

Evaluation der Usability von Software für niedergelassene Ärzte anhand des Beispiels von „orbImmun“

Bachelorarbeit
zur Erlangung des Titels

„Bachelor of Science Medizinische Informatik“

der Privaten Universität für Medizinische Informatik und Technik Tirol

vorgelegt von

Matthias Gruber
aus
Feldkirch, Vorarlberg

Innsbruck, 2004



Zusammenfassung

Im medizinischen Bereich ist die Qualität von Software besonders kritisch, weshalb der Thematik der Überprüfung von Qualität eine besondere Rolle zukommt. Eines der wichtigsten Unterscheidungsmerkmale in Verbindung mit Softwarequalität ist Usability (zu Deutsch: Gebrauchstauglichkeit). Usability setzt sich aus einer Vielzahl von Faktoren zusammen, hauptsächlich aus den Produktkriterien Effizienz, Effektivität und Zufriedenheit (siehe auch Heinsen 2003).

Diese Arbeit behandelt die Evaluation der Usability eines Softwaretools für niedergelassene Ärzte. Da Usability nicht direkt gemessen werden kann, wurde eine Vielzahl von Methoden entwickelt, um die Evaluation von Usability zu ermöglichen. Nach der grundlegenden Theorie werden die wichtigsten Methoden zur Usability Evaluation vorgestellt. Im Rahmen dieser Arbeit wurden vor allem eine Expertenevaluation unter Zuhilfenahme von Heuristiken (Norman 1988; Norman 1990; Nielsen 1993; Shneiderman 1998) und eine Fragebogenevaluation nach Vorbild der IsoMetrics Fragebögen (Willumeit 1996) durchgeführt. Für die Expertenevaluation wurde eine Sammlung von Heuristiken der angeführten Autoren erstellt, die aufgrund der Redundanzen und der großen Anzahl zunächst selektiert und für die eigentliche Evaluation frei zusammengestellt wurde. Diese Vorgangsweise empfiehlt auch (Nielsen 1993).

Gegenstand der Evaluation war die Software „orbImmun“, die von der Firma „Ablinger & Garber Publishing“ in Hall in Tirol betreut und gemeinsam mit der Österreichischen Ärztekammer vertrieben wird. orbImmun ist ein unter Microsoft Windows lauffähiges Reiseimpfprogramm, das niedergelassene Ärzte bei der Reiseberatung von Patienten unterstützt. Es bietet Informationen bezüglich Reisen außerhalb Österreich und Europa, der Auswahl von Impfstoffen und Medikamenten und hilft bei der Rezepterstellung.

Bei der internen Evaluation wurde bereits eine große Anzahl an Verbesserungsmöglichkeiten (Design, Ergonomie u.a.) festgestellt. Diese Ergebnisse wurden größtenteils durch die Fragebogenevaluation bestätigt und es wurden noch weitere Punkte entdeckt. Weiters kam auch eine große Anzahl von neuen Ideen und interessanten Anregungen zusammen.

Insgesamt wurden viele Verbesserungspunkte gefunden, von denen einige beim vorliegenden Programm nur schwer zu korrigieren sein werden. Die Empfehlungen gehen deshalb in Richtung einer umfassenden Neuentwicklung.

Abstract

Within medical applications the quality of computer software is crucial. For this reason medical software should always be tested for quality. One of the primary features of software-quality is usability, which consists of factors like the product criteria efficiency, effectiveness and satisfaction (Heinsen 2003).

This thesis describes the usability evaluation of a computer software product for licensed doctors and other medical staff. Usability being a property which cannot be easily measured must be tested using specific usability evaluation methods, which are described in the theoretical part of this thesis. Part of the practical work undertaken are an expert evaluation using heuristics from (Norman 1988; Norman 1990; Nielsen 1993; Shneiderman 1998) and a series of interviews using a questionnaire inspired by the IsoMetrics questionnaires by (Willumeit 1996). For the expert evaluation, a collection of heuristics from the authors above mentioned was used. These heuristics were selected due to inherent redundancies and combined for the actual evaluation. This approach is also recommended by (Nielsen 1993).

Object of evaluation was the computer program “orbImmun” supported by the company „Ablinger & Garber Publishing“ in Hall in Tyrol and published together with the Austrian “Ärzttekammer” (General Medical Council (UK) / State Medical Board of Registration (US)). orbImmun is a software for travel vaccination which runs on Microsoft Windows and helps doctors consulting patients about health and travelling. It offers information concerning travels outside Austria and Europe, selection of vaccines and medication and provides prescriptions.

The internal evaluation opens up a wide range of improvements concerning design and ergonomics. Most of these findings are confirmed by the questionnaire, which also revealed additional important aspects. Furthermore quite an amount of new ideas and interesting suggestions has been gathered.

Altogether, there are many findings suggesting correction or improvement. Some of them would be quite difficult to realise on the existing software. The author’s recommendation is to re-develop the software from scratch rather than trying to fix the existing product.

Inhalt

I. EINLEITUNG	6
1. Aufgabenstellung, Abgrenzung	6
2. Aufbau der Arbeit	6
II. EVALUATION VON USABILITY - THEORETISCHE GRUNDLAGEN	7
1. Begriffserklärungen	7
1.1 Grundlegende Begriffe	7
1.2 Produktkriterien: Effizienz, Effektivität und Zufriedenheit	12
1.3 Priorisierung von Usability Problemen	12
2. Motivation	13
2.1 Einleitung	13
2.2 Vorteile von Usability	15
3. Wirtschaftliche Aspekte	16
3.1 Einleitung	16
3.2 Kosten und Nutzen	16
4. Normen und Zertifizierung	19
4.1 Einleitung	19
4.2 DIN EN ISO 9241	20
4.3 Weitere themenverwandte Normen	20
4.4 Zertifizierung	21
5. Expertenevaluation	21
5.1 Einleitung	21
5.2 Methoden	22
5.3 Evaluatoren	24
5.4 Heuristiken	24
5.5 Durchführung	24
5.6 Priorisierung	24
5.7 Ergebnisdarstellung und Empfehlungen	25
6. Fokusgruppen	25
6.1 Einleitung	25
6.2 Durchführung	25
6.3 Analyse und Bericht	26

7. Evaluationsfragebögen	27
7.1 Einleitung	27
7.2 Qualitätskriterien	27
7.3 Durchführung	27
7.4 Auswertung	28
7.5 IsoMetrics.....	29
7.6 AttrakDiff.....	30
 III. FALLBEISPIEL „orbImmun“	32
1. Einleitung.....	32
2. Projektbeschreibung	32
2.1 orbImmun.....	32
2.2 Projekt „orbImmun Redesign“	32
3. Vorgehensweise	33
3.1 Einleitung	33
3.2 Interne Evaluation	33
3.3 Fragenkatalog	39
3.4 Externe Evaluation	40
3.5 Ergebnispräsentation	40
4. Ergebnisse	41
4.1 Einleitung	41
4.2 Ergebnisse der internen Evaluation	41
4.3 Ergebnisse der externen Evaluation	46
4.4 Empfehlungen	49
5. Diskussion	49
 IV. VERZEICHNISSE	50
1. Bibliographie	50
2. Abbildungen	51
3. Formeln und Gleichungen.....	51
4. Tabellen.....	51
 V. INDEX	52
 CURRICULUM VITAE.....	61

I. Einleitung

1. Aufgabenstellung, Abgrenzung

Um Software qualitativ zu beurteilen ist es erforderlich, die Gebrauchstauglichkeit (Usability) der Software zu untersuchen. Zu diesem Zweck existieren eine Reihe von gut beschriebenen Evaluationsmethoden und Heuristiken, die in dieser Arbeit vorgestellt werden. Diese Methoden wurden am Projekts orbImmun angewandt, um die Gebrauchstauglichkeit der Software fundiert zu beurteilen. Dies wird im praktischen Teil der Arbeit beschrieben.

2. Aufbau der Arbeit

Nach Abschnitt I (Einleitung) folgen die zwei Hauptteile der Arbeit, nämlich Abschnitt II: „Evaluation von Usability – Theoretische Grundlagen“ und Abschnitt III: „Fallbeispiel orbImmun“. Die theoretischen Grundlagen (Abschnitt II) lassen sich den Themen „Usability“ und „Usability Evaluation“ zuordnen. Abschnitt II enthält die wichtigsten theoretischen Voraussetzungen, die zur Durchführung einer kleinen Usability Evaluation wie im Fallbeispiel orbImmun (Abschnitt III) benötigt werden. Im folgt das Literatur- und Abbildungsverzeichnis und der Index.

II. Evaluation von Usability - Theoretische Grundlagen

In diesem Abschnitt werden die theoretischen Grundlagen der Themen Usability (Kapitel 1 bis 4) und Usability-Evaluation (Kapitel 5 bis 7) behandelt. Diese wurden beim Projekt orbImmun eingesetzt, um die Software zu bewerten (siehe Abschnitt III). Die Ergebnisse dienen dem Auftraggeber als Entscheidungshilfe im weiteren Projektverlauf.

Im ersten Kapitel werden die grundlegenden Begriffe definiert und erklärt. Die Motivation (Kapitel 2) führt direkt zu den Vorteilen von Usability und den wirtschaftlichen Aspekten (Kapitel 3). Normen und Zertifizierung (Kapitel 4) sind wichtige Grundlagen für die Evaluation von Usability (Kapitel 5 bis 7). In diesen Kapiteln werden Methoden zur Usability Evaluation behandelt.

1. Begriffserklärungen

1.1 Grundlegende Begriffe

1.1.1 Usability

Der Begriff Usability bedeutet ins Deutsche übersetzt „Gebrauchstauglichkeit“, also die Angemessenheit oder Eignung eines Produktes für einen bestimmten Anwendungszweck. Das Produkt kann sowohl materiell als auch immateriell (wie beispielsweise Software) sein. Häufig werden in der Literatur oder im Marketing die Begriffe „benutzerfreundlich“ oder „user friendly“ als Synonyme verwendet. Diese Begriffe sind nicht wirklich zutreffend, da es für ein gebrauchstaugliches Produkt keineswegs genügt, einfach nur „benutzerfreundlich“ zu sein.

In der ISO Norm 9241 in Teil 11 (siehe auch Kapitel 4) wird Usability folgendermaßen definiert:

“The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.“

(Übersetzung: Das Ausmaß in dem ein Produkt von bestimmten Benutzern für das Erreichen bestimmter Ziele mit Effektivität, Effizienz und Zufriedenstellung im Zusammenhang der Anwendung benutzt werden kann.)

Die Definition aus der ISO-Norm macht deutlich, dass Usability keine Eigenschaft des Gegenstandes allein ist, sondern auch immer in Abhängigkeit vom Benutzer, vom Nutzungskontext (gewissermaßen von den Arbeitsbedingungen) und von den Zielen gesehen werden muss (Weers 2000). Es werden in der Norm des weiteren drei Faktoren genannt, aus denen sich Usability zusammensetzt. Diese werden in Abschnitt 1.2 erörtert. Diese Faktoren und somit der Begriff Usability betrifft alle Aspekte eines Produktes, mit denen der Mensch zu tun hat .So schreibt auch (Nielsen 1993):

“Usability applies to all aspects of a system, with which a human might interact, including installation and maintenance procedures.”

(Übers.: Der Begriff Usability lässt sich auf alle Aspekte eines Systems anwenden, mit denen der Benutzer interagieren kann, inklusive Installation und Wartung.)

(Nielsen 1993)

Es erweist sich als sinnvoll „Usability“ wie in der ISO Norm 9241 als Oberbegriff für eine Menge von Attributen zu betrachten, aus denen sich die Gebrauchstauglichkeit eines Produktes zusammensetzt. In der Definition von Usability in (Nielsen 1993) werden folgende Hauptattribute unterschieden:

- **Learnability** (Erlernbarkeit): Ein System sollte leicht erlernbar sein, so dass der Benutzer damit möglichst schnell produktiv arbeiten kann.
- **Efficiency** (Effizienz): Das System sollte effizient arbeiten, damit – sobald der Benutzer den Umgang mit dem System beherrscht – eine möglichst hohe Produktivität gegeben ist.
- **Memorability** (hier: „Einprägsamkeit“): Das System sollte einprägsam sein, so dass ein Benutzer auch nach längerer Pause nicht wieder alles neu lernen muss.
- **Errors** (Fehler): Das System sollte eine geringe Fehlerrate haben, so dass der Benutzer kaum Fehler machen kann, und falls der Benutzer Fehler macht, diese leicht zu beheben sind. „Katastrophale“ Fehler dürfen außerdem nicht vorkommen.
- **Satisfaction** (Zufriedenstellung): Das System sollte „angenehm“ zu benutzen sein, so dass die Benutzer es mögen und damit zufrieden sind.

Wie man sieht, beinhaltet diese Auflistung viele subjektive Kriterien, vor allem was den letzten Punkt „Satisfaction“ betrifft. Daher muss sich jegliche Evaluation der Usability eines Produktes auch stark an den Personen der Zielgruppe des Produktes orientieren. Denn die Praxis zeigt: Selbst wenn theoretisch alle Voraussetzungen für „hohe Usability“ gegeben sind, kann ein Produkt bei den Kunden aus subjektiven Gründen nicht gut ankommen:

„Usability wird sowohl durch objektive als auch durch subjektive Kriterien beeinflusst, daher ist ‚Usability‘ nicht nur in der Informatik, sondern auch in der Psychologie ein Thema. Objektive Kriterien wie Effizienz und Effektivität stehen dem subjektiven Kriterium ‚Zufriedenheit‘ gegenüber und variieren in ihrer Gewichtung je nach Projekt.“

(Sven Heinsen 2003)

Um solche subjektiven Kriterien zu untersuchen ist es sinnvoll, eine möglichst große Menge an Meinungen einzuholen und Methoden der Statistik zu verwenden, um möglichst klare Aussagen zu erhalten. In den Kapiteln 6 (Fokusgruppen) und 7 (Evaluationsfragebögen) werden dazu die gängigsten Methoden zumindest für kleine bis mittlere (Software-) Produkte vorgestellt.

1.1.2 Usability Engineering

Dieser Begriff wird vor allem in der Informatik häufig verwendet und drückt aus, dass es sich um eine praktische Disziplin handelt. Andere bevorzugen auch „User centered Design“ oder „Software Ergonomie“ (v.a. im deutschen Raum). Der Begriff „Usability Engineering“ drückt jedoch am deutlichsten aus, dass es sich dabei um einen Prozess handelt. Außerdem erinnert der Begriff an „Software Engineering“, womit Usability Engineering eng zusammenhängt. Mit Usability Engineering ist also ein Entwicklungsprozess gemeint, im Laufe dessen die Usability eines Produktes verbessert bzw. sichergestellt werden soll.

Diese Disziplin ist im Bereich der Softwareentwicklung noch relativ jung, vermutlich auch deswegen, weil sie erst mit graphischen Betriebssystemen wie Windows® oder MacOS® und graphischen Schnittstellen wie dem „X Window System“, die mit WIMP- Benutzerschnittstellen (**W**indows, **I**cons, **M**aus, **P**ointer) ausgestattet sind, groß wurde. Denn erst mit der steigenden Komplexität der Systeme und somit auch der grafischen User Interfaces („Benutzeroberfläche“, kurz: UI) stieg auch der Bedarf an Usability.

Solange Computer hauptsächlich von Experten über Befehlszeileninterpreter wie DOS® oder UNIX® bedient wurden, welche sich mit „Batch-Dateien“ oder selbst geschriebenen Programmen zu helfen wussten, gab es wenig Bewusstsein für Usability Fragestellungen. Erst durch die große Verbreitung von Computersystemen und der Einführung von graphischen Benutzeroberflächen (in Folge „GUI“) stieg auch das Usability - Problembewusstsein, da sich aufgrund der gestiegenen Komplexität und der zunehmenden Größe der Anwendungen Anwendungsfehler häuften (Thaller 2002) und immer mehr Benutzer durch die Vielzahl von Funktionen und deren Komplexität überfordert wurden (Pürer 2001).

„The same technology that simplifies life by providing more functions in each device also complicates life by making the device harder to learn, harder to use.“

(Übers.: Dieselbe Technologie, die das Leben vereinfacht, indem sie mehr Funktionen für jedes Gerät verfügbar macht, verkompliziert auch das Leben, indem sie die Benutzung des Gerätes schwerer zu erlernen und zu bedienen macht.)

Paradoxon der Technologie, (Norman 1990)

1.1.3 HCI (Human-Computer Interaction)

Synonyme: CHI (computer-human interaction), MMI (man-machine interface), HMI (human-machine interface), OMI (operator-machine interface) und andere.

Übersetzt wird HCI im Deutschen meist mit „Mensch-Maschine Kommunikation“, obwohl „Mensch-Maschine Interaktion“ meiner Meinung nach treffender wäre, da man mit Maschinen - zumindest zum Zeitpunkt, zu dem diese Arbeit verfasst wird - weniger kommuniziert als eben interagiert.

Die Interaktion findet über diverse Ein- und Ausgabegeräte wie beispielsweise Maus, Tastatur, Touchscreen und andere statt. Diese Schnittstellen stellen das Nadelöhr der HCI dar, eine natürliche Grenze, die alle Möglichkeiten der Kommunikation oder Interaktion mehr oder weniger stark limitiert. So kann man mit einer Tastatur nicht so schnell Informationen eingeben, wie durch die natürliche Sprache. Umgekehrt findet auch beim Output von Informationen durch das Programm meist eine Auslese statt, da es dem menschlichen Benutzer im Gegensatz zur Maschine nicht möglich ist, tausende Informationen zugleich aufzunehmen und zu verarbeiten.

Eine Usability-basierte Optimierung der Schnittstellen zwischen Mensch und Maschine verbessert die Interaktion oder Kommunikation, womit bessere Usability erreicht wird.

Wenn man nun mit einer Maschine interagiert, also beispielsweise Daten zur Auswertung in ein Programm eingibt, dann wird die HCI nicht nur durch das GUI und die Eingabegeräte eingeschränkt, sondern auch durch die Fähigkeit der Software Informationen zu interpretieren.

Eine gute Definition der HCI findet man in den „ACM SIGCHI Curricula for Human-Computer Interaction.“ von (Hewett, Baecker et al. 1992):

“Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.”

(Übers.: “Human-computer interaction“ ist eine Disziplin, die sich mit dem Design, der Evaluation, und der Implementierung von interaktiven Computersystemen für menschliche Benutzer und dem Studium der wichtigsten umgebenden Phänomene beschäftigt)

(Hewett, Baecker et al. 1992)

(Ruser 2003) definiert HCI folgendermaßen:

„Mensch-Computer Interaktion beschäftigt sich mit der Analyse, Gestaltung und Evaluation von Informations- und Kommunikationssystemen, wobei der Mensch mit seinen individuellen und sozialen Bedürfnissen im Mittelpunkt der Betrachtung steht.“

Hervorzuheben ist hier der Begriff „Evaluation“, um den es in dieser Arbeit hauptsächlich geht. Die Evaluation von Computersystemen für Menschen gehört zu einem wichtigen Teilaspekt der HCI.

