird die Liste der E-Mail-Konten umgestaltet. Die einzelnen E-Mail-Konten werden deutlicher voneinander getrennt dargestellt. Die Link zum Bearbeiten des Kontos wird größer als die Links zum Einblenden dargestellt und er wird in *E-Mail-Konto bearbeiten* umbenannt. Dafür gibt es Teilüberschriften für jedes E-Mail-Konto wie *E-Mail-Konto [Name]*.

Statusmeldung:

Der Bereich für die Statusmeldung wird etwas großzügiger gestaltet werden. Die Texte werden größer und fettgedruckt angezeigt oder der Hintergrund der Statusmeldung wird in der ursprünglichen Textfarbe dargestellt (Textfarbe dann hell).

Löschen von E-Mail-Adressen:

Anstatt des Links zum Löschen wird es neben dem Ändern-Button ein Lösch-Button geben.

Löschen Sicherheit:

Es wird vor dem endgültigen Löschen einer E-Mail-Adresse eine Sicherheitsabfrage gestellt. Diese wird in Form eines Popup-Feldes mit der Frage *Möchten Sie die [betroffene E-Mail-Adresse] endgültig und unwiderruflich Löschen?* im Sichtbereich des Lösch-Buttons gestellt werden. Das Popup kann mit *Ja* (Löschen zustimmen) und *Nein* (Löschen abbrechen) geschlossen werden. Dieser Mechanismus wird auch bei anderen Löschvorgängen zur Erhaltung der Konsistenz angewendet werden.

Hinweise:

Auf der E-Mail Übersichtseite wird noch mal deutlich gemacht, welche Funktionen auf der Seite E-Mail-Konto bearbeiten möglich sind und wie man dort hin gelangt. Für Begriffe sollte eine andere Informationsform gefunden werden, die auf jeder Seite alle Begriffserklärungen anzeigen kann (z. B. Feld *wichtige Begriffe*, das bei Bedarf eingeblendet werden kann)

Da der Weg von der E-Mail Übersicht in die einzelnen Konten durch den Nutzertest wiederholt als problematisch bewertet wurde (nach der Heuristischen Evaluation wurde das Problem schon versucht zu lösen) zeigt es, dass einerseits diese Veränderung für den Nutzer noch nicht ausreichend war und andererseits in einem nächsten Nutzertest dieser Aspekt nochmals kontrolliert werden sollte.

Lösungsvorschläge für den Administratorbereich

Offene Aufgaben:

Die Startseite wird in *Status Übersicht* und der entsprechende Menüpunkt *Start* in *Status* umbenannt. Dies soll helfen, diese Seite besser als Leitseite für den Status des Systems wahrzunehmen.

Suchkriterien:

Dem Auswahlmenü der Suchkriterien wird *Alias* hinzugefügt. Dieser wird als Standardeintrag ausgewählt sein.

Alle anzeigen:

Der Link zum Anzeigen aller Mailboxen wird aus dem Suchbereich entfernt und rechts daneben angezeigt. Evtl. muss die Schrift des Links noch vergrößert werden oder eine andere Form der Hervorhebung stattfinden.

Wechsel zwischen den Mailboxen:

Es wird auf der Mailbox bearbeiten Seite im rechten Teil eine Übersicht der anderen Mailboxen desselben Kunden geben, über dessen Links diese direkt erreicht werden können.

Löschen von Aliases:

Anstatt des Links zum Löschen wird es neben dem Ändern-Button ein Lösch-Button geben.

6.3.5 Bedeutung der Ergebnisse

Insgesamt können die Nutzertests als erfolgreicher Schritt des Usability Engineering für die Webanwendung angesehen werden. Es konnte festgestellt werden, dass die Mehrzahl der nach der Heuristischen Evaluation behobenen Probleme nicht wieder aufgetreten sind (soweit das im Rahmen der Aufgaben überprüft wurde). Zwei Probleme im Kundenbereich sind trotz der Lösungsversuche auch im Nutzertest aufgetreten und haben sogar zu erheblichen Schwierigkeiten geführt. Der Weg zur E-Mail-Konten Bearbeitung war durch den angebotenen Link den Kunden nicht klar. Dazu kann beigetragen haben, dass einigen das Konzept eines E-Mail-Kontos noch nicht bewusst war. Außerdem unterstützen die Statusmeldungen im Kundenbereich den Nutzer immer noch nicht ausreichend. Hier muss über eine Neupositionierung oder zumindest Hervorhebung zur Lösung dieses Problems nachgedacht werden (siehe auch Lösungen Kundenbereich im vorigen Abschnitt).

Zudem gab es im Kundenbereich zwei weniger ernsthafte Probleme, die weiterhin bestanden haben, aber die Kunden auch nicht sehr behindert haben. Das waren zum einen die Direktlinks, zum anderen die Überschriften, die nicht genug betont waren, damit die Kunden sie zur Orientierung wahrnahmen. Die Direktlinks wurden von den Kunden nicht benutzt. Das kann bedeuten, dass sie nicht genügend hervorgehoben waren (so wie das die Experten eingeschätzt hatten) oder aber dass es für die Kunden keine wichtige Beschleunigungsfunktion war. Aus den Beobachtungen des Nutzertests ist letzteres wahrscheinlicher.

Im Administratorbereich bestand das Usability Problem bezüglich des Links zur Mailbox (aus den Suchergebnissen heraus) auch noch, hatte aber wesentlich weniger Einfluss auf die Testteilnehmer als im Kundenbereich. Als ein Grund dafür können die Erfahrungen der Administratoren angesehen werden. Ihnen sind diese Art der Navigation vertrauter und die Konzepte dahinter bekannt. Die schwerwiegenden Probleme in diesem Bereich wurden alle erst durch den Nutzertest aufgedeckt. Dies beweist wie wichtig der Einsatz mehrere Usability Evaluationen ist und dass sich Expertenevaluationen und Nutzertests dabei gegenseitig ergänzen können.

Schlusswort

Ziel dieser Diplomarbeit war die benutzergerechte Konzeption und prototypischen Entwicklung einer Webanwendung, mit der Kunden und Administratoren Webhosting Leistungen verwalten können. Dazu wurden Usability Evaluationen in zwei Entwicklungsschritten der Anwendung durchgeführt. Die Heuristische Evaluation wurde als expertenorientierte Methode zur Beurteilung der Design Entwürfe der Webanwendung verwendet. Der Einsatz dieser Methode als Fernevaluation stellte sich jedoch als organisatorisch aufwendiger heraus, als eine Heuristische Evaluation als die Haupt-Methode des Discount Usability Engineering sein sollte. Trotzdem wurden noch viele Usability Probleme gefunden, obwohl während des Entwurfs bereits versucht wurde, allgemeine Usability Richtlinien umzusetzen. Da die geplante Webanwendung sehr fachspezifisch ist, mussten aber auch spezielle Richtlinien dafür beachtet werden. Eine Erkenntnis aus dieser Evaluation war deshalb, dass selbst bei Einhaltung von Usability Prinzipien die Usability Evaluation immer eine notwendige Kontrollinstanz sein sollte.

Bevor eine Evaluation mit Nutzern möglich war, musste ein funktionstüchtiger Prototyp entwickelt werden. Dabei wurde versucht, einen Großteil der Usability Probleme aus den Entwürfen zu beseitigen. Es wurde sich für den Einsatz eines PHP Frameworks namens *Symfony* entschieden, welches die Entwicklung durchaus beschleunigt hat. Außerdem bot es unterstützende Funktionen, um Web 2.0 Konzepte wie AJAX zu integrieren. Im Sinne einer technikbewussten Gestaltung der Benutzerschnittstelle, wie es die Software-Ergonomie fordert, wurden diese Konzepte zum Wohle des Benutzers eingesetzt. Dennoch bleibt abzuwarten, in welchen Umfang die neuen Möglichkeiten der Interaktion durch Web 2.0 Technologien sich für die Verbesserung der Usability etablieren. Diese Thematik wurde in der vorliegenden Diplomarbeit aber nicht näher betrachtet.

