



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Bachelorarbeit

Zur Erlangung des akademischen Grades
eines Bachelor of Science (B.Sc.)
über das Thema

Konzeption und Umsetzung einer Open Source Anwendung für mehr Gehältertransparenz

Eingereicht im Fachbereich 4 Informatik, Kommunikation und Wirtschaft,
Studiengang Informatik und Wirtschaft
der Hochschule für Technik und Wirtschaft Berlin

Autorin:

Sarah Ganter, 544827

1. Prüferin (HTW Berlin):
2. Prüfer:

Prof. Dr. Juliane Siegeris
Bastian Albers

Berlin, 15. März 2017

... it is impossible
to design an optimal
user interface just by
giving it your best try.

Jacob Nielsen, 1993

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	III
1. Einleitung.....	1
1.1 Motivation	1
1.2 Ziel der Bachelorarbeit	3
1.3 Aufbau der Bachelorarbeit	3
2. Ausgangssituation	4
2.1 Auftraggeber.....	5
2.2 Die Anwendung „What do juniors get paid“.....	6
3. Grundlagen.....	10
3.1 Entstehung des Usability Engineerings	10
3.2 Definition und Begriffserklärung	11
3.3 User Centered Design.....	14
3.4 Methoden der Usability Evaluation	14
3.4.1 User-Test vs. Usability Inspektion	16
3.4.2 Heuristische Evaluation	18
3.4.3 Simplified Thinking Aloud	22
3.4.4 Fazit.....	24
4. Spezifikation der Evaluation.....	25
4.1 These und Annahme	25
4.2 Durchführung der Heuristischen Evaluation.....	26
4.2.1 Vorevaluatives Training	27
4.2.2 Evaluationsdurchgänge	27
4.2.3 Auswertung und Kategorisierung der Ergebnisse	28
4.2.4 Debriefing	31
4.3 Adaption der Methode	31
4.3.1 Vorevaluatives Training	32

Inhaltsverzeichnis

4.3.2 Evaluationsdurchgänge	34
4.3.3 Auswertung und Kategorisierung der Ergebnisse	35
4.3.4 Debriefing	37
5. Auswertung der Testergebnisse	37
5.1 Gegenüberstellung der Heuristiken	37
5.2 Probleme des Schweregrad 4	40
5.2.1 Fehlende Information auf der Startseite	40
5.2.2 Fehlende Filtermöglichkeiten.....	44
5.2.3 Fehlende System-Rückmeldung	45
5.2.4 Fehlender Zurück-Button	47
5.2.5 Fehlendes Impressum	48
5.3 Probleme der neue Heuristiken	49
5.3.1 Zielgruppe ist unklar	49
5.3.2 Wert der Daten	50
6. Bewertung, Validierung und Ausblick	52
6.1 Bewertung.....	52
6.2 Validierung der Ergebnisse.....	53
6.3 Ausblick	55
Literaturverzeichnis	IV
Anhangsverzeichnis	X

Abbildungsverzeichnis

Abbildung 1: geschlechtsspezifische Gehaltsentwicklung in Berliner Startups	2
Abbildung 2: Startseite der Anwendung „What do juniors get paid“	7
Abbildung 3: Formular-Seite der Anwendung „What do juniors get paid“ ...	8
Abbildung 4: Stories-Seite der Anwendung „What do juniors get paid“.....	9
Abbildung 5: Modell der System acceptability	13
Abbildung 6: Durchschnittliche Verteilung gefundener Usability Probleme, unterteilt in die drei Expertentypen	20
Abbildung 7: Kosten- und Nutzenbalance entsprechend der Anzahl an Evaluatoren	21
Abbildung 8: Verhältnis zwischen gefundenen Usability Problemen und Anzahl der Evaluatoren	21
Abbildung 9: Annahme der Verteilung der verletzten Heuristiken (vor der Evaluation).....	26
Abbildung 10: Skala zur Bestimmung des Schweregrads	30
Abbildung 11: Verteilung der Heuristiken	38
Abbildung 12: Verteilung des Schweregrads innerhalb der gesamten Anwendung.....	39
Abbildung 13: Verteilung des Schweregrads pro Heuristik.....	40
Abbildung 14: „Negative aging“ Effekt der Abbruchrate einer Webseite ..	42
Abbildung 15: Formularfeld „Arbeitsschwerpunkt“	50
Abbildung 16: Auflistung aller neu erstellten github Issues	53

1. Einleitung

Die Einleitung dieser Arbeit setzt sich aus der Motivation sowie dem Ziel und dem Aufbau der Arbeit zusammen. Im Folgenden werden die einzelnen Punkte genauer erläutert.

1.1 Motivation

In einer aktuellen Pressemitteilung des Statistischen Bundesamtes zeigt sich, dass Frauen im Schnitt 21% weniger verdienen als Männer. Der Verdienstunterschied ist so hoch, dass Deutschland eines der Schlusslichter im europäischen Vergleich bildet. Dieser Wert bezieht sich auf das Gender Pay Gap in allgemeiner Form, jedoch liegt der Unterschied bei vergleichbarer Tätigkeit und Qualifikation immer noch bei 7%. Vgl. [D_Statis 2016], [D_Statis 2013]

Der Berliner Startup Salary Report zeigt ein ähnliches Bild, wonach Frauen mit weniger Gehalt einsteigen als Männer und sich das Gehaltsgefälle über die Jahre immer mehr vergrößert. So steigt der Unterschied von 200€ beim Einstiegsgehalt auf über 1000€ bei mehr als 10 Jahren Arbeitserfahrung, siehe Abbildung 1. Vgl. [Backes et al. 2016] S.30 f.

1. Einleitung

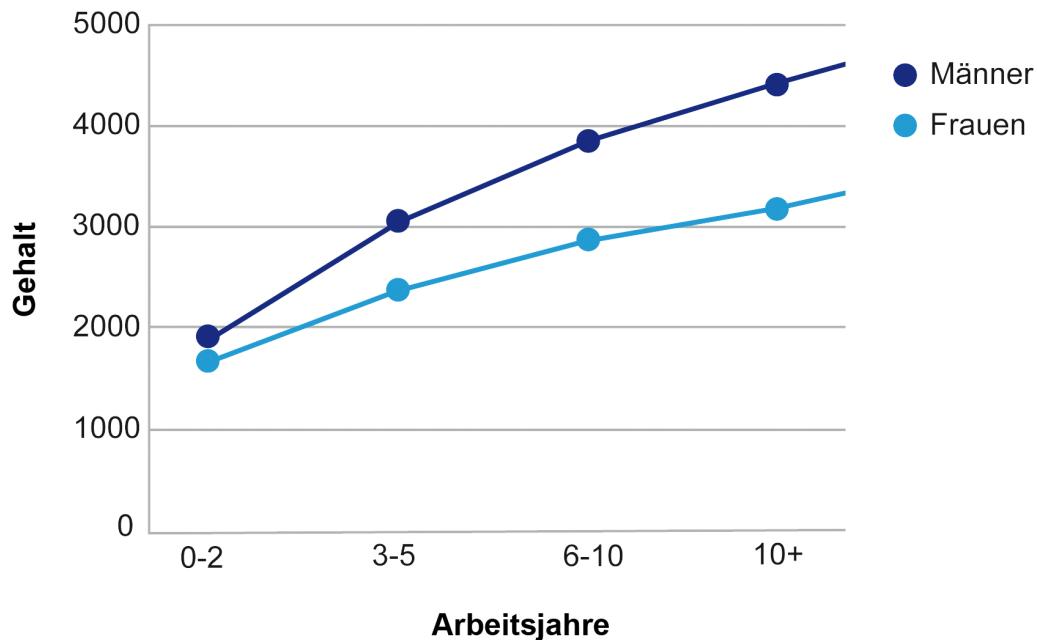


Abbildung 1: geschlechtsspezifische Gehaltsentwicklung in Berliner Startups nach [Backes et al. 2016] S. 31 [eigene Abb.]

Natürlich sind es nicht nur Frauen, die in Gehaltsverhandlungen gedrückt werden, aber die Studien zeigen, dass es ein generelles Ungleichgewicht und eine Ungerechtigkeit in der Bezahlung gibt.

So soll die vorliegende (Web-)Anwendung all diesen Menschen ein Werkzeug an die Hand geben, sich in Vertragsverhandlungen besser zu positionieren und Gehaltsangebote vergleichbar zu machen. Wenn Gehälter transparent offengelegt würden, dann wäre eine solche Ungerechtigkeit schlicht nicht mehr möglich.

Um möglichst viele Menschen zur Teilnahmen an der Umfrage der Anwendung zu bewegen, da nur eine breite Masse an Daten einen Mehrwert für die User bietet, sollte die Anwendung nach Usability-Kriterien entwickelt werden, d.h. sie sollte für den User einfach nutzbar sein und ihn bei der Erfüllung von Aufgaben angemessen unterstützen.

Gute Usability kann heutzutage ein entscheidendes Kriterium für den Erfolg oder Misserfolg einer Webseite sein. Aus diesem Grund ist es

1. Einleitung

wichtig, die Usability umfangreich zu prüfen, was mittels sogenannter Usability Studien geschieht. Ein wichtiger Bestandteil dieser Studien sind Evaluationen, mit denen Probleme identifiziert werden, was schlussendlich zur Verbesserung der Webseite führen soll.

1.2 Ziel der Bachelorarbeit

Ziel dieser Arbeit ist es, die Usability der Anwendung „What do juniors get paid“ zu untersuchen und mögliche Schwachstellen und Probleme aufzudecken. Hierfür muss eine passende Evaluationsmethode ausgewählt und durchgeführt werden. Die durch die Evaluation entstandene Problemanalyse soll den Entwicklerinnen der Anwendung zur Verfügung gestellt werden, um den Prozess der Entwicklung voranzutreiben und die Gestaltung der Anwendung hinsichtlich Usability-Kriterien positiv zu beeinflussen.

Darüberhinaus ist es das Ziel, mit dieser Arbeit ein Open Source Projekt zu unterstützen. Die Evaluation der Anwendung soll es Anderen ermöglichen, produktiv an der Weiterentwicklung mitzuarbeiten, indem Lösungsvorschläge aus den gefundenen Problemen erarbeitet werden. Diese sollen in konkrete Arbeitsaufgaben übertragen und zur Weiterentwicklung bereitgestellt werden.

1.3 Aufbau der Bachelorarbeit

Die vorliegende Arbeit wird wie folgt gegliedert: Zu Beginn wird in Kapitel 2 die zu entwickelnde Anwendung vorgestellt, sowie die Organisation, die hinter der Anwendung steht.

In Vorbereitung der durchzuführenden Evaluation wird in Kapitel 3 der Begriff Usability und ihre Notwendigkeit im Kontext der Softwareentwicklung erläutert. Es werden die Unterscheidungskriterien verschiedener Evaluationsmethoden vorgestellt, um anschliessend die eingesetzten Evaluationsmethoden (Heuristische Evaluation und

2. Ausgangssituation

Simplified Thinking Aloud) genauer vorzustellen. Diese werden dem klassischen User-Test gegenübergestellt und es wird erläutert, warum ein Einsatz eben dieser Methoden für die vorliegende Anwendung sinnvoll ist.

In Kapitel 4 geht es um die Spezifikation der Evaluation und deren Durchführung. Es wird zu Beginn eine These aufgestellt, welche Ergebnisse aus der Evaluation zu erwarten sind. Anschliessend folgt eine theoretische Beschreibung aller Phasen der Heuristischen Evaluation, um nachfolgend die Durchführung der Phasen für die konkrete Evaluation der Anwendung zu erläutern.

Nach der Durchführung der Evaluation werden in Kapitel 5 die Testergebnisse ausgewertet. Hierfür wird zunächst die in Kapitel 4 aufgestellte These dem tatsächlichen Ergebnis gegenübergestellt. Anschliessend werden spezielle Probleme, die sich aus der Evaluation ergeben haben, genauer beleuchtet und es werden Verbesserungsvorschläge erbracht.

Zum Abschluss wird im Kapitel 6 die Durchführung der Evaluationsmethode nochmal zusammengefasst und kritisch betrachtet. Die Validierung der Ergebnisse und der Ausblick auf eine Weiterentwicklung der Anwendung, unter Berücksichtigung der gefundenen Probleme, bilden jedoch einen positiven Abschluss dieser Arbeit.

2. Ausgangssituation

In diesem Kapitel wird der aktuelle Stand der zu entwickelnden Anwendung „What do juniors get paid“ zu Beginn dieser Arbeit erläutert. Zunächst wird in Abschnitt 2.1 auf die Organisation eingegangen, welche hinter der Anwendung steht und die Entwicklung maßgeblich vorantreibt. Im Anschluss wird die Anwendung samt ihres Funktionsumfanges in

2. Ausgangssituation

Abschnitt 2.2 genauer beleuchtet. Berücksichtigt werden hierbei die Motivation für die zu entwickelnde Anwendung, sowie der Aufbau und die technischen Grundlagen.

2.1 Auftraggeber

Hinter der zu entwickelnden Anwendung steht die Organisation „Juniors Are Awesome“, im Folgenden abgekürzt durch JAA. Es handelt sich hierbei um eine nicht offizielle Organisation, 2015 gegründet von Tam Eastley und Sara Regan. In Zuge dieser Arbeit wurde ein schriftliches Interview mit den beiden Gründerinnen geführt, in dem sie ihre Beweggründe und Motivation zur Gründung ihrer Organisation deutlich machen (siehe Anhang S. 1 ff.). Als Junior Entwicklerinnen haben sie die Erfahrung gemacht, dass wenige über die Schwierigkeiten und Ängste sprechen, unter denen BerufsanfängerInnen oft leiden, deshalb wollten sie einen geschützten Raum schaffen, wo sich BerufsanfängerInnen untereinander austauschen und unterstützen können. Die Mitgliedschaft bei JAA funktioniert durch Einladung und richtet sich ausschließlich an Juniors, d.h. BerufsanfängerInnen die entweder eine Einstiegsposition besetzen oder suchen, bzw. nicht über mehrjährige Berufserfahrung im technischen Bereich verfügen. Durch diese strenge Auswahl soll der geschützte Raum gewahrt bleiben. Menschen, die nicht in diese Kategorie fallen, werden nicht in die Gruppe aufgenommen. Die Auswahl der Mitglieder wird ehrenamtlich von den beiden Gründerinnen durchgeführt. Aufgrund der geografischen Lage ist ein Großteil der Mitglieder in Berlin ansässig, die Organisation richtet sich jedoch an Juniors weltweit und besitzt ebenso Mitglieder beispielsweise aus England und den USA. In Zukunft möchten die JAA weiter wachsen und gern Workshops anbieten und eine Junior-freundliche Konferenz organisieren. Aus Zeitmangel beschränkt sich der Umfang der Organisation jedoch momentan auf einen

2. Ausgangssituation

gemeinsamen Slack-Channel¹, ein Job-Board für Juniorstellen und die zu entwickelnde Anwendung „What do juniors get paid“.

2.2 Die Anwendung „What do juniors get paid“

Die Erfahrungen der beiden Gründerinnen zeigen, dass BerufsanfängerInnen ihren Wert oft nicht richtig einschätzen können und verunsichert sind, wenn es um Gehaltsverhandlungen geht. „This can lead to some situations where companies take unfair advantage of people who don't know their worth in wages.“ [ReEs 2016] S. 3 Um dieses Problem zu lösen, soll mit Hilfe der Anwendung eine repräsentative Menge an Junior-Gehältern mit Hilfe eines Fragebogens ermittelt werden.

Aufbau

Die Anwendung² ist als Open Source Projekt angelegt, d.h. dass der Quellcode öffentlich einsehbar ist und dem User das Recht eingeräumt ist, die Anwendung selbst auszuführen, zu verändern, zu verteilen und zu verbessern. Vgl. [JaegMetz 2006] S. 1 f. Die Anwendung steht zur gemeinsamen Zusammenarbeit bei github³ bereit. Github ist eine Plattform für Entwickler, die neben der Versionskontrolle durch git⁴ ebenso ein soziales Netzwerk bereitstellt, welches ihnen ermöglicht, ihre Aktivitäten (z.B. Mitwirkung an Projekten) öffentlich zu machen. Vgl. [Thung et al. 2013]

Wie viele Projekte auf github ist auch die Anwendung „What do juniors get paid“ dort öffentlich zu sehen und kann von unterschiedlichen Entwicklerinnen bearbeitet werden. Die Basisfunktionalität wurde größtenteils von den beiden Gründerinnen umgesetzt und wird zunehmend von der JAA-Community erweitert. Die Mitwirkung funktioniert nach dem Partizipationsprinzip. So können bei github verschiedene *Issues*

¹ Kommunikationstool für Gruppen: <https://slack.com>

² <https://github.com/rubycorns/whatdojuniorsgetpaid>

³ <https://github.com>

⁴ <https://git-scm.com>

2. Ausgangssituation

(Arbeitsaufgaben, Verbesserungen oder Fehlerbehebung) angelegt werden, die von jedem bearbeitet werden können und nach erfolgreichem *Review* (Qualitätsprüfung) in das Projekt eingepflegt werden.

Die Anwendung ist (bislang) einfach gehalten und gliedert sich in drei Abschnitte:

- Startseite
- Stories - Eine Auflistung aller eingereichten Daten
- Formular - Das Formular zum Erfassen der Daten

Auf der ersten Seite, der Startseite, kann der Nutzer entscheiden, ob er die gesammelten Daten der Umfrage ansehen möchte („See all stories“) oder selbst Daten zur Umfrage hinzufügen möchte („Submit your own story“). Hierfür wird jeweils ein Button zur Verfügung gestellt, siehe Abbildung 2.

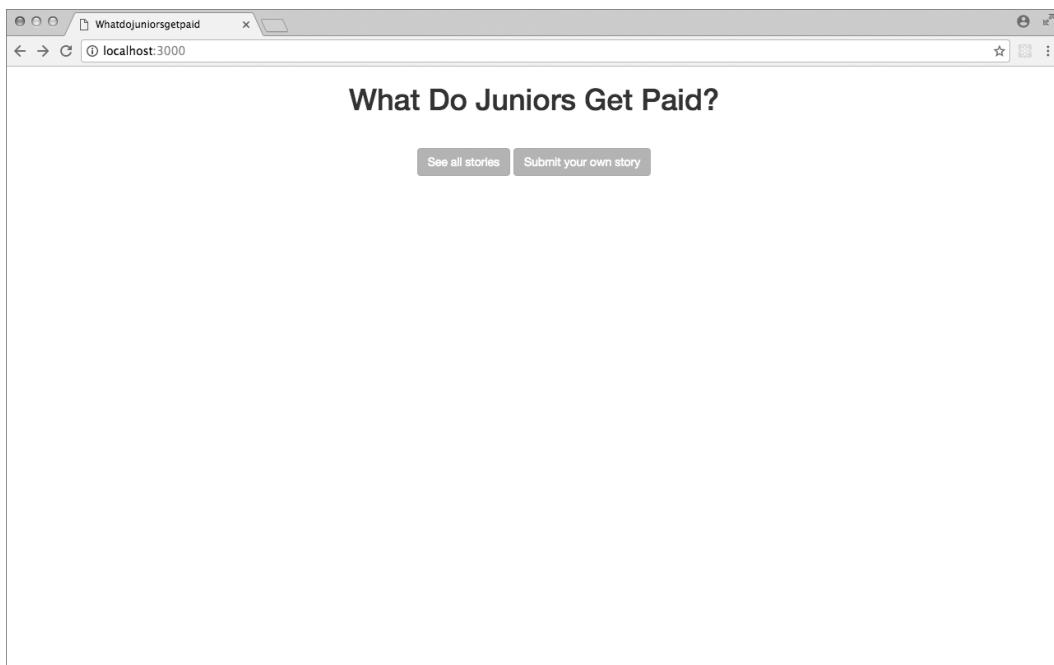


Abbildung 2: Startseite der Anwendung „What do juniors get paid“ [eigene Abb.]

2. Ausgangssituation

Entscheidet sich der User durch Klick auf den rechten Button („Submit your own story“) dafür, selbst Daten hinzuzufügen, gelangt er zu einem Formular. Dieses ist mit einem Einleitungstext versehen und soll den User dazu anregen, seine eigenen Daten hinzuzufügen. Das Formular selbst ist einseitig gestaltet und unterteilt sich in drei Themenbereiche. Durch einen Button am Ende des Formulars lassen sich die eingegebenen Daten einreichen, siehe Abbildung 3.

The screenshot shows a web browser window with the title bar 'Whatdojuniorsgetpaid'. The address bar displays 'localhost:3000/junior_stories/new'. The main content area is titled 'Create a new Junior story'. It contains the following text:
Thanks so much for taking time to fill out this form! Your input will go a long way towards helping other tech newbies.
Fill in as much or as little information as you like. In the following section, the only required fields are "freelancer" (because freelance wages can be quite high and risk throwing off our data), "salary" (because that's what this is all about), and "currency".
Please keep in mind that this will be published for everyone to see. That being said, this is completely anonymous, so we encourage you to share your story with others.