Einen guten Überblick über die Themen der HCI gibt Abbildung 1.1.3:

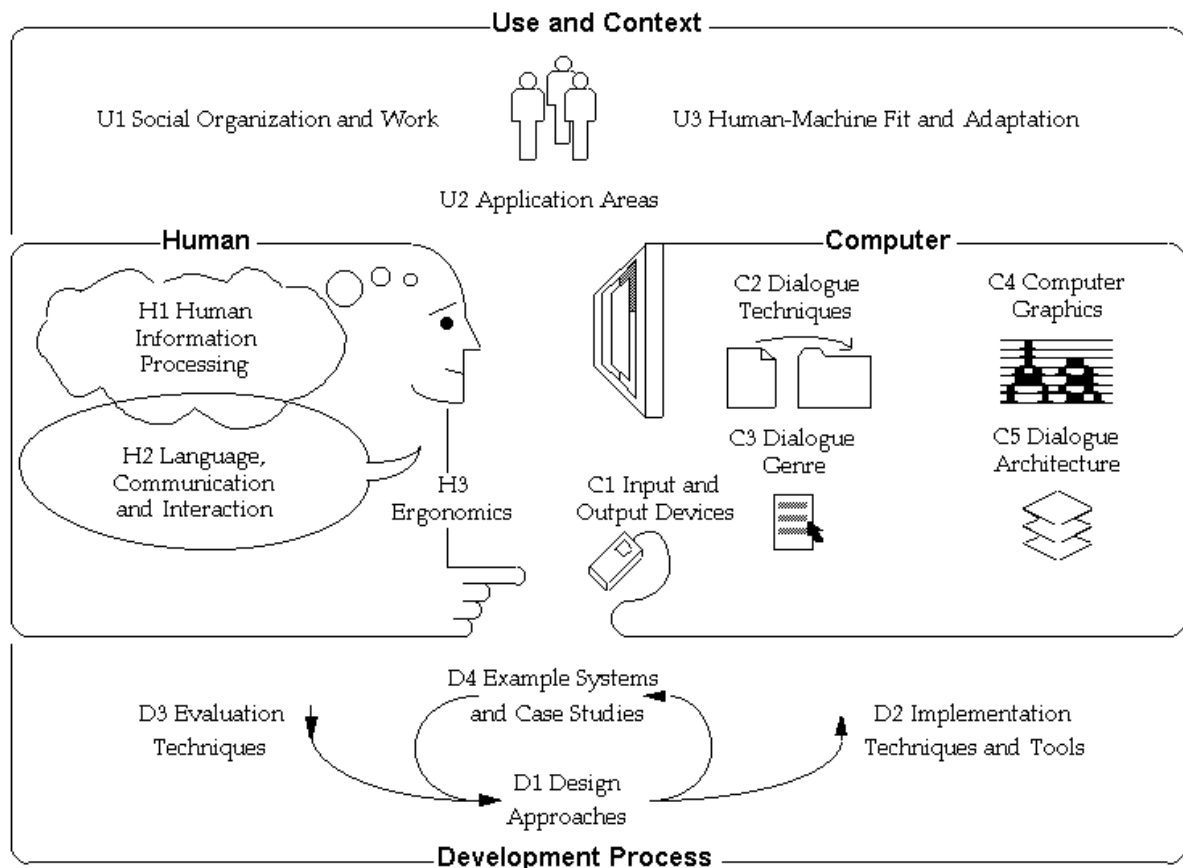


Abbildung 1.1.3, „Nature of HCI“ (Hewett, Baecker et al. 1992)

Legende

Übersetzt aus (Hewett, Baecker et al. 1992)

N	Natur der HCI
	N1 (Meta-)Modelle der HCI
U	Nutzen und Zusammenhang mit Computern
	U1 Mensch – soziale Organisation und Arbeit
	U2 Anwendungsfelder
	U3 Mensch-Machine Fitness und Adaption
H	Menschliche Charakteristiken
	H1 Menschliche Informationsverarbeitung
	H2 Sprache, Kommunikation, Interaktion
	H3 Ergonomie
C	Computersystem und Schnittstellenarchitektur
	C1 Ein- und Ausgabegeräte
	C2 Dialog-Techniken
	C3 Dialog-Gattung
	C4 Computergrafik
	C5 Dialog-Architektur
D	Entwicklungsprozess
	D1 Design - Herangehensweise
	D2 Implementierungstechniken
	D3 Evaluationstechniken
	D4 Beispielsysteme und Fallstudien
P	Projektpräsentationen und Untersuchungen

1.2 Produktkriterien: Effizienz, Effektivität und Zufriedenheit

1.2.1 Effizienz

Im Zusammenhang mit Usability bedeutet Effizienz, dass der User sein Ziel mit möglichst wenig Zeit- und Arbeitsaufwand erreichen kann. Beispielsweise steigert es oft die Effizienz einer Anwendung, wenn bestimmte Daten erfasst, und damit dem User das wiederholte Eingeben von ein und derselben Information erspart wird.

1.2.2 Effektivität

Effektivität heißt, dass das Produkt seine Funktion korrekt erfüllt, also das tut, was man sich davon erwartet.

1.2.3 Zufriedenheit

Mit Zufriedenheit ist gemeint, dass der Benutzer sich mit dem Produkt wohl fühlt, sich damit also zufrieden fühlt („Joy of Use“). Dieses Kriterium ist natürlich sehr subjektiv und schwer vorherzusagen. Emotionale Aspekte kommen hier ins Spiel, die oft nicht nur durch das Produkt selbst, sondern auch durch ganz andere mit dem Produkt irgendwie zusammenhängende Faktoren beeinflusst werden. So wird ein Arbeiter auch mit einer perfekten Software nicht zufrieden sein, wenn er durch diese Software beispielsweise seinen Arbeitsplatz gefährdet sieht. Aber nicht immer sind derartige Zusammenhänge so offensichtlich und daher ist die Messung des Faktors „Zufriedenheit“ sehr schwierig.

„Wenn die drei Kriterien Effektivität, Effizienz und Zufriedenheit für einen bestimmten Nutzungskontext in einer Software, Produkt oder Webseite erfüllt sind, ist es ein benutzerfreundliches Produkt. Je nach Interessen und Zielen werden die einzelnen Kriterien verschieden gewichtet sein.“
(Sven Heinsen 2003)

1.3 Priorisierung von Usability Problemen

Nach (Nielsen 2003) besteht die Schwere eines Usability-Problems aus einer Kombination folgender Faktoren:

- Frequenz: Wie oft taucht das Problem auf?
- Wirkung: Wie schwer ist das Problem für den Benutzer zu lösen?
- Hartnäckigkeit (persistence): Lässt sich das Problem dauerhaft lösen, oder taucht es immer wieder auf?

(Nielsen 2003)

Zusätzlich unterstreicht Nielsen die Bedeutung der Wirkung auf den Markt („market impact“) und betont, dass auch Probleme, die objektiv gesehen harmlos sind, sich trotzdem sehr stark ungünstig auswirken können.

Nielsen gibt in (Nielsen 2003) fünf Stufen für die Schwere eines Usability Problems an:

0. Kein Problem
1. Das Problem ist nur kosmetischer Natur: Korrektur nicht unbedingt nötig
2. Minderschweres Problem: Korrektur hat niedrige Priorität
3. Schweres Problem: Korrektur hat hohe Priorität
4. Usability Katastrophe: Korrektur muss unbedingt noch vor dem Release des Produktes erfolgen.

(Nielsen 2003)

2. Motivation

2.1 Einleitung

Von vielen Leuten wird Usability leider noch als Randthema der Informatik oder anderer Gebiete betrachtet. Die Wichtigkeit von Usability wird oft unterschätzt und es gibt eine Vielzahl von Missverständnissen bezüglich ihrer Rolle im Softwareentwicklungsprozess. Eine oft zitierte Aussage des Managements ist: „Das Produkt muss nicht evaluiert werden, es hat sich bisher noch kein Kunde beschwert.“ (Heinsen 2003)

Dies ist ein gefährlicher Trugschluss, da es vielen Benutzern zu aufwendig ist, sich über eine Usability-Problematik zu beschweren, die ihnen selbst vielleicht sowieso nebensächlich vorkommt.¹ Dies wurde auch beim Fallbeispiel „orbImmun“ (Kapitel III) festgestellt.

Eine weitere Aussage, die auf Missverständnissen zum Thema Usability beruht, lautet: „Wir brauchen keinen Usability Spezialisten, das machen unsere Entwickler sowieso schon“. Die Entwickler sind meist sogar indirekt daran beteiligt, dass Usability Probleme entstehen. Da Software Entwickler oder Programmierer in der Regel sehr fortgeschritten im Umgang mit Computern sind, benötigen sie viele der Mechanismen nicht, die weniger erfahrenen Benutzern helfen sollen. Die meisten Programme verfügen heutzutage über eine Menüsteuerung, deren Elemente wahlweise mit einem Zeigegerät (Maus) oder durch Tastenkombinationen gewählt werden können. Meist ist man mit der Tastatur schneller, vor allem, wenn man sich mit den Händen sowieso schon dort befindet, wie zum Beispiel bei Textverarbeitungsprogrammen. Aber diese Art der Menüsteuerung setzt Wissen über die richtigen Tastenkombinationen voraus und darf bei einem neuen Benutzer nicht vorausgesetzt werden.

¹ Daher kann es auch Aufgabe der Usability sein, Feedback - Mechanismen zu implementieren. Der neueste Microsoft® Media Player sowie das Microsoft® Betriebssystem Windows XP ® beinhalten bereits einen Mechanismus, der bei schweren Fehlern oder Abstürzen für die Übermittlung der relevanten Daten sorgt. Die elektronische „Beschwerde“ bzw. ein „Fehlerbericht“ ist also nur einen Mausklick entfernt.

Außerdem weiß der Programmierer ja schon während seiner Arbeit sehr genau, wie die Software funktioniert bzw. funktionieren soll. Daher wird er sich nur schwer in die Lage eines Benutzers versetzen können, der noch gar nichts über das Programm weiß. Auch in (Heinsen 2003) findet sich dazu eine klare Aussage:

„Eine Evaluation mit echten Benutzern zählt jedoch typischerweise weder zu ihrem [Anm.: Entwickler] Aufgabengebiet noch zu ihren Fähigkeiten. Entwickler und Designer haben keine „Außensicht“. Sie wissen bereits, wie das Produkt funktionieren soll, und sind deshalb in der Regel kaum in der Lage, sich in Nutzer hinein zu versetzen, die weder ihre umfangreichen IT-Kenntnisse haben noch sich mit dem jeweiligen Produkt schon auskennen. Ein externes oder zumindest unabhängiges Usability-Testing garantiert hier einen wesentlich objektiveren und kritischeren Blick. Dabei ist es wichtig, Tests mit echten Endnutzern durchzuführen.“

[...]

„Eine Zielgruppen-Definition und die Einbindung von Personen, die diesem Kreis entsprechen, ist für ein gutes Produkt unersetzlich.“

(Heinsen 2003)

Usability ist also kein Luxus oder ein „nice-to-have“ Feature, um das man sich als Entwickler nicht kümmern muss. Genauso wenig kann man Usability als Entwickler ohne Hilfestellung umsetzen. Da hohe Usability heutzutage als selbstverständlich vorausgesetzt wird und jeder Konsument bzw. Auftraggeber von einer neuen Software Gebrauchstauglichkeit von vorne herein erwartet, muss man sich vor jeder Softwareentwicklung (und auch bei anderen Produkten, bei denen Usability eine Rolle spielt) Gedanken darüber machen und dem Thema eine entsprechende Rolle im Software-Entwicklungsprozess zukommen lassen. Dies bestätigt auch Heinsen:

„Benutzerfreundlichkeit wird immer wichtiger und eine Missachtung kann zu einem entscheidenden Nachteil im Wettbewerb führen.“

(Heinsen 2003)

Genau wie ihre Auftraggeber und Vorgesetzten denken auch viele Entwickler und Tester, dass sie „das mit der Usability“ schon selbst im Griff hätten. Dabei machen sie oft den Fehler, sich selbst als typischen User zu sehen und sich beim Design des Produktes an ihren eigenen Bedürfnissen zu orientieren. Die Zielgruppe eines Software Produktes verfügt jedoch nur selten über das Wissen eines Programmierers und hat auch sonst oft eine völlig andere Herangehensweise an das Produkt (Heinsen 2003).

2.2 Vorteile von Usability

Gemäß (Sven Heinsen 2003) bietet die Integration von Usability in den Produktentwicklungsprozess folgende Vorteile:

- Usability schafft Erfolgssicherheit und Planbarkeit
 - Ein Usability orientierter Entwicklungsprozess deckt Risiken früher auf und verhindert so eine aufwändige Korrektur nach der Veröffentlichung.
- Usability spart Zeit
 - Usability Engineering hilft zwischen möglichen Features eines Produktes oder einer Webseite zu priorisieren
- Usability spart Geld
 - Jeder Euro, der in Usability investiert wird, spart 10 bis 100 Euro (IBM 2003)
 - Usability kann direkt zu Umsätzen führen
 - Speziell bei e-commerce C2C-Webseiten: je leichter man bestellen kann, umso mehr wird bestellt.
- Usability bringt neue Kunden
 - Usability orientierte Websites wie Amazon oder eBay: verkaufsfördernder Effekt
- Usability erhöht Marktwert und Loyalität
 - Usability orientierte Produkte führen zu einem höheren Vertrauen zum Anbieter.
- Usability schafft Vermarktungs-Vorteile
 - Usability ist ein wichtiges Kriterium für die Bewertung von Software und Websites. Zahlreiche Computerzeitschriften befassen sich mit Produkttests.
- Usability ist ein Qualitätsmerkmal

(Heinsen 2003)

Weiters bringt Usability auch wirtschaftliche Vorteile, die im folgenden Kapitel erläutert werden.

3. Wirtschaftliche Aspekte

3.1 Einleitung

Dieses Kapitel soll zeigen, dass sich Usability wirtschaftlich bezahlt macht und erklären, wie dieser wirtschaftliche Vorteil zustande kommt. Auf buchhalterische Methoden oder andere wirtschaftswissenschaftlichen Details wird nicht eingegangen, da dies den Rahmen der Arbeit sprengen würde.

3.2 Kosten und Nutzen

3.2.1 ARPU vs. AMPU

Eine der wichtigsten Messgrößen betreffend den wirtschaftlichen Erfolg einer gewinnorientierten Anwendung (darunter fallen auch Webseiten wie zum Beispiel Onlineshops) ist „ARPU“. Die Abkürzung steht für „average revenue per user“, was soviel bedeutet wie „durchschnittliche Einkunft pro Benutzer“. Diese Zahl gibt an, wie viel Gewinn ein Besucher im Durchschnitt einbringt. Dabei werden jedoch nicht die Kosten berücksichtigt, die pro Benutzer anfallen, wie zum Beispiel Kosten für Rechnungsstellung, Betreuung und andere. Daher muss man bei der Interpretation dieser Messgröße vorsichtig sein, ein niedriger ARPU muss nicht unbedingt schlecht sein. Speziell wenn es um Dienstleistungen oder Produkte geht, die im Vorhinein bezahlt werden, die Kosten auf Unternehmensseite also gering sind, kann der ARPU missverständlich sein. Dasselbe gilt verstärkt auch für den „mobile market“, da dort die Kauffrequenz der Kunden deutlich höher ist. Ein gutes Beispiel dafür ist der so genannte „short messaging service“ oder kurz „SMS“.

In (Global Information 2003) wird dargelegt, dass speziell für mobile operators „AMPU“ statt „ARPU“ verwendet werden sollte:

“Average Margin per User (AMPU) is the difference between the cost of serving a user and the revenue that user generates. AMPU can be negative or positive. The greater the AMPU, the greater the profit. The key to increasing AMPU comes from understanding the full costs of providing specific offerings.”
(Übers.: Die durchschnittliche Gewinnmarge pro Benutzer / Kunde ist die Differenz zwischen den Kosten, die durch die Bedienung des Benutzers / Kunden entstehen, und den Einnahmen, die der Benutzer / Kunde bringt. Dieser Wert kann sowohl negativ als auch positiv sein. Je höher der Wert, umso größer der Profit. Der Schlüssel dazu, wie man den Wert erhöhen kann, liegt im Verständnis der vollständigen Kosten für das Stellen von spezifischen Angeboten.)

(Global Information 2003)

Im vergangenen Jahrzehnt beklagte die Wirtschaft ständig sinkende ARPU (Global Information 2003), dafür stieg in einzelnen Geschäftsfeldern – ganz besonders im mobile market – die Frequenz der Geschäfte enorm. Für den Kunden bedeutet dies: mehr kaufen in weniger Zeit. Daher werden Kunden auch steigend Wert auf komfortable und gebrauchstaugliche Geschäftsabwicklungen legen, also auf Usability. Somit haben Unternehmen, die Usability berücksichtigen, einen Wettbewerbsvorteil.

3.2.2 Weitere wirtschaftliche Faktoren

Neben dem erwähnten Wettbewerbsvorteil gibt es eine Vielzahl anderer positiver Faktoren, die durch gezielten Einsatz von Usability optimiert werden können. Im folgenden nur ein kurzer Überblick über die wichtigsten Faktoren:

- Entwicklung und Implementierung:
 - Je eher eine Änderung im Entwicklungsprozess vorgenommen wird, desto kostengünstiger ist sie. Da ein benutzerorientierter Produktentwicklungsprozess die Entwicklung schon früh in die richtige Richtung lenkt, entsteht hier eine signifikante Kostensenkung (Heinsen 2003)
 - Daumenregel 1:6:10 (Landauer 1996)
- Pflege und Wartung
 - Usability Maßnahmen während der Entwicklung reduzieren den langfristig anfallenden Pflege- und Wartungsaufwand effektiv (Heinsen 2003)
- Support
 - Ein Produkt, das die Bedürfnisse seiner Zielgruppe erfüllt und für diese gut zu bedienen ist, wird naturgemäß weniger Support-Aufwand verursachen. Vor allem, wenn der Kundenservice durch eine persönliche Beratung wie z.B. in einem Call Center geleistet wird, zahlt sich ein geringeres Anfragevolumen schnell aus (Heinsen 2003).
- Sonstige Ausgaben
 - Einsparung von Weiterbildungsmaßnahmen: eine Software, die schneller verstanden werden kann, hat natürlich eine schnellere Einarbeitungszeit und erfordert damit weniger Weiterbildung (Heinsen 2003).
 - Weiteres Einsparungspotential liegt z.B. bei externen Beratungsaufwänden, Reisekosten, Kommunikationskosten (Heinsen 2003)

Ein großes Potential liegt des weiteren in der Steigerung der Produktivität durch verbesserte Usability. So schätzt Jakob Nielsen in (Nielsen 2002), dass mittelständische Unternehmen durchschnittlicher Größe durch Usability-Verbesserungen ihrer Intranets etwa 5 Millionen US Dollar an Produktivität gewinnen könnten und der ROI (siehe Abschnitt 3.2.3) über 1000% liegen würde.