Dafür wurde sich mit einer weiteren Methode der Usability Evaluation, dem Nutzertest, auseinandergesetzt, indem ein solcher für jeden der beiden Anwendungsbereiche mit dem Prototyp durchgeführt wurde. Da dies eine sehr komplexe Methode in ihrer klassischen Form ist, wurden die Tests vereinfacht geplant, um im Rahmen der Diplomarbeit durchführbar zu sein. Die Durchführung der Tests gestaltete sich in mancher Hinsicht dennoch problematisch. Zum einen führte der Prototyp aufgrund seines geringen Funktionsumfang dazu, dass die Aufgaben die Teilnehmer in Bereiche führten, die noch nicht implementiert waren. Somit erfüllten jene den geplanten Zweck teilweise nicht, boten aber dafür überraschende Einblicke in das Nutzerverhalten. Als Versuchsleiterin und Entwicklerin des Prototyps fiel es mir darüber hinaus schwer, die Teilnehmer nicht zu beeinflussen und meine eigenen Vorstellungen über die Verwendung der Anwendung in den Hintergrund zu stellen. Dennoch können die Nutzertests als erfolgreich angesehen werden, da neue Usability Probleme entdeckt wurden und ich zudem als Entwickler wertvolle Einblicke in das Nutzerverhalten bekommen habe.

Die Webanwendung wird nach Abschluss dieser Arbeit weiterentwickelt werden und Usability Probleme aus den Nutzertests dabei als Erstes beseitigt werden. Außerdem wurde von Seiten des Auftraggebers die Bereitschaft zu weiteren Usability Evaluationen signalisiert. Es wurde in dieser Diplomarbeit gezeigt, wie mit relativ wenig Aufwand Usability Evaluationen erfolgreich schon in frühen Entwicklungsphasen (wo sie besonders wertvoll sind) durchge-

führt werden können. Die Anwendung von Usability Richtlinien durch die Designer und Entwickler ist nur ein Schritt in die richtige Richtung. Nur müssen in Zukunft noch mehr der Nutzen und die Bedeutung von Usability Evaluationen, insbesondere von Nutzertests, bewusst gemacht werden, damit die *konkreten* Anwender und deren Verhalten mehr berücksichtigt werden. Es bleibt offen in wie weit, eine Anpassung oder Weiterentwicklung der Methoden dazu beitragen kann, insbesondere die Web Usability zu fördern.

Literaturverzeichnis

EBERLEH, OBERQUELLE & OPPERMANN 1994

EBERLEH, Edmund ; OBERQUELLE, Horst ; OPPERMANN, Reinhard: *Einführung*. In: EBERLEH, Edmund (Hrsg.) ; OBERQUELLE, Horst (Hrsg.) ; OPPERMANN, Reinhard (Hrsg.): *Einführung in die Software-Ergonomie : Gestaltung graphisch-interaktiver Systeme: Prinzipien, Werkzeuge, Lösungen*. 2., völlig neu bearb. Aufl. Berlin : Walter de Gruyter & Co., 1994, S. 1-6

FOWLER & STANWICK 2004

FOWLER, Susan ; STANWICK, Victor: *Web Application Design Handbook : Best Practices for Web-Based Software*. San Francisco : Morgan Kaufmann Publishers, 2004

GARRETT 2005

GARRETT, Jesse J.: *Ajax: A New Approach to Web Applications*. URL <http://www.adaptivepath.com/publications/essays/archives/000385.php>, 2005 (abgerufen im September 2006)

GI-FACHGRUPPE SOFTWARE ERGONOMIE 2005

GESELLSCHAFT FÜR INFORMATIK E.V. (GI) - FACHGRUPPE SOFTWARE ERGONOMIE (Hrsg.): *Curriculum Software Ergonomie*. URL <http://www.gui-design.de/curriculum.pdf>, 2005 (abgerufen im Juli 2006)

GLASER 1994

GLASER, Wolff R.: *Menschliche Informationsverarbeitung*. In: EBERLEH, Edmund (Hrsg.) ; OBERQUELLE, Horst (Hrsg.) ; OPPERMANN, Reinhard (Hrsg.): *Einführung in die Software-Ergonomie : Gestaltung graphisch-interaktiver Systeme: Prinzipien, Werkzeuge, Lösungen*. 2., völlig neu bearb. Aufl. Berlin : Walter de Gruyter & Co., 1994, S. 7-51

HERCZEG 2005

HERCZEG, Michael: *Software-Ergonomie Grundlagen der Mensch-Computer-Kommunikation*. 2., vollständig überarb. Aufl. München : Wissenschaftsverlag GmbH, 2005

HEUER 2003

HEUER, Jens: *Expertenevaluation*. In: HEINSEN, Sven (Hrsg.) ; VOGT, Petra (Hrsg.): *Usability praktisch umsetzen : Handbuch für Software, Web, Mobile Devices und andere interaktive Produkte*. München : Carl Hanser Verlag, 2003, S. 115-135

IEA 2006

THE INTERNATIONAL ERGONOMICS ASSOCIATION (IEA) (Hrsg.): *IEA - The International Ergonomics Association*. URL <http://www.iea.cc/ergonomics/>, 2006 (abgerufen im Juli 06)

KALBACH 2003

KALBACH, James: *Von Usability überzeugen*. In: HEINSEN, Sven (Hrsg.) ; VOGT, Petra (Hrsg.): *Usability praktisch umsetzen : Handbuch für Software, Web, Mobile Devices und andere interaktive Produkte*. München : Carl Hanser Verlag, 2003, S. 7-21

KRUG 2006

KRUG, Steve: *Don't Make Me Think! A Common Sense Approach to Web Usability*. Second Edition. Berkeley : New Riders, 2006

MANHARTSBERGER & MUSIL 2001

MANHARTSBERGER, Martina ; MUSIL, Sabine: *Web Usability : Das Prinzip des Vertrauens*. Bonn : Galileo Design, 2001

MAYHEW 1999

MAYHEW, Deborah J.: *The Usability Engineering Lifecycle*. San Diego : Academic Press, 1999

MORKES & NIELSEN 1997

MORKES, John ; NIELSEN, Jakob: *Concise, Scannable, and Objective: How to Write for the Web*. URL <http://www.useit.com/papers/webwriting/writing.html>, 1997 (abgerufen im August 2006)

NIELSEN 1993

NIELSEN, Jakob: *Usability Engineering*. San Diego : Academic Press, 1993

NIELSEN & MACK 1994

NIELSEN, Jakob ; MACK, Robert L.: *Executive Summary*. In: NIELSEN, Jakob (Hrsg.) ; MACK, Robert L. (Hrsg.): *Usability Inspection Methods*. New York : John Wiley & Sons, 1994, S. 1-23

NIELSEN 1994

NIELSEN, Jakob: *Heuristic Evaluation*. In: NIELSEN, Jakob (Hrsg.) ; MACK, Robert L. (Hrsg.): *Usability Inspection Methods*. New York : John Wiley & Sons, 1994, S. 25-62

NIELSEN CA. 1994

NIELSEN, Jakob: *How to Conduct a Heuristic Evaluation*. URL http://www.useit.com/papers/heuristic/heuristic_evaluation.html, ca. 1994 (abgerufen im Oktober 2006)

NIELSEN 2000

NIELSEN, Jakob: *Why you only need to test with 5 users*. URL <http://www.useit.com/alertbox/20000319.html>, 2000 (abgerufen im Oktober 2006)

NIELSEN 2001

NIELSEN, Jakob: *First Rule of Usability? Don't Listen to Users (Alertbox)*. URL <http://www.useit.com/alertbox/20010805.html>, 2001 (abgerufen im August 2006)

NIELSEN & LORANGER 2006

NIELSEN, Jakob ; LORANGER, Hoa: *Prioritizing Web Usability*. Berkeley : New Riders, 2006