1. Let's start with the basics

I am employed as a:

The gender I identify with is: female

I identify as a person of colour

Age: 36

I live in: Berlin

Country:

My educational background when it comes to programming is:

Abbildung 3: Formular-Seite der Anwendung „What do juniors get paid“ [eigene Abb.]

Durch das Absenden der Userdaten (durch Buttonklick im Formular), sowie durch Auswahl des linken Buttons auf der Startseite („See all stories“) gelangt der User zur Ansicht aller eingereichten Daten.

Diese werden in Form sogenannter ‘Stories’ dargestellt. Alle Stories werden als Liste auf einer Seite ausgegeben, wobei jede Story als Fließtext dargestellt wird. Die Stories können per Button als CSV-Datei heruntergeladen werden, siehe Abbildung 4.

2. Ausgangssituation

The screenshot shows a web browser window with the title 'Whatdojuniorsgetpaid' and the URL 'localhost:3000/junior_stories'. The page is titled 'Junior Stories:' and features a 'Download as CSV' button. Below the title, there are five story snippets, each starting with 'I am working as a ...'. The stories are separated by horizontal lines.

- I am working as a junior developer. I identify as female. I am 31 years old. I live in the city of Berlin. I live in the country of DE. My educational field when it comes to programming is studied computer science or related field at university. I am not a freelancer. This is my first junior/intern job. My primary technology at work is Ruby (Rails, Sinatra etc...). I am a backend developer. I have been working at my current job for less than 1 year. I work 1 days per week. I work remotely. My company consists of less than 10 people. My tech team consists of 1 - 5 people. My company is 2 years old. My salary is 20000 € per year before taxes. On a scale of 1 - 5, my happiness level in my job is a 5.
- I am working as a intern. I identify as ef. I live in the city of Ewrt. I am a freelancer. I work 3 days per week. My salary is 1000 \$ (Canadian) per year before taxes.
- I am working as a junior developer. I identify as female. I am a person of colour. I am 27 years old. I live in the country of DE. My educational field when it comes to programming is other. I am a freelancer. This is not my first junior/intern job. My primary technology at work is Perl. I have been working at my current job for 1 - 2 years. My salary is 90000 € per year before taxes.
- I am working as a intern. I am not a freelancer. My salary is 6789 € per year before taxes. Additional information: drtzuknbtvret
- I am working as a junior developer. I identify as female. I am 43 years old. I live in the city of Berlin. I live in the country of DE. My educational field when it comes to programming is studied computer science or related field at university. I am a freelancer. This is not my first junior/intern job. My primary technology at work is Swift. I am a frontend developer. I have been working at my current job for 1 - 2 years. I work 5 days per week. I work remotely. My company consists of 10 - 30 people. My tech team consists of 5 - 10 people. My company is 30 years old. My salary is 24,000 £ per year before taxes. On a scale of 1 - 5, my happiness level in my job is a 5. Additional information when it comes to my happiness: jkljkljldgkjljflj8738487577028083098981^1212424352^+aaaa>CSKLJQKFJL054876792984908. Additional information: vkljfklkjlgsp958498679843090f0üpdkdkcc-bsxüwqfrf90uz
- I am 0 years old. My educational field when it comes to programming is other. I am not a freelancer. This is my first junior/intern job. My primary technology at work is Java. I am a other developer. I work 2 days per week. I do not work remotely. My company is 0 years old. My salary is 24,000 € per year before taxes. On a scale of 1 - 5, my happiness level in my job is a 2.

Abbildung 4: Stories-Seite der Anwendung „What do juniors get paid“ [eigene Abb.]

Technische Grundlage

Da beide Gründerinnen Ruby-Entwicklerinnen sind, wurde die Anwendung mit Ruby on Rails⁵ umgesetzt. Rails ist ein Web Application Framework, geschrieben in der Programmiersprache Ruby⁶. Dieses stellt Entwicklerinnen ein Gerüst zur Verfügung, welches alles beinhaltet, was zur Entwicklung einer Webseite benötigt wird. Ruby on Rails zeichnet sich durch eine kompakte Sprache und durch die Modularisierung nach dem Model View Controller Prinzip aus. Ebenso wird ein Server bereitgestellt, sowie eine Installation einer SQLite3 Datenbank zum Speichern der Daten. Vgl. [rubyonrails] Im Frontend wird aktuell nur HTML und CSS eingesetzt, wobei das Styling durch das Bootstrap-Framework⁷ generiert wird und demnach keine speziellen Anpassungen im Corporate Design besitzt.

⁵ <http://rubyonrails.org>

⁶ <https://www.ruby-lang.org/de/>

⁷ <http://getbootstrap.com>

3. Grundlagen

Kapitel 3 setzt sich ausschliesslich mit der Thematik der Usability auseinander. Zunächst wird in Abschnitt 3.1 die Entstehung des Usability Engineerings erläutert, um die Notwendigkeit von Usability in der Softwareentwicklung deutlich zu machen. Eine genauere Erläuterung des Begriffs, sowie Definitionsmöglichkeiten werden in Kapitel 3.2 näher beleuchtet. In Abschnitt 3.4 werden die verschiedenen Methoden der Usability Evaluation erklärt, um anschliessend drei der Methoden (User Test, Heuristische Evaluation und Simplified Thinking Aloud) gegenüberzustellen. In einem abschliessenden Fazit werden die Gründe für die anschliessend eingesetzte Evaluationsmethode dargelegt.

3.1 Entstehung des Usability Engineerings

Das Internet hat sich in den letzten Jahrzehnten enorm entwickelt und hat längst Einzug in das tägliche Leben der meisten Menschen gehalten. Während das Internet (bzw. sein Vorläufer ARPAnet⁸) bis Anfang der 90er Jahre nur wenigen Institutionen, wie dem Militär und akademischen Einrichtungen, zugänglich war, finden seit 1993 Einzelpersonen und Unternehmen den Weg dorthin. Vgl. [MusKe 2003] S. 2 Die Verwendungsmöglichkeiten sind inzwischen viel größer geworden; neben Informationsbeschaffung dient es heute vor allem der Interaktion und Kommunikation. Die Bereitstellung von Internetseiten allein reicht dabei nicht aus, sie müssen für den Besucher in gleichem Maße nutzbar sein.

Der Begriff des Usability Engineerings wurde bereits 1993 von Jakob Nielsen durch die Veröffentlichung seines gleichnamigen Buches geprägt. Er beschreibt, dass User zunehmend weniger gewillt sind, schlecht nutzbare Anwendungen zu tolerieren und gute User Interfaces somit den

⁸ Advanced Research Projects Agency Network of the Department of Defense

3. Grundlagen

entscheidenden Ausschlag bei der Produktdifferenzierung auf dem Markt ausmachen können.

Wo es in Zeiten vor der Revolution des Personal Computers in Ordnung war, dass nur ausgebildete Fachleute Computer bedienten, weil diese für sehr spezielle Zwecke ausgebildet waren, müssen Computerprogramme heute leicht zugänglich und verständlich sein. Vgl. [Nie 1993], S. 8 f.

Usability Engineering sei ein Prozess, der Hand in Hand mit dem Prozess der klassischen Software-Entwicklung einhergehen sollte. Dafür sei eine enge Kooperation zwischen den beteiligten Designern, Usability-Professionals und Software-Entwicklern nötig, ebenso wie die Einbeziehung der späteren Nutzer oder Vertreter der Zielgruppe, um deren Bedürfnisse und Anforderungen zu erfassen. Hierbei wäre eine iterative Vorgehensweise zu bevorzugen. Vgl. [SaBr 2011], S. 87

Diese iterative Vorgehensweise sei üblicherweise Bestandteil des Usability Engineering Lifecycle, bestehend aus Anforderungsanalyse, Design/Test/Development und Installation und beschreibt ein ständiges Re-Design des Produkts, bis alle Usability-Fehler behoben sind. Vgl. [Mayhew 1999] S. 6 ff. Das Ausmaß der Iterationsdurchgänge richte sich hierbei nach Umfang der Evaluationsmethode. Vgl. [Mayhew 1999] S. 20

3.2 Definition und Begriffserklärung

Eine allgemein gebräuchliche Definition für den Begriff Usability ist die internationale Norm 9241-11:

Usability ist „das Ausmaß, in dem ein Produkt durch bestimmte BenutzerInnen in einem bestimmten Nutzungskontext genutzt werden kann, um Ziele effektiv, effizient und zufriedenstellend zu erreichen.“ [DIN EN ISO 9241-11] S. 4

Auch im deutschen Sprachgebrauch wird heute hauptsächlich der englische Begriff Usability verwendet. Übersetzungsversuche wie Userfreundlichkeit, Brauchbarkeit oder Gebrauchstauglichkeit decken den

3. Grundlagen

Begriff der Usability nicht vollständig ab, sondern beschreiben vielmehr nur ein Teilgebiet der Usability, wie z.B. Effektivität, Effizienz oder Befriedigung. Auch die inhaltliche Abgrenzung zu den Begriffen Usefulness⁹ und Utility¹⁰ geht verloren.

Deutlich wird dies durch die Aufzeichnungen Jakob Nielsens. Zu Beginn der Revolutionierung des Personal Computers setzte sich zunächst der Begriff der *user friendliness* durch. Für Nielsen ist dieser Begriff nicht angemessen.

„[...] users don't need machines to be friendly to them, they just need machines that will not stand in their way when they try to get their work done.“ [Nie 1993] S. 23

Weiter sei der Begriff zu eindimensional gewählt, da verschiedene User unterschiedliche Bedürfnisse hätten. Was der eine User als besonders freundlich und hilfreich empfindet, kann für den nächsten mühsam und anstrengend sein. Vgl. [Nie 1993] S. 23

Aufgrund dieser mangelnden Abdeckung ersetzt Nielsen den Begriff der *user friendliness* durch den Begriff Usability - ein Konzept, welches sich aus mehreren Kriterien zusammensetzt.

Er beschreibt Usability als ein „Qualitätsmerkmal, wie einfach etwas zu benutzen ist“, [NieLo 2006] S. xvi welches im größeren Kontext der *system acceptability* zu betrachten ist, siehe Abbildung 5.

„[...] which basically is the question of whether the system is good enough to satisfy all the needs and requirements of the users and other potential stakeholders [...].“ [Nie 1993] S. 24

⁹ Brauchbarkeit, Nützlichkeit (aus dem Englischen)

¹⁰ Nutzwert, Zweckmäßigkeit (aus dem Englischen)

3. Grundlagen

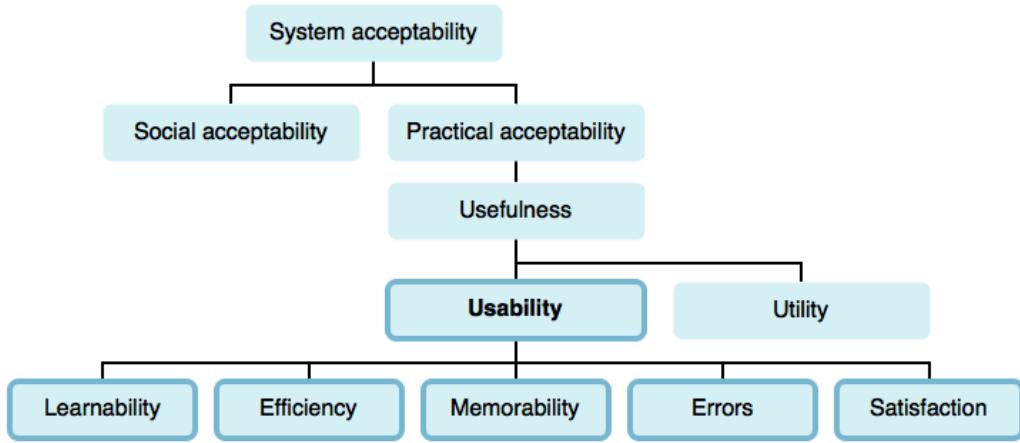


Abbildung 5: Modell der System acceptability nach [Nie 1993] S.25 [eigene Abb.]

Für Nielsen sind die Begriffe Usability und Utility gleichsam bedeutend. Utility sei ein Merkmal der Funktionalität eines Designs, welches die Fragestellung beantworten soll, ob der Benutzer genau das Ziel erreichen kann, was er benötigt. Vgl. [Nie 2012] „It matters little that something is easy if it's not what you want.“ [Nie 2012] Vereint man beide Begriffe miteinander, ist laut Nielsen der Begriff der Usefulness erfüllt.

Im Vergleich zur oben genannten Definition der internationale Norm 9241-11 schlüsselt Nielsen den Begriff der Usability in fünf Qualitätsmerkmale auf, welche es überhaupt ermöglichen, Usability greifbar und messbar zu machen.

- Erlernbarkeit (Learnability): Wie leicht können Benutzer Aufgaben bei der erstmaligen Benutzung eines Systems bewältigen?
- Effizienz (Efficiency): Wie schnell können Benutzer diese Tasks erledigen, nachdem sie mit dem System vertraut sind?
- Einprägsamkeit (Memorability): Wie leicht fällt es Benutzern, das System nach einer längeren Nutzungspause zu verwenden?
- Fehleranfälligkeit (Errors): Wie viele Fehler unterlaufen Nutzern bei der Verwendung des Systems und wie schwerwiegend sind diese?

3. Grundlagen

- Befriedigung (Satisfaction): Wie angenehm ist es dem Nutzer, das vorliegende System zu verwenden? Vgl. [Nie 2012]

3.3 User Centered Design

Für ein gutes Design ist es empfehlenswert, eine Anwendung frühzeitig mit echten Usern zu testen, um deren Anforderungen und Bedürfnisse an ein Produkt zu ermitteln. Auch wenn alle bekannten Design-Prinzipien befolgt werden, ist es unmöglich, ein optimales User Interface zu kreieren, ohne die User direkt zu befragen.

„Users have infinite potential for making unexpected misinterpretations of interface elements and for performing their job in a different way than you imagine.“ [Nie 1993] S. 10

Des Weiteren schreibt Nielsen, dass Designer keine User seien und es ein Irrglauben sei anzunehmen, ein Designer wüsste genau, was der User möchte. Designer unterschieden sich durch ihre mentale Einstellung und ihr zu tiefes Wissen über das Produkt von den Usern. Durch diesen Wissensvorsprung fällt es schwer, unbefangen zu sein und keine generellen Annahmen über die Funktionalität eines Produkts zu machen, die der User ohne tieferes Verständnis über die Technik nicht machen kann. Vgl. [Nie 1993] S.13 „Knowing about a system is a one-way street. One cannot go back to knowing nothing.“ [Nie 1993] S.13 Daher sollte sichergestellt werden, dass das User Interface stetig durch Evaluation validiert wird und bei Bedarf in Funktionalität und Design angepasst wird. Vgl. [Nie 1993] S.10 f.

3.4 Methoden der Usability Evaluation

„Evaluation allgemein bezeichnet eine systematische und möglichst objektive Bewertung eines geplanten, laufenden oder abgeschlossenen Projekts“ [SaBr 2011] S. 23 und soll dieses in Bezug auf

3. Grundlagen

Zielerreichungsgrad eines bestimmten Vorhabens bewerten oder bestimmte Fragestellungen klären. Die erworbenen Erkenntnisse können Hinweise zur Verbesserung eines laufenden Projekts liefern oder hilfreich für die Planung kommender Projekte sein. Um konkrete Entscheidungs- oder Problemlösungsprozesse zu unterstützen, sollte eine Evaluation immer glaubwürdige und nützliche Informationen liefern. Vgl. [SaBr 2011] S. 23 f.

Evaluationsmethoden lassen sich grundlegend in folgende Unterscheidungskriterien aufteilen:

- formative und summative Evaluationsmethoden
- formale und informelle Evaluationsmethoden
- analytische und empirische Evaluationsmethoden

Das erste Unterscheidungskriterium basiert auf dem Zeitpunkt, zu dem die Evaluation durchgeführt wird. Die formative Evaluation findet bereits während des Entwicklungsprozesses statt, um Schwächen und Probleme im Design frühzeitig zu erkennen und iterativ verbessern zu können. Vgl. [GeHaDü 1999] Die summative Evaluation wird hingegen oft nach Abschluss der Fertigstellung durchgeführt. Ziel ist es, Erkenntnisse für weitere (ähnliche) Produkte zu erhalten, indem das fertige Produkt mit ähnlichen Produkten oder mit allgemeinen Standards verglichen wird. Vgl. [RedBias et al. 2002] Während summative Evaluationsmethoden nur zu einer globalen Bewertung der Anwendung ohne konstruktive Verbesserungsvorschläge führen, ist es das Ziel der formativen Evaluation, konkrete Verbesserungsmöglichkeiten aufzuzeigen. Vgl. [SaBr 2011] S. 120

In der Praxis findet sich eine Vielzahl an Projekten, die auf frühzeitige Evaluationen verzichten und erst gegen Ende der Projektlaufzeit eine summative Evaluation zur Bestätigung der abschliessenden Gebrauchstauglichkeit durchführen. Da es jedoch eher die Regel ist, dass

3. Grundlagen

gravierende Mängel festgestellt werden, können diese zu diesem Zeitpunkt nur noch mit einem hohen finanziellen und zeitlichen Aufwand ausgeglichen werden. Vgl. [SaBr 2011] S. 24 f. „Die kontinuierliche und frühzeitige Beteiligung von Usability-Experten während der Analysephase ist somit auch aus betriebswirtschaftlichen Gründen unbedingt ratsam.“ [SaBr 2011] S. 25

Bei den formalen und informellen Evaluationsmethoden basiert die Unterscheidung auf der genauen Spezifikation, die zur Durchführung angewandt wird. Die meisten Usability-Inspektionen entsprechen einer informellen Evaluationsmethode, welche sich auf bestimmte Faustregeln guter Usability und auf das Wissen und die Erfahrung der User bzw. Experten beruft. Formale Evaluationen sind hingegen eher schwer durchführbar, da hierfür genaue Modelle benötigt werden und sie jeweils nach einem spezifischen Schema ablaufen, um die Usability eines Produkts zu bewerten. Somit sind diese Methoden auch schwer skalierbar und auf komplexere Interfaces anwendbar. Vgl. [NieMa 1994] S. 2

Die Unterscheidung zwischen analytischer und empirischer Evaluation besteht darin, welche Personengruppe die Evaluation durchführt. Bei der empirischen Evaluation sind es echte User, die ein User Interface testen. Bei der analytischen Evaluationsmethode hingegen wird das Interface durch Usability-Experten getestet, welche sich in die Rolle der User hineinversetzen sollen und ihre Beurteilung auf Grund bestimmter Usability-Richtlinien und ihres eigenen Domänenwissens treffen. Vgl. [SaBr 2011] S. 119

3.4.1 User-Test vs. Usability Inspektion

Der Begriff Usability-Inspektion wurde 1994 von Nielsen und Mack eingeführt und beschreibt eine Art der formativen Evaluationsmethoden, bei der die Evaluation von Usability-Experten anstelle von „echten“ Usern durchgeführt wird. Vgl. [SaBr 2011] S. 142 Der Begriff ist angelehnt an

3. Grundlagen

Code-Inspektionen, wie sie in der Programmierung eingesetzt werden, hat aber in der Durchführung wenig Ähnlichkeit mit diesen. Vgl. [NieMa 1994] S. 2

Der klassische User-Test ist die wohl häufigste Methode der empirischen Evaluation. Hierbei ist, wie im vorangegangene Kapitel bereits erwähnt, entscheidend, dass ein User Interface mittels echter User getestet wird. Die Auswahl der User sollte dabei ein breites Spektrum der anvisierten Zielgruppe abbilden, wobei die ausgesuchten Tester das User Interface noch nicht kennen sollten. Vgl. [SaBr 2011] S. 167 Unter Beobachtung von Usability-Experten wird das zu testende Interface anhand realistischer Aufgaben erprobt. In welcher Form die Beobachtung stattfindet, hängt vom Umfang und Budget des Tests ab. Üblicherweise werden User-Tests jedoch in einem Usability-Labor durchgeführt, welches mit umfangreichem Ton- und Videoequipment ausgestattet ist. Laut Sarodnick und Brau spielt das technische Equipment eine entscheidende Rolle, da mittels Video- und Tonaufzeichnung das Verhalten der Testpersonen genau analysiert werden kann. Oft kommen mehrere Kameras zum Einsatz, die aus verschiedenen Blickwinkeln aufzeichnen, um keine wichtige Interaktion zu verpassen. Neben den Testleitern bietet ein klassisches Usability-Labor mit einem abgeschotteten Kontrollraum ebenfalls die Möglichkeit zusätzliche Beobachter in Gestalt von Entwicklern, Auftraggebern und weiteren Usability-Experten teilnehmen zu lassen. Vgl. [SaBr 2011] S. 168 f. Durch die verschiedenen Ton- und Bildaufzeichnungen sowie mögliche Protokolle, Interviews und Messungen ergibt sich eine Fülle an Informationen, die im Anschluss an den User-Test ausgewertet und analysiert werden müssen. Vgl. [SaBr 2011] S. 163

Beide Arten der Usability-Evaluation haben ihre Vor- und Nachteile und müssen individuell je nach Zielsetzung ausgewählt werden. Obgleich Nielsen ein Fürsprecher der Usability-Inspektion ist, ist es für gute

3. Grundlagen

Usability jedoch unumgänglich, tatsächliche User zu befragen. Er sagt dazu:

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

Ebenso sei es in bestimmten Fällen sinnvoll, auf Inspektionsmethoden zurückzugreifen, die eine Problemanalyse bei niedrigeren Kosten bereitstellen. Vgl. [Nie 1993] S.165

Es sei auch zu erwähnen, dass die besten Ergebnisse erzielt werden, wenn mehrere Methoden kombiniert werden. Oft werden bei User-Tests und Inspektionen verschiedene Probleme aufgedeckt, die keine Überschneidungen haben und von der jeweils anderen Methode übersehen wurden. Somit lässt sich in Kombination mehrerer Methoden eine möglichst flächendeckende Problemanalyse anstellen. Vgl. [NieMa 1994] S. 2

3.4.2 Heuristische Evaluation

Die Heuristische Evaluation ist die informellste der Inspektionsmethoden und stützt sich auf Usability-Prinzipien (Heuristiken), nach denen ein User Interface geprüft und bewertet wird. Vgl. [NieMa 1994] S. 5 Sie zählt zum sogenannten Discount Usability Engineering, ein Begriff, der bereits 1989 von Nielsen eingeführt und seitdem, nicht zuletzt durch die Heuristische Evaluation, als eines der bekanntesten Verfahren dieser Art sehr propagiert wird. Vgl. [Nie 1994] S. 25, [SaBr 2011] S. 213 Viele Evaluationsmethoden scheinen zu aufwändig und kostenintensiv, so dass diese Vorhaben oft spätestens auf Entwickler- oder Managerebene gestoppt werden, was im schlimmsten Fall dazu führen kann, dass am Ende kein Test durchgeführt wird. Vgl. [Nie 1993] S. 17

3. Grundlagen

Die Heuristische Evaluation charakterisiert sich im Gegenzug zu empirischen Verfahren, wie dem User-Test durch:

- Zeitersparnis
- geringere Kosten
- geringere Komplexität

Zur Minimierung der Komplexität beschränkt Nielsen die Usability-Prinzipien auf zehn Heuristiken, mit Hilfe derer ein User Interface geprüft werden soll. Generell gibt es verschiedene Heuristiken, die allgemein oder auch spezieller für ein spezifisches Interface formuliert sein können. Ebenso ist es auch möglich, eigene Heuristiken zu erstellen. Die Heuristiken nach Nielsen sind jedoch so gestaltet, dass sie allgemein auf alle Projekte anwendbar sind und eine große Bandbreite an möglichen Problemen eines User Interfaces abdecken. Um die Heuristiken richtig anzuwenden, benötigen die Tester eine gewisse Usability-Expertise.