Insgesamt sind Usability verbesserte Produkte erfahrungsgemäß zum Verkaufszeitpunkt ausgereifter und weniger fehleranfällig, daher weisen sie auch eine gesteigerte Wettbewerbsfähigkeit vor allem in Märkten mit sehr ausgereiften Produkten auf.

Eine Daumenregel des Marketing besagt, dass ein verlorener Kunde zehn weitere davon abhält das Produkt zu kaufen. Sinngemäß heißt es auch bei (Albers 2003):

„Nicht zuletzt ist jeder verlorene Kunde ein potenzieller Multiplikator für die Verbreitung von schlechter Stimmung im Kunden- und Interessentenstamm.“

(Albers 2003)

Fest steht, dass sich Effekte wie Mundpropaganda - vor allem in einem so schnellen Medium wie dem Internet – besonders stark auswirken.

3.2.3. TCO – Total Cost of Ownership

TCO steht für „Total Cost of Ownership“ (Gesamtbetriebskosten), ein Netto-Ansatz, der vor allem dazu verwendet wird Kosten längerfristig abschätzen zu können. Es handelt sich dabei um einen interessanten Kandidaten für einen Usability Business Case, da Usability-Projekte zumeist vergleichsweise geringe Investitionsvolumina besitzen und sich nicht über einen längeren Zeitraum als drei bis sechs Monate erstrecken (Heinsen 2003).

Ein Nachteil der Methode ist allerdings die Tatsache, dass der (finanzielle) Nutzen einer Investition dabei unberücksichtigt bleibt. Dadurch ist TCO am ehesten zur Kostenoptimierung geeignet.

TCO wird folgendermaßen berechnet:

$TCO_{Abs.} = \sum \text{Kosten}_{Investition} - \sum \text{Kosten}_{Alternative}$	
$TCO_{Abs.}$:	„Gesamt-Betriebskosten“ eines Produkts bzw. eines kostenpflichtigen Service. (Kosten / Zeitraum)
$\sum \text{Kosten}_{Investition}$:	Summe aller Kosten einer im Zusammenhang mit der Optimierung eines Produktes bzw. kostenpflichtigen Service geplante Investition pro Zeitraum.
$\sum \text{Kosten}_{Alternative}$:	Summe aller Kosten einer im Zusammenhang mit der Optimierung eines Produktes bzw. eines kostenpflichtigen Service geplanten alternativen Investition

Gleichung 1, „TCO“ (Heinsen 2003)

Im letzten Punkt ist zu beachten, dass im Sinne des TCO auch die Beibehaltung des Status quo, also [...] die schlichte Unterlassung der Usability-optimierenden Maßnahmen, eine Kostenalternative darstellt. Das Projektvorhaben wird dann positiv bewertet, wenn die Gesamtkosten der Alternative die der geplanten Investition übersteigen, was sich bei der vorgeführten Absolutkalkulation in einem negativen (!) TCO-Wert ausdrückt (Heinsen 2003).

3.2.4 ROI – Return on Investment

Mit dem ROI - auf Deutsch: „Rendite einer Investition“ – wird versucht, den Gesamtertrag einer Investition abzuschätzen, das heißt Gewinne und Kosten werden gegeneinander abgewogen. Die Vorgehensweise der Kalkulation ist jedoch leider nicht standardisiert. Um den ROI möglichst detailreich darzustellen, gibt es Formeln wie beispielsweise bei (Doost 1992), die mit 6 oder mehr Faktoren nicht gerade einfach sind.

Einfacher ist dagegen die so genannte „**Dupont Formel**“:

$$\text{ROI} = \frac{\sum \text{Einkünfte}}{\sum \text{Kosten}}$$

Gleichung 2: Dupont Formel

$$\text{ROI} = \left(\frac{\sum \text{Einkünfte}}{\sum \text{Kosten}} * 100 \right) - 100$$

Gleichung 3: ROI in Prozent

Mit der zweiten Formel lässt sich ausrechnen, wie viel Prozent der investierten Summe nach einer bestimmten Zeit erwirtschaftet werden kann.

Beispiel: Ein kleines Softwareprojekt. (stark vereinfacht)

Angenommen die Startkosten betragen 10.000 Euro und der Support und die Weiterentwicklung (laufende Kosten) kostet 1000 Euro im Monat. Weiters nehmen wir an, dass der Auftraggeber im ersten Jahr nur 10.000, im zweiten Jahr 60.000 und im dritten Jahr 120.000 Euro verdient. Dann beträgt der ROI nach der zweiten Formel im ersten Jahr ca. -54%, was einem Verlust entspricht, im zweiten Jahr schon ca. 106%, womit dann die Kosten gedeckt wären und im dritten Jahr 313%, ein deutlicher Gewinn.

4. Normen und Zertifizierung

4.1 Einleitung

Normen werden oft als Werbung, genauso aber auch als Vertragsbestandteile oder für Gesetzestexte herangezogen. Solange es nicht gesetzlich vorgeschrieben ist und auch vertraglich nichts anderes vereinbart wurde, ist die Einhaltung von Normen seitens der Hersteller eines Produktes jedoch freiwillig.

Schon seit den 80er Jahren werden Normen für Benutzerschnittstellen interaktiver Software entwickelt. Eine der wichtigsten Normen in Zusammenhang mit Usability ist die DIN EN ISO 9241, die ergonomische Anforderungen bei computerunterstützter Büroarbeit beschreibt sowie die EU-Richtlinie 90/270/EWG.

DIN steht für „Deutsches Institut für Normung“, CEN (hier kurz EN) für „Comité Européen de Normalisation“ und ISO für „International Standards Organisation“, drei verschiedene wichtige Normungsinstitute. Im Falle der DIN EN ISO 9241 wurde die Norm von allen drei erwähnten Instituten anerkannt und wortgleich übernommen.

Leider werden Normen meist nur eingehalten, wenn dies vertraglich vereinbart wurde, da die Berücksichtigung von Normen einen zusätzlichen Aufwand und Kosten verursacht. Allein der Download von einem der 17 Teile der erwähnten Norm von der ISO Homepage (<http://www.iso.ch>) kostet über 100 Euro. Für kleine Softwareprojekte ist dies daher nur selten rentabel. Dennoch hat es viele Vorteile, Normen zu implementieren, unter anderem, weil man so über eine Überprüfungsmöglichkeit verfügt, mit der gezeigt werden kann, ob und in welchem Ausmaß das betrachtete Produkt den gängigen Standards entspricht.

4.2 DIN EN ISO 9241

Diese Norm beschäftigt sich vor allem mit Bürotätigkeiten am PC-Arbeitsplatz, was insofern verständlich ist, da es sich hier um einen gesetzlich geregelten Bereich handelt. Der Titel dieser Norm lautet “Ergonomic requirements for office work with visual display terminals (VDTs)” (Zu Deutsch: “Ergonomische Anforderungen für Büroarbeit mit VDT’s”).

Die für diese Arbeit wohl am meisten relevanten Teile der Norm (es gibt 17 Abschnitte) sind Teil 11 mit dem Titel “Guidance on Usability“, der als Grundlage für die anderen Teile dient und Teil 10, der sieben wichtige Designgrundsätze enthält. Einen Überblick über die Teile der Norm, die für Software relevant sind, gibt die folgende Tabelle 4.1 von (Weers 2000).

Tabelle 1 „DIN EN ISO 9241“ nach (Weers 2000)

Teil	Titel
10	Dialogue Principles
11	Guidance on Usability
12	Presentation of information
13	User Guidance
14	Menu dialogues
15	Command dialogues
16	Direct manipulation dialogues
17	Form filling dialogues

4.3 Weitere themenverwandte Normen

Weitere Normen, welche die Usability von Software betreffen sind:

- Die EU-Bildschirmrichtlinie **EU 90/270 EWG** (in Deutschland unter der Bezeichnung „**Bildschirmarbeitsverordnung**“ umgesetzt) hat eine europaweite gesetzliche Initiative zur Sicherung der Gebrauchstauglichkeit von Anwendungssoftware angestoßen (Heinsen 2003) .
- Die **DIN ISO/IEC 12119** Norm ist mit der DIN EN ISO 9241 eng verbunden und beschreibt einen Prüfstand für Softwarequalität, wobei vor allem Mindeststandards für marktgängige Produkte definiert werden.
- Die **ISO 14915** Norm behandelt vor allem Gestaltungsprinzipien für multimediale Inhalte.
- Die **ISO 13407** Norm behandelt den Usability-Engineering Prozess für interaktive Software.
- Für den Bereich Icons und Symbole ist speziell die Norm **ISO/IEC 11581** von Bedeutung.

Abschließend gilt noch festzuhalten, dass es speziell bei Softwareprojekten von enormer Wichtigkeit ist, relevante Normen von Beginn an zu berücksichtigen. Die Gründe dafür liegen in den Besonderheiten der Software-Entwicklung: Je später in einem Softwareprojekt eine Änderung vorgenommen wird, umso aufwendiger – und somit teurer – ist sie (Landauer 1996; Heinsen 2003).

4.4 Zertifizierung

Die im vorigen Kapitel aufgeführten Normen enthalten wenige Hinweise dazu, wie man Software auf ihre Konformität prüfen kann (Heinsen 2003). Die Deutsche Akkreditierungsstelle Technik e.V. (kurz: DATech) hat daher ein Prüf- und Zertifizierungsschema für die Gebrauchstauglichkeit von Software entwickelt. Wenn ein Prüflabor oder eine Zertifizierungsstelle die Anforderungen der DATech und der DIN 17025 erfüllt, kann sie durch die DATech akkreditiert werden.

Eine Zertifizierung besteht in einer unabhängigen Bestätigung der Prüfung durch eine nach DIN 45011 unabhängig vom Prüflabor arbeitende Zertifizierungsstelle. Ein Zertifikat wird vergeben, wenn der Zertifizierer den Bericht der Prüfstelle für stimmig und nachvollziehbar befindet. Dieses Zertifikat kann auch zeitlich begrenzt sein oder an eine bestimmte Version der Software gebunden sein. Die Zertifizierungsstelle überwacht dann während der Gültigkeitsdauer des Zertifikates, ob die Voraussetzungen gegeben sind und geht eventuellen Beschwerden nach (Heinsen 2003).

Für die Usability-Zertifizierung von Websites gibt es derzeit noch keinen einheitlichen Standard, was man an der großen Anzahl unterschiedlicher Zertifikate und Prüfaussagen im Web sehen kann.

5. Expertenevaluation

5.1 Einleitung

Dieses Kapitel beschäftigt sich mit der Evaluation von User Interfaces durch Experten. Es werden Methoden und Heuristiken für die Usability-Bewertung von Software vorgestellt, sowie die Ergebnisdarstellung behandelt.

(Nielsen 1993) unterscheidet Usability Testing (repräsentative User führen vorgegebene Aufgaben mit der Software durch) von Usability-Inspektion durch Experten, welche ein User Interface nach ihrer Erfahrung und mit Hilfe vorgegebener Richtlinien untersuchen. Beide Methoden haben folgende Vor- und Nachteile:

Tabelle 2 "Expertenevaluation" nach (Heinsen 2003)

Inspektionsmethoden z.B. heuristische Evaluation	Empirische Verfahren z.B. Usability Tests
Begutachter beurteilen ein Interface basierend auf ihren Erfahrungen und Richtlinien	Repräsentative Nutzer beurteilen 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 aufwendig sein
Auch gut für die Beurteilung nicht-funktionaler Prototypen geeignet	Ideal für Untersuchungen funktional vollständiger Systeme
Keine Erhebung quantitativer Maße	Geeignet zur Erhebung von quantitativen Maßen (Fehlerrate, Zeitdauer, subjektive Zufriedenheit)

Anzumerken gilt außerdem, dass eine Expertenevaluation sich immer auf verschiedene Gestaltungsrichtlinien (Heuristiken) stützen muss. Im Gegensatz zu klassischer Software (wie beispielsweise Windows Anwendungen) gibt es für neuere Anwendungsgebiete (z.B. Mobile Applications u.a.) noch sehr wenige Erfahrungswerte, was bei der Auswahl und Anpassung eines Evaluationsverfahrens berücksichtigt werden sollte.

Wenn ein Auftraggeber eine Aussage über die Qualität eines bestehenden Produktes benötigt, sind aus Akzeptanzgründen Usability-Tests den Inspektionsmethoden vorzuziehen, wenn die erforderlichen Aufwände vom Auftraggeber übernommen werden (Heinsen 2003).

5.2 Methoden

5.2.1 Cognitive Walkthrough

Bei dieser Methode nach (Lewis 1990) führen die Evaluatoren vorgegebene, korrekte Handlungsabläufe an dem System durch. Für jeden Handlungsschritt werden vier Aspekte beurteilt, die eine erfolgreiche

Aufgabenverarbeitung durch den Nutzer verhindern können:

- Der Nutzer versucht nicht, den richtigen Effekt zu erzielen.
- Der Nutzer erkennt nicht, dass die korrekte Aktion zur Verfügung steht.
- Der Nutzer stellt keine Verbindung zwischen der korrekten Aktion und dem gewünschten Effekt her.
- Der Nutzer erhält keine Rückmeldung über seine (erfolgreiche) Aktion.

Insgesamt konzentriert sich der Cognitive Walkthrough auf die leichte Erlernbarkeit einer Anwendung, das heißt wie weit ein Nutzer eine Anwendung durch Exploration erlernen kann (Heinsen 2003).

5.2.2 Formale Usability-Inspektionen

Formale Inspektionen nach (Kahn 1994) stellen eine sehr gründliche Methode dar, ein Interface zu bewerten.

Der Nachteil dieser Methode liegt im organisatorischen und zeitlichen Aufwand, da die Methode ein Team von Moderatoren und Inspektoren sowie die Vorbereitung der Inspektionsteilnehmer voraussetzt. Daher kann diese Methode am sinnvollsten bei einer längerfristigen Produktentwicklung eingesetzt werden (Heinsen 2003).

Auch diese Methode orientiert sich an vorgegebenen Aufgaben oder Szenarien, zusätzlich werden hier aber auch noch Nutzerprofile berücksichtigt. Dementsprechend kommen aufgabenbezogene sowie nutzerspezifische Heuristiken zum Einsatz.

5.2.3 Heuristische Evaluation

Die heuristische Evaluation wurde vor allem von Jakob Nielsen entwickelt und hat als oberstes Ziel die maximale Effizienz bei der Begutachtung eines Interface (Nielsen 1990; Nielsen 1994). Die Begutachtung erfolgt in Form einer freien Exploration, das heißt die Evaluatoren bewegen sich frei durch das zu untersuchende Interface. In einem ersten Durchgang verschaffen sich die Evaluatoren einen generellen Überblick über das System. Im zweiten Durchgang wird dann besonderes Augenmerk auf Interaktionselemente wie zum Beispiel Navigation gelegt.

In beiden Durchgängen wird das Interface mit Hilfe von Usability-Heuristiken überprüft und Verletzungen dieser Gestaltungsprinzipien werden dokumentiert. Am Ende werden die einzelnen Beobachtungen und Probleme bezüglich Häufigkeit und Auswirkungen beurteilt.

Die Methode ist sehr schnell anzuwenden, laut (Nielsen 1994) lassen sich schon mit einer geringen Menge von Evaluatoren in einer typischen Evaluationsdauer von ca. 2 Stunden die meisten Usability Probleme identifizieren. Sven Heinsen nennt in (Heinsen 2003) jedoch auch folgende Nachteile der Methode:

- Durch das explorative, nicht aufgabenorientierte Vorgehen ist es gerade bei größeren Systemen schwierig, die Begutachtung aller relevanten Bereiche sicherzustellen.
- Da auf die Definition von Nutzerklassen verzichtet wird, ist die Anwendung verschiedener Heuristiken problematisch. Beispielsweise ist es nicht einfach zu beurteilen, ob das System „die Sprache des Benutzers spricht“, wie es eine gängige Usability Heuristik verlangt, wenn der Benutzer nicht definiert wurde.
- Aus dieser Methode werden keine direkten Gestaltungsvorschläge abgeleitet (Nielsen 1994). Gerade diese bieten aber den Mehrwert für einen Auftraggeber, der die Bewertung seines Produktes wünscht.

Aus diesen Gründen empfiehlt Heinsen die „klassische“ heuristische Evaluation vor allem für Projekte, die sich noch in der Entwicklung befinden, und die meisten Ergebnisse in das eigene Entwicklungsteam zurückfließen. Für die Bewertung umfangreicher bestehender Systeme stellt Heinsen als Erweiterung der heuristischen Evaluation den „Heuristischen Walkthrough“ vor.

5.2.4 Heuristischer Walkthrough (Heinsen 2003)

Der heuristische Walkthrough erweitert die „klassische“ heuristische Evaluation vor allem in folgenden zentralen Punkten:

- Einbeziehung von Nutzereigenschaften
- Aufgaben- / Szenarienbasiertes Vorgehen
- Ableitung von Gestaltungsempfehlungen

Die Evaluation wird in fünf Phasen durchgeführt:

1. Festlegung des Untersuchungsgegenstandes
 - a. Was wird evaluiert
 - b. Wer sind die Nutzer
 - c. Welchen Umfang soll die Evaluation haben
 - d. Welche Aufgaben / Szenarien werden bearbeitet
 - e. Für welche Zielgruppe sollen die Ergebnisse aufgearbeitet werden
2. Auswahl der Evaluatoren (Anzahl, Qualifikation)
3. Festlegung der benutzten Heuristiken
4. Durchführung der Begutachtung
5. Aufbereitung und Präsentation der Ergebnisse

5.3 Evaluatoren

Bei der Auswahl der Evaluatoren sind verschiedene Aspekte zu berücksichtigen. Zum einen muss eine sinnvolle und mit dem vorhandenen Budget finanzierbare Anzahl bestimmt werden. Es gibt einen direkten Zusammenhang zwischen der Anzahl der Evaluatoren und der Anzahl an identifizierten Problemen (Nielsen 1994), wobei ein einzelner Evaluator lediglich ca. 35% der Fehler in einem Interface findet (Nielsen 1994).

Unter Umständen kann auch der Einsatz von so genannten „Doppelexperten“ vonnöten sein, vor allem wenn die zu untersuchende Software ein bestimmtes Fachwissen oder Fachvokabular voraussetzt.