NIST 2006

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST): *Industry Usability Reporting : Increasing the Visibility of Software Usability*. URL <http://www.nist.gov/iusr>, 2006 (abgerufen im Oktober 2006)

NORM DIN EN ISO 9241 TEIL 11

NORM DIN EN ISO 9241 TEIL 11 2005. Ergonomie der Mensch-System-Interaktion : Anforderungen an die Gebrauchstauglichkeit – Leitsätze

PIROLI & CARD 1999

PIROLI, Peter ; CARD, Stuart K.: *Information Foraging*. URL <http://www2.parc.com/istl/groups/uir/publications/items/UIR-1999-05-Pirolli-Report-InfoForaging.pdf>, 1999 (abgerufen im August 2006)

ROSENFELD & MORVILLE 1998

ROSENFELD, Louis ; MORVILLE, Peter: *Information Architecture for the World Wide Web*. First Edition. Sebastopol : O'Reilly & Associates, Inc, 1998

SCHWEIBENZ & THISSEN 2003

SCHWEIBENZ, Werner ; THISSEN, Frank: *Qualität im Web : Benutzerfreundliche Webseiten durch Usability Evaluation*. Berlin : Springer-Verlag, 2003

THE CREOLE PROJECT 2006

THE CREOLE PROJECT (Hrsg.): *Creole – Trac*. URL <http://creole.phpdb.org/trac/>, 2006 (abgerufen im September 2006)

THE PROPEL PROJECT 2006

THE PROPEL PROJECT (Hrsg.): *Propel – Trac*. URL <http://propel.phpdb.org/trac/>, 2006 (abgerufen im September 2006)

THE SYMPHONY PROJECT 2006

THE SYMPHONY PROJECT (Hrsg.): *The symphony book*. URL <http://www.symfony-project.com/get/trunk/symfony-book.pdf>, 2006 (abgerufen im Juli 2006)

TOGNAZZINI 2003

TOGNAZZINI, Bruce: *First Principles of Interaction Design*. URL <http://www.asktog.com/basics/firstPrinciples.html>, 2003 (abgerufen im August 2006)

VOGT, HEUER & HEINSEN 2003

VOGT, Petra ; HEUER, Jens ; HEINSEN, Sven: *Ergebnisse präsentieren*. In: HEINSEN, Sven (Hrsg.) ; VOGT, Petra (Hrsg.): *Usability praktisch umsetzen : Handbuch für Software, Web, Mobile Devices und andere interaktive Produkte*. München : Carl Hanser Verlag, 2003, S. 221-229

VON GIZYCKI 2002

VON GIZYCKI, Vittoria: *Usability – nutzerfreundliches Web-Design*. In: BEIER, Markus (Hrsg.) ; VON GIZYCKI, Vittoria (Hrsg.): *Usability – nutzerfreundliches Web-Design*. Berlin : Springer, 2002, S. 1-17

WHARTON ET AL 1994

WHARTON, Cathleen ; RIEMAN, John ; LEWIS, Clayton ; POLSON, Peter: *The Cognitive Walkthrough Method: A Practitioner's Guide*. In: NIELSEN, Jakob (Hrsg.) ; MACK, Robert L. (Hrsg.): *Usability Inspection Methods*. New York : John Wiley & Sons, 1994, S. 105-140

WORLD WIDE WEB CONSORTIUM 2006

WORLD WIDE WEB CONSORTIUM (W3C) (Hrsg.): *Web Accessibility Initiative (WAI)*. URL <http://www.w3.org/WAI/>, 2006 (abgerufen im September 2006)

YAML.ORG 2006

YAML.ORG (Hrsg.): *YAML Ain't Markup Language*. URL <http://yaml.org/about.html>, 2006 (abgerufen im September 2006)

ZIMBARDO & GERRIG 1999

ZIMBARDO, Philip G. ; GERRIG, Richard J.: *Psychologie*. Berlin : Springer, 1999

Glossar

<i>Begriff</i>	<i>Erklärung</i>
Apache	Produkt der Apache Software Foundation und der meistverbreitete Webserver im Internet
dbmail	System zur Verwaltung von E-Mails in einer Datenbank (MySQL oder PostgreSQL)
E-Commerce	Elektronischer Handel, Handel über das Internet
Embedded Systems	Eingebettetes System; Rechnersystem, das weitgehend unsichtbar in verschiedenen Anwendungsbereichen und Geräten arbeitet (z. B. Unterhaltungselektronik)
Evaluation; Evaluator	Beschreibung, Analyse und Bewertung von Prozessen; Gutachter, Prüfer
Eye-Tracking	Blickmessung; Verfahren, mit dem gemessen wird, an welchen Punkten der Blick des Nutzers stehen bleibt, da nur dann Informationen wahrgenommen werden
Flash	proprietäre Entwicklungsumgebung des Herstellers Adobe Systems (früher Macromedia) zur Erstellung multimedialer Inhalte, sog. <i>Flashfilme</i>
Frames	Technik zur Unterteilung des Anzeigenbereichs eines Webbrowsers
Framework	unterstützende Programmierstruktur in der Softwareentwicklung, in der andere Software entwickelt und organisiert werden kann
Homepage	Startseite einer Webseite
Informationsarchitektur	Organisation von Information, um diese auffindbar zu machen
Instant Messenger	Software, um in Echtzeit mit anderen Teilnehmern zu kommunizieren (chatten)
Iteration, iterativ	Schrittweise Annäherung an die Lösung eines Problems durch wiederholte Anwendung desselben Verfahrens; schrittweise wiederholt
Kognition; kongnitiv	(psychologisch) die mentalen Prozesse eines Individuums zur Informationsverarbeitung, das Denken; auf Erkenntnis beruhend
Open Source	Programme, deren Quelltext eingesehen, weitergegeben und beliebig verändert werden kann
Persistenz; persistent	(Informatik) Fähigkeit, Daten in nicht-flüchtigen Speichermedien wie Dateisystemen oder Datenbanken zu speichern; dauerhaft

Photoshop	Bildbearbeitungsprogramm des Herstellers Adobe Systems
phpMyAdmin	freie PHP Anwendung zur Administration von MySQL Datenbanken
PocketPC	Bezeichnung für die Hardware eines PDA (Personal Digital Assistant) und teilweise das verwendete Betriebssystem darauf
Prototyping	Verfahren, bei denen ein Prototyp (Entwurfsmuster, Versuchsmodell) hergestellt wird
Refactoring	Vorgehensweise in der Softwareentwicklung bei der Lesbarkeit, Verständlichkeit, Wartbarkeit, Erweiterbarkeit und Struktur eines Computerprogramms verbessert werden
Transkription	Übertragung von Inhalten eines Films in eine schriftliche Form zur Analyse
Screen Design	Gestaltung der Bildschirmaufteilung
Screenreadern	Bildschirmleseprogramm; bietet Blinden und Sehbehinderten eine alternative Benutzerschnittstelle anstelle des Textmodus oder anstelle einer grafischen Benutzeroberfläche
Scrollbar	Bildlaufleiste, um den Inhalt eines Fensters zu verschieben
User Interface	Benutzeroberfläche, Benutzerschnittstelle
Software Engineering	Softwaretechnik; Bereitstellung und systematische Verwendung von Prinzipien, Methoden und Werkzeugen für die ingenieurmäßige Entwicklung und Anwendung von Softwaresystemen
Style Guide	Stilanleitung für die einheitliche Erstellung von Texten und sonstigen Kommunikationsmitteln
Stylesheet	Beschreibungssprache zur Definition von Formateigenschaften einzelner HTML-Elemente, siehe auch CSS
Template	(in einer Webanwendung) bietet eine Vorlage für verschiedene Inhaltsdaten zur Präsentation der Anwendung (HTML)
top-down	(Softwareentwicklung) wenn der Entwurf mit abstrahierten Objekten beginnt, die dann konkretisiert werden, wird von einer top-down Vorgehensweise gesprochen
Validierung	(von Benutzereingaben) Überprüfung, ob die Eingabe in dem vorgegebenen Wertebereich liegt oder einer vorgegebenen Wertemenge entspricht
Virtual Host	Funktion eines Webserver, damit mehrere Instanzen desselben Webservice auf einem physikalischen Server gehostet werden können

Webhosting	Unterbringung (Hosting) von Webseiten auf dem Webserver eines Internetdienstanbieters
Webmail	Dienste im Web, die die Verwaltung von E-Mails mit einem Webbrowser ermöglichen
Webpace	Speicherplatz, auf den über das Web zugegriffen werden kann

Anhang

A. Entwürfe und Style Guide

Dieser Anhang befindet sich auf der CD zur Diplomarbeit in dem Ordner *Entwurf*.

