



MAGISTERARBEIT

Konzeption, Implementierung und Evaluierung eines Web-Tools zur Unterstützung von verteilten heuristischen Evaluierungen.

zur Erlangung des akademischen Grades
Magister
(Mag. rer. soc. oec.)

ausgeführt am
Institut für Rechnergestützte Automation
Forschungsgruppe Industrial Software

der Technischen Universität Wien

unter der Anleitung von
Univ.-Prof. Dipl.-Ing. Dr. Thomas Grechenig und
Mag. Thomas Költringer

durch
Johannes Ortmann

Anschrift
Sieveringerstr. 103/4/3, 1190 Wien

Wien, 30.04.2007

Abstract

Bei der Entwicklung von Software kommt dem Usability Engineering und hier insbesondere dem User Centered Design Ansatz zentrale Bedeutung zu. Dabei stellt die heuristische Evaluierung die am häufigsten verwendete Inspektionsmethode dar, da sie von den frühen Stadien der Entwicklung bis hin zum fertigen Produkt angewendet werden kann, und somit besonders kosteneffizient ist. Bei einer heuristischen Evaluierung testen mehrere EvaluatorenInnen ein Interface, und beurteilen dessen Kompatibilität mit zuvor festgelegten Usability Prinzipien (Heuristiken).

Im Rahmen der vorliegenden Masterarbeit wurde ein Tool entwickelt, das Usability-ExpertenInnen bei der heuristischen Evaluierung von User Interfaces unterstützt. Das entwickelte Tool erfüllt zwei Funktionen: Einerseits dient es als zentrale Informationsplattform bei heuristischen Evaluierungen, andererseits beschleunigt es administrative Abläufe. Implementiert wurde diese Webapplikation in PHP 4 unter Verwendung von HTML 4.01 und CSS, sowie einer relationalen Datenbank. Die Funktionen des Tools unterstützen alle Aspekte der heuristischen Evaluierung und ermöglichen es den EvaluatorenInnen, ein Projekt gemeinschaftlich im Internet zu evaluieren.

Der Einsatz des Tools ist allerdings nicht nur auf das Web beschränkt, sondern kann auch als Intranetlösung eingesetzt werden. Es dient dabei zur zentralisierten Speicherung, erleichterten Auswertung und administrativen Unterstützung der ExpertenInnen, welche die heuristische Evaluierung durchführen. In Summe zeichnet sich das Tool vor allem durch die schnelle und effiziente Dokumentation von Usability Problemen aus, wodurch die Effektivität der Problemfindung signifikant gesteigert wird. Vor Abschluss der Implementierung wurden zwei Evaluierungsverfahren, ein Software Quality Test und danach ein User Acceptance Test, angewandt und deren Resultate umgesetzt. Das hier entwickelte Webtool erfüllt somit alle gestellten Anforderungen und trägt maßgeblich zu einer zeit- und kosteneffizienten Evaluierung bei.

Abstract

Usability engineering, particularly user centered design, plays a decisive role in software development. Specifically heuristic evaluations are considered to be the most popular inspection method, practiced by a small set of evaluators who examine an interface and judge its compliance with the previously defined principles (heuristics). Its crucially important advantage is the possibility to apply it both in early and in final stages of development, thus rendering this method significantly more cost-effective than other methods.

This thesis is motivated by the idea of supporting usability experts in the process of distributing heuristic evaluations of user interfaces. This was accomplished through the development of a new web application to enable centralized document management as well as to optimize administrative procedures.

This web application is implemented in PHP 4 and utilizes HTML 4.01 and CSS in addition to a MySQL database. It supports all aspects of heuristic evaluation processes and enables usability experts to evaluate a project jointly on networks such as the Internet. By allowing centralized storage of all necessary data, the application facilitates analyses and ensures administrative support for the usability experts performing a heuristic evaluation. The tool's main advantage, however, clearly consists in the possibility of documenting usability problems more efficiently.

During its development, the prototype was evaluated in the context of a software quality test as well as a user acceptance test. The conclusions drawn from these examinations formed the basis for further improvement in the tool's performance. As a result, the presented final version of the web tool meets all defined requirements and proved to be particularly time- and cost-effective in heuristic evaluation processes.

Danksagung

An dieser Stelle möchte ich all jenen danken, die durch ihre fachliche und persönliche Unterstützung zum Gelingen dieser Diplomarbeit beigetragen haben. Besonderer Dank gebührt meinen Eltern, die mir dieses Studium durch ihre Unterstützung ermöglicht haben.

Weiters bedanke ich mich bei Herrn Univ.-Prof. Dipl.-Ing. Dr. Thomas Grechenig und Mag. Thomas Költringer für die Betreuung meiner Diplomarbeit und die zahlreichen wissenschaftlichen Ratschläge, welche stets zur Verbesserung der Arbeit beigetragen haben. Schlussendlich sei all jenen gedankt, die mich während meines Studiums unterstützt und gefördert haben.

Inhaltsverzeichnis

1.	Einleitung	7
1.1.	Motivation.....	8
1.2.	Problemstellung	9
1.3.	Struktur der Arbeit.....	10
2.	User-Centered Design.....	11
2.1.	Usability Engineering – Definitionen.....	11
2.2.	User-Centered Design vs. System Centered Design	13
2.3.	User-Centered Design Prozess	14
2.4.	Methoden des User-Centered Design Prozesses	18
3.	Usability Inspection Methods.....	20
3.1.	Inspektionsmethoden im Software Entwicklungsprozess	20
3.2.	Inspektionsmethoden im Usability Engineering.....	21
3.3.	Beschreibung von Inspektionsmethoden	24
3.3.1.	Heuristic Evaluation	24
3.3.1.1.	Durchführung	24
3.3.1.2.	Diskussion und Bewertung.....	27
3.3.2.	Cognitive Walkthrough	29
3.3.3.	Heuristic Walkthrough	31
3.3.4.	Guideline Reviews	32
3.3.5.	Plurastic Walkthroughs	33
3.3.6.	Consistency Inspection	34
3.3.7.	Formal Usability Inspektion	34
3.4.	Zusammenfassung.....	37
4.	Entwicklung eines Tools zur Unterstützung der „Heuristic Evaluation“	39
4.1.	Anforderungen	39
4.2.	Prozess	40
4.2.1.	Vorbereitung	41
4.2.2.	Evaluierung	43
4.2.3.	Konsolidierung	44
4.2.4.	Berichterstellung	45
4.3.	Umsetzung	46
4.3.1.	Entwicklung & Software Architektur	46
4.3.2.	Rollenbeschreibung	48
4.3.2.1.	Rolle des/der ProjektadministratorIn / SuperadministratorIn	48

4.3.2.2.	Rolle des/der EvaluatorIn	49
4.3.3.	User Interface	49
4.3.3.1.	EvaluatorIn	49
4.3.3.2.	ProjektadministratorIn	53
4.3.3.3.	SuperadministratorIn	58
4.4.	Vergleich zu anderen Tools	59
4.4.1.	Mozilla UZReview	59
4.4.2.	QuestionPro	61
4.5.	Prozessunterstützung	62
5.	Evaluierung	64
5.1.	Software Quality Test	64
5.1.1.	Vorgangsweise und Zielsetzung	64
5.1.2.	Ergebnisse & Erfahrungen	65
5.2.	User Acceptance Test	66
5.2.1.	Vorgangsweise und Zielsetzung	66
5.2.2.	Ergebnisse & Diskussion	67
6.	Zusammenfassung und Ausblick	69
7.	Abbildungsverzeichnis	72
8.	Literaturverzeichnis	73
9.	Anhang	76
9.1.	Zehn Usability Heuristiken	76
9.2.	Datenbankmodell	78

1. Einleitung

Um in der Informationstechnologie langfristig wettbewerbsfähig zu bleiben, sind überzeugende Qualität und hohe Kundenzufriedenheit die zentralen Grundpfeiler einer strategischen Ausrichtung der Softwareentwicklung. Da die immer komplexer werdenden Aufgaben auf diesem Gebiet nicht mehr ausschließlich durch das Wissen der konkreten EntwicklerInnen gelöst werden können, sind zunehmend Innovationen und Entwicklungen neuer Anwendungen erforderlich. Dabei sollten die Anwendungen unter Berücksichtigung der Bedürfnisse ihrer zukünftigen BenutzerInnen hin gestaltet werden.

Die heuristische Evaluierung ist eine Methode, bei der die Usability einer bestimmten Benutzeroberfläche anhand gewisser Kriterien beurteilt wird. In der bereits von Jakob Nielsen und anderen UsabilityspezialistInnen beschriebenen Methode wird von einer ExpertInnengruppe versucht, anhand einer Liste mit Heuristiken möglichst viele Probleme in einem Userinterface zu finden und diese exakt zu dokumentieren, sodass eine adäquate Lösung für die aufgezeigten Probleme gefunden werden kann.

Im Vergleich zu klassischen Usabilitytests, bei denen potentielle BenutzerInnen ohne besondere Vorkenntnisse zum Einsatz kommen, bietet die heuristische Evaluierung durch das Heranziehen von ExpertInnen den großen Vorteil, dass sie schon in den frühen Phasen der Entwicklung angewendet kann, und ein schnelles und relativ kostengünstiges Feedback für die Designer darstellt. Daher werden heuristische Evaluierungsmethoden vom ersten Prototyp bis hin zum fertigen Produkt in diversen Entwicklungsstadien eingesetzt, wodurch etwaige Fehler zum frühest möglichen Zeitpunkt eliminiert und somit Zeit und Kosten gespart werden können.

Dennoch sollte eine heuristische Evaluierung keinesfalls als Ersatz für herkömmliche Usabilitytestmethoden verstanden werden, da ExpertInnen häufig gerade aufgrund ihrer umfassenden Vorkenntnisse jene Probleme übersehen, die sich weniger erfahrenen BenutzerInnen stellen.

Während heuristische Evaluierungsmethoden also einen grundlegenden Bestandteil moderner Softwareentwicklung ausmachen, ist deren Durchführung aber oftmals mühevoll und aufwändig. Das in dieser Arbeit beschriebene Web-Tool stellt eine Möglichkeit dar, den Prozess der heuristischen Evaluierung in der Praxis einfacher und effizienter zu gestalten.

1.1. Motivation

Der Verfasser hat im Vorfeld dieser Arbeit selbst einige heuristische Evaluierungen moderiert und organisiert. Aufgrund der dabei gesammelten Erfahrungen wurden zahlreiche Probleme ersichtlich, die bei einer Evaluierung ohne spezifische Softwareunterstützung zu einem beträchtlichen Mehraufwand führen.

Die gravierendsten Nachteile der derzeit verfügbaren Inspektionsmethoden liegen im Zeitmanagement und im dezentralisierten Dokumentmanagement, da die evaluierenden UsabilityexpertInnen sich üblicherweise nicht am gleichen Ort befinden, und außerdem meist unterschiedliche Systeme und Programme verwenden. Daher muss der Prozessverlauf erst mühevoll erarbeitet werden. Obwohl es sich bei unterschiedlichen Evaluierungsprojekten häufig um sehr ähnliche Probleme und Aufgaben handelt, fehlt bislang eine entsprechende Software zur Koordination und Unterstützung der ExpertInnen bei der heuristischen Evaluierung.

Zwar ist durchaus davon auszugehen, dass bei großen Beratungsfirmen interne Tools in der heuristischen Evaluierung Verwendung finden, diese sind jedoch nicht öffentlich zugänglich. In der Opensource-Community trifft man hingegen nur solche Tools an, die spezifisch auf bestimmte Produkte ausgerichtet sind, wie beispielsweise bei Mozilla UZReview.

Ein neutrales und universal einsetzbares Tool zur Unterstützung der heuristischen Evaluierung steht derzeit nicht zur Verfügung, obwohl es den Ablauf von Evaluierungsprozessen zweifellos auf vielfältige Weise verbessern könnte. Das Ziel der vorliegenden Masterarbeit ist somit die Entwicklung eines Web-Tools zur Unterstützung von UsabilityexpertInnen bei der Durchführung diverser Evaluierungsprojekte und zur Vereinfachung und Beschleunigung dieser Evaluierungen.

1.2. Problemstellung

Vor der Implementierung eines solchen Web-Tools musste ein detaillierter Anforderungskatalog erstellt werden, wobei der Administration der Evaluierungsprojekte und der Rollenverteilung eine zentrale Bedeutung zukommt.

Im Hinblick auf die funktionalen Anforderungen war es selbstverständlich wichtig, die Administration heuristischer Evaluierungsprojekte inklusive der dazugehörigen ModeratorInnen und EvaluatorInnen zu gewährleisten. Darüber hinaus sollte das Tool sowohl über auswählbare Heuristiken verfügen, als auch deren individuelle Gestaltung ermöglichen. Ein weiterer wesentlicher Aspekt ist die Mandantenfähigkeit, die die Rolle der am Evaluierungsprozess beteiligten Personen und die sich daraus ergebende Rechteverteilung berücksichtigt. Neben diesen und zahlreichen anderen funktionalen Anforderungen sollten bei der Implementierung des Tools auch nicht-funktionale Kriterien wie Performance, Zuverlässigkeit und Usability berücksichtigt werden.

Nach Erstellung des Anforderungskataloges erfolgte eine Auswahl der zu verwendenden Technologien, wobei die Entscheidung auf PHP als Skriptsprache und MySQL als Datenbank fiel. Ausschlaggebende Kriterien für diese Wahl waren neben optimaler technischer Eignung als Entwicklungsumgebung die Open Source-Verfügbarkeit der gewählten Tools. Im Anschluss daran wurde ein Paper Prototyp entwickelt, sowie die exakten Programmabläufe erstellt.

Im Zentrum der vorliegenden Arbeit steht die Implementierung des Web-Tools, das abschließend einer detaillierten Evaluierung und einer darauf basierenden Optimierung unterworfen wurde.

1.3. Struktur der Arbeit

Kapitel 2 beschäftigt sich mit dem User-Centered Design Ansatz, da dieser die Grundlage für benutzerorientierte Softwareentwicklung darstellt. Im Anschluss daran soll in Kapitel 3 ein kurzer Überblick über diverse Inspektionsmethoden vor allem dazu dienen, Nielsens Heuristic Evaluation, deren Durchführung vom im Rahmen dieser Arbeit entwickelten Web-Tool unterstützt werden soll, gegenüber anderen Methoden zu positionieren.

Auf der Basis dieser allgemeinen Konzeptualisierungen von User-Centered Design und Inspektionsmethoden wird in Kapitel 4 die Entwicklung des Web-Tools näher erläutert, wobei zunächst detailliert auf das zugrunde liegende Konzept eingegangen wird. Danach wird beschrieben, inwiefern die Funktionen des Tools die einzelnen an der Evaluierung beteiligten Personen unterstützen. Darüber hinaus werden die Besonderheiten des Tools im Vergleich zu ähnlichen Projekten, sowie dessen Softwarearchitektur und User Interface Design beleuchtet.

Kapitel 5 der vorliegenden Arbeit diskutiert Evaluierungen, die StudentInnen in mehreren Entwicklungsphasen vornahmen. Den Abschluss der Arbeit bilden eine Zusammenfassung und ein Ausblick in diesem Gebiet.

2. User-Centered Design

Als Einführung in die grundlegenden Konzepte benutzerorientierter Softwareentwicklung wird hier ausführlich der User-Centered Design Ansatz vorgestellt. Dazu werden zunächst verschiedene Definitionen von Usability Engineering angeführt und Unterschiede zu System Centered Design Ansätzen erläutert. Im Anschluss daran werden der eigentliche Prozess des User-Centered Design und dessen Methoden detailliert beschrieben.

2.1. Usability Engineering – Definitionen

Allgemein ausgedrückt beschäftigt sich Usability Engineering mit der Verbesserung der Gebrauchstauglichkeit interaktiver Systeme, es existieren jedoch diverse genauere Definitionen des Begriffs, von denen einige hier präsentiert werden sollen.

Eine allgemeine Definition von Usability Engineering von Jakob Nielsen, einer der bekanntesten Größen auf diesem Gebiet, lautet:

„Usability is the measure of the quality of the user experience when interacting with something – whether a web site, a traditional software application, or any other device the user can operate in some way or another.“ Nielsen, J. (1994)

Jakob Nielsen assoziiert mit Usability fünf Attribute:

Erlernbarkeit. Ein System sollte leicht erlernbar sein.

Effizienz. Das System sollte effizient in der Handhabung sein, sobald UserInnen mit dem System vertraut sind.

Merksbarkeit. UserInnen sollten das System auch nach längeren Arbeitspausen wieder ohne Schwierigkeiten benutzen können.

Fehler. Das System sollte grundsätzlich keine katastrophalen Fehler zulassen, UserInnen möglichst daran hindern Fehler zu machen und so gestaltet sein, dass dennoch auftretende Fehler unkompliziert zu beheben sind.

Zufriedenheit. Das System sollte die Erwartungen der BenutzerInnen hinreichend erfüllen und somit ein Gefühl von Zufriedenheit hervorrufen.

Eine globale Definition von Usability der International Organization for Standardization, die um die Durchsetzung einheitlicher internationaler Standards bemüht ist, lautet:

„Usability eines Produktes ist das Ausmaß, in dem es von einem bestimmten Benutzer verwendet werden kann, um konkrete Ziele in einem definierten Kontext effektiv,

effizient und zufrieden stellend zu erreichen“ (International Organization for Standardization, 1998).

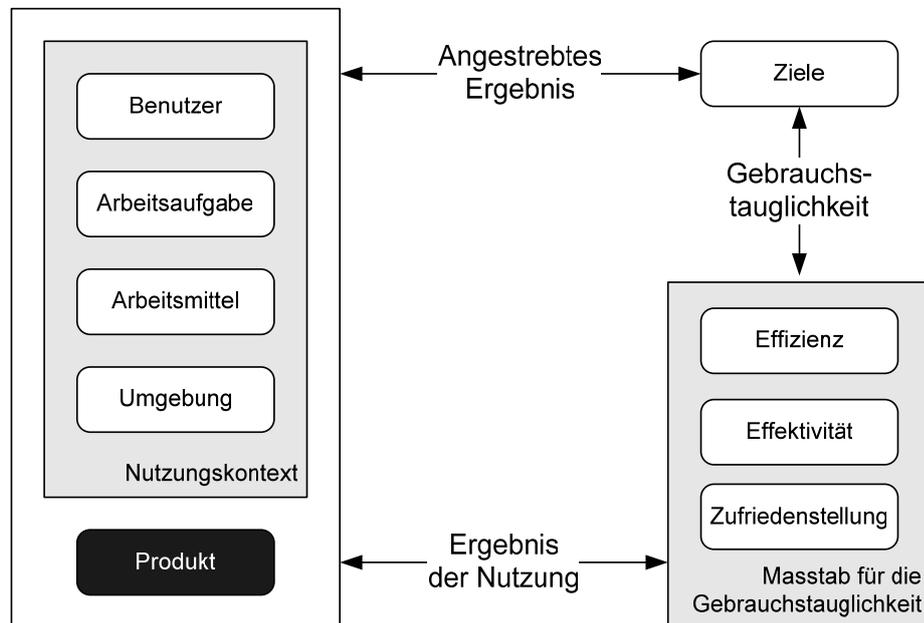


Abbildung 1 – Anwendungsrahmen der Gebrauchstauglichkeit

Usability Engineering beschreibt also den Prozess zur Produktion benutzbarer Software von der Anforderungsanalyse bis zum Support (Abbildung 1). Allerdings muss dieser Prozess keineswegs nur auf die Software bezogen sein. Auch bei Geräten des täglichen Gebrauchs, wie zum Beispiel einer Fernbedienung oder einem E-Herd, kann Usability Engineering eine Rolle spielen und somit jegliches von Menschen benutzte Objekt umfassen.

2.2. User-Centered Design vs. System Centered Design

Donald Norman (2002) erläutert in „The Design of Everyday Things“ anhand einer Reihe von Beispielen seine Ansichten über die Qualität bestimmter Formen von Design. Dabei gelingt es ihm, den Wert eines guten Designs ebenso zu vermitteln wie die Konsequenzen der Fehler eines schlechten Designs. Norman verwendet den Begriff User-Centered Design für ein Design, welches gezielt auf die Anforderungen der UserInnen abgestimmt wurde, statt diese Bedürfnisse anderen Kriterien, wie beispielsweise den Prinzipien der Ästhetik, unterzuordnen.

User-Centered Design umfasst eine Vielzahl verschiedener Modelle und Ansätze, durch deren Anwendung die (Software-)EntwicklerInnen darin unterstützt werden sollen, eine für den/die spätere/n UserIn optimale Produktlösung zu entwerfen. Bei diesen Herangehensweisen werden die Benutzeranforderungen schon von der ersten Phase des Design-Prozesses an in den Lifecycle eines Produkts miteinbezogen. Entscheidend sind hierbei die Miteinbeziehung realer BenutzerInnen in die Produktentwicklung sowie ein iterativer Designprozess.

Die Vorteile eines auf die BenutzerInnen abgestimmten Designprozesses für alle Beteiligten sind klar ersichtlich. Zunächst wirkt sich User-Centered Design für die UserInnen positiv aus, da sich diese an einem fehlerfreien und unkompliziert zu handhabenden Endprodukt erfreuen können, welches sowohl den benötigten Zeitaufwand als auch die Inanspruchnahme von Support minimiert. Dadurch ergibt sich für den/die individuelle/n BenutzerIn neben einer allgemeinen Zufriedenheit mit dem Produkt auch eine potentiell erhebliche Kostenersparnis (Mayhew, 1999).

Aufgrund dieser von den UserInnen wahrgenommenen Qualitäten des Produkts können die EntwicklerInnen mit einem erhöhten Absatz rechnen, was neben der durch eine verringerte Entwicklungszeit entstehenden Kostenersparnisse zu den signifikantesten Vorteilen des Einsatzes von User-Centered Design zählt (Mayhew, 1994). Im Hinblick auf die Entwicklung von Software besteht der Grundgedanke eines User-Centered Design Ansatzes also darin, dass nicht nur die Funktionalität sondern auch die Bedienbarkeit eines Programms maßgeblich berücksichtigt werden, um sowohl für die BenutzerInnen als auch die jeweiligen EntwicklerInnen optimale Ergebnisse zu erzielen.

2.3. User-Centered Design Prozess

Die DIN EN ISO 13407 "Benutzerorientierte Gestaltung interaktiver Systeme" (International Organization for Standardization, 1999) ist eine EU-Norm, die einen prototypischen benutzerorientierten Softwareentwicklungsprozess beschreibt. Erfüllt ein spezieller Entwicklungsprozess die in dieser Norm festgehaltenen Empfehlungen, so kann er als richtlinienkonform betrachtet werden.