Des Weiteren muss man – je nach Situation – auch entscheiden, ob man interne oder externe Evaluatoren heranziehen will. (Heinsen 2003) empfiehlt mindestens einen externen Evaluator pro Team und nennt folgende Nachteile bei rein internen Evaluationen:

- Interne Evaluatoren sind nicht unvoreingenommen
- Sie müssen unter Umständen eigene Entscheidungen in Frage stellen
- Es gibt Akzeptanzproblem beim Auftraggeber

5.4 Heuristiken

Heuristiken bilden die Grundlage einer Usability Evaluation. Benötigt werden in diesem Fall vor allem Gestaltungsrichtlinien, die allgemeingültige Anforderungen an gebrauchstaugliche Software stellen. Es gibt eine Vielzahl von solchen Heuristiken vor allem bei einschlägigen Autoren (siehe Teil III Kapitel 3.2.1 bis 3.2.4) sowie innerhalb der bereits vorgestellten Normen (siehe Kapitel 4). Zusätzlich existieren für einige Betriebssysteme so genannte Styleguides, wie zum Beispiel „The Microsoft Windows User Experience“ für Microsoft oder „Macintosh Human Interface Guidelines“ von Apple.

(Nielsen 1994) empfiehlt, die vorgegebenen Heuristiken bei Bedarf einfach zu erweitern.

5.5 Durchführung

Kleine, übersichtliche Software kann meist recht unkompliziert durch freie Exploration evaluiert werden. Bei großen oder komplexen Anwendungen stößt diese Vorgehensweise jedoch recht schnell an ihre Grenzen. Daher wird bei solchen Projekten eher eine aufgabenorientierte Vorgehensweise gewählt. Dabei folgen die Evaluatoren einer vom Auftraggeber definierten Aufgabenstellung. (Heinsen 2003) nennt dazu folgende Vorteile:

- Evaluatoren müssen nicht alle einzelnen Seiten oder Dialoge untersuchen
- Die wichtigsten Bereiche kommen dennoch zur Untersuchung
- Abfolgen von Seiten / Dialogen / Bereichen werden sichtbar

5.6 Priorisierung

Nachdem die Findings der Evaluation vorliegen, werden diese konsolidiert und priorisiert. Zuerst müssen die Findings zusammengefasst werden, sofern mehrere unabhängige Evaluatoren beteiligt waren. Im Anschluss wird versucht, die Dringlichkeit der einzelnen Probleme einzuschätzen.

Dabei wird das Problem nach den unter Abschnitt II, Kapitel 1.3 beschriebenen Aspekten beurteilt. Da es nicht immer praktikabel ist, alle Aspekte eines Problems einzeln zu bewerten, wird meist eine kombinierte Einschätzungsskala verwendet (siehe Abschnitt II Kapitel 1.3 sowie Nielsen 1994).

5.7 Ergebnisdarstellung und Empfehlungen

Nach Ende der Evaluation liegen die Ergebnisse als Liste von konsolidierten und priorisierten Findings vor. Um die Übersicht und die Verwertbarkeit der Ergebnisse insbesondere für den Auftraggeber zu erhöhen, sind noch weitere Verarbeitungsschritte nötig: Zum einen die Aggregation allgemeiner Problemfelder und zum anderen die Ableitung von Verbesserungsvorschlägen.

Wenn möglich sollte die Ableitung von Verbesserungsvorschlägen in Zusammenarbeit mit dem Entwicklungsteam erfolgen. Die Erarbeitung von Verbesserungsvorschlägen ist schon von Anfang an bei der Aufwandsschätzung zu berücksichtigen, da dies meist einen beachtlichen Aufwand verursacht (Heinsen 2003).

6. Fokusgruppen

6.1 Einleitung

Dieses Kapitel behandelt das Thema Fokusgruppen, welche eine Technik des Usability Engineering darstellen, mit deren Hilfe sowohl bestehende Interfaces verbessert, als auch neue entwickelt werden können. Nach einer kurzen Vorbereitungsphase (Klärung von Zielen und Themen, Zusammenstellung der Gruppe, Rollenverteilung) erfolgt die Diskussion (Durchführung) und im Anschluss werden die Ergebnisse analysiert, zusammen gefasst und es wird ein Bericht erstellt.

Fokusgruppen werden eingesetzt, um Bedürfnisse von Benutzern zu ermitteln, wobei diese Methode vor dem Design des Interface eingesetzt werden kann, aber auch wenn die Software bereits einige Zeit in Verwendung ist. Eine solche Gruppe besteht meist aus sechs bis neun Benutzern, die zusammengebracht werden, um neue Konzepte zu diskutieren (Nielsen 1993) .

Meist ist es ratsam, mehrere Fokusgruppen zu einem Thema zu bilden, da das Ergebnis einer einzelnen Gruppe nicht immer repräsentativ sein muss (Nielsen 1993; Heinsen 2003). Eine Sonderform stellen so genannte Benutzerclubs dar, welche sich wiederholt zum selben oder zu verschiedenen Themen zu Fokusgruppen zusammenfinden (Hassenzahl M. 1999).

6.2 Durchführung

Nachdem die Ziele und Themen einer Fokusgruppe geklärt wurden und deren Zusammenstellung feststeht, beginnt deren eigentliche Arbeit, welche hauptsächlich in einer moderierten Diskussion besteht. Jede Fokusgruppe verfügt über einen Moderator, der dafür verantwortlich ist, dass die Teilnehmer der Diskussion beim Thema bleiben. Wichtig dabei ist, dass der Moderator immer die Objektivität gegenüber dem Auftraggeber, dem zu behandelnden Thema und gegenüber den Teilnehmern wahrt. Bevor die Diskussion beginnen kann, muss der Moderator eine Liste von zu behandelnden Themen zusammenstellen, sowie Ziele bestimmen, so dass feststeht, welche Informationen gewonnen werden sollen (Nielsen 1993).

Wie alle Methoden, die nicht auf Messung der tatsächlichen Verwendung beruhen, sondern den Benutzer nach seiner oft subjektiven Meinung fragen, bergen auch Fokusgruppen die Gefahr, dass die Benutzer glauben etwas bestimmtes zu wollen, obwohl sie eigentlich etwas ganz anderes benötigen. Um diesem Umstand vorzubeugen empfiehlt Jakob Nielsen den Teilnehmern der Fokusgruppe möglichst konkrete Beispiele für die diskutierte Technologie bereitzustellen (Nielsen 1993).

Im Folgenden werden nun kurz die typischen drei Teile einer Fokusgruppe nach (Heinsen 2003) beschrieben:

6.2.1 Einleitung

- Zielorientierung
- Moderator stellt Ziele und Regeln vor
- Schaffung einer kommunikativen Arbeitsatmosphäre
- Vorstellungsrunde

6.2.2 Aktivitäten

- Durchführung der Aktivitäten, die festgelegt wurden
- Beispiele für Aktivitäten: Wunschkarten, idea seeding, Ideen und Konzepte bewerten, Fragestunde mit dem Auftraggeber
- Moderation
- Aktivitäten sollten für die Teilnehmer einfach und leicht nachvollziehbar sein.

6.2.3 Abschluss

- Gegebenenfalls weitere Ausarbeitung zentraler Punkte
- Zusammenfassung der Ergebnisse

6.3 Analyse und Bericht

Die einfachste Form der Analyse besteht darin, den Moderator am Ende einer Diskussion einen Bericht schreiben zu lassen, der die wichtigsten Ergebnisse übersichtlich zusammenfasst. Genauere Analysen sind möglich, aber meist aufgrund der Unstrukturiertheit einer Fokusgruppe schwer zu erstellen (Nielsen 1993). In den meisten Fällen verlangt der Auftraggeber einen solchen formellen Bericht vom Moderator. Folgende Aspekte sollten bei der Erstellung dieses Berichts berücksichtigt werden:

- **Anschaulichkeit:** Die Ergebnisse sind von zentraler Bedeutung, Beispiele (zum Beispiel Zitate aus der Diskussion) sind wichtig zur Veranschaulichung.
- **Pluralität der Perspektiven:** Die Ergebnisse der Fokusgruppe erleichtern dem Auftraggeber die Entscheidung, nehmen sie ihm aber nicht ab.
- **Vermeidung von Quantifizierungen:** Quantifizierungen sind die Aufgaben von Fragebögen und anderen Verfahren und bei solch kleinen Personenzahlen wie bei Fokusgruppen nicht sinnvoll.
- **Trade-Offs:** gesonderte Bewertung von einzelnen Ideen oder Konzepten, die besonders interessant sind; oft grundlegend für weitere Entscheidungen.

(Heinsen 2003)

Heinsen empfiehlt außerdem bei gegebener Möglichkeit die Ergebnisse beim Auftraggeber zu präsentieren, um dort mit weiteren Betroffenen diskutieren zu können.

7. Evaluationsfragebögen

7.1 Einleitung

Dieses Kapitel behandelt Qualitätskriterien bei der Erstellung von Evaluationsfragebögen, die Durchführung von Fragebogenbefragungen, Auswertung der Ergebnisse sowie zwei Beispiele für Software-Evaluationsfragebögen. Neben AttrakDiff (Hassenzahl M. 2002) werden die IsoMetrics-Fragebögen (Willumeit 1996) kurz vorgestellt, welche für diese Arbeit als Vorlage verwendet wurden.

Evaluationsfragebögen stellen eine Methode zur Erfassung von großen Datenmengen dar. Dabei erhält man vom einzelnen subjektive Daten, die auf den persönlichen Erfahrungen mit dem Gegenstand des Fragebogens beruhen. Durch die große Menge an Daten werden diese objektiviert und statistisch auswertbar. Je detaillierter der Fragebogen ist, umso größer sollte deshalb die Personenzahl sein (Nielsen 1993).

Mehrere Schritte sind zur Konstruktion eines Fragebogens nötig (Lienert G. 1994; Bortz J. 1995):

- Sammlung von Items
- Zuordnung von Items zu Subskalen
- Aussonderung ungeeigneter Items mithilfe Testtheorie

Dabei ist der Fragebogen empirisch auf die Einhaltung bestimmter Qualitätskriterien zu überprüfen und gegebenenfalls zu verfeinern (Heinsen 2003).

7.2 Qualitätskriterien

nach (Bortz J. 1995)

7.2.1 Hauptgütekriterien

- Objektivität
- Reliabilität
- Validität

7.2.2 Weitere Qualitätskriterien

- Ökonomie
- Nützlichkeit

Die Einhaltung dieser Qualitätskriterien ist wichtig, um den erhobenen Daten vertrauen zu können (Heinsen 2003).

7.3 Durchführung

Zu Beginn der Fragebogenuntersuchung werden die zu befragenden Nutzer ausgewählt. Spätestens dann muss die Art der Befragung bestimmt werden.

7.3.1 Befragung per Post

Im folgenden die wichtigsten Punkte, die bei dieser Befragungsart berücksichtigt werden müssen:

- Teilnehmer müssen in einem beiliegenden Anschreiben über den Initiator und die Thematik informiert werden (unter Angabe der Institution) (Heinsen 2003)
- Vertrauliche Handhabung der Daten
- Klärung von Fragen zur Datensicherheit
- Angabe eines Rücksendetermins
- Gefahr der „Selbstselektion“: Oft machen sich besonders zufriedene Nutzer nicht die Mühe, den Fragebogen auszufüllen (Heinsen 2003)
- Adressierte und frankierte Rückumschläge können den Rücklauf erhöhen (Armstrong J. S. 1988)

7.3.2 Befragung per Internet

Im Gegensatz zu anderen Befragungsmethoden gibt es bei Internetbefragungen die Problematik des Rücklaufes nicht, da dieser mit technischen Mitteln automatisch realisiert werden kann. Ansonsten ist diese Form der schriftlichen Befragung sehr ähnlich (Heinsen 2003).

7.3.3 Interviews

Interviews können telefonisch oder vor Ort stattfinden. In jedem Fall unterliegen sie Einschränkungen hinsichtlich des Termins. Sobald jedoch ein Termin mit dem Benutzer vereinbart wurde, ist es dem Interviewer normalerweise auch möglich, das Interview erfolgreich durchzuführen. Bei der Durchführung ist es besonders wichtig, dass der Interviewer neutral bleibt und den Befragten dazu bringt mit ganzen Sätzen zu antworten anstatt nur mit „Ja“ oder „Nein“ (Nielsen 1993).

Der Vorteil von Interviews liegt vor allem in der hohen Flexibilität, da der Interviewer beispielsweise sofort reagieren kann, wenn er den Eindruck gewinnt, dass der Befragte die Frage nicht richtig verstanden hat (Nielsen 1993).

7.4 Auswertung

Nachdem der Rücklauf abgeschlossen wurde, fallen quantitative und eventuell auch qualitative Daten (frei formulierbare Antworten) an, die ausgewertet werden müssen.

7.4.1 Auswertung quantitativer Daten

Die Itemwerte aller befragten Personen werden numerisch – meist in Form des Mittelwertes – zusammengefasst. Dabei ist unbedingt auf die Polung zu achten. Bei einer fünfstufigen Ratingskala müssen vor dem Summieren außerdem die Werte negativ formulierter Items nach der Formel $6 - x$ umgerechnet werden (Heinsen 2003).

Die Antwortskalen einiger Fragebögen sehen außerdem eine Möglichkeit vor, sich der Meinung zu enthalten. (wie zum Beispiel: „kann ich nicht beurteilen“) Diese Fälle müssen gesondert berücksichtigt werden.

Fragebögen mit mehr als 20% Auslassung sollten ausgeschlossen und gesondert analysiert werden (Gediga G. 1999). Für die IsoMetrics Fragebögen (siehe Kapitel 7.5) hat sich ansonsten die Ersetzung mit der Mitte der Skala als praktikabel erwiesen (Gediga G. 1999). Alternativ dazu können auch statistische Verfahren wie „missing data Prozeduren“ eingesetzt werden (Schafer 1997).

Sind mehrere verschiedene Versionen einer Software untersucht worden, empfiehlt sich zusätzlich die Erstellung einer Profilanalyse (Heinsen 2003).

7.4.2 Auswertung qualitativer Daten

Diese Daten liegen meist in Form von frei formulierten Antworten vor, die Schwachstellen oder besondere Stärken des Systems ansprechen, die nicht durch die geschlossenen Fragen des Evaluationsbogens abgedeckt werden konnten.

Aus diesen Antworten müssen zuerst die mit wenig aussagekräftigen Inhalten ausgeschlossen werden. Zusätzlich zur Bestimmung des Problemgehalts einer Anmerkung zur Software können diese beispielsweise nach funktionalen Aspekten kategorisiert werden. Diese Vorgehensweise gibt den Entwicklern konkrete Anknüpfungspunkte für Veränderungen oder Weiterentwicklungen (Heinsen 2003).

7.5 IsoMetrics

Der IsoMetrics Fragebogen ist ein Evaluationsfragebogen zum Thema Software auf der Basis der ISO 9241 Teil 10 (Willumeit 1996; Gediga G. 1999).

Der Fragebogen wurde an der Universität Osnabrück im Fachgebiet Arbeits- und Organisationspsychologie entwickelt. Von IsoMetrics gibt es zwei verschiedene Versionen, wobei IsoMetrics^S (s für short) nur die numerische Beantwortung der Fragen ermöglicht (quantitative Daten), während beim IsoMetrics^L (long) Fragebogen zusätzlich frei formulierbare Antworten vorgesehen sind, so dass auch qualitative Daten gewonnen werden können.

		stimmt nicht	stimmt wenig	stimmt mittelmäßig	stimmt ziemlich	stimmt sehr	
index	Aufgabenangemessenheit	1	2	3	4	5	Keine Angabe
A.1	Die Software zwingt mich, überflüssige Arbeitsschritte durchzuführen.						

Abbildung 1: Beispiel aus IsoMetrics^S Version 2.03 (Willumeit 1996)

	stimmt nicht	stimmt wenig	stimmt mittelmäßig	stimmt ziemlich	stimmt sehr	Keine Angabe
6.1 Die Software zwingt mich, überflüssige Arbeitsschritte durchzuführen.	1	2	3	4	5	

	nicht wichtig	wenig wichtig	mittelmäßig wichtig	ziemlich wichtig	sehr wichtig	Keine Angabe
Wie wichtig ist dieser Aspekt für Ihren Gesamteindruck von der Software?	1	2	3	4	5	

Können Sie konkrete Beispiele nennen, bei denen Sie dieser Aussage zustimmen können?

Abbildung 2: Beispiel aus IsoMetrics^L Version 2.03 (Willumeit 1996)

Der IsoMetrics Fragebogen umfasst sieben Subskalen, die den Gestaltungsgrundsätzen der ISO Norm 9241 Teil 10 entsprechen. Insgesamt umfasst der Fragebogen 75 Items, die auf einer Skala von 1 bis 5 oder mit „keine Angabe“ beantwortet werden können (siehe Abbildung 1). Bei der Langversion gibt es zu jedem Item eine zusätzliche Skala zur Bewertung seiner Wichtigkeit sowie Freiraum für konkrete Beispiele in Bezug auf das Item (siehe Abbildung 2).

Mit Hilfe von IsoMetrics können wichtige Informationen für die Bewertung von Software, insbesondere für die iterative Softwareentwicklung und die Prototypen-Evaluation gewonnen werden (Heinsen 2003).

Zu den IsoMetrics Fragebögen gibt es außerdem noch ein Handbuch (Gediga G. 1998), das Angaben zur Auswahl der Testpopulation, zu Testaufgaben, Prozeduren für die Datenauswertung, Berichterlegung und zum „Usability Review“ macht. Außerdem gibt es eine Internet-gestützte Variante sowie eine Version zur Evaluation von Web- und Multimediaanwendungen (Ollermann 2001).

Die Bearbeitungszeit der Fragebögen liegt bei ca. 30 bis 60 Minuten für die Kurzversion (IsoMetrics^S), beziehungsweise mindestens 2 Stunden für die lange Version IsoMetrics^L. Außerhalb von Forschungsprojekten ist die Nutzung des Fragebogens entgeltlich (<http://www.isometrics.uni-osnabrueck.de/> 29.12.2003; Heinsen 2003).

7.6 AttrakDiff

Während der zuvor vorgestellte IsoMetrics Fragebogen in erster Linie zur Bewertung der Gebrauchstauglichkeit von Software eingesetzt werden kann, erfasst der in diesem Kapitel vorgestellte AttrakDiff Fragebogen zusätzlich so genannte „hedonische“ Qualitäten der Software. Der Fragebogen besteht aus 23 Items, die in sieben Stufen unterteilt sind (siehe Abbildung 3).

Die Items sind außerdem zu drei Subskalen zusammengefasst: Pragmatische (ergonomische) Qualität (acht Items), hedonische Qualität (sieben Items) und Attraktivität (acht Items).

Beurteilung des Produkts Musterprodukt

Bitte geben Sie mit Hilfe der folgenden Wortpaare Ihren Eindruck zu Musterprodukt wieder.
Bitte klicken Sie in jeder Zeile eine Position an!

menschlich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	technisch
isolierend	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	verbindend
angenehm	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unangenehm
originell	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	konventionell
einfach	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	kompliziert
fachmännisch	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	laienhaft
hässlich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	schön
praktisch	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unpraktisch
sympathisch	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unsympathisch
umständlich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	direkt

1/3

abbrechen

weiter

Abbildung 3: AttrakDiff Beispiel (<http://www.attrakdiff.de> 29.12.2003)

Die Zuverlässigkeit der Skalen kann als sehr gut bezeichnet werden (Heinsen 2003) und es konnte gezeigt werden, dass pragmatische und hedonische Qualität voneinander unabhängig sind (Hassenzahl M. 2000; Hassenzahl M. 2002).