Dazu gehören folgende Dokumente:

- Datenbankschema der Maildatenbank (dbmail_schema.pdf)
- Datenbankschema der Webanwendungsdatenbank (wica_schema.pdf)
- Style Guide (style_guide.pdf)
- Grafische Entwürfe der Webanwendung (.jpg Dateien)

B. Dokumente zur Heuristischen Evaluation

Dieser Anhang befindet sich auf der CD zur Diplomarbeit in dem Ordner *Heuristische Evaluation*.

Dazu gehören folgende Dokumente:

- Flashfilm als Untersuchungsgegenstand der Evaluation (screendesign.swf, screendesign.html)
- Beschreibungen zum Flashfilm (beschreibung.pdf)
- Hinweisdokumente zu den einzelnen Schritten der Evaluation für die Experten (hinweise_heuristiken.pdf, hinweise_evaluation.pdf, hinweise_bewertung.pdf)
- Übersicht zum geplanten zeitlichen Ablauf für die Evaluation (zeitlicher_ablauf.pdf)

C. Heuristiken

Fehler und Status

Verständlichkeit von Statusmeldung

Bekommt der Benutzer präzise, verständliche Statusmeldungen über das Ergebnis nach einer ausgeführten Aktion?

Qualität der Statusmeldungen

Sind Statusmeldungen auf den ersten Blick sichtbar und als solche erkennbar?

Verständlichkeit von Fehlermeldungen

Sind die Fehlermeldungen in natürlicher Sprache ausgedrückt (keine Fehlercodes)?

Qualität der Fehlermeldungen

Beschreiben die Fehlermeldungen das Problem präzise und geben eine konstruktive Lösung vor?

Fehlerverhütung

Können Fehler bei der Bedienung durch das Interaktionsdesign vermieden werden?

Interaktion

Konsistenz der Interaktion

Erfolgt stets die gleiche Reaktion nach gleicher Aktion des Benutzers?

Effizienz der Benutzung

Sind häufig verwendete Interaktionselemente schnell erreichbar (z. B. Menü)?

Zweckgebundene Interaktionselemente

Ist die Gestaltung der Interaktionselemente bzw. das gesamte Interaktionskonzept so definiert, dass der Benutzer seine Ziele problemlos erreichen kann?

Beschleunigung der Benutzung

Können fortgeschrittene Benutzer die Interaktion mit dem System beschleunigen?

Leichtes Auffinden von Funktionen

Ist die Oberfläche so gestaltet, dass sie beim Auffinden von Inhalten und Funktionen dem Benutzer behilflich ist?

Benutzerfreiheit

Kann der Benutzer aus jeder Funktion problemlos aussteigen bzw. nach der Aktion zum Ausgangspunkt zurückkehren?

Standard von Interaktionselementen

Sind Interaktionselemente eindeutig als solche erkennbar (Links, Buttons, Felder, Boxen)?

Erkennen und Orientieren

Visuelle Konzepte

Werden visuelle Konzepte benutzt, die dem Benutzer aus der realen Welt vertraut sind?

Textverständnis

Nutzt das System Wörter und Phrasen, die dem Benutzer vertraut sind?

Klarer Zusammenhang zwischen Beschriftung und Funktion

Ist die Funktion von Elementen aus ihrer Benennung erkennbar?

Eindeutige Standortsbestimmung

Ist für den Benutzer immer klar ersichtlich, an welcher Position in der Navigation er sich befindet?

Konsistenz der Elemente

Sind wiederkehrende Elemente mit gleicher Funktionalität konsistent dargestellt bzw. bezeichnet?

Wieder erkennen statt sich erinnern

Sind Instruktionen zur Verwendung gut sichtbar bzw. bei Bedarf leicht auffindbar?

Gestaltung

Einfachheit der Darstellung

Sind zu jeder Zeit nur die tatsächlich notwendigen Informationen und Gestaltungselemente für den Benutzer sichtbar (auch in Suchergebnissen z. B.)?

Klare räumliche Anordnung

Sind zusammengehörige bzw. zusammenverwendbare Interaktionselemente durch räumliche Nähe und/oder farbliche Identität als solche erkennbar?

Passende Größenverhältnisse

Sind Inhaltselemente und Interaktionselemente nach ihrer Wichtigkeit in angemessene Größenverhältnisse gesetzt?

Lesbarkeit

Sind alle wiedergegebenen Inhalte erkennbar (lesbar)?

Eigene Vorschläge

D. Ergebnisse der Heuristischen Evaluation

In Tabelle 13 sind Probleme, die sehr kontrovers bewertet wurden, grau hinterlegt. In Tabelle 15 sind die Lösungen grau hinterlegt, deren Probleme mit der höchsten Priorität (3) eingestuft wurden.

Tabelle 13: Usability Probleme mit Lösungsvorschlägen der Experten und Bewertungsstufe der Ernsthaftigkeit (gerundeter Mittelwert)

<i>Fehler und Status</i>			
<i>Nr.</i>	<i>Problembeschreibung</i>	<i>Lösungsvorschlag</i>	<i>Stufe</i>
1	Statusmeldungen sind zu allgemein formuliert.	Sie sollten sowohl im positiven als auch im negativen Fall Auskunft über die genaue Veränderung geben (z. B. welcher Alias wurde geändert, der Status einer Aufgabe wurde inwiefern geändert).	2
2	Einige Formulierungen der Statusmeldungen geben Raum für Unklarheiten. (z. B. Status: Passwort konnte nicht geändert werden. Das <u>alte</u> Passwort bleibt gültig.)	Welches „alte“ ist hier gemeint? Besser wäre „bisher verwendete Passwort“.	2
3	Negative Statusmeldungen (rot) sind von Fehlermeldungen farblich und inhaltlich nicht unterscheidbar. Dies kann zu Unsicherheit beim Benutzer über seinen Einfluss auf das Geschehen führen. (z. B. Status: Passwort konnte nicht geändert werden. Das alte Passwort bleibt gültig.)		2
4	Statusmeldungen sind im Allgemeinen inkonsistent positioniert und können so nicht immer eindeutig der entsprechenden Aktion zugeordnet werden.	Es sollte lieber ein fester Bereich für die Statusmeldungen geschaffen werden oder sie sollten immer neben dem jeweiligen Aktionsbutton platziert werden.	2
5	Statusmeldungen bei der Bearbeitung von Aliases/E-Mail Adressen sind nicht konsistent platziert (im Adminbereich neben den Buttons, im Kundenbereich neben der Überschrift).	Neben den Buttons bietet eine besser Zuordnung bei mehreren Adressen, ist aber inkonsistent zu den restlichen Platzierungen der Statusmeldungen	2
6	Im Adminbereich sind die Fehlermeldungen auch in natürlicher Sprache.	Hier wären Fehlercodes aber durchaus angemessen und zielgruppenspezifisch richtig.	2
7	Fehlermeldungen sind zu allgemein formuliert.	Es sollte hinzugefügt werden, wo der Fehler aufgetreten ist (z. B. welche E-Mail Adresse existiert bereits).	2

