

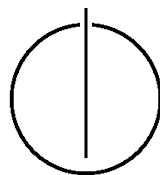
FAKULTÄT FÜR INFORMATIK

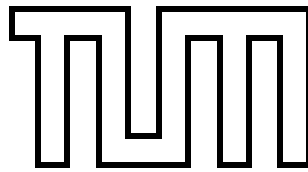
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelorarbeit in Wirtschaftsinformatik

**Identifikation von Bedienbarkeitsproblemen
in einem Enterprise 2.0 Werkzeug**

Stefan Wiesi





FAKULTÄT FÜR INFORMATIK

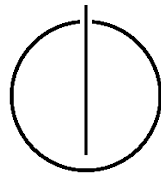
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelorarbeit in Wirtschaftsinformatik

Identifikation von Bedienbarkeitsproblemen in einem
Enterprise 2.0 Werkzeug

Identification of usability problems in an Enterprise 2.0
Tool

Bearbeiter:	Stefan Wiesi
Aufgabensteller:	Prof. Dr. Florian Matthes
Betreuer:	Alexander Steinhoff
Datum:	15. September 2011



Ich versichere, dass ich diese Bachelorarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 14. September 2011

Stefan Wiesi

Abstract

Während Web 2.0 in Unternehmen immer noch ein wichtiges Thema ist, nimmt auch der Konkurrenzkampf im Internet stetig zu. Aus diesem Grund versuchen Unternehmen, sich nicht ausschließlich durch eine höhere Anzahl an zur Verfügung stehenden Funktionen ihrer Software oder Services von anderen Unternehmen abzugrenzen, sondern achten auch immer mehr darauf, eine gute und intuitive Bedienbarkeit ihres Produkts sicherzustellen. Ziel dieser Arbeit ist es, Bedienbarkeitsprobleme eines Enterprise 2.0 Werkzeugs zu identifizieren. Dabei soll außerdem gezeigt werden, ob es sinnvoll ist, verschiedene Methoden miteinander zu kombinieren und welchen Mehrwert die Anwendung mehrerer Methoden mit sich bringt.

Dafür werden zunächst verschiedene Usability-Evaluationsmethoden genannt und bezüglich ihrer Charakteristiken, Vorgehensweisen sowie ihrer Vor- und Nachteile erläutert. Nachdem passende Evaluationsverfahren für das zu untersuchende System ausgewählt und anschließend angewendet wurden, konnte eine Vielzahl an Usability-Problemen aufgedeckt werden. Eine umfassende Betrachtung der gefundenen Probleme ergab dabei, dass die Durchführung mehrerer Methoden die Anzahl gefundener Probleme erheblich steigern kann. Verschiedene Methoden bringen außerdem unterschiedliche Arten von Problemen zum Vorschein. Der Output einer Methode kann zudem als Input eines anderen Verfahrens genutzt und somit Ressourcen gespart werden. Es ist daher trotz des Mehraufwands zu empfehlen, unterschiedliche Usability-Evaluationsmethoden miteinander zu kombinieren.

Inhaltsverzeichnis

Abstract	vii
Abbildungsverzeichnis	xi
Tabellenverzeichnis	xiii
1. Einleitung	1
1.1. Motivation	1
1.2. Aufbau der Arbeit	2
2. Usability	3
2.1. Definition und Abgrenzung	3
2.2. Kriterien für Usability	5
2.2.1. Inhalt	5
2.2.2. Design	5
2.2.3. Struktur	6
2.3. Usability-Probleme	6
3. Usability Evaluation	9
3.1. Expertenorientierte Evaluationsmethoden	9
3.1.1. Heuristische Evaluation	10
3.1.2. Cognitive Walkthrough	12
3.1.3. Pluralistic Walkthrough	13
3.2. Nutzerorientierte Evaluationsmethoden	15
3.2.1. Interviews	15
3.2.2. Fokusgruppen	16
3.2.3. Fragebögen	18
3.2.4. Logfile-Analyse	19
3.2.5. Usability-Tests	21
4. Planung und Durchführung ausgewählter Methoden	23
4.1. Das System	23
4.2. Auswahl der Methoden	26
4.3. Logfile-Analyse	27
4.3.1. Quick-Search	27
4.3.2. Präfix-Suche	28
4.4. Usability-Interviews	29
4.5. Usability-Testing	30

5. Auswertung	33
5.1. Registrierungs- und Anmeldeprozess	33
5.2. Wikis	35
5.3. Wiki-Seiten	37
5.4. Tags	41
5.5. Attribute	43
5.6. Attachments	45
5.7. Suche	48
5.7.1. Quick-Search	48
5.7.2. Präfix-Suche	52
6. Globale Betrachtung der gefundenen Usability-Probleme	57
7. Zusammenfassung und Ausblick	63
Appendix	67
A. Fragenkatalog Usability-Interviews	67
A.1. Offene Fragen	67
A.2. Geschlossene Fragen	67
Literaturverzeichnis	69

Abbildungsverzeichnis

2.1. User Experience - Usability (nach [SB06, S. 22])	4
3.1. Prozentualer Anteil gefundener Fehler mit steigender Anzahl Gutachtern (siehe [NM90])	11
4.1. TypeTags und Attribute in Tricia	24
4.2. Tricia Suchfunktion	25
4.3. Tricia Navigation	25
4.4. Piwik-Auswertung	27
4.5. Beispiel für die aufgerufene URL einer Präfix-Suche	28
5.1. Login und Registrierungsmöglichkeit	33
5.2. Tricia Registrierungs-Maske	34
5.3. Google Login - siehe http://www.google.com	34
5.4. Expert Settings bei der Wiki-Erstellung	35
5.5. Vergleich der Einstellungsmöglichkeiten während und nach der Wiki-Er- stellung	36
5.6. Erstellung einer neuen Wiki-Seite	37
5.7. Minimierte und maximierte Menü-Leiste	38
5.8. Erstellung einer Tabelle	38
5.9. Tabelleneigenschaften Darstellungsfehler	39
5.10. Änderung von Bildeigenschaften	40
5.11. Vergabe von Tags	41
5.12. Löschen von Tags	42
5.13. Hinzufügen von TypeTags	42
5.14. Hinzufügen von Attributen	43
5.15. Editieren von Attributen	44
5.16. Fehler beim Löschen eines Wertepaares	44
5.17. Hinzufügen eines Links	45
5.18. Aufrufen des Attachment-Verzeichnisses	46
5.19. Änderung der Navigation	46
5.20. Aufruf der zu löschenden Datei	47
5.21. Intranet-Suchfunktion	48
5.22. Überblick Piwik-Analyse	49
5.23. Piwik - Auswahl des ersten Vorschlags	50
5.24. Piwik - Auswahl des zweiten Vorschlags	51
5.25. Suche von Tags	52
5.26. Darstellung bereits ausgewählter Tags aus der Tag-Cloud	55

6.1. Anzahl gefundener Probleme je Testperson	57
6.2. Anzahl gefundener Probleme während des Testverlaufs	58

Tabellenverzeichnis

2.1. Usability-Kriterien	6
5.1. Usability-Probleme bei Registrierung und Login	34
5.2. Usability-Probleme bezüglich der Erstellung und Verwaltung von Wikis	37
5.3. Usability-Probleme beim Erstellen und Editieren von Wiki-Seiten	40
5.4. Usability-Probleme bei der Verwaltung von Tags	43
5.5. Usability-Probleme bei der Verwaltung von Attributen	45
5.6. Usability-Probleme bei der Verwaltung von Anhängen	47
5.7. Parameter-Kombination	53
5.8. Kombination von drei oder mehr Parametern	54
5.9. Usability-Probleme bei der Nutzung der Suchfunktion	55
6.1. Zusammenfassung der Usability-Probleme	61

1. Einleitung

1.1. Motivation

In Zeiten des Web 2.0 steigt die Anzahl im Internet angebotener Software und Services rasant und unaufhaltsam an. Firmen haben es somit immer schwerer, ihre eigenen Produkte zu vermarkten sowie sich von der Konkurrenz abzuheben. Der Fokus liegt hierbei jedoch nicht nur darauf, neue Kunden zu gewinnen, sondern auch bereits akquirierte Kunden weiterhin an das Produkt zu binden. Ein aktuelles Beispiel stellen die beiden sozialen Netzwerke Google+ und Facebook dar. Google+, welches sich noch in der Beta-Phase befindet, konnte die ersten Testnutzer mit einer einfachen und schnellen Handhabung begeistern. Kein anderes soziales Netzwerk war bisher in der Lage, so schnell eine Zahl von 10 Millionen Nutzern zu erreichen. Dies liegt zum einen natürlich an dem großen Bekanntheitsgrad von Google. Zum anderen konnte Google+ an einigen Stellen überzeugen, mit denen Facebook bereits lange Zeit zu kämpfen hatte. Darunter fallen zum Beispiel die Unzuverlässigkeit einiger Funktionen wie der Chat-Funktion oder die nur sehr schwer zu bedienenden Privatsphäre-Einstellungen. Google+ konnte hingegen genau bei diesen Aspekten punkten. Es wurde dem Nutzer beispielsweise besonders erleichtert, bestimmte Inhalte auch nur mit bestimmten Personengruppen zu teilen. Die dazu nötigen Schritte waren für alle Nutzergruppen einfach zu verstehen und durchzuführen. Nur wenige Wochen nach dem Start von Google+ veröffentlichte Facebook einige Neuerungen, die besonders auch auf eine nutzerfreundlichere Bedienung der Status-Funktion abzielten und somit versuchten, deren Usability zu verbessern.

Auch bei technischen Produkten gehört Usability zu den wichtigsten Kaufargumenten. Das wurde in letzter Zeit vor allem bei der Vermarktung von Smartphones deutlich, deren hohe „Nutzerfreundlichkeit“ von den Herstellern immer wieder angepriesen wurde. Produkthersteller aller Industriezweige bewegen sich dabei immer weiter zu einer mensch-zentrierten Gestaltung hin. Hinzukommend verpflichten beispielsweise 2008 veröffentlichte DIN-Normen die Hersteller von Medizinprodukten, deren Gebrauchstauglichkeit zu steigern, indem ein Usability-Engineering-Prozess durchgeführt und Ergebnisse in einem sogenannten „Usability-File“ dokumentiert werden müssen. Auch die seit 2005 rapide angestiegene Anzahl an Stellenausschreibungen für Usability-Professionals ist ein Indiz dafür, dass Usability eine immer größere Rolle spielt [SB06, Seite 17f]. Aus diesen Gründen ist abzuleiten, dass Unternehmen die Usability ihrer Produkte nicht nur verbessern sollten, sondern auch müssen. Dies gilt besonders auch für die Hersteller von Software und Websites.

In dieser Arbeit sollen Usability-Probleme eines Enterprise 2.0 Werkzeugs identifiziert werden. Hierfür bietet sich eine Vielzahl von Methoden an. Weischedel und Huizingh sagen, dass Firmen oft auf die Analyse von Logfiles und Clickstream-Daten zurückgreifen, um die Usability ihrer Website oder Software zu verbessern. Die zur Verfügung stehenden großen Datenmengen führen jedoch in vielen Fällen dazu, dass sich die Firmen aus-

schließlich auf diese Evaluationsmethode beschränken [WH06]. Aus diesem Grund soll in dieser Arbeit die Kombination verschiedener Methoden genauer untersucht werden. Dadurch kann anschließend auch gezeigt werden, wie sich die unterschiedlichen Methoden unterstützen und welche Vor- und Nachteile die Durchführung mehrerer Methoden mit sich bringt.

1.2. Aufbau der Arbeit

Die Arbeit erstreckt sich auf zwei Teile. Im ersten Teil wird eine Einführung zum Thema Usability gegeben. Dabei wird der Begriff sowohl definiert, als auch zum Beispiel von der User Experience abgegrenzt. Des Weiteren werden verschiedene Kriterien für Usability genannt und erläutert. Nachdem Usability-Probleme definiert wurden, werden in Kapitel 3 sowohl analytische, als auch empirische Evaluationsverfahren näher beschrieben. Der Fokus liegt hierbei vor allem auf deren typischen Charakteristiken, der Durchführung und den Vor- und Nachteilen der einzelnen Methoden.

Das zu evaluierende System wird zu Beginn des zweiten Teils skizziert. Anschließend werden drei Methoden aus den im ersten Teil genannten Evaluationsverfahren anhand ihrer Eignung ausgewählt. Die Planung und Vorbereitung der ausgewählten Methoden werden in Kapitel 4 behandelt. Es folgt eine Beschreibung der identifizierten Usability-Probleme, die in Kapitel 6 in einem globalen Kontext näher betrachtet werden. Abschließend werden die Ergebnisse der Arbeit zusammengefasst und Ansätze für weitere Analysen und Arbeiten vorgestellt.

2. Usability

Im ersten Kapitel der Arbeit soll der Begriff Usability zunächst definiert und abgegrenzt werden. Dabei wird er auch in Verbindung mit der Nutzererfahrung (User Experience) gebracht. Anschließend soll herausgearbeitet werden, was man unter Usability-Problemen versteht und mit welchen Evaluationsmethoden die Bedienbarkeit eines Systems untersucht und Fehler identifiziert werden können.

2.1. Definition und Abgrenzung

Wenn man sich mit Usability beschäftigt, kommt man immer auch mit dem Begriff der **Ergonomie** in Berührung. Ergonomie ist eine *„wissenschaftliche Disziplin, die sich mit dem Verständnis der Wechselwirkungen zwischen menschlichen und anderen Elementen eines Systems befasst und der Berufszweig, der Theorie, Prinzipien, Daten und Methoden auf die Gestaltung von Arbeitssystemen anwendet mit dem Ziel, das Wohlbefinden des Menschen und die Leistung des Gesamtsystems zu optimieren.“* [ISO04] Diese Definition stellt folglich nicht das technische System in den Mittelpunkt, sondern das Gesamtsystem, welches aus Mensch, Werkzeug, Aufgabe und Umgebung besteht [SB06].

Da in der Vergangenheit die Multifunktionalität eines Systems das größte Kriterium für dessen Nutzung war, trat bald auch der Begriff der „Benutzerfreundlichkeit“ immer häufiger in den Vordergrund. Dieser wurde schließlich jedoch durch die „Gebrauchstauglichkeit“ abgelöst. Ein gebrauchstaugliches System ist nicht nur komfortabel zu bedienen, sondern unterstützt den Nutzer auch bei der Erreichung seiner Ziele. Im Gegensatz zur Ergonomie ist die Usability keine eigenständige Disziplin, sondern eine *„Qualität eines technischen Systems“* und ein *„Ziel der Gestaltung nach den Erkenntnissen der Ergonomie“* [SB06, S. 19].

Während es für Usability eine ganze Reihe an möglichen Übersetzungen wie beispielsweise Gebrauchstauglichkeit, Benutzbarkeit, Benutzerfreundlichkeit, Qualität oder auch Nützlichkeit gibt, versucht die ISO-Norm 9241 eine allgemein gültige Definition zu liefern: *„Usability bezeichnet das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und mit Zufriedenheit zu erreichen.“* Hinzukommend besagt die Norm, dass die Maße für die drei Komponenten Effektivität, Effizienz und Zufriedenheit nicht allgemein festgelegt werden können, da diese vor allem vom Nutzungskontext und dem *„Zweck, für den die Gebrauchstauglichkeit zu beschreiben ist“*, abhängen [ISO99]. Betrachtet man nun die drei Komponenten im Kontext eines Users, der zum Beispiel ein bestimmtes Buch kaufen möchte, dann beschreibt die Effektivität das Erreichen des Ziels an sich. Der Aufwand, den der User betreiben muss, um das Ziel zu erreichen, wird hier außen vor gelassen. Dieser Aufwand spielt bei der Komponente Effizienz jedoch eine sehr große Rolle. Die Effizienz beschreibt den Einsatz, den der User aufbringen muss, um sein Ziel zu erreichen. Je

2. Usability

geringer der Einsatz, desto höher ist auch die Effizienz. Gibt der User hier beispielsweise den Titel des gesuchten Buches in die Suchmaske ein und erhält nach sehr geringer Ladezeit seine Ergebnisse, kann man von einer hohen Effizienz sprechen. Schließlich spielt die Zufriedenheit des Nutzers noch eine wichtige Rolle. Diese ist jedoch schwer zu bestimmen, da verschiedene Nutzer auch verschiedene Erwartungen haben. Erwartet ein Nutzer beispielsweise sein gesuchtes Buch schon in den ersten 5 Treffern und findet sein Buch auf der zweiten Seite an neunter Position, dann ist er sehr wahrscheinlich unzufrieden. Hat ein User die Erwartung, dass sein Buch innerhalb der ersten zehn Treffer liegen sollte, dann wäre er in diesem Beispiel bereits zufrieden. Möchte man hohe Zufriedenheit erreichen, müssen die Erwartungen der Nutzer also bereits im Voraus bekannt sein. [BG02, S. 2f]

Im Gegensatz zur Usability beschäftigt sich die **User Experience (UX)** nicht nur ausschließlich mit der tatsächlichen Nutzung eines Systems, sondern betrachtet sowohl dessen angenommene als auch verarbeitete Nutzung (siehe Abb. 2.1). Damit beinhaltet die User Experience auch positive sowie negative Gefühle der Nutzer, aber auch deren Vorlieben und Meinungen. Gerade deshalb kann sie zum Beispiel auch durch Werbung, die Medien oder die Verpackung eines Produkts beeinflusst werden. Eine gute Usability ist somit zwar ein wichtiger Faktor für eine positive User Experience, ist aber nur einer von mehreren Faktoren, die diese beeinflussen. Abschließend kann man sagen, dass die User Experience für Produkte, welche für einen Endverbraucher produziert werden, eine weitaus größere Rolle spielt. Arbeitssysteme werden hingegen öfter nach ihrer Usability evaluiert [SB06, S. 22].

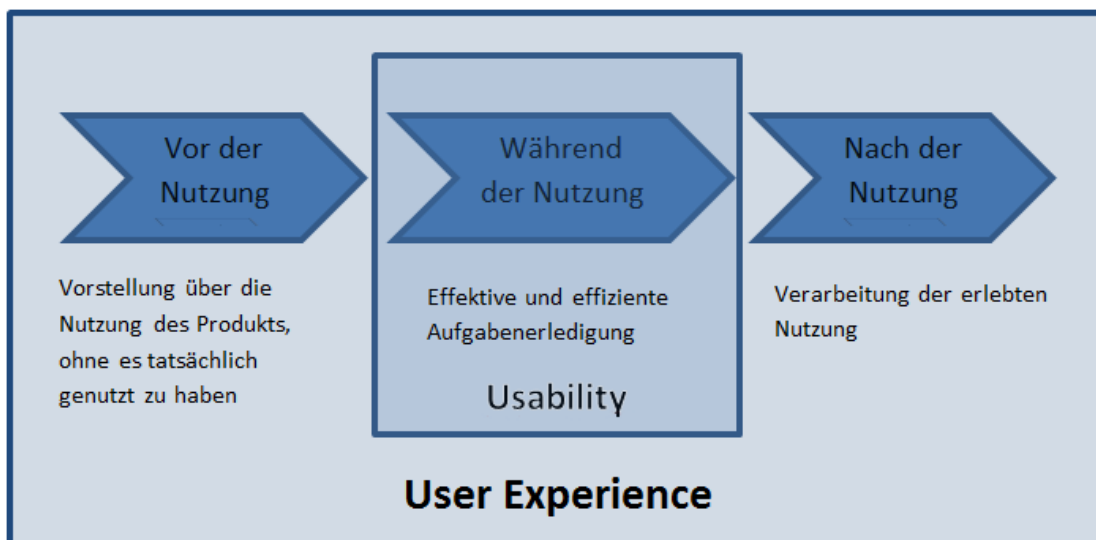


Abbildung 2.1.: User Experience - Usability (nach [SB06, S. 22])

2.2. Kriterien für Usability

Wie bereits erwähnt, tragen vor allem die drei Zielgrößen Effektivität, Effizienz und Zufriedenheit zu einer guten Usability bei. In diesem Abschnitt der Arbeit sollen diese drei Komponenten in Anlehnung an [BG02] in Beziehung mit den Dimensionen einer Website gebracht werden. Unter einer Website versteht man eine komplexe Struktur von verschiedenen Seiten, welche beliebig miteinander kombiniert werden können und oft mit unterschiedlichen Inhalten befüllt sind. Anhand dessen können die drei Dimensionen einer Website identifiziert werden: Inhalt, Design und Struktur.

Die Dimension **Inhalt** beschreibt alle Inhalte einer Seite und deren Darstellung. Das **Design** (auch Web-Design) stellt die visuelle Gestaltung dar. Die **Struktur** steht für die Anordnung der einzelnen Seiten. Während sich also die beiden Dimensionen Inhalt und Design auf einzelne Seiten einer Website beziehen, beschreibt die Struktur deren Zusammenhänge. Im Folgenden sollen nun die Dimensionen mit den Zielgrößen in Beziehung gesetzt werden, um somit passende Kriterien für die Usability einer Website ableiten zu können [BG02, S. 5].

2.2.1. Inhalt

Wie bereits geschrieben, sagt Effektivität aus, dass der User sein Ziel überhaupt erreicht. Bezogen auf den Inhalt einer Website bedeutet das, dass die gesuchte Information auf der Website auch vorhanden ist. Ist dies der Fall, hat der Inhalt einer Website eine hohe Effizienz, wenn die bereitgestellten Informationen vom Nutzer mit möglichst wenig Ressourceneinsatz gescannt und gefiltert werden können. Dies kann beispielsweise durch die Aufteilung der Informationen in verschiedene Blöcke und Spalten oder durch eine abwechslungsreiche Präsentation der Inhalte umgesetzt werden. Außerdem tragen kurze Ladezeiten zu einer hohen Effizienz bei. Eine hohe Anzahl an Bildern kann die Ladezeit der Website zwar um einiges verlängern, Bilder können jedoch dabei helfen, Informationen übersichtlicher zu gestalten. Hier gilt es somit zielgruppenspezifisch abzuwägen. Ein User ist zufrieden, sobald ihm nur noch die Informationen angeboten werden, die er auch benötigt. Online-Shops wie Amazon erreichen dies zum Beispiel, indem sie dem Nutzer hauptsächlich Inhalte anbieten, die auf seine hinterlegten Interessen und bereits gekauften Produkte zurückzuführen sind [BG02, S. 6f].

2.2.2. Design

Das Design beschäftigt sich in diesem Fall nicht mit der traditionell bekannten Aufbereitung von Inhalten, sondern lediglich mit der visuellen Gestaltung der Bildschirmoberfläche. Das Design einer Website ist bereits effektiv, wenn eine gestaltete Website vorliegt. Durch unterschiedliche Gestaltung einzelner Funktionen können diese voneinander getrennt werden und liefern somit optische Anhaltspunkte für den User. Eine Website ist effizient, sobald der User durch diese Art von Gestaltung fähig ist, eine Hierarchie der Module zu erkennen und diesen bestimmte Inhalte zuzuordnen kann. Der Designer kann dies zum Beispiel erreichen, wenn er die Primär- und Sekundärnavigation am oberen beziehungsweise unteren linken Rand der Seite anlegt. Dieses Prinzip hat der Großteil der User bereits erlernt und erkennt es somit schnell wieder. Die Anwendung bewährter Prinzi-

pien führt jedoch auch dazu, dass sich viele Websites visuell nur noch gering voneinander unterscheiden und somit ein Herausstechen aus der Menge erschwert wird. Eine Differenzierung kann jedoch beispielsweise durch unerwartete und ungewöhnliche Darstellungen, welche den User fesseln und somit die Zufriedenheit erhöhen, erreicht werden. Die Wahl der Darstellungsmethoden muss erneut nutzerspezifisch gewählt werden [BG02, S. 8f].

2.2.3. Struktur

Als letzte Dimension einer Website wird in diesem Abschnitt die Struktur auf die drei Komponenten überprüft. Viele einzelne Seiten, die in Verbindung zueinander stehen, bauen eine Website auf. Eine Website ist bereits effektiv, wenn Verbindungen zwischen den einzelnen Seiten vorhanden sind. Diese Verbindungen können einerseits durch eine vorhandene Navigation oder andererseits durch Links zu weiteren Inhalten der Website realisiert werden. Das Navigieren von Seite zu Seite wird als „Flow“ bezeichnet. Ein effizienter Flow zeichnet sich dadurch aus, dass der Nutzer ihn auch erkennt. Der User muss stets wissen, an welcher Stelle der Hierarchie er sich befindet. Außerdem müssen Rückschritte zu der vorherigen Seite ermöglicht werden. Die Zufriedenheit des Users kann zum Beispiel durch das Anbieten verschiedener Orientierungswege erreicht werden [BG02, S. 9].

Fasst man die beschriebenen Beispiele zusammen, ergibt sich folgende Tabelle (siehe [BG02, S. 11]):

	Inhalt	Design	Struktur
Effektivität	gesuchter Inhalt ist vorhanden	Modularität	Vorhandensein eines Navigationsflows
Effizienz	Scannbarkeit, unterschiedliche Präsentation des Inhalts	Vorhandensein einer optischen Hierarchie, Verwendung von Gestaltungsprinzipien	Eindeutigkeit in der Navigation
Zufriedenheit	Individualisierung der Inhalte	Progressivität	verschiedene Orientierungswege

Tabelle 2.1.: Usability-Kriterien

Insgesamt kann man sagen, dass es sich hierbei jedoch nur um Beispiele für Usability-Kriterien handelt und diese nicht auf alle Zielgruppen gleichermaßen angewendet werden können. Die Erwartungen der User sollten somit so früh wie möglich bekannt sein. Die Anpassung der Website an die Zielgruppe kann folglich sowohl vor dem Erstellen der Website, als auch während des Erstellungsprozesses geschehen [BG02, S. 11f].