III. Fallbeispiel „orbImmun“

1. Einleitung

In diesem Teil der Arbeit wird die Usability-Evaluation an einem konkreten Beispiel gezeigt. Zuerst wird in Kapitel 2 das betroffene Softwareprojekt und die Software vorgestellt. Kapitel 3 erläutert die Vorgangsweise der Evaluation, wobei auch die wichtigsten Usability-Heuristiken vorgestellt werden, die im Laufe des Projekts herangezogen wurden. Kapitel 4 beinhaltet die Ergebnisse dieser Evaluation, untergliedert in die einzelnen Untersuchungsmethoden aus Kapitel 3. Am Ende von Kapitel 4 finden sich dann die Empfehlungen, die auch bei der Ergebnispräsentation (Kapitel 3.5) an die Auftraggeber ergingen. Das Ende des Abschnittes bildet dann Kapitel 5 (Diskussion).

2. Projektbeschreibung

2.1 orbImmun

orbImmun ist ein unter Microsoft Windows lauffähiges Reiseimpfprogramm, das niedergelassene Ärzte bei der Reiseberatung von Patienten unterstützt. Es bietet gesundheitliche Informationen bezüglich Reisen außerhalb Österreichs und Europas, der Auswahl von Impfstoffen und Medikamenten und darüber hinaus elektronische Unterstützung bei der Rezepterstellung.

orbImmun ist seit 1994 am Markt und wird von der Firma „Ablinger & Garber Publishing“ in Hall in Tirol betreut und gemeinsam mit der Österreichischen Ärztekammer vertrieben. Mehrere hundert Ärzte österreichweit und einige Ämter sind Abonnenten von orbImmun. Zweimal im Jahr erfolgt ein kostenpflichtiges Update, um die Reiseimpfungsdatenbank auf dem neuesten Stand zu halten.

2.2 Projekt „orbImmun Redesign“

Ziel des Projekts „orbImmun-Redesign“ ist die inhaltliche sowie programmtechnische Verbesserung der aktuellen Version von orbImmun, vor allem was die Benutzerfreundlichkeit und das Informationsdesign betrifft.

Aus der Projektausschreibung:

Phase 1 des Projekts „Analyse und Evaluation der aktuellen Version“ umfasst folgende Tätigkeiten:

- Usability Evaluation des Programms.
- Durchführung von Interviews mit ca. fünf orbImmun-Benutzern (Ärzte)
- Verfassen eines Evaluationsberichts mit Verbesserungsvorschlägen.

Projektleiter ist DI Raimund Schatz (UMIT – Institut für Informationssysteme) und die Medizinisch-Wissenschaftliche Betreuung hat a. o. Prof. Dr. H. Kollaritsch (Wien)

Diese Arbeit beschäftigt sich mit Phase 1 des Projektes „orbImmun Redesign“, also der Analyse und Evaluation der aktuellen Version der Software.

3. Vorgehensweise

3.1 Einleitung

Die Evaluation der Software wurde in folgenden Schritten durchgeführt, welche in den anschließenden Kapiteln genauer beschrieben werden:

- Interne Evaluation
 - Experten evaluieren anhand allgemeingültiger Kriterien (Heuristiken) und Erfahrungswerten die zu untersuchende Software. (Expertenevaluation)
 - Interne Diskussion der Ergebnisse
- Entwicklung eines Fragenkatalogs
 - Unter Berücksichtigung der internen Evaluation
 - In Anlehnung an IsoMetric Fragebögen von (Willumeit 1996)
 - Speziell angepasste Versionen für Telefoninterviews und e.mail Bearbeitung
- Externe Evaluation mit Hilfe von Test-Usern
 - Befragungen (telefonisch, vor Ort, per e.mail)
- Zusammenfassung der Ergebnisse
 - Präsentation
 - Empfehlungen
 - Besprechung der weiteren Vorgehensweise

3.2 Interne Evaluation

Nach allgemeingültigen wissenschaftlichen Kriterien betreffend Design und Usability von Softwareprodukten sowie anhand von Erfahrungswerten wurde die vorliegende Software von DI Raimund Schatz und dem Autor mehrfach getestet und untersucht.

Dabei auftretende Probleme und Auffälligkeiten wurden bei der Erstellung der Fragebögen für die externe Evaluation berücksichtigt. Auf Schwächen, die in dieser Phase bereits bekannt wurden ging der Autor mit gezielten Fragen speziell ein.

Bei dieser Erstevaluation wurden vor allem Jakob Nielsens „Usability Heuristics“ aus (Nielsen 1993) herangezogen. Als Erweiterung wurden jedoch auch Heuristiken von Ben Shneiderman - „Acht Goldene Regeln“ aus (Shneiderman 1998) sowie Donald Norman - Vier bzw. sieben Prinzipien für gutes Design aus (Norman 1990) und (Norman 1988) verwendet. In den folgenden Abschnitten werden die wichtigsten Heuristiken kurz aufgeführt.

3.2.1 Die 10 Usability-Heuristiken von Jakob Nielsen

3.2.1.1. Visibility of system status (Sichtbarkeit des Systemstatus)

Der Benutzer soll erkennen können, was das System gerade macht. Speziell wenn längere Vorgänge ohne Feedback stattfinden können unerfahrene Anwender leicht zur Annahme gelangen, das System hätte sich „aufgehängt“. Wichtige Punkte sind vor allem Orientierung und Navigation:

Orientierung

- Seite (Dialoge) beschriftet
- Kapitel/Kategorie der Seite sichtbar

Navigation

- Links klar beschriftet
- Links als solche erkennbar
- Hilfestellung zu "sinnvollen" Wegen

3.2.1.2. Match between system and the real world (Übereinstimmung von System und Wirklichkeit)

Software soll sich an die Arbeitsumgebung und die zukünftigen Benutzer „anpassen“.

Konzepte

- Aufbau, Metaphern Zielgruppe(n) vertraut

Sprache

- Wörter, Fremdwörter
- Satzbau
- Ausdruck

3.2.1.3. User control and freedom (Nutzerkontrolle / -freiheit)

Fehler sollen für den Benutzer erkennbar und vor allem behebbar sein. Des weiteren darf sich ein Benutzer durch das Interface nicht eingeschränkt fühlen.

Fehlertoleranz

- undo
- redo
- home

Anpassbarkeit

- Abbruch jeder Aktion/ jedes Features möglich
- Lauffähigkeit auf verschiedenen Browsern / Systemen
- Veränderbarkeit des Layouts

3.2.1.4. Consistency and standards (Konsistenz und Standards)

Konsistenz und Standardisierung erleichtern dem Benutzer das Auffinden von wiederkehrenden Elementen auch in verschiedenen Softwareprodukten.

Konsistenz

- Begriffe
- Buttons
- Links
- Überschriften
- Ausdruck

Standards

- Web / Browser Standards

3.2.1.5. Error prevention (Fehlervorbeugung)

Fehlervorbeugung soll nicht nur bei der Programmierung eingesetzt werden (Programmfehler). Es sollte genauso auch versucht werden Benutzerfehler aufzufangen. Ein simples Beispiel dafür ist die so genannte „confirmation box“ – Eine Art Schutz vor potentiell gefährlichen Aktionen (Beispiel: Festplatte formatieren) wobei der Benutzer gefragt wird, ob er sich wirklich sicher ist, dass er die gewählte Aktion ausführen will.

3.2.1.6. Recognition rather than recall (Wiedererkennung nutzen statt Erinnerung)

- Objekte, Aktionen, Optionen sichtbar
- Verlauf erkennbar
- Beschriftung Seiten, Links
- Icons verständlich

Den Benutzer zu zwingen Aktionen auswendig auszuführen führt dazu, dass dieser das Programm für „schwierig“ oder „kompliziert“ klassifiziert.

3.2.1.7. Flexibility and efficiency of use (Flexibilität und Effizienz, Adaptivität)

Flexibilität

- Beschleuniger / Abkürzungen für Experten
- Unsichtbarkeit dieser Beschleuniger / Abkürzungen für Novizen
- Routinen für häufige Aktionen
- Suchfunktion
- Bookmarking

Effizienz

- kürzeste (zeitlich) Wege werden angeboten
- strategisch optimierte Abläufe

3.2.1.8 Aesthetic and minimalist design (Ästhetisches und minimalistisches Design)

- Design an Zielgruppe(n) angepasst (Layout)
- Design der Wahrnehmung angepasst
- Schriftgröße und -format
- Ebenen der Information
- auf jeder Ebene nur das Notwendige
- Links zu tiefergehenden Informationen
- „Chunking“ der Informationen

3.2.1.9 Help users recognise, diagnose, and recover from errors (Benutzer soll Fehler erkennen, diagnostizieren und beheben können)

- keine Codes, normaler Text
- Lösungswege vorschlagen

Ziel ist eine Verwirrung der Benutzer durch Insiderinformationen wie Fehlercodes zu vermeiden und dem Benutzer konstruktive Hilfe bei Problemen zu bieten.

3.2.1.10 Help and documentation (Hilfe und Dokumentation)

- Hilfefunktion anbieten
- schon bei sehr geringem Level an Schwierigkeit
- sofort, einfach erreichbar
- Integration der Hilfe - Links in Seiten
- Hilfefunktion als intelligenter Software Agent (stellt sich auf die Bedürfnisse des Users ein)

3.2.2 Acht Goldenen Regeln des Interface Designs nach (Shneiderman 1998)

3.2.2.1 Strive for consistency (Einheitlichkeit)

Unter „Consistency“ versteht man nicht nur die Einheitlichkeit von Befehlsabfolgen in ähnlichen Situationen, sondern auch die Terminologie in Buttons, Feldern, Menus, Zusatzhilfe etc. und das einheitliche Layout im Gesamterscheinungsbild wie Farbe, Schrift, und Ikonografie. Die Konsistenz kann in bestimmten Fällen aufgebrochen werden um Zusatzfunktionen zu betonen, oder in Fällen, in denen Funktionen gewisse Standards aufweisen müssen, die unveränderlich sind (z.B. Rückfrage vor dem Löschen, Sternchen bei Passworteingabe).

→ Siehe auch Heuristik 3.2.1.4

3.2.2.2 Enable frequent users to use shortcuts (Abkürzungen für geübte User)

Benutzer, die bereits das Computersystem kennen, möchten die Effizienz der Interaktion erhöhen, indem sie bestimmte Prozeduren abkürzen. So kann das System Shortcuts (um den Weg mit der Maus zu verkürzen) oder Makroprogrammierung (zur Zusammenfassung oft wiederholter aufeinanderfolgender Arbeitsschritte, die dann z.B. mittels Funktionstaste ausgeführt werden) anbieten.

→ Siehe auch Heuristik 3.2.1.7

3.2.2.3 Offer informative Feedback (Rückmeldung vom System)

Für jede Aktion, die der User setzt, soll vom System eine Rückmeldung gegeben werden, damit der Benutzer weiß, ob das System die Eingabe akzeptiert hat. Für häufige und unbedeutende Eingriffe kann die Rückmeldung sehr unscheinbar sein, bei wichtigen und seltenen dafür stärker, um dadurch ihren Einfluss auf das System zu verdeutlichen.

→ Siehe auch Heuristik 3.2.1.1

3.2.2.4 Design dialogs to yield closure (Klare Abfolgen bis zum Ende)

Sequenzen von Aktionen sollen so organisiert sein, dass sie einen klaren Anfang und ein klares Ende haben und der Benutzer immer weiß, welcher Schritt als nächster durchzuführen ist. Eine informative Rückmeldung beim Beenden einer Abfolge gibt dem Benutzer das Vertrauen, dass die Aktionen korrekt ausgeführt wurden. Außerdem zeigt es den Status an, wann die nächste Aktion gesetzt werden kann.

3.2.2.5 Offer error prevention and simple error handling (Fehlervermeidung und -handling)

Das System muss derart konzipiert sein, dass dem Benutzer keine schwerwiegenden Fehler unterlaufen können. So ist z.B. ein Auswahlm Menü einem Eingabefeld vorzuziehen (fixe Auswahlmöglichkeit, Vermeidung von Tippfehlern) oder die Eingabe von Text in Zahlenfeldern unmöglich zu machen. Das System soll im Falle eines Fehlers dem User die Rückmeldung geben, warum der Fehler aufgetreten ist, und wie er (mit möglichst geringem Aufwand) repariert werden kann. Fehler sollten niemals zu einem instabilen System führen bzw. soll der User die Möglichkeit besitzen, den ursprünglichen Zustand des Systems wiederherzustellen (z.B. Reset-Button).

→ Siehe auch Heuristiken 3.2.1.5 und 3.2.1.9

3.2.2.6 Permit easy reversal of actions (Leichtes Rückgängigmachen von Aktionen)

Befehle oder Befehlsketten müssen rückgängig gemacht werden können. Dadurch ermuntert man den User, Dinge auszuprobieren, die er noch nicht kennt. Außerdem kann er im Falle eines unerwünschten Ergebnisses den Befehl rückgängig machen und so eine Fehlerkorrektur vornehmen.

→ Siehe auch Heuristik 3.2.1.3

3.2.2.7 Support internal locus of control (Kontrollmöglichkeit für erfahrene User)

Erfahrene Benutzer von Computersystemen ziehen es vor, wenn das System so reagiert, wie sie es vorgeben und nicht, dass sie auf das System reagieren müssen. Unerwartet auftauchende Elemente u.ä., die den Benutzer frustrieren, sind zu vermeiden. Idealerweise kann der Benutzer sein Interface seinen Bedürfnissen anpassen (z.B. mittels Benutzerprofilen).

3.2.2.8 Reduce short-term memory load (Freihalten des menschlichen Kurzzeitgedächtnisses)

Um dieses Ziel zu erreichen, müssen aktuelle Informationen über das Interface (Eingabecodes, Abkürzungen etc.) für den Benutzer ständig erreichbar sein (z.B. über das Drücken der »F1«-Taste). Das Interface ist klar zu strukturieren, damit sich der Benutzer mit dem Inhalt der angebotenen Information beschäftigen kann und sich nicht auf die Bedienung des Interfaces konzentrieren muss. Deshalb ist auf Zusatzfunktionen wie Hilfesystem oder interaktive Lernsimulationen besonderes Augenmerk zu legen. Es ist wichtig, dass der User genügend Zeit zur Verfügung hat, damit er die

Bedienung des Interfaces lernt und sie versteht. User können sich meist nur an ca. „sieben plus oder minus zwei“ Informationen erinnern (Shneiderman 1998).

3.2.3 Vier Prinzipien für gutes Design von Donald Norman (Norman 1990)

3.2.3.1 State and action alternatives should be visible (Status und Alternativen sollen sichtbar sein)

Durch bloßes Hinsehen sollte der Benutzer den Zustand des Gerätes/Programms sowie alternative Aktionen erkennen können.

3.2.3.2 Conceptual model with a consistent system image (Konzeptionelles Modell mit konsistentem Systemabbild)

Der Designer soll dem Benutzer ein gutes mentales Modell bieten. Dazu gehören konsistent präsentierte Operationen mit Resultaten sowie ein unveränderliches und konsistentes „System image“.

3.2.3.3 Interface should include good mappings that reveal the relationship between stages. (Das Interface sollte über gutes Mapping verfügen, das die Beziehungen zwischen verschiedenen Stufen/Arbeitsphasen erkennbar macht.)

Die Beziehungen zwischen Aktionen und Resultaten sollen erkennbar sein, wie zwischen Schaltern und Effekten, und natürlich zwischen dem Zustand des Systems und den sichtbaren Anzeichen.

3.2.3.4 User should receive continuous feedback. (Der Benutzer sollte ständig Feedback erhalten.)

Der Benutzer soll ein vollständiges und kontinuierliches Feedback über die Resultate von Aktionen erhalten.

→ Siehe auch Heuristik 3.2.1.1

3.2.4 Sieben Prinzipien des Designs von Donald Norman (Norman 1988)

3.2.4.1 Use both knowledge in the world and knowledge in the head. (Nutze sowohl das Wissen im Kopf als auch das Wissen in der Umwelt.)

Wissen im Kopf (wie zum Beispiel die Bedeutung des Wortes „Abbrechen“) und Wissen in der Umwelt (wie zum Beispiel ein Knopf, der per definitionem dazu dient gedrückt zu werden, das Wissen über seine eigene Funktionsweise also bereits enthält) müssen optimal genutzt werden.

→ Siehe auch Heuristik 3.2.1.6

3.2.4.2 Simplify the structure of tasks. (Vereinfache die Struktur der Aufgaben.)

Die Strategie liegt hier in der Bereitstellung von Gedächtnisstützen. Es ist mit Hilfe der Technik das sichtbar zu machen, was sonst unsichtbar bliebe und auf diesem Weg Feedback und Steuerung zu verbessern, ohne dem Benutzer die Kontrolle zu entziehen.

3.2.4.3 Make things visible: bridge the gulfs of Execution and Evaluation. (Mache Dinge sichtbar: Überbrücke die Kluft der Ausführung und der Auswertung.)

Damit Benutzer die Auswirkungen ihrer Handlungen sehen können, müssen Aktionen auf der Ausführungsseite einer Handlung sichtbar gemacht werden. Außerdem sollte das System Hinweise auf seinen Zustand geben, die leicht wahrzunehmen und zu interpretieren sind.

→ Siehe Heuristik 3.2.1.1

3.2.4.4 Get the mappings right. (Benutze gutes Mapping.)

Ein natürliches Mapping muss folgende Beziehungen feststellbar machen:

- zwischen der Intention und den möglichen Handlungen
- zwischen den Handlungen und ihren Auswirkungen auf das System
- zwischen dem wirklichen und dem wahrnehmbaren Systemzustand
- zwischen dem wahrgenommenen Systemzustand und den Bedürfnissen, Erwartungen und Intentionen des Benutzers.

Hierbei spielen Bilder und Graphiken eine wichtige Rolle.

3.2.4.5 Exploit the power of constraints, both natural and artificial. (Nutze die Macht von natürlichen und künstlichen Einschränkungen.)

- damit der Benutzer das Gefühl hat, es sei nur eine einzig richtige Handlung möglich.
- damit die Zahl der möglichen Alternativen reduziert wird.

Zweck dieser Anweisung ist die Komplexität und somit die Beanspruchung des Benutzers gering zu halten.

3.2.4.6 Design for error. (Designe fehlerfreundlich.)

Rechne mit Fehlern und erleichtere das Rückgängigmachen von Aktionen, Unterstützung statt Bekämpfung des Benutzers.