8	Einige Formulierungen der Fehlermeldungen sind missverständlich.	„vorderen Teil“ der gewünschten Adresse; hier sollte besser ein anderer Ausdruck gefunden werden oder der entsprechende String angezeigt werden. Fehler: <u>Altes</u> Passwort fehlt. Bitte geben Sie ihr <u>aktuelles</u> Passwort ein. Altes fehlt, aktuelles eingeben? Hier sollten die Begriffe gleich sein. Fehler: <u>Neue E-Mail Adresse</u> fehlt. Bitte geben Sie den vorderen Teil der gewünschten Adresse ein. Sollte im Zusammenhang mit der Berichterstattung der „vorderen Teil“ Formulierung korrigiert werden. Fehler: Größenangabe fehlt. Bitte geben Sie <u>eine Zahl ein</u>. Besser: ...die neue Größe der Mailbox an.	3
9	Fehlermeldungen sind nicht eingabespezifisch formuliert.	Die Eingaben, die zum Fehler geführt haben, sollten mit erscheinen.	2
10	Systemfehler sind als Statusmeldungen nicht gut erkennbar.	Fehler in Nutzerfehler und Systemfehler unterteilen.	2
11	Unzureichender Lösungsvorschlag bei dem Fehler: Kein Mailkonto ausgewählt. Bitte wählen Sie ein Mailkonto aus.	Es sollte verständlich gemacht werden, dass E-Mail Adressen immer einem Mailkonto hinzugefügt werden müssen.	3
12	Die allgemeine Suchfunktion und die Mailboxen Suchfunktion konkurrieren miteinander auf einer Seite im Adminbereich.	Funktionalität sollte besser hervorgehoben werden (Beschriftung der allgemeinen Suche fehlt).	2
13	Beim Umbenennen von Aliases ist es möglich falsche Domains einzugeben.	Dies kann wie andersorts schon gemacht, durch eine Auswahl möglicher Domains verhindert werden.	3
Interaktion			
Nr.	Problembeschreibung	Lösungsvorschlag	Stufe
14	Links zum Einblenden von Infos verhalten sich inkonsistent gegenüber restlichen Links (es geht keine neue Seite auf).	Diese Links sollten deshalb anders hervorgehoben werden.	1
15	Passwort generieren wird durch eine Checkbox bewirkt. Passwort ändern durch einen Button.	Diese beiden Funktionen sollten getrennt werden oder beide über Buttons gesteuert werden.	2
16	Elemente zum Schnelleinstieg sind zwar gegeben, aber der Benutzer hat keine Möglichkeit diese zu individualisieren.		1
17	Der Admin kann beim Ändern des Passworts die Funktionen „Generieren“ und „Ändern“ nicht eindeutig unterscheiden.	Hier wäre eine konsistente Nutzung von Buttons besser.	3
18	Die Navigationspunkte unten (Kontakt, Hilfe etc) sind zu unscheinbar (klein, ungewöhnliche Position).	Eventuell sollten sie oben rechts positioniert werden.	1

19	In den Suchergebnissen fehlt der Tabellenkopf mit der Beschriftung für die Spalten. Dieser könnte das Verstehen der Inhalte fördern.		2
20	Kundendaten sind einmal über den Namen des Kunden, ein anderes Mal über „Kundendaten“ erreichbar.	Hier sollte beide Male der Name des Kunden verwendet werden.	2
21	Im Inhaltsbereich sind zwar einige Funktionen durch Rahmung und blauen Hintergrund abgegrenzt. Die Funktionen auf hellen Hintergrund sollten aber auch noch besser gerahmt werden, insbesondere um den Funktionsbereich der Buttons abzugrenzen.		1
22	Status: ... Bitte versuchen Sie es erneut. Bei anhaltendem Fehler nehmen Sie Kontakt mit dem <u>Administrator</u> von Netfarm auf.	Hier sollte die E-Mail Adresse des Administrators verlinkt werden.	2
23	Der Weg zur Bearbeitungsseite eines Mailkontos/einer Mailbox ist durch die ID nicht sofort erkennbar.	Hier könnte alternativ ein „Bearbeiten“ Link angeboten werden.	2
24	Der Weg zurück zu den Suchergebnissen ist nicht klar genug. Besonders nachdem eine Aktion auf einer Seite nach den Ergebnissen (Mailbox bearbeiten) ausgeführt wurde, kann der Zurück-Button des Browsers problematisch werden.	Evtl. Link „Suchergebnisse“ in Menü einbauen oder Link auf den Nachfolgeseite anbieten.	3
25	Die Begriffe Mailkonto, E-Mail, Webmail sind inhomogen verwendet.	Hier sollte eine Standardbenennung gewählt werden.	2
<i>Erkennen und Orientieren</i>			
<i>Nr.</i>	<i>Problembeschreibung</i>	<i>Lösungsvorschlag</i>	<i>Stufe</i>
26	Die Abgrenzung der funktionellen Bereiche klappt optisch nicht vollständig. Besonders auf der „Mailbox bearbeiten“ Seite ist nicht klar, ob der Button „Ändern“ bei den Nutzerdaten auch die Mailboxgröße ändern, da beides auf demselben Hintergrund liegen.	Hier sollte eine Rahmung Klarheit schaffen.	3
27	Hinweistext missverständlich: „vorhandene E-Mails zu solchen Adressen“		2
28	Begriffe wie Mailkonto, Hosting, Domain werden im Kundenbereich nicht erklärt.	Die Hinweise wäre eine gute Möglichkeit für Erklärungen.	3
29	Der Link „Info zum Server“ ist nicht selbsterklärend. Was erwartet den Benutzer hier?		2
30	Die Überschrift „Schnell: E-Mail Adresse anlegen“ ist zu fordernd. Der Benutzer könnte den Eindruck haben, dies „muss“ schnell erledigt werden.		1
31	Der Weg zur Bearbeitungsseite eines Mailkontos/einer Mailbox ist durch die ID nicht sofort erkennbar.	Hier könnte alternativ ein „Bearbeiten“ Link angeboten werden.	3

32	Wenn der „Passwort ändern“ Link auch auf die Mailbox bearbeiten Seite führt ist er überflüssig.	Anders wäre es, wenn er die Funktion auf derselben Seite einblendet würde (vgl. Alias einblenden). Dann sollte er aber auch anderes benannt werden.	2
33	Die Eingabefelder zum Ändern der E-Mail Adressen laden nicht zum direkten Ändern ein.	Evtl. könnte eine Beschriftung des Feldes wie „E-Mail Adresse umbenennen“ helfen.	2
34	Passwort generieren wird durch eine Checkbox bewirkt. Passwort ändern durch einen Button.	Diese beiden Funktionen sollte man trennen und beide über Buttons steuern.	2
35	Die Menüpunkte „Hilfe“ / „Kontakt“ usw. unterbrechen das System der Karteikartenreiter. Hier wird nicht mehr klar markiert, wo sich der Benutzer befindet.		2
36	„Mailkonto bearbeiten“ und „Mailbox bearbeiten“ erzeugen keine Hervorhebung im Menü und damit keine eindeutige Standortsbestimmung mehr. Es ist nicht klar, unter welchen Menüpunkt diese Seiten einzuordnen sind.		2
37	Die Begriffe Mailkonto, E-Mail, Webmail sind inhomogen verwendet.	Hier sollte eine Standardbenennung gewählt werden.	2
38	Hinweise fehlen auf der Startseite. Auch hier könnten Instruktionen oder Begriffserklärungen schon notwendig sein.		2
39	Im Admin-Bereich fehlt die Hilfe komplett.	Obwohl es professionelle Benutzer sein werden, so sollte doch eine Minimalhilfe verfügbar sein.	2
Gestaltung			
Nr.	Problembeschreibung	Lösungsvorschlag	Stufe
40	Im Adminbereich bilden der ausgewählte Menüpunkt und der Inhaltsbereich keine Einheit, obwohl sie mit derselben Farbe hinterlegt sind.	Evtl. sollte hier eine stärkere Rahmung des gesamten weißen Bereichs eingesetzt werden, oder die Menüpunkte mehr getrennt werden.	2
41	Der „Schnell-“ Bereich ist nicht genügend abgegrenzt, um seine besondere Rolle zu betonen.		1
42	Das Hauptmenü des Adminbereichs ist zu wenig hervorgehoben.	Es sollte wichtiger erscheinen.	1
43	Die Seitenüberschriften sind zu unauffällig.	Sie sollten stärker betont werden.	2
44	Die Links heben sich im schwarzen Text durch ihre Unterstreichungen zu wenig ab.	Evtl. sollten diese noch farbig gestaltet werden.	1
45	Der Link „Hinweise einblenden“ wirkt verloren und ohne Bezug. Er kann leicht übersehen werden.		2