2.3. Usability-Probleme

Bevor die Usability einer Website anhand von verschiedenen Verfahren evaluiert und somit Usability-Probleme aufgedeckt werden können, muss zunächst definiert werden, was genau man unter einem Usability-Problem versteht. Nielsen beschreibt Usability-Probleme beispielsweise als „Aspekte einer Benutzeroberfläche, die den vermeintlichen Grund für

eine verminderte Usability des betroffenen Systems für den Endnutzer darstellen“ [NM94, Seite 3]. Karat, Campell und Fliegel definieren ein Usability-Problem als „alles, was mit der Fähigkeit des Nutzers interferiert, seine Aufgaben effizient und effektiv zu komplettieren“ [KCF92]. Wie man sehen kann, vernachlässigen beide Definitionen mögliche begrenzte Fähigkeiten der potenziellen Nutzer eines Systems. Nutzer, die keine Domänenexperten sind, werden früher oder später auf scheinbare Usability-Probleme stoßen, wenn das System ausschließlich für die Nutzung durch Domänenexperten konzipiert wurde. Hinzu kommt auch der Anwendungskontext vernachlässigt. Versucht der Nutzer zum Beispiel eine Aufgabe mit dem System zu lösen, für dessen Bewerkstelligung das System nicht erstellt wurde, wird er in seinen Augen bald mit Usability-Problemen zu kämpfen haben [SB06, S. 25f].

Diesen beiden Problemen versuchen Sarodnick und Brau mit ihrer Definition eines Usability-Problems entgegenzuwirken:

„Ein Usability-Problem liegt vor, wenn Aspekte eines Systems es Nutzern mit hinreichender Domänenerfahrung unangenehm, ineffizient, beschwerlich oder unmöglich machen, in einem typischen Anwendungskontext die Ziele zu erreichen, für deren Erreichung das System erstellt wurde.“ [SB06, Seite 26]

Es gibt sowohl verdeckte als auch offene Usability-Probleme. Während man unter verdeckten Problemen Fehler versteht, die nur selten oder mit hohem Aufwand gefunden werden, da man auf sie oft nur auf Umwegen oder zufällig trifft, sind offene Probleme leicht zu entdecken. Es kann jedoch nicht pauschal gesagt werden, dass schwerwiegende Probleme meist offensichtlich und somit einfach zu finden sind und verdeckte Probleme oft nur minder schwer sind [SB06, S. 26].

3. Usability Evaluation

Die Evaluation der Usability stellt eines der zentralen Elemente des Usability-Engineerings dar und sollte vor allem durchgeführt werden, weil es für die Erstellung von Software oder Websites keine allgemeingültigen Lösungen gibt. Unter Usability-Engineering versteht man den „*methodischen Weg zur Erzeugung der Eigenschaft Usability*“ [SB06, Seite 23]. Es unterstützt zum Beispiel das Software-Engineering bei der Entwicklung technischer Systeme und ergänzt es um arbeitswissenschaftliche Aspekte. Einer Usability-Evaluation sollten immer auch Analyse- und Entwurfsphasen vorangehen, um somit passende Entwurfsentscheidungen treffen zu können. Wie in Kapitel 2 bereits beschrieben, müssen dazu die Bedürfnisse und Erwartungen der Nutzer möglichst frühzeitig bestimmt und die durchzuführenden Aufgaben des Systems abgegrenzt und beschränkt werden. Auf diesen Erkenntnissen aufbauend können dann später Usability-Ziele definiert und anschließend geprüft werden. Immer noch wird jedoch in einer Vielzahl von Projekten auf eine frühzeitige Durchführung von Aktivitäten des Usability-Engineerings verzichtet und die Usability erst am Ende des Prozesses evaluiert. Dabei gefundene Probleme lassen sich dann aber meist nur noch mit einem hohen finanziellen und zeitlichen Aufwand beheben [SB06, S. 23ff].

Für die Durchführung der Usability-Evaluation stehen viele verschiedene Methoden zur Verfügung. Diese lassen sich im allgemeinen in zwei Klassen aufteilen: die empirischen und analytischen Methoden. Die empirischen Methoden bestehen zum Großteil aus Bewertungs-Methoden, die die eigentlichen Nutzer des System zur Gewinnung von Erkenntnissen nutzen. In den analytischen Methoden versuchen Usability-Experten sich in die Lage der Nutzer zu versetzen oder beurteilen das System anhand von Richtlinien, Prinzipien, eigenem Wissen und Erfahrung [SB06, S. 119f].

In diesem Kapitel der Arbeit sollen sowohl einige empirische Methoden, also nutzerorientierte Verfahren, als auch analytische Verfahren, auch expertenorientierte Methoden genannt, beschrieben und erläutert werden.

3.1. Expertenorientierte Evaluationsmethoden

Nielsen bezeichnet expertenorientierte Evaluationsmethoden als „Usability-Inspektionen“ [NM94]. Dabei werden Bedienoberflächen von Gutachtern auf ihre Usability geprüft. Gutachter können in diesem Fall sowohl Usability-Experten, Software-Entwicklungs-Berater mit bestimmten Spezialgebieten und Endnutzer mit vorhandenem Domänenwissen, aber auch Fachleute aus anderen Bereichen sein. Die einzelnen Usability-Inspektionsmethoden verfolgen manchmal zwar geringfügig unterschiedliche Ziele, wobei das Finden von Usability-Problemen jedoch für alle das Hauptziel darstellt. Je nach Methode können die gefundenen Probleme verschieden gewichtet und anschließend bereits Verbesserungsvorschläge geäußert werden [NM94]. Dieser Abschnitt beschäftigt sich mit folgenden drei

analytischen Usability-Evaluationsmethoden: Heuristische Evaluation, Cognitive Walkthrough und Pluralistic Walkthrough.

3.1.1. Heuristische Evaluation

Bei der heuristischen Evaluation versuchen Usability-Experten, Probleme zu identifizieren, indem sie unter Einbeziehung allgemein gültiger Usability-Regeln, sogenannter Heuristiken, das System auf die Einhaltung dieser Heuristiken untersuchen. Die erste Liste genereller Heuristiken wurde von Nielsen und Molich (siehe [NM90]) veröffentlicht und umfasst neun Heuristiken. Diese Liste wurde jedoch mehrmals revidiert und überarbeitet und beinhaltet aktuell die folgenden zehn Heuristiken (nach [SB06, S. 147f]):

1. **Sichtbarkeit des Systemstatus:** Der Nutzer sollte vom System durch geeignetes Feedback in angemessener Zeit immer darüber informiert werden, was er gerade macht und wo er sich befindet.
2. **Übereinstimmung zwischen System und realer Welt:** Das System sollte die Sprache des Benutzers sprechen. Dafür soll es keine systemorientierten Begriffe, sondern Wörter und Formulierungen, die dem Nutzer geläufig sind, nutzen.
3. **Benutzerkontrolle und Freiheit:** Es muss immer ein „Notausgang“ für den Nutzer vorhanden sein, den er ohne große Probleme betätigen kann. Dafür sollten auch die Funktionen „Undo“ und „Redo“ unterstützt werden.
4. **Konsistenz und Standards:** Plattform-Konventionen müssen eingehalten werden. Den Nutzern muss klar sein, welche Begriffe die gleiche Bedeutung haben und welche Aktionen dasselbe Ziel verfolgen.
5. **Fehler vermeiden:** Das System soll so ausgelegt sein, dass Fehler bereits im Vorhinein vermieden werden.
6. **Erkennen vor Erinnern:** Objekte, Aktionen und Optionen sollten für den Nutzer immer sichtbar sein. Dadurch bleibt ihm erspart, sich bereits gesehene Informationen oder Abläufe zu merken.
7. **Flexibilität und effiziente Nutzung:** Es sollte Nutzern ermöglicht werden, von ihnen häufig durchgeführte Aktionen auf ihre Bedürfnisse anpassen zu können. Um das System für Anfänger und Experten besser bedienbar zu machen, kann das System zum Beispiel Beschleunigungsmöglichkeiten bereitstellen, die nur der erfahrene Nutzer erkennt und somit die eigene Interaktionsgeschwindigkeit steigern kann.
8. **Ästhetisches und minimalistisches Design:** Irrelevante Informationen sollten möglichst vermieden werden, da hierdurch die Sichtbarkeit von relevanten Informationen gefährdet sein kann.
9. **Unterstützung beim Erkennen, Verstehen und Bearbeiten von Fehlern:** Bereitgestellte Fehlermeldungen müssen für den Benutzer auch verständlich sein. Das Problem sollte deshalb in der Sprache des Nutzers beschrieben und mögliche Lösungsvorschläge aufgezeigt werden.

10. **Hilfe und Dokumentation:** Jedem Nutzer sollte der Zugang zu einer Dokumentation und Hilfe möglich sein. Die Hilfe sollte dabei leicht zu durchsuchen sein und die für den Nutzer relevanten Aufgaben und konkreten Schritte beinhalten.

Die Experten nehmen bei der Durchführung der heuristischen Evaluation die Rolle eines späteren Nutzers ein. Wenn einer der Experten eine Verletzung der oben genannten Heuristiken entdeckt, hat er meist einen Hinweis auf ein Usability-Problem gefunden. Die Anzahl der einzusetzenden Experten kann dabei stark variieren. Nielsen und Molich haben unter anderem vier verschiedene Benutzeroberflächen mit Hilfe der heuristischen Evaluation von Experten bewerten lassen. Bei den bewerteten Systemen handelte es sich beispielsweise um ein Videotex (siehe <http://en.wikipedia.org/wiki/Videotex>) und zwei Sprachdialog-Systeme. Die eingesetzten Experten hatten zudem verschiedene Hintergründe. Für das Videotex- und die beiden Sprachdialog-Systeme wurden Informatikstudenten eingesetzt, die eine Vorlesung über Usability hörten. Das vierte System, das bei Kunden einer Telefongesellschaft zum Einsatz kommen sollte, um Namen und Adressen zu bereits vorhandenen Telefonnummern finden zu können, wurde von Lesern der dänischen Zeitschrift „Computerworld“ begutachtet. Als Basis für die Evaluationen dienten sowohl zweimal auf Papier vorhandene Mock-ups des Systems, als auch bereits laufende Systeme [NM90].

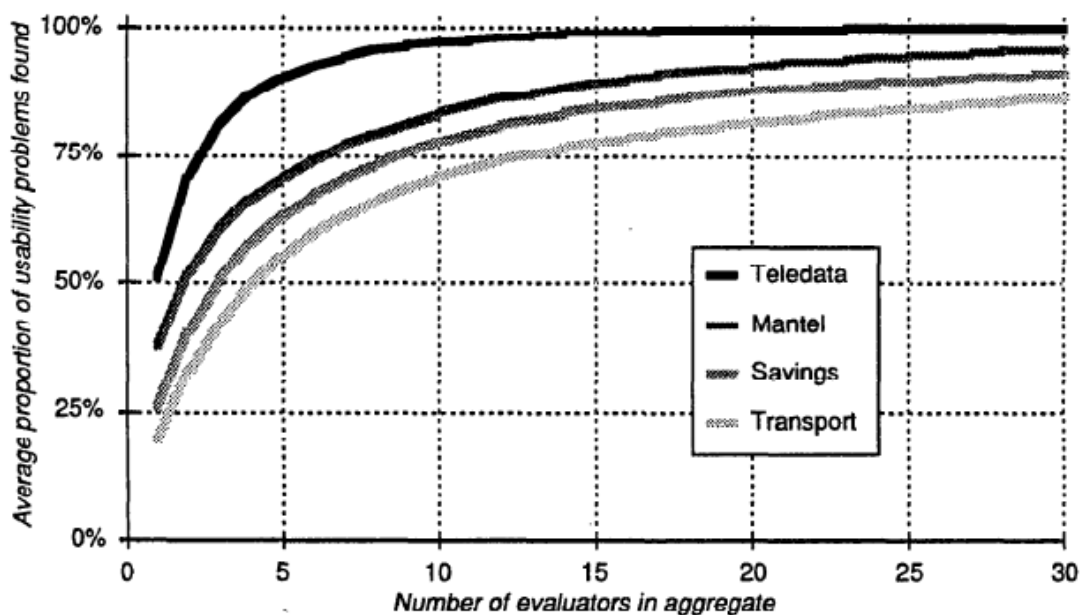


Abbildung 3.1.: Prozentualer Anteil gefundener Fehler mit steigender Anzahl Gutachtern (siehe [NM90])

Wie anhand Abbildung 3.1 zu sehen ist, steigt die Anzahl der gefundenen Fehler zwischen 1 und 5 Gutachtern rapide an und erreicht ab 10 Evaluatoren bereits fast ihr Maximum. Nielsen und Molich empfehlen daher bei der Durchführung einer heuristischen Evaluation zwischen 3 und 5 Experten zu wählen und falls noch zeitliche oder finanzielle Ressourcen vorhanden sind, diese in andere Methoden zu investieren.

Vor- und Nachteile

Zu den Vorteilen der heuristischen Evaluation gehört vor allem, dass die Methode günstig und intuitiv durchzuführen ist. Es ist zudem sehr einfach, Gutachter für diese Form der Bewertung zu motivieren. Die Durchführung der Methode erfordert wenig bis keinen Planungsaufwand und kann außerdem bereits früh im Entwicklungsprozess eingesetzt werden.

Einer der Nachteile der Methode ist zum Beispiel das Fehlen von Lösungsvorschlägen zu gefundenen Usability-Problemen. Die Ergebnisse können außerdem durch die verschiedenen Denkweisen der Experten verfälscht werden [NM90].

3.1.2. Cognitive Walkthrough

Der Cognitive Walkthrough (CWT) ist eine von mehreren Usability-Walkthrough-Methoden, um Benutzeroberflächen zu bewerten und Usability-Probleme zu finden. Weitere Vertreter der Walkthrough-Verfahren sind z.B. noch der Pluralistic Walkthrough oder der Soziotechnische Walkthrough. Der Pluralistic Walkthrough wird in einem folgenden Kapitel näher erläutert. Die verschiedenen Walkthrough-Methoden werden zwar immer nach dem gleichen Vorgehensmodell durchgeführt, machen dabei jedoch von unterschiedlichen Fragestellungen Gebrauch. Dabei werden einer Gruppe von Zielnutzern oder Experten zum Beispiel schriftliche Beschreibungen der Funktionen eines Systems oder Mock-Ups vorgelegt und diese mit Hilfe zuvor festgelegter Kriterien bewertet. Die Cognitive Walkthrough Methode beschäftigt sich grundsätzlich mit der Erlernbarkeit des Systems und zielt somit vor allem darauf ab, Probleme zu identifizieren, auf die unerfahrene Nutzer in ihrer Kennenlernphase der Software oder Oberfläche stoßen. Dafür werden für später durchzuführende Aufgaben vorab Ideallösungen entwickelt. Anschließend versuchen Experten die geforderten Aufgaben zu lösen, wobei auch darauf geachtet wird, die Fähigkeiten und Kenntnisse der späteren Zielgruppe zu berücksichtigen [SB06, S. 152].

Vorgehen

Der Cognitive Walkthrough Prozess lässt sich in folgende fünf Teilphasen gliedern (nach [WRLP93]):

1. **Definition der Input-Größen:** In dieser Phase des CWT soll zunächst geklärt werden, wer die späteren Nutzer des Systems sind. Dies kann sowohl eine sehr allgemeine Darstellung sein (z.B. „Microsoft User“), als auch eine detailliertere Beschreibung (z.B. „Mac-User über 50, die das Programm XY regelmäßig nutzen“). Des Weiteren müssen die von den Experten zu bearbeitenden Aufgaben definiert und passende Ideallösungen entwickelt werden. Hinzukommend müssen alle möglichen Reaktionen der Benutzeroberfläche bekannt sein und entweder auf Papier oder in Form eines bereits implementierten Systems zur Verfügung stehen.
2. **Bestimmung der Gutachter:** Die Gutachter, die die Bedienoberfläche evaluieren sollen, werden in der zweiten Phase des CWT bestimmt. Dies können unter anderem Design-Spezialisten, Software-Entwickler, Usability-Experten, aber auch Exper-

ten mit beispielsweise Marketing-Hintergrund sein. Der CWT kann sowohl individuell, als auch durch die Gruppe durchgeführt werden. Auch der Entwickler des zu testenden Systems kann dabei als Experte eingesetzt werden.

3. **Durchführung der Aktivitäten jeder einzelnen Aufgabe:** Im dritten Teil des CWT versuchen die Experten, die vorgeschriebenen Aufgaben zu lösen. Dabei gehen sie besonders auf folgende Fragen ein: Versucht der Nutzer, den richtigen Effekt zu erzielen? Erkennt der Nutzer, ob die korrekte Handlung zur Verfügung steht? Erkennt der Nutzer, ob die korrekte Handlung zum gewünschten Effekt führt? Erkennt der Nutzer, dass er die richtige Handlung getätigt hat?
4. **Erfassen der kritischen Informationen:** Kritische Informationen, wie beispielsweise gefundene Probleme oder Verbesserungsvorschläge, sollten so genau wie möglich dokumentiert werden. Dies kann zum Beispiel durch eine Videoaufzeichnung des CWT erreicht werden. Bei Gruppendiskussionen ist es zudem wichtig, kritische Informationen auf einem für alle Teilnehmer sichtbaren Medium festzuhalten (beispielsweise Flipchart).
5. **Revision des Systems um gefundene Probleme zu beheben:** Am Ende des CWT sollte das gesamte System nochmal revidiert und über gefundene Fehler diskutiert werden. Dabei sollte darauf geachtet werden, dass es oft einfacher ist, einzelne Aktionen aus dem Lösungsprozess zu eliminieren, als Fehler explizit zu beheben. Oft ist es auch einfacher, ein gefundenes Problem durch eine globale Umstrukturierung des Prozesses zu lösen, als es lokal zu beheben.

Vor- und Nachteile

Wie die heuristische Evaluation ist auch der CWT einfach, intuitiv und mit geringem Kostenaufwand durchzuführen. Darüber hinaus kann durch ihn der Lernprozess von späteren Nutzern bewertet werden. Hinzukommend kann der CWT auch sehr früh in der Entwicklungsphase zum Einsatz kommen, wenn beispielsweise noch kein Prototyp vorhanden ist.

Diese Art von Usability-Test wird oft jedoch von Experten und nicht von eigentlichen Nutzern durchgeführt. Diese versuchen zwar, die Fähigkeiten und den Wissenstand späterer Nutzer anzunehmen. Die eigene Erfahrung kann dabei aber nie ganz ausgeblendet werden. Auch wenn die eigentliche Durchführung der Methode nicht sehr aufwendig ist, müssen im Vorfeld sehr detaillierte Planungen getätigt werden. Die Planung der Aufgaben und vor allem die Entwicklung der Lösungen erfordern somit hohe zeitliche Ressourcen.

3.1.3. Pluralistic Walkthrough

Wie der CWT gehört auch der Pluralistic Usability Walkthrough (PUW) zu den Usability-Walkthrough-Methoden. Ihn zeichnet besonders aus, dass sowohl Usability-Experten und Produkt- oder Software-Entwickler, als auch repräsentative Nutzer des Systems bei der Durchführung des Evaluierungsprozesses mitwirken. Den PUW charakterisieren folgende fünf Eigenschaften (nach [HN07]):

3. Usability Evaluation

1. Miteinbeziehung von repräsentativen Nutzern, Produkt-Entwicklern und „Human Factor“ Experten.
2. Die Benutzeroberflächen des Systems werden in der gleichen Reihenfolge vorgestellt, wie später auch den tatsächlichen Nutzern.
3. Alle Teilnehmer versuchen, die Rolle der Nutzer anzunehmen.
4. Die Teilnehmer schreiben für jede Bildschirmdarstellung auf, welche Aktionen sie zur Erreichung der (Teil-)Ziele durchführen würden, bevor die Gruppe die einzelnen Darstellungen anschließend gemeinsam diskutiert.
5. Bei der Diskussion der Darstellungen kommen die repräsentativen Nutzer immer als erstes zu Wort.

Wie aus den typischen Eigenschaften des PUW hervorgeht, werden für diese Methode keine realen Prototypen oder fertigen Implementierungen eines Systems gebraucht. Es reicht bereits aus, Bilder des fiktiven Systems bereitstellen zu können, beispielsweise in der Form von Mock-Ups. Diese werden den Teilnehmern des PUW jeweils einzeln und in einer **festen** Reihenfolge vorgelegt. Die Teilnehmer markieren dann auf den einzelnen Ausdrucken, welche Aktionen sie zur Erreichung eines Ziels ausführen würden. Einzelne Handlungen müssen dabei Schritt für Schritt dokumentiert werden. Nachdem alle Evaluatoren alle Systembilder abgearbeitet haben, stellt der Leiter des Versuchs die zuvor erarbeitete Ideallösung der Aufgaben vor. Anschließend folgt eine Gruppendiskussion, bei der die individuellen Ergebnisse und Eindrücke jedes einzelnen Teilnehmers besprochen werden. Fragen über das System können während des Prozesses meist ausschließlich von den Produkt-Entwicklern beantwortet werden, da zu diesem Zeitpunkt der Entwicklungsphase oft noch keine Dokumentation vorhanden ist. Die Usability-Experten versuchen vor allem zwischen den Nutzern und Entwicklern zu vermitteln [SB06, S. 158f], [HN07].

Vor- und Nachteile

Einer der Hauptvorteile des PUW wurde bereits mehrfach genannt. Die Usability-Evaluation kann schon in einer sehr frühen Phase des Entwicklungsprozesses durchgeführt werden, auch wenn noch kein Prototyp des späteren Systems vorliegt. Der PUW kann jedoch natürlich auch genutzt werden, wenn bereits ein Prototyp vorhanden ist. Ein weiterer Vorteil der Methode ist, dass mögliche Lösungsvorschläge von Seiten der Usability-Experten, Produkt-Entwickler oder Nutzer sofort diskutiert und evaluiert werden können. Das sofortige Vorhandensein von Lösungen hat zudem den Vorteil, dass Teilnehmer gefundene Usability-Probleme überhaupt offenlegen. Dies hat den einfachen Grund, dass Menschen allgemein ungern kritisieren, ohne einen passenden Lösungsvorschlag parat zu haben. Hinzukommend können Fragen der Nutzer Hinweise auf die später noch zu erstellende Dokumentation des Systems geben und die Produkt-Entwickler in einem frühen Stadium des Projekts direktes Feedback von potentiellen Nutzern des Systems bekommen. Auch zu erwähnen sind der geringe Kosten- und Planungsaufwand.

Als Nachteil ist ganz klar zu sehen, dass obwohl der allgemeine Zeitaufwand für die Durchführung des PUW eher gering ist, der eigentliche Prozess die Zeit in Anspruch

nimmt, die der langsamste Teilnehmer zur Bearbeitung der Aufgaben benötigt. Des Weiteren können nicht alle möglichen Aktionen auf einem Systembild simuliert werden. Das einfache Durchforsten des Systems durch den Teilnehmer, was oft auch zu gewissen Lerneffekten führen kann, ist somit nicht möglich [NM94, Seite 69f].

3.2. Nutzerorientierte Evaluationsmethoden

Nach Holzinger ([Hol05]) sind Usability Evaluierungen, die mit Hilfe von Endnutzern eines Systems oder Produkts durchgeführt werden, die „grundlegendsten“ Testmethoden und meist „unabdingbar“. In diesem Kapitel werden Usability-Interviews, Fokusgruppen, Fragebögen, Logfile-Analysen und Usability-Tests genauer erläutert.

3.2.1. Interviews

Unter einem Interview versteht man ein geleitetes Gespräch zwischen einem Interviewer und einem Nutzer. Dabei steht vor allem im Fokus, eine spezifische Nutzergruppe eines Systems besser kennen zu lernen. Auf diesem Weg kann der Interviewleiter herausfinden, wie die Funktionen eines Systems von den Nutzern wahrgenommen, interpretiert und genutzt werden und kann somit Verbesserungspotential aufdecken. Das Interview kann sowohl als autonome Evaluierungsmethode durchgeführt, als auch zusammen mit anderen Methoden eingesetzt werden [Usa11].

Vorgehen

Die Durchführung eines Interviews basiert auf einer gründlichen Vorbereitungsphase. Man kann folgende Prozessschritte ableiten: [Usa11]

1. **Bestimmung der Durchführungsart:** Zunächst muss festgelegt werden, ob das Interview face-to-face oder telefonisch durchgeführt werden soll. Dabei sollte man jedoch ein direktes Interview dem telefonischen aus später genannten Gründen vorziehen. Bestehen jedoch große geographische Distanzen zwischen den Interviewpartnern, so kann es natürlich von Vorteil sein, das Interview telefonisch durchzuführen.
2. **Festlegung der Form:** In diesem Prozessschritt geht es darum, einen Interview-Leitfaden zu entwickeln. Zunächst muss bestimmt werden, ob das Interview offen oder strukturiert aufgebaut sein soll. Dementsprechend können Fragen offen oder geschlossen formuliert sein. Der Interview-Leitfaden sollte alle Themen und Fragen beinhalten und somit als Orientierungshilfe für den Interviewleiter dienen. Ein Interview sollte in der Regel mindestens 30, maximal 90 Minuten dauern.
3. **Rekrutierung der Nutzer:** Wenn notwendig, müssen Nutzer zunächst in verschiedene Nutzertypen gegliedert werden. Die Anzahl der durchzuführenden Interviews kann variieren. Für jeden Nutzertyp sollte man zwischen 6-10 Interviews führen. Mainkar empfiehlt zum Beispiel zwischen 5-7 Interviews [Mai11].
4. **Durchführung eines Pretests:** Der in Schritt 2 erstellte Leitfaden sollte zunächst an 1-2 Nutzern getestet werden. Dabei können Stärken und Schwächen des Leitfadens

entdeckt und gegebenenfalls angepasst werden. Außerdem kann festgestellt werden, ob das Interview den geplanten Zeitrahmen einhält.