Die besten Ergebnisse lassen sich erzielen, wenn Tester sowohl über Usability-Expertise als auch über Domänenwissen verfügen. Eine Studie Nielsens von 1992 zeigt, wie die Aufdeckung von Usability-Problemen exponentiell ansteigen kann je nach Expertise der Tester. So wurden in einer Heuristischen Evaluation eines Sprachdialogsystems von den Testern, die nur über Domänenwissen verfügten, 22% der Probleme erkannt, Tester mit Usability-Expertise fanden 41% der Probleme und Tester mit Expertise in beiden Bereichen deckten 60% der Probleme auf, siehe Abbildung 6. Vgl. [Nie 1992] S. 376

3. Grundlagen

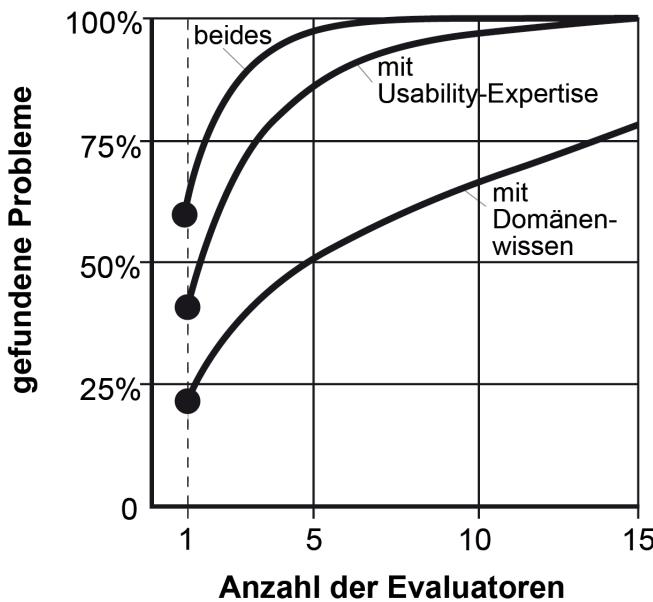


Abbildung 6: Durchschnittliche Verteilung gefundener Usability Probleme, unterteilt in die drei Expertentypen nach [Nie 1992] S. 377 [eigene Abb.]

In erwähnter Studie beschreibt Nielsen die Experten als Menschen, die bereits einen Abschluss oder mehrjährige Berufserfahrung im Usability-Bereich haben, eine offizielle Abgrenzung, was einen Usability-Experten ausmacht, gibt es jedoch nicht. So beschreibt er zwei Jahre später ebenfalls, dass man Experten innerhalb eines halbtägigen Seminars selbst generieren kann, was die Heuristische Evaluation zu einem vergleichsweise schnellen und kostensparenden Vorgang macht. Vgl. [Nie 1994] S. 25

Ein weiterer Faktor für Kostensparnis ist der Einsatz weniger Experten. Es ist sinnvoll, ein User Interface mit mehreren Experten zu testen, da verschiedene Menschen verschiedene Fehler finden. Erfahrungen der Vergangenheit zeigen, dass die Evaluation mit nur einem Experten 35% der Fehler aufdeckt und man bereits eine positive Anstiegsrate der gefundenen Fehler ab zwei Testern verzeichnen kann. Empfohlen ist es, die Evaluation mit 3-5 Experten durchzuführen, weil hierbei die Kosten- und Nutzenbalance am ausgeglichensten ist, siehe Abbildung 7. Sofern Budget und Zeit in einem Projekt keine Rolle spielen, können mehr Experten eingesetzt werden, ebenso, wenn es sich um kritische

3. Grundlagen

Anwendungsfälle handelt. Ansonsten können mit 3-5 Evaluatoren bereits zwei Drittel der Usability-Probleme in einem User Interface ausfindig gemacht werden, siehe Abbildung 8. Vgl. [Nie 1994] S. 32 f.

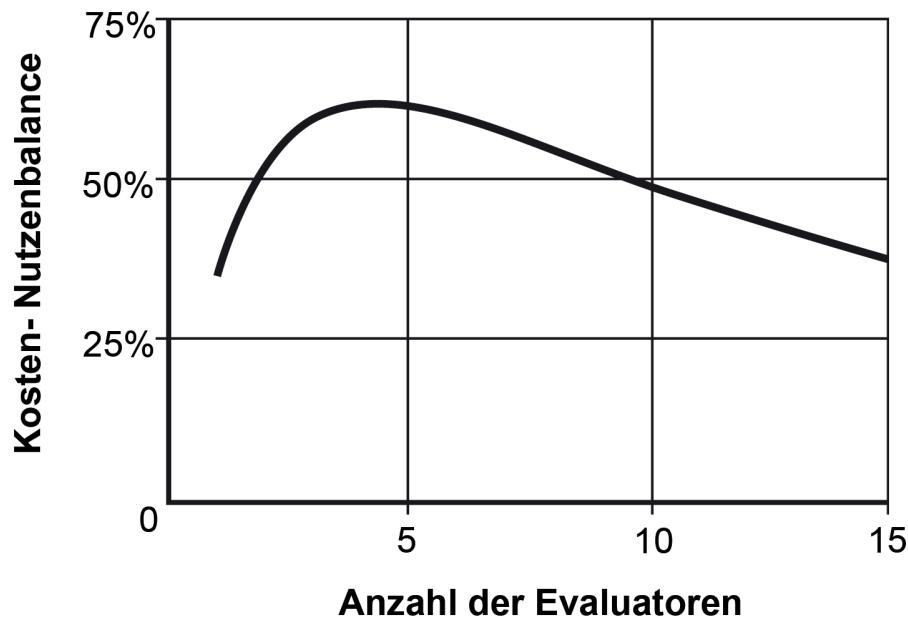


Abbildung 7: Kosten- und Nutzenbalance entsprechend der Anzahl an Evaluatoren nach [Nie 1994] S. 35 [eigene Abb.]

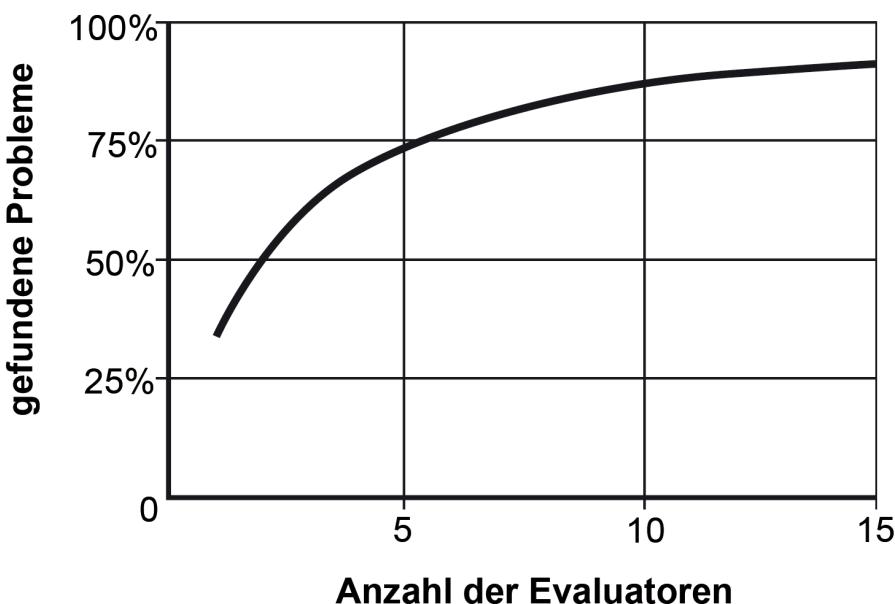


Abbildung 8: Verhältnis zwischen gefundenen Usability Problemen und Anzahl der Evaluatoren nach [Nie 1994] S. 33 [eigene Abb.]

3. Grundlagen

Des Weiteren ist die Durchführung einer Heuristischen Evaluation vergleichsweise schnell; die meisten Evaluationen lassen sich innerhalb eines Tages erledigen. Vgl. [Nie 1994] S. 25

Generell ist anzumerken, dass es in der eigenen Verantwortung liegt, in welchem Umfang man eine Heuristische Evaluation (oder eine andere Methode des Discount Usability Engineerings) durchführt. Je nach Sorgfalt und Genauigkeit der Durchführung der Evaluation lassen sich bessere Ergebnisse erzielen, dies ist jedoch eine Zeit- und Kostenfrage. Vgl. [NieMa 1994] S. 17

3.4.3 Simplified Thinking Aloud

Die Thinking Aloud Methode verlangt, dass ein Testuser während der Untersuchung eines Interfaces seine Eindrücke kontinuierlich verbalisiert. Hierdurch gibt er Einblick in sein eigenes Verständnis des Interfaces und ermöglicht dem Versuchsleiter, direkt nachzuvollziehen, an welcher Stelle es zu Missverständnissen kommt und welche Dialogelemente besondere Probleme bereiten.

Da das „laut denken“ sich für die meisten Menschen eher unnatürlich anfühlt, ist der Versuchsleiter immer dazu angehalten, die Testuser zum Sprechen zu animieren. Dies erfordert eine genaue Beobachtung des Testusers, da es wichtig ist, seinen Arbeitsfluss nicht zu stören und keine Erkenntnisse vorwegzunehmen. Der Versuchsleiter agiert zwar als Gesprächspartner, muss aber selbst neutral bleiben und darf den Testuser in seiner Meinung nicht beeinflussen. So soll der Versuchsleiter auch nicht auf Fragen direkt antworten, sondern den Testuser mittels Gegenfragen immer anregen, selbst nachzudenken.

Die Methode bietet die Möglichkeit, qualitative Daten mit Hilfe nur weniger User zu sammeln, ist aber nicht dazu geeignet, Aussagen über die Performance, d.h. die Geschwindigkeit des Users beim Erfüllen bestimmter Aufgaben zu treffen. Das Thinking Aloud kann zum einen durch den Prozess des Sprechens den Ablauf der Evaluation

3. Grundlagen

verlangsamen, zum anderen kann es sich positiv auf die Problemfindung auswirken durch die Tatsache, dass sich Testuser mehr selbst reflektieren und durch das laute Aussprechen Fehler schneller wahrnehmen. Beides würde zu einer Verfälschung statistischer Ergebnisse führen. Die gewonnenen qualitativen Daten hingegen geben einen wertvollen Einblick in das mentale Modell¹¹ der einzelnen Testuser und ermöglichen, informelle Kommentare über einzelne Aspekte des Interfaces zu sammeln, die bei anderen Testmethoden üblicherweise nicht erhoben werden. Zu beachten ist jedoch, dass die einzelnen Daten jeweils subjektive Eindrücke widerspiegeln. Es ist Sache des Versuchsleiters, die Kommentare zu interpretieren, indem genau erfasst wird, was der User in jedem Schritt macht und warum er das tut. Es darf am Ende keine Änderungen am Design aufgrund einzelner User-Aussagen geben. Diese sind rein subjektiv und sollten in aggregierter Form betrachtet werden. Vgl. [Nie 1993] S. 195-198 Wie Nielsen anmerkt, haben User nicht immer Recht: „[...] users will often have divergent opinions when asked about details of user interface design.” [Nie 1993] S. 12

Das Simplified Thinking Aloud unterscheidet sich vom (traditionellen) Thinking Aloud dadurch, dass es wesentlich einfacher durchführbar ist. Während die traditionelle Methode einen (im Idealfall) ausgebildeten Psychologen als Versuchsleiter, sowie umfangreiches Equipment in Form von Videoaufzeichnung verlangt, lässt sich die vereinfachte Methode mit schriftlicher Aufzeichnung seitens des Versuchsleiters durchführen. Im Vorfeld kann ein minimales Training des Versuchsleiters erfolgen, Nielsen sagt jedoch aus, dass auch methodisch primitive Experimente erfolgreich viele Usability Probleme aufdecken. Vgl. [Nie 1993] S. 18 f.

¹¹ Das Mentale Modell ist, was der User über ein System im Vorfeld denkt. Der Umgang mit einem System und die Erwartungshaltung ist stark vom individuellen Glaube jedes einzelnen Users abhängig. Es basiert nicht auf Fakten und steht oft im Kontrast zu dem mentalen Modell eines Designers. Vgl. [Nie 2010]

3. Grundlagen

Ebenso wie die Heuristische Evaluation gehört das Simplified Thinking Aloud zum Discount Usability Engineering und soll durch Minimierung des Aufwands und die damit verbundene Kostenersparnis den Zugang zu Evaluationsmethoden erleichtern.

3.4.4 Fazit

Aus organisatorischen Gründen ist die Heuristische Evaluation als Evaluationsmethode im Zuge dieser Arbeit zu bevorzugen. Hierfür gibt es verschiedene Gründe. Zum einen stehen die benötigten Expertinnen zur Durchführung der Evaluation bereits im näheren Umfeld zur Verfügung und müssen nicht akquiriert werden. Auch der Einsatz von umfangreichem Equipment wie einem geeigneten Raum zur Durchführung eines User-Tests, sowie Kamera- und Tonausstattung, fällt damit weg. Auch wenn der User-Test ebenfalls in kleinerem Format durchgeführt werden könnte, ist der Gesamtaufwand zur Durchführung wesentlich höher als bei der Heuristischen Evaluation. Denn es werden nicht nur mindestens zwei Versuchsleiter benötigt, die sich um Kamera und Versuchsablauf kümmern müssen, sondern auch die Vor- und Nachbereitung wird umfangreicher, da die gewonnenen Daten aus einem User-Test sowohl analysiert als auch interpretiert werden müssen.

Da sich die besten Ergebnisse erzielen lassen, wenn beide Evaluationsmethoden anwendet werden, schlägt Nielsen ein iteratives Vorgehen vor, in dem zuerst die Heuristische Evaluation angewandt und anschliessend ein User-Test durchgeführt wird. Der Fokus in der zweiten Iteration kann somit auf spezifischen Problemen liegen, die von der Heuristischen Evaluation übersehen wurden. Vgl. [Nie 1994] S. 57

Das wäre auch der Ansatz für diese Arbeit, jedoch kommt es im Falle der vorliegenden Anwendung zu einer seltenen Überschneidung von Expertinnen und Userinnen, so dass allein die Durchführung der Heuristischen Evaluation vermutlich bessere Ergebnisse erzielen wird als

die übliche Heuristische Evaluation ohne User-Expertise und das dazugehörige Domänenwissen. Die in Kapitel 4.3 beschriebenen Testteilnehmerinnen fallen als zukünftige Informatik-Absolventinnen und Werkstudentinnen genau in die anvisierte Zielgruppe der Anwendung, sind aber ebenso als Expertinnen für die Heuristische Evaluation geeignet, da sie über das nötige Fachwissen verfügen. Dieser Umstand lässt es zu, dass in Kombination mit dem Simplified Thinking Aloud ähnliche Erkenntnisse gewonnen werden können wie in einem User-Test.

4. Spezifikation der Evaluation

In diesem Kapitel erfolgt die Spezifikation der Evaluation, welche in Form der Heuristischen Evaluation in Kombination mit dem Simplified Thinking Aloud durchgeführt wird. In Abschnitt 4.1 wird zunächst eine These aufgestellt, welche Ergebnisse nach der Evaluation zu erwarten sind. Anschliessend wird in Abschnitt 4.2 die Heuristische Evaluation in allen Phasen genauer erklärt, um in Abschnitt 4.3 die Adaption eben dieser Phasen zu beschreiben. Dies geschieht unter Nennung der Testteilnehmerinnen und der zugrundeliegenden Heuristiken.

4.1 These und Annahme

Als Open Source Projekt steht für die Entwicklung der Anwendung kein Budget zur Verfügung. Es ist somit anzunehmen, dass neben der Entwicklungsarbeit noch kein besonderes Augenmerk auf der Usability lag, die einen zusätzlichen Aufwand bedeuten würde. Oft unterliegen Entwickler und Designer dem Irrglauben, dass ihre User wie sie selbst seien und sie dadurch wüssten, was sich der durchschnittliche User wünschte. Vgl. [Krug 2006] S. 128

Der bisherige Fokus bei der Entwicklung der Anwendung lag auf der Backend-Entwicklung und somit auf dem reibungslosen Ablauf der Programmlogik. Da die Anwendung bislang noch nicht getestet wurde,

4. Spezifikation der Evaluation

haben also ausschliesslich Entwickler bei der Gestaltung mitgewirkt. Steve Krug spricht in seinem Buch „Don't make me think!“ von einer professionellen Leidenschaft, die je nach Beruf einen unterschiedlichen Blickwinkel auf ein Produkt ausmacht. Entwickler gingen somit weniger mit einem ästhetischen, als mit einem funktionalen Gedanken an die Sache heran. Vgl. [Krug 2006] S. 126

Des Weiteren ist die vorliegende Anwendung mit drei Seiten sehr überschaubar gehalten und besitzt noch kein eigenes Styling. Aus diesen Gründen ist anzunehmen, dass die meisten Probleme im Design zu finden sind und somit auf die erste Heuristik, *Ästhetik und minimalistisches Design* fallen, siehe Abbildung 9.

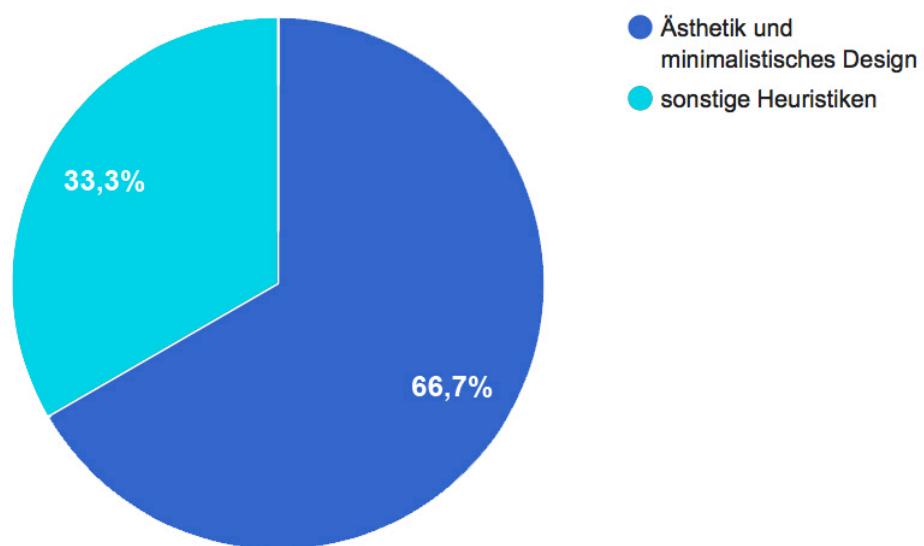


Abbildung 9: Annahme der Verteilung der verletzten Heuristiken (vor der Evaluation) [eigene Abb.]