Tabelle 14: Ergebnisse, die als Vorschläge der Experten eingeordnet wurden, mit Bewertungsstufe (gerundeter Mittelwert)

Nr.	Vorschlag	Stufe
1	Das gesamte Interaktionskonzept wechselt zwischen hierarchischer Struktur (Menü) und linearer Struktur (mehrere Funktionen auf einer Seite). Dies kann missverstanden werden oder der Benutzer findet sein Ziel nicht, obwohl beides an sich bekannte Konzepte sind.	1
2	Das visuelle Konzept der Karteikarten könnte für die Untermenü-Punkte übernommen werden.	1
3	E-Mail Adressen sollten schon eingeblendet sein im Kundenbereich. Die Mailkonto-Bezeichnungen sind evtl. zu wenig verständlich ohne dazugehörige E-Mail Adressen.	1
4	Statusmeldungen sollten einen extra Bereich erhalten, wo sie immer erscheinen.	1

Tabelle 15: Lösungen, die im Prototyp umgesetzt wurden, um Usability Probleme (Problem Nr.) zu beheben.

Problem Nr.	Lösung im Prototyp
1 und 7	Status- und Fehlermeldungen enthalten jetzt den Namen des Objekts, das bearbeitet wurde (z. B. der neue Name der E-Mail-Adresse)
4 und 5	Statusmeldungen werden jetzt immer unterhalb der dazugehörigen Teilüberschrift platziert.
8	Formulierungen verbessert (vorderer Teil -> lokaler Teil; altes Passwort -> aktuelles Passwort; Bitte geben Sie eine Zahl ein. -> Bitte geben Sie die neue Größe der Mailbox ein.)
11	Es ist standardmäßig immer ein Mailkonto ausgewählt.
12	Die allgemeine Suche wurde in diesem Prototyp noch nicht eingebaut, da der Funktionalitätsumfang diese noch nicht notwendig macht.
13	Es werden immer die verfügbaren Domains vorgeben, wenn Aliases angelegt oder geändert werden sollen.
14	Links zum Ein- und Ausblenden beginnen immer mit den Schlüsselwörter „Einblenden:“ und „Ausblenden:“.
15, 17 und 34	Passwort generieren und Passwort ändern werden jetzt beide über Buttons gesteuert.
19	Es wurde ein Tabellenkopf an den entsprechenden Stellen hinzugefügt.
20	Es wurde konsistent der Name des Kunden als Link zu seinen Daten verwendet.
21 und 26	Weißer Bereiche wurden entsprechend ihren Funktionszusammenhang auch mit einem Rahmen versehen.
23 und 31	Im Kundenbereich wurde der Link zur Bearbeitungsseite eines Mailkontos durch das Wort „bearbeiten“ ergänzt. Im Administratorbereich wurden vorerst keine Änderungen diesbezüglich vorgenommen.
24	Die Suchergebnisse sind jederzeit über den Zurückbutton des Browsers erreichbar ohne dass danach ausgeführte Änderungen beeinflusst werden.
25 und 37	Begriffe vereinheitlicht (Mailkonto -> E-Mail-Konto; E-Mail = E-Mail-Adresse außer der Menüpunkt „E-Mail“; Webmail = Webmail)

27	Die Texte des Hinweisbereichs wurden komplett überarbeitet.
28	Begriffe Mailkonto und E-Mail-Adresse werden in den Hinweisen erklärt. Domain soll auch in den Hinweisen erklärt werden, ist aber noch nicht Teil des Prototyps. Andere Begriffe, deren Erklärung kurz genug ist, werden über das Akronym-Tag erklärt (z. B. Weospace). Hosting sollte schon klar gestellt worden sein, bevor ein Besucher der Webseite Kunde wird (nicht Teil der Webanwendung).
30	Der Begriff „Schnell:“ wurde durch „Direkt:“ ersetzt.
32	An genannter Stelle wurde der Link in „Einblenden: Passwort ändern“ umbenannt.
33	Vor die entsprechenden Eingabefelder wurde die Beschriftung „Bearbeiten“ eingefügt.
36	„E-Mail-Konto bearbeiten“ wurde als Punkt des Untermenüs zu „E-Mail“ aufgenommen. Der Link erscheint in grauer Schrift und inaktiv, wenn der Nutzer sich nicht auf der Seite befindet. Der Link wie die anderen Menüpunkte auch hervorgehoben, wenn sich der Nutzer auf der Seite befindet.
38	Hinweise auf der Startseite hinzugefügt.
39	Link zur Hilfe im Administratorbereich hinzugefügt.
40 und 42	Es wurde die Menügestaltung des Kundenbereichs auch im Administratorbereich übernommen, da sie stimmiger ist und keine Probleme damit durch die Experten festgestellt wurden.
45	Der Link zum Ein- und Ausblenden der Hinweise wurde auf die Ebene des Untermenüs verlagert und bildet mit dem standardmäßig eingeblendeten Hinweisbereich farblich eine Einheit (hellblau hinterlegt).

E. Prototyp der Webanwendung

Der Prototyp der Webanwendung kann im Internet unter den URLs <http://wica.netfarm.at> (Kundenbereich) und <http://wica.netfarm.at/backend.php> (Administratorbereich) erreicht werden. Außerdem befinden sich auf der CD zur Diplomarbeit im Ordner *Prototyp* die Quellcode Dateien zum Stand des Prototyps beim Nutzertest. Es sei darauf hingewiesen, dass die Anwendung nur mit installiertem Framework lauffähig ist. Die Dateien des Frameworks, die nicht direkt zur Anwendung gehören, sind nicht auf der CD vorhanden.

F. Dokumente zu den Nutzertests

Dieser Anhang befindet sich auf der CD zur Diplomarbeit in dem Ordner *Nutzertests*.

Dazu gehören folgende Dokumente:

- Hinweisdokument zur Einführung der Teilnehmer in den Test und Einverständniserklärung zur Audioaufnahme (hinweise.pdf)
- Tabellen mit Ergebnissen der Eingangsinterviews (ergebnisse_interview.pdf)

G. Ergebnisse der Nutzertests

Tabelle 16: Usability Probleme im Nutzertest Kunden mit Bewertung durch den Versuchsleiter