→ Siehe auch Heuristik 3.2.1.5

3.2.4.7 When all else fails, standardise. (Wenn alles andere versagt: Standardisierung.)

Standardisierung erleichtert zumindest das Erlernen von Arbeitsabläufen.

→ Siehe auch Heuristik 3.2.1.4

3.3 Fragenkatalog

Im Anschluss an die Expertenevaluation wurden deren Ergebnisse für die Entwicklung eines Fragenkataloges herangezogen, um gezielt auf bereits entdeckte Mängel eingehen zu können. Abgesehen von solchen gezielten Fragestellungen waren produktspezifische und benutzerspezifische (Vokabular usw.; siehe auch Abschnitt II Kapitel 7) Eigenheiten zu berücksichtigen. Auch die IsoMetrics-Fragebögen (Willumeit 1996) dienten für einige Bereiche des Fragenkataloges als Vorlage.

Der Evaluationsbogen umfasst 34 Items, die in die Subskalen Arbeitsablauf, Übersichtlichkeit, Praktikabilität und weitere Fragen eingeteilt wurden. Davon beziehen sich sieben Items speziell auf Probleme, die bereits im Vorfeld bei der Expertenevaluation entdeckt wurden.

Der Fragenkatalog wurde dann für den Einsatz unter verschiedenen Bedingungen angepasst, so dass Versionen für Telefonbefragung, schriftliches ausfüllen und eine Version für e.mail Befragungen entstanden. Dazu wurden 15 Items des Telefonfragebogens als Fragen von weniger hoher Priorität gekennzeichnet, um bei Zeitdruck des Befragten flexibel reagieren zu können. Zum Einsatz kam bis auf eine Ausnahme die Telefonvariante, da alle Befragten sich für diese Befragungsart entschieden, bis auf eine Person, die den Bogen per e.mail bearbeitete.

Tabelle 3: Beispiele für Fragen aus dem orbImmun Evaluationsfragebogen

Subskala	Frage	Typus
Arbeitsablauf	Die Software zwingt mich, überflüssige Arbeitsschritte durchzuführen?	ja/nein
Arbeitsablauf	Sind die 2 Splash Screens beim Programmstart für sie wichtig und informativ?	1-5
Übersichtlichkeit	Wichtige Befehle und Funktionen lassen sich leicht auffinden?	1-5
Übersichtlichkeit	Bei Bedarf stellt die Software auch Hilfe und Anleitungen zur Verfügung?	ja/nein
Praktikabilität	Bereits eingegebene Daten bleiben auch im Fehlerfall verfügbar?	ja/nein
Praktikabilität	War die Handhabung der Lizenzdiskette für Sie problematisch?	ja/nein
weitere Fragen	Welche Features wären zusätzlich sinnvoll und praktisch?	Freitext

Die Fragen 2 und 6 beziehen sich speziell auf Besonderheiten der Software. Insgesamt waren 20 der Fragen mit ja/nein zu beantworten, bei drei Fragen waren frei formulierbare Antworten möglich und die restlichen Fragen waren mit einer fünfstufigen Skala zu bearbeiten.

3.4 Externe Evaluation

Nach erfolgreicher Terminvereinbarung wurden fünf orbImmun Benutzer (hauptsächlich Ärzte) kontaktiert, welche die Software anhand des Fragebogens evaluierten. Die Evaluation fand teilweise telefonisch und teilweise schriftlich (per e.mail) statt. Mehr Feedback konnte bei den Telefonbefragungen gewonnen werden, da die „live“-Kommunikation am Telefon gegenüber anderen Kommunikationsformen flexibler und somit ergiebiger ist (Nielsen 1993). Dennoch sind in den meisten Fällen andere Kommunikationsformen vorzuziehen, da Telefonbefragungen subjektiv oft als aufdringlich und lästig empfunden werden. Dank der ständig zunehmenden kontinuierlichen Spam-Überflutungen der letzten Jahre ist auch im Bereich Email die Sensibilität der Benutzer stark erhöht. In diesem Fall konnten jedoch Email- und Telefonbefragungen durchgeführt werden, da die Befragten bereits im Vorfeld informiert waren und den Termin der Befragung frei wählen konnten. Die Fragen zur Software wurden großteils vollständig und sinngemäß beantwortet.

3.5 Ergebnispräsentation

Die Ergebnisse der vorangegangenen Untersuchungen wurden vor dem Auftraggeber und anderen Betroffenen präsentiert. Dabei wurde zum einen ein Evaluationsbericht erstellt, welcher nach einer kurzen Einleitung und der Vorstellung der Vorgehensweise die Ergebnisse zusammenfasst. Der Schluss enthält Empfehlungen zur weiteren Vorgehensweise und im Appendix finden sich die vollständigen Ergebnisse der internen wie der externen Evaluation. Des weiteren wurden die Ergebnisse mit Hilfe einer kurzen Powerpoint Präsentation zusammengefasst vorgestellt, die in etwa dieselbe Gliederung aufwies.

4. Ergebnisse

4.1 Einleitung

Während bei der internen Evaluation strikt sequentiell vorgegangen wurde, waren die Fragen der externen Evaluation in folgende Themenbereiche gegliedert:

- Arbeitsablauf
- Übersichtlichkeit
- Praktikabilität
- Sonstiges

Ergebnisse der internen Evaluation wurden zum größten Teil durch die externe Evaluation bestätigt. Es folgt eine überblicksartige Zusammenfassung der wichtigsten Ergebnisse:

- Der sequentielle Ablauf des Bearbeitungsvorganges ist zu unflexibel, nur wenige User finden diesen erzwungenen Ablauf praktikabel. Dies dürfte die Hauptursache dafür sein, dass die Software betreffend Arbeitsablauf hauptsächlich negativ bewertet wurde.
- Das wenig ökonomische Formulardesign ist nicht übersichtlich genug. Dies geht so weit, dass einzelne Features von einem großen Teil der Befragten nie entdeckt wurden. Deshalb wurde das Programm betreffend Übersichtlichkeit ebenfalls eher schlecht bewertet.
- Auch betreffend Praktikabilität schnitt das Produkt nicht optimal ab.

Neben den vielen Kritikpunkten konnten die Benutzer allerdings auch einiges an konstruktiven Verbesserungsvorschlägen einbringen. Vieles davon ging in Richtung moderne Technologien (Internet) bzw. inhaltliche oder funktionelle Erweiterungen. Ein detaillierter Einblick in die Ergebnisse der Evaluation findet sich in den folgenden Kapiteln.

4.2 Ergebnisse der internen Evaluation

Bei der Internen Evaluation wurden viele Widersprüche zu den herangezogenen Heuristiken (siehe Kapitel 3.2.1 bis 3.2.4) gefunden. Im Folgenden werden beispielhaft einzelne der insgesamt ca. 20 Findings abgebildet, die sich von kleineren Verstößen gegen Designkonventionen über mittelschwere bis schwere Layout-Problemen bis hin zu „Usability Katastrophen“ (siehe Abschnitt II Kapitel 1.3) verteilen.

- orbImmun verfügt über einen Kopierschutz, der mit Hilfe einer Lizenzdiskette realisiert wurde. Abgesehen davon, dass diese Technologie stark veraltet ist, wurde auch bei der Überprüfung dieser Diskette ein Usability Problem festgestellt: Diese Überprüfung erfolgt nach dem ersten Start, welcher jedoch nicht automatisch nach der Installation erfolgt. Im Anschluss wurden drei Messageboxes eingesetzt, um die Überprüfung der Diskette zu realisieren (siehe Abbildung 4 bis Abbildung 6).



Abbildung 4: orbImmun Lizenzdiskette 1



Abbildung 5: orbImmun Lizenzdiskette 2



Abbildung 6: orbImmun Lizenzdiskette 3

Das Windows Symbol für „kritischen Abbruch“ (roter Kreis mit weißem X) eignet sich bei den ersten zwei Meldungen nicht, da es sich beim Inhalt der Messageboxen um eine Frage beziehungsweise um eine Information handelt. Es wären also die entsprechenden Windows Symbole „Question“ und „Info“ (ein kleines, blaues „i“ in einer weißen Sprechblase) einzusetzen. Abgesehen davon lässt sich der ganze Vorgang mit einer einzigen Messagebox abwickeln. Der Hinweis (Messagebox 3) zur erfolgreichen Prüfung der Lizenzdiskette kann weggelassen werden und der Text der erste Messagebox könnte wie folgt formuliert werden: „Bitte legen Sie die mitgelieferte Lizenzdiskette in das Diskettenlaufwerk ein und klicken sie dann auf OK.“ So lassen sich die zweite und die dritte Messagebox vermeiden und der Benutzer erspart sich zwei Mausklicks und zwei Meldungen.

Dieses Problem ist jedoch nicht sehr brisant (Stufe 1 nach Nielsen 2003), da dieser Vorgang nur einmal erfolgt (siehe Abschnitt II Kapitel 1.3).

- Befindet man sich in der Druckvorschau, ist die „close box“ (das „X“ zum Schließen der Anwendung rechts oben) deaktiviert. Will man das Programm nun beenden, kann nur „Ende“ in der Menüleiste ausgewählt werden, was (auch außerhalb der Druckvorschau) zu Fehlermeldungen führt. Wie auch an anderen Stellen im Programm wird auf der „error box“ ein Hilfebutton angezeigt, der nichts bewirkt. Der Text der Fehlermeldung lautet: „Alias ‚DRUCK‘ wurde nicht gefunden“ und ergibt wenig Sinn. Während der Hilfe-Button gar nichts bewirkt, führen „Abbrechen“ und „Ignorieren“ beide zum gleichen Ergebnis, nämlich zum Programmende ohne weitere Vorwarnung. Bereits eingegebene Daten gehen dadurch verloren.

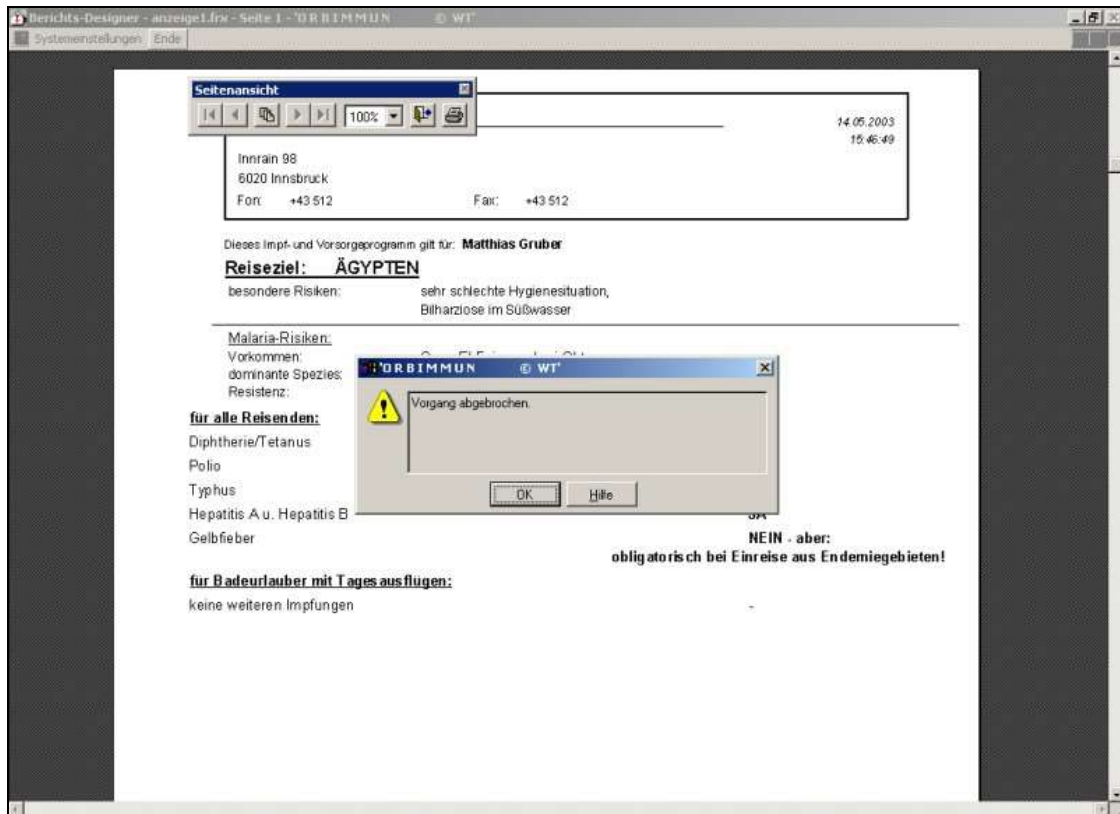


Abbildung 7: orbImmun Druckvorschau 1

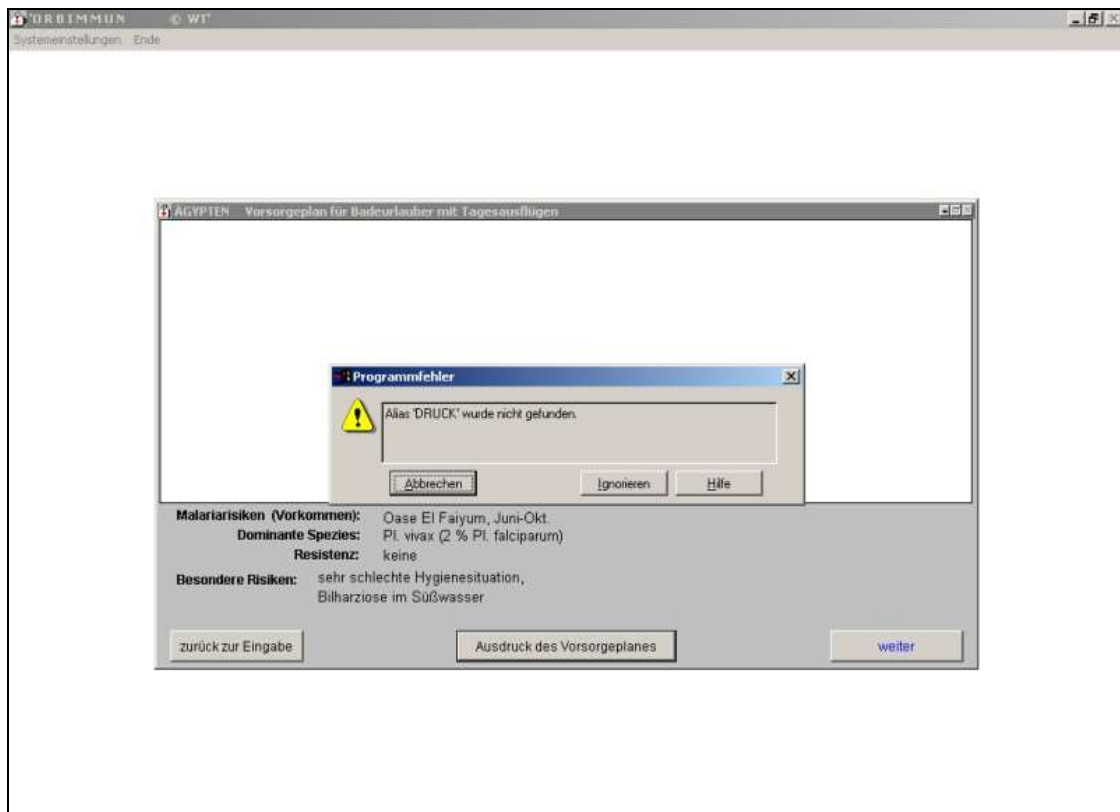


Abbildung 8: orbImmun Druckvorschau 2

Dieses Problem ist rein programmiertechnischer Natur und von mittlerer Priorität, da der Benutzer den Fehler umgehen kann, indem er die Druckvorschau verlässt, bevor er das Programm beendet. Dennoch stellt dies ein Problem dar, da zuvor eingegebene Daten verloren gehen können.

- orbImmun ist – im Widerspruch zum GUI Design, welches als MDI (Multiple Document Interface) vorliegt – ein assistentenähnliches Programm, da der Hauptteil der Arbeit in einzelnen, sequentiellen Schritten erfolgt. Assistentendesign wird für Arbeitsabläufe benutzt, die sequentiell sind und durch die nach Möglichkeit vor- und rückwärts navigiert werden kann. Dies wird meist durch einen „Zurück“- und einen „Weiter“-Button umgesetzt. Auch bei orbImmun gibt es bei allen Dialogen des Eingabevorgangs einen Button mit der Beschriftung „Zurück zur Eingabe“. Wenn dieser angeklickt wird, springt das Programm nicht zum vorhergehenden sondern zum ersten Dialog „Eingabe“ und verliert einen Teil der bereits eingegebenen Daten.



Abbildung 9: orbImmun „zurück“-Funktion 1



Abbildung 10: orbImmun „zurück“-Funktion 2

Dieses Problem ist als „Usability Katastrophe“ oder zumindest als Problem hoher Priorität (Stufe 3 nach (Nielsen 2003) einzuordnen, da es sehr hartnäckig ist, vom Benutzer nicht gelöst werden kann und immer auftritt, wenn der Benutzer eine Fehleingabe nicht sofort im selben Dialog bemerkt. Dies kann sich bei bis zu sieben Sessions pro Woche (siehe Kapitel 4.3.1) recht häufig ereignen. Statt „Zurück zur Eingabe“ sollte ein „Zurück“ zum vorhergehenden Dialog führen und bereits eingegebene Daten dürfen dabei nicht verloren gehen. Auch kann das MDI Design eingespart werden, was allerdings nur bei einer Neuprogrammierung sinnvoll ist. Eine nachträgliche Änderung des Software Designs bei einer bereits fertigen Software ist sehr aufwendig und fehleranfällig.

Weitere Probleme waren größtenteils sehr ähnlich zu den drei vorgestellten und waren nach (Nielsen 2003) in den Stufen 1 bis 3 einzuordnen (siehe Abschnitt II Kapitel 1.3).

4.3 Ergebnisse der externen Evaluation

4.3.1 Einleitung

Da der Evaluationsbogen bei einigen Fragen eine Skala von 5 Stufen zur Auswahl hatte, wurden diese Stufen immer so gepolt, dass die 5 wie im österreichischen Schulnotensystem dem schlechtest möglichen Ergebnis entsprach und die 1 dem bestmöglichen Ergebnis.

Das Programm orbImmun wird von den Befragten zwischen ein und sieben mal pro Woche benutzt. Die Dauer der Benutzung liegt zwischen 10 und 30 Minuten und ist pro Anwender relativ konstant, da pro Patient immer etwa derselbe Vorgang mit nur wenig Variationsmöglichkeiten durchgeführt wird. Unterschiede in der Bearbeitungsdauer waren hauptsächlich auf die Art der Bearbeitung zurückzuführen, da nicht alle Benutzer den Patienten nebenbei informieren. (Eine Person setzt die Software für Telefonberatungen ein, wodurch die Bearbeitungszeit eher die Obergrenze erreichte.) Daher entspricht orbImmun in Bezug auf die Benutzungsfrequenz und -dauer nahezu einem Mailclient oder einer ähnlich häufig benutzten Anwendung. Unter diesem Gesichtspunkt müssen auch die Usability - Anforderungen an die Software betrachtet werden.