5. **Durchführung des Interviews** (ergänzend nach [Mai11]): Bei der Durchführung des Interviews sollte man darauf achten, dass man ungestört bleibt. Des Weiteren sollte man zu Beginn den Grund und Zweck der Befragung erläutern. Die Beziehung zwischen Leiter und Interviewpartner hat großen Einfluss auf den Erfolg. Der Leiter muss vor allem versuchen, Vertrauen aufzubauen, neutral zu sein, den Nutzer **immer** mit Respekt zu behandeln und **nie** zu widersprechen. Zu Beginn sollten außerdem Fragen gestellt werden, die den Nutzer in seinen Antworten nicht zu sehr einschränken und möglichst einfach zu beantworten sind. Hinzukommend hat der Interviewer die Aufgabe, die Ergebnisse so detailliert wie möglich festzuhalten. Dafür eignet sich besonders, das Gespräch als Audio oder Video aufzuzeichnen. Notizen während des Interviews zu machen ist eine weitere Möglichkeit, Resultate festzuhalten, stellt jedoch für den Interviewer eine zusätzliche Belastung dar.

Vor- und Nachteile

Interviews lassen sich in allen Phasen des Entwicklungsprozesses anwenden. Vor allem bei Produktinnovationen kann somit frühes und sofortiges Feedback von Zielgruppen-Nutzern gesammelt werden. Durch die direkte Kommunikation mit den Nutzern können sowohl detailliertere Fragen besprochen, als auch aufkommende Unklarheit schnell beseitigt werden. Dies stellt besonders einen Vorteil gegenüber der Durchführung einer Umfrage mit Hilfe eines Fragebogens dar [Usa11].

Da im Gegensatz zu Gruppendiskussionen individuelle Interviews mit jeder Person einzeln durchgeführt werden müssen, sind sie besonders zeit- und kostenintensiv. Auch die Planung des Interviews, sowie die Rekrutierung der Partner ist sehr zeitaufwendig. Ein weiterer großer Nachteil ist vor allem, dass der Erfolg der Methode von den Fähigkeiten des Interviewers abhängt. Ist dieser beispielsweise unerfahren oder lenkt das Interview nicht in die richtige Richtung, können eventuell die gewünschten Ergebnisse nicht erzielt werden [Mai11].

3.2.2. Fokusgruppen

Kontio et al. beschreiben Fokusgruppen als Usability-Evaluationsmethode, die aus den offenen Interviews entstanden ist [KLB04]. Dabei werden qualitative Daten nicht bei einem individuellen Interview, sondern bei einer Gruppendiskussion erhoben. Die Gruppendiskussion, an der üblicherweise zwischen 3 bis 12 Teilnehmer mitwirken, wird von einem Moderator durchgeführt. Dieser sollte seine eigene Meinung möglichst nicht in die Diskussion einbringen und hat vor allem die Aufgabe, dass die vorher festgelegte Struktur der Diskussion eingehalten wird, sicherzustellen, dass die Probanden nicht zu weit vom eigentlichen Thema abweichen und, wenn nötig, tiefer nachzuhaken.

Kontio et al. gliedern außerdem den Durchführungsprozess in folgende Schritte [KLB04]:

- **Definition des Forschungsproblems:** Im ersten Schritt muss zunächst die Problemstellung, welche der Evaluation zugrunde liegt, festgelegt werden. Während Fokusgruppen sich sehr gut eignen, um zum Beispiel sofortiges Feedback für ein neues

Konzept zu erhalten, dabei Helfen einen Fragebogen zu erstellen, oder mögliche Usability-Probleme aufzudecken, sind sie weniger geeignet um beispielsweise Hypothesen zu testen oder quantitative Aussagen über ein Thema zu machen.

- **Planung des eigentlichen Events:** Die eigentliche Durchführung der Gruppendiskussion, für deren Ablauf ein strukturierter Plan erstellt wird, dauert in der Regel zwischen 2 und 3 Stunden. Die zu diskutierenden Themen müssen vorab definiert werden. Dabei muss die Zeit so bemessen werden, dass jedes Thema von den Teilnehmern verstanden werden kann und eine interaktive Diskussion möglich ist.
- **Auswahl der Teilnehmer:** Der Wert der gewonnenen Daten hängt vor allem von der Erfahrung und dem Wissen der Teilnehmer ab. Deshalb ist die Auswahl von erfahrenen und motivierten Teilnehmern ein kritischer Faktor für den Erfolg der Methode. Dabei hängt vor allem von der Forschungsfrage ab, ob die Teilnehmer bereits Erfahrung in Gruppendiskussionen haben müssen.
- **Leiten der Sitzung:** Am Anfang der Sitzung sollte der Moderator eine kurze Beschreibung der Gründe und Ziele der Diskussion geben. Dabei wird jedes Thema normalerweise nacheinander beschrieben. Der zuvor festgelegte Zeitplan sollte möglichst eingehalten, alle Themen abgehandelt und Feedback zu ihnen gegeben werden. Die Art der Diskussion kann zudem verschiedene Formen annehmen. Neben einer strukturierten Diskussion können auch Werkzeuge wie Brainstorming, Abstimmungen oder auch Rollenspiele angewendet werden. Langford und McDonagh [LM03] beschreiben in ihrem Buch zum Beispiel 38 verschiedene Methoden, welche in einer Gruppendiskussion angewendet werden können. Die Ergebnisse der Diskussion können beispielsweise auf Papier, Audio oder Video festgehalten werden.

Vor- und Nachteile

Fokusgruppen haben den Vorteil, dass sie eine große Menge an verwertbaren Daten von einer Gruppe von Befragten sehr schnell und nur mit geringem Kostenaufwand zur Verfügung stellen können. Würde man die festgelegten Themen beispielsweise in individuellen Interviews besprechen, würde dies ein Vielfaches an Zeit und somit Kosten in Anspruch nehmen. Der Moderator hat zudem die Möglichkeit, direkt mit den Teilnehmern zu sprechen. Somit können Unklarheiten sofort beseitigt oder entstandene Folgefragen gestellt werden. Während die Teilnehmer ihre Fragen und Antworten in eigenen Worten verfassen können, kann der Moderator auch deren Mimik und Gestik in die Evaluation einfließen lassen. Die Gruppendynamik führt außerdem dazu, dass Teilnehmer Dinge äußern, die sie in einem individuellen Interview nicht geäußert hätten. Hinzukommend sind Fokusgruppen sehr flexibel einzusetzen und die gewonnenen Daten sehr einfach zu verstehen [SSR07, Seite 42f].

Die bei einer Gruppendiskussion entstehende Dynamik kann jedoch nicht nur positiv sein, sondern auch die Kontrolle der Diskussion für den Moderator erschweren. Diesem Effekt kann jedoch durch verschiedene Techniken entgegengewirkt werden. Die soziale Akzeptanz der Teilnehmer untereinander könnte auch zu Problemen führen. Vertritt ein Teilnehmer zum Beispiel eine Meinung, die durch den Großteil der Gruppe abgelehnt

wird, kann dies dazu führen, dass der Teilnehmer von nun an weniger Informationen beisteuert. Die Zeitbeschränkung kann unter anderem dazu führen, dass komplexere Themen nicht von allen Teilnehmern verstanden werden. Deshalb sollten, wenn möglich, Teilnehmer mit ähnlichem Wissensstand ausgewählt werden [KLB04].

3.2.3. Fragebögen

Fragebögen werden häufig dazu benutzt, quantitative Aussagen von Nutzern zu gewinnen. Ein Fragebogen besteht meist aus einem Satz von Fragen und/oder Aussagen (Items). Die Fragen können zudem verschiedenen Unterthemen zugeteilt werden. Im Bereich der Usability Forschung werden Fragebögen den Teilnehmern häufig nach der Beendigung eines Usability-Tests vorgelegt, da der Beantwortung der Fragen eine hinreichende Auseinandersetzung mit dem System zugrunde liegen sollte. Bei der Bearbeitung des Fragebogens spielt der subjektive Einfluss, der auf einen Menschen wirkt, eine große Rolle. Die Teilnehmer des Tests müssen zur Beantwortung der Fragen das System mit einem ihnen zuvor bekannten System vergleichen. Dieser Vergleich wirkt sich folglich immens auf die Antworten der Nutzer aus. Je größer jedoch die Anzahl der Teilnehmer ist, desto kleiner wirkt sich der subjektive Einfluss auf das Gesamtergebnis aus. Die Antwortmöglichkeiten können entweder in Form von Multiple Choice, Ratingskalen oder Freitextfeldern vorliegen. Während die Wissenschaft in den letzten Jahren versucht hat, standardisierte Fragebögen bezüglich der Usability eines Systems zu entwickeln, können diese Fragebögen auch für eigene Zwecke angepasst und genutzt werden [SB06, S. 181f].

Bei der Erstellung eigener Fragebögen ist nach Sarodnick und Brau ([SB06, S. 183]) aber besonders auf die Einhaltung der folgenden Hauptgütekriterien zu achten.

- **Objektivität:** Die durch den Fragebogen gesammelten Ergebnisse sollten möglichst unabhängig vom Versuchsdurchführer sein.
- **Reliabilität:** Bei einer wiederholten Durchführung sollten annähernd ähnliche Ergebnisse festgestellt werden.
- **Validität:** Der Fragebogen sollte genau das messen, was er auch messen soll.

Fragebögen können sowohl in Papierform, als auch digital vorliegen und von den Befragten entweder selbst, durch einen Mittelsmann beispielsweise in einem Interview, oder auch am Computer bei zum Beispiel einer Online-Umfrage ausgefüllt werden [BG02, S. 79]. Aus diesem Grund ergeben sich für Fragebögen verschiedener Art auch unterschiedliche Vor- und Nachteile.

Vor- und Nachteile

Sarodnick und Brau beschreiben sowohl allgemeine, als auch auf computergestützte Fragebögen beschränkte Vor- und Nachteile [SB06, S. 184f].

Ein allgemeiner Vorteil von Fragebögen ist vor allem die geringe Beeinflussbarkeit durch Dritte. Aus diesem Grund haben Fragebögen meist eine hohe Objektivität. Außerdem ist durch die Gliederung in Unterthemen eine Trennung von Teilsystemen oder Funktionen möglich. Fragebögen können zudem in jeder Phase des Entwicklungsprozesses eingesetzt

werden. Im Bereich der Usability-Evaluation gibt es bereits eine Vielzahl an standardisierten Fragebögen. Computergestützte Fragebögen haben außerdem den Vorteil, dass sie sehr kostengünstig sind, da keine Materialkosten für deren Durchführung anfallen. Da die gesammelten Daten bereits in elektronischer Form vorliegen und auch während der Eingabe auf ein korrektes Format geprüft werden, sind Online-Fragebögen weniger fehleranfällig. Durch die Online-Bereitstellung eines Fragebogens können auch Ergebnisse mit sehr großer Stichprobenzahl erhoben werden. Hinzukommend fühlen sich Befragte im Internet oft anonym und beantworten so Fragen, auf die sie normalerweise nicht geantwortet hätten.

Den vielen Vorteilen stehen auch Nachteile gegenüber. Ein fest strukturierter Fragebogen schränkt den Nutzer oft zu sehr ein und bietet deshalb nur wenig Platz für freie Antworten. Die Entwicklung von standardisierten Fragebögen ist sehr aufwändig und somit kostenintensiv. Außerdem ist die Qualität der Ergebnisse sehr von der Qualität des Erhebungsinstruments abhängig. Computergestützte Fragebögen haben zudem den Nachteil, dass häufig Personen mit einer hohen Computere Expertise stärker vertreten sind, als bei Fragebögen, die auf Papier beantwortet werden. Die gewünschte Zielpopulation ist oft nur schwer abzugrenzen. Teilnehmer haben außerdem keine Möglichkeit, Verständnisfragen zu stellen, da sie bei einer Online-Umfrage ausschließlich auf sich allein gestellt sind. Durch die Unverbindlichkeit im Internet hat man als Versuchsleiter oft mit einer hohen Abbrecherquote bei Online-Fragebogen zu kämpfen.

3.2.4. Logfile-Analyse

Das Logfile eines Webservers ist „ein einfaches Textfile, das mit jedem Editor oder einfach mit dem Browser geöffnet werden kann“ ([Hei03, S. 101]. Der Webserver arbeitet im Grunde genommen wie ein Programm, das die von Browser-Nutzern an den Server gerichtete Anfragen beantwortet. Dabei werden die angefragten Dateien - meist in der Form von Dokumenten - an den Browser zurückgeliefert. Jede dieser Anfragen wird vom Webserver in einem Logfile festgehalten, indem er pro Anfrage eine Zeile in das Logfile schreibt. Auf diese Weise werden auch Anfragen, die nicht erfolgreich waren oder auf einen Fehler liefen, geloggt. Die Größe eines Logfiles hängt folglich von der Anzahl der Besucher der Website ab und kann, obwohl nur Zeichen im Logfile enthalten sind, zwischen 10 MByte und mehreren Gigabyte liegen. [Hei03]

Aufbau

Nach Heindl ([Hei03, S. 102]) und der Website e-teaching.org ([ete11]) enthält ein Logfile-Eintrag meist folgende Basis-Informationen:

- **IP oder DNS-Adresse des Clients:** Anhand der IP-Adresse können Anfragen einzelnen Nutzern zugeordnet werden. Heutzutage kann oft bereits an der IP-Adresse des Nutzers seine geographische Position bestimmt werden.
- **Identität des Clientrechners und Benutzers:** Während die Identität des Rechners meist nicht verfügbar ist, enthält das Logfile bei vorheriger Anmeldung des Nutzers dessen Benutzernamen.

- **Zeitpunkt der Anfrage:** Das Logfile enthält den sekundengenauen Zeitpunkt der Anfrage - gemäß Serverzeit.
- **Kommando:** Der Server speichert, ob eine GET oder POST Methode aufgerufen wurde.
- **Serverantwort:** Der Eintrag enthält außerdem den Antwortcode des Servers auf die Anfrage.
- **Dateiname und Pfadangabe:** Es werden der genaue Dateiname der angeforderten Datei und deren Speicherort festgehalten.
- **Dateigröße:** Dateigröße der angefragten Datei.

Außerdem werden sowohl die URL, die der Nutzer unmittelbar vor der Anfrage aufgerufen hat, als auch der Browser und das benutzte Betriebssystem des Nutzers als zusätzliche Informationen einer Anfrage gespeichert.

Auswertung

Die oben beschriebenen Informationen können auf unterschiedliche Art und Weise ausgewertet werden. Dafür steht vor allem eine Vielzahl an Softwarepaketen zur Verfügung, die entweder auf dem eigenen PC, dem Webserver, oder einem externen Dienstleister installiert werden können [Hei03].

Die Auswertung kann zum Beispiel zeigen, wie sich der gesamte Traffic einer Seite über einen bestimmten Zeitraum entwickelt. Außerdem kann die Anzahl der Aufrufe auf eine Website oder einen Inhalt bestimmt werden. Dadurch können Bereiche identifiziert werden, die Besucher besonders interessieren. Das Speichern von Sessions und vor allem deren Dauer kann Rückschlüsse auf die Intensität der Nutzung der Inhalte geben. Es ist auch möglich, den zurückgelegten Weg eines Nutzers zu verfolgen. Die Untersuchung solcher Klickpfade kann beispielsweise für die Verbesserung der Navigation genutzt werden. Hinzukommend können durch die Auswertung von Logfiles Inhalte gefunden werden, die von den Nutzern nicht genutzt werden oder sogar zum Verlassen der Seite führen. Werden Hinweise auf Probleme gefunden, können diese dann in weiteren Usability-Evaluationsmethoden genauer untersucht und anschließend behoben werden [ete11].

Vor- bzw. Nachteile

Einer der Hauptvorteile dieser Methode ist, dass die Besucher einer Seite nicht durch einen zusätzlichen Aufwand bei der Erhebung der Daten mitwirken müssen. Die Anfragen an einen Webserver werden in jedem Fall im Logfile gespeichert. Die Auswertung und Verwendung dieser ohnehin vorhandenen Daten ist ein positiver Nebeneffekt. Außerdem kann die Analyse von Logfiles eine gute Grundlage für weitere Untersuchungen sein.

Viele Browser nutzen den Cache-Speicher, um oft besuchte Seiten zwischenspeichern. Deshalb müssen eigentliche Anfragen oft nicht mehr an den Server gestellt werden, sondern können direkt bearbeitet und Seiten damit schneller dargestellt werden. Dies kann

unter Umständen zu einer Verfälschung der eigentlichen Daten führen. Auch die gemeinsame Nutzung von Rechnern kann zu ungenauen Daten führen, da unterschiedliche Benutzer oft nicht unterschieden werden können. Des Weiteren beeinflussen Webcrawler die Erhebung der Daten immens. Diese gilt es deshalb so gut wie möglich zu filtern. Auch Zugriffe von Entwicklern, die ihre Änderungen testen werden mitgeloggt und können so ein falsches Bild über die Nutzung bestimmter Inhalte liefern.

3.2.5. Usability-Tests

Der Usability-Test zählt wahrscheinlich zu den bekanntesten Usability-Evaluationsmethoden. In ihm wird ein System, das sich meist noch im Entwicklungsstadium befindet, von Nutzern getestet, indem sie zuvor definierte Aufgaben bearbeiten. Deshalb wird der Usability-Test oft auch als „Nutzertest“ oder „Benutzbarkeitstest“ bezeichnet [SB06, S. 163]. Um mögliche Stärken und Schwächen des Systems aufzudecken, werden die Nutzer bei der Bearbeitung der Aufgaben von Usability-Experten beobachtet. Dabei können auch beiläufige Äußerungen der Nutzer oder die benötigte Zeit Hinweise auf mögliche Verbesserungen geben.

Usability-Tests lassen sich in induktive und deduktive Tests unterteilen. Während induktive Tests der formativen Evaluation dienen und Usability-Probleme durch das Testen von frühen Versionen oder Prototypen des Systems entdeckt werden sollen, werden bei deduktiven Tests verschiedene Alternativen eines Systems miteinander verglichen oder die Leistungsfähigkeit eines einzelnen Systems ermittelt. Je nach Art des Tests können auch die Rahmenbedingungen variieren. Induktive Tests werden beispielsweise oft direkt am Arbeitsplatz des Nutzers durchgeführt. So können auch Probleme identifiziert werden, die durch das Arbeitsumfeld beeinflusst werden. Im Gegensatz dazu werden deduktive Tests meist im Labor durchgeführt, da für die Ausführung von Vergleichstests standardisierte Bedingungen von Nöten sind. Bei induktiven Tests müssen außerdem alle Funktionen des Systems implementiert sein. Dies ist vor allem für eine erleichterte Auswertung notwendig, da sonst vermeintliche Usability-Probleme von Fehlern, die aus realitätsfremden Bedingungen hervorgehen, unterschieden werden müssen. Bei deduktiven Tests ist hinzukommend besonders darauf zu achten, dass quantitative Aspekte, wie zum Beispiel die Antwortzeiten des Systems, so realitätsnah wie möglich sind. Beide Arten von Usability-Tests haben jedoch gemeinsam, dass vorher festgelegte Aufgaben von den Nutzern unter Beobachtung bearbeitet werden [SB06, S. 163ff].

Die Auswahl der richtigen Testpersonen ist ein wichtiger Faktor für den Erfolg der Methode. Damit mögliche Probleme nicht umgangen werden können, sollten die Nutzer das System vor dem Test nicht kennen. Es sollte jedoch auch darauf geachtet werden, dass sie eine gewisse Vorerfahrung mit vergleichbaren Systemen haben. Die Anzahl der heranzuziehenden Testpersonen kann sich von System zu System unterscheiden. Nach Nielsen reichen 5 bis 6 Personen aus, um 80% der Usability-Probleme zu ermitteln [Nie94]. Sarodnick und Brau hingegen sagen, dass *„der real gefundene Anteil der Probleme mitunter deutlich davon abweichen kann“* und somit für eine *„Absicherung der weitgehend vollständigen Erfassung also deutlich mehr Personen beteiligt werden müssen“* [SB06, S. 167]. Faulkner beschreibt in einer ausführlichen Analyse, dass bereits 5 Testpersonen ausreichen, um durchschnittlich 85% der vorhandenen Probleme zu identifizieren. Bei 10 Teilnehmer steigt dieser Wert auf knapp unter 95% [Fau03].

Es gibt verschiedene Möglichkeiten, um Daten während eines Usability-Tests zu erheben. Die Bearbeitung der Aufgaben durch die Nutzer kann beispielsweise via Video aufgezeichnet werden. Das hat vor allem den Vorteil, dass schnelle Sequenzen nochmal detailliert beobachtet und ausgewertet werden können. Man muss jedoch bedenken, dass die Auswertung der Videoaufzeichnungen sehr zeitintensiv ist. Nach der Durchführung des Usability-Tests können diese Aufzeichnungen auch dafür genutzt werden, um bestimmte Szenarien nochmal genauer mit dem Nutzer zu besprechen. Dies erlaubt dem Benutzer, sich besser in die jeweilige Situation zurückzusetzen. Des Weiteren stellen die in Kapitel 3.2.4 beschriebenen Logfiles, die während eines Usability-Tests gespeichert werden, eine Basis für weitere Auswertungen dar. Hinzukommend kann die Denkweise der Nutzer verstanden werden, indem die Nutzer dazu aufgefordert werden, ihr Vorgehen durch „lautes Denken“ zu rechtfertigen [SB06, S. 169f].

Vor- und Nachteile

Einer der größten Vorteile von Usability-Tests ist, dass die Usability-Probleme von eigentlichen Nutzern gefunden werden und sie somit eine hohe Validität haben. Wie bereits beschrieben, genügen zwischen 5 und 10 Testpersonen, um zwischen 85% und 95% der vorhandenen Probleme zu identifizieren. Außerdem erkennen die Entscheidungsträger die gefundenen Probleme auch als wirkliche Probleme an. Die erhobenen Daten können vielseitig genutzt werden. Sie können zum einen Basis für weiterreichende Evaluationen sein, und zum anderen während des Usability-Tests für gezielteres Nachfragen genutzt werden.

Die Durchführung eines Usability-Tests erfordert jedoch einen hohen Zeit- und somit auch Kostenaufwand. Die durchzuführenden Aufgaben müssen zunächst ermittelt und in Frage kommende Testpersonen identifiziert werden. Das kann beispielsweise bei Lösungen, die auf eine einzelne Abteilung zugeschnitten sind, zu Problemen führen, da hier nur wenige potentielle Nutzer zur Verfügung stehen.

4. Planung und Durchführung ausgewählter Methoden

Dieses Kapitel handelt von der Planung und Durchführung ausgewählter Usability-Evaluationsmethoden. Am Anfang des Kapitels wird zunächst ein Überblick über das zu untersuchende System gegeben. Dabei soll vor allem auf dessen Einsatzmöglichkeiten und die zur Verfügung stehenden Funktionen eingegangen werden. Anschließend werden auf das System anwendbare Evaluationsmethoden ausgewählt, deren Planung und Durchführung danach detaillierter beschrieben werden.

4.1. Das System

Bei dem zu untersuchenden System handelt es sich um ein Enterprise 2.0 Werkzeug. Koch und Richter definieren Enterprise 2.0 folgendermaßen [KR08, S. 16]:

„Enterprise 2.0 bedeutet [...] die Konzepte des Web 2.0 und von Social Software nachzuvollziehen und zu versuchen, diese auf die Zusammenarbeit in den Unternehmen zu übertragen. Dabei spielt allerdings die (Weiterentwicklung der) Unternehmenskultur eine bedeutende Rolle.“

Enterprise 2.0 Werkzeuge sind somit Web 2.0 Werkzeuge, die in Unternehmen für interne oder externe Zusammenarbeit genutzt werden. Dazu zählen beispielsweise Blogs, Wikis, Social Bookmarks, Tags, Mashups und Soziale Netzwerke [IBM11].