Die ISO 13407 wurde im Juni 1999 in ihrer deutschen Fassung als Norm bestätigt. Sie besteht in ihrem Aufbau sowohl aus Beschreibungen der Planung benutzerorientierter Gestaltung, als auch aus Erläuterungen zur Entwicklung interaktiver Systeme, wobei das Hauptaugenmerk auf einer Optimierung der Gebrauchstauglichkeit von Systemen liegt. Diese Erläuterungen beschreiben in prägnanter, übersichtlicher und verständlicher Form einen iterativen Entwicklungsprozess, der vordergründig von NutzerInnen- und Aufgabeneigenschaften bestimmt wird. Außerdem enthält die Norm diverse Richtlinien und Tabellen für das Berichten über benutzerorientierte Aktivitäten.

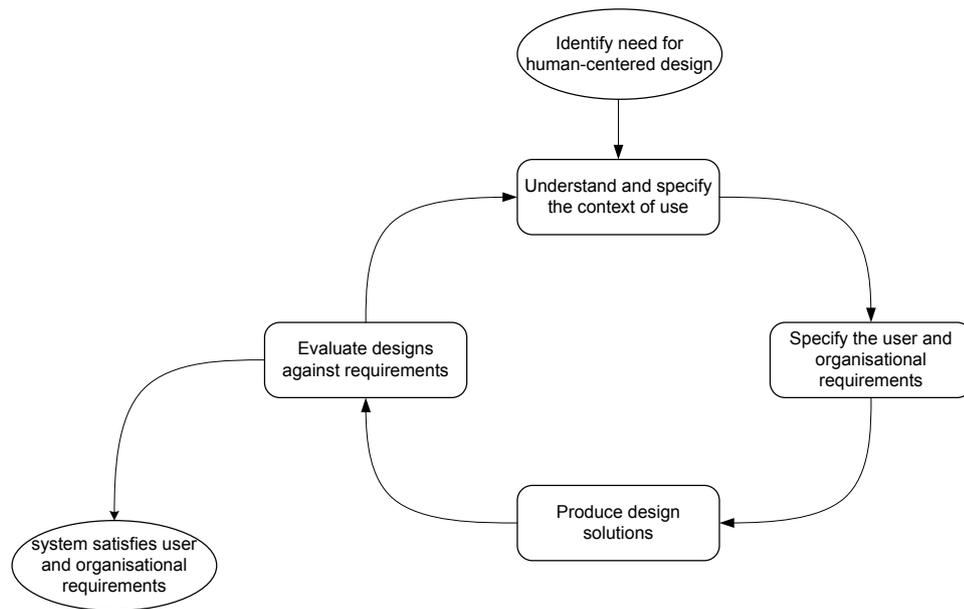


Abbildung 2 – ISO13407 User Centered Design Model

Die vier Phasen (a bis d) des User-Centered Design Model (Abbildung 2) werden iterativ so lange wiederholt bis das Produkt den spezifizierten BenutzerInnen- und Organisationsanforderungen entspricht.

a) Benutzerstudie und –analyse (Context of use)

In der ersten Phase des ISO 13407 Design Modells stehen das Verständnis und die Spezifizierung der Anwendungsparameter, welche durch die Kriterien (Charakteristiken) der zukünftigen BenutzerInnen definiert sind, im Mittelpunkt. Dazu zählen die Arbeitsaufgaben, das Umfeld und die Organisationsanforderungen.

Die Charakteristiken (Kriterien) der zukünftigen BenutzerInnen:

Wichtige Informationen über die BenutzerInnen umfassen deren Wissen, Fähigkeiten, Erfahrung, Ausbildung und Training aber auch physische Attribute, Gewohnheiten, Präferenzen und persönliche Möglichkeiten.

Die Arbeitsaufgaben für BenutzerInnen:

Die Aufgaben für die BenutzerInnen sollen nicht als Funktionen oder Eigenschaften des Systems beschrieben werden. Die Definition der Aufgaben sollte jedenfalls folgende Merkmale enthalten: Zielsetzung, Usability beeinflussende Faktoren, Sicherheits- und Gesundheitsimplikationen, Aktivitäten zwischen BenutzerInnen und Ressourcen.

Umfeld des Systems:

Unter dem Umfeld des Systems versteht man die Hardware und die Software, aber auch physische und soziale Aspekte.

b) Tätigkeitsanalyse (Specification of the User and Organisational Requirements)

Fast alle Design Prozesse umfassen wichtige Aktivitäten um funktionale und nicht funktionale Anforderungen festzuschreiben. Diese Aktivitäten und Anforderungen sollten für das User-Centered Design erweitert werden, um BenutzerInnen und Organisationsanforderungen besser und genauer zu definieren. Die ISO 13407 Norm empfiehlt, dabei folgende Aspekte zu beachten:

- Benötigte Leistung des neuen Systems im betrieblichen und wirtschaftlichen Sinn
- Gesetzliche Normen (Juristische Anforderungen), sowie darüber hinausgehende Sicherheits- und Gesundheitsaspekte
- Kooperation und Kommunikation zwischen den BenutzerInnen und anderen involvierten Personengruppen

- Die Verteilung der Benutzeraufgaben, Zufriedenheit der BenutzerInnen und deren Motivation
- Aufgabenperformance
- Arbeitsdesign und –organisation
- Management von Veränderungsprozessen
- Benutzbarkeit und Wartung
- User-Computer Interface und der Arbeitsplatz

Aus diesen Anforderungen werden die Kriterien für die Usability abgeleitet und die entsprechenden Spezifikationen festgelegt, sodass anschließend Tests für Effizienz, Effektivität und positive Akzeptanz durchgeführt werden können.

c) Prototyp (Production of Design Solutions)

Design Solutions sollten immer auf bereits bestehenden Standards aufgebaut werden, indem vorliegende wissenschaftliche Fakten und Theorien als Vorgaben für die Design Lösungen genutzt werden. Die Ergebnisse der kognitiven Wissenschaften aber auch der Ergonomie und Psychologie sollten dabei miteinbezogen werden. Im Konkreten können hier bereits bestehende User Interface Richtlinien, aktuelle Standards, Produktwissen von vergangenen Implementierungen und Marketinginformationen als durchaus nützliche Informationsquellen dienen.

Außerdem stellen Simulationen, Modelle und andere Prototypen effektive Methoden der Kommunikation zwischen den UserInnen und dem Entwicklungsteam dar. Die Darstellung von Prototypen erleichtert die Design Entscheidungen und erlaubt die Evaluierung mehrerer Konzepte vor einer endgültigen Entscheidung. Im Entwicklungsprozess sollte das Feedback der BenutzerInnen möglichst früh genutzt werden um die Qualität sowie die Vollständigkeit der Funktionsspezifikationen des Designs zu optimieren. Wenn in manchen Fällen die Einbeziehung von BenutzerInnen in frühen Stadien des Designprozesses schwierig ist, können auch ExpertInnen eine Evaluierung vornehmen. Diese Formen der Evaluierung können das wirkliche User Testing jedoch nicht vollständig ersetzen. Die Ergebnisse der Inspektion und die Benutzerkommentare über die Prototypen stellen einen wichtigen Input für zukünftige Designadaptierungen dar, welche die Usability entscheidend verbessern können.

d) Evaluierung und Anforderungen

Die Evaluierungsphase ist im Aufbau eines User-Centered Design unabdingbar, da Evaluierungen Informationen für die Verbesserung des Designs und wichtiges Feedback über die Erfüllung von User- und Organisationsanforderungen liefern. Ein wesentlicher Aspekt der Qualität von Evaluierungsergebnissen besteht in dem Kontext, in dem das System getestet wurde. Die benutzte Evaluierungsmethode hängt vom Systemumfeld, den Kosten und Zeitfaktoren und dem jeweiligen Entwicklungsstand ab. ExpertInnenevaluierungen sind meist schneller und kostengünstiger als von BenutzerInnen durchgeführten, und zeigen mit hoher Wahrscheinlichkeit gravierende Usabilityprobleme auf. Neben diesen Vorteilen können sie aber auf die Arbeit mit BenutzerInnen basierende Evaluierungen nicht ersetzen, sondern sind viel mehr als Ergänzung zu sehen.

Der größte Vorteil eines iterativen Designs besteht in der Möglichkeit, Änderungen bereits in den anfänglichen Implementationsstadien, und damit vergleichsweise kostengünstig, vorzunehmen. Daher ist es sehr wichtig Evaluierungen so früh wie möglich durchzuführen und während der Entwicklung iterativ einzusetzen.

Auch nach der Veröffentlichung eines Produktes sollten Evaluierungen der Usability durchgeführt und damit weiterhin so genannte Feldstudien betrieben werden. Dies begründet sich dadurch, dass manche Probleme und Anforderungen erst nach längeren Nutzungszeiten entstehen können, beispielsweise aufgrund von Veränderungen am Arbeitsplatz, der Arbeitsweise oder der Performance-Anforderungen. Für die beteiligten Usability-ExpertInnen können Daten von Helpdesks, aus Userfeedback, Feldstudien und Leistungsdaten daher auch nach der Freigabe eines Produkts zur Implementierung von Designverbesserungen dienen.

2.4. Methoden des User-Centered Design Prozesses

Für jede Phase der Produktentwicklung existieren mehrere Methoden zur Verbesserung der Usability (vgl. Abbildung 3). Manche dieser Methoden müssen mit wirklichen UserInnen durchgeführt werden, während andere ausschließlich für Expertenevaluierungen vorgesehen sind. Um die Usability eines Produktes zu gewährleisten und zu verbessern, sollten sich dessen EntwicklerInnen vorab für die im jeweiligen Fall adäquaten und umsetzbaren Inspektionsmethoden, unter anderem auch im Hinblick auf die damit verbundenen Kosten und das Heranziehen wirklicher BenutzerInnen, entscheiden.

Überblick der Usability Methoden

Requirements	Design	Implementation	Test	Post Release
User Survey	Design Guidelines	Style Guides	Diagnostic Evaluation	Post Release Testing
Interviews	Paper Prototyping	Rapid Prototyping	Performance Testing	Subjective Assessment
Contextual Inquiry	Heuristic Evaluation		Subjective Evaluation	User Surveys
User Observation	Parallel Design		Heuristic Evaluation	Remote Evaluation
Context	Storyboarding		Critical Incidence Technique	
Focus Groups	Evaluate Prototype		Experiments	
Brainstorming	Wizard of Oz		Cognitive Walkthrough	
Evaluating existing Systems	Interface Design Patterns		Heuristic Walkthrough	
Card Sorting	Cognitive Walkthrough		Guideline Review	
Affinity Diagramming	Heuristic Walkthrough		Consistency Inspection	
Scenarios of Use	Competitive Analysis		Formal Usability Inspection	
Task Analysis			Model Human Processor	
Requirements meeting				

Abbildung 3 – Überblick der Usability Methoden

In Abbildung 3 werden die verschiedenen Usability Methoden nach Entwicklungsphasen gegliedert aufgelistet. Die Tabelle wurde erstellt, um durch die Kombination der angeführten Methoden eine möglichst effiziente Auswahl zur Planerstellung zu ermöglichen. Durch das iterative Design werden die jeweiligen Entwicklungsphasen mehrere Male durchlaufen und die Methoden können anhand des Überblicks in der Tabelle problemlos variiert und adaptiert werden. Vor allem kompetitive Methoden lassen sich so klar darstellen. Außerdem wird eindeutig ersichtlich, dass in der Implementationsphase nur zwei Ansätze gewählt werden können, während in den meisten anderen Phasen der Entwicklung zahlreiche Methoden zur Auswahl stehen. Aufgrund der vielfältigen Herausforderungen an der Schnittstelle zwischen Usability Engineering und Software Engineering erscheint es wesentlich, die Synergien zwischen diesen beiden Disziplinen aufzuzeigen und zu fördern. Dadurch kann sowohl zur Zufriedenheit der BenutzerInnen als auch der DesignerInnen und EntwicklerInnen entscheidend beigetragen werden.

3. Usability Inspection Methods

Auf den folgenden Seiten sollen die Bedeutung von Usability Insepection in der Entwicklung von Software erläutert, und ein Überblick über verschiedene hierfür verwendete Methoden geboten werden. Besonderes Augenmerk liegt in diesem Zusammenhang auf der Heuristic Evaluation, da diese Methode die Basis für das im Rahmen dieser Arbeit entwickelte Tool darstellt.

3.1. Inspektionsmethoden im Software Entwicklungsprozess

Inspektionen werden verwendet, um Fehler in frühen Stadien der Softwareentwicklung zu finden und ein Softwareprodukt auf die Einhaltung von Standards und Vorgabedokumenten (z.B. Anforderungen) zu überprüfen. Aufgrund des hohen Formalismus und des relativ geringen Grades an Freiheiten bei der Durchführung von Inspektionen sind diese auch für Schulungszwecke innerhalb eines Projektteams gut einsetzbar. Denn hier werden sowohl der primäre Nutzen von Inspektionen, welcher im Auffinden von Fehlern besteht, als auch das Kennen lernen des Produkts und der Methode unterstützt. Inspektionen werden nach genau vordefinierten Phasen durchgeführt und sowohl die zu ermittelnden Messdaten als auch die jeweiligen Rollen werden genau vordefiniert.

Die Fagan Inspektion (Fagan, 1976) ist die ursprüngliche Form von Inspektionen und bildet die Basis für die meisten anderen Inspektionstechniken. Es gibt allerdings auch zahlreiche andere Softwareinspektionsmethoden, wie den Active Design Review (Parnas, 1985), die Two-Person Inspektion (Bisant, 1989), die N-Fold Inspektion (Martin, 1990) oder die Phased Inspektion (Knight, 1993), welche die traditionelle Fagan Inspektion zum Teil massiv weiterentwickelt haben.

Der 1994 erstmals von Jakob Nielsen eingeführte Begriff ‚Usability Inspection‘ leitet sich von diesen Softwareinspektionsmethoden ab und wurde für eine spezielle Sammlung von Methoden verwendet, welche drei spezielle Charakteristika aufwiesen:

- EvaluatorInnen werden nicht aus der User Community ausgewählt und sind UsabilityspezialistInnen.
- Die Evaluierung ist zeitlich kürzer als ein Usability Test.
- Die Evaluierung ist kostengünstiger als andere Tests.

Inspektionen sind gewöhnlich kostengünstiger und schneller, da sie es ermöglichen, Änderungen an einem Produkt bereits in den frühen Phasen seiner Entwicklung

durchzuführen. Insofern empfiehlt es sich, bei jeder Produktentwicklung auf Inspektionsmethoden zurückzugreifen.

3.2. Inspektionsmethoden im Usability Engineering

Typische Usability Tests mit AnwenderInnen sind häufig sehr zeitaufwändig und kostenintensiv. Daher können solche Tests nicht in jedem Softwareentwicklungsprojekt eingesetzt werden. Wenn der zeitliche oder finanzielle Aufwand eines klassischen Usability Tests für ein Software-Entwicklungsprojekt nicht tragbar ist, stellen Inspektionsmethoden eine wertvolle und kostengünstige Alternative zur Sicherstellung einer hohen Gebrauchstauglichkeit dar. Diese sollten aber weniger als Ersatz für klassische Usability Tests, sondern vielmehr als deren Ergänzung bzw. als Alternative in Situationen in denen der Einsatz anderer Testmethoden nicht möglich wäre, gesehen werden.

Usability Inspektion steht für eine Zusammenstellung von Methoden basierend auf so genannten EvaluatorInnen welche ein User Interface überprüfen oder verwandte Aspekte der Usability analysieren. Usability InspektorInnen können UsabilityspezialistInnen sein, aber auch Software-BeraterInnen mit spezieller Erfahrung oder NutzerInnen mit besonderem Content- oder Task-Wissen. Unter einer Usability Inspektion versteht man jedenfalls immer die Evaluierung eines User Interface aufgrund einer Beurteilung von InspektorInnen. Die verschiedenen Inspektionsmethoden variieren abhängig davon, wie diese Beurteilung ausgeführt wird und welche Kriterien die InspektorInnen für ihre Beurteilung heranziehen. Im Allgemeinen ist eine Inspektion immer von der Beurteilung der aus dem Feedback abgeleiteten Daten abhängig. Dabei werden vier Möglichkeiten bei der Evaluation unterschieden:

- automatisch - Analyse des User Interface durch eine Evaluierungssoftware
- empirisch - Analyse des User Interface durch wirkliche UserInnen
- formell - Einsatz exakter Modelle und Formeln um das User Interface zu beurteilen
- informell – basierend auf Wissen und Erfahrung des/r EvaluatorIn („Daumenregeln“)

Mit dem heutigen Stand der Technik können automatische Methoden noch nicht umgesetzt werden. Formale Methoden sind aber deshalb oft schwierig einzusetzen, da

sie nicht auf einem großen komplexen interaktiven User Interface anwendbar sind. Die am häufigsten benutzten Methoden umfassen die empirische Evaluierung von User Interfaces durch UserInnen selbst. Diese sind aber oftmals schwierig zu rekrutieren oder sehr kostspielig, wenn alle Aspekte des entstehenden Designs getestet werden sollen.

Inspektionsmethoden gehören zu den informellen Vorgehensweisen und können das User-Testing teilweise ersetzen. Da manche Probleme der Usability nur durch User-Testing und manche nur durch Inspektion gefunden werden, ist eine Kombinationsvariante die erfolgreichste Methode. Daher sollten in Software-Entwicklungsprojekten mit ausreichenden zeitlichen und finanziellen Ressourcen die Vorteile dieser beiden Methoden kombiniert werden, um optimale Ergebnisse zu erzielen. Inspektionen können hierbei vor allem im Vorfeld von Usability Tests durchgeführt werden, um zu verhindern, dass im Test Situationen auftreten, welche die Ergebnisse unbrauchbar machen. Ein Beispiel dafür sind Fehler die eine vollständige Durchführung des Tests verhindern könnten.

Das gemeinsame Ziel aller Inspektionsmethoden besteht darin, die Bedienoberfläche einer Software, einer Webseite bzw. eines Produktes durch ein oder mehrere GutachterInnen nach verschiedenen Kriterien bewerten zu lassen. Auf diese Weise können Usabilityprobleme frühzeitig erkannt und beseitigt werden. Die GutachterInnen können AnwenderInnen, Fachleute eines Anwendungsgebietes, SoftwareentwicklerInnen, GestalterInnen oder UsabilityexpertInnen sein.

Inspektionsmethoden, die auch Reviewmethoden genannt werden, sind dazu geeignet Bedienprobleme aufzuzeigen. Im Abschlussbericht einer Usabilityinspektion finden sich daher üblicherweise die Beschreibungen von gefundenen Usabilityproblemen mit den dazugehörigen Lösungsvorschlägen. Allerdings eignen sich Usability-Inspektionsmethoden nicht dazu, komplexere Bedienprobleme zu lösen. Man kann solche Probleme zwar identifizieren, für deren Behebung müssen aber adäquatere Usabilitymethoden, wie zum Beispiel Paper Prototyping, eingesetzt werden.

Usability Inspektionsmethoden können projektbegleitend in fast jeder Phase eines Software-Entwicklungsprojektes eingesetzt werden. Für ihre Durchführung werden in der Regel keine marktreifen Versionen der Software bzw. funktionale Prototypen benötigt. Meist sind schon Spezifikationen, Skizzen, Mock-Ups oder Produktbeschreibungen ausreichend, um erste verwertbare Ergebnisse zu erzielen.

Mit Hilfe von Inspektionen können komplexere Usabilityprobleme entdeckt werden, welche in klassischen Usability Tests nur mit Hilfe langer Testreihen bzw. einer großen Anzahl an ProbandInnen entdeckt worden wären.

Ein weiterer wichtiger Vorteil besteht darin, dass Usability Inspektionen auch in Software-Entwicklungsprojekten mit knappem Budget eingesetzt werden können. Sie werden deshalb auch unter dem Begriff „Discount Usability“ mit eingeordnet.

Die Nachteile von Inspektionsmethoden liegen vor allem in der subjektiven Einschätzung der InspektorInnen und den Grundlagen auf denen diese getroffen werden. GutachterInnen arbeiten zum Teil auf der Grundlage von Anwenderprofilen und Anwendungsszenarien, welche nicht aus empirischen Untersuchungen bzw. Daten abgeleitet wurden. Das bedeutet, dass die InspektorInnen mit Annahmen zu Anwenderprofilen und Anwendungsszenarien arbeiten. Diese Annahmen können teilweise verfälscht, unvollständig oder komplett falsch sein.

Die Ergebnisse einer Usability Inspektion basieren auf den Erfahrungen des/r GutachterIn, und stellen somit eine subjektive Einschätzung der Bedienoberfläche nach bestimmten Kriterien dar. Dadurch besteht die Gefahr, dass die Ergebnisse durch fehlende Kenntnisse und Erfahrungen der GutachterInnen im Anwendungsgebiet der Software oder das Übersehen von Usabilityproblemen aus technischen Gründen, wie zum Beispiel bei Problemen mit bestimmten Bildschirmauflösungen, verfälscht werden. Außerdem kann ein Mangel an Erfahrung mit der Durchführung von Inspektionen zur Folge haben, dass Anfängerfehler, die für ExpertInnen oftmals schwierig zu erkennen sind, übersehen werden. Zudem besteht immer auch das Risiko, dass sich andere subjektive Einflussfaktoren, wie beispielsweise die jeweilige Tagesverfassung der GutachterInnen, auf die Resultate niederschlagen.

3.3. Beschreibung von Inspektionsmethoden

3.3.1. Heuristic Evaluation

Die heuristische Evaluierung ist eine Usability Engineering Methode zur Untersuchung von Usability Problemen in einem User Interface Design welche im iterativen Design Prozess zu beachten sind, und wurde als Usability Methode erstmals von Nielsen und Molich (1990) veröffentlicht. Bei einer heuristischen Evaluierung testen mehrere EvaluatorInnen ein Interface und beurteilen dessen Kompatibilität mit den vorher festgelegten Usability Prinzipien bzw. Heuristiken (Nielsen, 2005).

3.3.1.1. Durchführung

Die Durchführung einer heuristischen Evaluierung besteht aus mehreren Abschnitten, welche folgendermaßen aufgegliedert werden können:

a) Training

Falls die EvaluatorInnen noch keine Erfahrungen mit heuristischer Evaluierung haben, wird zunächst eine Einführung in diese Methode erforderlich sein. Diese Einführung und deren Umfang müssen der jeweiligen Erfahrung der InspektorInnen angepasst werden. Je nach Art der Evaluierung ist ein Training der EvaluatorInnen sowohl über den Einsatzbereich als auch den Vorgang erforderlich, um ihnen das nötige Wissen zu vermitteln. Ein derartiges Training soll die EvaluatorInnen nicht zu ExpertInnen ausbilden, sondern die Idee des Einsatzbereichs des Systems vermitteln. Eine andere Möglichkeit das fehlende Einsatzwissen zu kompensieren, wäre eine Betreuung der InspektorInnen durch ExpertInnen während der Evaluierung. Da die EvaluatorInnen das System jedenfalls aus einer unbeeinflussten und freien Perspektive testen sollen, darf ein vorangehendes Training keine Details über das User Interface des zu prüfenden Systems enthalten. Falls durch die Evaluierung jedoch nur ein kleiner Teil des Systems geprüft werden soll, kann den EvaluatorInnen ein Task Szenario als Basis für die Evaluierung vorgegeben werden.