4.3.2 Arbeitsablauf

Nur eine Person war der Meinung, dass keine überflüssigen Arbeitsschritte erforderlich sind. Auch das Feedback und das Hilfs- und Informationsangebot wurde im Durchschnitt bei geringer Streuung mit 4 benotet. Auf die Frage „Die Software bietet mir einfache und praktikable Lösungen um fehlerhafte Schritte zu korrigieren?“ war nur eine Person der Meinung, dass dies zutreffe. Diese Antwort ist schwer nachzuvollziehen und vielleicht auf ungenaues Lesen der Frage zurückzuführen, da in der Software keinerlei Problemlösungsfunktionalität existiert, außer dem Button „Zurück zur Eingabe“, der aber nicht besonders hilfreich ist (siehe Kapitel 4.2).

Die logische Abfolge und Navigation durch die einzelnen Arbeitsschritte wurde durchschnittlich mit drei benotet, allerdings antwortete nur eine Person mit „ja“ auf die Frage, ob die einzelnen Arbeitsschritte der Software auch den realen Arbeitsabläufen entsprechen.

4.3.3 Übersichtlichkeit

Die Auffindbarkeit von Befehlen und Funktionen wurde von den Evaluationsteilnehmern durchgehend mit 3 (neutral) benotet. Die Terminologie (Verständlichkeit der verwendeten Ausdrücke usw.) wurde fast durchgehend mit 1 (sehr gut) benotet. Allerdings war auch die Aussage zur Aktualität der Software eindeutig: Auf die Frage „Wirkt die Bedienoberfläche der Software zeitgemäß auf Sie?“ wurde ausschließlich mit 5 (schlechtest mögliche Bewertung) geantwortet.

Auch in diesem Themenbereich befand sich noch eine Frage zur Fehlerbehandlung: „Wenn Fehler auftreten weiß ich anschließend warum und kann die Aktion die zu dem Fehler führte rückgängig machen?“. Nur eine Person war der Meinung, dass dies zutreffe. Dasselbe Ergebnis ergab die Frage „Bei Bedarf stellt die Software auch hilfreiche Informationen und Anleitungen zur Verfügung?“. Nur eine Person war der Ansicht, dass dies zutrifft. Das ist schwer nachvollziehbar, da die Evaluatoren bei der internen Evaluation keinerlei Hilfe oder ähnliches entdecken konnten.

Die Übersichtlichkeit der Papierausdrucke wurde zwar im Durchschnitt mit 3.4 benotet, allerdings ist dieser durch eine einzelne Bewertung mit 1 (sehr gut) nach unten verschoben. Die Antworten im einzelnen: Ein mal 5 (sehr schlecht), zwei mal 4 (schlecht), ein mal 3 (neutral) und einmal 1 (sehr gut).

Zwei Fragen konnten aufgrund der großen Streuung beim Gesamtergebnis nicht berücksichtigt werden:

- „Meldungen und Fehlermeldungen der Software sind konstruktiv, klar und verständlich?“

Zwei Personen antworteten mit „ja“, zwei mit „nein“ und eine Person machte keine Angabe.

- „Die Begriffe und Fachausdrücke in der Software entsprechen denen meiner Arbeitstätigkeit?“

Zwei Personen antworteten mit „ja“, eine mit „nein“ und zwei Personen machten keine Angabe.

Insgesamt ist die Übersichtlichkeit der Software besser zu bewerten als der Arbeitsablauf, dennoch sicher nicht völlig zufriedenstellend.

4.3.4 Praktikabilität

Die Bedienung der Software wurde einstimmig als leicht erlernbar angegeben. Allerdings wurde die Frage „Das Programm ist im Umfang der abgedeckten Funktionalität genau richtig?“ im Durchschnitt mit 3.5 benotet, wie auch die Handhabung der Lizenzdiskette: Eine Person machte keine Angabe, da ein Dritter die Installation vorgenommen hatte. Alle anderen empfanden die Handhabung der Lizenzdiskette als problematisch. Auch folgende zwei Fragen wurden eindeutig negativ beantwortet:

- Bereits eingegebene Daten bleiben auch im Fehlerfall verfügbar?

Diese Frage wurde einstimmig mit „**Nein**“ beantwortet.

- Die Software konnte problemlos und einfach installiert werden?

Drei Personen antworteten mit „**Nein**“, zwei machten keine Angabe mit der Begründung, dass sie diese nicht selbst installiert haben.

Auf die Frage, ob bei der Arbeit mit der Software Systemfehler („Abstürze“) auftraten, antworteten zwei der Befragten mit „Ja“. Bei der internen Evaluation war es nicht ohne weiteres möglich, das Programm zum Absturz zu bringen. Möglicherweise wurden anderweitige Fehler von den Befragten als „Abstürze“ oder „Systemfehler“ fehlinterpretiert.

Die Praktikabilität der Software ist nicht durchgehend schlecht, dennoch ist ein großes Verbesserungspotential gegeben.

4.3.5 Weitere Fragen

- Ist die Hilfefunktion ausreichend genau und übersichtlich?

Diese Frage wurde zur Bestätigung der internen Evaluation eingeführt. Nur eine Person antwortete mit „ja“, was schwer nachvollziehbar ist, da bei der internen Evaluation keine Hilfefunktion gefunden werden konnte.

- Ist das Gewicht des Patienten für diese Anwendung von Bedeutung?

Diese Frage stellte sich den Evaluatoren während der internen Evaluation, da ein Feld für das Gewicht des Patienten im Beratungsarchiv vorhanden ist, die Eingabe des Patientengewichts im Eingabeformular

jedoch nicht vorgesehen wurde. Nur eine Person machte keine Angabe, alle anderen Befragten antworteten mit „ja“. Folgende Information war den Kommentaren der Befragten zu entnehmen: „Speziell beim Thema Malariaphylaxe spielt das Gewicht des Patienten eine Rolle.“ Dies sollte für eine zukünftigen Version abgeklärt werden.

- **Bereitet Ihnen die Funktion des Buttons „Zurück zur Eingabe“ Probleme?**

Diese Frage ergab sich aus der internen Evaluation und wurde wie erwartet hauptsächlich mit „ja“ beantwortet, nur eine Person antwortete mit „nein“.

- **Ermöglicht die Software flexibles (nicht eingetragenes) arbeiten? (1-5)**

Durchschnitt 4.5, eine Person machte keine Angabe, eine Person antwortete mit 3, alle anderen mit 5.

Die folgenden drei Fragen waren mit frei formulierbaren Sätzen zu beantworten. Die zusammengefassten Antworten werden im Folgenden stichwortartig aufgelistet:

- **Welche Features wären zusätzlich sinnvoll und praktisch?**

- Intervalle der Teilimpfungen
- Angaben zur verschiedenen Applikation (Alternativen zur intramuskulären Gabe, wie bei Antikoagulierten)
- Differenzierte Verhaltenshinweise für einzelne Regionen
- Links
- Stichwortverzeichnis
- Zusätzliche Daten über Pharmafirmen wie Telefonnummer oder Adresse weil oft individuelle Fragen auftauchen und dann Rückfragen erforderlich sind.
- Update per Internet
- eine Server Version, damit mehrere Personen gleichzeitig daran arbeiten können. Die Probleme würden sich reduzieren, da dann die EDV für die Software zuständig wäre.

- **Welche Features sind überflüssig und könnten weggelassen werden?**

- Geburtsdatum ist sinnlos, da irrelevant, ungefähres Alter sinnvoller (z.B. 40-a, 45-a, etc.)

- **Sonstige Verbesserungsvorschläge / Anmerkungen:**

- Ich verwende die Software nur für spezielle Fälle, nicht für meine normale Arbeit, da sie nicht telefonberatungstauglich ist wegen des enormen Zeitaufwands und der hohen Komplexität.
- Die Informationen in der Software sind nicht genau genug, ständige Telefonanrufe am Institut für Tropenmedizin sind erforderlich. Speziell bei Kombinationen von Reisen bzw. Malariaphylaxe.
- Besser wäre eine Onlinelösung
- Datengrundlage sehr gut
- Zu umfangreich für Parteien
- Ich verwende zusätzlich die Software „RTAP“, da orblmmun für einige Bereiche meiner Arbeit nicht verwendbar ist.

Gerade diese letzten Anregungen sind sehr wertvoll, da sie viele konstruktive Vorschläge und Ideen enthalten. Viele der gewünschten zusätzlichen Features sind vermutlich leicht zu implementieren. Wenn die Attraktivität und die Usability durch solche Zusatzfunktionen gesteigert wird, ist eine Realisierung einzelner Komponenten (oder eine Anpassung bestehender Komponenten) in Betracht zu ziehen.

4.4 Empfehlungen

In Anbetracht der Anzahl der gefundenen Verbesserungsmöglichkeiten im Bereich Design und Ergonomie halten die Evaluatoren eine umfangreiche Verbesserung der Software für unumgänglich. Vor allem das UI Design, welches als „Multi Document Interface“ mit mehreren aufeinander folgenden Dialogen gelöst wurde, sollte überdacht werden: Da der Arbeitsvorgang aus mehreren sequentiellen Schritten besteht, wäre auch ein Assistentendesign in Betracht zu ziehen. Die derzeitige Lösung, welche ein Rückwärtsnavigieren nur zum ersten Dialog ermöglicht und dabei auch noch einen Teil der bereits eingegebenen Daten verwirft ist aus Usability - Gesichtspunkten sehr kritisch zu sehen, genauso wie das Fehlen einer Hilfefunktion. Unter Berücksichtigung der Tatsache, dass orbImmun auf veralteten Technologien beruht (orbImmun wurde 1994 entwickelt und verwendet noch ein diskettengestütztes Lizenz- und Updatesystem sowie eine Visual Fox Pro Datenbank), sind die Evaluatoren der Meinung, dass eine vollständige Neuentwicklung die effizienteste Art wäre das Programm wieder attraktiv zu machen.

Dies sollte zusätzlich noch anhand einer Aufwandsschätzung abgeklärt werden.

5. Diskussion

Bei der internen Evaluation bestand die Schwierigkeit, dass kein Mediziner (oder noch besser: Doppelsexperte) im Evaluatorenteam eingeschlossen war. Daher konnte die Software inhaltlich nicht überprüft und evaluiert werden. Es wurde versucht, dies mit Hilfe der Fragebogenevaluation zu kompensieren (fast alle Befragten sind Ärzte, allerdings keine Spezialisten auf dem Gebiet), was nach der Meinung des Autors nicht ausreichend ist. (siehe auch Abschnitt II, Kapitel 5.3 Evaluatoren) Außerdem war die Anzahl der Evaluatoren mit zwei Personen knapp bemessen. Ideal für dieses Projekt wäre nach Meinung des Autors eine Anzahl von drei Evaluatoren (siehe auch Nielsen 1994).

Dafür war der Einsatz der beschriebenen Usability-Heuristiken äußerst hilfreich. Wenn auch vieles davon für erfahrene IT Spezialisten Allgemeinwissen ist, fällt es mit Hilfe dieser Heuristiken viel leichter, die gefundenen Probleme systematisch zu betrachten. Dies ist auch wichtig für die anschließende Priorisierung.

Die externe Evaluation hätte eventuell auch postalisch erfolgen können. Die höhere Personenanzahl hätte möglicherweise die Akzeptanz und die Genauigkeit der Ergebnisse verbessert. Zwar traten keine Probleme auf, aber die Effizienz der Evaluation könnte gesteigert und die IsoMetrics Fragebögen (Willumeit 1996) in gleicher Art und Weise eingesetzt werden.

Die Durchführung der Telefonbefragung verlief reibungslos, allerdings konnte erwartungsgemäß nur ein geringer Prozentsatz der ausgewählten Testpersonen für die Befragung gewonnen werden.

Abschließend betrachtet wurde die Evaluation der Software erfolgreich durchgeführt und lieferte ein brauchbares und nachvollziehbares Ergebnis, welches vom Auftraggeber unter Zustimmung abgenommen wurde.

IV. Verzeichnisse

1. Bibliographie

- Albers, Haßmann, Somm, Tomczak (2003). *Verkauf*, Symposion.
- Armstrong J. S., Lusk E. J. (1988). "Return postage in mail surveys." *Public Opinion Quarterly* 51: S. 233 - 248.
- Bortz J., Döring N. (1995). *Forschungsmethoden und Evaluation für Sozialwissenschaftler*, Springer.
- Doost, Roger K. (1992). *ROI revisited*. New York, The CPA Journal Online.
- Gediga G., Hamborg K.-C. (1999). IsoMetrics: Ein Verfahren zur Evaluation von Software nach ISO 9241-10. H. Holling & G. Gediga: *Evaluationsforschung*, Hogrefe: S. 195 - 234.
- Gediga G., Hamborg K.-C. & Duntsch I. (1999). "The IsoMetrics Usability Inventory: An operationalisation of ISO 9241-10." *Behaviour and Information Technology* 18: S. 151 - 164.
- Gediga G., Hamborg K.-C., Willumeit Heinz (1998). *Das IsoMetrics Handbuch* (Version 1.15a), Osnabrücker Schriftenreihe Software-Ergonomie.
- Global Information, Inc. (2003). AMPU Not ARPU: A better metric for the wireless industry. <http://www.gii.co.jp>, 2.6.2003.
- Hassenzahl M., Platz A., Burmester M., Lehner K. (2000). *Hedonic and Ergonomic Quality Aspects Determine a Softwares Appeal.*, Addison-Wesley.
- Hassenzahl M., Prümper J. (1999). "Probleme und Konzepte software-ergonomischer Beratung am Beispiel eines Informationssystems im World Wide Web (WWW)." *Informatik - Forschung und Entwicklung* 14: S. 145 - 153.
- Hassenzahl M., Prümper J. (2002). "The effect of perceived hedonic quality on product appealingness." *International Journal of Human-Computer Interaction* 13: S. 479 - 497.
- Heinsen, Vogt (2003). *Usability praktisch umsetzen*, Carl Hanser Verlag München Wien.
- Hewett, Baecker, et al. (1992). "ACM SIGCHI Curricula for Human-Computer Interaction." <http://www.attrakdiff.de> (29.12.2003). AttrakDiff™, User Interface Design GmbH, Marc Hassenzahl (Universität Darmstadt).
- <http://www.isometrics.uni-osnabrueck.de/> (29.12.2003). IsoMetrics - Development of a software usability instrument, K.C. Hamborg (Universität Osnabrück).
- IBM (2003). *Cost justifying ease of use*.
- Kahn, M. J., Prail, A. (1994). *Formal Usability Inspections*, John Wiley and Sons.
- Landauer, T.K. (1996). *The Trouble with Computers*. Cambridge, MA, London, MIT Press.
- Lewis, Polson, Wharton, Rieman (1990). *Testing a walkthrough methodology for theory based design of walk-up-and-use interfaces.*, ACM CHI '90 Conference: 235 - 242.
- Lienert G., Raatz U. (1994). *Testaufbau und Testanalyse*, PVU.
- Nielsen, Jakob (1994). Heuristic Evaluation. *Usability Inspection Methods* S. 25-63. J. N. R. L. Mack, John Wiley and Sons.
- Nielsen, Jakob (2002). Intranet Usability: The Trillion-Dollar Question. <http://www.useit.com/alertbox/20021111.html>, 11.11.2002.
- Nielsen, Jakob (1993). *Usability Engineering*, Morgan Kaufmann / Academic Press.

- Nielsen, Jakob (2003). Severity Ratings for Usability Problems.
<http://www.useit.com/papers/heuristic/severityrating.html>, 20.10.2003.
- Nielsen, Molich (1990). "Heuristic Evaluation of User Interfaces." *ACM CHI '90 Conference*: 249-256.
- Norman, Donald (1988). *The Psychology of Everyday Things*. New York, Basic Books.
- Norman, Donald (1990). *The Design of Everyday Things*. New York, Doubleday Books.
- Ollermann, F. (2001). *Evaluation von Hypermedia-Anwendungen*, Diplomarbeit, Uni Osnabrück.
- Pürer, H. (2001). *Grundbegriffe der Kommunikationswissenschaft*, UVK Verlagsgesellschaft.
- Ruser, Cornelia (2003). *New HCI Visions and Future PDAs*. Zürich, Seminar aus dem Institut für Informatik, Universität Zürich.
- Schafer, J. L. (1997). *Analysis of Incomplete Multivariate Data*, Chapman & Hall.
- Shneiderman, Ben (1998). *Designing the User Interface: Strategies for Effective Human-Computer-Interaction*, Addison-Wesley Longman, Inc.
- Sven Heinsen, Petra Vogt (2003). *Usability praktisch umsetzen*, Carl Hanser Verlag München Wien.
- Thaller, Georg Erwin (2002). *Software-Test, Verifikation und Validation*. Hannover, Heinz Heise GmbH & Co KG.
- Weers, Michael (2000). Web-Usability. <http://www-cg-hci.informatik.uni-oldenburg.de/~airweb/Seminarphase/MichaelWeers/html/>, 18.05.2000.
- Willumeit, K.C. Hamborg; G. Gediga; Heinz (1996). "IsoMetrics - Fragebogen zur Evaluation von graphischen Benutzungsschnittstellen." *Universität Osnabrück - Fachbereich Psychologie* Version 2.00.