Das zu evaluierende System trägt den Namen Tricia und verinnerlicht einige von den oben genannten Enterprise 2.0 Werkzeugen. Tricia kann entweder lokal oder bei einem Hosting-Partner gehostet werden. Die Software läuft als Webanwendung und das System kann somit mit jedem Browser bedient werden. Den Nutzern des Systems ist es beispielsweise möglich, verschiedene Wikis anzulegen. Diese Wikis bestehen hauptsächlich aus Wiki-Seiten. Es können sowohl an die Wikis, als auch an die Wiki-Seiten Dateien angehängt und gewünschte Zugriffsrechte zugewiesen werden. Durch ein Parent-Child Konzept können die Wiki-Seiten innerhalb eines Wikis in eine Baumstruktur übertragen werden. Der Nutzer kann die Wiki-Seiten editieren, das bedeutet, dass er sowohl einfachen Text, als auch Tabellen, Bilder oder Verlinkungen hinzufügen und entsprechend formatieren kann. Wikis und Wiki-Seiten können mit beliebigen Tags ausgestattet werden. Durch das Hinzufügen von Tags können gesuchte Inhalte leichter gefunden und ähnliche Inhalte strukturiert werden. Tricia bietet dem Nutzer zudem Funktionen eines hybriden Wikis an. Es erlaubt ihm dabei zum Beispiel die Nutzung sogenannter TypeTags. Einer Wiki-Seite kann dabei ein bestimmter Typ zugewiesen werden. Gibt es beispielsweise zu jeder Vorlesung eine Wiki-Seite, dann sollte auch jede dieser Wiki-Seiten vom gleichen Typ, zum

4. Planung und Durchführung ausgewählter Methoden

Beispiel „Vorlesung“, sein. Die Nutzung von TypeTags ermöglicht es dem Nutzer zum Beispiel, eine Liste aller Vorlesungen abzurufen und deren Attribute in Tabellenform dargestellt zu bekommen. Die eben erwähnten Attribute sind ein weiterer Bestandteil von hybriden Wikis. Attribute können jeder Wiki-Seite hinzugefügt werden. Die Attribute werden als Key-Value-Pairs dargestellt, da sie aus einem Schlüssel und einem zugewiesenen Wert bestehen.

Types: course seminar	
Acronym	OSem
ECTS	6
Language	German
Lecturer	 Prof. Dr. Florian Matthes
Level	Bachelor Master
Module No.	IN2122
Organizer	 Christopher Schulz
Rhythm	Summer & Winter
SWS	2
Type	Seminar
Incoming Links	
Referenced by	 Christian Neubert  Christopher Schulz  Dr. Sabine Buckl  Student projects and job opportunities

Abbildung 4.1.: TypeTags und Attribute in Tricia

Typen wie „Währung“ oder „Datum“ zugeordnet werden. Neben einer Versionierung bietet Tricia auch die Möglichkeit, gelöschte Seiten oder Dateien wiederherzustellen.

Neben den Wiki-Funktionen ermöglicht Tricia dem Nutzer zudem, Dateien nicht nur an Wiki-Seiten anzuhängen, sondern diese auch einfach in ein Verzeichnis hochzuladen und mit gewünschten Tags auszustatten. Auf diese Weise ist es den Benutzern möglich, Dateien nur für bestimmte Personen oder Gruppen zugänglich und durch die Tags leichter auffindbar zu machen. Die Dateien können beliebig oft angesehen oder heruntergeladen werden.

Die eben erwähnten Gruppen erlauben es dem Nutzer, den Zugriff auf Inhalte aller Art zu regulieren und ermöglichen somit eine kontrollierte Kollaboration zwischen einzelnen

Abbildung 4.1 zeigt die Darstellung der Attribute auf einer Wiki-Seite. Es ist zu erkennen, dass Attribute sowohl aus reinem Text, als auch aus Verlinkungen zu Personen oder anderen Wiki-Seiten bestehen können. So können die Wiki-Seiten miteinander verknüpft und zuvor unstrukturierte Inhalte strukturiert werden. Die Abbildung zeigt außerdem, welche TypeTags für die Wiki-Seite vergeben wurden. Diese Wiki-Seite ist sowohl vom Typ „course“, als auch vom Typ „seminar“. Klickt man beispielsweise auf einen der beiden Typen, erhält man eine tabellarische Auflistung der Wiki-Seiten dieses Typs. Hinzukommend werden alle Referenzen auf diese Wiki-Seite dargestellt. Den TypeTags und Attributen von Wiki-Seiten können zudem bestimmte Formate und Typen zugeordnet werden. Das bedeutet, dass eine Wiki-Seite vom Typ „course“ beispielsweise standardmäßig mit den Attributen „Lecturer“ und „ECTS“ ausgestattet ist. Den Attributen können somit beispielsweise

Nutzern. Es können beispielsweise auch E-Mails an alle Mitglieder einer Gruppe gesendet werden. Dadurch kann sich der Nutzer sicher sein, dass bestimmte Inhalte auch nur an die gewollten Adressaten gelangen.

Eine weiteres Feature, das von Tricia unterstützt wird, ist die Verwaltung von Blogs. Nutzer können zum Beispiel persönliche Blogs erstellen, die nur von ihnen editiert werden können. Die Lese- beziehungsweise Schreibberechtigungen können jedoch auch erweitert und jederzeit angepasst werden. Somit können auch mehrere Nutzer gleichzeitig an ein und demselben Blog arbeiten. Neben der Definition von verschiedenen Blog-Themes, können Nutzer auch RSS-Feeds abonnieren.

Wie in Abbildung 4.2 zu sehen, werden durch die Suchfunktion nicht nur Wiki-Seiten durchsucht, sondern auch Blogs oder Dateien in die Suche eingebunden. Die Suchfunk-



Abbildung 4.2.: Tricia Suchfunktion

tion kann sowohl als Quick-Navigation, als auch als Präfix-Suche genutzt werden. Dabei werden dem Nutzer passende Suchergebnisse bereits beim Tippen vorgeschlagen. Diese können dann direkt angeklickt und aufgerufen werden. Hinzukommend können neue Wiki-Seiten auch direkt innerhalb der Suchmaske erstellt werden. Tricia wird zum Beispiel bei sebis - einem Lehrstuhl für Informatik der TU München - unter anderem für die Verwaltung der Lehrstuhl-Website eingesetzt. Auch das Intranet der Fakultät für Informatik der TU München (TUM) wird mit Tricia realisiert. Dabei wird Tricia zum Beispiel für die Abwicklung der Promotionsdurchläufe genutzt. Der obere Teil von Abbildung 4.3 zeigt

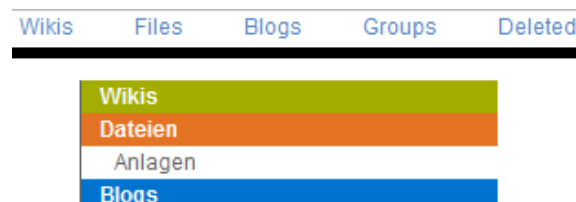


Abbildung 4.3.: Tricia Navigation

die Navigation von Tricia, die sich am oberen rechten Rand der sebis-Website befindet.

Im Gegensatz dazu befindet sich die Navigation der Intranet-Seite am linken Seitenrand - dargestellt im unteren Teil der Abbildung.

4.2. Auswahl der Methoden

In diesem Teil der Arbeit sollen nun aus den in Kapitel 3 beschriebenen Methoden verschiedene ausgewählt werden, um die Usability von Tricia zu untersuchen. Der Fokus soll dabei hauptsächlich auf den Wiki-Funktionen, den hybriden Eigenschaften und der Suchfunktion liegen. Blogs, Dateien und Gruppen werden dabei gegebenenfalls nur am Rande betrachtet. Tricia befindet sich zwar in ständiger Weiterentwicklung, es existieren jedoch bereits seit sehr langer Zeit Versionen, die alle beschriebenen Funktionen beinhalten und auch bereits von Kunden im alltäglichen Gebrauch eingesetzt werden. Da die in Abschnitt 3.1 genannten expertenorientierten Untersuchungsmethoden hauptsächlich in frühen Phasen des Entwicklungsprozesses eingesetzt werden, wenn beispielsweise noch offene Designfragen oder Prozessfolgen untersucht und geklärt werden müssen, eignen sich diese Methoden für die Usability-Evaluation von Tricia eher weniger. Außerdem stehen mit den eigentlichen Nutzern der Software die idealen Testpersonen für die in Abschnitt 3.2 erläuterten nutzerorientierten Methoden zu Verfügung, sodass die empirischen Methoden in diesem Fall den analytischen Methoden vorzuziehen sind.

Nun sollen verschiedene empirische Verfahren gewählt werden, um Usability-Probleme von Tricia zu identifizieren und zu untersuchen. Außerdem soll gezeigt werden, auf welche Weise sich die Methoden gegenseitig unterstützen und welche Art von Problemen durch welche Methoden gefunden werden können.

Da Tricia als Webanwendung läuft, bietet sich die Möglichkeit, die **Logfiles**, die vom Server gespeichert werden, zu analysieren. Auf diese Weise kann zum Beispiel die Nutzung bestimmter Funktionen über einen längeren Zeitraum untersucht und detailliert betrachtet werden. Dabei können einerseits Probleme bei der Nutzung einzelner Funktionen identifiziert werden, andererseits aber auch Funktionen entdeckt werden, die von Nutzern zum Beispiel überhaupt nicht genutzt werden. Bei der Durchführung von weiteren Analysen können dann beispielsweise dadurch gefundene Probleme genauer untersucht und somit besser verstanden werden.

Der Einsatz von Tricia als Intranet der Fakultät für Informatik der TUM bietet zudem die Möglichkeit, dass Nutzer, die das System regelmäßig nutzen, zu deren Problemen und Erfahrungen mit Tricia befragt werden können. Für eine solche Befragung kämen sowohl Fragebögen, als auch Gruppendiskussionen und Interviews in Frage. Da der Aufwand zur Erstellung eines Fragebogens sehr hoch ist und es nur wenige Nutzer des Intranets gibt, die auch die Wiki-Funktionen von Tricia regelmäßig nutzen, wird eine direkte Befragung der Mitarbeiter der Erstellung eines Fragebogens vorgezogen. Nun gilt es zu klären, ob eine Gruppendiskussion, ein Interview, oder beides durchgeführt werden soll. Aufgrund der stark begrenzten Nutzerzahl können nicht genügend Testpersonen gefunden werden, um sowohl eine Gruppendiskussion, als auch Einzel-Interviews durchzuführen. Da eine gemeinsame Terminfindung bei mehreren Testpersonen während der normalen Arbeitszeit oft sehr schwierig ist, die Teilnehmer unterschiedlich häufig mit dem System arbeiten und somit ungleiche Vorkenntnisse besitzen, und genügend Zeit für die Durchführung von Einzel-Interviews besteht, werden **Interviews** den Fokusgruppen in diesem Fall vor-

gezogen.

Als letzte Methode wird ein **Usability-Test** des Systems durchgeführt. Dabei soll vor allem untersucht werden, ob die Probleme, die bei regelmäßigen Nutzern des Systems auftreten, auch bei Personen auftreten, die das System noch nicht kennen. Bei der Durchführung des Usability-Tests soll zudem auf die Ergebnisse der Logfile-Analyse eingegangen werden, indem beobachtet wird, wie und warum die zuvor identifizierten Probleme bei der Nutzung des Systems in Erscheinung treten.

4.3. Logfile-Analyse

Da eine Logfile-Analyse des kompletten Systems zu aufwändig wäre, beschränkt sich die Logfile-Analyse von Tricia auf nur eine Funktion. Die eingebaute Suchfunktion, die eine der wichtigsten Funktionen von Tricia ist, bietet sich hierfür sehr gut an. Die Logfile-Analyse besteht deshalb aus zwei Teilen. Im ersten Teil wurde die Nutzung der Quick-Search auf Usability-Probleme untersucht. Im zweiten Teil wurde die Präfix-Suche näher betrachtet. Dabei galt es vor allem zu zeigen, welche Filterfunktionen von Nutzern genutzt werden und ob die Tag-Cloud zur weiteren Einschränkung der Suchergebnisse und somit zur Navigation genutzt wird. Die ausgewerteten Logfiles wurden vom Webserver des sebis-Lehrstuhls gespeichert.

4.3.1. Quick-Search

Page URL	Pageviews
quicksearch	1913
/used	1005
1	358
submit	218
2	191
3	68
4	33
5	20
6	18
undefined	2

Abbildung 4.4.: Piwik-Auswertung

Die Quick-Search Funktion von Tricia wird anhand des Open Source Webanalytik-Tools „Piwik“ (<http://de.piwik.org/>) untersucht. Piwik wertet dabei die Zugriffe und Eingaben der Quick-Search aus. Dadurch kann zum Beispiel bestimmt werden, der wievielte Vorschlag der automatischen Vervollständigung vom Nutzer ausgewählt wurde, und nach wie vielen eingegebenen Buchstaben er das Ergebnis ausgewählt hat. Abbildung 4.4 zeigt ein Beispiel einer Piwik-Auswertung. Auf diese Weise kann zum Beispiel gezeigt werden, ob der erste Vorschlag vom Nutzer auch am öftesten gewählt wird. Werden spätere Vorschläge öfter selektiert, kann das auf einen Fehler im Suchalgorithmus hindeuten. Die Anzahl der eingegebenen Buchstaben sind zudem ein Hinweis, wie schnell dem Nutzer die gesuchten Ergebnisse vorgeschlagen werden. Für die Untersuchung wurden die Daten von sieben Tagen herangezogen.

4.3.2. Präfix-Suche

Im Gegensatz zur Quick-Search, wurde die Präfix-Suche nicht mit der Hilfe von Piwik analysiert. Für die Untersuchung der Filterfunktion und Tag-Navigation wurden die Zugriffe eines bestimmten Zeitraums direkt aus dem Logfile des Webservers gezogen. Anschließend wurden alle Einträge der URL `http://wwwmatthes.in.tum.de/search` gefiltert und versucht, Einträge, die durch Bots oder Crawler entstanden sind, zu eliminieren. Zusätzlich wurden Einträge von Entwicklern, die ihre Funktionen testen, nicht in Betracht gezogen, da die Ergebnisse dadurch verfälscht werden könnten.

Die übrig gebliebenen URLs können folgende Parameter enthalten:

- **Query:** Der Query-Parameter gibt den Inhalt des Suchfeldes an, kann also bei einer leeren Suche auch leer sein.
- **Must:** Die Tags, die vom Nutzer in der Tag-Cloud zur Einschränkung der Suchergebnisse ausgewählt wurden, werden im Must-Parameter übergeben. Da nach und nach mehrere Tags ausgewählt werden können, können auch mehrere Must-Parameter gesetzt sein.
- **Kinds:** Der Kinds-Parameter speichert den Filter für den Typ der Suche. Hier kann also festgelegt werden, ob das Suchergebnis zum Beispiel ein Wiki, eine Wiki-Seite, eine Datei oder ein Blog-Post sein soll.
- **Spaces:** Der Spaces-Parameter gibt an, wo sich das Suchergebnis befinden soll. Damit kann der Filter beispielsweise auf ein bestimmtes Wiki gelegt werden.
- **TypeTags:** Der TypeTag, den das Ergebnis haben soll, wird im TypeTags-Parameter festgelegt.
- **AttributeName:** Hat man ein bestimmtes TypeTag ausgewählt, kann zusätzlich durch das AttributeName-Parameter ein bestimmtes Attribut ausgewählt werden.

Hinzukommend gibt es noch die Parameter „containsInvalidLinks“ und „containsInvalidValues“, die hier aber nicht näher erläutert werden. Abbildung 4.5 zeigt ein Beispiel

`/search?kinds=wikiPage&spaces=%2Fwikis%2Fsebis&typeTags=staff&query=ba`

Abbildung 4.5.: Beispiel für die aufgerufene URL einer Präfix-Suche

für die URL eines Logfile-Eintrags. Diese URL zeigt beispielsweise, dass das Wort „ba“ in das Suchfeld eingegeben wurde. Das Ergebnis soll eine Wiki-Seite sein, sich im sebis-Wiki befinden und den TypeTag „staff“ beinhalten. Auf diese Weise kann bestimmt werden, welche Filterfunktionen von Nutzern wie häufig genutzt werden. Außerdem kann untersucht werden, ob Filterfunktionen miteinander kombiniert werden, und wie die Tags zur Einschränkung beziehungsweise Navigation genutzt werden. Werden bestimmte Funktionen wenig oder überhaupt nicht genutzt, kann dies zum Beispiel ein Hinweis darauf sein, dass die Funktion nicht verstanden oder auch nicht gebraucht wird. In weiteren Analysen kann dann beispielsweise gezielter untersucht werden, warum die Nutzer diese Funktionen falsch verwenden.

Auch für die Auswertung der Logfile-Einträge der Präfix-Suche wurde der Datensatz einer Woche betrachtet. Nach dem Filtern der Bot-, Crawler- und Entwickler-Einträge waren noch knapp 2000 Einträge vorhanden. Die Ergebnisse der Evaluation werden in Kapitel 5 zusammen mit den Ergebnissen der Interviews und Usability-Tests erläutert.

4.4. Usability-Interviews

Wie in Abschnitt 4.2 bereits kurz erwähnt wurde, stellen die Nutzer des Intranets der Fakultät für Informatik die idealen Interviewpartner für eine Usability Evaluation dar. Zu diesem Zweck wurden fünf Nutzer des Systems identifiziert, die unter anderem auch mit den Wiki-Funktionen von Tricia arbeiten. Vier dieser Nutzer waren weiblich, arbeiteten in verschiedenen Lehrstuhl-Sekretariaten und hatten somit ein eher gering ausgeprägtes technisches Hintergrundwissen. Der männliche Interviewpartner war Informatiker und ist deshalb schon des öfteren mit Systemen dieser Art in Kontakt gekommen.

Da die Interviews während der Arbeitszeit der Nutzer durchgeführt wurden, wurde für die Dauer eines Interviews jeweils eine Zeit zwischen 30 und 60 Minuten angestrebt. Interviews sollten, wie in Kapitel 3.2.1 geschildert, auch nicht länger als 90 Minuten dauern. Der Zeitrahmen wurde bei jedem Interview eingehalten, wobei das längste Interview mit kurzen Unterbrechungen etwa 55 Minuten dauerte. Da die Interviews am Arbeitsplatz der Nutzer stattfanden, kam es zwar vereinzelt zu Unterbrechungen, der Großteil der Zeit konnte jedoch ungestört genutzt werden.

Um einen besseren Eindruck von den Interviewpartnern zu bekommen, wurden die ersten beiden Interviews vor allem dazu genutzt, den zuvor erarbeiteten Fragenkatalog auf Fehler und Vollständigkeit zu überprüfen. Dadurch konnten Verbesserungen und notwendige Anpassungen noch vor den restlichen Interviews eingearbeitet werden.

Der Fragenkatalog

Der anfänglich erarbeitete Fragenkatalog enthielt sowohl **offene**, als auch **geschlossene** Fragen. Nach den ersten beiden Interviews stellte sich aber zum Beispiel heraus, dass manche Funktionen wenig bis gar nicht benutzt werden. Der Fragenkatalog wurde daraufhin so modifiziert, dass explizite Fragen zu den nicht genutzten Funktionen in eher offene Fragen geändert wurden. Statt nach der Funktionsweise eines Features zu fragen, wurde von nun an mehr danach gefragt, was der Nutzer unter dem Feature versteht und warum er es nicht nutzt. Zu den offenen Fragen zählten zum Beispiel folgende:

- Wozu wird das System hauptsächlich genutzt?
- Gibt es allgemeine Probleme mit der Nutzung des Systems?
- Gibt es Verbesserungsvorschläge für das System?
- ...

Geschlossene Fragen waren zum Beispiel:

- Wie wird ein Wiki angelegt?

- Wie werden Tags hinzugefügt?
- Wie wird ein Attachment hochgeladen?
- ...

Im Anhang dieser Arbeit befindet sich der genutzte Fragenkatalog. Dieser sollte jedoch eher als Leitfaden interpretiert werden, da die Dynamik eines Gesprächs oft dazu führen kann, dass manche Fragen nicht gestellt oder neue Fragen eingebracht werden.

Am Anfang der Interviews wurde vom Interviewleiter immer eine kurze Einführung über den Hintergrund und Zweck des Gesprächs gegeben. Während die offenen Fragen, die dem Nutzer einen großen Spielraum für Antworten bieten und somit eine bessere Vertrauensbasis zwischen dem Leiter und dem Befragten aufgebaut werden kann, am Anfang des Gesprächs gestellt wurden, wurden die geschlossenen Fragen im späteren Verlauf des Gesprächs zur Identifizierung von Usability-Problemen genutzt. Alle fünf Interviews wurden in einem Zeitraum von zwei Wochen gehalten. Während der Gespräche wurden vom Interviewleiter handschriftliche Notizen zu den Ergebnissen gemacht. Zur Nachbearbeitung und um den Befragten so viel Aufmerksamkeit wie möglich zu schenken, wurden die Gespräche mit vorheriger Einwilligung der Befragten auf Audio aufgezeichnet. Für die Ergebnisse der Interviews siehe Kapitel 5.

4.5. Usability-Testing

Nach den Usability-Interviews wurden die Usability-Tests geplant und durchgeführt. Dieser Prozess lässt sich folgendermaßen unterteilen:

1. **Definition der durchzuführenden Aufgaben:** Wie in Kapitel 3 beschrieben, sollten einer Usability Evaluation immer auch Analyse- und Entwurfsphasen vorangehen, in denen die Aufgaben, die vom System zu erfüllen sind, zu definieren sind. Vor der Durchführung der Usability-Tests mussten somit Kernfeatures und Funktionen von Tricia identifiziert werden. Da sich die Usability-Evaluation besonders auf die Wiki-Funktionalität von Tricia beschränkt, ließen sich die folgenden für den Nutzer wichtige Funktionen identifizieren:
 - Registrieren und Anmelden am System
 - Anlegen und Bearbeiten von Wikis
 - Anlegen und Bearbeiten von Wiki-Seiten
 - Definieren einer Wiki-Homepage
 - Nutzung des Parent-Child Konzepts
 - Verwendung von Tags und TypeTags
 - Verwendung von Attributen
 - Verwaltung von Attachments
 - Nutzung der Suchfunktion

2. **Auswahl der Testpersonen:** Für die Durchführung des Tests wurden fünf Testpersonen ausgewählt, die das System noch nicht kannten. Diese Testpersonen waren zwischen 20 und 50 Jahren alt, hatten verschiedene berufliche Hintergründe und auch unterschiedliche Vorkenntnisse mit ähnlichen Systemen. Auf diese Weise sollte untersucht werden, ob manche Probleme entweder speziell nur bei Personen auftreten, die wenig technische Erfahrung haben, oder aber auch global vorhanden sind.
3. **Bereitstellung der Testumgebung:** Für die Durchführung des Usability-Tests wurde eine lokale Installation von Tricia bereitgestellt. Die lokale Installation enthielt zuvor angelegte Testdaten, damit die Nutzer sich ein Bild von dem System machen konnten. Der Test wurde nicht im Usability-Labor, sondern in einer ruhigen Umgebung mit wenig Umwelteinflüssen durchgeführt.
4. **Durchführung des Usability-Tests:** Am Anfang des Tests bekamen die Testpersonen eine kurze Einführung in das System. Dabei wurde sowohl der Zweck der Untersuchung erläutert, als auch die Grundfeatures von Tricia erklärt. Es wurde jedoch nicht auf die genaue Funktionsweise der Features eingegangen. Nach der Einführung stellte der Testleiter den Probanden Aufgaben, die sie zu bearbeiten hatten. Zunächst sollten sie sich am System registrieren und anmelden. Anschließend sollte jede Testperson ein eigenes Wiki anlegen. Dabei kamen die Teilnehmer das erste Mal mit den Tags in Berührung. Nach der Erstellung des Wikis sollte eine Homepage für das Wiki erstellt werden. Dazu musste zunächst eine neue Wiki-Seite im richtigen Wiki angelegt und diese schließlich als Homepage für das Wiki definiert werden. Im weiteren Verlauf hatten die Testpersonen die Aufgabe, eine Wiki-Seite zu einem frei wählbaren Thema (zum Beispiel Autos, Urlaub) anzulegen. Um das Parent-Child Prinzip zu testen, sollten die Nutzer dann durch weitere Wiki-Seiten realisierte Unterthemen aufbauen. Der Inhalt dieser Wiki-Seiten sollte dabei formatiert werden. Die Testpersonen sollten sowohl Tabellen und Bilder einbinden, als auch Attachments an die Wiki-Seite anhängen, verlinken und wieder löschen. Zum Abschluss des Tests sollten die Nutzer die Nutzung der Suchfunktion simulieren. Dabei wurde auch auf die Verwendung der Tag-Cloud eingegangen.

Der Usability-Test dauerte im Durchschnitt etwa 90 Minuten. Das Gespräch zwischen Testleiter und Testperson kann als interaktiv beschrieben werden. Dabei wurden die Testpersonen zum lauten Denken angeregt und beim Auftreten von Problemen, die bereits durch die Interviews bekannt waren, genauer nachgehakt. So weit wie möglich wurden nur Verständnisfragen beantwortet. Die Nutzer waren deshalb bei der Bearbeitung der Aufgaben auf sich allein gestellt. Konnte eine Aufgabe nicht bearbeitet werden, wurden vom Versuchsleiter einzelne Hinweise auf die Lösung gegeben. Die einzelnen Tests wurden nicht auf Video aufgezeichnet, da der Mehrwert einer Nachbereitung den hohen Aufwand nicht rechtfertigen würde. Allgemein kann gesagt werden, dass die Handhabung und Funktionalitäten des Systems von den Testpersonen als sehr positiv beschrieben wurden.

5. Auswertung

Kapitel 5 handelt von den Usability-Problemen von Tricia, die durch die Usability-Interviews (siehe 4.4), die Usability-Tests (siehe 4.5) und der Logfile-Analyse (siehe 4.3) aufgedeckt werden konnten. Dabei geht es in diesem Abschnitt primär darum, die entdeckten Usability-Probleme zu beschreiben und mögliche Verbesserungsvorschläge zu äußern. Im folgenden werden die Usability-Probleme verschiedenen Bereichen oder Funktionen von Tricia zugeordnet.