Typischerweise sollten derartige heuristische Untersuchungen nicht länger als zwei Stunden dauern. Falls sie dennoch mehr Zeit beanspruchen (etwa bei sehr komplexen Nutzerschnittstellen), ist es ratsam sie in mehrere Teile aufzusplitten, die sich dann jeweils auf nur einen Bereich des Interface konzentrieren. Während der Untersuchung sollte der/die EvaluatorIn das Interface mehrere Male

durchgehen und alle Aspekte des Interfaces aufgrund der vorgegebenen Heuristiken (siehe 7.1 für ein Beispiel) auf Unstimmigkeiten kontrollieren. Prinzipiell müssen die EvaluatorInnen selbst entscheiden, wie gründlich sie das Interface testen möchten. Zwei Durchgänge sollten aber mindestens durchlaufen werden, da viele Fehler erst beim zweiten Durchlauf festgestellt werden, während der erste Durchgang primär dem kennen lernen des Systems dient.

b) Evaluation

Die EvaluatorInnen müssen das Interface bei der heuristischen Evaluierung unabhängig voneinander inspizieren, und sollten noch keine Zwischenergebnisse diskutieren bevor alle die Evaluierung beendet haben. Eine Evaluierung dauert möglicherweise ein bis zwei Stunden. Falls das Interface eine große Anzahl an Dialogen enthält oder sehr komplex ist, kann die Evaluierung auch in mehrere Sitzungseinheiten aufgeteilt werden. Die InspektorInnen sollten üblicherweise zwei Evaluierungsdurchläufe durchführen. Der erste Durchlauf soll den Evaluierenden einen Überblick über die Bedienung, den allgemeinen Umfang und die Interaktion mit dem Interface verschaffen. Falls ein Task Szenario vorgeben ist, wird dieses bereits im ersten Durchlauf eingebunden. Der zweite Durchgang der Evaluierung ist zwecks einer detaillierten Analyse der Dialoge und der Elemente anhand der Liste der Usability Heuristiken durchzuführen. Diese zwei Durchgänge sind Vorschläge, aber keineswegs in jedem Fall verbindlich. EvaluatorInnen können selbst entscheiden, wie sie die Evaluierung gestalten. Um sich einen Überblick zu verschaffen, ist aber ein erster Durchgang durchaus ratsam.

Die EvaluatorInnen sollten alle von ihnen erkannten Usability Probleme – unabhängig von deren Schweregrad - festhalten. Die jeweilige Problembeschreibung sollte präzise ausgeführt sein. Das bedeutet, dass Screenshots wie auch eine genaue Anleitung zur Reproduktion des Problems hinzugefügt werden sollten. Jedes der identifizierten Probleme ist üblicherweise einer oder mehreren Usability Heuristiken zugeordnet, wobei die Sammlungen der Heuristiken durchaus den Umständen angepasst werden können. Die Ergebnisse können schriftlich aufgezeichnet werden oder auch einer zentralen Beobachtungsstelle, wo die Eindrücke der EvaluatorInnen beim Testen des Interface gesammelt bzw. eigene Eindrücke des/r BeobachterIn festgehalten werden, mitgeteilt werden. Die schriftliche Aufzeichnung der Probleme bietet den Vorteil einer formalen Präsentation (für das Management), stellt aber auch einen

höheren Aufwand für die EvaluatorInnen dar. Außerdem müssen die Schriftstücke gelesen und dann von EvaluierungsmanagerInnen zusammengefügt werden.

Beim Einsetzen eines/r EvaluierungsmanagerIn kommt es einerseits zu einem zusätzlichen Aufwand bei jeder Evaluierungssession, andererseits verringert sich aber die Arbeit der EvaluatorInnen, was den zusätzlichen Vorteil hat, dass die Ergebnisse schnell nach der Evaluierungssession bekannt gegeben und bei der Evaluierung aufgetretene Probleme sofort geklärt werden können.

c) Merging & Severity Rating

Die durch die EvaluatorInnen gefundenen Probleme werden in einer Liste zusammengefasst, um danach das endgültige Severity Rating durchzuführen. Diese Liste sollte auch Aufschluss darüber geben, welche EvaluatorInnen jedes dieser Probleme gefunden haben, da mit Hilfe dieser Information die Korrelation zwischen der Anzahl der EvaluatorInnen und der Anzahl der gefunden Probleme berechnet werden kann. Diese Erkenntnisse können bei späteren heuristischen Evaluierungen zur Optimierung der Anzahl der EvaluatorInnen dienen.

Den Resultaten der EvaluatorInnen soll in dieser Phase ein Severity Ranking zugeordnet werden. Dazu wird zunächst jedem Problem ein Severity Rating zugeteilt, welches unter anderem auf der Frequenz, also der Häufigkeit mit der ein Problem auftritt, basiert. Weitere Kriterien zur Bewertung der Schwere eines Problems bestehen in seiner Auswirkung, also dem Schwierigkeitsgrad seiner Bewältigung, und der Hartnäckigkeit (Persistenz) des Problems, die vor allem dann gegeben ist, wenn der/die UserIn ein Problem nicht nur bei der ersten, sondern auch bei späteren Anwendungen eines User Interface als störend empfindet.

d) Writing a Report

Das so genannte Industry Usability Reporting Project (National Institute of Standards and Technology, 1997) hat 1999 ein Common Industry Format (CIF) für Usability Test Reports herausgegeben. Andrews änderte dieses ab und fasste die für die heuristische Evaluierung relevanten Aspekte zusammen:

- Titelseite - Name und Version des Produkts, Name des/r TestleiterIn, Name des/r ReportschreiberIn, Datum

- Executive Summary - Identität und Beschreibung des Produkts, Zusammenfassung der verwendeten Methode, Zusammenfassung der Resultate
- Evaluierungsumfeld - Daten der BenutzerInnen, sowie die verwendete Hardware und Software jedes/r Evaluators
- Positive Eindrücke - einige positive Aspekte des Prototypen mit Screenshots
- Analyse der Hauptprobleme - die schwerwiegendsten Probleme (höchste Severity)
- Liste aller gefundenen Probleme
- Appendix - Logs der EvaluatorsInnen, Rohdaten usw., falls erwünscht

3.3.1.2. Diskussion und Bewertung

Bei der heuristischen Evaluierung ist es für eine einzelne Person fast unmöglich alle Usability Probleme in einem Interface zu finden. Diversen Studien (Nielsen, 1992) zufolge findet ein/e EvaluatorIn nur durchschnittlich 35 Prozent der bekannten Usability Probleme in einem Interface. Aus diesem Grund ist es effektiver, bei dieser Methode mehrere EvaluatorsInnen zu involvieren, um möglichst viele Probleme zu lokalisieren. Ein geeignetes Minimum sind drei EvaluatorsInnen, weil bis zu dieser Zahl die Anzahl der gefundenen Probleme stark ansteigt. Bei kritischen Projekten ist jedoch eine größere Anzahl an EvaluatorsInnen heranzuziehen, um optimale Resultate zu erzielen. Um die ideale Anzahl an EvaluatorsInnen für eine heuristische Evaluierung herauszufinden, kann auch eine Cost/Benefit Analyse durchgeführt werden.

Nielsen und Landauer (1993) präsentierten ein solches Modell, basierend auf folgender Prognoseformel für die Anzahl der in der heuristischen Evaluierung gefundenen Usability Probleme: $Probleme\ gefunden(i) = N(1 - (1 - \lambda)^i)$

Wobei *Probleme gefunden(i)* für die aggregierte Anzahl der verschiedenen gefundenen Usability Probleme der *i* verschiedenen EvaluatorsInnen steht, *N* für die totale Anzahl der gefundenen Usability Probleme im Interface und λ für die proportionale Anzahl der gefundenen Usability Probleme jedes/r EvaluatorsIn, wobei diese aber von Projekt zu Projekt variieren wird.

Nielsen und Mack (1994) konnten nach ihren Studien folgendes Cost/Benefit Modell berechnen (Abbildung 4).

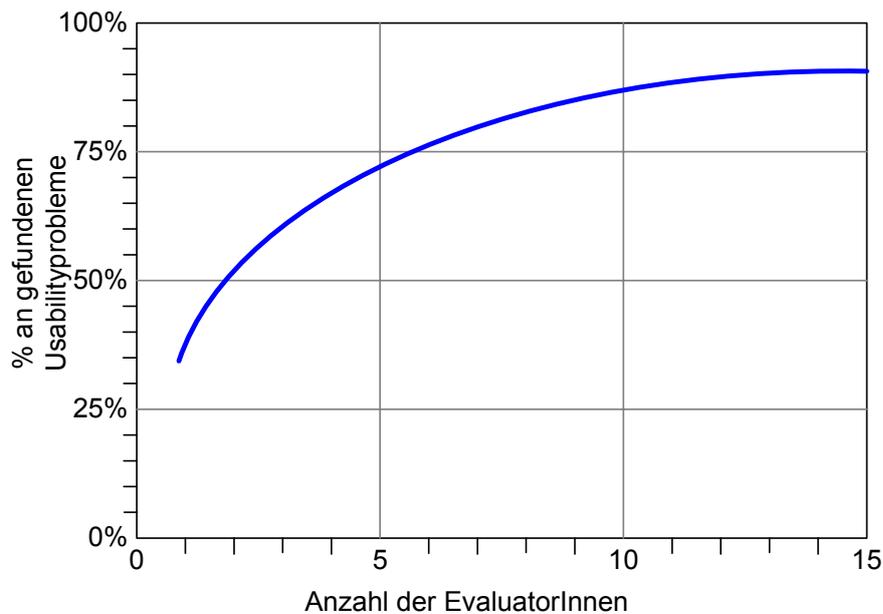


Abbildung 4 – Cost/Benefit Model

Die heuristische Evaluierung kann sowohl schwerwiegende als auch vernachlässigbare Usability Probleme aufdecken. Die Wahrscheinlichkeit, ein schwerwiegendes Usabilityproblem zu finden, ist im Allgemeinen höher als jene, ein vernachlässigbares zu entdecken. Nielsen errechnete aufgrund der sechs von ihm durchgeführten Studien, dass die Wahrscheinlichkeit, dass EvaluatorInnen ein schwerwiegendes Usabilityproblem erkennen, bei 42 Prozent liegt, während vernachlässigbare Probleme nur in 32 Prozent der Fälle erkannt werden. Obwohl schwerwiegende Usability Probleme einfacher zu finden sind, werden nach Nielsen viel mehr vernachlässigbare Probleme als schwerwiegende gefunden, da sich die schwerwiegenden Probleme häufig überlappen. Durch das Severity Rating können große, markante Probleme leicht markiert werden, aber auch kleine, die genauso relevant sind, gekennzeichnet werden. Ein großer Vorteil der heuristischen Evaluierung besteht darin, dass die EvaluatorInnen das System nicht wirklich benutzen müssen, um erste Probleme aufzeigen zu können. Daher würde sogar ein User Interface als Paper Prototyp ausreichen und es muss kein fertiges System vorhanden sein bevor die Evaluierung beginnen kann. Dies erlaubt eine frühe Einbeziehung der heuristischen Evaluierung in den Usability Engineering Lifecycle.

3.3.2. Cognitive Walkthrough

Der Cognitive Walkthrough (Polson, 1992) besteht aus zwei Teilen und beginnt mit der Vorbereitungsphase, welche sich in vier Abschnitte gliedert:

a) Userdefinition

In diesem Teil wird die Frage „Wer wird dieses System benutzen?“ geklärt, was teilweise einfach sein kann, wenn es sich um einen Alltagsgegenstand, wie um einen Bankomat, handelt. Jedoch sollte man spezielles Hintergrundwissen der UserInnen bzw. deren Ausbildung nicht ignorieren, da diese Faktoren deren jeweilige Herangehensweise maßgeblich beeinflussen können.

b) Definition der Input Konditionen für den Walkthrough

Da es oftmals mit großem Aufwand verbunden wäre, alle Optionen eines User Interface zu testen, wählt man eine sinnvolle Anzahl an repräsentativen Tasks aus, um für diese ein Benchmarking einzusetzen. Die Auswahl gerade dieser repräsentativen Aufgaben ist schwierig und sollte basierend auf den Marketingstudien, der Anforderungsanalyse und den Konzepttests erarbeitet werden. Manche dieser repräsentativen Tasks sollten notwendigerweise die Kernfunktionalität der Applikation testen. Zudem sollten auch unterschiedliche Kombinationen dieser Kernfunktionalitäten getestet werden, um die Möglichkeit eines Usabilityproblems im Taskflow aufzudecken. Diese Tasks sollten so realistisch wie möglich ausführbar sein und im Arbeitskontext stattfinden, um einen gewöhnlichen Einsatz zu simulieren. Dazu gehören auch die zukünftige Datenstruktur für das Interface, sowie plausible Testdaten (und Testdatengröße) in der Datenbank, welche die BenutzerInnen auch später so gebrauchen werden.

c) Definition der Tasks und der Aktionssequenzen für jeden Task

Ein weiterer wichtiger Aspekt der Vorbereitungsphase eines Cognitive Walkthrough besteht in der Definition, wie BenutzerInnen ihre Aufgabe betrachten sollten, bevor sie das jeweilige Interface kennen lernen. Des Weiteren erfolgt eine Beschreibung der Sequenzen und Aktionen die der/die UserIn durchläuft um den Task erfolgreich zu beenden. Diese Aktionen können einfache Bewegungen, wie Drücken der „y“ Taste oder den Cursor auf das Menu „Datei“ führen, sein. Es kann sich aber auch um Aktionssequenzen handeln, wie den Login in ein System. Um ein möglichst effektives Tutorial für einen Task zu erstellen, sollte die Gründlichkeit der Definitionen der Usererfahrung angepasst werden.

d) Interface

Die Beschreibung des Interface muss alle Reaktionen des Interface auf die Aktionen in dem Task enthalten. Wenn die Applikation bereits implementiert ist, sollten alle nötigen Informationen schon in der Softwareentwicklungsphase dokumentiert werden. Die Beschreibung sollte der Usererfahrung angepasst werden, damit kein zusätzlicher Overhead entsteht.

Nach Abschluss der Vorbereitungsphase erfolgt die Analysephase. Hier werden zunächst alle Aktionen und deren Lösungswege überprüft. Dazu versucht man, sich in die Rolle von UserInnen zu versetzen und den Lösungsweg aus deren Sicht nachzuvollziehen, um dadurch zu erkennen, wie von den UserInnen die richtige Lösung gewählt wurde. Von den EvaluatorInnen sollte dabei von folgenden Fragestellungen ausgegangen werden:

„Wird der/die UserIn den richtigen Effekt erzielen?“,

„Wird der/die UserIn die richtige Aktion finden?“,

„Wird der/die UserIn die richtige Aktion mit dem richtigen Effekt assoziieren?“

und „Wird der/die UserIn nach erfolgreichem Abschluss der Aktion einen Fortschritt feststellen können?“

Dies sind nur exemplarische Fragen, die durch detaillierte Richtlinien (Guidelines) ersetzt werden können.

Während der Evaluierung ist es wichtig einige Informationen, beispielsweise mit Hilfe multimedialer Aufnahmegeräte, festzuhalten. Probleme, die die Usability eines User Interface einschränken können umfassen unter anderem Timeouts und physisch schwierig durchführbare Aktionen, wie beispielsweise Tastenkombinationen.

Das Endergebnis eines Cognitive Walkthrough besteht aus den Success und/oder Failure Stories für jeden Task, welche sich aus den Taskbeschreibungen und entweder einer erfolgreichen „Story“ oder der Verletzung eines der Kriterien und einem Beispiel warum der/die UserIn die Aktion nicht erfolgreich beenden konnte, zusammensetzen. Bei einer Gruppenevaluierung sind vorzugsweise für alle sichtbare Materialien für die Dokumentation zu verwenden und drei Listen zu führen, wobei die erste für alle UserInnen relevante Informationen beinhaltet, während sich die zweite mit allen Argumenten der Gruppe über die Gruppenstory befasst und auf der dritten Liste die Nebeneffekte bzw. Designprobleme festgehalten werden.

3.3.3. Heuristic Walkthrough

Der heuristische Walkthrough ist eine kombinierte Methode, teils bestehend aus der heuristischen Evaluierung, teils aus dem Cognitive Walkthrough und wurde von Sears 1997 veröffentlicht. Die Methode versucht, die Nachteile der heuristischen Evaluierung und des Cognitive Walkthrough aufzuheben und deren Vorteile zu unterstützen.

Die heuristische Evaluierung ist nicht strukturiert und der einzige Leitfaden besteht in den Heuristiken. Der Cognitive Walkthrough hingegen gibt eine sehr detaillierte Struktur vor, was EvaluatorInnen die das System frei nutzen wollen möglicherweise einschränkt. Ein Heuristic Walkthrough hat zwei Iterationen, von denen die erste Iteration mit dem Cognitive Walkthrough identisch ist, indem eine Liste an priorisierten Aufgabenbeschreibungen und gedankenfokussierenden Fragen abgearbeitet wird. Die EvaluatorInnen können durch die zukünftigen Benutzeraufgaben eine Einsicht in das User Interface erlangen. Die zweite Iteration allerdings ist der heuristischen Evaluierung nachempfunden, bei der die EvaluatorInnen ihr aufgabenorientiertes Wissen frei einsetzen können.

Des Weiteren betont Sears die Wichtigkeit von drei Dokumenten, die während der Evaluierung benötigt werden. Zunächst ist es von Bedeutung eine Taskliste zu erstellen, welche sich aus einer priorisierten Sammlung von häufigen und wichtigen Aufgaben, die normalerweise im User Interface genutzt werden, zusammensetzt. Diese Liste kann auch Tasks enthalten, die die EvaluatorInnen jedem Teil des Systems exponieren. Die EvaluatorInnen können das System dann frei erkunden, aber sollten diese Prioritäten beachten, um angemessene Aufgaben des Interface auszuwählen.

Weiters verwendet man gedankenfokussierende Fragen, um die erste Iteration, welche dem Cognitive Walkthrough nachempfunden ist, zu unterstützen. Vier Fragen sollten den EvaluatorInnen helfen, die Benutzergedanken zu beleuchten während sie ein System bedienen und dessen Lernbarkeit testen:

„Wird der/die BenutzerIn wissen, was er/sie als nächstes tun soll?“

„Wird der/die BenutzerIn die Steuerung erkennen, um den nächsten Teil der Aufgabe zu erledigen?“

„Sobald ein/e BenutzerIn die Steuerung findet, kann er/sie sie verwenden?“

„Wenn ein/e BenutzerIn die richtige Aktion ausführt, wird er/sie einen Fortschritt erkennen?“

Das dritte benötigte Dokument zur Durchführung eines Heuristic Walkthrough besteht in einer Heuristikliste. Da die zweite Iteration des heuristischen Walkthrough im Wesentlichen einer heuristischen Evaluierung gilt, werden für diese letzte Phase der

Inspektionsmethode Usability Heuristiken benötigt. Hierfür kann jede etablierte Heuristikliste verwendet werden, die aber zuvor ausgewählt werden muss.

Iteration 1: Aufgabenorientierte Evaluierung

In der ersten Phase eines Heuristic Walkthrough werden die EvaluatorInnen wie zuvor erwähnt durch eine Taskliste unterstützt. Durch die Priorisierung der Aufgaben im Hinblick auf deren Wichtigkeit und Nutzungshäufigkeit, können die EvaluatorInnen den Hauptnutzungszweck eines User Interface mühelos erkennen. Jede/r EvaluatorIn kann aber auch darüber hinaus gehende Aufgaben ausführen, da der Ablauf nicht genau festgelegt ist. Die Taskliste und die Priorisierung dienen nur als Hilfsmittel für die EvaluatorInnen, um ihnen die Auswahl wichtiger Teile des Interface zu erleichtern. Die EvaluatorInnen sollten das System möglichst unvoreingenommen benutzen, wobei sie auch die oben angesprochenen gedankenfokussierenden Fragen, beachten sollten.

Iteration 2: Freie Evaluierung

Während der zweiten Phase des Heuristic Walkthrough können die EvaluatorInnen jeden Teil des Systems komplett frei erkunden. Sie sollten sich in der ersten Iteration ein ausreichendes Grundwissen über das System angeeignet haben, um nun das Interface anhand der vorgegebenen Heuristiken evaluieren zu können.

3.3.4. Guideline Reviews

Bei einem Guideline Review wird ein User Interface anhand einer Liste von Guidelines auf Übereinstimmungen geprüft. Guidelines gibt es in verschiedenen Abstraktionsgraden und sie sind meist als große Sammlungen zusammengefasst. Eine der ersten populären Sammlungen von Guidelines wurde für die U.S. Airforce von Smith und Moiser 1986 ausgearbeitet, wobei 944 Guidelines in sechs Kategorien gestaffelt waren. Heute sind Sammlungen an Guidelines von der International Standardization Organization zu bekommen, wie unter anderem „ISO 9241-11 Anforderungen an die Gebrauchstauglichkeit“. Das Problem bei großen Sammlungen besteht darin, jede Regel der Reihe nach anzuwenden, da diese Guidelines nicht auf ein bestimmtes Interface abgestimmt sind. Guideline Reviews empfehlen sich daher nur für InspektorInnen die schon große Erfahrung auf dem Gebiet haben und mit Guidelines umzugehen wissen.