<i>Anmeldung</i>		
<i>Usability Problem</i>	<i>Kundenzitat -/verhalten</i>	<i>Bewertung</i>
<i>Passwort-Eingabe:</i> Bei dem Eingabefeld für das Passwort gibt es keine Hilfestellung, was bei der Eingabe beachtet werden muss.	„Muss ich die Groß-/Kleinschreibung beachten?“ Hat alle Kunden ziemlich genervt, dass sie gleich zu Beginn eine Fehlermeldung hatten.	2 (Alle Kunden hatten das Ausrufezeichen nicht eingegeben und erhielten bereits beim Einstieg eine Fehlermeldung und waren entsprechend irritiert dadurch.)
<i>Fehlermeldung:</i> Die Fehlermeldung wegen falschem Passwort ist nicht hilfreich genug.	„Also ich hab Groß-/Kleinschreibung beachtet. (schaut nochmal auf den Aufgabenzettel) Achso, das Rufzeichen hab ich vergessen.“	2 (Alle Kunden konnten sich im zweiten Anlauf erfolgreich anmelden. Dieses Problem sollte nicht dauerhaft stören.)
<i>Kategorien und Standortbestimmung</i>		
<i>Usability Problem</i>	<i>Kundenzitat-/verhalten</i>	<i>Bewertung</i>
<i>Kategorie Datenbank:</i> Die Kategorie <i>Datenbank</i> wird nicht verstanden oder die Nutzer haben völlig andere Vorstellungen zu dem Begriff.	Ein Kunde würde neue Leistungen beispielsweise unter Datenbank bestellen. „Datenbank, weiß ich jetzt nicht, keine Ahnung“	2 (Der Begriff war allen Kunden auf die ein oder andere Art unklar, sollte den Kunden aber nicht dauerhaft stören.)
<i>Kategorie Mein Konto:</i> Die Kategorie <i>Mein Konto</i> wird mit der Vorstellung aus dem Bankwesen verknüpft bzw. ist nicht selbsterklärend.	Bei den Erwartungen zu den Kategorien haben sich alle Kunden schwer getan, zu dieser Kategorie ihre Vorstellungen zu äußern.	1 (Dieses Problem beeinflusst den Nutzer nicht dauerhaft, da er nach einmaliger Nutzung seine Vorstellungen angepasst haben sollte.)
<i>Untermenü:</i> Die Standortbestimmung durch das Untermenü im Bereich E-Mail ist nicht eindeutig.	„Weil durch das unterlegte seh ich ja, wo ich grad bin, durch das blasse.“ (zeigt auf das grau geschriebene E-Mail-Konto bearbeiten) Auf einer anderen Seite sieht er das fettgedruckte wieder als Standort.	1 (Das Problem hat den betreffenden Kunden nicht bei seiner Arbeit beeinflusst.)
<i>Überschriften:</i> Überschriften werden kaum wahrgenommen und können somit die Standortbestimmung auch nicht unterstützen.		1 (Das Problem muss nicht dringend geändert werden, da es die Kunden nicht sehr bei ihrer Arbeit beeinflusst hat.)
<i>Startseite</i>		
<i>Usability Problem</i>	<i>Kundenzitat-/verhalten</i>	<i>Bewertung</i>
<i>Direktlinks:</i> Die Direktlinks werden nicht genutzt oder entsprechen nicht den regelmäßigen Aufgaben des Kunden.	Die Frage bezüglich regelmäßiger Aufgaben ergab Abweichungen von den als Direktlinks angebotenen Aufgaben.	1 (Für Kunden scheint das Menü zur Navigation ausreichend, da es keiner als umständlich empfand.)

<i>Leistungen:</i> Der Status der Leistungen wird nicht oder nicht seiner Bedeutung entsprechend verstanden.	„Was heißt Domains 1 2, ein halbes? Damit fang ich nichts an.“ „10 MB Webspace...“ (stutzt) „Achso, na das (zeigt auf die Zahl vor dem Schrägstrich) ist mein Status den ich jetzt hab und nach dem Schrägstrich ist das, auf was man aufrüsten kann.“	3 (Dieses Problem mindert die Zufriedenheit und Wohlwollen des Kunden mit und gegenüber der Anwendung, weil Unklarheit über die Vertragsleistungen besteht. Drei der Testkunden waren sichtlich irritiert über die Angaben.)
<i>E-Mail Übersicht und E-Mail-Konto bearbeiten</i>		
<i>Usability Problem</i>	<i>Kundenzeit/-verhalten</i>	<i>Bewertung</i>
<i>E-Mail-Adresse anlegen:</i> Die Zerteilung der Eingaben beim Anlegen einer E-Mail-Adresse ist nicht selbsterklärend.	Zwei Kunden haben in das Eingabefeld für den lokalen Teil bereits die komplette E-Mail-Adresse eingegeben. Das Domainfeld wurde dabei nicht beachtet.	3 (Es hat zwar nur zwei Kunden beeinflusst, diese wurden aber erheblich in ihrer Arbeit beeinträchtigt.)
<i>E-Mail-Konto bearbeiten:</i> Der Weg von der Übersicht ins E-Mail-Konto ist schwer zu finden.	Die Hälfte der Kunden fand den Weg ohne Hilfe gar nicht, die andere mehr oder weniger problemlos. Die betroffenen Kunden waren recht verzweifelt bei diesem Schritt.	4 (Die betroffenen Kunden konnten die Aufgabe ohne Hilfe durch den Versuchsleiter nicht beenden. Auch nach der Hilfe war ihnen der Link noch nicht klar. Das Problem hat damit großen Einfluss auf die Effektivität des Anwenders.)
<i>Statusmeldung:</i> Statusmeldungen beim Anlegen und Löschen von E-Mail-Adressen werden nicht als Abschluss des Vorgangs wahrgenommen.	„Jetzt müsst ich wahrscheinlich noch sagen <i>Speichern</i> . Aber speichern kann ich Passwort.“ „Denken Sie, die Sache ist jetzt abgeschlossen? – „Nein. (sieht die Statusmeldung) Oh ja, doch. Das heißt, ich könnte jetzt ne zweite anlegen.“	3 (Wenn der Abschluss einer Aufgabe nicht klar erkannt wird, führt das zu Unsicherheit und Unzufriedenheit beim Nutzer. Nur einer der Kunden hatte keine Probleme, das Ende einer Aufgabe klar zu definieren, alle anderen waren sich unsicher oder erkannten erst auf Nachfrage die Statusmeldung.)
<i>Löschen von E-Mail-Adressen:</i> Der Link zum Löschen von E-Mail-Adressen wird nur schwer gefunden.	(Schaut sich um, findet aber anscheinend das Löschen nicht.) „Wo ist das Löschen?“ (schaut weiter) „Ach da is es.“	3 (Diesen Link haben alle Kunden nicht intuitiv gefunden, da er an unerwarteter Stelle war.)
<i>Löschen Sicherheit:</i> Es fehlt eine Sicherheitsabfrage, bevor das Löschen wirksam wird. Dies verunsichert den Kunden über die Auswirkungen seiner Handlung.	„Jetzt müsst ich wahrscheinlich noch sagen <i>Speichern</i> “. Aber speichern kann ich Passwort. Gibt's sonst noch was zu speichern?“	3 (Obwohl es nur für zwei Kunden offensichtlich unerwartet war, war es allen nicht bewusst, dass das Löschen endgültig war und bedeutet bei ungewollter Nutzung einen relativ hohen Aufwand zur Wiederherstellung. Dies verunsichert den Kunden über die Auswirkungen seiner Handlung und kann zu Unzufriedenheit führen.)

<i>Hinweise:</i> Die Hinweise sind teilweise nicht kontextbezogen.	In einer anderen Situation hatte sich ein Kunde zwar die Erklärungen zu E-Mail-Konto und E-Mail-Adresse durchgelesen, konnte sie auf der nächsten Seite aber schon nicht mehr anwenden.	3 (Es wird eine hohe Erinnerungsleistung erwartet, besonders wenn die Informationen wirklich neu für den Nutzer sind. Dadurch ist der Nutzer weniger effektiv.)
--	---	---

Tabelle 17: Usability Probleme im Nutzertest Administratoren mit Bewertung durch den Versuchsleiter