5.1. Registrierungs- und Anmeldeprozess

Bevor die Nutzer Wiki-Funktionen wie zum Beispiel das Erstellen und Editieren von Wikis beziehungsweise Wiki-Seiten nutzen können, müssen sie einen Account beim System registrieren und sich anschließend anmelden. Da die Nutzung der Wiki-Funktionen nur Administratoren möglich ist, wurden die in den Usability-Tests von den Nutzern erstellten User vom Versuchsleiter umgehend freigeschaltet. Der Registrierungs- und Login-Prozess wurde ausschließlich bei der Durchführung der Usability-Tests untersucht, da die Interview-Partner bereits Accounts hatten und wegen regelmäßiger Nutzung des Systems auch keine Probleme beim Login hatten.

Der Link zur Registrierung befindet sich neben der Login-Box am oberen rechten Rand der Seite und wurde von allen Nutzern sofort entdeckt (siehe Abbildung 5.1). Nach dem



Abbildung 5.1.: Login und Registrierungsmöglichkeit

Klicken auf den Registrierungslink haben die Nutzer die Wahl zwischen dem Anfordern eines neuen Passworts für einen bereits bestehenden Account, oder die Erstellung eines neuen Mitgliederprofils. Nachdem die zweite Option gewählt wurde, sehen sich die Nutzer mit der in Abbildung 5.2 dargestellten Maske konfrontiert. Obwohl oberhalb der Maske beschrieben wird, dass der Nutzer seinen vollen Namen eingeben soll, kam es hier zu einem ersten Problem. Während manche Nutzer den Hinweis gelesen hatten und ihren Vor- und Nachnamen in das Feld eintrugen, füllten andere Nutzer das Feld beispielsweise nur mit ihrem Nachnamen oder einem Nickname aus. Dies könnte jedoch verhindert werden, indem zum Beispiel die Eingabe von Vor- und Nachname voneinander getrennt gefordert wird. Auch ein Hervorheben des Hinweises kann hier zur Vermeidung von Falscheingaben führen.

5. Auswertung

Nach dem erfolgreichen Registrieren sollten sich die Nutzer nun am System mit der in Abbildung 5.1 dargestellten Maske anmelden. Auch hier hatten einige Nutzer Probleme, die richtigen Angaben im Benutzerfeld zu machen. Während das System die Eingabe einer E-Mail Adresse erwartet, befüllten vor allem Nutzer, die bei

der Registrierung einen Nickname verwendet hatten, das Feld nicht mit ihrer angegebenen E-Mail, sondern mit dem Nickname. Nachdem das System einen Hinweis auf den Fehler gegeben hatte, waren alle Probanden fähig, sich ohne weitere Probleme einzuloggen. Auch hier kann aber der Nutzer soweit eingeschränkt werden, dass er keine Falscheingabe tätigt. Hierfür könnte das E-Mail Feld zum Beispiel mit dem Inhalt „E-Mail“ vordefiniert sein. Es könnte jedoch auch der Ansatz von Google genommen werden, die bei ihrer Login-Maske ein Beispiel angeben (siehe Abbildung 5.3). Tabelle 5.1 zeigt, bei welchen Usability-Testern

I want to create a profile for this site.

Please enter your full name and choose a password for this site:

Name	<input type="text"/>
Password	<input type="password"/>
Password - Again	<input type="password"/>

Abbildung 5.2.: Tricia Registrierungs-Maske

Abbildung 5.3.: Google Login - siehe <http://www.google.com>

diese beiden Fehler aufgetreten sind, wobei ein „X“ das Auftreten des Fehlers markiert, „TP“ für Testperson und „IP“ für Interviewpartner steht. Trat ein Fehler nicht auf, wird ein „-“ eingetragen. Die Reihenfolge der Teilnehmer wird bei den Tabellen weiterer Probleme beibehalten, sodass am Ende eine zusammengefasste Tabelle erstellt werden kann. In diesem Fall traten beide Fehler bei drei von fünf Testpersonen auf. Besonders hervorzuheben ist, dass alle Nutzer, die einen Fehler beim Ausfüllen der Registrierung gemacht haben, auch eine Falscheingabe beim Login getätigt haben. Dies ist ein Hinweis darauf, dass ein Fehler unter Umständen auch zu einem zweiten Fehler führen kann.

	TP1	TP2	TP3	TP4	TP5	IP1	IP2	IP3	IP4	IP5
Falscheingabe Name	X	-	X	X	-	-	-	-	-	-
Falscheingabe Login	X	-	X	X	-	-	-	-	-	-

Tabelle 5.1.: Usability-Probleme bei Registrierung und Login

5.2. Wikis

Dieser Abschnitt beschäftigt sich mit Usability-Problemen, die bei der Erstellung und dem Editieren von Wikis auftreten. Bei der Durchführung der Interviews ist dabei besonders aufgefallen, dass Wikis von Entwicklern des Systems erstellt werden und die befragten Nutzer selten beziehungsweise nur geringfügige Änderungen an bereits bestehenden Wikis vornehmen.

Ein neues Wiki kann durch einen Klick auf „New Wiki“ im Wiki-Bereich des Systems erstellt werden. Dieser Link befindet sich am oberen rechten Rand der Seite und wurde sowohl von manchen Interviewpartnern, als auch vereinzelt von den Usability-Testpersonen nicht sofort entdeckt. Es ist jedoch schwer, hierfür ohne tiefere Untersuchungen Verbesserungen vorzuschlagen. In weiteren Analysen könnte deshalb zum Beispiel untersucht werden, ob eine Replatzierung des Buttons, die Änderung der Schriftfarbe, oder das Ändern der Bezeichnung auf „Create a new Wiki“ zu einer besseren Vermeidung des Problems führen.

Nachdem der Nutzer auf den Link geklickt hat, kann er verschiedene Eigenschaften des Wikis festlegen. Dabei können beispielsweise Tags, die in Abschnitt 5.4 näher beschrieben werden, vergeben werden, Schreib- und Leseberechtigungen zugeteilt, und sogenannte „Expert Settings“, dargestellt in Abbildung 5.4, definiert werden.

Die Interviewpartner und Usability-Tester hatten keine Probleme damit, die unterschiedlichen Lese- und Schreibzugriffe zu verstehen und voneinander abzugrenzen. Beide Parteien konnten sich unter den Expert Settings jedoch nur sehr wenig vorstellen. Während die Nutzer des Intranets die Experteneinstellungen überhaupt nicht beachtetten, was wahrscheinlich vor allem daran lag, dass sie normalerweise mit vorgefertigten Wikis arbeiteten, überlegten die Usability-Tester längere Zeit, welche Eingaben getätigt werden sollten und entschieden sich schließlich dafür, keine Änderungen vor-

Expert Settings	
Theme	<input type="text"/> <i>Available themes: defaultStyle</i>
Show Comments	<input checked="" type="checkbox"/> <input type="checkbox"/> Set for all existing pages
Enable \$TriciaScript\$	<input checked="" type="checkbox"/> <input type="checkbox"/> Set for all existing pages
Show Templates	<input checked="" type="checkbox"/> <input type="checkbox"/> Set for all existing pages

Abbildung 5.4.: Expert Settings bei der Wiki-Erstellung

zunehmen. Experteneinstellungen sollten nur von Nutzern geändert werden, die sich mit dem System über längere Zeit beschäftigt haben. Da beide Parteien letztendlich keine Änderungen vorgenommen haben, ist es zu keinen weiteren Problemen gekommen. Unabsichtliche Änderungen der Nutzer könnten zusätzlich noch erschwert werden, indem die Experteneinstellungen standardmäßig ausgeblendet werden. Dies könnte zum Beispiel durch ein Drop-Down realisiert werden.

Nach der Erstellung des Wikis hatten die Usability-Tester die Aufgabe, eine Homepage für ihr Wiki zu erstellen. Dazu musste eine neue Wiki-Seite angelegt werden. Die dabei aufgetretenen Probleme werden im nächsten Abschnitt der Auswertung beschrieben und

5. Auswertung

erläutert. Die Zuordnung der Homepage kann in den Wiki-Einstellungen getätigt werden. Einige der Usability-Tester konnten die gewünschte Einstellung nicht finden und konnten erst durch Hinweise auf den richtigen Weg gelenkt werden. Bei der Erstellung des Wikis gibt es keine Möglichkeit, eine Homepage zu definieren. Dies lässt sich logischerweise vermuten, weil noch keine Wiki-Seiten für dieses Wiki zur Verfügung stehen. Da die Nutzer bei der Erstellung des Wikis jedoch keine Einstellungsmöglichkeit für die Homepage sehen, suchen sie anschließend auch an dieser Stelle nicht mehr danach. Dies kann dann unter anderem dazu führen, dass die Option an ganz anderen Stellen im System gesucht und nicht gefunden wird. Abbildung 5.5 zeigt im oberen Teil einen Ausschnitt der Einstellungsmöglichkeiten nach der Wiki-Erstellung und im unteren Teil einen Ausschnitt der Einstellungen während der Wiki-Erstellung. Es ist zu sehen, dass die Definition der Homepage erst nach der Erstellung des Wikis vorgenommen werden kann und der Nutzer davor auch keinen Hinweis darauf bekommt, dass die Einstellung in dieser Maske getätigt werden kann. Der Nutzer könnte hierbei zum Beispiel entweder durch einen Hin-

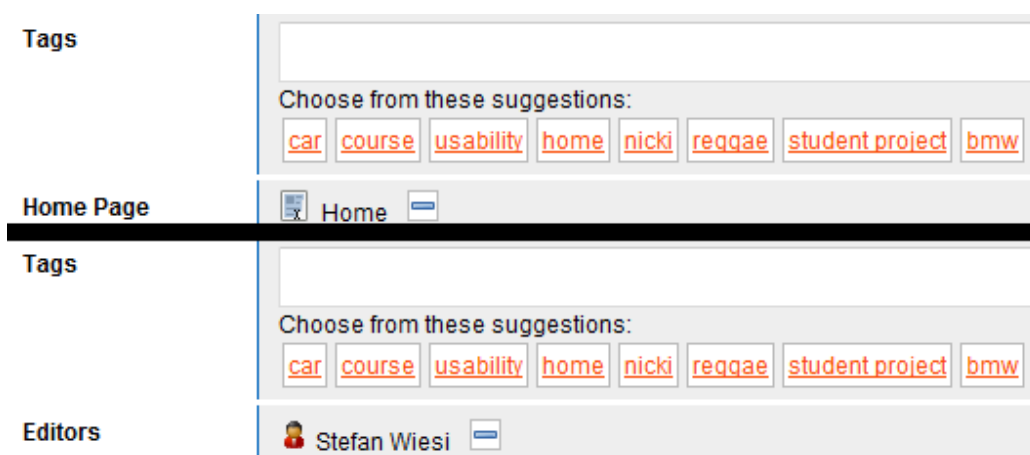


Abbildung 5.5.: Vergleich der Einstellungsmöglichkeiten während und nach der Wiki-Erstellung

weis während der Erstellung des Wikis darauf aufmerksam gemacht werden, dass die Homepage hier definiert werden kann, oder aber die Einstellungsmöglichkeit wird schon implementiert, eine Wiki-Seite kann aber noch nicht als Homepage zugeordnet werden.

Den Interviewten und Usability-Testern wurde außerdem die Frage gestellt, wie man alle Wiki-Seiten eines Wikis anzeigen kann, während man sich auf einer Wiki-Seite des Wikis befindet. Dafür wurde oben rechts neben den Optionen zum Editieren oder Erstellen einer neuen Wiki-Seite die Funktion „Browse this Wiki“ eingebaut. Während die Usability-Tester die Aktion nur selten wahrnahmen und einen Umweg über die Navigation machten, erzählten auch die Nutzer des Intranets, dass sie lange Zeit nicht wussten, dass die Funktion „Browse this Wiki“ dieses Verhalten hervorruft. Ein Nutzer hatte den Button sogar trotz regelmäßiger Nutzung noch nie betätigt. Auf die Frage, was die Nutzer sich denn unter der Bezeichnung vorstellten, konnten nur wenige das hervorgerufene Verhalten des Buttons korrekt beschreiben. Tabelle 5.2 fasst noch einmal die in diesem Abschnitt beschriebenen Probleme zusammen. Wurde eine Funktion während der Interviews nicht behandelt, so wird auch dann ein „-“ in die jeweilige Zelle eingetragen.

	TP1	TP2	TP3	TP4	TP5	IP1	IP2	IP3	IP4	IP5
New Wiki Button	-	-	X	X	-	X	-	-	-	-
Homepage definieren	-	-	X	X	X	-	-	-	-	-
Expert Settings	X	X	X	X	X	X	-	X	X	X
Browse Wiki Funktion	X	-	X	X	-	X	-	-	X	-

Tabelle 5.2.: Usability-Probleme bezüglich der Erstellung und Verwaltung von Wikis

5.3. Wiki-Seiten

Nachdem die Usability-Tester ein Wiki erstellt hatten, bekamen sie die Aufgabe, eine Homepage für das eben erstellte Wiki zu definieren. Dazu mussten sie innerhalb des Wikis eine neue Wiki-Seite anlegen. Sowohl beim Anlegen der neuen Seite, als auch beim Einfügen und Editieren von Texten, Bildern und Tabellen, kam es dabei zu verschiedenen Problemen. Nutzer versuchten zum Beispiel, gemachte Fehler oder Änderungen durch das Drücken auf den Zurück-Button des Browsers wieder rückgängig zu machen. Normalerweise erscheint nach dem Betätigen dieses Buttons ein Pop-up, welches den Nutzer darauf hinweist, dass nicht gespeicherte Inhalte verloren gehen könnten. Dies war jedoch nicht immer der Fall, und es kam mehrmals vor, dass deswegen Nutzer bereits getätigte Eingaben wiederholen mussten. Dieses Nutzer-Verhalten konnte jedoch ausschließlich bei den Usability-Tests beobachtet werden. Das Vorgehen der Interviewpartner kann als sehr vorsichtig beschrieben werden. Entwürfe werden dabei sehr häufig gespeichert, und auch Einstellungs-Änderungen nur vorgenommen, wenn sich die Person über die ausgelösten Aktionen der Änderungen vollkommen im Klaren waren. Hier sollte der Nutzer beim Versuch, die Seite ohne vorheriges Speichern zu verlassen, **immer** darauf hingewiesen werden, dass Änderungen, die nicht gespeichert wurden, unter Umständen verloren gehen könnten.

Obwohl bei der Erstellung einer neuen Wiki-Seite bereits in der Überschrift auf das betreffende Wiki hingewiesen wird (siehe Abbildung 5.6), und auch im weiteren Verlauf der Erstellungsmaske die Zuordnung von Wiki und Wiki-Seite nochmal dargestellt wird, be-

Add Page to Wiki "Home Wiki"

Abbildung 5.6.: Erstellung einer neuen Wiki-Seite

richteten die Interviewpartner, dass sie sich oft nicht sicher waren, in welchem Wiki sie die neue Wiki-Seite anlegen. Auch Usability-Tester konnten auf Nachfrage vereinzelt nicht sagen, in welchem Wiki sich die Wiki-Seite, die sie eben erstellen, befinden wird. Eine Anpassung des Systems, das hier zu einer besseren Vermeidung des Problems führt, kann hier jedoch nur schwer gefunden werden, da sowohl die Überschrift sehr groß, als auch die in der Maske zusätzlich vorhandene Zuordnung für den Nutzer leicht zu verstehen sein sollte.

Menü-Leiste

Bei den in Abbildung 5.7 gezeigten Aktionen und Formatierungsmöglichkeiten, wurden von den Nutzern und Interviewpartnern zwei Probleme entdeckt, wobei der obere Teil des Bildes die minimierte Leiste, und der untere Teil die maximierte Leiste zeigt. Der Nutzer kann die maximierte Leiste durch ein Klicken des weißen Pfeiles mit roter Umrandung aktivieren. Ein Problem, das den meisten Interviewpartnern bei ihrer Arbeit mit



Abbildung 5.7.: Minimierte und maximierte Menü-Leiste

dem System aufgefallen ist, und auch mehrmals als Verbesserungsoption genannt wurde, ist die fehlende Möglichkeit, Texte zu unterstreichen. Hinzukommend sahen manche Interviewpartner die komplette Menü-Leiste während des Interviews zum ersten Mal. Dies kann vor allem daran liegen, dass vom System gespeichert wird, ob der Nutzer die minimierte oder maximierte Menü-Leiste aktiviert hat. Klickt ein Nutzer nie auf den Pfeil zur Darstellung der gesamten Leiste, wird das System ihm diese auch in Zukunft standardmäßig nicht vorschlagen. Zur Verbesserung der Formatierungsmöglichkeiten sollte die Unterstreichen-Funktion unbedingt hinzugefügt werden. Außerdem könnte dem Nutzer standardmäßig die ausgeklappte Menü-Leiste angezeigt werden. Auf diese Weise sieht er von Anfang an alle Optionen und kann diese auf Wunsch auch wieder ausblenden.

Tabellen

Ein weiterer Bereich, bei dem sowohl die Interviewpartner, als auch die Usability-Tester sehr viele Schwierigkeiten hatten, ist das Anlegen und Editieren von Tabellen. Durch das Wählen eines in der Menü-Leiste vorhandenen Buttons, gelangt der Nutzer zu der in Abbildung 5.8 dargestellten Maske. Hier kann der Nutzer die Anzahl der Reihen und Spalten bestimmen. Außerdem kann die Ausrichtung der Zellen, die Stärke des Rahmens und die Tabellenüberschrift festgelegt werden. Mit den Optionen für „Cellpadding“ und „Cellspacing“ konnten sowohl die Interviewten, als auch die Usability-Tester wenig bis nichts anfangen. Vor allem aber die äußerst wichtige Funktion der „Class“, durch die beispielsweise Zellumrandungen sichtbar gemacht werden können, wurde von den Nutzern nicht verstanden und deswegen auch nicht genutzt. Manche Interviewpartner waren deshalb nicht in der Lage, eine Tabelle nach ihren Ansprüchen zu

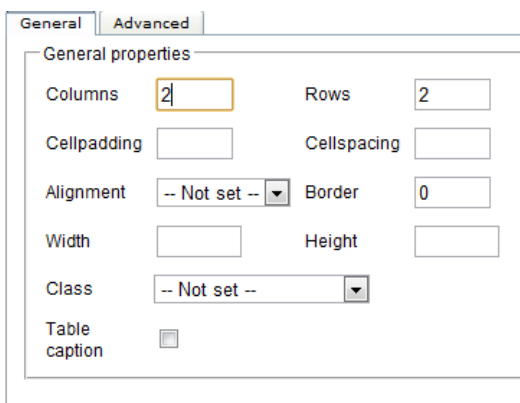


Abbildung 5.8.: Erstellung einer Tabelle

erstellen und diese mit bereits vorhandenen Daten zu befüllen und suchten deshalb nach Hilfe bei den Entwicklern. Hinzukommend wurden die Optionen für Fortgeschrittene nicht verstanden. Dies wird in diesem Fall jedoch nicht als Usability-Problem gewertet, da die Interviewpartner nicht zu der Gruppe der Power-User gezählt werden können und die Usability-Tester das System bisher noch nicht kannten. Abbildung 5.9 zeigt jedoch einen Darstellungsfehler, der als Usability-Problem verstanden werden kann. Die Nutzer waren dabei nicht in der Lage zu erkennen, welche Optionen sich hinter den deplatzierten Buttons versteckten. Das Problem trat dabei browserunabhängig auf.

Hatten die Usability-Tester eine Tabelle erstellt, so sollten sie diese befüllen. Dabei wurde vor allem beobachtet, dass die Nutzer Probleme mit der Navigation hatten. Die Navigation zu anderen Zellen innerhalb einer Tabelle erfolgt in Tricia durch die Pfeil-Tasten. Viele waren es durch die Programme Microsoft Excel oder Word jedoch gewohnt, die nächste Zelle

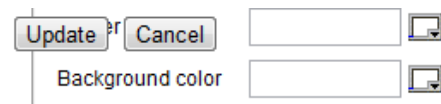


Abbildung 5.9.: Tabelleneigenschaften Darstellungsfehler

in einer Zeile durch Betätigen der Tabulator-Taste auszuwählen. Auch das Springen in die untere Zelle durch das Benutzen der Enter-Taste führte die Nutzer deshalb nicht zum gewünschten Ergebnis. Manche wählten deshalb den umständlichen Weg und selektierten die gewünschte Zelle mit der Maus. Nutzer, die so reagierten, waren zusätzlich nicht in der Lage, die Pfeil-Tasten als Navigationsmöglichkeit zu identifizieren.

Um die Navigation innerhalb einer Tabelle zu vereinfachen, sollte diese an die oben beschriebene Navigation angeglichen werden. Damit Nutzer die Wirkung verschiedener Tabelleneigenschaften besser erkennen können, könnte in Anlehnung an das Editieren von Bildern (wird im nächsten Abschnitt erläutert) eine Beispieltabelle gezeigt werden, die die vom Nutzer gemachten Einstellungen beispielhaft anzeigt.

Bilder

Die Interviewpartner beschrieben keine Probleme beim Hochladen oder Editieren von Bildern. Auch die Usability-Tester hatten keine Probleme damit, Bilder in eine Wiki-Seite einzubauen oder die Eigenschaften für diese anschließend zu ändern. Dies hängt vor allem von der in Abbildung 5.10 dargestellten Maske ab, die die vom Nutzer festgelegten Eigenschaften umgehend beispielhaft anzeigt.

Struktur

Wie bereits öfter gesagt, können Wiki-Seiten durch ein Parent-Child Konzept einander zugeordnet und somit strukturiert werden. Den Interviewpartnern war dieses Konzept meist bekannt, es wurde jedoch so gut wie nicht genutzt, da dessen Anwendung keinen zusätzlichen Nutzen für die Interviewten erbracht hätte. Manche der Usability-Tester konnten sich unter dem Parent-Child Konzept jedoch nichts vorstellen und waren somit nicht in der Lage die Wiki-Seiten innerhalb eines Wikis in einer Struktur zu erfassen. Beim Erstellen einer Wiki-Seite hat der Nutzer zusätzlich noch die Möglichkeit, Tags und Type-

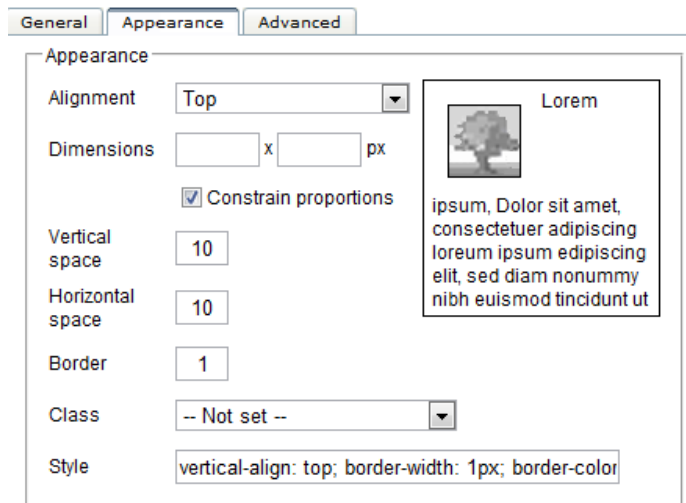


Abbildung 5.10.: Änderung von Bildeigenschaften

Tags zu vergeben, Attribute zu definieren oder Attachments hochzuladen. Die dabei aufgetretenen Probleme werden in den kommenden Kapiteln erläutert. Tabelle 5.3 fasst nun die in diesem Abschnitt beschriebenen Probleme zusammen. Dabei fällt vor allem auf, dass sowohl die Interviewpartner, als auch die Usability-Tester sehr viele Probleme bei der Handhabung von Tabellen hatten und ohne Hilfe die gewünschten Ziele auch oft nicht erreichen konnten. Fehlende Formatierungsmöglichkeiten fielen jedoch ausschließlich den Nutzern auf, die regelmäßig und über einen längeren Zeitraum mit dem System gearbeitet hatten. Der Darstellungsfehler wurde durch die Interviews nicht erkannt, weil sich die Interviewten mit den Einstellungen für Fortgeschrittene nicht beschäftigten und auch während des Interviews über diese Einstellungen nicht gesprochen wurde.

	TP1	TP2	TP3	TP4	TP5	IP1	IP2	IP3	IP4	IP5
Zurück-Button	-	X	X	-	X	-	-	-	-	-
Zuordnung Wiki-Seite	X	-	-	-	X	X	-	-	X	-
Unterstreichen-Option	-	-	-	-	-	X	-	X	X	-
Maximierte Menüleiste	X	-	-	-	X	X	-	-	X	-
Tabelleneigenschaften	X	X	X	X	X	X	-	X	X	X
Darstellungsfehler	X	X	X	X	X	-	-	-	-	-
Tabellennavigation	X	X	X	X	X	-	-	-	-	-
Parent-Child Konzept	-	X	X	-	X	-	-	-	-	-

Tabelle 5.3.: Usability-Probleme beim Erstellen und Editieren von Wiki-Seiten

5.4. Tags

Wie in mehreren Abschnitten der Arbeit bereits beschrieben, können Nutzer sowohl bei der Erstellung von Wikis, als auch beim Anlegen von Wiki-Seiten Tags hinzufügen. Wiki-Seiten können zudem mithilfe von TypeTags verschiedene Typen zugeordnet werden. Usability-Tester und Interviewpartner beschrieben Tags als „Schlagwörter“, die für Inhalte vergeben werden, um diese besser finden zu können. Während der Durchführung der Usability-Interviews fiel aber vor allem auf, dass die Nutzer so gut wie nie Tags oder TypeTags verwendeten. Auf die Frage, warum sie keine Tags nutzen, wurden zum Beispiel folgende Antworten gegeben:

„Ich verwalte nur zwei Wiki-Seiten. Warum sollte ich da anfangen Tags zu vergeben, wenn ich meine Inhalte doch schnell über die Navigation finden kann? [...] Sobald ich mehrere Wiki-Seiten habe, würde ich jedoch gegebenenfalls anfangen, Tags zu vergeben.“

„Wir [die Sekretärinnen] haben uns schon öfter überlegt, für unsere Wiki-Seiten Tags zu vergeben, haben es aber dann nie wirklich gemacht.“

Auch der Unterschied zwischen Tags und TypeTags war vielen Nutzern nicht klar. Während der Usability-Tests hätten viele der Nutzer, ohne dass ein expliziter Hinweis darauf gegeben wurde, überhaupt keine Tags vergeben. Manche fühlten sich durch die mehrfache Eingabe sogar genervt:

„Jetzt habe ich bereits für das Wiki und die Wiki-Seite Tags vergeben, warum muss ich noch TypeTags vergeben oder einer angehängten Datei Tags zuweisen?“

Aus diesen Sichtweisen geht hervor, dass die Nutzer den globalen Nutzen der Tags nicht verstehen, da sie die Vergabe von Tags nur zu ihrem eigenen Vorteil nutzen würden. Manchen Befragten schien nicht klar zu sein, dass das Benutzen von Tags besonders auch die Suche anderer Nutzer unterstützt.