4.2 Durchführung der Heuristischen Evaluation

Im Folgenden wird die Durchführung der Heuristischen Evaluation genauer erläutert.

Die Heuristische Evaluation besteht aus vier Phasen:

4. Spezifikation der Evaluation

- Vorevaluatives Training
- Evaluationsdurchgänge
- Auswertung und Kategorisierung der Ergebnisse
- Debriefing

4.2.1 Vorevaluatives Training

Vor der eigentlichen Evaluation sollte eine Einführung in die zugrunde liegenden Heuristiken, sowie in die zu testende Anwendung gegeben werden, um eine einheitliche Terminologie sicherzustellen. Wie umfangreich dieses vorevaluative Training gestaltet wird, hängt davon ab, über welches Wissen die Experten bereits verfügen. Wie bereits erwähnt, können Evaluatoren mit weniger Expertise in einem umfangreicheren Training zu Experten ausgebildet werden; handelt es sich aber bereits um Usability-Experten, kann dieses Training verkürzt werden und sich bei Bedarf auf die genauere Erläuterung der Anwendung beschränken, um für das nötige Domänenwissen zu sorgen. Dies umfasst nicht nur die grobe Erläuterung der Anwendung, sondern auch die Fragestellung, welches Problem die Anwendung lösen soll und wer die Zielgruppe ist. Dies ist deshalb wichtig, damit sich die Evaluatoren in die Sichtweise der tatsächlichen Nutzer versetzen können und nach bestem Wissen als deren Stellvertreter agieren können. Hilfreich ist hierfür die Beschreibung eines typischen Szenarios, wobei darauf geachtet werden sollte, nicht zu viel von der zu testenden Anwendung zu zeigen, damit die Experten unvoreingenommen an die Evaluation herangehen. Vgl. [NieMa 1994] S. 39, [SaBr 2011] S. 218 f.

4.2.2 Evaluationsdurchgänge

Die eigentliche Evaluation wird von jedem Experten unabhängig und allein durchgeführt, um eine gegenseitige Beeinflussung zu vermeiden. Wie sich der Ablauf der Evaluation gestaltet, ist den Experten selbst überlassen, empfohlen ist es jedoch, dass jeder Experte das Interface mindestens

4. Spezifikation der Evaluation

zweimal durchgeht. Der erste Durchgang dient dazu, sich mit der Anwendung vertraut zu machen. Hierfür kann ggf. ein vorgestelltes Szenario¹² genutzt werden, um ein Gefühl für den Ablauf der Anwendung zu bekommen. In weiteren Durchläufen prüft der Experte die einzelnen Elemente in Bezug auf die anzuwendenden Heuristiken. Der Fokus liegt hierbei auf der Identifizierung möglicher Probleme, nicht auf deren Verbesserung.

Als Dokumentation der gefundenen Probleme sind schriftliche Protokolle ausreichend, welche entweder von den Evaluatoren selbst angefertigt werden oder von den beisitzenden Beobachtern bzw. dem Versuchleiter, indem die Evaluatoren ihre Entdeckungen verbalisieren.

Wenn möglich, sind die Experten dazu angehalten, sowohl eine Einordnung des gefunden Problems in eine der gegebenen Heuristiken vorzunehmen, als auch eine Einschätzung des Schweregrads des Problems.

In den Studien von Nielsen ergab sich jedoch, dass eine zu große Involviering des Versuchsleiters mittels genauerer Nachfrage die Experten zu sehr verunsicherte, weshalb Nielsen davon abrät, den Arbeitsfluss zu sehr zu unterbrechen.

Die Dauer eines Evaluationsdurchgangs richtet sich nach der Komplexität der Anwendung, beträgt in der Regel aber 1-2 Stunden. Nielsen empfiehlt bei längeren Sitzungen, die Evaluation in einzelnen Abschnitte zu teilen.
Vgl. [Nie 1994] S. 28

4.2.3 Auswertung und Kategorisierung der Ergebnisse

Nach den Evaluationsdurchgängen wird durch den Versuchsleiter eine überschneidungsfreie Gesamtliste aller gefundenen Usability-Probleme erstellt. Abhängig von der Komplexität der Anwendung sollte ein unterschiedlicher Grad der Granulation gewählt werden. Eine niedrige Granulation sollte gerade bei komplexen Anwendungen gewählt werden,

¹² ein Szenario ist eine spezifisch vorgegebene Testaufgabe um die Komplexität zu verringern. Vgl. [Nie 1993] S. 18

4. Spezifikation der Evaluation

wohingegen der Grad der Granulation bei einfachen Anwendungen (mit weniger Masken) höher sein darf. Die Erstellung der Gesamtliste wie auch das Filtern und Auswerten der vielen Informationen stellt sich in der Praxis oft schwierig dar und benötigt eine gewisse Erfahrung seitens des Versuchsleiters. Vgl. [SaBr 2011] S. 221 f.

Mit etwas zeitlichem Abstand werden die dokumentierten Probleme nach Schweregrad bewertet und kategorisiert. Dieser Schritt ist wichtig, um eine Priorisierung der Probleme zu erhalten. Im Normalfall können nicht alle Probleme in der nachfolgenden Iteration behoben werden, da jedes Projekt einen gewissen Zeit- und Budget-Rahmen hat.

Nach Nielsen werden hierfür vier verschiedene Faktoren herangezogen:

- Frequenz: Wie häufig taucht das Problem auf?
- Einfluss: Wie sehr stört das Problem den Arbeitsablauf?
- Persistenz: Kann das Problem bei Kenntnis umgangen werden oder taucht es wiederholt auf?
- Markteinfluss: Erzeugt das Problem einen ungünstigen Eindruck des Gesamtprodukts?

Zur Bewertung der Probleme wird erneut auf die Evaluatoren zurückgegriffen, welche anhand der Gesamtliste aller gefundenen Probleme eine Einschätzung des Schweregrads vornehmen sollten. Da die Beurteilung sowohl die eigenen gefundenen Probleme als auch die der anderen Experten umfasst, ist es notwendig, eine ausreichende Beschreibung der Probleme bereitzustellen. Dies kann mit Hilfe von beigefügten Screenshots oder den Aufzeichnungen aus den einzelnen Evaluationssitzungen erfolgen. Wenn die Möglichkeit besteht, kann auch das getestete System erneut zur Verfügung gestellt werden, Nielsen weist jedoch darauf hin, dass dies den Prozess der Bewertung verlangsamen würde.

4. Spezifikation der Evaluation

Die Evaluatoren sollten zur Dokumentation des Schweregrads folgende von Nielsen konzipierte Skala heranziehen:

0	Ich stimme nicht zu, dass das ein Usability Problem ist	-
1	Kosmetisches Problem	Braucht nicht behoben zu werden, außer es ist genug Zeit
2	Geringfügiges Usability Problem	Die Behebung hat untergeordnete Priorität
3	Bedeutendes Usability Problem	Die Behebung des Problems hat hohe Priorität
4	Usability Katastrophe	Das Problem muss behoben werden!

Abbildung 10: Skala zur Bestimmung des Schweregrads nach [Nie 1994] S. 47 ff. [eigene Abb.]

Sinnvoll sei es, dass so viele der Evaluatoren wie möglich an der Bewertung der Probleme teilnehmen. Je größer die Zahl der Teilnehmer, desto verlässlicher wird das Ranking ausfallen, so Nielsen. Die Aussagekraft der Bewertung des Schweregrads seitens einer einzigen Person sei so gering, dass keine Investitionsentscheidungen daraufhin getroffen werden sollten. Vgl. [Nie 1994] S. 49 f.

Sarodnick und Brau empfehlen in ihrem Buch, die Auswertung direkt in einer Gruppe vorzunehmen, da somit eine Diskussion entstehen kann, welche das Ergebnis aussagekräftiger machen soll. Wie sich jedoch in der Praxis zeigt, kann es schwierig sein, alle Evaluatoren zu einer Stellungnahme zu bewegen. Vgl. [SaBr 2011] S. 224

Auf Grund dessen ist es fraglich, ob eine Bewertung in der Gruppe nicht zu Lasten der Verlässlichkeit geht, die durch eine möglichst hohe Teilnehmerzahl an Evaluatoren gegeben ist.

Nielsen schlägt nämlich eine Auswertung in Form von Fragebögen vor, welche nach der Evaluation an alle Beteiligten geschickt werden. Er weist explizit darauf hin, dass die Evaluatoren ihre Bewertung individuell und

unabhängig voneinander vornehmen sollten. Ausgehend davon, dass die Evaluatoren das Interface bereits kennen, brauchen die Teilnehmer durchschnittlich 30 Minuten zur Bewertung. Vgl. [Nie 1994] S. 48

4.2.4 Debriefing

Das Debriefing wird im Anschluss an die Evaluationsdurchgänge abgehalten und kann optional durchgeführt werden. Damit bietet sich die Möglichkeit, die Heuristische Evaluation um konkrete Designvorschläge zu erweitern, statt nur auf Probleme hinzuweisen, ohne konkrete Hilfestellung zu geben. Im Idealfall sind alle Personen, die an der Evaluation beteiligt waren, anwesend, das betrifft sowohl die Evaluatoren als auch den Versuchsleiter. Ziel des Debriefings ist die Diskussion des Interfaces und die potenzielle Verbesserung der gefundenen Usability Probleme. Um mögliche Verbesserungen direkt zu adressieren, sind ebenfalls die Entwickler und Designer der Anwendung am Debriefing beteiligt. Die gewonnenen Daten der Evaluation sollen im Debriefing dazu genutzt werden, bereits erste konkrete Verbesserungsvorschläge bezüglich des Designs zu erarbeiten. Ebenso bedeutsam ist es auch, die von den Evaluatoren als positiv empfundenen Aspekte des Designs zu benennen, da sie in der Heuristischen Evaluation an sich nicht beachtet werden. Vgl. [Nie 1994] S. 32, S. 41 ff.

4.3 Adaption der Methode

Testteilnehmer

Zur Durchführung der Heuristischen Evaluation wurden fünf Expertinnen gesucht, welche die zu untersuchende Anwendung im Hinblick auf zehn vorgegebene Heuristiken prüfen sollten. Als Kriterium für die nötige Expertise wird gute Kenntnis der Heuristischen Evaluation als Erhebungsmethode, sowie der einzelnen Heuristiken vorausgesetzt.

4. Spezifikation der Evaluation

Bei den Expertinnen handelt es sich um Studentinnen der Informatik, welche kurz vor ihrem Bachelorabschluss stehen. Sie zeichnen sich dadurch als Expertinnen aus, da sie bereits erfolgreich an einem Usability Seminar teilgenommen haben und somit die nötige Kenntnis der Heuristischen Evaluation mitbringen.

Langjährige Berufserfahrung im Bereich der Usability war nicht gefordert, sondern im Gegenteil eher unerwünscht. Da sich die Anwendung an BerufseinsteigerInnen in der IT richtet, kommt es somit zu einer Überschneidung von Expertinnen und Zielgruppe, wodurch sich mehr Ergebnis-Vielfalt in der Evaluation ergeben könnte.

Zudem muss bei jedem Evaluationsdurchgang eine Versuchsleiterin anwesend sein, welche die Anwendung gut kennt und den Expertinnen für Fragen und Hilfestellung zur Verfügung steht. Diese Rolle wurde durch die Verfasserin der Arbeit selbst ausgefüllt.

4.3.1 Vorevaluatives Training

Da alle durchführenden Expertinnen erst kürzlich ihre Kenntnisse in Usability und Heuristischer Evaluation erlangten, wird auf eine umfangreiche Einführung in die Methodik verzichtet. Auch die Einführung in die Anwendung an sich wird sehr kurz gehalten. Durch die Überschneidung von Zielgruppe und Expertinnen ist der unvoreingenommene Blick auf die Anwendung eine gute Möglichkeit, zu sehen, wie intuitiv bedienbar und verständlich diese ist. Außerdem ist die Anwendung sehr überschaubar, wodurch auf eine umfangreiche Einführung und die Vorgabe eines Szenarios verzichtet werden kann. Anstelle eines gemeinsamen Trainings wird jede Expertin einzeln über Sinn und Zweck, sowie Ablauf der Evaluation aufgeklärt. Sie werden darüber informiert, dass sie mindestens zwei Durchläufe machen sollen, um sich mit der Anwendung insgesamt vertraut zu machen und um anschliessend die einzelnen Dialogelemente genauer zu prüfen. Augenmerk wird vornehmlich auf eine entspannte Atmosphäre gelegt und einen guten Austausch zwischen Expertin und Versuchsleiterin. Die

4. Spezifikation der Evaluation

Expertinnen werden darüber aufgeklärt, dass sie zu jedem Zeitpunkt Rückfragen an die Versuchsleiterin stellen dürfen. Zur besseren Einordnung der Heuristiken bekommt jede Expertin ein Merkblatt, auf dem alle Heuristiken in kurzen Sätzen erklärt sind:

1. Ästhetik und minimalistisches Design

Interfaces sollten so einfach wie möglich gehalten sein, es sollte nur die Information bereitgestellt werden, die der User benötigt. Zusammenhängende Elemente sollten sichtbar sein. Die Gestaltung des Interface sollte den User in Effektivität und Effizienz unterstützen.

2. Anpassung zwischen System und reeller Welt

Das System sollte in Wort und Phrase die natürliche Sprache des Users sprechen und nicht des Systems.

3. Wiedererkennen statt sich erinnern

Die Bedienung sollte weitestgehend selbsterklärend sein. Das System sollte Eingaben des Users speichern und bereitstellen.

4. Konsistenz und Standards

Benennung und Position von Dialogelementen sollte auf allen Seiten konsistent sein. Gleiche Computerbefehle sollten immer den selben Effekt haben.

5. Sichtbarkeit des Systemstatus

Das System sollte den User ständig darüber informieren, was es tut und wie es Usereingaben interpretiert.

6. Benutzerkontrolle und –freiheit

Der User sollte ständig die Möglichkeit haben, einen Vorgang zu verlassen, z.B. durch Cancel- oder Undo-Funktion.

7. Flexibilität und Effizienz

4. Spezifikation der Evaluation

Erfahrene User sollte die Möglichkeit haben, die Benutzung des Systems zu beschleunigen, mit Hilfe von Shortcuts.

8. Hilfe beim Erkennen, Diagnostizieren und Beheben von Fehlern

Fehlermeldungen sollten in natürlicher Sprache, präzise, konstruktiv und positiv (höflich) formuliert sein.

9. Fehler vermeiden

Das System sollte den User durch Auswahlmöglichkeiten und Wiederbestätigung (nach Eingabe des Users) bei der Fehlervermeidung unterstützen.

10. Hilfe und Dokumentation

Das System sollte bei Bedarf Hilfe anbieten. Hilfe und Dokumentation sollte in ansprechender Form, verständlich aufbereitet und lösungsorientiert sein. Vgl. [Nie 1995]

4.3.2 Evaluationsdurchgänge

Jeder Evaluationsdurchgang dauert 1 - 1 1/2 Stunden und wird jeweils unabhängig voneinander durchgeführt. Die Expertinnen bekommen einen Laptop gestellt, an welchem sie die Anwendung prüfen können.

Anstelle einer klassischen Heuristischen Evaluation wird eine Mischung aus dieser und dem Simplified Thinking Aloud durchgeführt. Dies ermöglicht es, sowohl Probleme ausfindig zu machen, als auch direkte Verbesserungsvorschläge zu erfassen. Da es in diesem Fall zu einer Überschneidung von Expertinnen und tatsächlichen Userinnen der Zielgruppe kommt, können die Äusserungen wertvolle Erkenntnisse für die Auswertung liefern. Der Austausch zwischen Versuchsleiterin und Expertin befähigt die Expertinnen auf natürliche Weise, Vorschläge zu machen und laut über Probleme nachzudenken. Neben dem Auffinden der Probleme dürfen sie auch aufzeigen, ob ihnen etwas besonders gut gefällt. Alle Anmerkungen werden der Versuchsleiterin gegenüber

4. Spezifikation der Evaluation

verbalisiert und von dieser in einem Formular erfasst. Das Formular sieht bereits eine Einordnung in die Heuristiken vor, welche aber noch keine endgültige Aussagekraft hat und im Zuge der Auswertung angepasst werden kann. Die Einordnung wird bei Bedarf gemeinschaftlich von Expertin und Versuchsleiterin vorgenommen.

Die Entscheidung, dass die Versuchsleiterin die Protokolle jeder Sitzung anfertigt, wurde bewusst getroffen. Wie Nielsen bereits schreibt, macht es die Auswertung am Ende einfacher, da anstelle vieler verschiedener Aufzeichnungen nur ein Set an Protokollen aus einer Hand aufgearbeitet werden muss. Die Ergebnisse sind somit schneller verfügbar und leichter zu erfassen. Vgl. [Nie 1994] S. 27 Somit ist für das Protokoll zwar keine strenge Struktur vorgegeben, da die Anfertigung und Auswertung der Protokolle von der selben Person durchgeführt wird, es hat sich aber folgendes Vorgehen abgezeichnet:

- Auswahl der Heuristik
- Benennung des Fundortes (Startseite, Stories oder Form)
- ggf. wird die genaue Stelle benannt (z.B. das genaue Formularfeld)
- Problem dokumentieren
- ggf. Verbesserungsvorschlag dokumentieren

4.3.3 Auswertung und Kategorisierung der Ergebnisse

Zur Auswertung der Ergebnisse wird zunächst eine Liste aller gefundenen Probleme erstellt. Hierbei ist darauf zu achten, dass es keine Überschneidungen gibt und die gleichen Probleme der einzelnen Fragebögen aggregiert werden. Aufgrund der Überschaubarkeit der Anwendung wird eine hohe Granulation gewählt. Die Probleme werden zunächst in die einzelnen Seiten der Anwendung aufgeteilt (Startseite, Stories, Formular), sowie in einen Abschnitt, der die Anwendung allgemein betrifft. Alle Probleme, die im Formular aufgetreten sind, werden wiederum den einzelnen Dialogfeldern oder dem Formular allgemein zugewiesen.

4. Spezifikation der Evaluation

Beim Erstellen dieser Liste wird auf eine Wertung der Probleme verzichtet, sie dient zunächst lediglich dazu, alle gefundenen Probleme zu sortieren. Es werden alle gefundenen Probleme aufgenommen und nach Möglichkeit zusammengefasst, jedoch wird auf die Dokumentation aller Probleme bezüglich Rechtschreibung und Grammatik verzichtet, da sich die Anwendung noch im Entwicklungsstadium befindet und den Expertinnen hier das Fachwissen der englischen Sprache fehlt.

Des Weiteren werden erbrachte Verbesserungsvorschläge, die sich durch das Thinking Aloud ergeben haben, ebenfalls in der Liste notiert, sowie die Häufigkeit des Auftretens eines Problems. Da die Einteilung der Probleme in die einzelnen Heuristiken bereits während der Evaluation erfolgt ist, kommt es hier zu einer Redundanz, da die Expertinnen die Probleme teilweise unterschiedlich kategorisiert hatten. An dieser Stelle werden von der Versuchsleiterin Anpassungen vorgenommen, um eine einheitliche Konsistenz zu erhalten. Auch die betreffende Heuristik wird anschliessend in die Gesamtliste aufgenommen.

Alle beteiligten Expertinnen haben sich bereit erklärt, eine Beurteilung des Schweregrads vorzunehmen. Dieser Schritt erfolgt mit zeitlichem Abstand zur eigentlich Evaluation per Mail. Jede der Teilnehmerinnen bekommt die Gesamtliste aller gefundenen Probleme zugeschickt und gibt eine persönliche Einschätzung ab. Die Ergebnisse werden aggregiert und es wird ein Mittelwert seitens der Versuchsleiterin gebildet. In der Regel wird hierbei eine Mehrheitsentscheidung getroffen, je nach Verteilung des Schweregrads und Divergenz der Einschätzung werden jedoch Abstufungen gemacht. Des Weiteren ist eine unterschiedliche Bereitschaft zur hohen Risikobewertung bei den Expertinnen zu erkennen. So geht beispielsweise die Bewertung des Schweregrads 4 (Usability Katastrophe) bei fünf Teilnehmerinnen weit auseinander. Bei drei der Expertinnen wurde dieser Schweregrad zweimal bis neunmal vergeben, zwei weitere Teilnehmerinnen vergaben den Schweregrad 22 und 32 Mal. Auch dieser Aspekt wird im Zweifelsfall bei der Aggregation der Ergebnisse berücksichtigt.