<i>Startseite</i>		
<i>Usability Problem</i>	<i>Kundenzitat/-verhalten</i>	<i>Bewertung</i>
<i>Kategorie:</i> Die Kategorie <i>Extras</i> ist zu unspezifisch und weckt beim Nutzer keine klaren Vorstellungen darüber.	„Extras – kann ich mir jetzt nicht viel vorstellen.“	2 (Drei Administratoren hatten gar keine konkreten Vorstellungen dazu und würde es dementsprechend auch nicht intuitiv nutzen. Da sich in dieser Kategorie aber keine Hauptfunktionen befinden, wurde das Problem als weniger schwerwiegend eingestuft.)
<i>Schnellzugriff:</i> Entspricht nicht, den erwünschten Aufgaben, die Administratoren häufig abarbeiten müssen oder zu denen sie einen Status sehen wollen.	Zwei Administratoren würden sich eher eine Übersicht oder zumindest einen Schnellzugriff auf Zahlungseingänge und offene Rechnungen wünschen. „Domain anlegen und so das muss ich nicht schnell machen, aber Zahlungen muss ich täglich eintragen und das muss schnell gehen.“	2 (Es hatte auf die aktuellen Aufgaben keinen Einfluss, sollte aber in einem später Test nochmals überprüft werden, wenn die dazugehörige Funktionalität vorhanden ist.)
<i>Offene Aufgaben Status:</i> Die Statusauswahl der offenen Aufgaben sind nicht logisch nachvollziehbar und führt zu Unsicherheit bei der Benutzung.	Alle Administratoren, die die entsprechende Aufgabe unter offene Aufgaben erledigt haben, waren skeptisch, welchen Status sie jetzt wählen sollten. „Vielleicht sollt ich mir vorher aussuchen welche Stati es gibt. Aber nachdem ich ja nicht weiß welche Stati es gibt, tu ich mir bissle schwer herauszufinden welche Stati es gibt.“	2 (Die Statusmeldungen sind nicht selbsterklärend. Trotzdem konnten die meisten Administratoren die betreffende Aufgabe ohne weitere Probleme erfüllen. Auch dieses Problem muss bei erweitertem Funktionsumfang nochmals beurteilt werden.)
<i>Offene Aufgaben:</i> Die offenen Aufgaben wurden nach dem Verlassen der Startseite nicht für die Erfüllung weiterer Aufgaben in Betracht gezogen. Diese Funktion schien nicht selbsterklärend genug.	„Domains würd ich aber unter ‚Domains‘ genauso suchen, wenn ich schon einmal im System unterwegs bin.“ „Ja, wenn ich mich neu einloggen würde, und ich würd es auf der Startseite sehen, dann würd ich es auf der Startseite machen.“	3 (Grundsätzlich kann ein Nutzer natürlich den von ihm bevorzugten Weg nehmen. In diesem Fall beeinflusst es die Effizienz während der Arbeit aber enorm. Drei Administratoren hätten die Statusänderung über das Menü durchgeführt, welches erheblich länger gedauert hätte (Suche notwendig).)

<i>Suche</i>		
<i>Usability Problem</i>	<i>Kundenzitat/-verhalten</i>	<i>Bewertung</i>
<i>Suchkriterien:</i> Die Suchkriterien waren nicht ausreichend, um die Suche einzuschränken.	„Hm, nach E-Mail-Adresse kann man nicht suchen.“	4 (Jeder Administrator war davon betroffen und das Finden einer E-Mail-Adresse wurde ihnen dadurch sehr erschwert, was ihre Effektivität und Effizienz beeinflusst hat.)
<i>Alle anzeigen:</i> Der Link zum Anzeigen aller Mailboxen wird an der aktuellen Position neben dem Suchbutton nicht wahrgenommen.	(Administrator befindet sich auf das der Mailboxen Suchseite) „Mir fehlt eine Gesamtliste.“ (nachgefragt, warum er den Link dort nicht gesehen hat) „... in der Regel sind Suchformulare so, dass man einen Suchknopf und eine Zurücksetzenknopf hat. Deshalb hab ich den Button automatisch als Zurücksetzen wahrgenommen.“	3 (Administratoren, die diese Funktion nutzen wollten, haben sie nicht gefunden und mussten sich alle Ergebnisse auf ineffizientere Weise anzeigen lassen. Da die Suchfunktion ein zentrales Element der Administratorbereichs ist, wurde dieses Problem als schwerwiegend eingestuft.)
<i>Suchergebnis:</i> Der Link zum Bearbeiten der Mailbox ist zu unscheinbar.	„Jetzt seh ich zwar das Suchergebnis, aber ich würde halt gerne über das Suchergebnis auf den User kommen. Geht das jetzt über die Mailbox-ID?... Das ist mir nicht eindeutig genug.“	1 (Es hat nur zwei Administratoren gestört und sollte nicht dauerhaft ein Problem darstellen.)
<i>Suchergebnis Inhalt:</i> Die Spalte <i>Kunde</i> in der Ergebnisliste wurde missverstanden.	Es gab bei zwei Administratoren Verständnisprobleme darüber, ob der Kundename nun zweimal existiert oder ob der Kunde zwei Mailboxen hat. „Eigentlich sollte es den Kunden Max Mustermann nur einmal geben.“	1 (Es hat nur zwei Administratoren betroffen und diese nicht erheblich in ihrer Arbeit gestört.)
<i>Mailbox bearbeiten</i>		
<i>Usability Problem</i>	<i>Kundenzitat/-verhalten</i>	<i>Bewertung</i>
<i>Wechsel zwischen den Mailboxen:</i> Es ist auf einer Mailbox bearbeiten Seite nicht ersichtlich welche weiteren Mailboxen der Kunde noch hat. Ein Wechsel dorthin ist nicht direkt möglich.	„Also in dieser Mailbox ist die E-Mail-Adresse nicht. Wenn der Kunde mehrere Mailboxen hat, wüsst ich jetzt adhoc nicht, wie ich danach suchen sollte.“	3 (Es führte bei den betroffenen Administratoren zu Einbußen in der Effizienz. Aufgrund der Aussagen der Administratoren bezüglich regelmäßiger Aufgaben kann davon ausgegangen werden, dass oft auf Mailboxen zugegriffen wird. Das Problem würde dabei dauerhaft stören.)

<i>Löschen von Aliases:</i> Der Link zum Löschen von Aliases wird nur schwer gefunden.	„Was mir ein wenig abgeht ist die Löschfunktion. Einfach leer machen (deutet auf das Eingabefeld für den Alias) und ändern ist eine problematische Variante. (schaut noch mal hin) AH! Da drunter ist das.“	3 (Es war für alle Administratoren nicht intuitiv, weil sie den Link immer neben den Ändern-Button erwartet hatten und hat sie deshalb Zeit gekostet beim Erledigen der Aufgabe. Es ist sinnvoller die Darstellung an die Erwartungen des Nutzers anzupassen, da dies zu mehr Zufriedenheit mit der Anwendung führt.)
<i>Farbe:</i> Die Aliases sind zu stark hervorgehoben und haben somit eine zu dominante Position auf der Seite.	"Hm, das war jetzt etwas verwirrend das jetzt die Aliases gleich da stehen." – Warum? – „Na, weil das Blau da so hervorschießt.“	1 (Es hat nur einen Administrator betroffen und würde ihn aber nicht dauerhaft stören.)

Thesen

Interaktionen zwischen Mensch und Maschine können nur verbessert werden, wenn die Benutzungsschnittstelle an den Menschen angepasst wird.

Usability Evaluationen müssen in den Entwicklungsprozess eines Produkts eingebunden werden, um dessen Usability zu optimieren.

Durch die hohe Vielzahl von Webseiten ist Usability ein entscheidendes Qualitätsmerkmal im Web geworden.

Bei der Entwicklung von Webanwendungen müssen fachspezifische Usability Richtlinien beachtet werden.

Die modernen Internet-Technologien können dazu beitragen, die Usability von Webanwendungen zu verbessern.

Die Einhaltung von allgemeinen Usability Richtlinien kann nicht die Usability Evaluation des Produkts mit zielgruppenspezifischen Nutzern ersetzen.

Mit der Heuristischen Evaluation werden viele leichte Usability Probleme gefunden.

Der Nutzertest ist die wichtigste Methode der Usability Evaluation.

Nutzertests sind erfolgreicher, wenn der Prototyp ausreichend Funktionalität zum Testen bietet.

Expertenorientierte und nutzerorientierte Methoden der Usability Evaluation ergänzen sich und erhöhen bei gemeinsamer Anwendung während des Usability Engineering die Qualität der Endprodukts.

Für die Usability Evaluation haben sich in der Praxis Methoden des Discount Usability Engineering als schnelle und erfolgreiche Formen erwiesen.

Selbstständigkeitserklärung

Ich versichere, dass ich die Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Melanie Reichel, 28. November 2006