Beim Hinzufügen von Tags - dargestellt in Abbildung 5.11 - hatten manche Nutzer ihre Probleme.

Tags können einerseits direkt aus einer Vorschlag-Liste ausgewählt werden, andererseits können neue Tags durch manuelle Eingabe hinzugefügt werden. Das Bestätigen eines neuen Tags wird dabei mit der Enter-Taste erzielt. Einige der Nutzer - auch Interviewpartner,

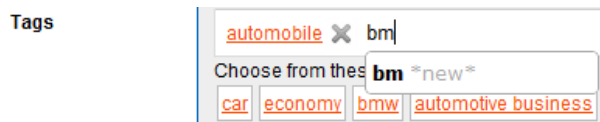


Abbildung 5.11.: Vergabe von Tags

die darum gebeten wurden, testweise neue Tags hinzuzufügen - erkannten nicht, dass ein neues Tag durch „Enter“ bestätigt wird, und trennten verschiedene Tags mit einem Komma. Auf die Frage, warum sie die Tags mit einem Komma separierten, kam zum Beispiel folgende Antwort:

5. Auswertung

„Bei YouTube werden unterschiedliche Tags durch ein Komma getrennt. Auch bei Facebook hat man früher verschiedene Interessen, Filme oder Bücher durch ein Komma voneinander abgrenzen können.“

Abbildung 5.11 zeigt außerdem, dass bereits vorhandene Tags in der Vorschlag-Liste nicht auftauchen. Obwohl das Tag „bmw“ bereits existiert, wird es dem Nutzer bei der Eingabe von „bm“ nicht vorgeschlagen. Dieses Problem ist jedoch nur wenigen Usability-Testern aufgefallen.

Nutzer können Tags wieder löschen, indem sie mit dem Mauszeiger über das zu löschende Tag fahren und anschließend darauf klicken. Außerdem können Tags durch Benutzung der Return-Taste wieder entfernt werden. Beim Löschen der Tags durch einen Mauszeigerklick (siehe Abbildung 5.12) hatte keiner der Interviewpartner und Usability-Tester ein Problem. Verwendeten die Nutzer aber zum Beispiel zur Beseitigung von Eingabefehlern die Return-Taste, kam es manchmal vor, dass das voranstehende Tag gelöscht wurde. Das war der Fall, da ein voranstehendes Tag bereits entfernt wird, wenn das danach stehende Leerzeichen entfernt wird. Da während des Vergabe-Prozesses Tags normaler-

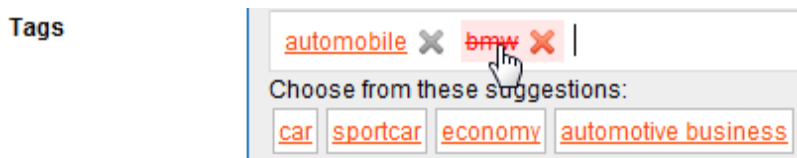


Abbildung 5.12.: Löschen von Tags

weise vor den TypeTags vergeben werden, und die Nutzer somit den Umgang mit Tags bei der Vergabe von Tags lernen konnten, traten beim Hinzufügen und Löschen von TypeTags (siehe Abbildung 5.13) nur noch vereinzelt die gleichen Probleme auf. Die in diesem

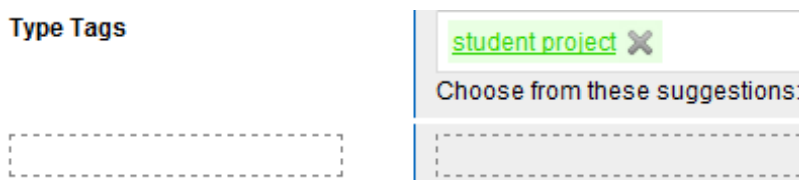


Abbildung 5.13.: Hinzufügen von TypeTags

Abschnitt erläuterten Usability-Probleme werden noch einmal in Tabelle 5.4 zusammengefasst.

Um die Nutzer zur Verwendung von Tags anzuregen, müssen den Nutzern deren Vorteile näher gebracht werden. Dies kann zum Beispiel durch die besondere Betonung der Vorteile bei Vorführungen oder Schulungen bezüglich des Systems erreicht werden. Es gilt noch weiter zu untersuchen, wie den Nutzern der Unterschied zwischen Tags und TypeTags besser verständlich gemacht werden kann. Da der beim Löschen von Tags beschriebene Fehler von nur sehr geringer Bedeutung ist, und der Löschprozess ansonsten von allen Nutzern intuitiv gemeistert werden konnte, sollte dieser auch nicht zwingend angepasst werden. Bereits vorhandene Tags sollten dem Nutzer jedoch so früh wie möglich zur Auswahl angezeigt werden. Dadurch kann auch verhindert werden, dass verschiedene Tags

	TP1	TP2	TP3	TP4	TP5	IP1	IP2	IP3	IP4	IP5
Tags hinzufügen	-	X	-	X	X	-	-	X	-	-
Tags löschen	X	-	-	-	X	-	-	-	-	-
Tags \neq TypeTags	X	X	X	-	-	X	-	X	X	-
Vorschlag-Liste	-	-	-	X	X	-	-	-	-	-

Tabelle 5.4.: Usability-Probleme bei der Verwaltung von Tags

für das gleiche Thema angelegt werden. Ein Beispiel hierfür wäre, wenn sowohl das Tag „Enterprise 2.0“, als auch das Tag „Enterprise2.0“ im System hinterlegt sein würden. Beim Hinzufügen von Tags kann in weiteren Tests analysiert werden, ob ein Hinweis darauf, dass ein neues Tag mit „Enter“ angelegt wird, den Prozessschritt für die jeweiligen Nutzer intuitiver macht.

5.5. Attribute

Attribute wurden - wie zuvor Tags und TypeTags - von den Interviewpartnern so gut wie nicht benutzt. Dies hat vor allem den Grund, dass die Befragten keinen Nutzen darin sahen, ihre Wiki-Seiten mit Attributen auszustatten. Da die Verwendung von Attributen bei den meisten der angelegten Wiki-Seiten nicht sinnvoll gewesen wäre, konnten nur die Usability-Tests zur Untersuchung des Themas genutzt werden. Dabei traten bei der Durchführung der Tests immer wieder zwei Probleme auf. Nutzer ließen sich zum Beispiel durch die Vorschlag-Liste leicht verleiten, eine neue Wiki-Seite für einen bisher nicht vorhandenen Begriff anzulegen, obwohl dies in den meisten Fällen jedoch nicht nötig gewesen wäre. Abbildung 5.14 zeigt zur besseren Veranschaulichung, wie ein Attribut vom

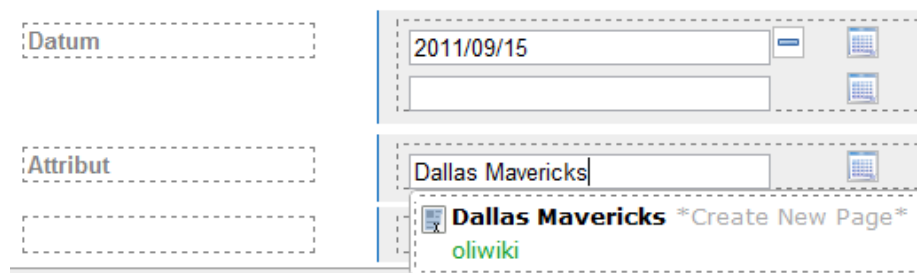


Abbildung 5.14.: Hinzufügen von Attributen

Nutzer hinzugefügt werden kann. Man kann erkennen, dass es dem Nutzer bereits gelungen ist, das Attribut mit der Bezeichnung „Datum“ zu erstellen und ihm einen entsprechenden Wert zuzuweisen. Dem Attribut „Attribut“ will der Nutzer in diesem Fall den Wert „Dallas Mavericks“ zuweisen. Da durch die Attribute auch auf andere Wikis, Wiki-Seiten oder Personen verlinkt werden kann, werden dem Nutzer in einer Vorschlag-Liste passende Inhalte zur Verlinkung vorgeschlagen. Kann das System jedoch keinen passenden Vorschlag liefern, so erscheint als einziger Eintrag ein Link, der die Erstellung einer neuen Wiki-Seite auslöst. Nutzer fühlten sich hierbei durch die einzige Aus-

5. Auswertung

wahrscheinlichkeit gezwungen, auf den vorgeschlagenen Eintrag zu klicken und waren sich dabei nicht bewusst, dass dadurch eine neue Wiki-Seite angelegt wird. Hinzukommend hatte der Großteil der Usability-Tester Probleme, angelegte Attribute wieder zu löschen. Attribute können auf zwei Arten gelöscht werden.

Zum einen lassen sich komplette Wertepaare löschen, indem der Nutzer den Editieren-Button, der sich unter anderem neben jedem Attribut einer Wiki-Seite befindet (siehe Abbildung 5.15), klickt und anschließend die Option „Delete“ wählt.



Abbildung 5.15.: Editieren von Attributen

Zum anderen können Attribute auch innerhalb der Editieren-Maske einer Wiki-Seite gelöscht werden. Dazu muss der Nutzer den zugewiesenen Wert des Attributs löschen und seine Änderung anschließend speichern. Attribute, denen kein Wert zugewiesen wurde, werden auf der Wiki-Seite danach nicht mehr angezeigt. Wurde vom Nutzer jedoch sowohl der linke, als auch der rechte Teil des Wertepaares entfernt, konnten die gemachten Änderungen nicht mehr gespeichert werden (siehe Abbildung 5.16). Das System meldete

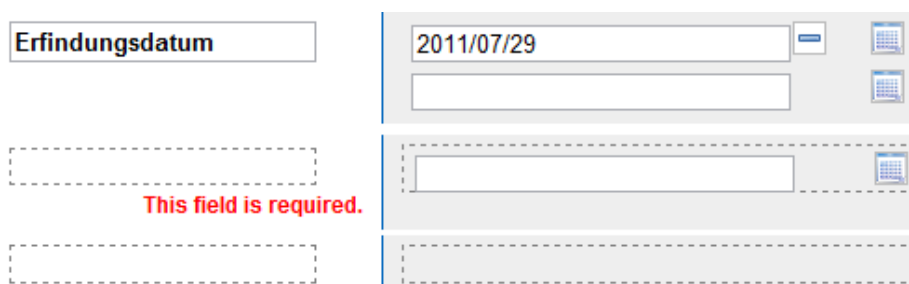


Abbildung 5.16.: Fehler beim Löschen eines Wertepaares

hierbei, dass kein vollkommen leerer Eintrag in den Attributen vorhanden sein darf. Die Nutzer waren deshalb gezwungen, den linken Teil des Paares immer ausgefüllt zu lassen. Die zuerst beschriebene Möglichkeit zum Löschen von Attributen wurde zudem nur von einem der Teilnehmer der Tests entdeckt. Die anderen Testpersonen waren somit nicht in der Lage, Attribute ganz zu entfernen.

Einer der Interviewpartner äußerte außerdem den Wunsch, Zeilenumbrüche oder Auflistungen bei der Definition von Attributen verwenden zu können. Bisher ist es nur möglich, unformatierten Text einzutragen.

Aus Tabelle 5.5 wird ersichtlich, dass Nutzer, die die erste der beschriebenen Löschoptionen nicht finden konnten, auf die Nutzung der zweiten Option beschränkt waren. Da die Nutzer bei der Verwendung dieser Möglichkeit jedoch auf einen Fehler getroffen sind, konnten sie die gewünschten Attribute nicht löschen. Auch hier führte der erste Usability-Fehler die Nutzer zu einem weiteren. Erweitert man nun die zweite Option um die Funktionsweise der ersten Funktion, so kann zumindest eines der Usability-Probleme vermieden werden. Es muss außerdem untersucht werden, ob die Erweiterung

der Attribut-Formatierungen technisch möglich ist und ob es sinnvoll ist, diese zu implementieren. Die Vorschlag-Liste sollte so angepasst werden, dass dem Nutzer sofort klar ist, dass durch einen Klick eine neue Wiki-Seite angelegt wird. Es wäre zudem sinnvoll, vor dem Erstellen der Seite ein Pop-up erscheinen zu lassen, durch das der Nutzer noch einmal explizit auf die durchgeführte Aktion hingewiesen wird und selbige nochmals bestätigt werden muss.

	TP1	TP2	TP3	TP4	TP5	IP1	IP2	IP3	IP4	IP5
Attribute hinzufügen	X	X	-	X	-	-	-	-	-	-
1. Lösch-Option	X	X	X	X	-	-	-	-	-	-
Attribute löschen	X	X	X	X	-	-	-	-	-	-
Attribut-Formatierung	-	-	-	-	-	-	X	-	-	-

Tabelle 5.5.: Usability-Probleme bei der Verwaltung von Attributen

5.6. Attachments

Bei der Verwaltung von Anhängen stießen sowohl die Interviewpartner, als auch die Usability-Tester auf verschiedene Probleme. Um eine Datei an ein Wiki oder eine Wiki-Seite anzuhängen, gibt es unterschiedliche Möglichkeiten. Der Nutzer kann zum Beispiel auf die Dateien über ein auf seinem PC eingebundenes SMB-Laufwerk zugreifen. Dadurch können von ihm Dateien einfach und bequem in seinem Explorer verwaltet und dabei hochgeladen, gelöscht oder geändert werden. Zusätzlich bietet aber auch das Webinterface verschiedene Wege, Anhänge hochzuladen und zu nutzen. Hierfür kann der Nutzer beispielsweise innerhalb der Editieren-Maske einer Wiki-Seite die Aktion „Insert a link“ aufrufen und sieht sich anschließend mit der in Abbildung 5.17 dargestellten Maske konfrontiert. Die Maske zeigt dem Nutzer unter anderem die Position des Ordners in dem er



Abbildung 5.17.: Hinzufügen eines Links

sich befindet, die bereits für eine Verlinkung zur Verfügung stehenden Dateien (in diesem

5. Auswertung

Fall „Einleitung.tex“) und lässt den Nutzer über den „Choose File“-Button weitere Dateien auswählen, die anschließend mit der Option „Upload“ in das Verzeichnis hochgeladen werden können. Um eine Datei zu verlinken, muss der Nutzer die entsprechende Datei anklicken. Diese Funktion wurde von zwei der fünf Usability-Testern nicht entdeckt. Die Testpersonen waren in der Lage, eine gewünschte Datei auszuwählen und hochzuladen, wussten aber nicht, dass die Datei nach dem Hochladen noch einmal explizit zur Verlinkung selektiert werden muss.

Da die Mehrzahl der Anhänge von den Interviewpartnern auch verlinkt werden mussten, nutzten die meisten auch diese Option, um Anhänge hochzuladen. Das Webinterface bietet jedoch noch einen weiteren Weg zur Verwaltung von Dateien. Nutzer können das Attachment-Verzeichnis des Wikis oder der Wiki-Seite auch direkt über einen Reiter aufrufen, der sich unter der „Breadcrumbs“-Navigation der Seite befindet (siehe Abbildung 5.18).

Dabei kann der Nutzer auf den Reiter klicken, einmalig ein Anhänges-Verzeichnis erstellen, und anschließend neue Ordner oder Dateien anlegen. Während einer der Interviewpartner, der seine Dateien normalerweise über ein SMB-Laufwerk verwaltet, nicht in der Lage war, diesen Reiter zu finden, hatten die Usability-Tester

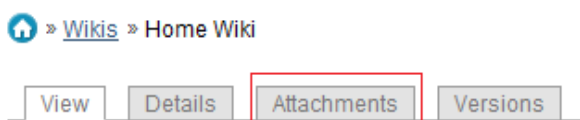


Abbildung 5.18.: Aufrufen des Attachment-Verzeichnisses

keine Probleme damit, ihn ausfindig zu machen. Klickt man auf den Reiter „Attachments“, ändert sich die Navigation der Seite (siehe Abbildung 5.19). Man kann erkennen, dass sich

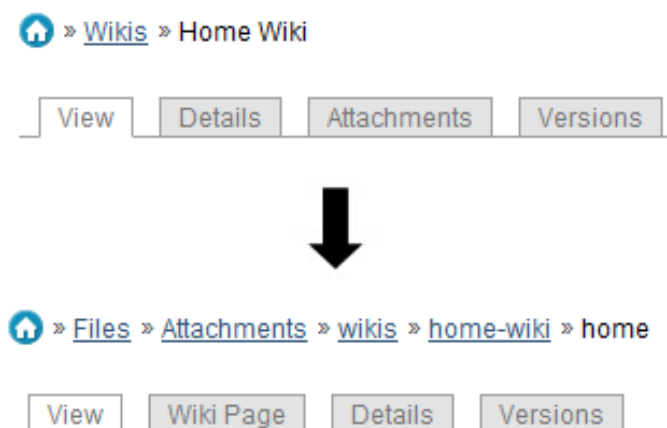


Abbildung 5.19.: Änderung der Navigation

der Nutzer nun nicht mehr innerhalb der Wikis, sondern innerhalb der Dateien (hier „Files“) befindet. Durch das Ändern der Navigation, hatten manche Nutzer Probleme damit, wieder auf die zuvor besuchte Wiki-Seite zu gelangen. Ihnen war nicht klar, dass der Rei-

ter „View“ in diesem Fall zur Darstellung des Attachments-Verzeichnisses benutzt wird und man den Reiter „Wiki Page“ wählen muss, um zur Ausgangsseite zurückzugelangen.

Während alle Usability-Tester in der Lage waren, eine Datei an eine Wiki-Seite anzuhängen, schilderten zwei der Interviewpartner, dass sie lange Zeit Probleme hatten, Dateien hochzuladen, weil sie die entsprechenden Funktionen nicht finden konnten. Zur Lösung des Problems waren sie dabei zum Beispiel auch auf die Hilfe der Entwickler angewiesen.

Um hochgeladene Dateien wieder zu löschen, gibt es zwei Wege. Einerseits kann man über das SMB-Laufwerk gewünschte Dateien ganz einfach entfernen, andererseits können angehängte Dateien auch über das Webinterface gelöscht werden. Dazu muss der Nutzer das Attachment-Verzeichnis des Wikis oder der Wiki-Seite öffnen, in dem sich die zu löschende Datei befindet. Anschließend müssen die Eigenschaften der Datei durch einen Klick auf das blaue „i“-Symbol dargestellt in Abbildung 5.20 - geöffnet werden, um die Datei schließlich mit einem Klick auf den „Delete“-Button zu entfernen. Interviewpartner, die ein SMB-Laufwerk nutzten, hatten keine Probleme bei der Löschung von Dateien, wurden aber während der Durchführung des Interviews gebeten, eine Datei über das Webinterface zu löschen. Dieser Prozess konnte jedoch nur von einem Interviewpartner und einem Usability-Tester erfolgreich durchgeführt werden. Nutzer, die Dateien über die Editieren-Maske der Wiki-Seite verlinkten, nahmen zum Beispiel an, dass die Löschung der Verlinkung auch zur Löschung der Datei führt. Andere Testpersonen waren zwar in der Lage, das entsprechende Attachment-Verzeichnis zu öffnen, wollten dann aber bereits auf dieser Seite den „Delete“-Button drücken, der jedoch die Löschung des gesamten Verzeichnisses auslöst. Den Testpersonen war nicht klar, wie die zu löschende Datei selektiert werden kann.

Des Weiteren werden Dateien standardmäßig im aktiven Fenster des Browsers geöffnet. Dies hat während der Usability-Tests zum Beispiel dazu geführt, dass Nutzer, die davon ausgegangen waren, dass die Datei in einem neuen Tab des Browsers geöffnet wurde, das aktive Tab geschlossen haben und somit ungewollt nicht mehr zum System zurückkehren konnten.

Attachments Folder

Tags: no tags assigned

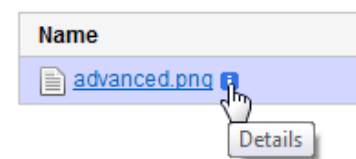


Abbildung 5.20.: Aufruf der zu löschenden Datei

	TP1	TP2	TP3	TP4	TP5	IP1	IP2	IP3	IP4	IP5
Navigation	X	-	-	-	X	-	-	-	-	-
Verlinkung	X	-	X	-	-	-	-	-	-	-
Attachment-Reiter	-	-	-	-	-	-	-	-	X	-
Datei hochladen	-	-	-	-	-	X	-	-	X	-
Datei löschen	X	X	X	-	X	X	-	X	X	X
aktives Tab	-	X	X	-	-	-	-	X	-	-

Tabelle 5.6.: Usability-Probleme bei der Verwaltung von Anhängen

Tabelle 5.6 zeigt, dass Nutzer vor allem bei der Löschung von Dateien über das Webinterface Probleme hatten. Dieser Prozess könnte vereinfacht werden, indem man den Prozessschritt zur Auswahl der zu löschenden Datei ändert. Statt auf das blaue „i“-Symbol klicken zu müssen, könnte eine Checkbox neben jeder Datei platziert werden, die bei Bedarf selektiert werden kann. Somit könnten auch mehrere Dateien auf einmal gelöscht werden. Um zu vermeiden, dass Nutzer nach dem Öffnen einer Datei das falsche Fenster schließen, könnten Dateien standardmäßig in einem neuen Browser-Tab geöffnet werden. Da Dateien und Wikis getrennt voneinander betrachtet werden müssen, ist es sehr schwer, eine Änderung der Navigation nach der Wahl des Attachment-Reiters zu umgehen. Um die Nutzer jedoch so wenig wie möglich zu verwirren, könnte die Anordnung der Reiter beibehalten werden. Dazu müssten innerhalb der Datei-Umgebung nur die beiden Reiter „Wiki-Page“ und „Details“ miteinander getauscht werden.

5.7. Suche



Abbildung 5.21.: Intranet-Suchfunktion

Die Suchfunktion ist die einzige Funktion von Tricia, die durch alle drei Usability-Evaluationsmethoden untersucht wurde. Der Großteil der Interviewpartner berichtete, dass sie die Suchfunktion so gut wie nicht benutzen. Dies lässt sich auf zwei Gründe zurückführen. Zum einen ist die Darstellung der Suchfunktion auf der Intranet-Seite sehr unterschiedlich zur Darstellung auf der Lehrstuhl-Seite. Während die Suchfunktion sich in der rechten oberen Ecke des Intranets befindet und zudem um ein Vielfaches kleiner als die Suchleiste des Lehrstuhls ist (siehe Abbildung 5.21), wurde die Suchleiste der sebis-Seite als großes Eingabefeld in der Seitenmitte platziert. Zum anderen bezeichneten sich die Interviewpartner selbst als „Inhalt-Verfasser“ und nicht als „Inhalt-Sucher“. Die gewünschten Seiten könnten von ihnen ohne Probleme und schnell über die Navigation mit wenigen Klicks erreicht werden. Auf Nachfrage hatten die Interviewten auch keine Probleme, die beiden Optionen der Suchfunktion - Quick-Search und Präfix-Suche - zu identifizieren und ohne Fehler zu nutzen.

Die Logfile-Analyse wurde, wie in Kapitel 4.3 bereits beschrieben, in zwei Teilen durchgeführt. Die Ergebnisse werden nun in den beiden folgenden Abschnitten der Arbeit näher erläutert.