4.3.4 Debriefing

Das Debriefing findet mit etwas zeitlichem Abstand zur Auswertung und Kategorisierung der gefundenen Probleme statt. Teilnehmende Parteien sind die beiden Entwicklerinnen der Anwendung, sowie die Versuchsleiterin der Evaluation. Als Grundlage einer Diskussion stellt die Versuchsleiterin die Ergebnisse der Evaluation vor, wobei das Hauptaugenmerk auf den schwerwiegendsten Problemen des Schweregrads 4 liegt. Vorliegende Verbesserungsvorschläge aus der Evaluation werden vorgestellt, aus der Diskussion jedes Problems ergeben sich jedoch ebenfalls neue Ideen und Verbesserungen. Ziel des Debriefings ist es, die besprochenen Probleme in github Issues zu übertragen, um die Problembehebung und eine Weiterentwicklung der Anwendung zu gewährleisten. Eine genauere Beschreibung des Debriefings findet sich in Abschnitt 6.3.

5. Auswertung der Testergebnisse

Nachdem die Evaluation durchgeführt und ausgewertet wurde, erfolgt im folgenden Kapitel die Präsentation der Ergebnisse. Zunächst werden in Abschnitt 5.1 die tatsächlichen Ergebnisse in Form von gefundenen Problemen pro Heuristik mit der vorangegangen These verglichen. Anschliessend werden die schwerwiegendsten Probleme genauer benannt, wobei in Abschnitt 5.2 alle Probleme des Schweregrad 4 besprochen werden und in Abschnitt 5.3 die Probleme, welche die neu erstellen Heuristiken betreffen.

5.1 Gegenüberstellung der Heuristiken

Die folgende Abbildung 11 zeigt die Gesamtverteilung der Heuristiken. Hierbei lässt sich der prozentuale Anteil je Heuristik ablesen und wieviele Probleme jeweils darunter fallen.

5. Auswertung der Testergebnisse

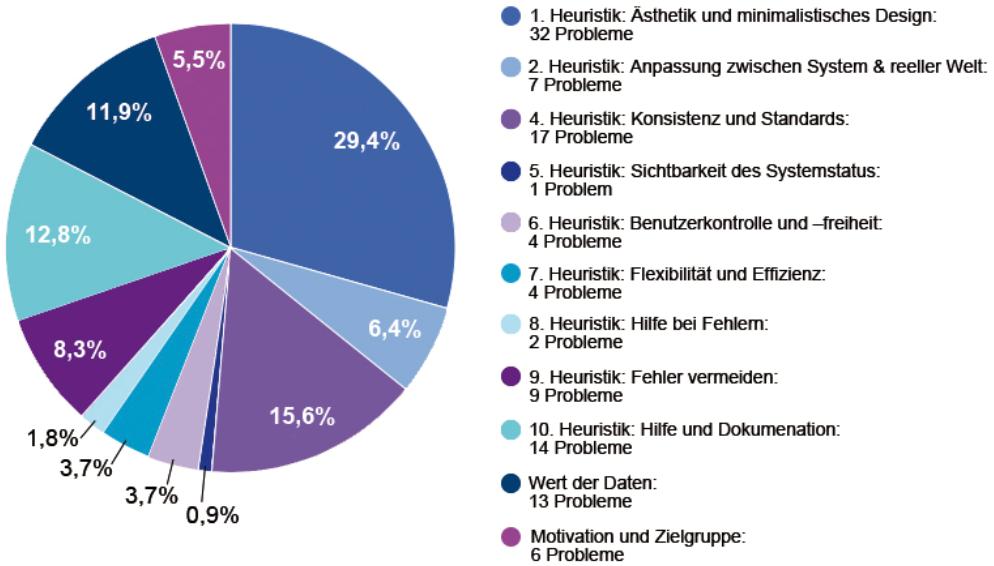


Abbildung 11: Verteilung der Heuristiken [eigene Abb.]

Im Gegensatz zur aufgestellten These, dass die gefundenen Probleme zu zwei Dritteln das Design und somit die erste Heuristik, *Ästhetik und minimalistisches Design*, betreffen werden, stellt es sich so dar, dass fast alle Heuristiken verletzt werden und sich relativ ausgewogen verteilen. Insbesondere wenn die Bewertung des Schweregrads berücksichtigt wird, zeigt sich, dass gerade die Heuristiken, die insgesamt wenig betroffen sind, die schwerwiegendsten Probleme aufweisen. Zwar macht die erste Heuristik (*Ästhetik und minimalistisches Design*) gut ein Viertel der gefundenen Probleme aus, und ist somit auch am meisten betroffen, weist jedoch nur sechs Probleme des Schweregrads 3 (Bedeutendes Usability Problem) und 4 (Usability Katastrophe) auf, ebenso wie die vierte Heuristik (*Konsistenz und Standards*). Es ist aber anzumerken, dass diese beiden Heuristiken damit bereits den Spitzensatz an schwerwiegenden Problemen innehaben. Betrachtet man den Schweregrad 4 (Usability Katastrophe) jedoch gesondert, zeigt sich, dass fünf Probleme mit diesem Schweregrad bewertet wurden, wobei diese sich gleichmäßig auf die

5. Auswertung der Testergebnisse

Heuristiken 1, 4, 5, 6 und 7 verteilen. Abbildung 12 zeigt die Verteilung des Schweregrads innerhalb der gesamten Anwendung. Es ist zu erkennen, dass der Hauptanteil der gefundenen Probleme auf den Schweregrad 2 fällt, was einem Usability Problem mit untergeordneter Priorität entspricht. Die insgesamt 34 Probleme mit dem Schweregrad 3 und 4 hingegen sind schwerwiegend und sollten sofort gelöst werden. Wie die Gesamtverteilung des Schweregrads pro Heuristik ausfällt, lässt sich aus Abbildung 13 entnehmen.

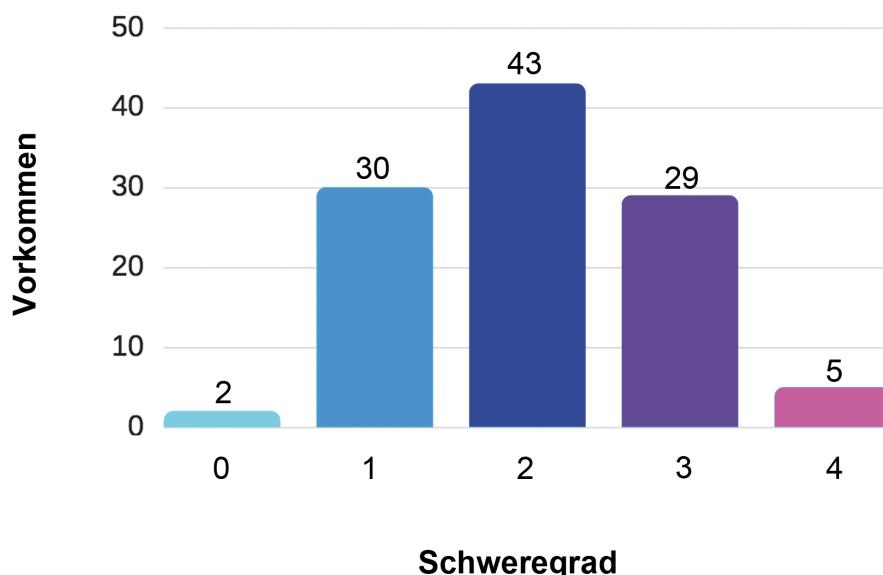


Abbildung 12: Verteilung des Schweregrads innerhalb der gesamten Anwendung [eigene Abb.]

5. Auswertung der Testergebnisse

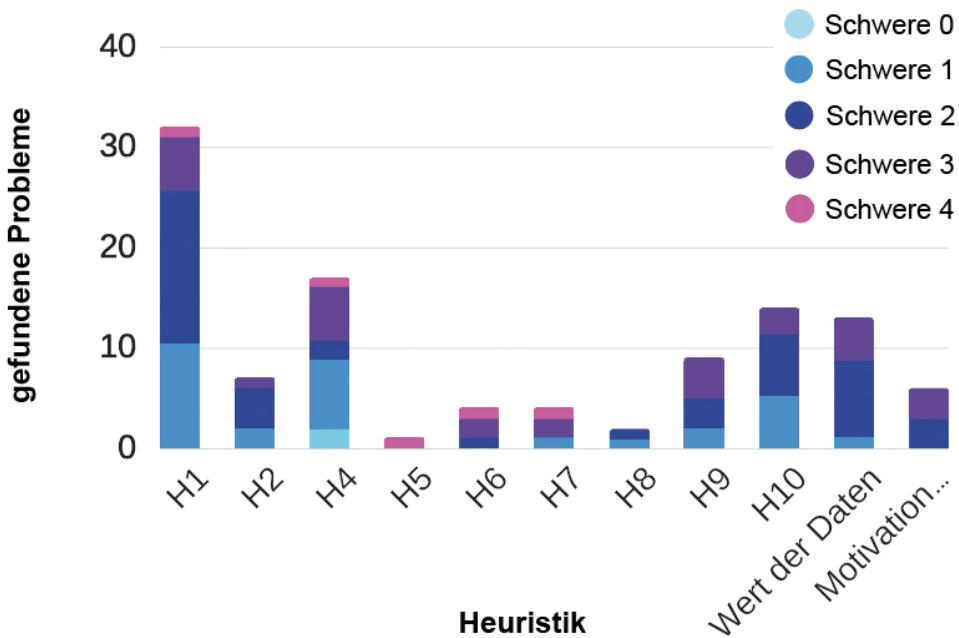


Abbildung 13: Verteilung des Schweregrads pro Heuristik [eigene Abb.]

5.2 Probleme des Schweregrad 4

Wie in Abschnitt 4.2.3 beschrieben, müssen Probleme mit dem Schweregrad 4 in jedem Fall behoben werden, da sie einer Usability Katastrophe gleichkommen. Alle diese Probleme waren derart auffällig, dass sie von fast allen Expertinnen gefunden und durchgehend sehr hoch im Schweregrad beurteilt wurden. Im Folgenden werden alle fünf dieser Probleme genauer beschrieben.

5.2.1 Fehlende Information auf der Startseite

Problembeschreibung

Das erste Problem mit dem Schweregrad 4 betrifft die erste Heuristik, *Ästhetik und minimalistisches Design*. (siehe Anhang S. 5, #5) In diesem Fall ist es weniger ein überladenes Design, was den User zu erschlagen droht, sondern die Startseite ist zu minimal gehalten. Wie in Abbildung 2

5. Auswertung der Testergebnisse

zu sehen ist, befinden sich auf der Startseite der Anwendung nur der Titel und zwei Buttons. Weitere Informationen werden vermisst. Bereits in den Evaluationsdurchgängen zeigte sich, dass die User auf den ersten Blick nicht verstehen, worum es in dieser Anwendung geht und was vom User verlangt wird. Die Anwendung musste explorativ durchgeklickt werden, um den Sinn und Zweck der Anwendung zu begreifen. Dieser Umstand kann sich sehr negativ auswirken, denn User entscheiden innerhalb der ersten Sekunden, ob sie auf einer Webseite verweilen oder nicht.

Eine Studie von Chao Liu et al. zeigt, dass beim Surfen im Netz das Phänomen des „negative aging“ zu erkennen ist, welches besagt, dass die Abbruchrate von Usern einer Webseite über die Zeit abnimmt. Im Gegenzug heißt das jedoch, dass die ersten 10 Sekunden eines Besuchs für die Webseite besonders kritisch sind. Dies trifft insbesondere bei Webseiten zu, die nicht auf Unterhaltung ausgelegt sind, sondern beispielsweise Themen wie Computer und Bildung betreffen. User sind mit Webseiten dieser Kategorie weniger gnädig und beurteilen diese in den ersten Sekunden härter. Vgl. [LiuWhDu 2010] Nielsen greift diese Erkenntnisse auf und stellt sie wie folgt dar:

5. Auswertung der Testergebnisse

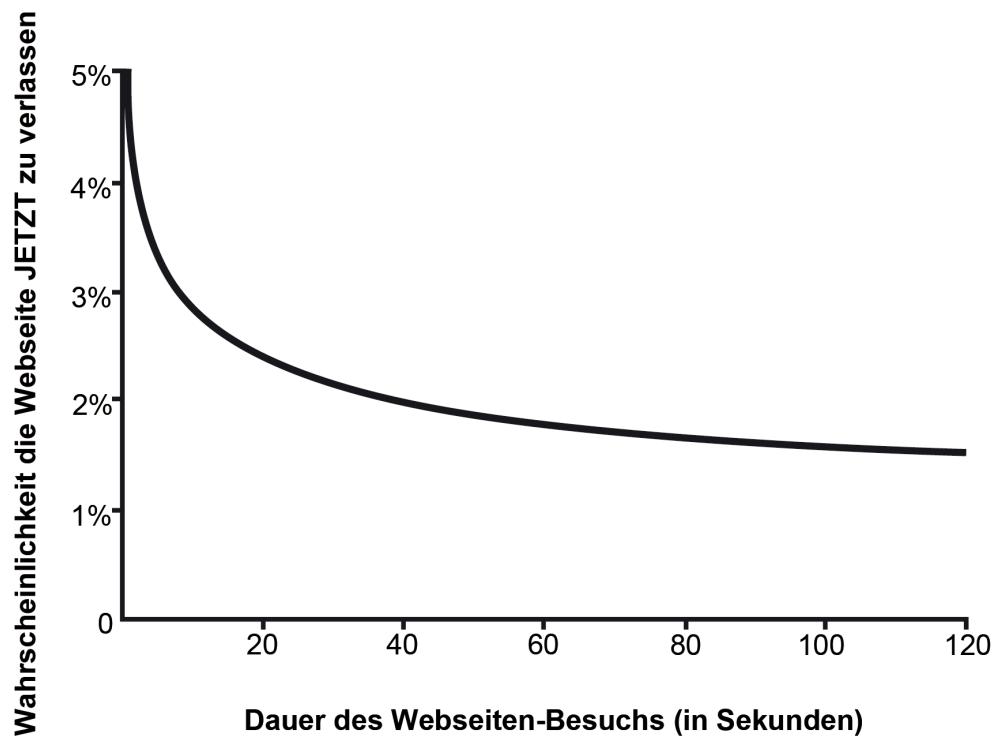


Abbildung 14: „Negative aging“ Effekt der Abbruchrate einer Webseite nach [Nie 2011] [eigene Abb.]

User seien auf Grund von Erfahrungen mit schlecht designten Webseiten eher skeptisch. „To gain several minutes of user attention, you must clearly communicate your value proposition within 10 seconds.“ [Nie 2011] Weiter beschreibt er, dass User mit zunehmender Erfahrung rücksichtsloser in ihrer Beurteilung werden. Vgl. [NieLo 2006] S. 28

Bei der anvisierten Zielgruppe der Anwendung handelt es sich um sehr erfahrene und computeraffine User. Je nach Auslegung der Zielgruppe sogar um Designer und Usability-Experten. Es ist also anzunehmen, dass diese ein besonders kritisches Auge auf die Anwendung haben werden und, wie von Nielsen prognostiziert, weniger wohlwollend mit dem Design umgehen werden. Erschwerend kommt hinzu, dass es sich hier um eine freiwillige Anwendung handelt, die auf die Mithilfe von Usern angewiesen ist. Ziel ist es, User dazu zu bewegen, ihre Daten zur Verfügung zu stellen, um einen Mehrwert für die Gemeinschaft zu bieten und andere Junior-Entwickler von ihren Daten profitieren zu lassen. Insbesondere die

5. Auswertung der Testergebnisse

Startseite muss also derart gestaltet sein, dass genügend Anreiz gegeben ist, User zur Bereitstellung von Daten zu bewegen.

Besonders, wenn die Startseite keinerlei Information über ihren Zweck bereitstellt, wird der User der Chance zur Teilnahme beraubt, da er gar nicht weiß, was von ihm verlangt wird.

Verbesserung

Auf Grund der Kürze der Zeit, die ein User initial auf der Webseite verbringt, müssen „[...] Botschaften ultraschlank sein und den Nagel auf den Kopf treffen.“ [NieLo 2006] S. 28

Nielsen formuliert folgende Punkte die eine Webseite dem User innerhalb der ersten 30 Sekunden bieten muss:

- auf welcher Webseite befindet sich der User
- welche Vorteile bietet die Webseite/Organisation
- Information über die Organisation (z.B. Produkte, Entwicklungen)
- Wahlmöglichkeiten des Users und in welcher Form er durch die Webseite navigieren kann Vgl. [NieLo 2006] S. 28

Das klingt nach relativ viel Inhalt, soll jedoch derart gestaltet sein, dass User nicht viel Text lesen müssen. Nielsen beschreibt einen Richtwert von 10-20 Wörtern auf einer Webseite als realistischen Wert, um User zu informieren und gleichzeitig nicht zu überfordern. Vgl. [NieLo 2006] S. 28

Auch Steve Krug sagt User überflögen Webseiten nur anstelle sie zu lesen. Vgl. [Krug 2006] S. 22

Ähnlich wie bei einer Zeitschrift sei es hilfreich, User mit einem Aufhänger zu locken. Dies kann beispielsweise ein beliebtes Feature sein oder ein besonderes Leistungsmerkmal. Vgl. [Krug 2006] S. 96

Denkbar wäre es auch, den Aufhänger visuell zu gestalten anstelle von Text. So schlugen die Expertinnen während der Evaluation vor, die Bedienung der Webseite in Form einer Bildreihe bzw. eines Comicstrips darzustellen. Ein weiterer Vorschlag war es, die „most recent stories“

5. Auswertung der Testergebnisse

anzuzeigen, damit der User ein Gefühl dafür bekommt, was ihn auf der Webseite erwartet.

5.2.2 Fehlende Filtermöglichkeiten

Problembeschreibung

Das zweite Problem des Schweregrads 4 betrifft die siebte Heuristik, *Flexibilität und Effizienz* (siehe Anhang S. 5, #13). Alle fünf befragten Expertinnen empfanden es als problematisch, dass in der Darstellung der Daten die gesuchten Informationen schwierig zu finden seien. Wie in Abschnitt 2.2 beschrieben, werden diese als Stories in einem Fließtext angezeigt. Der User müsste sich demnach durch alle Stories durchlesen, um die für ihn relevanten Informationen herauszufinden. Nielsen beschreibt, Interfaces sollten zwar generell für den durchschnittlichen User ohne spezielle Kenntnisse benutzbare sein, jedoch sollten erfahrene User die Möglichkeit haben, Vorgänge auch schneller zu bearbeiten. Vgl. [Nie 1993] S. 139

Dazu zählt nicht nur das Bereitstellen von Shortcuts, sondern auch die individuelle Anpassbarkeit von Interfaces, um das System für den einzelnen User effizienter nutzbar zu machen. Vgl. [Nie 1994B] S. 153

Verbesserung

Eine einfache Lösung wäre das optische Hervorheben bestimmter Schlagwörter. Jedoch wäre das keine individuelle Lösung für den einzelnen User. Sie würde auf Mutmaßungen beruhen, was der User gern sehen möchte und wäre standardmäßig für alle User gleich. Jedoch könnte man somit wenigstens das Gehalt deutlich hervorheben, denn wie einige Expertinnen anmerkten, ist gerade diese entscheidende Information schwer in den Stories zu finden.

Eine individuelle Lösung wäre es, Filtermöglichkeiten einzubauen, wie auch von den Expertinnen in der Evaluation gewünscht. Der Vorschlag war es, mehrfache Filter einzubauen, nach denen der User seine

5. Auswertung der Testergebnisse

gewünschten Ergebnisse eingrenzen kann. Ebenso war gewünscht, die Daten analysieren zu können, in Form visueller Graphen, so dass der User z.B. mögliche Gehaltsunterschiede zwischen Geschlechtern, Hautfarbe, Stadt oder Programmiersprachen erkenne kann. Hierzu wurde auch eine Durchschnittsberechnung aller Daten gewünscht.