3.3.5. Plurastic Walkthroughs

Mit dem Begriff „Pluralistische Inspektion“ (Bias, 1994) ist die systematische Gruppenevaluierung gemeint. Dabei begutachtet eine Gruppe aus repräsentativen AnwenderInnen, Software-EntwicklerInnen, GestalterInnen und UsabilityexpertInnen in einem Workshop gemeinsam die Gebrauchstauglichkeit einer Bedienoberfläche. Der Vorteil dieser Methode liegt in den unterschiedlichen Fähigkeiten, Erfahrungen und Sichtweisen der InspektorInnen. Im Vorfeld der Inspektion werden verschiedene Anwendungsszenarien erarbeitet und möglichst detailliert beschrieben. Die Beschreibungen sollten die einzelnen Bedienungsschritte enthalten, welche für die Bewältigung einer bestimmten Aufgabe nötig sind. Sollte es für eine Aufgabe mehrere Lösungswege geben, wird jene genutzt, welche vermutlich am meisten verwendet wird, bzw. werden im Bedarfsfall die alternativen Lösungswege ebenfalls geprüft. In der Regel werden pluralistische Inspektionen durch eine/n UsabilityexpertIn moderiert. Teilnehmende SoftwareentwicklerInnen und GestalterInnen beantworten Fragen zur Gestaltung und schlagen Lösungen für gefundene Probleme vor. Die eigentliche Inspektion wird aber von den AnwenderInnen durchgeführt. Dazu werden die einzelnen Aufgaben von dem/r ModeratorIn erläutert. Danach werden die TeilnehmerInnen aufgefordert, Vorschläge zu notieren, wie sie diese Aufgabe mit der gezeigten Bedienoberfläche lösen würden bzw. welchen Weg sie wählen würden, um eine Lösung zu finden.

Nachdem der/die ModeratorIn den vorgesehenen Lösungsweg gezeigt hat, werden die von den TeilnehmerInnen entwickelten Lösungswege vorgestellt und diskutiert. Die SoftwareentwicklerInnen sollten sich an der Diskussion der alternativen Lösungsansätze zunächst nicht beteiligen, da AnwenderInnen in der Regel großen Respekt vor der Fachkenntnis der SoftwareentwicklerInnen haben. Durch ein zu frühes Erläutern von bestimmten und vor allem technischen Zusammenhängen, könnten einfallsreiche Ideen von AnwenderInnen aus Respekt vor den komplexen Tatsachen nicht eingebracht werden. Erst gegen Ende der Diskussion sollten die SoftwareentwicklerInnen erklären, warum die Bedienoberfläche für das jeweilige Anwendungsszenario so gestaltet wurde. Um die Diskussion überschaubar und moderierbar zu gestalten, sollten an einer pluralistischen Inspektion etwa 6-10 AnwenderInnen, eine im Vergleich zu den NutzerInnen etwas geringere Anzahl an SoftwareentwicklerInnen, sowie ein bis zwei UsabilityexpertInnen teilnehmen. Pluralistische Inspektionen werden in der Regel mit einem Fragebogen zur Gebrauchstauglichkeit beendet, wobei die TeilnehmerInnen mit Hilfe von Skalen die

subjektive Gebrauchstauglichkeit beurteilen. Die Ergebnisse der Fragebögen einer Inspektion werden dann mit den Ergebnissen anderer pluralistischer Inspektionen der gleichen Bedienoberfläche verglichen. Diese Inspektionsmethode eignet sich besonders für das Sammeln von Anwenderfeedbacks zu Prototypen oder marktreifen Softwareprodukten. Ein weiterer Vorteil besteht darin, dass das Anwenderfeedback neben Hinweisen auf Usabilityprobleme auch zu neuen Entwicklungsideen führen kann.

3.3.6. Consistency Inspection

Diese Methode wird vorwiegend bei großen Projekten vorgenommen (Wixon, 1994). Dabei werden die einzelnen Komponenten auf ihre Konsistenz zum Gesamtprojekt untersucht. Ähnlich dem Pluralistic Walkthrough untersuchen ExpertInnen das Interface Schritt für Schritt, wobei sie besonderes Augenmerk auf die Einheitlichkeit der Produktelemente legen. Diese Analysemethode empfiehlt sich für ein frühes Stadium der Entwicklungsphase, bietet sich aber auch für ein komplettes Re-Design an.

3.3.7. Formal Usability Inspektion

Eine formale Usabilityinspektion (Kahn, 1994) dient zur Überprüfung der potentiellen Taskperformance eines Produkts. Diese Aufgabe übernimmt ein/e ProdukterzeugerIn (z.B.: ProgrammiererIn) und ein Team, welches nach Defekten im Ablauf sucht. Dabei wird ein Inspektionsteam bestehend aus vier bis acht Personen eingesetzt. Jede der Personen ist ein/e InspektorIn. Außerdem werden noch drei zusätzliche Personen benötigt: Ein/e ModeratorIn leitet die Meetings, und verteilt und sammelt alle relevanten Informationen. Weiters sollte der/die BesitzerIn oder UrheberIn des Programms miteinbezogen werden, da diese/r allfällige Defekte später reparieren und die Kommunikation zum Softwaredevelopmentteam übernehmen kann. Eine dritte Person ist der/die RecorderIn oder SchreiberIn, der/die die im Meeting gesammelten Informationen protokolliert. In einem zweiten Schritt gelangen die Unterlagen des Produkts zur Verteilung. Das so genannte Inspection Paket besteht aus dem Cover Letter, der Produktinformation, der Begleitdokumentation, den User Profilen, den Task Szenarios, dem Task Performance Model (Abbildung 5), den Heuristiken, dem Defect Logging Form und der Logistik. Der nächste Schritt ist dann ein Kickoff Meeting, bei dem alle TeilnehmerInnen das erste Mal aufeinander treffen. Der/Die ProduktentwicklerIn und der /die ModeratorIn übergeben den übrigen TeilnehmerInnen des Meetings das Inspection Paket und erläutern dieses.

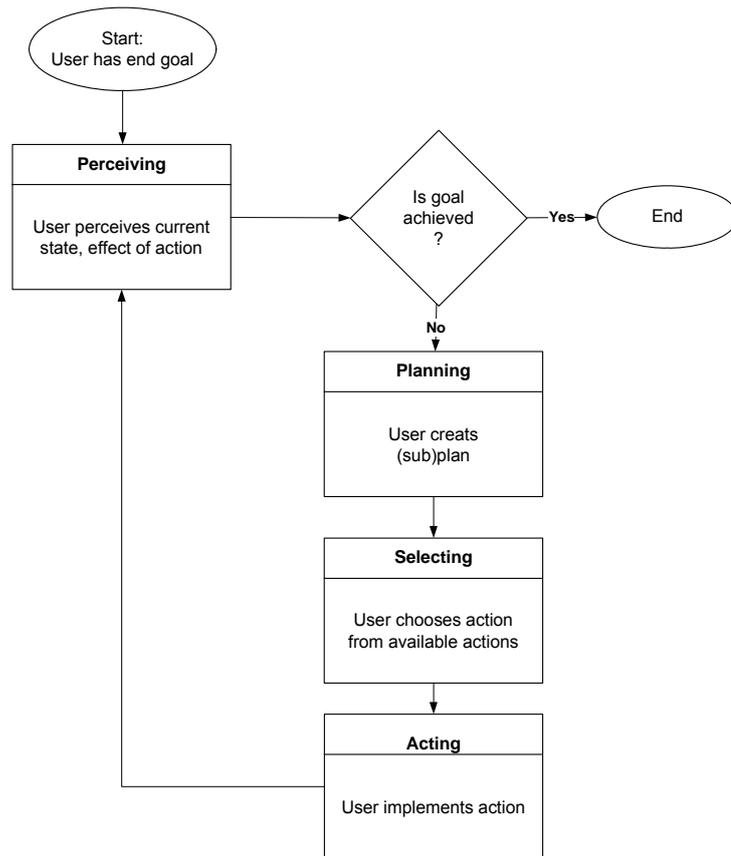


Abbildung 5 – Task Performance Model

Im Anschluss daran werden den InspektorInnen ihre Aufgaben erklärt, welche in der Suche nach Problemen mit Hilfe von User Profiles, Task Szenarios bzw. des Task Performance Modells besteht. Es folgt die Preparation Phase, in der jede/r InspektorIn alleine arbeitet. Anhand der ihnen zugeteilten Benutzerprofile durchlaufen die InspektorInnen nun das Task Szenario. Wenn der Fall eintritt, dass ein/e InspektorIn eine Aufgabenstellung nicht auf Anhieb lösen kann, wird dieser als Usabilityproblem dokumentiert. Der/Die InspektorIn benutzt dabei auch das Task Performance Modell, sowie die Heuristiken um Probleme zu identifizieren. Nachdem sich alle Teammitglieder unabhängig voneinander vorbereitet haben, findet ein Logging Meeting statt, um die gefunden Defekte zu aggregieren und dadurch gegebenenfalls weitere Defekte zu identifizieren. Bei diesem Meeting befragt der/die ModeratorIn die InspektorInnen anhand der User Profile/Task Szenarios zu möglichen Problemen.

Nach dem Abschluss des Logging Meetings beginnt die Rework Phase, in der Defekte identifiziert und an die zuständigen Stellen weitergeleitet werden, um korrigiert zu werden. In dieser Phase können auch Severity Ratings bzw. eine Gruppierung der Defekte erfolgen.

Nach dem Rework findet noch ein Follow-Up statt, bei dem die Ergebnisse der Untersuchung, wie die gefunden Defekte bzw. die in das Projekt investierten Stunden oder auch die Anzahl von verifizierten Defekten und reparierten Defekten, verteilt werden. Nach der Diskussion der Prozessqualität und -effizienz endet die formale Inspektion (Abbildung 6).

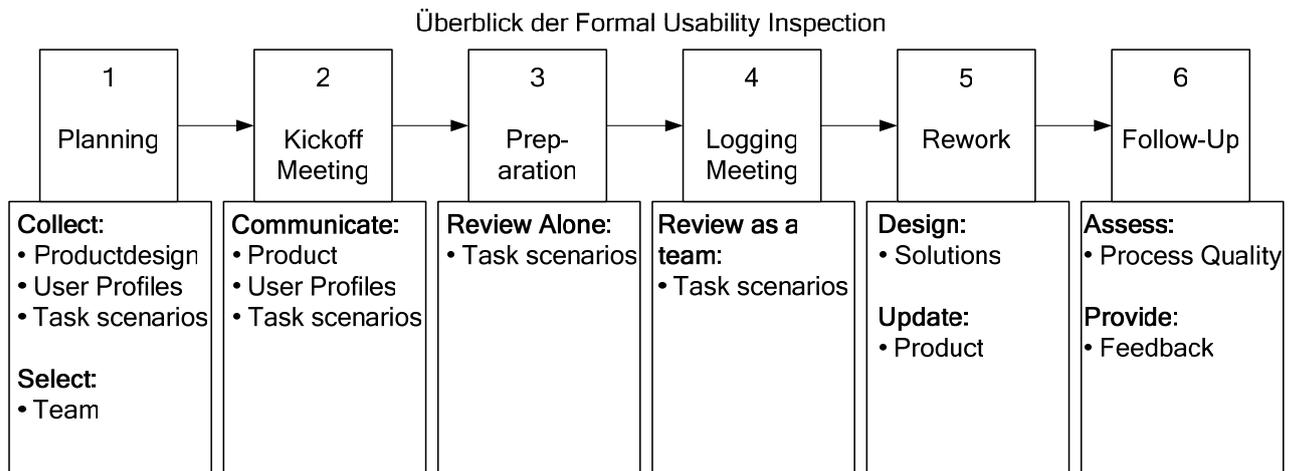


Abbildung 6 – Überblick der Formal Usability Inspection

3.4. Zusammenfassung

Wie in diesem Kapitel deutlich gemacht wurde, dienen Inspektionsmethoden oder Reviewmethoden primär dem Aufzeigen von Bedienungsproblemen. Die Bedienoberfläche einer Software, Website bzw. eines Produktes im Allgemeinen, wird durch einen oder mehrere InspektorInnen anhand gewisser Kriterien evaluiert, wodurch eine frühzeitige Beseitigung von Usabilityproblemen ermöglicht wird. Im Abschlussbericht einer Usabilityinspektion finden sich daher üblicherweise detaillierte Beschreibungen der gefundenen Usabilityprobleme und Vorschläge zu deren Lösung. Da Usabilityinspektionen über den Vorteil verfügen auch in Software-Entwicklungsprojekten mit knappem Budget eingesetzt werden zu können, werden sie häufig mit dem Begriff „Discount Usability“ bezeichnet. Viele Inspektionsmethoden lassen sich durch den Einsatz von Informationstechnologie massiv verbessern. Diese Verbesserungen umfassen insbesondere eine Minimierung des Arbeitsaufwands bei administrativen Prozessen, die Standardisierung von Daten sowie die Möglichkeit einer zentralen Speicherung von Daten, wenn diese als vorteilhaft erachtet wird. Darüber hinaus wäre eine Unterstützung des/r ProjektleiterIn bei der Auswahl der Testkriterien denkbar. Die Anwendung von Informationstechnologie bei Inspektionsprozessen unterstützt also die GutachterInnen bzw. InspektorInnen und minimiert den Organisationsaufwand möglicher ModeratorInnen und ProjektleiterInnen.

Tabelle 1 – Überblick der Usability Inspektionsmethoden

	Heuristic Evaluation	Cognitive Walkthrough	Heuristic Walkthrough	Plurastic Walkthrough	Guideline Reviews	Consistency Inspection	Formal Usability Inspection
Herangehensweise	Heuristiken	Tasks	Heuristiken	Scenarios	Guidelines	Standards	Scenarios
Geschwindigkeit	schnell/ mittel	mittel/ langsam	mittel/ langsam	mittel/ langsam	langsam	langsam	mittel
Kosten	niedrig	niedrig	niedrig	niedrig	mittel	mittel	niedrig
Anwendungsphase	Design- und Testphase	Design- und Testphase	Design- und Testphase	Design- und Testphase	Design- und Testphase	Design- und Testphase	Design- und Testphase
Anzahl der benötigten Evaluatoren	2-8	2-8	2-8	6-12	2-8	2-8	2-8
Entwickelt von	Jakob Nielsen	Basierend auf Lewis & Polson's CE + info processing model	Michael J. Muller, Lisa Matheson, Colleen Page, and Robert Gallup	Randolph G. Bias	Sidney L. Smith and Jane N. Mosier	Dennis Wixon, Sandra Jones, Linda Tse, and George Casaday	Michael J. Kahn and Amanda Prail

Tabelle 1 bietet einen Überblick über die verschiedenen in diesem Kapitel beschriebenen Inspektionsmethoden, von denen zwei Heuristiken verwenden. Vergleicht man den für den Inspektionsprozess benötigten Zeitaufwand, so muss festgestellt werden, dass die Geschwindigkeit zwar je nach Aufwand und Flexibilität der TeilnehmerInnen variiert, aber gewisse Methoden wie Guideline Reviews und Consistency Inspection dennoch tendenziell langsamer sind als beispielsweise die Heuristic Evaluation. Die im Rahmen des Inspektionsprozesses entstehenden Kosten halten sich bei allen Methoden in Grenzen, was ein wichtiges Kriterium von Inspektionsmethoden im Allgemeinen darstellt. Die Kosten korrelieren aber selbstverständlich mit dem benötigten Zeitaufwand sowie mit der Anzahl der herangezogenen InspektorInnen und differieren daher ebenfalls je nach Projekt. Obwohl es ratsam ist, immer möglichst viele EvaluatorInnen mit ein zu beziehen, sind bis auf den Pluralistic Walkthrough alle der in Tabelle 1 angeführten Methoden bereits mit zwei EvaluatorInnen durchführbar. Darüber hinaus bieten alle Inspektionsmethoden den großen Vorteil, sowohl in der Design- als auch in der Testphase eines Produkts anwendbar zu sein. In vielen Fällen empfiehlt es sich daher, bei der zweiten Iteration eine andere Inspektionsmethode zu wählen als bei der ersten, um mögliche Schwachstellen einer Methode durch eine andere auszugleichen.

4. Entwicklung eines Tools zur Unterstützung der „Heuristic Evaluation“

Als zentrales Ziel dieser Arbeit wurde ein Tool zur Unterstützung von SoftwareexpertInnen bei der heuristischen Evaluierung von User Interfaces entwickelt. Das vorliegende Kapitel widmet sich nun einer detaillierten Beschreibung dieses Tools, von anfänglichen Zielsetzungen, über Herausforderungen bei der Implementierung bis hin zu übersichtlichen Erläuterungen des Funktionsumfangs des Tools und der Vorteile die es gegenüber einer nicht durch das Tool unterstützten Evaluierung bietet.

4.1. Anforderungen

Das im Rahmen der vorliegenden Arbeit entwickelte Tool soll UsabilityexpertInnen bei der heuristischen Evaluierung und der Bewertung mit Userinterface Guidelines unterstützen. Es soll sowohl als zentrale Informationsplattform dienen, als auch administrative Aufgaben übernehmen können.

Funktionale Anforderungen an dieses Tool umfassen das Anlegen und die Administration von Evaluierungsprojekten, sowie das „severity rating“ (Bewertungsskala) der einzelnen Guidelines. Jede/r berechnigte/r BenutzerIn kann mit diesem Tool ein Projekt anlegen und mit den notwendigen Zugangsberchtigungen versehen. ExpertInnengruppen können somit völlig getrennt voneinander Projekte evaluieren. Zudem gibt es für jede/n EvaluatorIn bei der Evaluierung eine Uploadfunktion, welche das Beifügen von Screenshots für eine genauere Beschreibung des Problems ermöglicht. Ein Fragebogen ist in diesem Tool ebenfalls enthalten um damit den Hintergrund, die Erfahrung und die Daten des/r BenutzerIn einschätzen zu können. Am Ende jeder Evaluierung steht die Kumulationsphase, in der die Ergebnisse aufgelistet werden und doppelte oder ähnliche Probleme in ein „Über-Problem“ zusammengefasst werden können. Zusätzlich werden auch alle hinzugefügten Screenshots bzw. Bilder übersichtlich dargestellt, um die gefundenen Probleme genauer abgrenzen und darstellen zu können.

Der Einsatz des Tools soll nicht nur auf das Web beschränkt sein, sondern auch als Intranetlösung verwendet werden können. Es dient dabei zur zentralisierten Speicherung, erleichterten Auswertung und administrativen Unterstützung der ExpertInnen, welche die heuristische Evaluierung durchführen.

Ein wesentlicher Vorteil des Tools besteht darin, Usability Probleme schnell zu dokumentieren und dadurch die Effektivität bei der Problemfindung zu steigern. Zusätzlich werden Formatierungsprobleme, die ohne Einsatz des Tools auftraten, vermieden und somit Zeit und Ressourcen gespart.

4.2. Prozess

Der Prozess der verteilten heuristischen Evaluierung besteht aus vier Arbeitsschritten. Der erste Arbeitsschritt ist die Vorbereitung, welche das Einstellen der Settings und die Projektdefinition umfasst. Danach kann die Evaluierungsphase gestartet werden. Während dieser Phase sollen die EvaluatorInnen alle Usabilityprobleme des zu prüfenden Objekts im Tool festhalten. Die Aufgabe des/r ProjektadministratorIn ist in dieser Phase auf die Beobachtung des Projektfortschritts beschränkt. Der dritte Arbeitsschritt besteht in der Konsolidierung, bei der der/die ProjektdadministratorIn ähnliche oder sich überschneidende Probleme kumulieren oder gruppieren kann, sodass bei der nachträglichen Berichterstellung eine gegliederte Liste als Output entsteht. Die einzelnen Phasen der verteilten heuristischen Evaluierung sollen nun mit Hilfe von UML 2.0 Aktivitätsdiagrammen (Abbildung 7) veranschaulicht werden (Kecher, 2006).

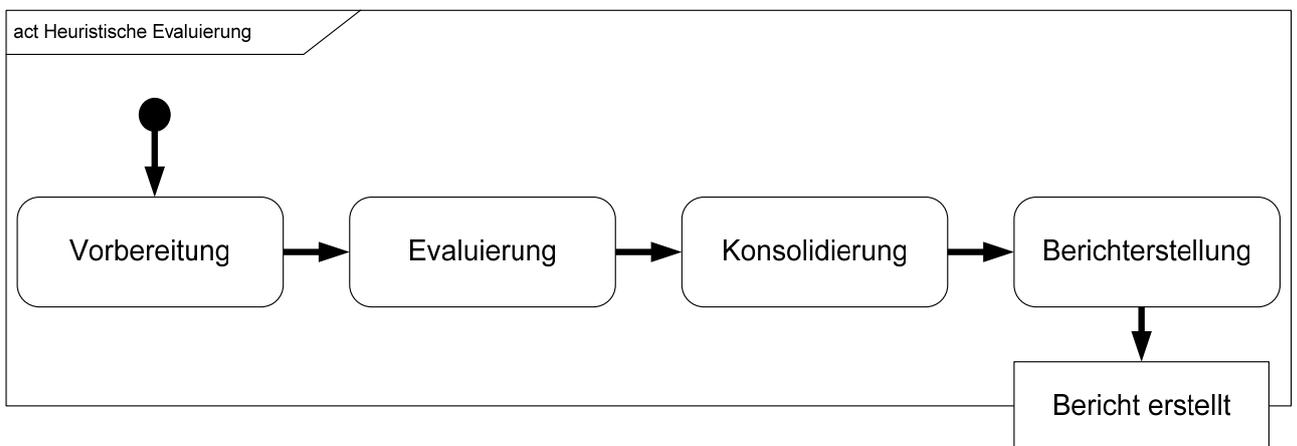


Abbildung 7 - Aktivitätsdiagramm - Distributed Heuristic Evaluation Prozess

4.2.1. Vorbereitung

Der/Die SuperadministratorIn übernimmt die Rolle des Webmasters und kann alle Funktionen ausführen, welche auch das Anlegen der Projektadministratorenaccounts umfassen. ProjektadministratorInnen und SuperadministratorInnen haben die Möglichkeit, die Settings des Tools zu bearbeiten. Dies beinhaltet zunächst die Bearbeitung der Heuristik-Vorlagen (Heuristic Sets), in denen Richtlinien mit Beschreibungen und Beispielen verfasst, editiert, gelöscht und gruppiert werden können. Darüber hinaus können Projekt- und SuperadministratorInnen aber auch die Fragebogen-Vorlagen (General Questions Sets) verfassen, editieren und löschen. Nachdem die Vorlagen erstellt oder bearbeitet wurden, kann der/die ProjektadministratorIn nun ein Projekt anlegen. Hier erfolgen zunächst allgemeine Angaben, wie der Titel, die Beschreibung, und der entsprechende Link zum Projekt. Danach werden die Heuristic Sets, ein General Questions Set und ein Rating Set (Bewertungsskala), sowie die beteiligten EvaluatorInnen und ProjektadministratorInnen ausgewählt und ein Datum, bis zu dem die Evaluierung abzuschließen ist, festgelegt. Das angelegte Projekt kann bis zum Zeitpunkt seiner Veröffentlichung bearbeitet werden, und sobald das Projekt released wird erhält jede/r EvaluatorIn ein Email, in dem der Beginn der Evaluierung bekannt gegeben wird (vgl. Abbildung 8).

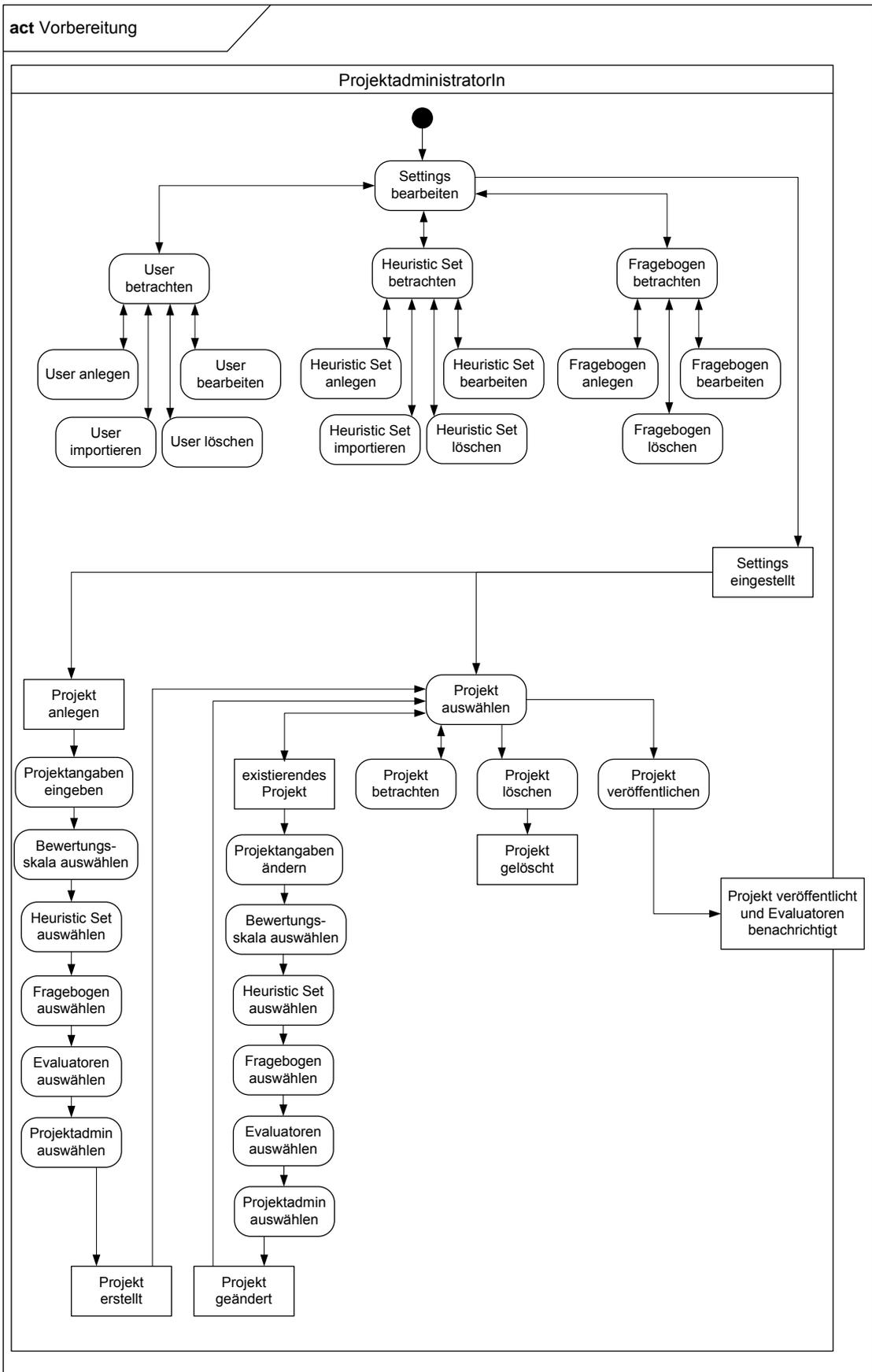


Abbildung 8 - Aktivitätsdiagramm – DHEP Vorbereitung

4.2.2. Evaluierung

Die EvaluatorInnen können sich, sobald das Projekt veröffentlicht wurde, einloggen und das angegebene Projekt bis zum Ende der Evaluierungsphase evaluieren. Jede/r EvaluatorIn muss zuerst ein Evaluierungsprojekt auswählen, danach kann er/sie den Fragebogen beantworten und die Heuristiken betrachten. Falls das geprüfte Usabilityobjekt gegen eine der Heuristiken verstößt, dokumentiert der/die EvaluatorIn ein Usabilityproblem und kann auch einen Screenshot beifügen. Der/Die ProjektadministratorIn kann den Fortschritt der EvaluatorInnen während der Evaluierungsphase beobachten und falls nötig das Finalisierungsdatum ändern oder das Projekt vorzeitig schließen. Sobald das Projekt geschlossen wurde beginnt die Konsolidierungsphase (vgl. Abbildung 9).

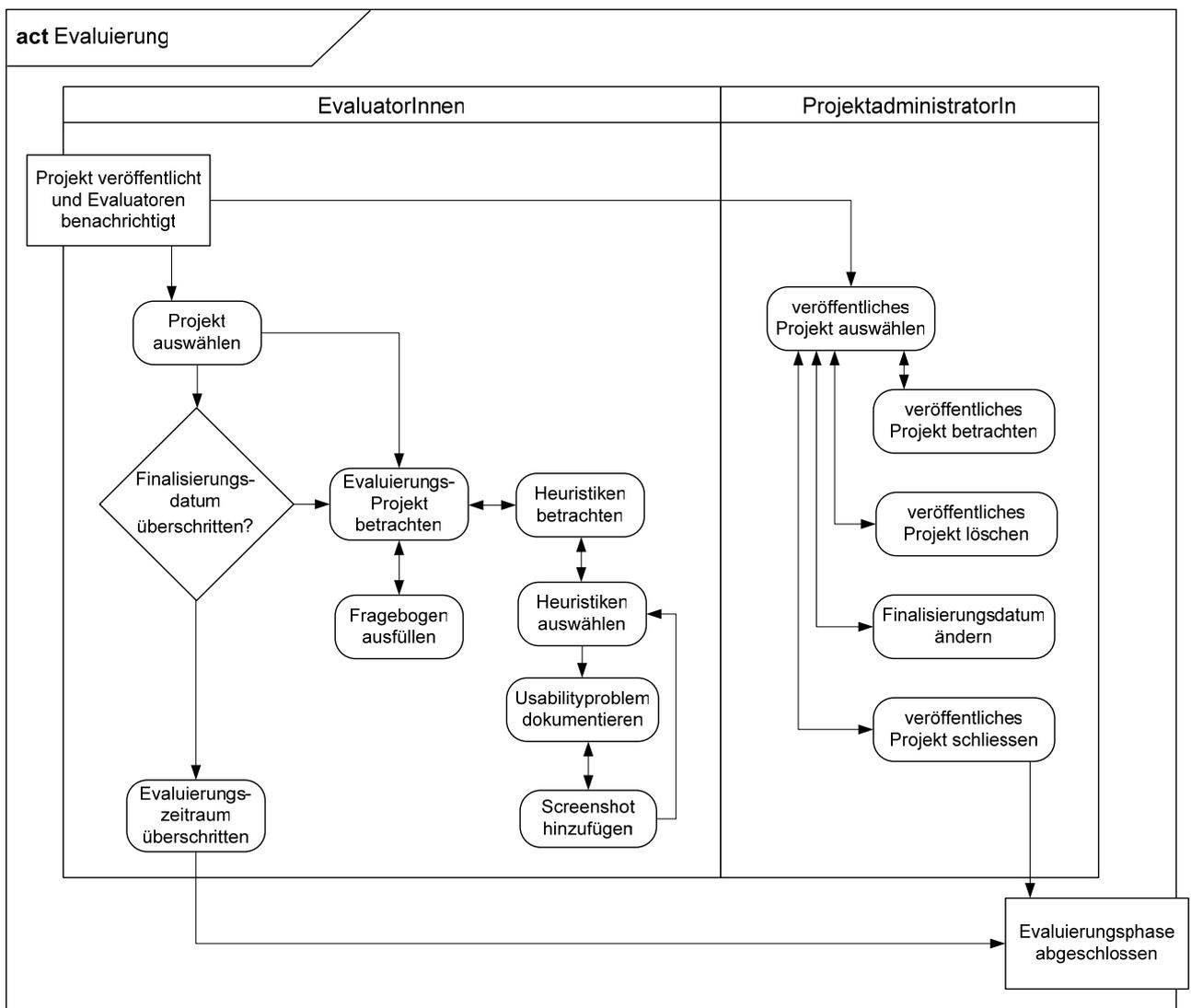


Abbildung 9 - Aktivitätsdiagramm – DHEP Evaluierung

4.2.3. Konsolidierung

Die Kumulationsphase beginnt, sobald die Evaluierung endgültig abgeschlossen wurde. In dieser Phase muss der/die ProjektadministratorIn ähnliche oder zusammenhängende Probleme erkennen, und diese in ein „Über“-Problem zusammenfassen. Im Anschluss daran kann die Gruppierungsfunktion des Tools genutzt werden, um die Probleme nach inhaltlichen Schwerpunkten zu ordnen und so einen übersichtlich gegliederten Report zu erzielen (vgl. Abbildung 10).

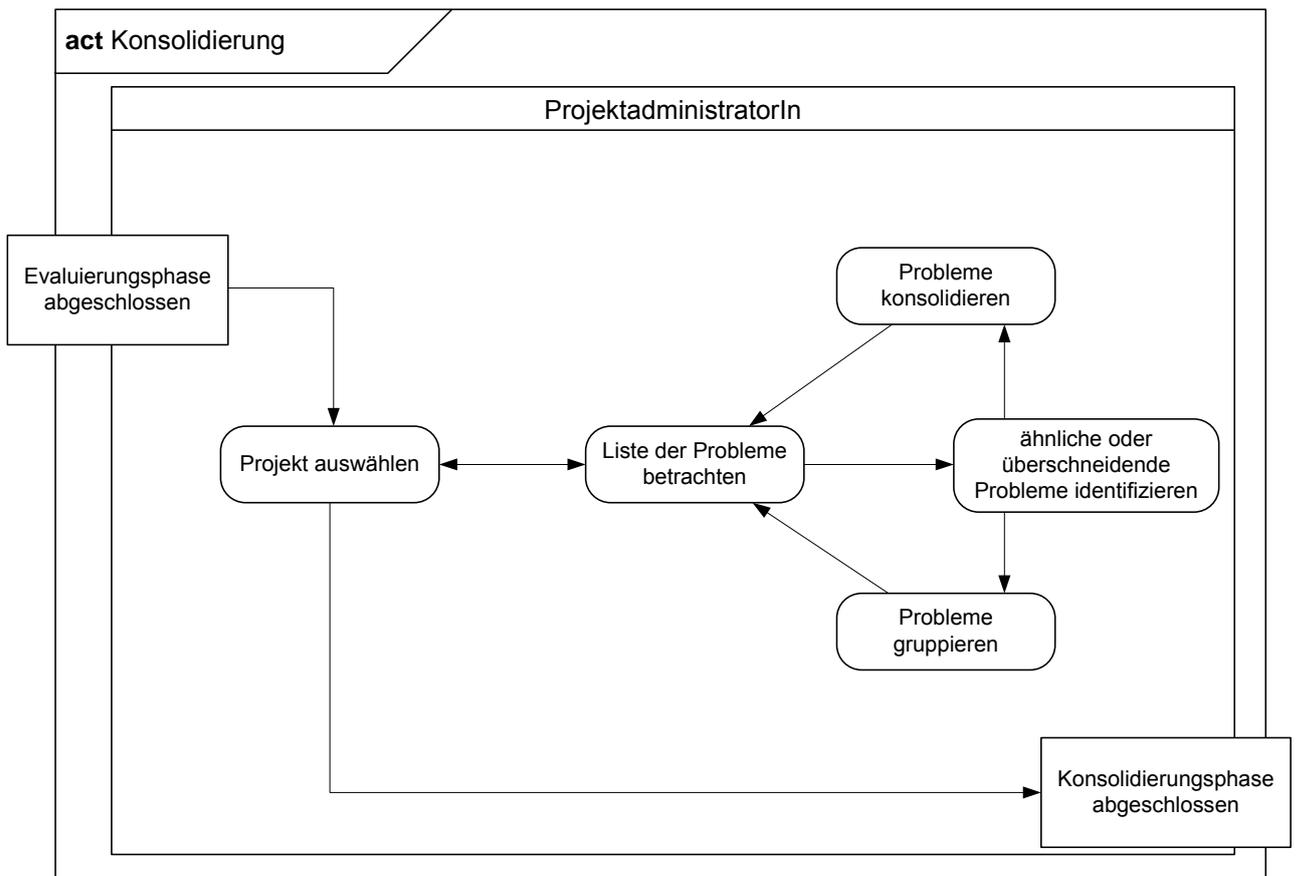


Abbildung 10 - Aktivitätsdiagramm – DHEP Konsolidierung

4.2.4. Berichterstellung

Der/die ProjektadministratorIn kann nach der Konsolidierungsphase jederzeit ein Projekt auswählen und den Bericht drucken oder exportieren (vgl. Abbildung 11).

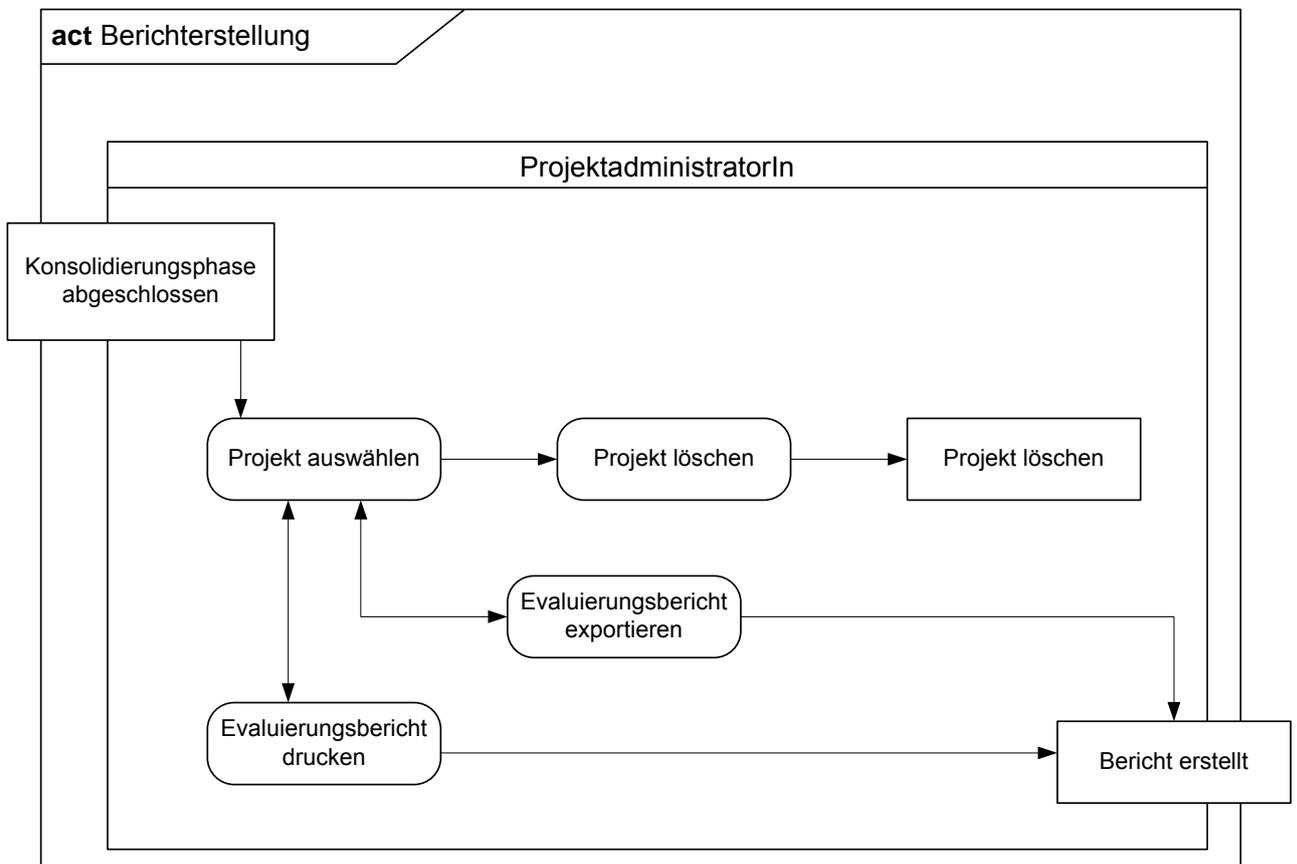


Abbildung 11 - Aktivitätsdiagramm – DHEP Berichterstellung

4.3. Umsetzung

4.3.1. Entwicklung & Software Architektur

Am Beginn der Entwicklung des Tools wurde in einer ersten Paper Prototyping Session (Snyder, 2003) innerhalb der Forschungsgruppe Industrial Software an der TU Wien ein erster Grundriss des Tools geschaffen. Im Anschluss daran erfolgte zunächst die Erstellung des ersten User Interface und dessen Präsentation, welche eine konstruktive Diskussion der Oberfläche und deren anschließende Überarbeitung nach sich zog. Erst nach mehreren Iterationen, während der noch zahlreiche Details verbessert und das Layout verändert wurde, konnte letztendlich eine Beta-Version erstellt werden, die allen ursprünglich gestellten Anforderungen gerecht wurde.

An dieser ersten Version des Tools wurde ein Software Qualitätstest und ein User Acceptance Test durchgeführt. Dabei konnten noch einige Probleme aufgezeigt und behoben werden. Dadurch wurde eine Perfektionierung der vorliegenden finalen Version des Tools vorgenommen. In Kapitel 5 wird dies näher erläutert.

Bei der Wahl der Softwarekomponenten wurde eine Webapplikation auf Basis des LAMP Designs (Kunze, 1998) entwickelt. LAMP steht für den kombinierten Einsatz von Linux Anwendungen, um dynamische Webseiten zur Verfügung zu stellen. Die einzelnen Buchstaben stehen für die verwendeten Komponenten: Linux, Apache, Mysql, PERL/PHP. Es wurde die bereits etablierte PHP (2007) Version 4.x mit einer Datenbankbindung genutzt. Bei der Installation von PHP musste die Grafikbibliothek GD frei geschaltet werden, um die Funktionalität für Screenshot Previews zu gewährleisten. Als Datenbank wurde MySQL 4.x (MySql AB 2007) gewählt. Somit wurden ausschließlich OpenSource-Packages verwendet, wodurch die Ansprüche einer Discount Usability Methode hinsichtlich Kosteneffizienz erfüllt werden konnten.

Model-View-Controller Darstellung

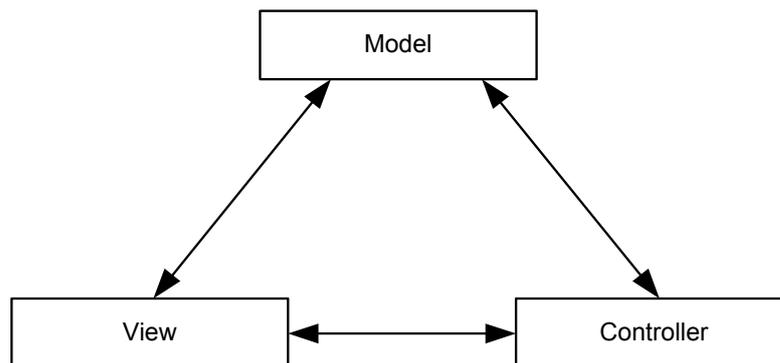


Abbildung 12 – MVC Darstellung

Als Software Architektur wurde der Model-View-Controller (MVC)-Ansatz (Abbildung 12) gewählt, und das Tool auch danach ausgerichtet (Dustdar, 2003). Der Begriff Model-View-Controller (Krasner, 1988) bezeichnet ein Architekturmuster zur Aufteilung von Softwaresystemen in drei Einheiten: Datenmodell (engl. *Model*), Präsentation (engl. *View*) und Programmsteuerung (engl. *Controller*). Ziel des Modells ist ein anpassungsfähiges Softwaredesign, um unter anderem eine spätere Änderung oder Erweiterung einfach zu gestalten und die Wiederverwendbarkeit der einzelnen Komponenten zu ermöglichen. Der Ansatz hilft, bei großen Anwendungen eine gewisse Übersicht und Ordnung zu gewährleisten, indem die Komplexität durch Modularität vermindert wird.

Das Modell befindet sich in der Datenbank, alle Controllerfunktionen im Verzeichnis „functions“ und die restlichen Dateien dienen lediglich zur Präsentation des Modells.

Der View des Webtools basiert auf den W3C (2007) konformen HTML 4.01 Transitional und CSS, was die Anpassung durch Trennung des Inhalts vom Aussehen erleichtert. Dies ermöglicht eine breite Browser-Kompatibilität und jeder Browser mit aktiviertem Javascript (Zakas, 2005) kann die Website annavigieren und bedienen. Javascript wurde um erweiterte interaktive Aktionen zu gewährleisten als clientseitige Skriptsprache genutzt. Die Controllerfunktionen dienen zur Kommunikation mit der Datenbank und werden nur bei Benutzung geladen und ausgeführt. Um dies zu illustrieren wird ein reales Beispiel aus der Webapplikation vorgestellt, welches den Ablauf des Logins darstellt und die Interaktion zwischen den Einheiten deutlich macht.

In Abbildung 13 greift der/die BenutzerIn mit dem Webbrowser auf den View „index.php“ zu. Der View kommuniziert mit dem Controller, der die Session Funktionen bereitstellt und den Loginstatus des/r BenutzerIn im Modell überprüft. Da der/die NutzerIn im

Modell nicht eingeloggt ist, signalisiert der Controller dem View, dass der/die NutzerIn zur Loginseite weitergeleitet werden muss.