5.7.1. Quick-Search

Die Quick-Search Option der Suche wurde mit Piwik analysiert. Abbildung 5.22 stellt einen Überblick über die Nutzung der Quick-Search dar. Die Auswertung zeigt, mit welcher Häufigkeit welche Seiten aufgerufen wurden. Als unterschiedliche Seiten sind in diesem Fall die unterschiedlichen Wahlmöglichkeiten der Vorschlag-Liste zu verstehen. Es ist zu erkennen, dass die Quick-Search in einem Zeitraum von einer Woche 1005 Mal benutzt

Page URL	Pageviews	Unique Pageviews	Bounce Rate	Avg. time on page	Exit rate
quicksearch	1913	1486	25%	1 min 4s	45%
/used	1005	623	20%	20s	26%
1	358	342	74%	2 min 3s	61%
submit	218	199	44%	1 min 0s	57%
2	191	190	83%	1 min 7s	63%
3	68	61	67%	1 min 23s	66%
4	33	33	0%	3 min 11s	48%
5	20	20	0%	2 min 13s	40%
6	18	16	0%	1 min 41s	38%
undefined	2	2	0%	4 min 34s	50%

Abbildung 5.22.: Überblick Piwik-Analyse

wurde. Eine Nutzung wird bereits durch das Aktivieren des Eingabefelds gezählt. Nachdem der Nutzer das Feld aktiviert hat, beginnt er mit der Eingabe eines Begriffs. Daraufhin werden ihm nach der Eingabe jedes Buchstabens passende Inhalte in einer Liste angeboten, die er durch einen Klick aufrufen kann. Nutzt der Suchende die Quick-Navigation nicht, sondern betätigt die Enter-Taste oder klickt auf das Lupensymbol, so wird ein „submit“ gezählt und eine Ergebnisseite angezeigt. Bei 1005 Benutzungen wurde demnach 218 Mal eine Präfix-Suche angestoßen. Das bedeutet, dass knapp ein Fünftel der Nutzer entweder den gesuchten Inhalt in der Vorschlag-Liste nicht vorgefunden hat, oder aber die Quick-Navigation einfach nicht benutzen wollte. Betrachtet man die Aufrufe der verschiedenen Auswahlmöglichkeiten, so ist zu erkennen, dass der erste Vorschlag der Liste auch mit Abstand am häufigsten angeklickt wurde (358 Mal). Während Option 2 noch 191 Mal ausgewählt wurde, klickten die Nutzer nur noch 68 Mal auf den 3. Vorschlag. Vorschläge 4 bis 6 haben kontinuierlich weniger Zugriffe. Diese Zahlen lassen eine positive Bilanz über den Suchalgorithmus ziehen. Da die ersten Vorschläge am häufigsten ausgewählt werden und die Zahl der Zugriffe stetig abnimmt, werden den Nutzern folglich die gesuchten Inhalte weiter oben angezeigt. Addiert man nun die Zahlen der Quick-Navigation und der Präfix-Suche, so erhält man in diesem Fall 908 Aufrufe. Bei den weiter oben genannten 1005 Nutzungen bedeutet dies, dass das Eingabefeld zwar 97 Mal ausgewählt, aber keine Suche durchgeführt wurde. Dies kann verschiedene Gründe haben. Einerseits kann es sein, dass der Nutzer während der Eingabe den gesuchten Inhalt bereits an einer anderen Stelle der Seite entdeckt hat und deshalb die Suche nicht vollendet und auf einen anderen Inhalt auf der Seite klickt. Andererseits kann diese Zahl auch durch eine automatische Aktivierung des Suchfeldes in die Höhe getrieben werden. Dies kann beispielsweise geschehen, wenn die Eingabemaske automatisch angewählt wird, nachdem der Nutzer auf die Startseite der Lehrstuhl-Website navigiert hat.

5. Auswertung

Zur genaueren Analyse werden nun die Zahlen der Bounce- und Exit-Rate genauer betrachtet. Während die Bounce-Rate den Prozentteil der Nutzer angibt, die nach ausschließlichem Besuch dieser Seite die Website ganz verlassen, gibt die Exit-Rate den Anteil der Nutzer an, die nach dem Besuch der Seite die Website verlassen. Im ersten Moment denkt man häufig, dass die beiden Zahlen eigentlich das gleiche ausdrücken. Nutzer, die aber von der Seite „bouncen“, haben außer dieser Seite keine einzige andere Seite der Website besucht. Sie sind zum Beispiel durch eine Suchmaschine auf diese Seite gelangt und verlassen diese umgehend wieder, ohne auf eine andere Seite der Website zu navigieren. Nutzer, die zu einer Erhöhung der Exit-Rate führen, verlassen zwar die Website auf dieser Seite, sind aber durch andere Inhalte der Website auf diese Seite gestoßen. Eine sehr hohe Bounce- und Exit-Rate können demnach darauf schließen lassen, dass die Nutzer zum Beispiel die gesuchten Inhalte auf der Seite nicht finden. Es kann aber auch sein, dass Nutzer die Seite einfach nicht ansprechend finden und sie deshalb verlassen. Da hier jedoch eine Suchfunktion untersucht wird, sind diese Zahlen unter einem anderen Blickwinkel zu betrachten. Eine hohe Bounce- und Exit-Rate können demnach auch zustande kommen, wenn der Suchende den gewünschten Inhalt gefunden hat und es für ihn keinen weiteren Grund gibt, auf der Website zu bleiben. Deshalb muss in diesem Fall auch die Zeit, die der Nutzer auf den Seiten verbracht hat, in Betracht gezogen werden. Die Auswertung zeigt, dass die Bounce- und Exit-Rate der ersten drei Vorschläge sehr hoch sind. Im ersten Moment ist dies ein negatives Zeichen. Betrachtet man jedoch die auf den Seiten verbrachte Zeit, die zwischen einer und zwei Minuten schwankt, so lässt dies darauf schließen, dass die Suchenden nicht nur kurz den Inhalt der Seite überflogen haben, sondern eine gewisse Zeit auf der Seite beschäftigt waren. Auch die durchschnittliche Verweildauer von knapp über drei Minuten auf der vierten Auswahl der Liste, lässt darauf schließen, dass die Nutzer sich längere Zeit auf der angeklickten Seite aufgehalten haben. Die hohe Bounce- und Exit-Rate in Verbindung mit der durchschnittlich höheren Verweildauer sind somit ein Zeichen, dass die Nutzer auf der ausgewählten Seite fündig wurden und danach keinen Grund mehr hatten, auf der Website zu bleiben.

Page URL	Pageviews	Unique Pageviews	Bounce Rate	Avg. time on page	Exit rate
quicksearch	1913	1486	25%	1 min 4s	45%
/used	1005	623	20%	20s	26%
1	358	342	74%	2 min 3s	61%
4	58	57	80%	1 min 25s	58%
3	58	51	67%	2 min 47s	86%
5	45	44	80%	59s	64%
6	39	39	100%	1 min 37s	69%

Abbildung 5.23.: Piwik - Auswahl des ersten Vorschlags

Abbildung 5.23 zeigt einen Ausschnitt darüber, wie viele Buchstaben ein Nutzer einge-

ben musste, bis er in diesem Fall den ersten Eintrag der Vorschlag-Liste angeklickt hat. Die meisten Aufrufe wurden dabei mit 58 Mal nach drei beziehungsweise vier Buchstaben getätigt. Die hohe Bounce- und Exit-Rate und die lange Verweildauer lassen auch hier darauf schließen, dass die vorgeschlagenen Inhalte auch die gesuchten waren. Außerdem ist zu erkennen, dass bei steigender Buchstabenanzahl die Seitenaufrufe sinken. Dies ist ein Hinweis darauf, dass bereits wenige Buchstaben ausgereicht haben, bis die passenden Einträge in der Liste vorhanden waren. Zum Vergleich soll nun ein Ausschnitt über

☰ 2	191	190	83%	1 min 7s	63%
⊕ 4	30	30	100%	1 min 24s	63%
⊕ 5	24	24	0%	1 min 57s	38%
⊕ 12	21	21	100%	12s	76%
⊕ 10	16	16	0%	21s	69%
⊕ 11	13	13	0%	2 min 23s	69%

Abbildung 5.24.: Piwik - Auswahl des zweiten Vorschlags

die Auswertung des zweiten Eintrags der Liste untersucht werden (siehe Abbildung 5.24). Man kann erkennen, dass auch der zweite Eintrag am häufigsten bereits nach vier oder fünf eingegebenen Buchstaben ausgewählt wird. Hier ist jedoch interessant, dass danach 12, 10 und 11 eingetragene Buchstaben die höchsten Aufrufe triggern. Diese hohe Anzahl an Buchstaben kommt hierbei durch das Kopieren von ganzen Sätzen oder langen Begriffen zustande. Diese werden dabei von den Nutzern einfach in die Suchmaske kopiert und die Vorschlag-Liste anschließend nach passenden Inhalten durchsucht. Hinzukommend ist zu sehen, dass die Durchschnittszeit, die die Nutzer auf diesen Seiten verbringen, nach 12 und 10 eingegebenen Buchstaben bei 12 beziehungsweise 21 Sekunden liegt. Das ist ein Zeichen dafür, dass die gewünschten Inhalte auf der aufgerufenen Seite wahrscheinlich nicht vorgefunden werden konnten.

Die Ergebnisse erstrecken sich auch über die weiteren Vorschläge, die im Vergleich zu den ersten beiden Vorschlägen eher selten ausgewählt wurden. Es ist jedoch hervorzuheben, dass einer Auswahl eines weiter unten auftauchenden Eintrags eine immer länger werdende Eingabe von Buchstaben vorausging. Auch das ist ein Hinweis darauf, dass den Nutzern bereits nach wenigen Eingaben passende Inhalte als erste Einträge der Vorschlag-Liste angeboten wurden. Die Logfile-Analyse der Quick-Search konnte somit keine Usability-Probleme der Nutzer aufdecken. Es wurde vielmehr gezeigt, dass der Suchalgorithmus gut funktioniert und den Nutzern die gesuchten Inhalte nach wenigen Eingaben und mit der richtigen Priorität angezeigt werden.

Die Usability-Tester hatten unter anderem die Aufgabe, nach bestimmten Inhalten zu suchen. Dabei fiel bei der Verwendung der Quick-Search nur ein Problem auf. Suchten die Nutzer nach einem Tag, so wurden ihnen verschiedene Inhalte in der Vorschlag-Liste angezeigt. Abbildung 5.25 zeigt zur Veranschaulichung die Suche nach dem Wort **oder** Tag „Usability“.

Zwei Testpersonen konnten hierbei nicht sagen, warum der zweite Eintrag in der Vor-

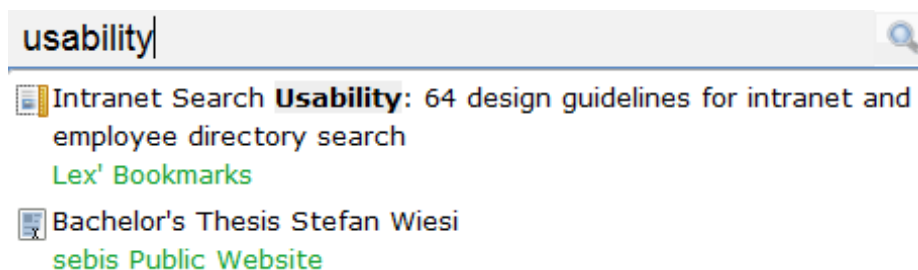


Abbildung 5.25.: Suche von Tags

schlag-Liste erscheint. Während im ersten Eintrag das Vorkommen des Wortes „Usability“ hervorgehoben wird, gibt es im zweiten Eintrag keinen Hinweis auf das eingegebene Suchwort. In diesem Fall wird der Eintrag jedoch angezeigt, weil für diese Wiki-Seite unter anderem das Tag „Usability“ hinterlegt wurde.

5.7.2. Präfix-Suche

Bevor die Usability-Evaluationen durchgeführt wurden, wurde aus der seit langem vorhandenen Volltext-Suche eine Präfix-Suche. Wie im vorherigen Kapitel bereits beschrieben, führt der Nutzer eine Präfix-Suche aus, indem er einen Begriff in die Suchleiste eingibt und entweder „Enter“ betätigt, oder das Lupen-Symbol anklickt. Daraufhin werden dem Nutzer passende Inhalte, die zuvor vereinzelt auch schon in der Vorschlag-Liste aufgetaucht sind, detaillierter auf einer Ergebnisseite angezeigt. Dort hat der Nutzer zusätzlich die Möglichkeit, die Ergebnisseite durch die Verwendung verschiedener Filter zu verfeinern. Bevor die Ergebnisse der Parameter-Auswertung genauer betrachtet werden, muss gesagt werden, dass die Umstellung von einer Volltext- auf eine Präfix-Suche mit großer Wahrscheinlichkeit zur Vermeidung von Usability-Problemen geführt hat. Aus der Piwik-Analyse der Quick-Search geht unter anderem hervor, dass Nutzer oft nur Teile eines Wortes eintippen und dann eine Präfix-Suche triggern. Bei einer Volltext-Suche hätte dies zur Folge, dass dem Nutzer so gut wie keine passenden Ergebnisse angezeigt werden, obwohl ihm bereits in der Quick-Search-Leiste, die auf einer Präfix-Suche basiert, passende Einträge angeboten wurden. Dies hätte zu einer Verwirrung der Nutzer führen können, kann aber nun nicht mehr vorkommen.

Wie in Kapitel 4.3 bereits detailliert erläutert, können die Eingaben und angewendeten Filter der Präfix-Suche durch in der URL gespeicherte Parameter untersucht werden. Dazu wurde jeder Eintrag der Logfiles im Zeitraum von einer Woche, der die Suchfunktion betrifft, auf enthaltene Parameter untersucht. Hinzukommend sei zu erwähnen, dass die Filter fast beliebig miteinander kombiniert werden können, jedoch nur das Must-Parameter mit sich selbst kombiniert werden kann. Das bedeutet, dass mehrere Tags hintereinander zur Einschränkung des Ergebnisses ausgewählt werden können. Tabelle 5.7 zeigt die verschiedenen Kombinationen von jeweils zwei Parametern. Da das Must-Parameter mehrmals in der URL vorkommen kann, wird zum Beispiel ein Aufruf der Parameter Query, Must und Must als Aufruf von Query und Must gewertet. Die Kombination Must und Must zeigt folglich die Auswahl von zwei oder mehreren Tags. Da die Kombination von drei oder mehreren Filtern eher selten vorkommt, wird diese im weiteren Verlauf des Ka-

pitels gesondert betrachtet. Das Vorkommen von nur einem Parameter wird in der obersten Zeile dargestellt. Das zu untersuchende Logfile enthielt etwa 2100 Einträge. An der in

	Query	Must	Kinds	Spaces	TypeTags	AttributeName	Σ
-	444	904	4	10	4	-	1366
Query	-	14	13	15	5	-	47
Must	14	245	38	10	11	2	320
Kinds	13	38	-	1	36	-	88
Spaces	15	10	1	-	6	-	32
TypeTags	5	11	36	6	-	-	58
AttributeName	-	2	-	-	-	-	2
Σ	491	1224	92	42	62	2	1913

Tabelle 5.7.: Parameter-Kombination

der Tabelle dargestellten Summe von 1913 Kombinationen ist zu erkennen, dass über 90% der durchgeführten Suchen maximal zwei Parameter enthielten. Betrachtet man nun die Zahl der einzeln aufgerufenen Parameter von 1366, so zeigt sich, dass etwa 65% der Anfragen mit nur einem Parameter ausgeführt wurden. Dies lässt darauf schließen, dass der Großteil der Nutzer keinen Filter zur Einschränkung der Suchergebnisse wählt. Genauer gesagt wurden 444 Anfragen getätigt, die nur das Suchwort enthielten. Mehr als doppelt so oft wurde jedoch ein einzelnes Tag angeklickt. Die Auswahl eines einzelnen Tags ohne davor ein Suchwort eingegeben zu haben lässt darauf schließen, dass der Nutzer seine ausgeführte Suche nicht einschränken wollte, sondern vielmehr sich passende Inhalte zu einem Tag hat anzeigen lassen. Die einzelne Auswahl der anderen Parameter ist deshalb so gering, weil diese Filter-Optionen eigentlich nur zur Einschränkung der Ergebnisse genutzt werden und somit meist in Verbindung mit einem Suchwort auftreten.

Bei der Untersuchung der verschiedenen Kombinationen tritt nur eine Zahl besonders in den Vordergrund. Die Kombination mehrerer Tags zur Navigation - in diesem Fall nicht zur Einschränkung der bereits ausgeführten Suche - wurde 245 mal durchgeführt. Dies entspricht einem Gesamtanteil von knapp über 10%. Die anderen Kombinationen treten hingegen in sehr geringer Anzahl auf. Hier gilt es, in weiteren Analysen zum Beispiel zu zeigen, warum diese Kombinationen so selten auftreten. Dies kann einerseits daran liegen, dass die gesuchten Inhalte bereits auf den ersten Seiten gefunden werden, andererseits aber auch ein Hinweis darauf sein, dass die Nutzer nicht verstehen, welche Einschränkungen durch die einzelnen Filter erzielt werden können. Des Weiteren ist zu erwähnen, dass Tags einzeln und in Kombination mit anderen Filtern am häufigsten (1224 Mal) verwendet wurden. Dabei ist jedoch besonders hervorzuheben, dass nur 14 der Suchanfragen anschließend durch die Auswahl von Tags weiter eingeschränkt wurden. Hinzukommend gaben Nutzer 92 Mal an, von welcher Art das Suchergebnis ist. Dagegen wurde nur 42 Mal ausgewählt, in welchem Bereich sich die gesuchten Inhalte befinden. Außerdem ist zu erwähnen, dass der Filter für vorhandene Attribute nur zweimal - in Verbindung mit dem Must-Parameter - ausgewählt wurde. Das zustande kommen dieser Kombination ist jedoch sehr fraglich, da das AttributeName-Parameter eigentlich nur in Verbindung mit dem TypeTags-Parameter gewählt werden kann.

Tabelle 5.8 zeigt die Auftritts-Häufigkeit von Kombinationen von drei oder mehr Parametern, die mehr als 9 Mal verwendet wurden:

	Anzahl
Must+Kinds+Spaces	103
Must+TypeTag+AttributeName	55
Spaces+TypeTag+AttributeName	13
Must+Spaces+TypeTag+AttributeName	10
Andere	38
Σ	219

Tabelle 5.8.: Kombination von drei oder mehr Parametern

Hierbei sind verschiedene Aspekte hervorzuheben. Zum einen wurde die Kombination der Must-, Kinds- und Spaces-Parameter 103 Mal angewandt. Dies ist die unter allen Kombinationen am zweithäufigsten genutzte Verwendung von Filter-Optionen. Es kann dabei unter Umständen sein, dass die hohe Zahl auf einen Power-User zurückzuführen ist, der durch die Verwendung genau dieser Kombination gesuchte Inhalte finden kann. Zum anderen kann die Zahl aber auch durch einen Entwickler manipuliert worden sein, der mehrmals die Funktionsweise seiner Implementierung getestet hat und dessen Einträge bei der Säuberung des Logfiles nicht als Entwickler-Einträge identifiziert werden konnten. Des Weiteren enthalten drei der vier dargestellten Kombinationen das AttributeName-Parameter. Dieser Parameter tritt folglich zum Großteil erst bei einer Verwendung von drei oder mehr Parametern auf, was auch auf die bereits erläuterten Voraussetzungen zurückgeführt werden kann. Zählt man die in Tabelle 5.7 und 5.8 aufgezählten Anfragen zusammen, so erhält man eine Gesamtzahl von 2132 Anfragen.

Aus der Logfile-Auswertung der Präfix-Suche können somit keine eindeutigen Schlüsse auf bestehende Usability-Probleme gezogen werden. Zusammenfassend kann jedoch gesagt werden, dass Filteroptionen selten miteinander kombiniert werden. Hinzukommend ist die Verwendung von Tags sehr beliebt. Dabei werden einzelne Tags jedoch häufiger angeklickt, als sie miteinander kombiniert werden. Ziel der zur Verfügung stehenden Tag-Cloud ist es jedoch, die Kombinierung verschiedener Tags zum Finden von gesuchten Inhalten zu benutzen. Auch hier ist in weiteren Analysen zu zeigen, wie dieser Prozess für die Nutzer besser dargestellt oder verbessert werden kann.

Teilnehmer der Usability-Tests wurden gebeten, verschiedene Suchanfragen durchzuführen. Da aus der Logfile-Analyse bereits bekannt war, dass Nutzer nur sehr selten die Filter-Optionen nutzen, wurden die Testpersonen nach der Bedeutung der verschiedenen Filter gefragt. Keiner der Teilnehmer hatte dabei Probleme, die richtige Auswirkung der Filter zu beschreiben. Auf die Frage, ob sie die Filter-Optionen bei ihrer Suche nutzen würden, sagte die Mehrzahl, dass sie sie nur nutzen würden, falls die gesuchten Inhalte nicht auf den ersten beiden Seiten der Ergebnis-Seite angezeigt werden. Die Testpersonen wurden außerdem dazu aufgefordert, bestimmte Inhalte über die Verwendung der Tag-Cloud zu finden. Diese arbeitet mit logischen UND-Verknüpfungen. Wählt man ein Tag aus, wird die Auswahl der Ergebnisse auf alle Inhalte eingeschränkt, die mit dem ausgewählten Tag ausgestattet sind. Daraufhin verändert sich die dargestellte Tag-Cloud und

bietet dem Nutzer weitere Möglichkeiten zur genaueren Eingrenzung. Wählt der Nutzer aus der neuen Tag-Cloud ein weiteres Tag, werden nach einer Aktualisierung alle Inhalte angezeigt, die sowohl das erste, als auch das zweite Tag beinhalten. Die bereits ausgewählten Tags werden dem Nutzer oberhalb der Tag-Cloud angezeigt (siehe Abbildung 5.26). Zwei der fünf Usability-Tester gingen in diesem Fall davon aus, dass die Auswahl des ersten Tags keinen Einfluss auf die angezeigte Auswahl des zweiten Tags hat. Ihnen war somit nicht klar, dass Tags, die nacheinander ausgewählt wurden, ohne den Zurück-Knopf zu betätigen, mit einem logischen UND verknüpft werden.

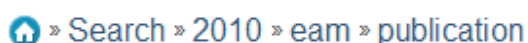


Abbildung 5.26.: Darstellung bereits ausgewählter Tags aus der Tag-Cloud

Die in Kapitel 5.7.1 und 5.7.2 beschriebenen Probleme werden in Tabelle 5.9 zusammengefasst. Um den Nutzern zu verdeutlichen, dass ein bestimmter Such-Eintrag wegen dem Vorhandensein eines Tags erscheint, könnten zum Beispiel passende Tags in der Vorschlagsliste angezeigt und farbig hinterlegt werden. Auf diese Weise sollte den Nutzern sofort klar sein, warum dieser Eintrag vom System vorgeschlagen wird. Die Darstellung der UND-Verknüpfung könnte zudem testweise verändert werden. Dadurch kann in weiteren Analysen untersucht werden, ob eine Änderung der Darstellung die Verwendung der Tag-Cloud für die Mehrzahl der Nutzer besser verständlich macht, oder ob die Fehlinterpretation der Usability-Tester nur ein Einzelfall war.

	TP1	TP2	TP3	TP4	TP5	IP1	IP2	IP3	IP4	IP5
Fehlender Tag-Hinweis	-	X	-	X	-	-	-	-	-	-
UND-Verknüpfung	-	-	X	X	-	-	-	-	-	-

Tabelle 5.9.: Usability-Probleme bei der Nutzung der Suchfunktion

6. Globale Betrachtung der gefundenen Usability-Probleme

Kapitel 6 betrachtet die in Kapitel 5 beschriebenen Usability-Probleme in einem größeren Kontext. Dabei soll zum Beispiel untersucht werden, welche Arten von Problemen durch die jeweiligen Untersuchungsmethoden gefunden werden konnten. Des Weiteren soll gezeigt werden, in welchem Maße sich die verschiedenen Methoden gegenseitig unterstützen und ergänzen können. Die Anzahl gefundener Probleme zu bestimmten Zeitpunkten der Untersuchungen gibt außerdem an, wie viele Testpersonen notwendig sind, um den Großteil aller Probleme aufzudecken. Zu einer besseren Veranschaulichung werden alle Probleme in Tabelle 6.1 auf Seite 61 zusammengefasst. Die Logfile-Analyse ist dabei kein Teil der Tabelle, da sie keine eindeutigen Usability-Probleme aufgedeckt hat, sondern nur Hinweise auf möglicherweise bestehende Probleme liefern konnte.