5.2.3 Fehlende System-Rückmeldung

Problembeschreibung

Ein weiteres Problem des Schweregrads 4 besteht darin, dass User nach Abschicken des Formulars keine Rückmeldung des Systems bekommen (siehe Anhang S. 7, #34). Im Falle des erfolgreichen Absendens des Formulars und dementsprechend erfolgreicher Bereitstellung der eigenen Daten werden die User automatisch auf die Seite mit den Stories weitergeleitet. Dieses Problem fällt unter die fünfte Heuristik, *Sichtbarkeit des Systemstatus*. Nielsen schreibt dazu, ein System solle in jedem Fall Feedback bereitstellen und das auch, bevor ein Fehler auftaucht. So soll auch positives Feedback und anteiliges Feedback während einer Usereingabe gegeben werden. Bekommt der User keine Rückmeldung des Systems, ist es für ihn schwer nachzuvollziehen, ob das System überhaupt arbeitet oder eventuell abgestürzt ist. Weiter führt er aus, das Feedback solle nicht generisch formuliert sein, sondern zur Usereingabe passen und diese erneut aufgreifen und formulieren. Dies dient nicht nur zur Information, sondern auch dazu, den User vor ungewollten Handlungen zu schützen. Als Beispiel führt er hier das Löschen einer Datei an. Gutes Feedback erkläre dem User so explizit wie möglich, was passiert und das bereits vor der endgültigen Ausführung, um Fehler zu vermeiden. Vgl. [Nie 1993] S. 134

Im Falle der vorliegenden Anwendung merkt der User durch die automatische Weiterleitung zwar, dass etwas passiert - es ist also auszuschliessen, dass das System abgestürzt sein könnte - in welcher Form die Formulareingaben jedoch gespeichert wurden, bleibt unbekannt.

5. Auswertung der Testergebnisse

Erschwerend kommt hinzu, dass die Usereingaben auf der Stories-Seite nicht direkt zu sehen sind, da die Stories chronologisch geordnet sind und die neueste Story am Seitenende landet.

Verbesserung

Ein Popup-Fenster oder einen Notification-Banner, d.h. eine Einblendung über die gesamte Breite am oberen Seitenrand, mit einer Benachrichtigung über das erfolgreiche Einreichen der Daten, wäre wohl die naheliegendste Verbesserung, jedoch lässt sich das beschriebene Problem mit einem anderen Problem des Schweregrads 3 verbinden. Dies betrifft die fehlende Wiederbestätigung seitens des Users, bevor die Daten eingereicht werden (siehe Anhang S. 7, #33). Dieses Problem wiederum fällt unter die neunte Heuristik, *Fehler vermeiden*. Obgleich beide Probleme verschiedene Heuristiken betreffen, ist eine Überschneidung der einzelnen Heuristiken zu erkennen, denn beide Heuristiken sollen den User unter Anderem vor Fehleingaben schützen. Die Wiederbestätigung hat im besten Fall jedoch den Vorteil, dass User ihre Eingaben nochmal überprüfen und ändern könnten. Es geht also nicht nur darum zu entscheiden, ob die Daten wirklich eingereicht werden sollen, sondern ermöglicht es auch, einzelne Fehler im Formular zu korrigieren. Um das umzusetzen, gibt es verschiedene Möglichkeiten. Zum einen könnte ein einfaches Popup-Fenster bereitgestellt werden, welches den User um Wiederbestätigung bittet, zum anderen könnte der User durch die Betätigung des Absende-Buttons auf eine Seite weitergeleitet werden, wo die Usereingaben noch einmal dargestellt werden und gegebenenfalls geändert werden können. Diese Art der Vorschaufunktion findet sich z.B. oft bei Bestellvorgängen oder Formulareingaben im Internet und gibt dem User Kontrolle über die eigenen Eingaben. Beide Varianten wurden von den Expertinnen während der Evaluation vorgeschlagen und sind je nach Verfügbarkeit der Ressourcen umzusetzen.

Obgleich eine Expertin während der Evaluation der Meinung war, durch die Wiederbestätigung des Users sei die Rückmeldung des Systems

5. Auswertung der Testergebnisse

eventuell überflüssig, da selbsterklärend, sollte dennoch zusätzlich positives Feedback über das erfolgreiche Einreichen der Daten erfolgen. Es sei wichtig, Konversationen reibungslos und sanft ablaufen zu lassen und am Ende eine positive Rückmeldung für eine weiterhin gute Beziehung zu gewährleisten. Im besten Fall erreicht man das mit einer positiven Anerkennung für die Teilnahme z.B. einer Danksagung. Vgl. [JarGaf 2009] S. 115

5.2.4 Fehlender Zurück-Button

Problembeschreibung

Ein weiteres Problem mit dem Schweregrad 4, welches die gesamte Anwendung betrifft, ist das Fehlen eines Zurückbuttons (siehe Anhang S. 15, #103). Dieses Problem fällt in die sechste Heuristik, *Benutzerkontrolle und -freiheit*. User möchten die Kontrolle über ein System haben und selbst bestimmen, ob und wann sie einen Vorgang verlassen. Vgl. [Nie 1993] S. 138 Da es auf keiner Seite der Anwendung einen Zurückbutton gibt, muss der User sich mit Hilfe des Browserbuttons durch die Seiten navigieren. Besonders kritisch ist dies, nachdem der User vom Formular auf die Stories-Seite weitergeleitet wurde. An dieser Stelle gibt es keinen direkten Weg mehr zur Startseite. Durch die Browernavigation gelangt der User zunächst wieder auf die Formular-Seite. Besonders kritisch ist hier anzumerken, dass alle Daten noch im Zwischenspeicher sind und somit im Formular sichtbar sind. Ein Problem, welches bei der Evaluation nicht gefunden wurde, jedoch aus Datenschutzgründen ebenfalls behoben werden sollte.

Verbesserung

Wie von den Expertinnen gewünscht, sollte eine Navigationsleiste bereitgestellt werden, die das einfache Navigieren auf jede Seite der Anwendung mit einem Klick möglich macht. Auch hier kann die Lösung dieses Problems mit einem anderen Problem kombiniert werden. Die

5. Auswertung der Testergebnisse

Expertinnen wünschten sich generell einen allgemeinen Rahmen für die Webseite (Header und Footer), wie sie es von anderen gängigen Webseiten gewohnt sind (siehe Anhang S. 15, #104). Dieses Problem wurde der vierten Heuristik, *Konsistenz und Standard* zugeordnet.

5.2.5 Fehlendes Impressum

Problembeschreibung

Das letzte gefundene Problem mit dem Schweregrad 4 besteht darin, dass innerhalb der Anwendung kein Impressum vorhanden ist (siehe Anhang S. 15, #105). Dieser Umstand ist zum einen rechtlich nicht haltbar, denn das Gesetz sieht eine Impressumspflicht für Webseiten vor Vgl. [TMG 2007] S. 4 f. Zum anderen mindert ein fehlendes Impressum und somit fehlende Transparenz auf der Webseite das Vertrauen der User. Zum aktuellen Stand ist es den Usern nicht ersichtlich, wer hinter der Webseite steht. Es gibt somit keine Möglichkeit, mit den Betreiberinnen der Webseite in Kontakt zu treten. Eingeordnet wurde dieses Problem in die vierte Heuristik, *Konsistenz und Standards*. Hierbei geht es weniger um die Konsistenz von Seiteninhalten innerhalb der gesamten Anwendung, sondern um einen Standard, den User von anderen Webseiten kennen und auch auf der vorliegenden Anwendung erwarten. Nielsen schreibt dazu, dass Interface Standards den User befähigen, Handlungen von einem System auf ein anderes zu übertragen. Vgl. [Nie 1993] S. 227

Verbesserung

Wie im echten Leben würdigten User solche Internetseiten, die offen mit ihren Informationen umgehen und diese im Vorfeld klar und transparent darstellen. Darunter fielen nicht nur Kontaktinformationen, wie man sie in einem Impressum erwartet, sondern beispielsweise auch FAQ's und AGB's. Der User sollte also nicht das Gefühl haben, nach diesen Informationen suchen zu müssen, sie sollten gut sichtbar auf der Webseite

5. Auswertung der Testergebnisse

platziert werden. Nielsen schlägt hierfür eine Utility Navigation¹³ im Header vor. Vgl. [Nie 2016] Auch denkbar und eventuell bevorzugt wäre die Darstellung im Footer. Beide Varianten sind im Internet zu finden und bilden einen konsistenten Standard. Wie jedoch in Abschnitt 5.2.4 bereits beschrieben, wünschten sich die Expertinnen einen allgemeinen Rahmen mittels Header und Footer, somit wäre die geforderten Informationen an dieser Stelle gut aufgehoben. Der Footer sollte zunächst als Minimalanforderung das Impressum beinhalten und kann bei Bedarf um weitere gewünschte Informationen erweitert werden.

5.3 Probleme der neue Heuristiken

Neben den Problemen des Schweregrad 4 wurden zahlreiche weitere Probleme gefunden, die ebenfalls mit einer hohen Priorität behoben werden sollten. Im Zuge der Evaluation wurden gemeinschaftlich mit den Expertinnen zwei neue Heuristiken gebildet. Die Heuristik *Motivation und Zielgruppe* hinterfragt die Motivation und den Anreiz für die User, an der Umfrage der Anwendung teilzunehmen. Von den Expertinnen vermutet, wird die Motivation durch die unklar formulierte Zielgruppe negativ beeinflusst. Bei der Heuristik *Wert der Daten* geht es um fehlende Informationen, die der Anwendung und den erhobenen Daten aus dem Fragebogen einen zusätzlichen Mehrwert bieten könnten.

5.3.1 Zielgruppe ist unklar

Problembeschreibung

Ein Problem, welches an mehreren Stellen des Formulars auftritt und der Mehrheit der Expertinnen aufgefallen ist, betrifft die unklare Formulierung der Zielgruppe (siehe Anhang S. 16). Wenn ein User nicht sicher ist, ob die Anwendung für ihn gemacht ist und ob er überhaupt zur Zielgruppe

¹³ Sekundär Navigation, die z.B. geschäftliche Informationen bereitstellt, eine Sprachauswahl oder eine Login-Option

5. Auswertung der Testergebnisse

gehört, wird er vermutlich nicht an der Umfrage teilnehmen. Noch schlimmer ist es, wenn sich User durch bestimmte Wortwahl explizit von der Anwendung ausgeschlossen fühlen. Möglicherweise haben sie sich bereits für die Teilnahme an der Umfrage entschieden und merken beim Ausfüllen des Formulars, dass die Formularfelder nicht zu ihrer Profession passen. Den Expertinnen sind während der Evaluation einige dieser Fallstricke aufgefallen. So ist beispielsweise durch das unterschiedliche Wording von Junior, Junior-Developer und Developer beispielsweise nicht wirklich klar, wer die Zielgruppe ist.

Verbesserung

Im ersten Schritt sollte die Zielgruppe bereits bei der ersten Benutzung der Anwendung sichtbar sein. Dies ließe sich durch eine einfache Definition auf der Startseite sicherstellen. Je nachdem, wie die Zielgruppe der Anwendung definiert wird, sollten ebenfalls die betreffenden Formularfelder so abgeändert werden, dass z.B. zusätzliche Arbeitsschwerpunkte ebenfalls abgebildet werden, wie beispielsweise UI/UX-Design, Produktmanagement oder Administration, und der User sich nicht unter einem generischen Term, wie „other“ einordnen muss, siehe Abbildung 15.

The image shows a screenshot of a survey application. At the top, there is a question: "My focus is mostly:". Below it is a dropdown menu with four options: "backend", "frontend", "fullstack", and "other". The "backend" option has a checkmark next to it, indicating it is selected. A faint watermark at the bottom of the dropdown menu reads: "Thinking of my current job for the following questions".

Abbildung 15: Formularfeld „Arbeitsschwerpunkt“ [eigene Abb.]

5.3.2 Wert der Daten

Problembeschreibung

5. Auswertung der Testergebnisse

An vielen Stellen im Formular und auch auf der Stories-Seite sind den Expertinnen fehlende Informationen aufgefallen, die sie sich für eine bessere Auswertung der Daten wünschen würden. Das betrifft zum einen zusätzlich gewünschte Formular-Abfragen, ob der User in einem Startup arbeitet, wieviele Urlaubstage er bekommt, oder wieviele Arbeitsstunden er tatsächlich leisten muss (siehe Anhang S. 9, #59-61). Zum anderen betrifft es bestehende Formularfelder, bei denen den Expertinnen die Auswahlmöglichkeit oft zu ungenau ist und sie sich eine breitere Auswahlmöglichkeit wünschten. Im Zweifelsfall würden sich User aus Mangel an Auswahlmöglichkeiten falsch oder unter einer generischen „other“-Option einordnen.

Verbesserung

Alle Probleme dieser Heuristik lassen sich relativ einfach umsetzen, da sie lediglich des Hinzufügens eines weiteren Formular-Feldes oder der Erweiterung bestimmter Dropdown-Menüs bedürfen. Sofern aber nicht alle Probleme umgesetzt werden sollten, ist es empfehlenswert, sich um die Probleme mit dem höchsten Schweregrad zu kümmern. Zum einen betrifft dies den Wunsch der Expertinnen, dass das „Country“-Feld ein Pflichtfeld sein sollte (siehe Anhang S. 10, #66), da sich der Gehaltsdurchschnitt von Land zu Land sehr unterscheiden kann und die Pflichtangabe die Daten besser vergleichbar mache. Zum anderen betrifft das sowohl die Erweiterung des Dropdown für die „primary technologies“ (siehe Anhang S. 11, #74), als auch für „currency“ (siehe Anhang S.13, #92), wo sich die Expertinnen ein breiteres Angebot an Auswahlmöglichkeiten wünschten. Zu guter Letzt wäre da noch das Formular-Feld „I work the following numbers of days a week“ (siehe Anhang S. 12, #80). Die Aussagekraft dieser Formular-Abfrage hat den Großteil der Expertinnen verwirrt, denn Wochenarbeitstage, ohne die Stundenanzahl pro Tag zu kennen, haben eine sehr geringe Aussagekraft. Der Wunsch ist es hier, direkt die Wochenarbeitsstunden anstelle der -tage abzufragen.

6. Bewertung, Validierung und Ausblick

Das letzte Kapitel der vorliegenden Arbeit beinhaltet eine Zusammenfassung der Arbeit in Abschnitt 6.1 im Hinblick auf die vorangegangene Zielsetzung und eine kritische Beurteilung der ausgewählten Evaluationsmethode. Abschließend erfolgt in Abschnitt 6.2 die Validierung der Ergebnisse, und in Abschnitt 6.3 wird ein Ausblick auf die mögliche Weiterentwicklung der Anwendung gegeben.

6.1 Bewertung

Ziel der Arbeit war es, die Usability der Anwendung „What do juniors get paid“ zu beurteilen und mögliche Schwachstellen aufzuzeigen. Zu diesem Zweck wurde eine Heuristische Evaluation in Kombination mit dem Simplified Thinking Aloud durchgeführt. Als Ergebnis entstand eine Gesamtliste aller gefundenen Probleme der Anwendung.

Mit fünf Expertinnen wurde tatsächlich eine sehr breite Masse an Problemen gefunden, jedoch gab es 62 von 109 Problemen, die je nur von einer Expertin gefunden wurden, so dass sich schwer voraussagen lässt, ob eine größere Menge an Expertinnen einen Mehrwert für die Ergebnisse geliefert hätte. Einen Mehrwert hingegen hätte der Einsatz mehrerer Expertinnen wahrscheinlich an der Stelle der Kategorisierung der gefunden Probleme gebracht. Auf Grund der individuell sehr unterschiedlichen Schweregrad-Bewertung (siehe Kapitel 5.1) ist der Mittelwert durchschnittlich sehr hoch ausgefallen. Eine breitere Masse an Expertinnen bei der Kategorisierung des Schweregrads hätte vermutlich ein differenzierteres Ergebnis gebracht. Ebenfalls wurde bei der Auswahl der Expertinnen aus organisatorischen Gründen nicht auf eine gute Repräsentation der Zielgruppe geachtet. Die Expertinnen waren alle Frauen eines Informatik-Studiengangs. Eine diversere Abbildung der Zielgruppe, wie z.B. die Befragung von Männern oder People of Colour hätten die Ergebnisse eventuell beeinflusst.

6. Bewertung, Validierung und Ausblick

Dennoch ist gerade die Kombination aus Heuristischer Evaluation und Simplified Thinking Aloud positiv zu bewerten. Die Heuristische Evaluation bietet einen guten Leitfaden, die Anwendung aus allen Blickwinkeln zu betrachten. Das Simplified Thinking Aloud hingegen ermöglichte einen Blick in das mentale Modell der Expertinnen/Userinnen und machte es ihnen möglich, Lösungsvorschläge zu entwickeln. Der Fokus lag somit nicht auf der reinen Problemfindung, sondern war gleichermaßen lösungsorientiert.

6.2 Validierung der Ergebnisse

Aus dem Debriefing (siehe Anhang S. 17 ff.) mit den beiden Entwicklerinnen der Anwendung (siehe Abschnitt 4.3.4), konnten 15 Issues abgeleitet werden, die bereits in github angelegt wurden, siehe Abbildung 16.

① 15 Open ✓ 0 Closed		Author ▾	Labels ▾	Milestones ▾	Assignee ▾	Sort ▾
①	Restrict currency dropdown to just US-Dollar	#44 opened 4 minutes ago by sarahkami	0 of 4			
①	Add a bigger range to "Company Size"	#43 opened 14 minutes ago by sarahkami			1	
①	Add more selection to "educational background"	#42 opened 16 minutes ago by sarahkami			1	
①	Change "Country" form-field to required field	#41 opened 23 minutes ago by sarahkami			1	
①	Add addition info to the term "intern"	#40 opened 25 minutes ago by sarahkami			1	
①	Add form-field "I work for a startup"	#39 opened 28 minutes ago by sarahkami				
①	Change the form-field "Number of days" to "Hours per week"	#38 opened 30 minutes ago by sarahkami			1	
①	Add Impressum to the Website	#37 opened 34 minutes ago by sarahkami				
①	Add timestamp to each story	#36 opened 35 minutes ago by sarahkami				
①	Add Edit/Check-Mode after form-submit	#35 opened 37 minutes ago by sarahkami	0 of 3			
①	Change order of stories (from newest to oldest)	#34 opened 42 minutes ago by sarahkami				
①	Redirect to specific story (after submit the form)	#33 opened 43 minutes ago by sarahkami				
①	Add system feedback after submitting the form	#32 opened an hour ago by sarahkami				
①	Display random junior stories on landing page	#31 opened an hour ago by sarahkami				
①	Add explanatory text on landing page	#30 opened an hour ago by sarahkami	0 of 2			

Abbildung 16: Auflistung aller neu erstellten github Issues [eigene Abb.]

6. Bewertung, Validierung und Ausblick

Alle Probleme des Schweregrad 4 wurden als solche anerkannt und sollen in der weiteren Entwicklung behoben werden. Zwei der Probleme sind den Entwicklerinnen bereits selbst aufgefallen und wurden eigenständig in github Issues übertragen. Dies betrifft die fehlende Filterfunktion (siehe Abschnitt 5.2.2) und die fehlende Navigation, die auch den fehlenden Zurück-Button (siehe Abschnitt 5.2.4) einschliesst. Auch die Vorschläge der Expertinnen wurden gut aufgenommen, so sollen beispielsweise ein paar Junior Stories auf der Startseite angezeigt werden, damit der User ein Gefühl dafür bekommt, was ihn erwartet. Ebenso soll ein Editiermodus eingeführt werden, der es ermöglicht, die eingereichte Story vor Veröffentlichung nochmal zu überprüfen.

Fast alle Verbesserungswünsche, den Wert der Daten betreffend, sollen zukünftig umgesetzt werden. Ausnahmen bilden hierbei zwei Zusatzfelder: die Erweiterung des Technology-Dropdowns und des Currency-Dropdowns. Beides soll zugunsten der Datenauswertung nicht erweitert werden. Das Currency-Dropdown soll sogar wegfallen, so dass User ihr Gehalt nur noch in US-Dollar angeben können (siehe Anhang S. 20 f.).

Was die unklare Zielgruppe anbelangt, ist es tatsächlich so, dass sich die Anwendung zunächst nur an EntwicklerInnen richtet (siehe Anhang S.22). Hierdurch sind viele der gefundenen Probleme hinfällig, da die Expertinnen automatisch davon ausgegangen waren, dass weitere technische Berufsgruppen eingeschlossen seien. Dies widerum zeigt, dass die Zielgruppe in jedem Fall klar adressiert werden sollte, am besten auf der Startseite, was ebenfalls in einem github Issue festgehalten wurde.

Beide Entwicklerinnen waren sehr zufrieden und dankbar für die gewonnenen Erkenntnisse. Wie Tam Eastley sagte, seien sie durch die Evaluation wieder neu motiviert, an der Anwendung weiterzuarbeiten.

6.3 Ausblick

Das allgemeine Ziel ist es, die Anwendung „What do juniors get paid“ ins Internet zu stellen und für die Allgemeinheit zugänglich zu machen. Dies ist ein laufender Prozess, der bereits zu Beginn der Arbeit gestartet war. Da die Anwendung ehrenamtlich entwickelt wird, geht der gesamte Prozess der Entwicklung jedoch langsam voran. Durch die vorgenommene Usability-Evaluation konnten konkrete Arbeitsaufgaben in Form von github Issues abgeleitet werden, die es den Mitgliedern der JAA ermöglichen soll, sich besser in die Entwicklung einzubringen. Viele der Mitglieder haben Interesse daran, an der Weiterentwicklung mitzuarbeiten, wissen aber oft nicht, was genau sie machen sollen.