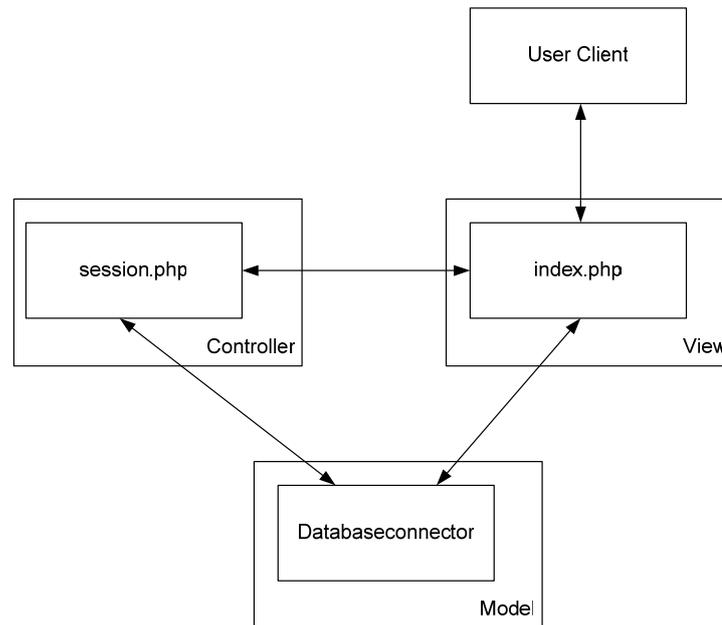


Abbildung 13 – Erster Aufruf der Webseite / MVC

4.3.2. Rollenbeschreibung

4.3.2.1. Rolle des/der ProjektadministratorIn / SuperadministratorIn

Der/Die ProjektadministratorIn und der/die SuperadministratorIn verfügen in der Distributed Heuristic Evaluation über dieselbe Rolle, wobei sich nur ihre unterschiedlichen Webadministratorenrechte unterscheiden. So sind beide unter anderem für die Organisation des Projekts und das Training der EvaluatorInnen zuständig, was in der klassischen Heuristischen Evaluation dem Aufgabenbereich des/der ProjektleiterIn entspricht. Bei der durch das Tool unterstützten Evaluierung muss der/die ProjektadministratorIn das Projekt definieren. Dies umfasst die Einteilung der EvaluatorInnen und die Auswahl der Heuristiken, der Bewertungsskala und des Fragebogens. Sobald der/die ProjektadministratorIn das definierte Projekt anschließend veröffentlicht, werden alle zuvor eingeteilten EvaluatorInnen darüber benachrichtigt, dass die Evaluierungsphase beginnt. Während die EvaluatorInnen das Usabilityobjekt überprüfen, kann der/die ProjektadministratorIn jederzeit ihre Fortschritte beobachten und notfalls auch die Deadline verlängern. Nach der Evaluierungsphase kann der/die ProjektadministratorIn die gefundenen Probleme mit Hilfe des Tools in der so

genannten Kumulationsphase weiter bearbeiten, sodass schließlich ein Evaluierungsbericht generiert werden kann.

4.3.2.2. Rolle des/der EvaluatorIn

Nach dem Training wird der/die EvaluatorIn von dem/r ProjektadministratorIn als EvaluatorIn im Tool registriert und einer Projektgruppe zugeteilt. Der/Die EvaluatorIn kann sich nach dem Start der Evaluationsphase in die Webapplikation einloggen und das Evaluierungsprojekt auswählen. Sobald das Evaluierungsprojekt aufgerufen wurde, können die EvaluatorInnen nun das Interface alleine inspizieren und sollen anhand der verschiedenen Heuristiken Probleme des zu prüfenden Objekts identifizieren und festhalten, sowie einen Fragebogen beantworten. Nachdem die EvaluatorInnen keine Probleme mehr feststellen können und den Fragebogen ausgefüllt haben, ist die Evaluierung für sie beendet. Die Rolle des/r EvaluatorIn weicht von jener bei der klassischen Evaluierung kaum ab und unterscheidet sich nur durch die Möglichkeit der räumlichen Entfernung und der Dokumentierung mit Hilfe des Tools.

4.3.3. User Interface

Bei der Entwicklung des Tools wurden drei verschiedene Benutzerrollen mit verschiedenen Rechten implementiert. In diesem Kapitel werden die Funktionalitäten der verschiedenen BenutzerInnen mit den jeweiligen Rechten erläutert. Beginnend bei dem/r EvaluatorIn mit den niedrigsten Rechten werden aufbauend über den/die ProjektadministratorIn bis zum/zur SuperadministratorIn die Funktionen und Unterschiede erläutert.

4.3.3.1. EvaluadorIn

Die Sektion „Evaluation“ (Abbildung 14) ist für jede/n registrierte/n BenutzerIn einsehbar. Hier werden alle offenen Evaluierungsprojekte für den/die jeweils eingeloggte/n BenutzerIn angezeigt und man kann zwischen drei verschiedenen Ansichten wählen:



Abbildung 14 - Evaluationsübersicht

General Questions

In diesem Teil (Abbildung 15) wird dem/r BenutzerIn ein Fragebogen für vorher festgelegte Hintergrundinformationen präsentiert, sodass dem/der ProjektadministratorIn mehr Information über die Person und die jeweilige Testumgebung erhält.

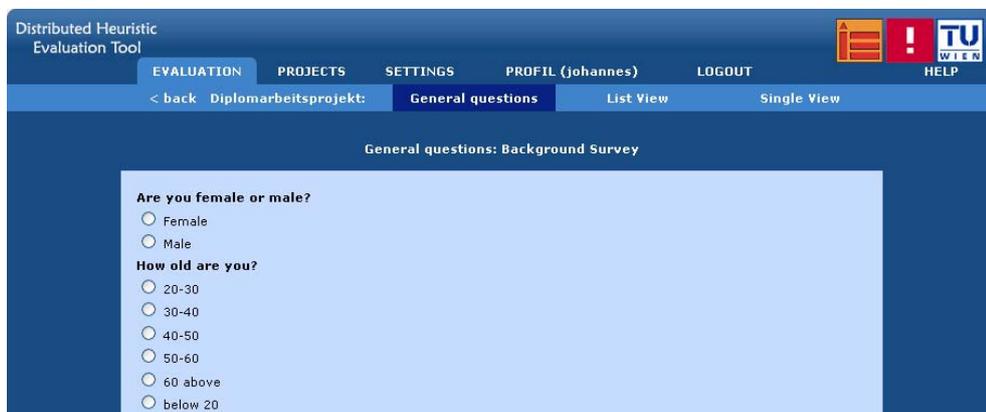


Abbildung 15 – Screenshot der Evaluationsansicht General Questions

Heuristic – List View

Dieser Teil (Abbildung 16) der Evaluierungssektion besteht in einer Listenansicht der Heuristiken. Daraus können die EvaluatorInnen einen Überblick über die Richtlinien gewinnen um dann direkt zu einer speziellen Richtlinie zu navigieren und ein Usabilityproblem dokumentieren zu können.

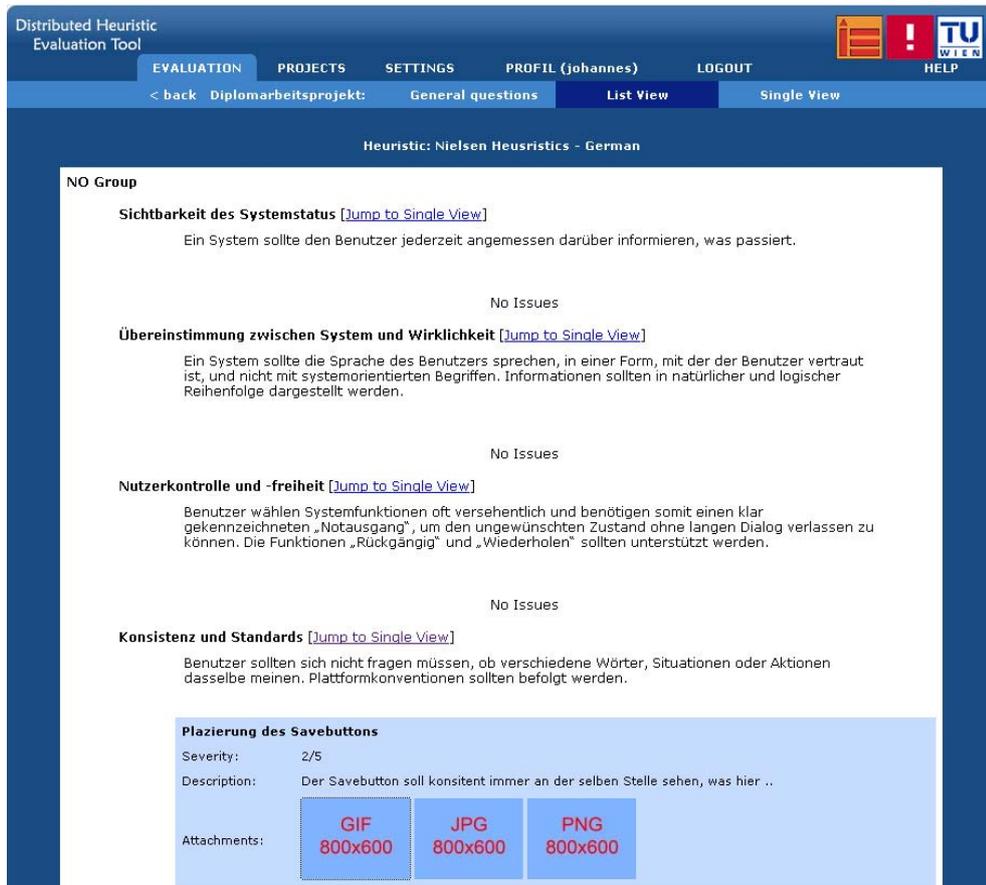


Abbildung 16 - Screenshot der heuristischen Evaluationsansicht – List View

Heuristic – Single View

Der/Die EvaluatorIn kann in dieser Sektion (Abbildung 17) Usabilityprobleme festhalten. Hierbei muss er/sie den Titel, die Beschreibung und das Severity Rating angeben, um ein Problem hinzufügen zu können. Zusätzlich können dem Problem auch noch Bilder (jpg, gif, png) für Screenshots beigefügt werden. Unter diesem Eingabeformular sind alle Usabilityprobleme aufgelistet, welche bereits zuvor dokumentiert wurden.

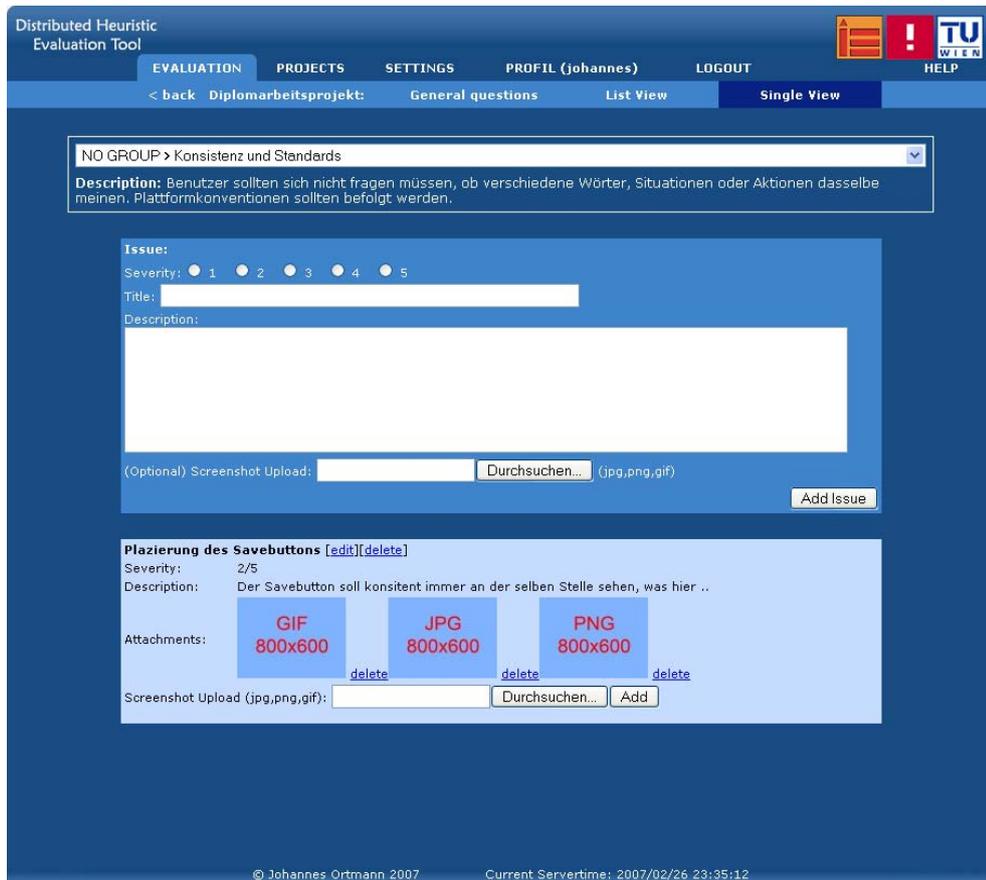


Abbildung 17 – Screenshot der heuristischen Evaluationsansicht – Single View

Profil

Die Menüoption „Profil“ (Abbildung 18) ist für jede/n BenutzerIn vorhanden und dient zur Editierung der eigenen Userdaten.

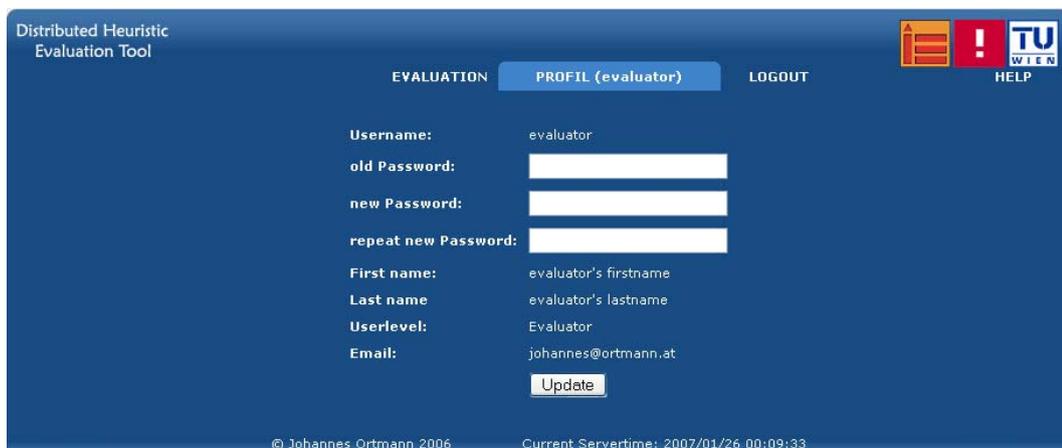


Abbildung 18 - Profil

Login/Logout

Diese Funktionen dienen zur eindeutigen Identifizierung für die Sessionsteuerung des Evaluationstools.

4.3.3.2. ProjektadministratorIn

Projects

Diese Option ist nur für ProjektadministratorInnen und SuperadministratorInnen einsehbar, da nur diese die geeignete Zugangsberechtigung haben. In dieser Sektion können sämtliche Projektaktionen im Verlauf einer Evaluierung durchgeführt werden (Abbildung 19).

Distributed Heuristic Evaluation Tool

EVALUATION PROJECTS SETTINGS PROFIL (Johannes) LOGOUT

ADD PROJECT

Preview/not released Projects:

ID	Projectname	Status	Creation	Finalization	Options
No Projects					

released Projects:

ID	Projectname	Status	Creation	Finalization	Options
58	test	Evaluationphase	02.01.2007	2.1.2007	[reset finalization date] [view] [delete] [close]
54	Test-Project	Evaluationphase	22.10.2006	15.11.2006	[reset finalization date] [view] [delete] [close]
52	A very new Project	Evaluationphase	22.10.2006	24.1.2007	[reset finalization date] [view] [delete] [close]
60	Realtest	Evaluationphase	06.01.2007	10.1.2007	[reset finalization date] [view] [delete] [close]

Closed Projects:

ID	Projectname	Status	Creation	Finalization	Options
53	An old project	Cumulationphase	22.10.2006	22.10.2006	[view] [Cumulation] [generate Report] [delete]
51	My new Project	Cumulationphase	22.10.2006	23.10.2006	[view] [Cumulation] [generate Report] [delete]

© Johannes Ortmann 2006 Current Servertime: 2007/01/10 01:20:36

Abbildung 19 - Projektübersicht

Dies beginnt mit dem Hinzufügen eines neuen Projekts (Abbildung 20), dessen Titel, Beschreibung und optionalem Link. Zusätzlich müssen ein Heuristik Set, ein General Questions Set, ein Rating Set, sowie die beteiligten ProjektadministratorInnen und EvaluatorInnen ausgewählt werden. Die einzige Einschränkung besteht darin, dass ProjektadministratorInnen nur Projekte einsehen können, zu welchem sie als ProjektadministratorIn hinzugefügt worden sind.

Distributed Heuristic Evaluation Tool

EVALUATION PROJECTS SETTINGS PROFIL (johannes) LOGOUT

HELP

Add new project:

Title:*	Diplomarbeitprojekt
Description:*	Eine Projektbeschreibung kann man hier platzieren.
Link:*	http://ortmann.at/diplomarbeit/
Finalization Date:	27 February 2007

Settings for this Project:

Heuristics set	Nielsen Heuristics - German
General questions set	Background Survey
Rating set	Licert Scale 1-5

Evaluators in Project	Available Evaluators
Ortmann Johannes	Admin Projekt Athur Evalu evaluator's lastname evaluator's firstname Jahoda Michael Költringer Thomas projectadmin's lastname projectadmin's firstname Strasser Thomas

Projectadmins	Available Admins
Ortmann Johannes	Admin Projekt Jahoda Michael Költringer Thomas projectadmin's lastname projectadmin's firstname thomas thomas

Create

* All Fields need to be filled out

© Johannes Ortmann 2007 Current Servertime: 2007/02/26 23:14:02

Abbildung 20 – Projekt anlegen

Vom Anlegen des Projekts bis zu dessen Veröffentlichung (Release) kann der/die ProjektadministratorIn noch Änderungen an der Zusammenstellung des Projekts durchführen. Erst nach der Veröffentlichung können die EvaluatorInnen mit der Überprüfung des Projektes beginnen. Die Administratorinnen können aber auch während der Evaluierung den Status einsehen.

Sobald das Finalisierungsdatum überschritten wurde, wird der Zugang für die EvaluatorInnen gesperrt und das Projekt kann für die Kumulationsphase geschlossen werden. Der/Die ProjektadministratorIn hat nun die Möglichkeit, mehrere gleiche oder ähnliche Probleme zu einem Problem zusammenzufassen und auch Gruppierungen zu erstellen (Abbildung 21).

Distributed Heuristic Evaluation Tool

EVALUATION PROJECTS SETTINGS PROFIL (johannes) LOGOUT

BACK

Cumulation Phase - Merging

Project: Diplomarbeitprojekt
 Description: Eine Projektbeschreibung kann man hier platzieren.
 Link: http://ortmann.at/diplomarbeit/
 Due til: 28. 2. 2007

Merging View Grouping View

No Issues

Frames from Johannes Ortmann
Violates: Ästhetisches und minimalistisches Design
 Severity: 4/5
 Description: Dialoge sollten keine Informationen beinhalten, die irrelevant sind oder selten benötigt werden. Jede zusätzliche Informationseinheit in einem Dialog konkurriert mit den relevanten Informationen.
 Attachments: No Attachments

Systemstatus inkorrekt from Johannes Ortmann
Violates: Sichtbarkeit des Systemstatus
 Severity: 4/5
 Description: Ein System sollte den Benutzer jederzeit angemessen darüber informieren, was passiert.
 Attachments: No Attachments

Plazierung des Savebuttons from Johannes Ortmann
Violates: Konsistenz und Standards
 Severity: 2/5
 Description: Benutzer sollten sich nicht fragen müssen, ob verschiedene Wörter, Situationen oder Aktionen dasselbe meinen. Plattformkonventionen sollten befolgt werden.
 Attachments: **GIF 800x600** **JPG 800x600** **PNG 800x600**

Merge Issue

BACK

© Johannes Ortmann 2007 Current Servertime: 2007/02/28 15:56:46

Abbildung 21 – Projektadministration - Kumulationsphase

Nach dem Abschluss des Projekts durch den/die AdministratorIn kann ein Report generiert werden, welcher eine ausführliche Problemliste - geordnet nach Severity, Gruppen und den dazugehörigen Pre-Evaluierungsfragen - enthält.

Settings

Die Sektion Settings beinhaltet drei Untergruppen (Heuristic, General Questions, Users), welche nur für ProjektadministratorInnen und SuperadministratorInnen zugänglich sind.

Heuristic

Diese Sektion dient zum Anlegen und zur Bearbeitung der Vorlagen für die Heuristik Sets für die Projekte (Abbildung 22). Ein Heuristik Set enthält Heuristikgruppen bzw. die Heuristiken selbst. Eine Heuristik muss einen Titel, eine Beschreibung und ein Beispiel umfassen. Daneben besteht auch die Möglichkeit eines Imports mittels Comma-Separated-Values Dateien.

Distributed Heuristic Evaluation Tool

EVALUATION PROJECTS **SETTINGS** PROFIL (johannes) LOGOUT

Heuristic General questions Users

HELP

Edit heuristic

Name of the heuristic set: Nielsen Heuristics - German

Add guideline group:

Name of the guideline:

Guideline description:

Example:

Select group: No group

No group

0.1 Sichtbarkeit des Systemstatus ↑ ↓ 📄 ✕	
Description:	Ein System sollte den Benutzer jederzeit angemessen darüber informieren, was passiert.
Example:	

0.2 Übereinstimmung zwischen System und Wirklichkeit ↑ ↓ 📄 ✕	
Description:	Ein System sollte die Sprache des Benutzers sprechen, in einer Form, mit der der Benutzer vertraut ist, und nicht mit systemorientierten Begriffen. Informationen sollten in natürlicher und logischer Reihenfolge dargestellt werden.
Example:	

0.3 Nutzerkontrolle und -freiheit ↑ ↓ 📄 ✕	
Description:	Benutzer wählen Systemfunktionen oft versehentlich und benötigen somit einen klar gekennzeichneten „Notausgang“, um den ungewünschten Zustand ohne langen Dialog verlassen zu können. Die Funktionen „Rückgängig“ und „Wiederholen“ sollten unterstützt werden.
Example:	

Abbildung 22 – Heuristic Sets bearbeiten

General Questions

Diese Sektion beinhaltet Funktionen zum Anlegen von Fragen mit vorgegebenen oder freien Antwortmöglichkeiten. Die Möglichkeiten zu deren Sortierung, Bearbeitung und Löschung sind ebenfalls im Tool enthalten (Abbildung 23).

Distributed Heuristic Evaluation Tool

EVALUATION PROJECTS SETTINGS PROFIL (johannes) LOGOUT

Heuristic General questions Users

HELP

Edit general questions

Name of general questions: Background Survey Update

Question:

Answer type: Text Answer Other Field:

Add

1.

Question: Are you male or female?

Answer Type: Radio Buttons + Other Field: off

Answers: Add

1. Female [delete]

2. Male [delete]

2.

Question: How old are you?

Answer Type: Radio Buttons + Other Field: off

Answers: Add

1. 40-50 [delete]

Abbildung 23 – General Questions Sets bearbeiten

Users

Diese Sektion dient dem Usermanagement. Es können UserInnen angelegt, editiert und gelöscht werden (Abbildung 24). Ein Useraccount umfasst sechs Daten: einen Vornamen, einen Nachnamen, eine Email-Adresse, ein Passwort, einen Benutzernamen und die Benutzerrechte. Ebenfalls im Tool enthalten ist die Möglichkeit eines Imports einer Datei mit Userdaten. Der/Die ProjektadministratorIn kann jedoch nur EvaluatorInnen anlegen und diese editieren.

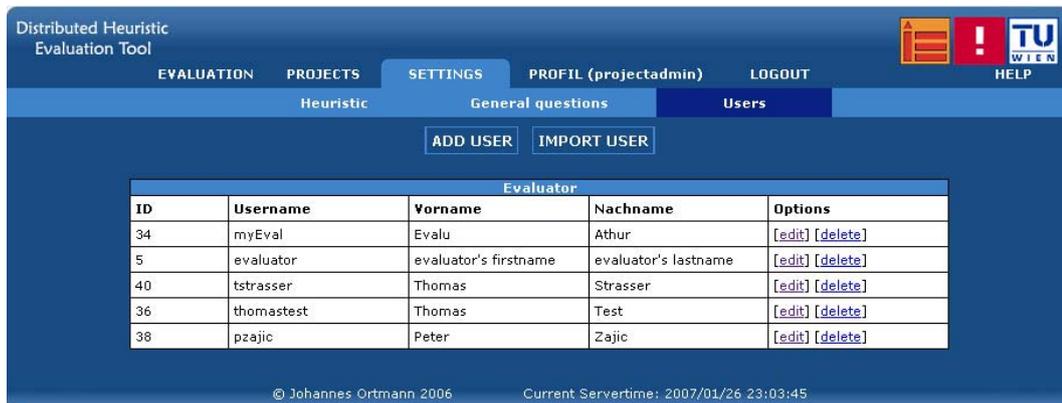


Abbildung 24 – Projektadmin Usermanagement

4.3.3.3. SuperadministratorIn

Users

Diese Sektion dient dem Usermanagement. Es können UserInnen angelegt, editiert und gelöscht werden (Abbildung 25). Ein Useraccount umfasst sechs Daten: einen Vornamen, einen Nachnamen, eine Email-Adresse, ein Passwort, einen Benutzernamen und die Benutzerrechte. Ebenfalls im Tool enthalten ist die Möglichkeit eines Imports einer Datei mit Userdaten. Der größte Unterschied zum Projektadmin in dieser Sektion ist, dass auch ProjektadministratorInnen und SuperadministratorInnen angelegt, editiert und gelöscht werden können. Zusätzlich kann sich ein/e SuperadministratorIn die IP Logs jedes/r UserIn anzeigen lassen, um die Transparenz der Evaluierung zu gewährleisten.

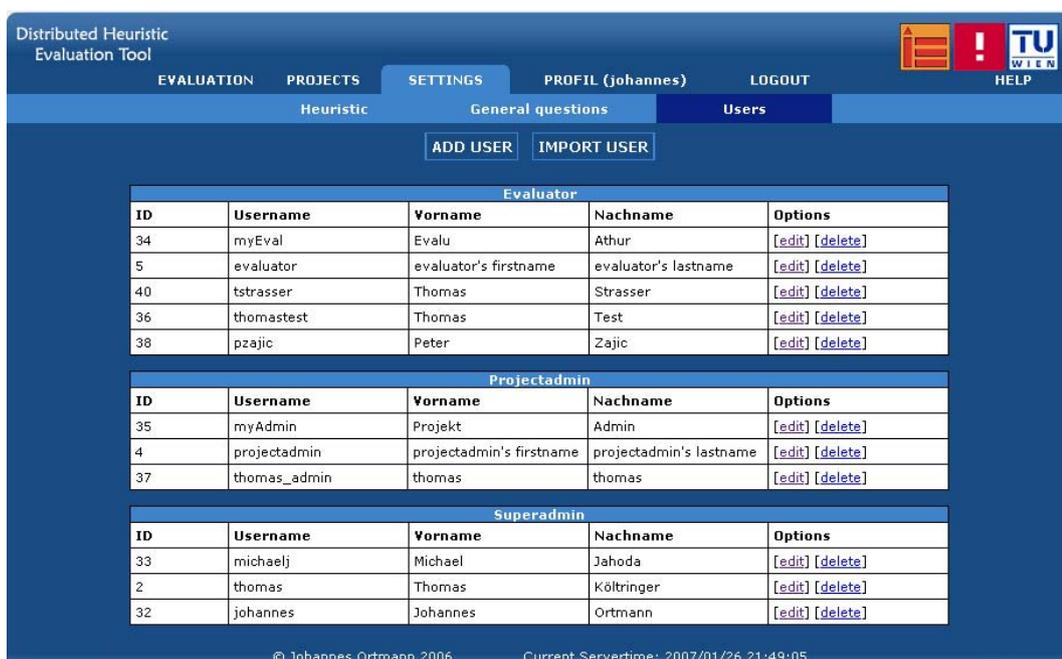


Abbildung 25 – Superadmin Usermanagement

Projects

Diese Option ist nur für ProjektadministratorInnen und SuperadministratorInnen einsehbar, da nur diese über die geeignete Zugangsberechtigung verfügen. In dieser Sektion können alle möglichen Projektaktionen im Verlauf einer Evaluierung durchgeführt werden. Hier kann der/die SuperadministratorIn im Gegensatz zum /zur ProjektadministratorIn alle Projekte einsehen, um die Administration des Tools zu gewährleisten.

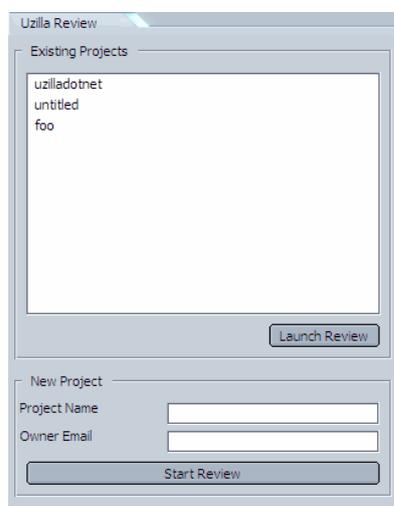
4.4. Vergleich zu anderen Tools

Das in dieser Arbeit beschriebene Tool wurde ausschließlich für die Unterstützung der heuristischen Evaluierung implementiert und soll das Konzept des Computer Aided Usability Engineering (CAUSE) verfolgen. Wie bereits dargelegt sind derzeit keine anderen Usability Tools zugänglich, die über einen ähnlichen Funktionsumfang verfügen und ebenso allgemein eingesetzt werden können. Dennoch sollen in diesem Kapitel einige Tools vorgestellt werden, die zwar nur sehr bedingt mit dem Distributed Heuristic Evaluation Tool vergleichbar sind, dessen Entwicklung aber durchaus inspiriert haben.

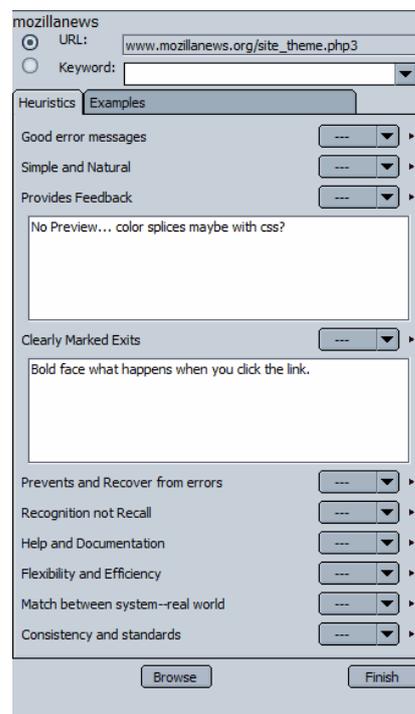
4.4.1. Mozilla UZReview

Das UZReview-Tool wurde von der Mozilla Entwickler Community (Mozilla Foundation 2003) entwickelt. Es handelt sich hierbei um eine Sidebar von Mozilla Firefox, in der Websites evaluiert werden können (Abbildung 26). Die Voraussetzung zur Benutzung des Tools ist die Installation des Browsers Mozilla und von UZReview. Mit Hilfe des Tools können Evaluierungen von Websites nach Heuristiken durchgeführt, und Probleme anhand des Keyword oder der URL dokumentiert werden. Hierbei handelt es sich um eine Open Source Lösung, die jedoch seit Jahren nicht mehr von der Community weiterentwickelt wurde. Der Grund dafür war vermutlich, dass dieses Tool nur die Fehlernotierung unterstützt, nicht aber die komplette heuristische Evaluierung. Die gefundenen Probleme werden als XML Datei auf dem lokalen PC gespeichert. Es fehlt eine gemeinsame Plattform bzw. die Möglichkeit der Kommunikation zu einer zentralen Ablage, was einen überflüssigen Overhead zur Folge hat. Die EvaluatorInnen müssen mit XML Files hantieren um Heuristiken zu wechseln oder Issues zu speichern, was zwar im Hinblick auf das Übertragungsprotokoll von Vorteil ist, den/der

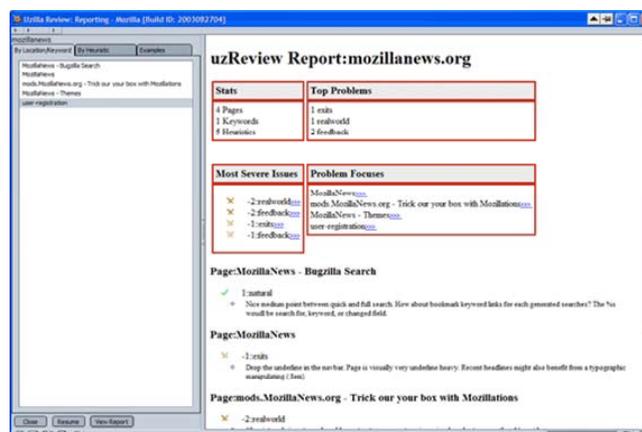
ProjektadministratorIn aber vor die mühevoll Aufgabe stellt diese XML Dateien auszuwerten und in einen Bericht zu fassen. Außerdem ist keine organisatorische Übersicht vorhanden, wo Projekte getrennt evaluiert und EvaluatorInnen bzw. ProjektadministratorInnen gewissen Projekten zugeordnet werden können. Diese organisatorischen Tätigkeiten müssen hier noch immer extern erledigt werden, was das Fehlen einer zentralen Informationsplattform umso problematischer erscheinen lässt. Es ist daher offensichtlich, dass die schwierig durchführbare Kumulation und die folglich auch erschwerte Reportgenerierung vor allem für den/die ProjektadministratorIn von Nachteil sind.



UZReview - Start Up Screen



UZReview - Conducting a Review



UZReview – Report

Abbildung 26 – Mozilla UZReview

4.4.2. QuestionPro

Ein Beispiel für ein Fragebogen-Tool stellt die Website www.questionpro.com (Abbildung 27) dar, welche sich auf Fragebögen und Umfragen spezialisiert hat und diese im großen Stil anbietet (Survey Analytics Enterprise Survey Software, 2006). Die Funktionen der Survey Software umfassen neben dem individuell gestaltbaren Fragebogen-Design auch zahlreiche Optionen zur Publikation, Datensammlung und Analyse. Die Firma kann schon auf einige Jahre Erfahrung zurückgreifen, beschränkt sich aber ausschließlich auf Surveys. Das Tool wird von den KundInnen meist benutzt um Web- oder Emailumfragen schnell und effizient durchführen zu können. Zusätzlich ersparen sich die Unternehmen jeglichen IT Aufwand, da dieser in der Gebühr enthalten ist und die Ressourcen dafür von QuestionPro bereitgestellt werden.

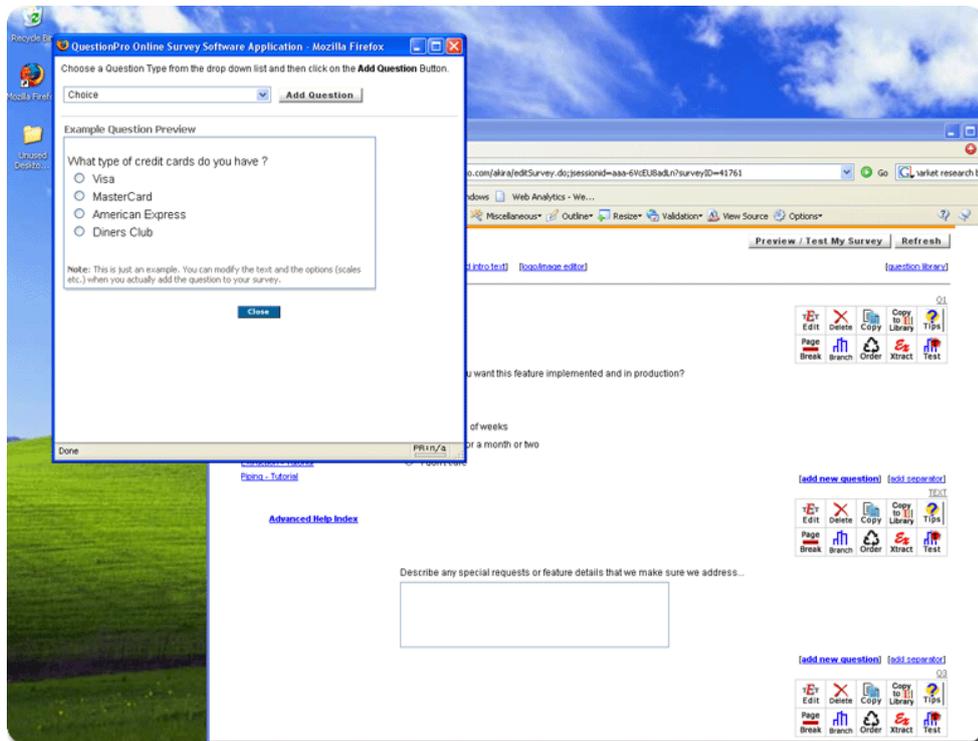


Abbildung 27 – Question Pro

4.5. Prozessunterstützung

Die computerunterstützte heuristische Evaluierung mit Hilfe des im Rahmen dieser Arbeit erstellten Prototyps zeichnet sich im Vergleich zur klassischen heuristischen Evaluierung durch eine Reihe von Vorteilen aus.

Bereits in der Vorbereitungsphase unterstützt der Prototyp den/die UsabilityexpertIn durch vordefinierte auswählbare Guidelines, welche problemlos abgeändert, neu angelegt und importiert werden können. Ein weiterer Vorteil besteht darin, dass sowohl vordefinierte Fragebögen, welche aber individuell abänderbar sind, als auch neu angelegte Fragebögen verwendet werden können. Im Anschluss daran wird eine vollständige Liste der beteiligten EvaluatorInnen und ProjektadministratorInnen erstellt, welche nach dem Release des Projekts automatisch per E-Mail verständigt werden. Dadurch gestaltet sich die Erstellung eines Evaluierungsprojekts für den/die UsabilityexpertIn deutlich unkomplizierter und weniger zeitaufwändig als bei klassischen Evaluierungsverfahren.

Auch während der Evaluierungsphase selbst sind die zahlreichen Vorteile des Prototyps gegenüber bestehenden Methoden der Evaluierung klar ersichtlich. Hier überzeugt das Webtool den/die EvaluatorIn zunächst durch die vereinfachte Eingabe von Daten im Fragebogen. Des Weiteren ist der Prototyp übersichtlich gestaltet und ermöglicht selbst das Hinzufügen von Bild-Dateien. Die bedeutendste Innovation des Webtools in der Evaluierungsphase besteht jedoch darin, dass die EvaluatorInnen gefundene Probleme in ein vorgegebenes Formular eintragen können, wodurch eine zentrale Speicherung und ein einheitliches Format der Daten gewährleistet werden.

Dies wirkt sich insbesondere in der im Anschluss erfolgenden Kumulations- und Severity Rating - Phase positiv aus, da die ProjektadministratorInnen aufgrund der zentralen Speicherung und homogenen Formatierung problemlos eine einheitliche Liste erstellen können. Zudem werden die von den EvaluatorInnen hinzugefügten Bilder automatisch als Thumbnails (Vorschaubilder) dargestellt, was den zusätzlichen Vorteil mit sich bringt, die Ladezeit der Seite gering zu halten. Darüber hinaus vereinfacht sich die Aufgabe der ProjektadministratorInnen in dieser Phase dadurch, dass mehrere zusammenhängende oder identische Problembeschreibungen in nur wenigen Schritten kumuliert werden können. Außerdem kann nach der Kumulierungsphase eine Gruppierung der Ergebnisse durchgeführt werden, welche bei der Reportgenerierung die Probleme zum Beispiel nach Themen oder Fehlerarten zusammenstellt. Am Ende der Reportgenerierung kann zusätzlich zur Vorschau auch eine Druckansicht gewählt werden, welche für die Weiterbearbeitung klar strukturiert ist.

Es erscheint daher offensichtlich, dass im Vergleich zur klassischen heuristischen Evaluierung bei der durch den Prototyp unterstützten heuristischen Evaluierung zahlreiche administrative Schritte entfallen, wodurch ein Zeitersparnis entsteht, das sich nicht zuletzt auch finanziell positiv auswirkt. Dieser Vorteil wird, wie bereits angesprochen, während aller Phasen des Evaluierungsprozesses klar ersichtlich. Weiters können durch den Einsatz des Webtools die Anfahrtskosten der beteiligten Personen minimiert werden, da aufgrund der möglichen Verwendung eigener Test Equipments (Home PC) ein flexiblerer Ablauf der Evaluierung gewährleistet ist.

5. Evaluierung

Um die Qualität des Webtools zu gewährleisten und etwaige Fehler frühzeitig auszuschließen, wurde das Tool schon ab der Fertigstellung des ersten funktionstauglichen Prototypen im Rahmen eines Software Quality Tests evaluiert. Später wurde ergänzend dazu ein User Acceptance Test durchgeführt. Diese beiden Evaluierungsverfahren werden in diesem Kapitel ebenso beschrieben wie die daraus gewonnenen Resultate.

5.1. Software Quality Test

5.1.1. Vorgangsweise und Zielsetzung

Nach der ersten Entwicklungsphase wurde ein Software Qualitätstest durchgeführt, um eine Qualitätssicherung und -verbesserung zu erzielen (Zuser, 2004). Hierbei war es vor allem wichtig, auf die Korrektheit, Konsistenz und Sicherheit des Tools zu achten. Außerdem wurden im Rahmen dieses Tests die Einhaltung der Standards und die Kompatibilität mit allen Browsern überprüft. Um das Tool anhand dieser Qualitätskriterien zu testen, führten mehrere StudentInnen der TU Wien einen Blackbox-Test durch. Ausgehend von den Anforderungen an das Tool wurden Testfälle erarbeitet, die eine Einschätzung des Funktionsumfangs und der Qualität ermöglichen sollten. Das System wurde bei diesem Test als Ganzes betrachtet, und nur sein Außenverhalten wurde in die Testergebnisse miteinbezogen. Die für diese Überprüfung heranzuziehenden Testfälle wurden anhand der Use Cases ausgewählt und nach der Häufigkeit ihrer Nutzung und ihrer Wichtigkeit im Gesamtprozess gewichtet. Bei diesen Testfällen handelte es sich um einen gesamten Evaluierungsdurchlauf mit einem zusätzlichen Qualitätstest sämtlicher Aktionen. Als Testwerkzeuge kamen marktübliche Internet-Browser sowie die eigenen PCs der StudentInnen als Umgebung zum Einsatz. Die Zielsetzung des Qualitätstests bestand darin, jede Aktion zu testen, sowie einen Belastungstest während eines praxisnahen Evaluierungsablaufs durchzuführen.

5.1.2. Ergebnisse & Erfahrungen

Die statistische Auswertung der Ergebnisse zeigte, dass die meisten Probleme in der Konsistenz der Beschriftung gefunden wurden. Es wechselten sich teilweise deutsche und englische Begriffe ab, was auf die späte Entscheidung, das Tool mit englischen Begriffen umzusetzen, zurückzuführen ist. Diese Inkonsistenz kann UserInnen verwirren und ein einheitlicher Sprachgebrauch ist die Grundbedingung dafür, dass die BenutzerInnen die Webapplikation in der Praxis einsetzen können. Die zweithäufigste Problemgruppe stellte die versteckte bzw. fehlende Bestätigung bei Benutzeraktionen dar. Bei Handlungen wie der Abspeicherung des Fragebogens fiel die Antwort des Systems nicht klar genug aus und wurde deshalb von dem/r BenutzerIn häufig nicht wahrgenommen. Dies führte dazu, dass NutzerInnen manchmal nicht wussten, ob sie eine Aktion erfolgreich durchgeführt hatten und nochmals dorthin navigierten, um die Eingabe zu überprüfen, was die Effizienz der Handhabung des Tools beträchtlich minderte.

Durch den Qualitätstest mit Hilfe mehrerer gängiger Browser konnten an der Webapplikation auch Inkonsistenzen des Layouts wahrgenommen werden. Durch die Benutzung von CSS kam es in manchen Browsern zu Problemen mit der Darstellung, die jedoch ohne großen Aufwand gelöst werden konnten. Zudem konnten in manchen Sektionen des Tools Formatierungsunterschiede in der Ausrichtung erkannt werden, und auch Hyperlinks wurden teilweise unterschiedlich gekennzeichnet, was sich störend auf das Gesamtbild der Webapplikation auswirkte.

Die beiden gravierendsten Probleme, welche jedoch sofort nach dem Qualitätstest behoben werden konnten, wurden bei der Kompatibilitätsüberprüfung durch die verschiedenen Browser gefunden. Das erste Problem umfasste die Gruppenbildung bei der Projektdefinition. Hierfür war eine Javascript Funktion nötig, die eine übersichtliche Zuteilung von EvaluatorInnen und ProjektadministratorInnen zum Projekt ermöglichte, aber mit manchen Browsern nicht kompatibel war, was ein fatales Szenario zur Folge hatte, da keine Projekte angelegt werden konnten. Dies konnte jedoch erfolgreich behoben werden. Das zweite und zugleich letzte Problem bestand in einer inkorrekten Inputweiterleitung für die Editierung von Problemen des/r EvaluatorIn. Dies führte zum Versagen der Funktion und der/die BenutzerIn konnte diese Aktion nicht ausführen.

Nach Bekanntgabe der Ergebnisse dieses Qualitätstests wurde eine weitere Iteration durchgeführt, um mit Abschluss der Arbeit eine möglichst fehlerfreie Version des Tools zu erhalten.