2. Abbildungen

Abbildung 1, „Nature of HCI“ (Hewett, Baecker et al. 1992)	11
Abbildung 2: Beispiel aus IsoMetrics ^S Version 2.03 (Willumeit 1996).....	29
Abbildung 3: Beispiel aus IsoMetrics ^L Version 2.03 (Willumeit 1996)	30
Abbildung 4: AttrakDiff Beispiel (http://www.attrakdiff.de 29.12.2003).....	31
Abbildung 5: orbImmun Lizenzdiskette 1	42
Abbildung 6: orbImmun Lizenzdiskette 2.....	42
Abbildung 7: orbImmun Lizenzdiskette 3.....	42
Abbildung 8: orbImmun Druckvorschau 1	43
Abbildung 9: orbImmun Druckvorschau 2	44
Abbildung 10: orbImmun „zurück“-Funktion 1	45
Abbildung 11: orbImmun „zurück“-Funktion 2.....	45

3. Formeln und Gleichungen

Gleichung 1, „TCO“ (Heinsen 2003).....	18
Gleichung 2: Dupont Formel	19
Gleichung 3: ROI in Prozent.....	19

4. Tabellen

Tabelle 1 „DIN EN ISO 9241“ nach (Weers 2000)	20
Tabelle 2 „Expertenevaluation“ nach (Heinsen 2003)	21
Tabelle 3: Beispiele für Fragen aus dem orbImmun Evaluationsfragebogen.....	40

V. Index

A

Abbruch.....	34
Abkürzungen.....	35, 36
Abonnenten	32
Absolutkalkulation	18
Absturz	47
Acht Goldene Regeln	33
Adaptivität.....	35
Adresse.....	48
Aesthetic	36
Aggregation.....	25
Aktivitäten.....	26
Aktualität.....	46
Akzeptanzproblem	24
Alter	48
Alternativen.....	38
Amazon	15
AMPU	16
Anfragevolumen.....	17
Anmerkungen.....	48
Anpassbarkeit	34
Anschaulichkeit	26
Antikoagulierte.....	48
Antwortskalen	28
Anwendungsfehler	9
Anwendungsfelder	11
Apple.....	24
Applikation	48
Arbeitsablauf.....	39, 41, 46
Arbeitsabläufe	46
Arbeitsatmosphäre	26
Arbeitsphasen.....	38
Arbeitspsychologie.....	29
Arbeitsschritte	46
Arbeitsumgebung	34
ARPU	16
Ärzte.....	2, 32, 40

Ärztchammer	2, 32
Assistentendesign.....	44
AttrakDiff.....	27, 30
Attraktivität	30, 49
Aufbau der Arbeit	6
Auffindbarkeit.....	46
Aufgaben.....	22, 23
Aufgabenstellung	24
Aufgabenverarbeitung.....	22
Auftraggeber	14, 25
Aufwandsschätzung	25, 49
Ausführung	39
Ausgaben	17
Auswahlmenü	37
Auswertung	28, 29, 39
average revenue per user.....	16

B

Batch-Dateien	9
Beanspruchung des Benutzers	39
Bearbeitungsdauer.....	46
Bearbeitungszeit.....	46
Bedienoberfläche	46
Befehlsketten	37
Befehlszeileninterpreter	9
Befragung.....	28
Befragungen.....	33
Beispielsysteme	11
Benutzer	25
Benutzerclubs.....	25
Benutzerfehler.....	35
benutzerfreundlich	7
Benutzerfreundlichkeit.....	32
Benutzerprofile	37
Benutzerschnittstellen	19
Benutzungsfrequenz.....	46
Beratungsarchiv	47
Bereiche	24

Berichterlegung	30
Betriebssystem	13
Betriebssysteme	24
Bildschirmarbeitsverordnung	20
Bildschirmrichtlinie	20
Bookmarking.....	35
Browser Standards	35
Büroarbeit	19
Bürotätigkeiten.....	20
Button.....	48
Buttons	35
C	
C2C	15
Call Center	17
CEN	19
Checklisten.....	21
CHI.....	9
Chunking.....	36
close box	43
Cognitive Walkthrough.....	22
Comité Européen de Normalisation	19
Command dialogues.....	20
computer-human interaction	9
confirmation box	35
consistency	36
Consistency	35
constraints	39
D	
DATech.....	21
Datenauswertung.....	30
Datengrundlage	48
Datenmengen	27
Datensicherheit.....	28
Dauer der Benutzung	46
Daumenregel des Marketing	17
Definition der HCI	10
Design	25, 33, 36, 38
Design - Herangehensweise	11
Designer	14
Designgrundsätze	20

Designkonventionen	41
Designmängel	49
Deutsche Akkreditierungsstelle Technik e.V.....	21
Deutsches Institut für Normung	19
Dialog-Architektur	11
Dialoge.....	24
Dialog-Gattung	11
Dialog-Techniken	11
Dialogue Principles	20
Dienstleistungen.....	16
DIN	19
DIN 17025	21
DIN 45011	21
DIN EN ISO 9241.....	19, 20
DIN ISO/IEC 12119	20
Direct manipulation dialogues	20
Diskettenlaufwerk	42
Diskussion.....	25, 49
Dokumentation.....	36
Doppelexperten	24
DOS	9
Druckvorschau	43, 44
Dupont Formel	19
durchschnittliche Einkunft pro Benutzer	16
E	
e.mail Befragungen	40
eBay	15
e-commerce.....	15
Effektivität	12
Efficiency	8
Effizienz.....	8, 12, 22, 35
Einarbeitungszeit.....	17
Eingabefeld	37
Eingabeformular	47
Einheitlichkeit	36
Einprägsamkeit	8
Einschätzungsskala	24
Einschränkungen.....	39
Einsparungspotential.....	17
elektronische Beschwerde.....	13
Empfehlungen.....	33, 49

Empirische Verfahren	21	externen Evaluator	24
EN	19	F	
Entwickler	13, 14	Fachausdrücke	47
Entwicklung	17	Fachvokabular	24
Entwicklungsprozess	17	Fachwissen	24
Entwicklungsteam	23, 25	Fallbeispiel	32
Erfolgssicherheit	15	Fallstudien	11
Ergebnisdarstellung	25	Features	41, 48
Ergebnispräsentation	40	Feedback	13, 34, 37, 38, 40, 46
Ergebnisse	41	Feedback Mechanismen	13
Ergonomic requirements for office work with visual display terminals	20	Fehleingabe	45
Ergonomie	11	Fehler	8
Ergonomiemängel	49	Fehlerbehandlung	46
ergonomische Anforderungen	19	Fehlerbericht	13
Ergonomische Anforderungen für Büroarbeit mit VDT's	20	Fehlercodes	36
ergonomische Qualität	30	Fehlermeldungen	43, 47
Erlernbarkeit	8, 22	Fehlerrate	21
error box	43	Fehlertoleranz	34
Error prevention	35	Fehlervermeidung	37
Errors	8	Fehlervorbeugung	35
EU 90/270 EWG	20	Findings	24, 41
EU-Bildschirmrichtlinie	20	Flexibilität	35
EU-Richtlinie 90/270/EWG	19	Flexibility	35
Evaluation	21, 39	Fokusgruppe	26
Evaluation von Usability	7	Fokusgruppen	25
Evaluationsbericht	32	Form filling dialogues	20
Evaluationsbogen	39, 46	Formale Usability-Inspektionen	22
Evaluationsdauer	23	Formulardesign	41
Evaluationsfragebogen	29	Fragebögen	33
Evaluationsfragebögen	27	Fragebogenbefragungen	27
Evaluationstechniken	11	Fragebogenuntersuchung	27
Evaluationsteilnehmer	46	Fragenkatalog	33, 39
Evaluatoren	22, 23, 24	Fragestunde	26
Execution	39	Freitext	40
Experten	21	Frequenz	12
Expertenevaluation	21 , 33	Funktionalität	47
Exploration	22, 24	G	
Externe Ergebnisse	46	Gebrauchstauglichkeit	7, 30
Externe Evaluation	33, 40	Geburtsdatum	48
		Gedächtnisstützen	38

Gesamtbetriebskosten	18
Gesamtergebnis	47
Gesamtertrag	18
Geschäftsabwicklungen	16
Gesetzestexte	19
Gestaltungsempfehlungen	23
Gestaltungsprinzipien	20, 23
Gestaltungsrichtlinien	22, 24
Gestaltungsvorschläge	23
Gewicht des Patienten	47
Gewinn	18, 19
Gewinnmarge	16
Goldenen Regeln des Interface Designs	36
graphische Benutzeroberflächen	9
GUI	9
GUI Design	44
Guidance on Usability	20

H

Handlungsabläufe	22
Hartnäckigkeit	12
Hauptgütekriterien	27
HCI	9
hedonische Qualität	30
hedonische Qualitäten	30
Heuristiken	21, 22, 23, 24, 41
Heuristische Evaluation	22
Heuristischer Walkthrough	23
Hilfe	36, 46
Hilfebutton	43
Hilfefunktion	36, 47
Hilfesystem	37
HMI	9
Human-Computer Interaction	9
human-machine interface	9

I

Icons	20
idea seeding	26
Ikonografie	36
Implementierung	17
Implementierungstechniken	11

Informationsangebot	46
Informationsdesign	32
Initiator	28
Inspektionsmethoden	21
Inspektoren	22
Installation	41, 47
Intention	39
Interaktion	9, 36
Interaktionselemente	22
Interessentenstamm	17
Interface	22, 24, 25, 34, 37
International Standards Organisation	19
Interne Ergebnisse	41
interne Evaluation	46
Interne Evaluation	33, 41
Interne Evaluatoren	24
Internet	17, 28, 41, 48
Internetbefragungen	28
Internetupdate	48
Intervall	48
Interviewer	28
Interviews	28, 32
Investition	18
Investitionsvolumina	18
ISO	19
ISO 13407	20
ISO 14915	20
ISO 9241 Teil 10	29
ISO Homepage	19
ISO Norm	
9241-11	7
ISO Norm 9241 Teil 10	30
ISO/IEC 11581	20
IsoMetric	33
IsoMetrics	27, 28, 29, 39
IsoMetrics ^L	29
IsoMetrics ^S	29
Items	27
Itemwerte	28
iterative Softwareentwicklung	30

J

Joy of Use 12

K

Kalkulation 18

Kauffrequenz 16

Kombinationen von Reisen 48

Kommunikation 9

Kommunikationsformen 40

Kommunikationskosten 17

Komplexität 9, 48

Konformität 21

konsistentes Systemabbild 38

Konsistenz 35

konstruktive Hilfe 36

Konsument 14

Kontrollmöglichkeit 37

Konzepte 34

Konzeptionelles Modell 38

Kopierschutz 41

Korrektur 15

Kosten 18

Kosten und Nutzen **16**

Kostenalternative 18

Kostenoptimierung 18

Kostensenkung 17

kritischer Abbruch 42

Kunden 16

Kundenservice 17

Kurzzeitgedächtnis 37

L

Lauffähigkeit 34

Layout 41

Learnability 8

Lernsimulationen 37

Links 35, 48

Lizenzdiskette 40, 41, 42, 47

M

Macintosh Human Interface Guidelines 24

MacOS 9

Mailclient 46

Makroprogrammierung 36

Malariaprophylaxe 48

man-machine interface 9

mapping 38

Mapping 38, 39

market impact 12

Maus 9, 13

MDI 44

MDI Design 45

Media Player 13

Memorability 8

Menschliche Informationsverarbeitung 11

Mensch-Machine Fitness und Adaption 11

Mensch-Maschine Interaktion 9

Mensch-Maschine Kommunikation 9

mentales Modell 38

Menu dialogues 20

Menüleiste 43

Menüsteuerung 13

Messagebox 41, 42

Messgrößen 16

Meta Modelle der HCI 11

Metaphern 34

Methoden 21, 22

Microsoft 2, 13, 24, 32

Microsoft Windows User Experience 24

Mindeststandards 20

minimalist 36

missing data 28

Missverständnisse 13

MMI 9

Mobile Applications 22

mobile market 16

mobile operators 16

Modelle der HCI 11

Moderation 26

Moderator 25

Moderatoren 22

Motivation **13**

Multimediaanwendungen 30

multimediale Inhalte	20
Multiple Document Interface	44
Mundpropaganda	17

N

Nadelöhr der HCI.....	9
Natur der HCI.....	11
Navigation.....	22, 34, 46
Netto-Ansatz	18
Neuprogrammierung	45
Normen	19, 20
Normungsinstitute	19
Nutzereigenschaften.....	23
Nutzergruppen.....	21
Nutzerklassen.....	23
Nutzerkontrolle	34
Nutzerprofile	22
Nützlichkeit.....	27

O

Objektivität	27
Ökonomie.....	27
OMI.....	9
Onlinelösung	48
Onlineshops.....	16
operator-machine interface.....	9
orbImmun.....	13, 32
Orbimmun	32, 40, 41, 44, 46, 48, 49
Orbimmun Redesign	32
Organisationspsychologie	29
Orientierung	34

P

Papier-ausdrucke.....	47
Paradoxon der Technologie.....	9
Patient	46
Patienten.....	2, 32
Patientengewicht	47
PC-Arbeitsplatz	20
persistence.....	12
Personenzahl	27
Pflege	17

Pharmafirmen.....	48
Planbarkeit	15
Pluralität	26
Polung	28
Post	28
Pragmatische Qualität	30
Praktikabilität.....	39, 41, 47
Präsentation.....	23, 33
Presentation of information.....	20
Prinzipien	38
Priorisierung.....	12, 24
Priorität	44, 45
Problemfelder.....	25
Problemgehalt	29

Produkt

Eignung.....	7
Produkte	16
Produktentwicklung	22
Produktentwicklungsprozess.....	17
Produktivität.....	17
Produktkriterien	12
Profilanalyse	29
profit	16
Profit	16
Programmende	43
Programmfehler	35
Programmierer	13
Projektausschreibung	32
Projektbeschreibung.....	32
Projektleiter.....	32
Prototypen.....	21, 30
Prototypen Evaluation.....	30
Prüflabor	21

Q

qualitative Daten	29
Qualitätskriterien.....	27
Quantifizierungen	26
quantitative Daten	28, 29
Question	42

R

Ratingskala.....	28
recall.....	35
Recognition	35
redo	34
Reiseberatung.....	2, 32
Reiseimpfprogramm.....	2, 32
Reiseimpfungsdatenbank	32
Reisekosten	17
Reliabilität.....	27
Rendite einer Investition	18
Repräsentative Nutzer	21
Reset-Button	37
Resultate.....	38
Return on Investment	18
revenue.....	16
Rezepterstellung.....	2, 32
Risiken	15
ROI.....	17, 18 , 19
RTAP	48
Rücklauf.....	28
Rückmeldung	22, 37
Rückumschläge	28

S

Satisfaction	8
Satzbau.....	34
Schnittstellen.....	9
Schnittstellenarchitektur.....	11
Schriftgröße.....	36
Schulnotensystem.....	46
Seiten.....	24
Selbstselektion	28
Server Version.....	48
Sessions pro Woche	45
short messaging service.....	16
shortcuts	36
Sichtbarkeit	34
Skala.....	40, 46
SMS	16
Software	19, 24, 25, 33, 40

Software Agent	36
Software Design.....	45
Software Engineering.....	9
Software Ergonomie	9
Softwareentwicklung	14, 30
Softwareentwicklungsprozess	13
Software-Entwicklungsprozess	14
Software-Evaluationsfragebögen	27
Softwareprodukte	33
Softwareprojekt.....	19
Softwarequalität	20
Spam	40
Splash Screens	40
Sprache	34
Standardisierung	35, 39
standards	35
Standards.....	35
Startkosten	19
Status.....	38
Steigerung der Produktivität	17
Steuerung	38
Stichwortverzeichnis.....	48
Streuung	46, 47
Stufen	
für die Schwere eines Usability Problems.....	13
Styleguides.....	24
subjektive Zufriedenheit	21
Subskalen	27, 30
Suchfunktion	35
Support.....	17
Support-Aufwand.....	17
Symbol	42
Symbole	20
System image	38
system status	34
Systemabbild.....	38
Systemfehler	47
Systemstatus.....	34
Systemzustand	39
Szenarien.....	22, 23

T

Tastatur	9, 13
Tastenkombinationen	13
TCO	18
TCO-Wert	18
Teilimpfung.....	48
Telefon	40
Telefonbefragung	40
Telefonberatung	46
Telefoninterviews.....	33
Telefonnummer	48
Terminologie	36, 46
Terminvereinbarung	40
Testaufgaben	30
Testpopulation.....	30
Testtheorie.....	27
Test-User.....	33
Themen der HCI.....	10
Tippfehler.....	37
Total Cost of Ownership	18
Touchscreen	9
Trade-Offs	26
Tropenmedizin	48

U

Übersichtlichkeit	39, 41, 46, 47
Umfang	47
Umsatz	15
undo	34
UNIX.....	9
Update	32, 48
Usability	7 , 33, 49
Definition	7 , 8
Faktoren	7
Kriterien, objektiv / subjektiv.....	8
Usability - Anforderungen	46
Usability Business Case	18
Usability Engineering	9 , 25
Usability Evaluation.....	32
Usability Heuristics	33
Usability Katastrophe.....	13, 41, 45

Usability Problem	41
Usability Probleme.....	12
Usability Review.....	30
Usability Testing	21
Usability-Bewertung	21
Usability-Engineering	20
Usability-Evaluation	32
Usability-Inspektion.....	21
Usability-Inspektionen.....	22
User centered Design	9
User control.....	34
user friendly	7
User Guidance.....	20
User Interface.....	21
User Interfaces	21

V

Validität	27
VDTs.....	20
Verbesserungsvorschläge.....	32, 48
verkaufsfördernder Effekt	15
Verlust.....	19
Verständlichkeit	46
Vertragsbestandteile.....	19
Visibility	34
Vorstellungsrunde	26
Vorteile von Usability	15

W

Wartung	17
Wartungsaufwand	17
Weiterbildung	17
Weiterbildungsmaßnahmen	17
Wettbewerb.....	14
Wettbewerbsfähigkeit	17
Wettbewerbsvorteil	16, 17
Wiedererkennung	35
WIMP	9
Windows	2, 9, 32
Windows Symbol.....	42
Windows XP	13
Windows, Icons, Maus, Pointer	9

Wirkung	12
Wirtschaftliche Aspekte	16
wirtschaftliche Faktoren	17
Wissen	38
Wunschkarten	26

Z

Zeigegerät	13
Zeitaufwand	48
Zertifikat	21
Zertifizierer	21

Zertifizierung	19, 21
Zertifizierungsschema	21
Zertifizierungsstelle	21
Zielgruppe	17, 23
Zufriedenheit	12
Zufriedenstellung	8
Zurück zur Eingabe	44, 46, 48
Zusatzfunktionen	49

Curriculum Vitae

30. Oktober 1978: geboren in 6800 Feldkirch

1985-1989: Volksschule

1989-1997: Bundesgymnasium Gallusstraße in Bregenz

Juni 1997: Matura (in den Fächern Deutsch, Englisch, Französisch, Mathematik, Biologie und Chemie)

1997 / 1998: Präsenzdienst in der Dauer von 8 Monaten

WS 1998: Beginn eines Studiums der Medizin an der Universität Innsbruck

Januar 2002: Einstellung bei der Firma „holzweg e-commerce solutions“ (Dipl. Inform. Univ. Christoph Holz) in Innsbruck als „Software-Entwickler“

August 2002: Beginn des Softwareentwicklungsprojektes „Acoustic Selling ® AudioManager“ mit dem Auftraggeber Antenne Soundworks (Schwarzach, Inhaber: Marnus Flatz)

WS 2002: Wechsel vom Medizinstudium zum Studium der medizinischen Informatik an der UMIT

30. November 2002: Beendigung des Arbeitsverhältnisses bei der Firma „holzweg“ und Vorbereitung auf die Selbstständigkeit

März 2003: Anmeldung des Gewerbes „Dienstleistungen in der automatischen Datenverarbeitung und Informationstechnik“

Ab Februar 2005: Teilnahme an den Projekten „Forschungsnetzwerk“ und „wearable computing“ an der UMIT als studentischer Mitarbeiter unter der Leitung von Univ. Prof. Dr. Paul Lucovicz

Hiermit erkläre ich an Eides statt, die Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet zu haben.

Matthias Gruber
Innsbruck, am 17.03.2005