Im Zuge dieser Arbeit wurden zunächst nur ein paar der gefundenen Probleme in github Issues übertragen. Ziel ist es jedoch, nach und nach weitere der Probleme anzusprechen und Lösungsmöglichkeiten zu entwickeln. Ebenfalls bieten sich die Evaluationsergebnisse als Grundlage an, ein konkretes Designkonzept für die Anwendung zu erstellen. Durch die Überscheidung von Userinnen und Expertinnen ergaben sich aus der Evaluation konkrete Ideen, wie die Userinnen die Anwendung erfassen und was sie sich von ihr wünschen würden.

Mit einer Anwendung, die nach Usability-Prinzipien gestaltet wurde, sollen in Zukunft genügen Daten gesammelt werden, um Missstände in ungerechter Bezahlung aufzudecken. Die Anwendung soll Menschen ein Werkzeug an die Hand geben, sich in Gehaltsverhandlungen besser zu positionieren und schlussendlich einen kleinen Anteil daran tragen, das Gender Pay Gap zu schliessen.

Literaturverzeichnis

[Backes et al. 2016]

Backes, Jan et al. (2016): The Berlin Startup Salary Report. An analysis of earnings in Berlin's entrepreneurial landscape. URL: <https://storage.googleapis.com/js-journal/Berlin-Startup-Salary-Report-Jobspotting-May2016.pdf>, Abruf am 9.3.2017

[D_Statis 2016]

Statistisches Bundesamt (2016): Verdienstunterschied zwischen Frauen und Männern in Deutschland bei 21 %. Auswirkung des Mindestlohns vor allem im Osten spürbar. URL: https://www.destatis.de/DE/PresseService/Presse/Pressemitteilungen/2016/03/PD16_097_621pdf.pdf?__blob=publicationFile, Abruf am 9.3.2017

[D_Statis 2013]

Statistisches Bundesamt (2013): Frauenverdienste – Männerverdienste: Wie groß ist der Abstand wirklich? URL: https://www.destatis.de/DE/Publikationen/STATmagazin/Verdienste/Arbeitskosten/2013_03/Verdienste2013_03.html, Abruf am 9.3.2017

[DIN EN ISO 9241-11]

Europäische Norm, E. N. (1999). ISO 9241-11: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten; Teil 11: Anforderungen an die Gebrauchstauglichkeit-Leitsätze. DIN Deutsches Institut für Normung eV. (Hrsg.) Beuth Verlag: Berlin.

[GeHaDü 1999]

Literaturverzeichnis

Gediga, Günther; Hamborg, Kai-Christoph; Düntsche, Ivo (1999): The IsoMetrics usability inventory: an operationalization of ISO 9241-10 supporting summative and formative evaluation of software systems URL:
https://www.researchgate.net/profile/Ivo_Duentsch/publication/243779554_The_IsoMetrics_usability_inventory_An_operationalisation_of_ISO_9241-10/links/559b8ac608ae99aa62ce5548.pdf, Abruf am 6.1.2017

[JaegMetz 2006]

Jaeger, Till; Metzger, Axel (2006): Open Source Software. Rechtliche Rahmenbedingungen der Freien Software, Verlag C.H. Beck: München

[JarGaf 2009]

Jarrett, Caroline; Gaffney, Gerry (2009): Forms that Work. Designing Web Forms for Usability, Morgan Kaufmann: San Francisco et al.

[Krug 2006]

Krug, Steve (2006): Don't make me think. Web Usability. Das intuitive Web. 2. Aufl., MITP-Verlags GmbH & Co. KG: Frechen

[LiuWhDu 2010]

Chao, Liu; White, Ryen W.; Dumais, Susan(2010): Understanding web browsing behaviors through weibull analysis of dwell time, in: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval (pp. 379-386). ACM: New York.

[Mayhew 1999]

Mayhew, Deborah J. (1999): The Usability Engineering Lifecycle. A

practitioner's handbook for user interface, Morgan Kaufmann: San Francisco et al.

[MusKe 2003]

Musciano, Chuck; Kennedy, Bill (2003): HTML & XHTML. Das umfassende Referenzwerk, O'Reilly Verlag: Köln

[Nie 1992]

Nielsen, Jakob (1992): Finding usability problems through heuristic evaluation, in: Proceedings of the SIGCHI conference on Human factors in Computing Systems (pp. 373-380). ACM: New York.

[Nie 1993]

Nielsen, Jakob (1993): Usability Engineering, Morgan Kaufmann: San Francisco et al.

[Nie 1994]

Nielsen, Jakob (1994): Heuristic Evaluation, in: Nielsen, Jakob; Mack, Robert L. (Hrsg.): Usability Inspection Methods, John Wiley & Sons, Inc: New York et al.

[Nie 1994B]

Nielsen, Jakob (1994): Enhancing the explanatory power of usability heuristics, in: Proceedings of the SIGCHI conference on Human Factors in Computing Systems (S. 152-158) ACM: New York.

[Nie 1995]

Nielsen, Jakob (1995): 10 Usability Heuristics for User Interface Design, URL: <https://www.nngroup.com/articles/ten-usability-heuristics>, Abruf am 28.12.2016

Literaturverzeichnis

[Nie 2010]

Nielsen, Jakob (2010): Mental Models, URL:
<https://www.nngroup.com/articles/mental-models/>, Abruf am
28.2.2017

[Nie 2011]

Nielsen, Jakob (2011): How Long Do Users Stay on Web Pages?,
URL: <https://www.nngroup.com/articles/how-long-do-users-stay-on-web-pages/>, Abruf am 24.2.2017

[Nie 2012]

Nielsen, Jakob (2012): Usability 101: Introduction to Usability, URL:
<https://www.nngroup.com/articles/usability-101-introduction-to-usability>, Abruf am 6.11.2016

[Nie 2016]

Nielsen, Jakob (2016): Trustworthiness in Web Design: 4 Credibility Factors, URL: <https://www.nngroup.com/articles/trustworthy-design/>, Abruf am 26.2.2017

[NieLo 2006]

Nielsen, Jakob; Loranger, Hoa 2006: Web Usability, Addison-Wesley: München et al.

[NieMa 1994]

Nielsen, Jakob; Mack, Robert L. (1994): Usability Inspection Methods, John Wiley & Sons, Inc: New York et al.

[RailsGirls 2010]

RailsGirls (2010): Press Kit: Rails Girls aims to give tools for women to build their ideas. URL: <http://railsgirls.com/press.html>, Abruf am 28.2.2017

[RedBias et al. 2002]

Redish, Janice; Bias, Randolph G.; Bailey, Robert; Molich, Rolf; Dumas, Joe; Spool, Jared M. (2002):

Usability in Practice: Formative Usability Evaluations — Evolution and Revolution, URL:

<https://pdfs.semanticscholar.org/5e61/743fccfd4ccf43b0788a389544bc8bef889d.pdf>, Abruf am 6.1.2017

[ReEs 2016]

Regan, Sara; Eastley, Tam (2016): Interview mit Sara und Tam (schriftlich), 24.11.2016

[rubyonrails]

rubyonrails.org (o.J.): Getting Started with Rails URL:
http://guides.rubyonrails.org/getting_started.html, Abruf am 2.3.2017

[SaBr 2011]

Sarodnick, Florian; Brau, Henning (2011): Methoden der Usability Evaluation. Wissenschaftliche Grundlagen und praktische Anwendung, 2. Aufl., Verlag Hans Huber: Bern.

[Thung et al. 2013]

Thung, Ferdian et al. (2013): Network structure of social coding in GitHub. In: 17th european conference on Software maintenance and reengineering (S. 323-326), Research Collection School Of Information Systems: Singapur.

[TMG 2007]

§ 5 Telemediengesetz - Allgemeine Informationspflicht, idF der Bekanntmachung vom 26. Februar 2007 (BGBI. I S. 179), zuletzt

Literaturverzeichnis

geändert durch Artikel 1 des Gesetzes vom 21. Juli 2016 (BGBl. I S. 1766) URL: https://www.gesetze-im-internet.de/tmg/_5.html, Abruf am 26.2.2017

Anhangsverzeichnis

1.	Interview with Sara Regan and Tam Eastley	1
2.	Überschneidungsfreie Gesamtliste aller gefundenen Probleme	5
3.	Probleme der Heuristik "Motivation und Zielgruppe"	16
4.	Ergebnisprotokoll des Debriefings.....	17

1. Interview with Sara Regan and Tam Eastley

- Founder of Juniors Are Awesome -

Datum: 24.11.2016

Art: Schriftlich

Question: Other form what I can find on the official website, are there any good informations about „Juniors Are Awesome“?

Because we are not an official organisation, that is registered, there isn't much more to what we do apart from

- 1) Run a Chat that Juniors can join and talk to each other/get advice in
- 2) Provide advice from our experiences as junior developers
- 3) Hopefully develop some useful stories and tools to help juniors such as the pay calculator

Question: How did you found the organisation, what was the motivation?

I forget the exact moment it started, but we were talking about life and struggles faced by juniors (we were both juniors at the time), and thought that many other people must also have these experiences, but they don't seem to talk about them. It was then that we decided to start the slack group. There are numerous reasons why one might think juniors aren't very open with their struggles: they want to serve as good role models for other people learning to code, impostor syndrome, stress. Additionally, there are many views put forth

within the programming field that prevent juniors from speaking up: the prevailing view that one must be passionate about coding in order to succeed, that programming cannot be taught, that one must be incredibly intelligent (and that to question/show confusion demonstrates one's own shortcomings) etc...

The silence around junior's experiences was upsetting to us, and we thought that the whole community would be a lot better if everyone opened up, shared their experiences, and realized that they were not alone. Everyone is confused and struggling along, but we must also celebrate our small successes, and what better way than to do that in a group.

Question: What are the future plans (if any)?

We've toyed with the idea of hosting a junior conference. In it we would present junior friendly talks and possibly also host workshops on beefing up your CV. Due to time limitations in our own lives, however, this is currently unfeasible.

Question: You want to add something about yourself, why you got involved with this?

Not much more to say aside from what was already said in the motivation section.

Question: What is your technical background?

We are both (junior) developers. Before this we worked in other non-programming fields, more oriented towards Social Sciences. Both of us were introduced to programming through Rails Girls and are career changers.

Question: How much are Rubycorns (Rails Girls Study Group) or Rails Girls itself involved in JAA?

Rubycorns are not involved any more than the code being in the Rubycorns repo. Just Tam and Sara. There are a handful of rubycorns in JAA, by rubycorns and JAA are two completely separate entities. (Tam just put JAA up in the rubycorns repo because she didn't know where else to put it). Rails Girls are also not involved at all.

Question: Who mainly was involved in creating the App (so far), Rubycorns all together or the two of you?

The Rubycorns did not build WhatDoJuniorsGetPaid. It was started by Tam, and a few motivated juniors from JAA contributed. The plan was to have it as an open source project that all of JAA could work on, learn on together, and launch together. But unfortunately due to time constraints, this hasn't been possible (yet).

Question: What was the motivation for the App?

To provide some number on what Junior developers get paid. One of the most common questions we heard in the Slack channel is 'this company i am interviewing asking me how much I want, what do I say?' or 'am I getting paid fairly?', and most of the time a Junior developer doesn't know, or feels scared to ask for what they would get paid in a previous job if they are career changers. This can lead to some situations where companies take unfair advantage of people who don't know their worth in wages. We were hoping this app could remedy some of these problems, and give Juniors more confidence because they have data to support their case. We found that the data on other websites was either lacking/out-of-date, available on a paid-for basis OR didn't exist for a junior-level person.

Question: What are the future plans for the App?

We'd like to improve the front end, introduce graphs, and launch it. It would be great if it could be a place with a number of technologies, so that everyone can play around with it, try stuff out, and help the community in the process. There are many people in JAA, and not all of them are rubyists, so it would be great if the app could reflect this.

2. Überschneidungsfreie Gesamtliste aller gefundenen Probleme

Startseite				
#	Schweregr.	Problem	Heuristik (ggf.) Vorschlag zur Verbesserung	Vorkommen
1	1	Buttons sind unterschiedlich groß, so dass der Eindruck einer unterschiedlichen Gewichtung entsteht/Gewichtung unklar.	1	I
2	2	Buttons und Text auf Buttons sind zu klein, Text zu lang.	1	I
3	2	Farbe der Buttons mit heller Schrift könnte schwer zu erkennen sein (Kontrast ist zu gering).	1	I
4	3	Der Begriff „Stories“ ist verwirrend (Was ist damit gemeint?) (Buttontext nicht klar).	2 Lieber Data statt Stories verwenden	III
5	4	Es werden keine Infos bereitgestellt, der User weiß nicht worum es geht.	1 - Most recent Stories anzeigen, damit der User weiß, worum es geht - Bilder einbinden	IIII
6	3	Startseite wirkt uneinladend (Startseite zu nackt).	1	III
7	3	Logo und Organisation fehlt (Wer steht dahinter?).	1	I
8	1	Text und Buttons sind zu weit oben angeordnet.	1 Lieber zentrieren	II
Stories				
#	Schweregr.	Problem	Heuristik (ggf.) Vorschlag zur Verbesserung	Vorkommen
9	3	Die einzelnen Stories sind nicht klar voneinander abgehoben, die Trennung ist nicht klar erkennbar.	1	IIII
10	3	Eine erklärende Einleitung zu den Stories fehlt, somit ist der Sinn und Zweck auf den ersten Blick nicht erkennbar.	1	II
11	2	Die Eingabefelder aus dem Formular sind in den Stories nicht wiederzuerkennen. (Der Fließtext wirkt vergleichsweise inkonsistent.)	4 Entweder Lückentext zum Ausfüllen (als Formular) oder Stories auch als Auflistung	I
12	2	Gleiche Inhalte in den einzelnen Stories sind nicht gut vergleichbar.	1 Lieber Stories als Auflistung gestalten	I
13	4	Die gesuchten Informationen werden nicht gefunden, da Filtermöglichkeiten fehlen.	7 Mehrfaache Filtermöglichkeit ist gewünscht, ebenso wie Datenanalyse und Durchschnittsberechnung	IIII

#	Schweregr.	Problem	Heuristik	(ggf.) Vorschlag zur Verbesserung	Vorkommen
14	2	Die Möglichkeit des CSV-Exports ist verwirrend, was soll man damit machen?	10	- Auswahlmöglichkeit, in welchem Format heruntergeladen möchte ist gewünscht (z.B. pdf, doc) - für den CSV-Export ist ein anderes Wording gewünscht, eher „Info“	III
15	2	CSV-Export funktioniert nicht richtig. Daten werden zwar heruntergeladen, aber nicht ordentlich angezeigt. (Daten unübersichtlich und schlecht lesbar)	10		II
16	2	Stories wirken unpersönlich.	Motivation und Zielgruppe	Gerne als Fakepersönlichkeit (trotz Anonymität) darstellen, z.B. als Steckbrief, mit Avatar	I
17	2	Text klingt zu computer-generiert. Das ist unschön/anstrengend zu lesen..	2		II
18	2	Info, wann Story eingereicht wurde fehlt - für bessere Bewertung/Interpretation. (Der User hat z.B. Probleme die Infos im Kontext Gehaltsentwicklung über die Zeit zu bewerten.)	Wert der Daten		II
19	1	Anzeige wieviele Stories vorhanden sind fehlt.	1		I
20	2	Die breite Gestaltung und die Länge des Textes ist unschön für das Auge und der User möchte es evtl. nicht lesen. Fließtext ist schwierig und unübersichtlich!	1		II
21	1	Die Schrift der Stories ist zu klein, das ist schwierig und anstrengend zu lesen.	1		I
22	2	Scale 1-5 wird im Text nicht erklärt, der User weiß nicht, welcher Wert für was steht.	1		I
23	1	„I live in DE“ ist unschön zu lesen, da es unnatürliche Sprache ist - evtl. auch nicht zu verstehen.	2		II
24	0	Download (CSV-Export) wird vom User unten erwartet.	4		I
25	2	„Other“ in den Stories stört den Lesefluss und hat keine Aussage.	2		I
26	3	Reihenfolge ist sehr schwierig, weil die neueste Story am Ende landet. Der User muss scrollen um die neueren Einträge zu sehen.	1	Lieber most recent story am Anfang, Sortierung umdrehen	I

27	2	Anordnung als Liste schwierig. Der User muss irgendwann endlos scrollen um alle Stories zu sehen. Sehr unbequeme Handhabung.	1		1
Formular allgemein					
#	Schweregr.	Problem	Heuristik	(ggf.) Vorschlag zur Verbesserung	Vorkommen
28	2	Inhaltliche Blöcke sind zu wenig voneinander abgegrenzt.	1	gerne weitere Unterteilung bei „Job Specifics“ und „Company“ gewünscht	II
29	2	Die Länge des Formulars erschlägt einen.	1	Lieber mehrere Seiten als ein langes Formular	II
30	1	Die Seite ist zu kahl und uneinladend.	1		I
31	2	Labels sind oft zu lang/breit. Es ist sowohl unschön fürs Auge, als auch unklar, welche Eingabe gefordert ist.	1		III
32	1	Einleitungstext ist zu lang/breit und die wesentlichen Infos sind nicht hervorgehoben.	1	- Man könnte z.B. die Erläuterung zu required fields weglassen, da es durch *- Kennzeichnung selbsterklärend ist - required fields kann als Auflistung gestaltet werden, anstelle von Klammerauflistung im Fließtext	I
33	3	Wiederbestätigung des Users, nach Abschicken des Formulars („Create Junior story“) fehlt.	9	Sowas wie „Möchten Sie das wirklich abschicken?“	III
34	4	Bestätigung/Rückmeldung nach Abschicken des Formulars („Create Junior story“) fehlt, der User weiß somit nicht wirklich was passiert.	5		III
35	3	Vorausgefüllte Felder (Platzhalter) sind nicht erkennbar, da Text auch nicht verschwindet, wenn das Feld aktiv ist (besonders bei „Happiness Factor“ auffällig).	10	Lieber vorausgefüllte Felder mit „e.g.“ kennzeichnen	III
36	3	Auswahl bei Ja/Nein Abfragen (bzw. 2 Optionen) ist anstrengend.	4	Lieber Checkboxen oder Radio-Buttons verwenden	III
37	3	Bei max. 5 Einträgen, sollten Dropdownmenüs komplett angezeigt werden - bequemere Benutzbarkeit für den User.	4	z.B. als Radiobuttons oder Checkboxen	II
38	1	*Kennzeichnung sollte hinter dem Label stehen und Erläuterung zu required fields sollte direkt dabei stehen.	8		I

#	Schweregr.	Problem	Heuristik	(ggf.) Vorschlag zur Verbesserung	Vorkommen
39	1	„Others“ sollte bei jedem Dropdown mit viel Auswahl angeboten werden, das wirkt konsistenter.	4		I
40	3	„Other“- Optionen sind zu generisch und sagen nichts aus.	7	Lieber ein zusätzliches Freitextfeld bei „other“	III
41	2	Unklar, was mit Happiness-Daten passiert, wenn sie nicht veröffentlicht werden. Inwieweit werden die Daten verarbeitet, gespeichert? Wozu?	10		II
42	2	Fehlermeldung verschwindet nach Korrektur der Fehleingabe nicht automatisch.	10		I
43	2	Fehlermeldung „Can't be blank“ klingt zu unhöflich.	8		I
44	3	Freitextfelder können eine Gefahr für Attacken sein (Einschleusen bösartiger Skripte etc.).	9		II
45	3	?Kennzeichnung ist bei einigen Felder vorhanden, aber hat keine Funktion. Das ist verwirrend für den User. Er würde an dieser Stelle eine Hilfestellung erwarten.	10		II
46	1	Der Einleitungstext beginnend mit „Danke“ wirkt komisch. Der User erwartet eher eine Einladung zuerst und das Danke am Schluss.	1		I
47	2	Dropdown-Regler am Ende des Eingabefeldes nicht erkennbar, schlecht sichtbar.	1		II
48	1	Spelling-Check nach dem Ausfüllen des Formulars anbieten, da es sonst zu Fehleingaben führen kann und somit die Auswertung schwierig macht.	9		I
49	3	Bei geforderter Zahleneingabe sind vereinzelte Sonderzeichen möglich, das kann zu Fehleingaben führen.	9	Lieber mit Zahlenspinner arbeiten und Eingrenzung der Zahlen (Dreistellig bis 999)	-
50	3	Bei Zahleneingaben (z.B. Alter) sind negative Werte möglich, das führt zu Fehlern..	9		-
51	3	Dem User ist unklar, welche Daten genau veröffentlicht werden.	10		-
52	1	Labels haben teilweise einen Doppelpunkt, teilweise nicht, das wirkt inkonsistent.	4		-