5.2. User Acceptance Test

5.2.1. Vorgangsweise und Zielsetzung

Die besondere Bedeutung von heuristischen Evaluierungen in der Software-Entwicklung wurde in dieser Arbeit bereits hinreichend erläutert, und es war daher selbstverständlich, dass auch die hier beschriebene Webapplikation nach ihrer Entwicklung einem User Acceptance Test durch Usability-ExpertInnen unterzogen wurde. Die Zielsetzung dieser Evaluierung bestand in der Überprüfung der Usability und der intuitiven Bedienung des erstellten Tools in einem Echtzeittest. Als EvaluatorInnen fungierten drei StudentInnen der Technischen Universität Wien, die mit Hilfe dieses Tools einige ihnen bekannte und populäre Webseiten inspizierten. Um eine möglichst genaue und nachvollziehbare Bewertung zu erstellen, wurden sie gebeten die Think-Aloud Methode anzuwenden und ihre Eindrücke zu schildern. Die Evaluierung erfolgte ohne vorherige konkrete Einschulung, um die intuitive Bedienung zu testen. Weiters wurde den StudentInnen vorgegeben, das Webtool Out-of-the-box einzusetzen. Das Ziel war hierbei, einen möglichst praxisorientierten Einsatz zu simulieren, um eventuelle Usabilityprobleme bei der Bedienung festzustellen. Danach sollten die StudentInnen einschätzen, wie gut oder wie schlecht der Prozess der heuristischen Evaluation durch das Tool unterstützt wird, und eine Bewertung der Unterstützung in den verschiedenen Phasen vornehmen, wobei ihre Benotung vor allem anhand des Kriteriums der Zeitersparnis durch das Tool erfolgte.

Die Aufgabe der StudentInnen bestand darin, im Tool ein Evaluierungsprojekt mit drei EvaluatorInnen und einem/r ProjektadministratorIn zu erstellen. Vor der Erstellung mussten sie zunächst die Heuristikvorlagen und die Fragebogenvorlagen kontrollieren, beziehungsweise je nach Einsatzgebiet editieren. Nach der Definition der Heuristik- und Fragebogenvorlagen erfolgte das Anlegen eines Projekts, wobei ein Titel, eine Beschreibung und ein Link angegeben wurden. Dann wurden die Heuristikvorlage, die Fragebogenvorlage, eine Bewertungsskala und die zuständigen ProjektadministratorInnen und EvaluatorInnen ausgewählt. Nach dem Evaluierungsstart überprüften die EvaluatorInnen die Webseiten anhand der Heuristiken im Single-View oder List-View des Tools auf Probleme. Außerdem wurden sie aufgefordert, unter ‚General Questions‘ einige persönliche Angaben zu machen. Sobald die Evaluierungsphase abgelaufen war, wurden die von den EvaluatorInnen dokumentierten Probleme vom/von der ProjektadministratorIn kumuliert und thematisch zusammengefasst, wodurch im Anschluss ein übersichtlicher und in vieler Hinsicht

aufschlussreicher Report generiert werden konnte. Dieser enthielt dann eine Auswertung der Fragebögen und eine Liste der zuvor zusammengestellten Probleme, welche die Vorlage für den Bericht durch den/die ProjektadministratorIn darstellte.

5.2.2. Ergebnisse & Diskussion

Die kritische Auseinandersetzung mit den Ergebnissen ergab zwei grundsätzliche Aussagen:

Einerseits konnten durch die Evaluierungen wichtige Erkenntnisse über Schwachstellen des Tools gewonnen werden. Die Umsetzung dieser Erkenntnisse führte zu einer entscheidenden Verbesserung der Qualität des Tools. Bei der Sektion List-View wurden nicht die Issues der Heuristiken angezeigt, sondern nur die Guidelines. Einige EvaluatorInnen wünschten sich jedoch eine Gesamtübersicht über die Heuristiken und die dazu bereits festgehaltenen Probleme. Darüber hinaus wurde in der Druckansicht ein Formatierungsproblem aufgezeigt, welches dadurch entstanden war, dass die Tables verschachtelt dargestellt wurden. Diese unübersichtlich gegliederte Druckvorlage stellte für den/die ProjektadministratorIn einen potenziell vermeidbaren Mehraufwand dar. Als zusätzliches Problem erwies sich die Tatsache, dass bei den generierten Reports einzelne Heuristiken den dazugehörigen Issues teilweise nicht eindeutig zugeordnet werden konnten. Einige EvaluatorInnen bemängelten auch das Fehlen von verschiedenen HTML-Title Tags, welche dem/r UserIn bei mehreren geöffneten Fenstern einen Überblick erleichtern sollten. Alle diese Anmerkungen wurden als äußerst konstruktiv empfunden und umgehend in das Tool integriert. Durch diese Abänderungen wird die künftige Verwendung des Tools zweifelsfrei noch intuitiver ablaufen, was als entscheidende Verbesserung angesehen werden muss.

Andererseits wurden aber auch einige Mängel aufgezeigt, welche offenbar nur aufgrund einer zu strikten Auslegung der Projektdefinition oder unzureichende Erfahrungen seitens der EvaluatorInnen als problematisch empfunden wurden. So berichteten mehrere EvaluatorInnen, dass sie Usabilityprobleme gefunden hätten, die keiner der festgelegten Heuristiken zuzuordnen waren. Da das zu testende Objekt jedoch anhand der von dem/r ProjektadministratorIn vorgegebenen Heuristiken evaluiert werden sollte, ist die Dokumentation eines Problems das gegen keine dieser Richtlinien verstößt prinzipiell nicht sinnvoll. Werden dennoch Probleme aufgezeigt, die keiner Heuristik zugeordnet werden können, legt dies den Schluss nahe, dass das vorliegende Heuristik-Set nicht optimal definiert wurde.

Eine andere Anregung bestand darin, in der Navigationsleiste Funktionen wie beispielsweise den Backbutton zu unterbinden. Darin sahen die EvaluatorInnen die Möglichkeit, ein doppeltes Abspeichern von Daten zu verhindern, falls der/die UserIn beispielsweise durch das erneute Laden einer Seite einen HTTP-POST Event wiederholt. Dieser minimale Vorteil steht aus Sicht des Verfassers jedoch in keinerlei Verhältnis zu den beträchtlichen Einschränkungen der Benutzerfreiheiten, die eine solche Reduktion der Funktionen innerhalb der Navigationsleiste zur Folge hätten. Denn durch ein Ausblenden von Taskleisten könnten auch andere wichtige Funktionen, wie etwa das Drucken des Reports, nicht mehr über diese ausgeübt werden. Des Weiteren zweifelte ein Evaluator an der Anwendersicherheit, insbesondere im Hinblick auf die Passwortspeicherung. Diese Frage konnte jedoch damit beantwortet werden, dass alle Passwörter in der Datenbank als Hashes verschlüsselt abgespeichert werden, was einen sehr hohen Sicherheitsstandard gewährleistet. Insgesamt hatten die EvaluatorInnen einen positiven Eindruck von der Webapplikation. Sie erkannten sofort den Nutzen des Tools und lobten auch die Unterstützung der heuristischen Evaluierung. Insbesondere der Projektleiter war begeistert, da er durch die erleichterte Administration am meisten von dem Tool profitierte.

6. Zusammenfassung und Ausblick

Als Kernstück der vorliegenden Arbeit wurde ein Distributed Heuristic Evaluation Tool entwickelt, das entscheidende Vorteile auf dem Gebiet der Evaluierung von Softwareprodukten mit sich bringt. Ein wesentlicher Teil der Arbeit befasste sich daher mit der Entstehungsgeschichte und der detaillierten Beschreibung der Funktionsweise des Tools.

Bei der Entwicklung von Software kommt dem Usability Engineering und hier insbesondere dem User-Centered Design Ansatz zentrale Bedeutung zu. Da Usabilityinspektionen sich gegenüber anderen Inspektionsmethoden dadurch auszeichnen, dass sie schon in den frühen Phasen eines Softwareentwicklungsprozesses angewendet werden können und darüber hinaus ein schnelles und relativ kostengünstiges Feedback für die DesignerInnen darstellen, sind sie heute nahezu unersetzlich geworden. Daher wurde im Anschluss eine Übersicht über verschiedene Inspektionsmethoden erarbeitet, welche dazu dienen sollte die heuristische Evaluierung gegenüber anderen Methoden zu positionieren. Zudem wurden die Signifikanz und das Einsatzgebiet der verschiedenen Methoden detailliert erläutert. Hier wurde deutlich, dass die Heuristische Evaluierung eine der am häufigsten verwendeten Inspektionsmethoden ist, da sie von den frühen Stadien der Entwicklung bis hin zum fertigen Produkt angewendet werden kann. Diese Tatsachen bildeten die Basis der Entscheidung, ein Tool zur Unterstützung der heuristischen Evaluierung zu entwerfen, das eine noch effizientere Inspektion von Usabilityobjekten ermöglichen könnte. Kapitel 4 dieser Arbeit widmete sich daher zunächst einer Beschreibung der Konzeption dieses Distributed Heuristic Evaluation Tools. Nachdem erste Paper Prototypen in mehreren Sessions erstellt, und in einen grundlegenden Designentwurf zusammengefasst worden waren, erfolgte die Umsetzung des ursprünglichen Prototyps. Dafür wurden PHP als Entwicklungsumgebung und Mysql als Datenbank ausgewählt, da diese beiden Technologien durch eine GPL Lizenz frei verfügbar waren. Weitere Iterationen der Softwareentwicklung folgten, bis eine erste Version des voll funktionsfähigen Prototyps entstand. Diese wurde weiterentwickelt bis sie letztlich sämtliche Anforderungen erfüllte, die ursprünglich an das Tool gestellt worden waren. So erlaubt die finale Version ein unkompliziertes Anlegen und Administrieren von Evaluierungsprojekten, insbesondere im Bezug auf die individuelle Gestaltung von Heuristiken und Fragebögen. Das Tool ermöglicht es jedem/r berechtigten BenutzerIn, ein Projekt anzulegen und mit den notwendigen spezifischen Zugangsberechtigungen zu versehen. Dadurch können ExpertInnengruppen völlig unabhängig voneinander

Projekte evaluieren. Darüber hinaus gibt es für jede/n EvaluatorIn bei der Evaluierung eine Uploadfunktion, welche das Beifügen von Screenshots zur genaueren Erläuterung eines Problems ermöglicht. Der Fragebogen in diesem Tool dient vor allem einer Einschätzung des Hintergrunds und der Erfahrung des/r BenutzerIn. Am Ende jeder Evaluierung steht die Kumulationsphase, in der die gefundenen Usabilityprobleme aufgelistet werden und doppelte oder ähnliche Ergebnisse von dem/r ProjektadministratorIn in ein „Über-Problem“ zusammengefasst werden können. Zusätzlich werden auch alle hinzugefügten Screenshots bzw. Bilder übersichtlich dargestellt, um die gefundenen Probleme genauer darstellen zu können. Das Tool kann darüber hinaus nicht nur im Web zum Einsatz kommen, sondern auch als Intranetlösung Anwendung finden. Unabhängig vom Umfang des Anwendungsgebietes ermöglicht das Tool die zentralisierte Speicherung und erleichterte Auswertung von Daten, und unterstützt die ExpertInnen, welche die heuristische Evaluierung durchführen, darüber hinaus auch bei administrativen Aufgaben. Nach der Implementierung des Distributed Heuristic Evaluation Tools erfolgte dessen Evaluation durch mehrere StudentInnen der Technischen Universität Wien, welche in Kapitel 5 näher beschrieben wurde. Hierbei sollten die intuitive Bedienung und die Usability getestet, sowie erste Rückmeldungen der UsabilityexpertInnen eingeholt werden. Dieser Prozess stellte sich als äußerst aufschlussreich dar und die Implementierung der Ergebnisse konnte wesentlich zur Verbesserung der Qualität des Tools beitragen. Der Abschluss dieser weiteren Entwicklungsphase resultierte in der nun vorliegenden finalen Version des Tools. Wie deutlich erkennbar, entfallen bei der durch den Prototyp unterstützten heuristischen Evaluierung im Vergleich zur klassischen heuristischen Evaluierung zahlreiche administrative Schritte, wodurch eine wesentliche Zeitersparnis anfällt, die sich nicht zuletzt auch finanziell positiv auswirkt. Weiters lobten die ExpertInnen den guten Überblick über den Fortschritt des Usabilityprojekts, wie auch die hohe Flexibilität der Einsetzbarkeit.

Allgemein betrachtet ist aber festzuhalten, dass sich die IT-Unterstützung von Usabilitymethoden noch in der Anfangsphase befindet, wobei sie insbesondere hinsichtlich der Dokumentation von Problemen durchaus über großes Potential verfügt. Dass zur Inspektion von Software meist noch traditionelles Papier zum Einsatz kommt, erscheint paradox und machte eine IT-Unterstützung von Inspektionsmethoden geradezu überfällig. Im Rahmen dieser Arbeit wurde jedoch nur eine von mehreren sehr bekannten Usabilitymethoden mit Hilfe von Informationstechnologie unterstützt, während bei den anderen Methoden nach wie vor großer Entwicklungsbedarf an

effizienten Tools herrscht. Als Vision sollte es ein Softwarepaket geben, das verschiedene Inspektionsmethoden unterstützt, daher wäre es durchaus denkbar und wünschenswert, eine Entwicklung weiterer Tools voranzutreiben, um in ferner Zukunft vielleicht sogar eine Software mit einer Sammlung von Tools zur Durchführung verschiedener Inspektionsmethoden zu implementieren.

7. Abbildungsverzeichnis

Abbildung 1 – Anwendungsrahmen der Gebrauchstauglichkeit	12
Abbildung 2 – ISO13407 User Centered Design Model	14
Abbildung 3 – Überblick der Usability Methoden	18
Abbildung 4 – Cost/Benefit Model	28
Abbildung 5 – Task Performance Model	35
Abbildung 6 – Überblick der Formal Usability Inspection	36
Abbildung 7 - Aktivitätsdiagramm - Distributed Heuristic Evaluation Prozess	40
Abbildung 8 - Aktivitätsdiagramm – DHEP Vorbereitung	42
Abbildung 9 - Aktivitätsdiagramm – DHEP Evaluierung	43
Abbildung 10 - Aktivitätsdiagramm – DHEP Konsolidierung	44
Abbildung 11 - Aktivitätsdiagramm – DHEP Berichterstellung.....	45
Abbildung 12 – MVC Darstellung	47
Abbildung 13 – Erster Aufruf der Webseite / MVC	48
Abbildung 14 - Evaluationsübersicht	50
Abbildung 15 – Screenshot der Evaluationsansicht General Questions.....	50
Abbildung 16 - Screenshot der heuristischen Evaluationsansicht – List View	51
Abbildung 17 – Screenshot der heuristischen Evaluationsansicht – Single View	52
Abbildung 18 - Profil	52
Abbildung 19 - Projektübersicht.....	53
Abbildung 20 – Projekt anlegen.....	54
Abbildung 21 – Projekadminstration - Kumulationsphase	55
Abbildung 22 – Heuristic Sets bearbeiten	56
Abbildung 23 – General Questions Sets bearbeiten.....	57
Abbildung 24 – Projektadmin Usermanagement	58
Abbildung 25 – Superadmin Usermanagement.....	58
Abbildung 26 – Mozilla UZReview	60
Abbildung 27 – Question Pro.....	61

8. Literaturverzeichnis

Bias, R. (1991) Walkthroughs: Efficient collaborative testing. IEEE Software

Bias, R. (1994) The pluralistic usability walkthrough: Coordinated empathies. In Usability inspection methods, John Wiley & Sons Inc.

Bisant D. B. & Lyle J. R. (1989) A Two-Person Inspection Method to Improve Programming Productivity. IEEE Transactions on Software Engineering.

Dustdar, S., Gall, H. & Hauswirth, M. (2003) Software-Architekturen für Verteilte Systeme. Springer-Verlag

Fagan, M.E. (1976) Design and Code Inspections to Reduce Errors in Program Development. IBM System Journal.

International Organization for Standardization (1998) ISO 9241-11 Ergonomic Requirements for Office Work with Visual Display Terminals – Part 11: Guidance on Usability. <http://www.iso.org/>, 8.1.2007

International Organization for Standardization (1999) ISO 13407 Human-Centred Design Processes for Interactive Systems. <http://www.iso.org/>, 9.1.2007

Kahn, M. J. & Prail, A. (1994) Formal usability inspections In Nielsen, J. & Mack, Usability Inspection Methods, John Wiley & Sons

Kecher, C. (2006) Das umfassende Handbuch. Galileo Press

Knight, J. C. & Myers, E. A. (1993) An improved Inspection Technique. Communications of the ACM

Krasner, G. & Pope, S. (1988) MVC A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80. Journal of Objectoriented Programming (JOOP).

- Kunze, M. (1998) LAMP: Freeware Web Publishing System with Database Support. <http://www.heise.de/ct/english/98/12/230/>, 25.1.2007
- Martin, J. & Tsai, W. T. (1990) N-Fold Inspection: A Requirements Analysis Technique. Communications of the ACM
- Mayhew, D. & Bias R. (1994) Cost-Justifying Usability. Morgan Kaufmann. ISBN 0120958104
- Mayhew, D. (1999) The Usability Engineering Lifecycle. A Practitioner's Handbook for User Interface Design. Morgan Kaufmann. ISBN 1558605614
- Mozilla Foundation (2003) Uzilla Review. <http://uzilla.mozdev.org/> , 24.10.2006
- MYSQL AB (2007) Mysql: The world's most popular open source database. <http://www.mysql.org/>, 1.12.2006
- National Institute of Standards and Technology (1997) Industry Usability Reporting Project. <http://zing.ncsl.nist.gov/iusr/>, 12.2.2007
- Nielsen J. & Molich R. (1990) Heuristic evaluation of user interfaces. ACM CHI'90 Conference
- Nielsen, J. (1992) Finding usability problems through heuristic evaluation. ACM CHI
- Nielsen, J. (1994) Usability Engineering. Morgan Kaufmann. ISBN 0125184069
- Nielsen, J. (2005) Ten Usability Heuristics. http://www.useit.com/papers/heuristic/heuristic_list.html, 3.10.2006
- Nielsen, J. & Landauer, T. K. (1993) A mathematical model of the finding of usability problems. Proceedings ACM/IFIP INTERCHI'93 Conference
- Nielsen J. & Mack R. (1994) Usability Inspection Methods. John Wiley & Sons. ISBN 0471018775

Norman, D. (2002) The Design of Everyday Things. B&T. ISBN 0465067107

Parnas D.L. & Weiss D.M. (1985) Active design reviews: Principles and practices. 8th International Conference on Software Engineering.

PHP (2007) Hypertext Preprocessor. <http://www.php.net>, 1.2.2007

Polson, P.G., Lewis, C., Rieman, J., & Wharton, C. (1992) Cognitive walkthroughs: A method for theory-based evaluation of user interfaces. International Journal of Man-Machine Studies

Sears, A. (1997) Heuristic walkthroughs: Finding the problems without the noise. International Journal of Human-Computer Interaction

Smith & Mosier (1986) Guidelines for Designing User Interface Software. US Department of Commerce - National Technical Information Service. ISBN 9992080418

Snyder, C. (2003) Paper Prototyping Paper Prototyping – The fast and easy way to design and refine user interfaces. Morgan Kaufmann.

Survey Analytics Enterprise Survey Software (2006) QuestionPro Survey Software – <http://www.questionpro.com>, 12.11.2006

Wixon, D., Jones, S., Tse, L., and Casaday, G. (1994) Inspections and design reviews: Framework, history, and reflection. In Usability inspection methods, John Wiley & Sons Inc.

World Wide Web Consortium (2007). World Wide Web Consortium. <http://www.w3.org/>, 4.1.2007

Zakas, N. C. (2005) Professional JavaScript for Web Developers. Wiley & Sons.

Zuser W., Grechenig T. & Köhle M. (2004) Software Engineering mit UML und dem Unified Process, 2. Auflage, Verlag Pearson Studium

9. Anhang

9.1. Zehn Usability Heuristiken

Zehn auf Deutsch übersetzte Usability Heuristiken von Jakob Nielsen:

Sichtbarkeit des Systemstatus

Ein System sollte den Benutzer jederzeit angemessen darüber informieren, was passiert.

Übereinstimmung zwischen System und Wirklichkeit

Ein System sollte die Sprache des Benutzers sprechen, in einer Form, mit der der Benutzer vertraut ist, und nicht mit systemorientierten Begriffen. Informationen sollten in natürlicher und logischer Reihenfolge dargestellt werden.

Nutzerkontrolle und –freiheit

Benutzer wählen Systemfunktionen oft versehentlich und benötigen somit einen klar gekennzeichneten „Notausgang“, um den ungewünschten Zustand ohne langen Dialog verlassen zu können. Die Funktionen „Rückgängig“ und „Wiederholen“ sollten unterstützt werden.

Konsistenz und Standards

Benutzer sollten sich nicht fragen müssen, ob verschiedene Wörter, Situationen oder Aktionen dasselbe meinen. Plattformkonventionen sollten befolgt werden.

Fehlervorbeugung

Ein sorgfältiges Design, das das Auftreten von Problemen verhindert, ist noch besser als gute Fehlermeldungen.

Erkennen anstatt Erinnern

Objekte, Aktionen und Optionen sollten sichtbar sein. Der Benutzer sollte sich Informationen nicht von einem Abschnitt des Dialogs bis zu einem anderen merken müssen. Instruktionen für den Systemgebrauch sollten leicht auffindbar sein.

Flexibilität und Effizienz

Akzeleratoren („Programmzeitverkürzer“) können unbemerkt von neuen und unerfahrenen Benutzern die Interaktion für Expertennutzer beschleunigen, so dass ein System sowohl von unerfahrenen als auch erfahrenen Benutzern bedient werden kann. Benutzern sollte es möglich sein, häufige Aktionen auf sie zuzuschneiden.

Ästhetisches und minimalistisches Design

Dialoge sollten keine Informationen beinhalten, die irrelevant sind oder selten benötigt werden. Jede zusätzliche Informationseinheit in einem Dialog konkurriert mit den relevanten Informationen.

Benutzern sollte geholfen werden, Fehler zu erkennen, zu diagnostizieren, und sich von diesen wieder zu „erholen“

Fehlermeldungen sollten in einfacher Sprache formuliert sein, das Problem exakt beschreiben und eine konstruktive Lösung vorschlagen.

Hilfe und Dokumentation

Obwohl es besser ist, wenn ein System ohne Dokumentation verwendet werden kann, mag es notwendig sein, Hilfe und Dokumentation bereitzustellen. Diese Informationen sollten leicht zu durchsuchen und auf die Aufgabe des Benutzers fokussiert sein, zudem sollten sie konkrete Schritte, die vorgenommen werden müssen, aufzählen, und dabei nicht zu umfangreich sein.

9.2. Datenbankmodell