Die Tabelle beinhaltet insgesamt 30 gefundene Probleme, wobei bei der Erstellung und Verwaltung von **Wiki-Seiten** mit einer Anzahl von acht Problemen die meisten Fehler aufgedeckt werden konnten. Im Durchschnitt wurden folglich in etwa vier Probleme je untersuchter Funktion oder Bereich gefunden. Dabei konnte durch die Interviewpartner auf 14 Usability-Probleme - knapp unter 50% aller Fehler - hingewiesen werden. Im Gegensatz dazu, war es den Usability-Testern möglich, 26 der 30 Probleme (87%) zu identifizieren, wovon 16 Probleme durch die Usability-Interviews noch nicht bekannt waren. Die Usability-Tester entdeckten somit mehr neue Probleme, als zunächst durch die Interviewpartner festgestellt werden konnten. Abbildung 6.1 stellt die Anzahl gefundener Probleme

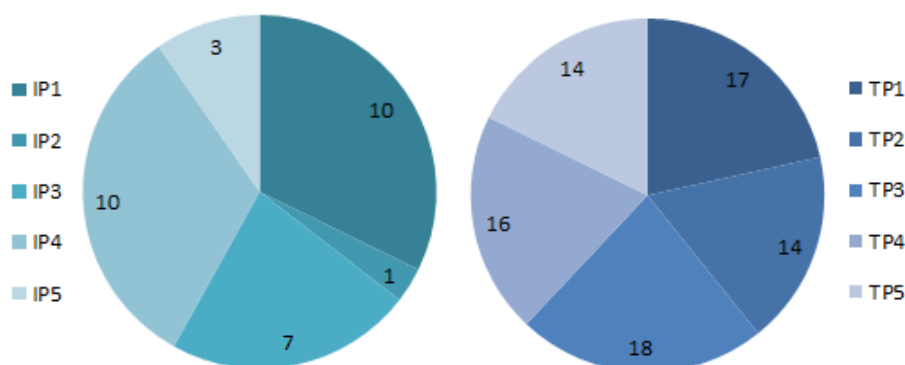


Abbildung 6.1.: Anzahl gefundener Probleme je Testperson

je Testperson dar. Hierbei ist vor allem zu erkennen, dass das Maximum und Minimum an aufgedeckten Fehlern bei den Interviewpartnern sehr weit voneinander entfernt liegen, während die Anzahl an Funden der Usability-Tester nur in geringem Maße voneinander abweichen. Dieses Phänomen lässt sich wie folgt erklären. Obwohl alle Usability-Tester

6. Globale Betrachtung der gefundenen Usability-Probleme

unterschiedliche berufliche Hintergründe und verschiedene Vorkenntnisse im Umgang mit Systemen wie Tricia hatten, war ihnen jedoch die Funktionsweise des Systems komplett neu. Einer der fünf Interviewpartner hatte hingegen einen Informatik-Hintergrund und arbeitete schon seit längerer Zeit mit dem System. Wie aus der zusammenfassenden Tabelle hervorgeht, hatte diese Testperson keinerlei Probleme mit der Handhabung des Systems, sondern fühlte sich lediglich durch das Fehlen einer Formatierungs-Option in seinem Handeln eingeschränkt. Hinzukommend war eine der Interviewten eine Power-Userin, die das System täglich nutzte, und somit mit den Funktionen sehr vertraut war. Hierbei wird deutlich, dass je regelmäßiger man ein System nutzt, desto weniger Probleme hat man im Umgang mit den einzelnen Funktionen. Das liegt vor allem auch daran, dass Power-User, die auf Probleme treffen, sich Umwege suchen, um dennoch das gewünschte Ziel zu erreichen. Nachdem dieser Umweg eine Weile genutzt worden ist, wird das Problem von nun an nicht mehr als solches erkannt und taucht deshalb auch eher selten in Usability-Interviews auf. Interviewpartner, die das System zum Beispiel wöchentlich oder monatlich nutzten, konnten sich an viel mehr Probleme erinnern. Obwohl manche Testpersonen bereits Antworten und Lösungen auf die aufgetretenen Probleme hatten, stolperten sie immer wieder über die gleichen Fehler. Zusammenfassend bedeutet dies jedoch nicht, dass man nur unerfahrene Nutzer interviewen sollte. Zu Vergleichszwecken ist es daher zu empfehlen, sowohl erfahrene, als auch Benutzer des Systems, die weniger Erfahrung mitbringen, in Usability-Interviews zu befragen.

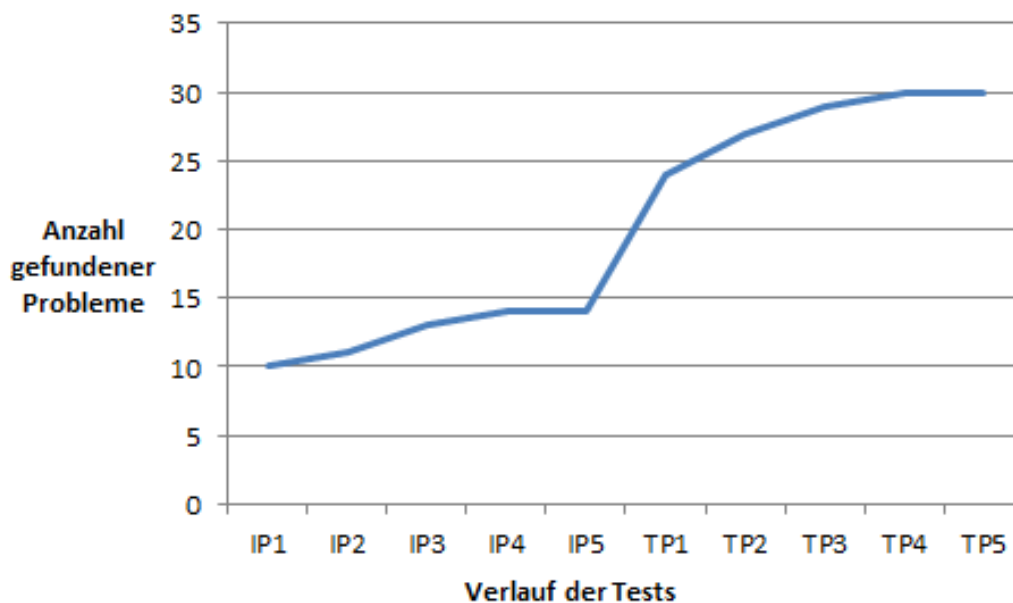


Abbildung 6.2.: Anzahl gefundener Probleme während des Testverlaufs

Abbildung 6.2 zeigt die Anzahl gefundener Probleme im zeitlichen Verlauf der Usability-Interviews und -Tests. Daraus wird ersichtlich, dass der erste Interviewpartner bereits ein Drittel aller Probleme identifizieren konnte. Im weiteren Verlauf der Interviews wurden jedoch nur noch vier weitere Probleme erkannt. Es ist daher anzunehmen, dass sich die Anzahl der entdeckten Probleme mit der Durchführung weiterer Interviews nicht mehr

signifikant steigern lässt. Der erste Usability-Tester stieß hingegen auf 10 neue Probleme. Aber auch im zweiten Teil des Graphen ist zu erkennen, dass sich die Fehleranzahl an einem bestimmten Wert angleicht. Auch hier ist zu vermuten, dass ein zusätzlicher Mehraufwand nicht den gewünschten Ertrag bringen würde. Der Verlauf des Graphen zeigt jedoch deutlich, dass die Kombination verschiedener Usability-Evaluationsmethoden einen sehr großen Mehrwert mit sich bringt. Die zusätzliche Durchführung der Usability-Tests führte zu einer doppelt so hohen Anzahl von Usability-Problemen. Obwohl allein durch die Usability-Tests 26 der 30 Probleme aufgedeckt werden konnten, dürfen die Usability-Interviews in dieser Rechnung nicht vernachlässigt werden. Durch die Durchführung der Interviews konnten einige Funktionen und Bereiche von Tricia identifiziert werden, die später auch den Usability-Testern zur Evaluation vorgelegt wurden. Außerdem war dadurch möglich zu zeigen, dass einige der Probleme sowohl bei regelmäßigen Benutzern, als auch neuen Nutzern des Systems auftraten.

Nun gilt es genauer zu untersuchen, welche Problemarten von welchen Methoden besser gefunden werden konnten. Dabei stechen vor allem bei den Interviewpartnern drei Bereiche hervor, bei denen so gut wie keine Probleme identifiziert wurden. Es handelt sich dabei um die Bereiche Registrierung und Login, Verwaltung von Attributen, und Verwendung der Suchfunktion. Da während des Login-Prozesses Fehler meist nur beim ersten Mal auftreten, kommen diese in Usability-Interviews nur sehr selten zur Sprache. Das liegt unter anderem auch daran, dass sich die Nutzer meist nur an die Probleme erinnern, auf die sie regelmäßig stoßen. Obwohl die Vergabe von Attributen und die Nutzung der Suchfunktion in den Interviews explizit angesprochen wurden, wurden in diesen beiden Bereichen keine Probleme aufgedeckt. Im Gegensatz dazu konnten durch die Usability-Tests einige Probleme dieser Funktionen ermittelt werden. Hier wird deutlich, dass Funktionen, die von den Interviewten wenig bis gar nicht genutzt werden, in den Interviews nur am Rande angesprochen werden sollten. Man sollte dabei jedoch darauf achten, dass nicht nach der Funktionsweise gefragt wird, sondern die Gründe gesucht werden, warum die Funktionen nicht benutzt wurden. Deshalb ist es sinnvoll, wie mehrmals bereits gesagt wurde, die ersten beiden Interviews als Test-Basis zu nutzen, um den Fragenkatalog bestmöglich anpassen zu können. Ein großer Vorteil der Usability-Interviews ist jedoch, dass Probleme, die von regelmäßigen Nutzern des Systems beschrieben werden, mit sehr großer Wahrscheinlichkeit auch bei Nutzern auftreten, die das System noch nicht kennen. In diesem Fall wurden von den Interviewten nur vier Probleme beschrieben, die durch die Usability-Tests nicht identifiziert werden konnten, was wiederum bedeutet, dass 72% der Probleme durch die Usability-Tests bestätigt wurden. Fehlende Funktionen wurden zudem nur von den Interviewpartnern genannt. Es war dabei mehreren Sekretärinnen aufgefallen, dass die Unterstreichen-Option innerhalb der Editieren-Maske der Wikis und Wiki-Seiten nicht vorhanden ist. Ein Interviewpartner äußerte zudem den Wunsch, dass auch Aufzählungen oder Absätze bei der Verwaltung von Attributen möglich sein sollten. Während regelmäßige Nutzer auch auf Grenzen bestimmter Funktionen treffen, testen Nutzer, die keine Erfahrung mit dem System haben, ausschließlich die vorhandene Funktionsweise. Usability-Tests können dagegen sehr gut eingesetzt werden, um spezifische Funktionen eines Systems zu testen. Dazu müssen jedoch zunächst Schlüssel-Funktionen des Systems identifiziert und bereits bestehende Problem-Bereiche abgegrenzt werden. Durch die Logfile-Analyse konnte zum Beispiel festgestellt werden, dass die Tag-Cloud nur in sehr geringem Maße zur Einschränkung der Suche genutzt wird, einzel-

ne Tags jedoch sehr oft angeklickt werden. Die von den Interviewpartnern ermittelten Problem-Bereiche konnten in den Usability-Tests intensiver untersucht werden. Einige der Sekretärinnen beschwerten sich beispielsweise über das Anlegen und Verwalten von Tabellen. Durch die Teilnehmer der Usability-Tests konnte danach getestet werden, mit welchen Tabellen-Funktionen Nutzer die meisten Probleme haben.

Zusammenfassend kann somit festgehalten werden, dass einzeln durchgeführte Usability-Evaluationsmethoden durchaus in der Lage sind, eine gewisse Anzahl an Usability-Problemen zu finden. Da jedoch bestimmte Probleme auch nur durch bestimmte Methoden aufgedeckt werden können und - wie in dieser Arbeit gezeigt - man Erkenntnisse der einen Methode als Basis für eine andere Methode verwenden kann, macht es durchaus Sinn, verschiedene Methoden miteinander zu kombinieren. In diesem Fall lieferten zum Beispiel die Logfile-Analyse und Usability-Interviews sehr viel Input, der die Planung und Durchführung der Usability-Tests um einiges erleichtern konnte. Hinzukommend sollte bei Usability-Interviews darauf geachtet werden, dass sowohl erfahrene, als auch unerfahrene Nutzer des Systems befragt werden, die außerdem das System unterschiedlich häufig nutzen. Usability-Tests müssen hingegen immer mit Nutzern der Zielgruppe durchgeführt werden, um möglichst realitätsnahe Ergebnisse erzielen zu können.

	TP1	TP2	TP3	TP4	TP5	IP1	IP2	IP3	IP4	IP5
Falscheingabe Name	X	-	X	X	-	-	-	-	-	-
Falscheingabe Login	X	-	X	X	-	-	-	-	-	-
New Wiki Button	-	-	X	X	-	X	-	-	-	-
Homepage definieren	-	-	X	X	X	-	-	-	-	-
Expert Settings	X	X	X	X	X	X	-	X	X	X
Browse Wiki Funktion	X	-	X	X	-	X	-	-	X	-
Zurück-Button	-	X	X	-	X	-	-	-	-	-
Zuordnung Wiki-Seite	X	-	-	-	X	X	-	-	X	-
Unterstreichen-Option	-	-	-	-	-	X	-	X	X	-
Maximierte Menüleiste	X	-	-	-	X	X	-	-	X	-
Tabelleneigenschaften	X	X	X	X	X	X	-	X	X	X
Darstellungsfehler	X	X	X	X	X	-	-	-	-	-
Tabellennavigation	X	X	X	X	X	-	-	-	-	-
Parent-Child Konzept	-	X	X	-	X	-	-	-	-	-
Tags hinzufügen	-	X	-	X	X	-	-	X	-	-
Tags löschen	X	-	-	-	X	-	-	-	-	-
Tags ≠ TypeTags	X	X	X	-	-	X	-	X	X	-
Vorschlag-Liste	-	-	-	X	X	-	-	-	-	-
Attribute hinzufügen	X	X	-	X	-	-	-	-	-	-
1. Lösch-Option	X	X	X	X	-	-	-	-	-	-
Attribute löschen	X	X	X	X	-	-	-	-	-	-
Attribut-Formatierung	-	-	-	-	-	-	X	-	-	-
Navigation	X	-	-	-	X	-	-	-	-	-
Verlinkung	X	-	X	-	-	-	-	-	-	-
Attachment-Reiter	-	-	-	-	-	-	-	-	X	-
Datei hochladen	-	-	-	-	-	X	-	-	X	-
Datei löschen	X	X	X	-	X	X	-	X	X	X
aktives Tab	-	X	X	-	-	-	-	X	-	-
Fehlender Tag-Hinweis	-	X	-	X	-	-	-	-	-	-
UND-Verknüpfung	-	-	X	X	-	-	-	-	-	-

Tabelle 6.1.: Zusammenfassung der Usability-Probleme

7. Zusammenfassung und Ausblick

Im abschließenden Kapitel der Arbeit sollen der Ablauf und die erzielten Ergebnisse noch einmal zusammengefasst und Ansätze für weitere Analysen und Arbeiten genannt werden. Nachdem zunächst die verschiedenen Definitionen von Usability erläutert wurden und Usability anschließend von Begriffen wie der „User Experience“ oder „Ergonomie“ abgegrenzt wurde, konnte eine Brücke zu Usability-Problemen geschlagen werden. Daraufhin wurden sowohl nutzerorientierte, als auch expertenorientierte Evaluationsverfahren in Kapitel 3 hinsichtlich ihrer Vorgehensweisen genauer erläutert. Dabei zeigte sich, dass Usability-Evaluationsmethoden in allen Phasen des Entwicklungsprozesses eingesetzt werden können. Es wurde deutlich, dass der Einsatz von Inspektionsmethoden bereits in sehr frühen Phasen der Entwicklung möglich ist, vor allem, wenn nur ein Prototyp vorhanden ist und kein fertiges Produkt oder tatsächliche Nutzer des Systems zur Verfügung stehen. Nutzerorientierte Verfahren sind hingegen meist viel aufwändiger zu planen und durchzuführen und dadurch sehr kostenintensiv, können aber Probleme aufdecken, die bei einer tatsächlichen Nutzung des Endprodukts auftreten.

Bevor eine Software oder Website hinsichtlich ihrer Bedienbarkeit evaluiert werden kann, müssen zunächst die Schlüsselfunktionen und Kernbereiche des Systems bestimmt und passende Evaluationsmethoden ausgewählt werden. In diesen Prozess müssen sowohl die zur Verfügung stehenden finanziellen Mittel und Nutzer des Systems, als auch der aktuelle Stand der Entwicklungsphase mit einbezogen werden. Hinzukommend treffen Firmen oft auf die Frage, ob sie eine oder mehrere Usability-Evaluationsmethoden einsetzen sollen. Aus der globalen Betrachtung der in dieser Arbeit beschriebenen Usability-Probleme konnten dazu unter anderem folgende Aspekte abgeleitet werden:

1. Die Durchführung einer einzelnen Usability-Evaluationsmethode bringt eine bestimmte Anzahl von Problemen zum Vorschein, die aber durch weiteren Aufwand so gut wie nicht überschritten werden kann.
2. Unterschiedliche Methoden decken unterschiedliche Arten von Problemen auf. Methoden, die mit langfristigen Nutzern des Systems durchgeführt werden, eignen sich zum Beispiel, um fehlende Funktionen zu identifizieren, während sich durch Nutzer, die das System noch nicht kennen, der Lernprozess und die Grundfunktionalität des Systems bewerten lassen.
3. Die Kombination mehrerer Methoden zieht, ausgenommen den Mehraufwand, fast ausschließlich positive Effekte nach sich.
 - Es wird eine höhere Anzahl an Problemen gefunden.
 - Durch mehrmaliges Auftreten gewisser Probleme, werden diese bestätigt.
 - Der Output einer Methode kann als Input einer anderen verwendet werden, wodurch wiederum Ressourcen gespart werden können.

4. Evaluationsmethoden decken nicht nur vorhandene Fehler des Systems auf, sondern geben auch Hinweise auf eventuell bestehende Probleme. Die Logfile-Analyse der Präfix-Suche konnte beispielsweise keine eindeutigen Fehler ausmachen, lieferte aber Hinweise auf mögliche Probleme.
5. Werden keine Probleme gefunden, ist dies ein Hinweis auf eine gute Funktionalität des Systems. Die Logfile-Analyse der Quick-Search hat zum Beispiel ergeben, dass der Suchalgorithmus die Ergebnisse liefert, die auch von den Nutzern erwartet werden.

Es ist somit zu empfehlen, mehrere Methoden miteinander zu kombinieren. Dazu muss situationspezifisch jedoch der Mehraufwand dem Nutzen gegenübergestellt und dadurch die passende Vorgehensweise gewählt werden.

Im Laufe der Arbeit wurden zudem bereits einige Ansätze für weitere Analysen oder Arbeiten genannt. Es wurde eine Vielzahl an Problemen aufgedeckt. Diese sollten jedoch unterschiedlich priorisiert werden, um entscheiden zu können, welche Probleme über kurze oder längere Zeit behoben werden sollten. Dabei müssen sowohl die Auswirkungen der einzelnen Fehler, als auch das zur Verfügung stehende Budget in den Priorisierungsprozess mit eingebunden werden. Führt ein bestehendes Problem zum Beispiel dazu, dass der Nutzer eine bestimmte Funktion nicht nutzen kann und dadurch an der Vollendung eines Prozesses gehindert wird, so sollte das Problem umgehend beseitigt werden. Des Weiteren wurden speziell durch die Logfile-Analyse Hinweise auf möglicherweise bestehende Usability-Probleme gegeben, die jedoch durch die anderen Methoden nicht ausreichend untersucht werden konnten. Anhand der Logfiles wurde beispielsweise gezeigt, dass Tags von den Nutzern sehr oft genutzt, aber nur selten miteinander kombiniert werden. Usability-Tests deckten daraufhin auf, dass manchen Nutzern nicht klar war, dass die Auswahl mehrerer Tags zu einer genaueren Einschränkung der Ergebnisse führt. Es konnte jedoch nicht genauer untersucht werden, wie die Tag-Cloud angepasst werden könnte, um dem Nutzern die gewollte Funktionsweise verständlich zu machen.

Abschließend sollte noch einmal hervorgehoben werden, dass in Zeiten von immer schneller wachsender Konkurrenz im Web, der Fokus nicht nur darauf liegen sollte, eine möglichst große Anzahl an Funktionen bereitzustellen, sondern vor allem auch darauf geachtet werden sollte, eine gute Usability der angebotenen Funktionen sicherzustellen. Hat ein Nutzer Probleme mit der Bedienung der Website oder Software, ist das Konkurrenzprodukt oft nur wenige Klicks entfernt. Um deshalb möglichst früh einen hohen Grad an Usability zu erreichen, sollten Usability-Evaluationsmethoden so früh wie möglich und an mehreren Zeitpunkten des Entwicklungsprozesses eingesetzt werden.

Appendix

A. Fragenkatalog Usability-Interviews

A.1. Offene Fragen

- Wozu wird das System hauptsächlich genutzt?
- Gibt es allgemeine Probleme bei der Bedienung des Systems?
- Welche Prozessschritte sind nicht intuitiv genug?
- Gibt es offensichtliche Fehler im System (auch die GUI betreffend)?
- Gibt es Verbesserungsvorschläge für das System?
- Wird die Seite als übersichtlich empfunden?

A.2. Geschlossene Fragen

Wikis & Wiki-Seiten

- Was ist ein Hybrid Wiki?
- Wie wird ein Wiki angelegt?
- Was bedeuten die vergebenen Berechtigungen?
- Wie wird eine neue Wiki-Seite angelegt?
- Was ist das Parent-Child Konzept? (Wie können Wiki-Seiten strukturiert werden?)
- Welche Formatierungsmöglichkeiten gibt es?
- Wie werden Bilder oder Tabellen hinzugefügt?
- Was bedeuten die Expert Settings

Tags & TypeTags

- Was sind Tags?
- Wie werden Tags & TypeTags hinzugefügt?

- Was ist der Unterschied zwischen Tags und TypeTags?
- Wozu können Tags benutzt werden?
- **Warum werden Tags nicht benutzt?**

Attribute

- Was versteht man unter Attributen?
- Wie werden Attribute hinzugefügt?
- Wie werden Attribute gelöscht?
- **Warum werden Attribute nicht benutzt?**

Attachments

- Wie können Attachments hochgeladen werden?
- Wie werden Attachments verlinkt?
- Wie werden Dateien gelöscht?

Suche

- Wie wird die Suche benutzt?
- Werden beide Komponenten der Suchfunktion erkannt?
- Was ist eine Tag-Cloud?
- Werden Filteroptionen genutzt?
- Durchsucht die Suche auch Tags?
- **Warum wird die Suchfunktion nicht benutzt?**

Literaturverzeichnis

- [BG02] BEIER, Markus (Hrsg.) ; GIZYCKI, Vittoria von (Hrsg.): *Usability. Nutzerfreundliches Web-Design*. Springer, 2002
- [ete11] ETEACHING.ORG: *Logfile Analyse*.
<http://www.e-teaching.org/didaktik/qualitaet/logfile/>, aufgerufen am 23. Mai 2011
- [Fau03] FAULKNER, Laura: Beyond the Five-User Assumption: Benefits of Increased Sample Sizes in Usability Testing. In: *Behavior Research Methods, Instruments, & Computers* 35 (2003), Nr. 3
- [Hei03] HEINDL, Eduard: *Logfiles richtig nutzen: [Webstatistiken erstellen und auswerten]*. Bonn : Galileo Press, 2003 (Galileo Computing)
- [HN07] HOLLINGSIED, Tasha ; NOVICK, David G.: Usability inspection methods after 15 years of research and practice. In: *Proceedings of the 25th annual ACM international conference on Design of communication*. New York, NY, USA : ACM, 2007 (SIGDOC '07), S. 249–255
- [Hol05] HOLZINGER, Andreas: Usability engineering methods for software developers. In: *Communications of the ACM* 48 (2005), Nr. 1, S. 71–74
- [IBM11] IBM: *Enterprise 2.0 - Tools*.
<http://www-01.ibm.com/software/de/web20/glossar.html>, aufgerufen am 23. August 2011
- [ISO99] Norm DIN EN ISO 9241-11 Januar 1999. *Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten Teil 11: Anforderungen an die Gebrauchstauglichkeit - Leitsätze*
- [ISO04] Norm DIN EN ISO 9385 2004. *Grundsätze der Ergonomie für die Gestaltung von Arbeitssystemen*
- [KCF92] KARAT, Claire-Marie ; CAMPBELL, Robert ; FIEGEL, Tarra: Comparison of empirical testing and walkthrough methods in user interface evaluation. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 1992 (CHI '92), S. 397–404
- [KLB04] KONTIO, Jyrki ; LEHTOLA, Laura ; BRAGGE, Johanna: Using the Focus Group Method in Software Engineering: Obtaining Practitioner and User Experiences. In: *ISESE*, IEEE Computer Society, 2004, S. 271–280

- [KR08] KOCH, Michael ; RICHTER, Alexander: *Enterprise 2.0 - Planung, Einführung und erfolgreicher Einsatz von Social Software in Unternehmen*. München : Oldenbourg, 2008
- [LM03] LANGFORD, Joe ; MCDONAGH, Deana: *Focus groups: supporting effective product development*. Taylor & Francis, 2003
- [Mai11] MAINKAR, Rahul: *Usability Interviews: A guide for interviewers*.
<http://www.rahulmainkar.blogspot.com/2007/11/usability-interviews-guide-for.html>, aufgerufen am 15. Mai 2011
- [Nie94] NIELSEN, Jakob: *Usability engineering*. AP Professional, 1994
- [NM90] NIELSEN, Jakob ; MOLICH, Rolf: Heuristic evaluation of user interfaces. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*. New York, NY, USA : ACM, 1990 (CHI '90), S. 249–256
- [NM94] NIELSEN, Jakob ; MACK, Robert L. ; MACK, Robert L. (Hrsg.): *Usability Inspection Methods*. Wiley John & Sons, 1994
- [SB06] SARODNICK, Florian ; BRAU, Henning: *Methoden der Usability Evaluation: Wissenschaftliche Grundlagen und praktische Anwendung*. Huber, Bern, 2006
- [SSR07] STEWART, David W. ; SHAMDASANI, Prem N. ; ROOK, Dennis W.: *Focus groups: theory and practice*. SAGE Publications, 2007 (Applied social research methods series)
- [Usa11] USABILITYTOOLKIT: *Interview*.
<http://http://www.usability-toolkit.de/usability/usability-methoden/interview/>, aufgerufen am 15. Mai 2011
- [WH06] WEISCHEDEL, Birgit ; HUIZINGH, Eelko K. R. E.: Website optimization with web metrics: a case study. In: *Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet*. New York, NY, USA : ACM, 2006 (ICEC '06), S. 463–470
- [WRLP93] WHARTON, Cathleen ; RIEMAN, John ; LEWIS, Clayton ; POLSON, Peter: *The Cognitive Walkthrough Method: A Practitioner's Guide* / Institute of Cognitive Science, University of Colorado. Boulder, Colorado, 1993. – Forschungsbericht