#	Schweregr.	Problem	Heuristik	(ggf.) Vorschlag zur Verbesserung	Vorkommen
53	2	Wording: „Junior“, „Junior Dev“, „Developer“ ist nicht klar. Wer ist die Zielgruppe?	Motivation und Zielgruppe	I	
54	2	Umgang mit Datenschutz ist dem User nicht klar. Eine genauere Info ist gewünscht.	10	Gerne als Link oder Mouseover im Einleitungstext	-
55	1	Basic-felder (Geschlecht, Alter...) eher am Ende platzieren, wegen Stereotype-Threat.	1		-
56	3	Cancel-Button wird vermisst.	6		-
57	2	Erklärung zu „required fields“ in Einleitungstext bricht einfach ab. „Currency“ wird nicht weiter erklärt.	1		-
58	1	Häckchen am Beginn des Dropdowns ist verwirrend. (Bei primary technology) hat eine Userin hierdurch eine Mehrfachauswahl vermutet.)	1		II
59	2	Abfrage: „Startup – ja/nein“ gewünscht.	Wert der Daten	-	-
60	1	Abfrage: „Urlaubstage“ gewünscht, da dies eine Art Bonus/Vergünstigung sein kann.	Wert der Daten	-	-
61	2	Abfrage: „Contracted Hours“ und „Honest Hours“ gewünscht.	Wert der Daten	Gerne auch zusammen arrangieren	-
Einzelne Formular-Felder					
#	Schweregr.	Problem	Heuristik	(ggf.) Vorschlag zur Verbesserung	Vorkommen
62	2	Auswahlmöglichkeit schränkt den User zu sehr ein. Nicht alle Jobs haben eine „Junior“-Bezeichnung im Titel.	Wert der Daten	- z.B. „It's not called Junior Position, but I just finished education“ - gerne an dieser Stelle eine Auswahl mit Teilzeit, Vollzeit, Werkstudent ...	II
63	2	Art/Level und Job sind vermischt.	Wert der Daten	Lieber Trennung: z.B. Art: Festangestellt, Teilzeit... , Jobbezeichnung: ... + Freitextfeld	I

#	Schweregr.	Problem The gender I identify with is: female	Heuristik	(ggf.) Vorschlag zur Verbesserung	Vorkommen
64	2	Gender-Freitextfeld zwar aus politischer Sicht positiv, aber kann zu Fehlern führen - ungewohnt für den User.	9	- lieber als Dropdown gestalten - könnte als Schieberegler gestaltet werden, somit müsste man sich nicht eindeutig zuordnen	III
65	1	<input type="checkbox"/> I identify as a person of colour „Person of colour“ unklar. Wie ist die Definition, wo fängt der Begriff an?	10	- lieber eindeutiger als eigenes Feld kennzeichnen - evtl. mit Angabe welche Colour	II
		I live in: Berlin Country			
66	3	Land sollte ein Pflichtfeld sein, da es wichtig für die Vergleichbarkeit der Daten ist (Es gibt große Gehaltsunterschiede alleine innerhalb der EU).	Wert der Daten		I
67	2	Country-Dropdown und City-Freitext wirkt inkonsistent.	4		I
68	0	City- und Country-Auswahl sollte umgedreht werden.	4		II
69	2	„City“-Label evtl. nicht direkt erkennbar. Der User weiß nicht unbedingt was er eingeben soll.	9	Lieber „City I live in“ als Label	I
70	2	City und Country lieber deutlicher zugehörig machen.	1	Lieber in gleicher Zeile anordnen	II
		My educational background when it comes to programming is: 			
71	1	Genauere Erklärung zu den Menüinträgen gewünscht (z.B. Bootcamp).	10	Gerne als Mouseover	I

#	Schweregr.	Problem	Heuristik	(ggf.) Vorschlag zur Verbesserung	Vorkommen
			Wert der Daten	- Gerne Mehrfachauswahl zulassen - zusätzlich Gewünscht ist: Studiert BA/MA (fertig/abgebrochen), noch Student BA/MA, Ausbildung	III
72	2	User hat Probleme sich genau zuzuordnen, weil Auswahlmöglichkeiten vermisst werden und evtl. mehrere Auswahlmöglichkeiten passen.			
		This is my first junior/intern job in the tech sector of this nature			
73	2	Die Frage ist zu lang, das Wording ist unklar. My primary technology at work is:	1	1	
74	3	Keine Flexibilität, nicht alle Technologien sind abgebildet.	Wert der Daten	- Gerne Mehrfachauswahl zulassen - zusätzliche Option „other“ und/oder Freitextfeld gewünscht	III
75	3	Auswahlmöglichkeit ist nicht alphabetisch sortiert, der User findet sich somit schlechter zurecht. My focus is mostly:	7		I
76	3	Zu wenig Auswahl, es werden nicht alle Jobs/Bereiche abgebildet - schließt viele Teilnehmer aus. I have been working at my current job for the following number of years:	Motivation und Zielgruppe	Zusätzliches Freitextfeld wünschenswert	II
77	1	Unterschiedung zwischen beiden Arbeitsjahren nicht ganz klar.	1		I
78	1	Auswahlmöglichkeit redundant.	9	Lieber Auswahl als: 1-2, 3-4, 4 und mehr	I

#	Schweregr.	Problem	Heuristik	(ggf.) Vorschlag zur Verbesserung	Vorkommen
			Motivation und Zielgruppe		I
79	2	„Developer“ ist zu eingegrenzt; da es sich ja auch um andere Jobs handeln kann.			
		I work the following numbers of days a week:			
80	3	Wording ist sehr verwirrend für den User. Number of days == 40h/Woche? Wochenarbeitsstage haben sehr wenig Aussagekraft.	Wert der Daten	Lieber Wochenarbeitsstunden abfragen, da Wochenarbeitsstage irrelevant	III
		I work mostly remote			
81	1	Genauere Erklärung zu remote gewünscht. Der User weiß nicht genau, was damit gemeint ist.	10		I
		My company is about this size:			
82	2	Information zu ungenau, da Einteilung zu grob - Aussagekraft dadurch gering.	Wert der Daten	Bei über 50 – eine genauere Einteilung gewünscht, da es einen Unterschied zwischen 50, 500, 5000 gibt	II
83	1	„Company Size“ ist unklar. Was genau ist gemeint? Nur Standort, Filialen weltweit, Jahresumsatz? Der User benötigt mehr Informationen oder eine Hilfestellung.	10		I
		My tech team is about this size:			
84	1	Genauere Erklärung zu „Tech Team“ gewünscht. Der User weiß nicht, was genau damit gemeint ist: das gesamte Team, die gesamte IT-Abteilung, das Projekt-Team etc.	10		I

#	Schweregr.	Problem	Heuristik	(ggf.) Vorschlag zur Verbesserung	Vorkommen
		My company is about (x) years old:			
		3			
85	1	Formulierung als „Lückentext“ wirkt im Vergleich zum Rest des Formulars inkonsistent.	4		I
86	1	Auswahlmöglichkeit lieber direkt im Text einbauen, das ist besser lesbar/bedienbar. ... * I make about this much per year before taxes	1	„My company is about _____ years old“	I
		20 000			
		* Currency			
87	1	User sind oft an Monatsgehalt gewöhnt, die Umrechnung in das Jahresgehalt ist umständlich.	7	Gerne Wahlmöglichkeit, ob Angabe in Jahres- oder Monatsgehalt	I
88	2	Die Zugehörigkeit zwischen Gehaltsabfrage und Currency wird auf den ersten Blick nicht klar.	1	Lieber nebeneinander darstellen	III
89	1	Bei Gehaltsabfrage – Hilfe durch Tausender-Trennung (z.B. Whitespace in der Darstellung, ähnlich wie bei IBAN).	4		II
90	2	Was passiert mit Komma- und Punkt trennung (Sonderzeichen generell)? Gilt British oder American English?	2		II
91	1	Der Begriff „Salary“ wird im Eingangstext erwähnt, fehlt aber im entsprechenden Formular-Feld. Wording ist hier inkonsistent und verwirrend.	4		I
92	3	Currency-Dropdown bietet zu wenig Auswahl.	Wert der Daten		I

#	Schweregr.	Problem	Heuristik	(ggf.) Vorschlag zur Verbesserung	Vorkommen
		The following section is a bit more personal. We will not publish it if you won't want us to. <input checked="" type="checkbox"/> You have my consent to publish this slightly more personal information about myself.			
93	2	Vorausgewählte Zustimmung ist ungewünscht - User fühlt sich bevormundet. I am happy in my job (5 being very happy, 1 being very unhappy)	6	Stattdessen pro-aktive Zustimmung gewünscht	I
94	3	Skala und Rating nicht klar erkennbar (Mapping-Faktor nicht erfüllt). Erklärung sollte von 1-5 gehen, nicht andersrum.	4	Gerne visuell darstellen, z.B. als Sternchen	
95	1	My happiness level at work is because of... I don't get enough support, everyone is super awesome, I wish there was more candy in the workplace	2		I
96	2	Das Freitextfeld bietet zu viel Eingabefreiheit und macht es schwer, die Infos später aus den Stories auszulesen. Der Begriff „Happiness-Level“ taucht vorher nirgends auf, das ist verwirrend für den User. Anything else you'd like to tell us?	9	Lieber eine Kategorieauswahl + others (Freitextfeld)	I
97	1		4		I
98	2	Der User weiß nicht, ob diese Infos auch veröffentlicht werden. Genauere Infos dazu fehlen. Worauf bezieht sich die Frage? Handelt es sich hierbei um Feedback?	10		II
99	2	Button ist zu nah am Rand platziert, ohne Margin. Benennung des Buttons ist verwirrend. Der User erwartet einen „Submit“-Button.	1		
100	1		4		I

Create Junior story

gesamte App				
#	Schweregr.	Problem	Heuristik (ggf.) Vorschlag zur Verbesserung	Vorkommen
			Motivation und Zielgruppe	
101	3	User weiß nicht, an wen sich die App richtet - unklar, ob es nur um „Developer“ geht.		III
102	3	Wording zwischen Formular und Stories unklar und inkonsistent.	4	I
103	4	„Zurück“-Buttons fehlen innerhalb der gesamten App. User müssen die Browseroption wählen. Besonders kritisch ist dies, wenn sie ein Formular absenden, automatisch zu den Stories weitergeleitet werden und wieder auf die Startseite wollen. Es gibt hier keinen direkten Weg, sondern nur eine Rückweg über das Formular.	6	IIII
104	3	Navigation/Allgemeiner Rahmen (Header,Footer) fehlt.	4	III
105	4	Impressum fehlt. Der User weiß nicht, wer hinter der Seite steht, wem die Daten gegeben werden, wie die BetreiberInnen kontaktiert werden können.	4	IIIII
106	3	Die Motivation zur Teilnahme an der Umfrage, zur Bereitstellung der eigenen Daten ist nicht klar.	Motivation und Zielgruppe	I
107	2	Infos über die Aktualität der Seite fehlen (ist die Seite noch aktuell, wird sie gewartet...?).	Wert der Daten	I
108	2	Da die App sich an internationale User richtet, ist die Gestaltung nur in Englisch schwierig und schließt User aus.	2	Lieber Angebot in Deutsch und Englisch (Deutsch, da die Seite aus Deutschland kommt, und der Großteil des Zielpublikums zunächst in Deutschland ist und Englisch für generelle Internationale Verständigung)
109	3	Unklar, ob es die Option gibt Daten endgültig löschen zu lassen.	6	I

3. Probleme der Heuristik „Motivation und Zielgruppe“

#	Schweregr.	Problem	Heuristik	(ggf.) Vorschlag zur Verbesserung	Vorkommen
16	2	Stories wirken unpersönlich.	Motivation und Zielgruppe	Gerne als Fakepersönlichkeit (trotz Anonymität) darstellen, z.B. als Steckbrief, mit Avatar	I
53	2	Wording: „Junior“, „Junior Dev“, „Developer“ ist nicht klar. Wer ist die Zielgruppe?	Motivation und Zielgruppe		I
76	3	Zu wenig Auswahl, es werden nicht alle Jobs/Bereiche abgebildet - schließt viele Teilnehmer aus.	Motivation und Zielgruppe	Zusätzliches Freitextfeld wünschenswert	II
79	2	„Developer“ ist zu eingegrenzt, da es sich ja auch um andere Jobs handeln kann.	Motivation und Zielgruppe		I
101	3	User weiß nicht, an wen sich die App richtet - unklar, ob es nur um „Developer“ geht.	Motivation und Zielgruppe		III
106	3	Die Motivation zur Teilnahme an der Umfrage, zur Bereitstellung der eigenen Daten ist nicht klar.	Motivation und Zielgruppe		I

4. Ergebnisprotokoll des Debriefings

Datum: 2.3.2017

Zeit: 18:30 – 19:30 Uhr

Teilnehmerinnen: Tam Eastley, Sarah Ganter, Sara Regan

Protokollantin: Sarah Ganter

Im Folgenden werden die Ergebnisse zusammengefasst, welche sich aus der Besprechung jedes Problems ergeben haben:

Probleme: Schweregrad 4

Problem 1: Fehlende Information auf der Startseite (siehe Anhang S. 5, #5)

- **Github Issue soll angelegt werden.**
- Es soll ein kurzer Erläuterungstext (Wer/Was/Warum) auf der Startseite hinzugefügt werden.
- Es sollen Junior Stories auf der Startseite angezeigt werden.
- Die Stories sollen vorausgewählt werden können, damit keine unseriösen Angaben auf der Startseite landen. Dies soll mit Hilfe einer Kennzeichnung in der Datenbank passieren.
- Innerhalb der vorausgewählten Stories soll es eine zufällige Anzeige geben, welche Stories angezeigt werden, damit der User nicht immer dieselben Stories sieht.
- Es soll eine Auswahlmöglichkeit im Formular bereitgestellt werden, die es den Usern ermöglicht, zu entscheiden, ob ihre Story auf der Startseite angezeigt werden darf.
- Die Stories können so dargestellt werden, dass der User nur einen kleinen Ausschnitt sieht, mit einer „Read more“-Option.

Problem 2: Fehlende Filtermöglichkeiten (siehe Angang S. 5, #13)

- Hierfür wurde bereits ein github Issue angelegt.
- Ebenso wurde bereits ein github Issue für die Datenanalyse angelegt, die Daten vergleichbar machen soll.

Problem 3: Fehlende System-Rückmeldung (siehe Anhang S. 7, #34)

- **Github Issue soll angelegt werden.**
- System-Rückmeldung soll eingebaut werden (die Art wurde nicht spezifiziert)
- Ein Editiermodus nach Absenden des Formulars soll eingebaut werden.
- Das hiermit verbundene Problem der Datensicherheit, weil andere Menschen die Daten einsehen könnten, wenn der User den Computer verlässt, soll durch Sessions (Sitzungen) gelöst werden. (Die Eingaben verschwinden automatisch, wenn eine Session abgelaufen ist.)
- Der Redirect nach Abschicken des Formulars soll auf die neu angelegte Story leiten, nicht auf die Story-Index Seite. Hierfür muss jede Story ihre eigene Seite bekommen.

Diskussion: Wie wird das Problem zwischen Vertrauen und Anonymität gelöst?

- Es gibt die Möglichkeit, dass sich User z.B. über ihr github-Konto in der App einloggen könnten. Dies würde gewährleisten, dass (nur) seriöse Daten eingegeben werden, weil sicher gestellt ist, dass es die User wirklich gibt. Die App wäre dann aber nichtmehr komplett anonym.
- Die Login-Funktion wird vorerst nicht umgesetzt, die Diskussion bleibt aber noch offen.

Problem 4: Fehlender Zurück-Button (siehe Anhang S. 15, #103)

- Hierfür wurde bereits ein github Issue angelegt.

Problem 5: Fehlendes Impressum (siehe Anhang S. 15, #105)

- **Github Issue soll angelegt werden.**
- Inhalt und Position müssen von den Entwicklerinnen noch spezifiziert werden.

Probleme: Wert der Daten

Problem: Information, wann Story eingereicht wurde fehlt (siehe Anhang S. 6, #18)

- **Github Issue soll angelegt werden.**
- Die Information soll durch einen Timestamp bereitgestellt werden.

Problem: Zusätzliche Abfragen werden vermisst (siehe Anhang S. 9, #59-61)

- Startup: **Github Issue soll angelegt werden.**
- Urlaubstage: Github Issue soll nicht angelegt werden, da die Urlaubstage in Amerika strikt sind (die App richtet sich an internationale User).
- Contracted Hours/Honest Hours: Github Issue soll nicht angelegt werden, da die Abfrage verwirrend wirkt.
- Zusätzliche Abfrage nach Stock-Options kann optional auch umgesetzt werden.

Problem: Auswahlmöglichkeit des „employed“-Feld schränkt User zu sehr ein (siehe Anhang S. 9, #62-63)

- **Github Issue soll angelegt werden.**
- Das Label „I am employed as a:“ sollte in etwas wie „My level is...“ umbenannt werden.

- Da „intern“ im englischen auch Werkstudent beinhaltet, sollte eine Zusatzinformation bereitgestellt werden, die den Term besser erklärt.

Problem: „Land“ sollte Pflichtfeld sein (siehe Anhang S. 10, #66)

- **Github Issue soll angelegt werden.**

Problem: „Educational background“ bietet zu wenig Auswahlmöglichkeit (siehe Anhang S. 11, #72)

- **Github Issue soll angelegt werden.**
- Auswahlmöglichkeit sollte um Bachelor und Masterabschluss erweitert werden, sowie um die Auswahlmöglichkeit, ob man es aktuell studiert oder abgeschlossen hat.

Problem: Es werden nicht alle Technologien abgebildet (siehe Anhang S. 11, #74)

- Je größer die Auswahl wird, desto schwieriger ist die Datenauswertung.
- Es ist besser, die Technologien zu begrenzen.
- Hierbei ist auch interessant, herauszufinden, ob es Technologien gibt, die gar nicht mehr von Juniors eingesetzt werden. (Es muss also eine Mischung aus aktuellen, modernen Technologien sein und altmodischen.)

Problem: Wording „Number of days“ ist sehr verwirrend (siehe Anhang S. 12, #80)

- **Github Issue soll angelegt werden.**
- Die Abfrage soll von „number of days“ zu „hours per week“ geändert werden.

Problem: Die Abfrage „Company Size“ ist zu ungenau (siehe Anhang S. 12, #82)

- **Github Issue soll angelegt werden.**
- Genaue Abgrenzung muss noch spezifiziert werden, evtl. 50, 500, mehr als 1000.

Problem: Currency-Dropdown bietet zu wenig Auswahlmöglichkeit (siehe Anhang S. 13, #92)

- **Github Issue soll angelegt werden**, jedoch soll die Auswahl weiter begrenzt werden.
- Es soll nur US-Dollar zur Auswahl stehen, was die Daten besser vergleichbar macht. Das ganze Auswahlfeld kann somit wegfallen.
- Es soll ein Text hinzugefügt werden, der darum bittet, das Gehalt in US-Dollar umzurechnen.
- Später kann ein Umrechnungstool hinzugefügt werden.

Diskussion: Was passiert wenn der Markt einbricht? Sind die Daten dann nutzlos?

- Das Problem kann durch den Timestamp gelöst werden. Der User kann somit nachvollziehen, wie der Dollarkurs zur Einreichung der Story steht/stand.
- Es könnte ein Aktienmarkt-Ticker an der Seite hinzugefügt werden.

Problem: Informationen über die Aktualität der Seite fehlen (siehe Anhang S. 15, #107)

- das Problem ist durch die Einführung eines Timestamps auf der Stories-Seite gelöst.

Problem: Motivation und Zielgruppe

Problem: Die Zielgruppe ist unklar (siehe S. 16, #53, #76, #79, #101)

- Die App richtet sich tatsächlich nur an Entwickler und soll weitere technische Berufe vorerst nicht abbilden. Diese Entscheidung wurde zugunsten der Auswertbarkeit/Vergleichbarkeit der Daten getroffen, da die Gehälter von unterschiedlichen Berufsgruppen sehr auseinander gehen können.
- Die Zielgruppe soll in dem Erklärungstext auf der Startseite (siehe Problem 1) adressiert werden.

Name: Ganter

Vorname: Sarah

Matrikel-Nr.: 544827

Fachbereich: 4

Studiengang: Informatik und Wirtschaft

Eigenständigkeitserklärung

Ich versichere, dass ich die vorliegende Bachelorarbeit mit dem Titel:
„Konzeption und Umsetzung einer Open Source Anwendung für mehr
Gehältertransparenz“ selbstständig verfasst und keine anderen als die
angegebenen Hilfsmittel benutzt habe.

Die Stellen, die anderen Werken dem Wortlaut oder dem Sinn nach
entnommen wurden, habe ich in jedem einzelnen Fall durch die Angabe
der Quelle, auch der benutzten Sekundärliteratur, kenntlich gemacht.

Ich versichere, dass ich bisher keine Prüfungsarbeit mit gleichem oder
ähnlichem Thema bei einer anderen Hochschule oder Prüfungsbehörde
vorgelegt habe.

Ort/Datum:

Unterschrift: