



Julius-Maximilians-Universität Würzburg
Institut für Informatik
Lehrstuhl für Kommunikationsnetze (Informatik III)

Emulation Framework for the Subjective Evaluation of Delays in Web and Business Applications

Emulationsumgebung zur
Untersuchung der subjektiven
Verzögerungswahrnehmung in
Web- und
Businessapplikationen

Masterarbeit im Fach Informatik
vorgelegt von

Valentin Fischer



Julius–Maximilians–Universität Würzburg
Institut für Informatik
Lehrstuhl für Kommunikationsnetze (Informatik III)

Emulation Framework for the Subjective Evaluation of Delays in Web and Business Applications

Emulationsumgebung zur Untersuchung der subjektiven Verzögerungswahrnehmung in Web- und Businessapplikationen

Masterarbeit im Fach Informatik
vorgelegt von

Valentin Fischer

geboren am 28. September 1988 in Pawlodar

Angefertigt am
Lehrstuhl für Kommunikationsnetze (Informatik III)
Julius–Maximilians–Universität Würzburg

Betreuer:

Prof. Dr.-Ing. P. Tran-Gia
Dr. rer. nat. Thomas Zinner
Dipl.-Inform. Matthias Hirth

Abgabe der Arbeit:
31.03.2016

Erklärung

Ich versichere, die vorliegende Masterarbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur verfasst zu haben. Darüber hinaus versichere ich, die Arbeit bisher oder gleichzeitig keiner anderen Prüfungsbehörde zur Erlangung eines akademischen Grades vorgelegt zu haben.

Würzburg, den 31.03.2016

(Valentin Fischer)

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen und verwandte Arbeiten	3
2.1	Qualitätsbegriff in technischen Systemen	3
2.1.1	Quality of Service vs. Quality of Experience	3
2.2	Methoden zur Erfassung der subjektiven Wahrnehmung	4
2.3	Verzögerungsarten	6
2.3.1	Eingabeverzögerung	6
2.3.2	Ladeverzögerung	7
2.4	Crowdsourcing	7
2.4.1	Anwendungsbeispiele	7
2.4.2	Microworkers.com	9
2.4.3	Crowd als Forschungsgruppe	9
2.5	Verwandte Arbeiten	10
2.5.1	Einfluss von Wartezeiten und Verzögerungen im psychologischen Zusammenhang	11
2.5.2	Einfluss von Wartezeiten und Verzögerungen in technischen Systemen und webbasierten Diensten	13
2.5.3	Analyse des Einflusses von Verzögerungen in Web- und Business Applikationen	14
3	Framework	17
3.1	Verwendete Webtechnologie	17
3.1.1	PHP	17
3.1.2	HTML	18
3.1.3	JavaScript und JQuery	19
3.1.4	CSS	19
3.2	Framework Übersicht	19
3.3	Vorverarbeitung	20
3.4	Editor	21
3.4.1	Implementierung	21
3.4.2	Erstellung eines emulierten Prozesses	24
3.4.3	Aufbau der generierten Emulationen	25
3.5	Nachbearbeitung	26
3.6	Anwendungsbeispiel: Amazon.com Emulation	27
3.6.1	Vorverarbeitung	27
3.6.2	Erstellung der Emulation	28
3.6.3	Nachbearbeitung	29
3.7	Erweiterungsmöglichkeiten	29

4	Methodik und Umsetzung	30
4.1	Durchführung der Studie mittels Crowdsourcing	30
4.2	Untersuchungsparameter	32
4.3	Nutzergruppen	33
4.4	Filterkriterien	33
4.5	Experimente	35
4.5.1	Amazon Funktionstest	35
4.5.2	Amazon Emulation	35
4.5.3	Zusammenfassung der durchgeführten Untersuchungen	38
5	Auswertung	40
5.1	Charakterisierung der Nutzergruppen	40
5.1.1	Struktur der Teilnehmer von Microworkers.com	40
5.1.2	Struktur der Referenzgruppe	42
5.2	Filterung der Daten	43
5.2.1	Filter	44
5.2.2	Anwendung der Filter	46
5.3	Auswertung	48
5.3.1	Einfluss der eingestellten Verzögerung auf die QoE	48
5.3.2	Einfluss der Ladeanimation auf die QoE	50
5.3.3	Einfluss der Vorlagenart der Emulation auf die QoE	52
5.3.4	Einfluss der Anweisungsart auf die QoE	54
5.4	Vergleich der Daten der Nutzergruppen	56
5.5	Akzeptanz der Webapplikation und Einzelfeedback	57
6	Fazit und Ausblick	59
	Abbildungsverzeichnis	62
	Tabellenverzeichnis	63
	Literaturverzeichnis	64

1 Einleitung

Verzögerungen bei der Nutzung verteilter Anwendungen haben einen starken Einfluss auf das Anwendungsverhalten und damit auf die vom Nutzer erfahrene Anwendungsqualität, die Quality of Experience (QoE). Ab einer gewissen Dauer verursachen sie Missmut, unterbrechen Gedankengänge, oder lenken uns von unseren eigentlichen Zielen ab. Sie stören während der Freizeit, zum Beispiel beim Surfen im Internet oder bei der Betrachtung von Videos, aber auch im Arbeitsumfeld, nämlich bei Jobs, bei denen mit technischen Systemen interagiert wird. Je nach Arbeit und benutzter Anwendung können Verzögerungen einen signifikanten Anteil der Arbeitszeit ausmachen. So wird zum einen die Produktivität des Arbeiters reduziert, zum anderen hat er auch das Gefühl langsam voran zu kommen, was den Spaß an der Arbeit mindert.

Untersuchungen zur QoE in interaktiven Systemen - insbesondere in web-basierten Systemen - fokussieren sich auf Lade- und Wartezeiten. In der Regel werden Studien entweder in einer kontrollierten Umgebung, in der relevante System- und Anwendungsparameter nachvollzogen werden können, mit einer kleinen Anzahl an Teilnehmern durchgeführt [1, 2, 3, 4], oder in einer offenen Umgebung, in der Rückschlüsse auf relevante Parameter nur bedingt oder gar nicht möglich sind, mit einer großen Anzahl Teilnehmer [5, 6, 7, 8]. Neben diesen Forschungsarbeiten existieren keine Arbeiten über den Einfluss von Verzögerung auf andere Anwendungen, und auch keine standardisierten Bewertungsmethoden. Dies liegt vor allem an den folgenden Herausforderungen. i) Vorhandene Systeme können nicht oder nur sehr schwer so angepasst werden, dass unterschiedliche Verzögerungen untersucht werden können. Dies gilt insbesondere für E-Commerce- und Unternehmensanwendungen. Im Falle von netzbasierten Anwendungen können Netzemulatoren genutzt werden um Verzögerungen zu emulieren. Dieses Verfahren ist fehleranfällig [9] und die durchgeführten Studien können, aufgrund möglicher Änderungen der benutzten Systeme, in der Regel nicht reproduziert werden. ii) Selbst wenn die Systeme angepasst werden können, ist dies nur durch Experten möglich. Eine JavaScript basierte Manipulation einer Webseite, wie in [4] durchgeführt, benötigt Kenntnisse in der Web Entwicklung und ist für Laien auf diesem Gebiet nicht möglich. Entsprechend können solche Studien nur von Forschern mit fachspezifischem Wissen durchgeführt werden.

Das erste Ziel dieser Arbeit ist es, ein Werkzeug zur Verfügung zu stellen, mit welchem das Erstellen solcher Untersuchungen erleichtert werden soll. Dabei soll es möglich sein, beliebige Arten von Applikationen als Attrappen nachzubauen und diese dann für die Untersuchungen zu verwenden. Um das Ganze zu vereinfachen sollen Screenshots bzw. als HTML heruntergeladene Webseiten als Vorlagen verwendet werden. Durch diese Methode soll der aufwändige und kostenintensive Nachbau von Applikationen vermieden werden, wodurch eine größere Gruppe die Möglichkeit erhält z.B. Verzögerungen in Software zu untersuchen. Ein weiteres Ziel dieser

Arbeit ist die Untersuchung des Einflusses der vorgestellten Methode auf die Bewertung der QoE. Zu diesem Zweck wird mittels der Methode eine E-Commerce Untersuchung für ein weit verbreitetes Anwendungsbeispiel erstellt und durch verschiedene Nutzer bewertet. Dabei soll der Einfluss spezifischer Designaspekte auf die Nutzerwertung herausgearbeitet werden. Dies beinhaltet den Einfluss der Führung des Nutzers durch die Aufgabe, wie die Untersuchung erstellt wurde, und in wie weit die Nutzerwertung durch Designaspekte wie Ladeanimationen beeinflusst werden kann. Zuletzt soll die Toleranz der Nutzer bezüglich der Wartezeiten evaluiert und Grenzwerte bezüglich der maximalen Wartezeiten identifiziert werden.

Diese Arbeit ist wie folgt gegliedert. Kapitel 2 beginnt mit den für das Verständnis benötigten Grundlagen. Dabei werden Begriffe rund um das Thema Web QoE und Crowdsourcing näher erläutert. Zusätzlich werden verwandte Arbeiten basierend auf deren Ergebnissen untersucht, um das Testszenario weitestgehend anpassen und vergleichbare Testbedingungen bereitstellen zu können. In Kapitel 3 wird der im Rahmen dieser Arbeit erstellte Editor vorgestellt und erklärt. Hierbei werden sowohl technische Aspekte erläutert als auch die Funktionsweise im Detail sowie ein Anwendungsbeispiel präsentiert. Folgend wird in Kapitel 4 die Methodik für die Integration der mit dem Editor erstellten Tests in die Testumgebung beschrieben. Weiter werden die Testparameter und die Nutzergruppen vorgestellt, die bei der Durchführung der Testfälle angewendet wurden und teilgenommen haben. Anschließend werden in Kapitel 5 die Ergebnisse der durchgeführten Experimente gezeigt. Zunächst werden Filterkriterien für die Datensätze zur Identifikation von falschen Angaben und Cheatern beschrieben. Auf den restlichen Daten erfolgt die Auswertung. Fokus liegt hierbei auf der vom Nutzer erfahrenen QoE und deren Einflussfaktoren. Abschließend folgt in Kapitel 6 eine Zusammenfassung des hier vorgestellten Editors, der durchgeführten Experimente und der wichtigsten Aussagen.

2 Grundlagen und verwandte Arbeiten

Das folgende Kapitel beschäftigt sich mit der Erläuterung grundlegender Begriffe aus dem Bereich Qualität im Web sowie einer Beschreibung der Begriffe bezüglich der zu untersuchenden Faktoren, wie z.B. Verzögerungen, die zum Verständnis dieser Arbeit beitragen sollen. Weiterhin wird Einsicht in verwandte Arbeiten gegeben, die bereits Qualität im Web und anderen Bereichen untersucht haben. Dazu gehören neben Verzögerungen in Audio- und Videostreaming auch Verzögerungen in Webseiten. Ebenso wie Beispiele aus dem täglichen Leben, z.B. die Bedienung an der Kasse im Schnellrestaurant. Anhand der Szenarien und Ergebnissen der verwandten Arbeiten soll mit einem ähnlichen Testaufbau geprüft werden, ob oder wie sich die Ergebnisse mittels des hier vorgestellten Tests unterscheiden bzw. rekonstruiert werden können.

2.1 Qualitätsbegriff in technischen Systemen

Im Allgemeinen beschreibt Qualität das Maß, in welchem ein Produkt gewisse Anforderungen oder Merkmale erfüllt [10]. Die Betrachtung der Qualität ist aus subjektiver bzw. objektiver Sicht möglich. Die subjektive Sicht beschreibt eine personenbezogene Bewertung, Eigenschaften wie z.B. „hübsch, interessant oder fair“, während sich die objektive Bewertung auf messbare Eigenschaften wie Höhe, Geschwindigkeit oder Dauer bezieht. Angesichts dessen lassen sich für das Web zwei Qualitätsbegriffe unterscheiden, die Quality of Experience (QoE) und die Quality of Service (QoS). Basierend auf diesen Kategorien werden für die QoS Messungen durchgeführt und die Ergebnisse ausgewertet, während zur Bestimmung der QoE persönliche Bewertungen von Nutzern, z.B. über einen Dienst, mittels einer Umfrage gesammelt und ausgewertet werden.

2.1.1 Quality of Service vs. Quality of Experience

Quality of Service fasst nach [11] alle Charakteristiken eines Telekommunikationsdienstes anhand seiner Fähigkeit zum Erfüllen gegebener und implizierter Anforderungen, die der Nutzer an den Dienst hat, zusammen. Einfacher gesagt ist damit die Gesamtleistung eines Dienstes - speziell auf die vom Nutzer erfahrene Dienstgüte bezogen – gemeint [12].

Um die QoS zu messen werden quantitative Größen, wie z.B. Jitter, Paketverlust, Durchsatz oder Verzögerung, bestimmt. Dies entspricht technischen Leistungsgrößen, die anhand von analytischen Daten, Simulationen oder empirischen Messungen erhoben werden. Die Auswertung der Resultate kann zur Analyse und Optimierung

eines Dienstes genutzt werden, z.B. um fehlende Ressourcen zu erkennen und bereitstellen zu können.

Im Laufe der Zeit entwickelte sich die Meinung, dass die QoS nicht ausreichend Aussagekraft besitzt, um die Eigenschaften, die Fülle und die Qualität eines Dienstes anhand der technischen Leistungsgrößen vollständig zu beschreiben [13]. Für diesen Zweck erweitert die QoE das QoS Konzept, indem es eine weitere Sicht auf die rein technischen Daten gewährt. So werden technische Systeme, Anwendungen uvm. vereint.

„Quality of Experience (QoE) is the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and / or enjoyment of the application or service in the light of the user’s personality and current state.“ lautet die Definition nach [13].

Demnach wird der Grad an Zufriedenheit bzw. Ärger, den ein Nutzer bei einer Applikation oder eines Dienstes empfindet, beschrieben. Die subjektive Wahrnehmung setzt sich zusammen aus dem Grad der Erfüllung seiner Erwartungen oder Zufriedenheit bezüglich der Nützlichkeit eines Systems oder Dienstes. Sie ist abhängig von seiner Persönlichkeit und aktuellen Verfassung.

Einem Nutzer werden Stimuli zur Bewertung vorgeführt. Seine Bewertung wird in der Regel im Rahmen einer Umfrage erhoben. Auf diese Weise lässt sich beispielsweise die Nutzerfreundlichkeit eines Programms bewerten. So wird die QoE überall dort verwendet, wo verschiedenste Systeme und Applikationen analysiert werden müssen und die QoS nicht ausreichend ist.

Um die QoE in Zahlen ausdrücken zu können ist eine Messgröße notwendig, da sich subjektive Wahrnehmungen wie „gut“ oder „sehr gut“ nicht als solches auswerten lassen. Die Messgrößen sind z.B. bezüglich einer Bewertungsskala, wie 1-5, 0-100 usw., zu bestimmen. Die unterschiedlichen Bewertungen (Opinion Scores) für gleiche Stimuli werden dann in der Regel als Mittelwerte zusammengefasst.

2.2 Methoden zur Erfassung der subjektiven Wahrnehmung

Für die Bestimmung der QoE aufgrund von subjektiven Wahrnehmungen sind verschiedene Methoden möglich. In [14] werden die Verfahren anhand der Bewertung von Fernsehbildern und kurzen Videosequenzen beschrieben. Die Bewertungen werden mittels verschiedener Skalen, wie in Abbildung 2.1 dargestellt, zu einem bestimmten Kontext abgegeben und anschließend weiter verarbeitet.

- Single Stimulus (SS)

Bei dieser Methode wird ein Bild vorgeführt. Die Bewertung kann kategorisch numerisch z.B. auf einer Skala von 1-10 erfolgen. Ein kategorisch adjektivischer Ansatz wäre beispielsweise eine Bewertung der Qualität in „1 schlecht“ oder „5 ausgezeichnet“. Je nach Kontext kann z.B. nach „1 sehr störend“ oder „5 nicht

Wert	Qualität
1	schlecht
2	dürftig
3	ausreichend
4	gut
5	ausgezeichnet

(a) kategorisch adjektivische Skala zur Bewertung der Qualität nach [14]

Wert	Beeinträchtigung
1	sehr störend
2	störend
3	ein wenig störend
4	wahrnehmbar, nicht störend
5	nicht wahrnehmbar

(b) kategorisch adjektivische Skala zur Bewertung der Qualität mit Beeinträchtigungen nach [14]

Wert	Qualität
-3	viel schlechter
-2	schlechter
-1	ein wenig schlechter
0	identisch
+1	ein wenig besser
+2	besser
+3	viel besser

(c) kategorisch adjektivische Skala zur Bewertung der Qualität im Vergleich nach [14]

Tabelle 2.1: Verschiedene Bewertungsskalen für die Erfassung der subjektiven Wahrnehmung

wahrnehmbar“ gefragt werden. Weiter sind auch adjektivische und numerische Ansätze ohne die Einteilung in Kategorien möglich.

- Double Stimulus Impairment Scale (DSIS)
In diesem Fall werden zwei Bilder gezeigt. Als erstes das Quellbild und als zweites das veränderte Quellbild. Daraufhin wird die Qualität des zweiten Bildes nach einer kategorisch adjektivischen Beeinträchtigungsskala bewertet, wobei die Beeinträchtigung z.B. als „1 sehr störend“ oder „5 nicht wahrnehmbar“ definiert wird.
- Double Stimulus Continuous Quality Scale (DSCQS)
Das Szenario ist hier wie bei DSIS. Die Bewertung erfolgt jedoch auf einer nicht kategorischen adjektivischen oder numerischen Skala.
- Stimulus Comparison (SC)
In dieser Methode wird die Qualität aufgrund eines Vergleichs bestimmt. Dazu werden zwei Bilder nebeneinander vorgeführt. Die Bewertung kann z.B. auf einer kategorisch-adjektivischen Vergleichsskala, wie „viel schlechter“ oder „viel besser“, erfolgen. Ein nicht kategorischer adjektivischer Ansatz wäre z.B. eine Skala beginnend mit „viel besser“ und endend auf „viel schlechter“.
- Single Stimulus Continuous Quality Evaluation (SSCQE)
Hier geht es um die Bewertung einer kurzen Videosequenz. Die Bewertung

wird mit einer nicht kategorisch adjektivischen Skala kontinuierlich während der Vorführung aufgenommen.

- Simultaneous Double Stimulus for Continuous Evaluation (SDSCE)
Bei diesem Verfahren wird wie bei SSCQE bewertet. Statt der beeinträchtigten Sequenz allein werden Referenz und Testsequenz nebeneinander abgespielt.

Basierend auf diesen Methoden werden die Nutzerwertungen (Opinion-Scores) gesammelt. Sofern es sich um nicht numerische Bewertungen handelt, werden den Kategorien Werte zugeordnet, wie z.B. in Tabelle 2.1a, 2.1b und 2.1c zu sehen ist. Für die Auswertung der Daten werden die Nutzerwertungen für gleiche Stimuli in der Regel als Mittelwerte zusammengefasst, dem Mean-Opinion-Score.

Der Mean-Opinion-Score (MOS) wurde ursprünglich für die Bewertung der Sprachqualität in Konversationen verwendet. Bei den Untersuchungen in [15] sollte ein Freiwilliger Audiodateien anhören und nach diskreten Werten von 1-5 bewerten. Dieser Test wird als „absolut category rating test (ACR)“ bezeichnet. Im Falle der Sprachqualität können z.B. Rauschen, Verzögerungen oder Echos die erfahrene Dienstgüte beeinträchtigen. Mit der Popularität von Videos im Netz, wie z.B. Youtube, fand diese subjektive Bewertungsmethode im Laufe der Zeit Anwendung beim Video-streaming, folgend von vielen anderen Bereichen.

In den hier durchgeführten Untersuchungen werden die Bewertungen mit einer kategorisch adjektiven Skala (Tabelle 2.1b) bezüglich der Beeinträchtigung durch die Verzögerungen abgegeben. Anschließend wird das arithmetische Mittel der Ergebnisse bestimmt. Die resultierenden Mean-Opinion-Scores werden für die Auswertungen verwendet.

2.3 Verzögerungsarten

Der hauptsächliche Untersuchungsfaktor dieser Arbeit sind Verzögerungen und deren Einfluss auf die wahrgenommene Quality of Experience. Bei Verzögerungen muss zwischen zwei verschiedenen Arten differenziert werden.

2.3.1 Eingabeverzögerung

Die Eingabeverzögerung beschreibt Wartezeiten, die mit I/O-Geräten zusammenhängen, sprich Eingaben von Maus und Tastatur. Dies tritt insbesondere bei sogenannten „thin-client“ Architekturen auf, bei der Applikationen auf einem zentralen Server installiert werden statt einzeln auf separaten Klienten. Der Nutzer greift über das Internet auf die Anwendungen zu und die vom Nutzer erfahrene Dienstgüte hängt hier stark von Parametern der QoS ab. Bei solchen Fernzugriffen können Eingabeverzögerungen auftreten. Das heißt, man sieht die Eingabe auf dem entfernten Rechner nicht zeitgleich mit der Eingabe am Gerät. Der Einfluss dieser Verzögerungen auf die vom Nutzer wahrgenommene Dienstgüte wurde für Office Anwendungen in [16] und [17] untersucht. Solche Wartezeiten treten bei einer lokalen Nutzung von Applikationen höchstens in Folge eines ausgelasteten Systems auf.

2.3.2 Ladeverzögerung

Eingabeverzögerungen sind jedoch nicht Teil der Untersuchungen dieser Arbeit, vielmehr geht es um die Ladeverzögerungen im Webkontext, auch „Page-Loading-Time (PLT)“ genannt. Aus technischer Sicht entspricht das der „round-trip-time“, die sich aus der Zeit, die eine Anfrage braucht um an dem Server anzukommen, der Zeit, die der Server zur Verarbeitung der Anfrage benötigt und der Zeit, bis die Antwort beim Nutzer ankommt, zusammen.

Diese Art von Verzögerung tritt immer dann auf, wenn Inhalte geladen oder nachgeladen werden müssen. Das Nachladen von Elementen ist asynchroner Inhalt und wird per sogenannten „Ajax“-Aufrufen realisiert. Dies können verkettete Auswahl-elemente auf Webseiten sein oder auch z.B. ein Chat der mit „polling“ umgesetzt ist.

2.4 Crowdsourcing

Jeff Howe beschreibt in seinem Artikel „The Rise of Crowdsourcing“ [18] 2006 erstmals die Entstehung von Crowdsourcing. Es ist vom Prinzip her angelehnt an das verwandte Outsourcing, welches durch Auslagern von Arbeit an Gruppen in anderen Ländern mit dem Ziel Kosten zu sparen definiert ist. Beim Crowdsourcing wird jedoch durch Aufteilung der Jobs in kleinere Aufgaben die Verteilung auf eine Masse, der „Crowd“, ermöglicht.

Howe erklärt wie durch den Fortschritt in der Technologie, durch innovative Plattformen und den Preisverfall für professionelles Equipment die Lücke zwischen Amateuren und Fachleuten weiter geschlossen wird. Zusätzlich ermöglicht das Internet den Zugang zu den kleinen Teilaufgaben für größere Gruppen, die die Bearbeitung übernehmen können. Abbildung 2.1 zeigt die grundlegende Verteilung der Arbeit zwischen Arbeitgeber und Arbeitnehmer vom Outsourcing zum Crowdsourcing.

Auf der linken Seite wird die Granularität der Tasks dargestellt, die mit jeweils der nächsten Stufe immer feiner wird. Auf der ersten Stufe wird ein ganzes Projekt über Outsourcing vermittelt, während auf der vierten Ebene sogenannte Tasks bzw. Microtasks an die Crowd übergeben werden. Durch die Verteilung der Microtasks in die Crowd kann ein Unternehmen auf das Wissen und die Arbeitskraft der Gesamtheit zugreifen. Zusätzlich öffnet sich durch die geringen Kosten für Jobs ein großer Ressourcenpool für Jedermann und viele Projekte, die ansonsten möglicherweise keine Chance hätten.

2.4.1 Anwendungsbeispiele

Crowdsourcing findet sowohl in kommerziellen als auch in freien Projekten Anwendung. Kommerzielle Projekte sind z.B. „Amazon Mechanical Turk“ und „Microworkers“. Auf diesen Plattformen können Auftraggeber Kampagnen mit Jobs online stellen. Die Jobs werden über die Plattform an Nutzer verteilt und können gegen eine Vergütung erfüllt werden. Ein Nutzer kann sowohl Auftraggeber als auch Auftragnehmer sein. Die Plattformen stellen bereits einige Templates zur Erstellung von

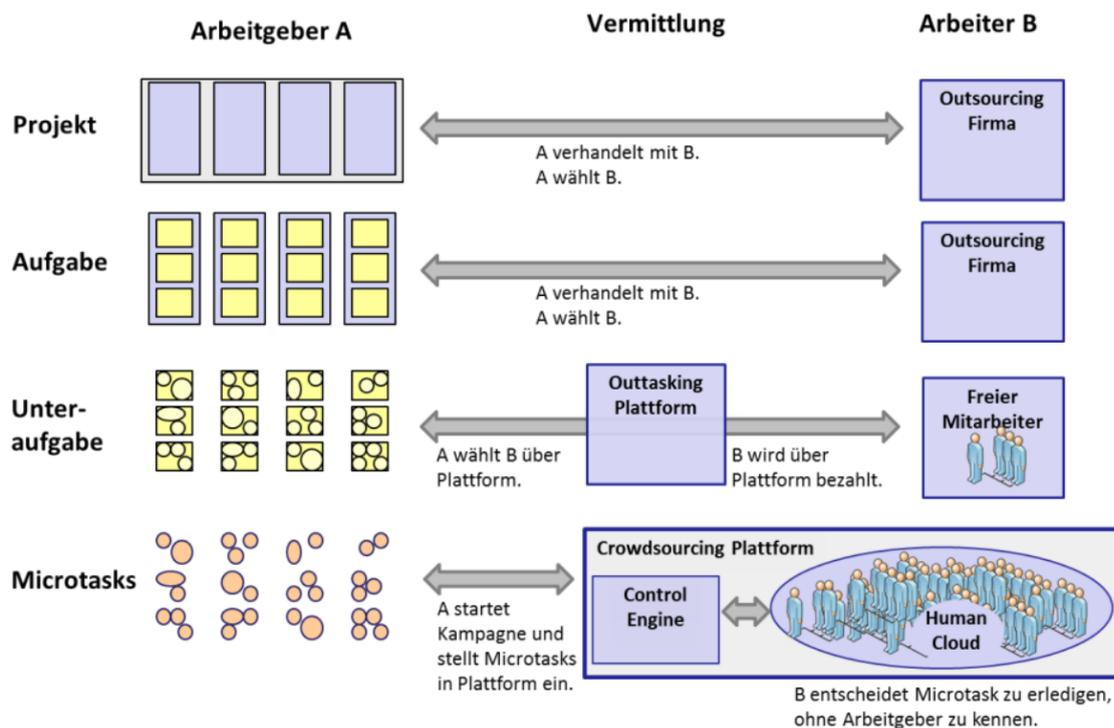


Abbildung 2.1: Crowdsourcing Granularität der Tasks im Zusammenspiel von Arbeitgeber und Arbeitnehmer nach [19].

Kampagnen zur Verfügung. Sie können jedoch komplett individuell gestaltet und integriert werden. Die Nutzer der Microworkers-Plattform werden im Rahmen dieser Arbeit als Teilnehmer für die Studien eingesetzt.

„Wikipedia – die freie Enzyklopädie“ [20] sagt durch seinen Titel aus, dass es aus freien Inhalten besteht. Es gehört zu den bekanntesten freien Projekten. Alle Nutzer können ihr Wissen zur Enzyklopädie beitragen und Artikel erstellen. Es existiert keine redaktionelle Überprüfung und Korrektur. Das System basiert darauf, dass sich die Nutzer untereinander selbst kontrollieren und korrigieren, was einen großen manuellen Aufwand erfordert. Ein ähnliches nicht kommerzielles Projekt ist z.B. „OpenStreetMap“ [21]. OpenStreetMap ist eine veränderbare Karte der Welt. Sie steht frei zur Verfügung und das Projekt zielt darauf ab, neue und interessante Ideen durch diese Daten zu fördern.

Zu Forschungszwecken wird Crowdsourcing für die Bestimmung der QoE für unterschiedliche Anwendungen/ Dienste verwendet. In [22] wird die QoE für Youtube-Videos mittels Crowdsourcing bestimmt und analysiert. Dabei wird der Einfluss von sogenannten „stallings“ (Pausen während Streaminginhalten) als hauptsächliches Kriterium, für die vom Nutzer erfahrene Dienstgüte, identifiziert. Es wird eine kosteneffiziente und flexible Methode vorgestellt, die zur Bestimmung der QoE von Streaminginhalten mittels Crowdsourcing verwendet werden kann. Problematisch ist hier die Verlässlichkeit der erhobenen Daten. Diese müssen mit zusätzlichen Mechanismen verbessert/ validiert werden. In den hier durchgeführten Experimenten wird diese Methode für die Bestimmung der QoE von Webapplikationen angepasst.

Eine andere Anwendung um den QoE mittels Crowdsourcing zu bestimmen, fin-

det sich bei „Burger King“. Seit einigen Jahren sammelt die Kette Kundenfeedback über die erfahrenen Wartezeiten während des Besuchs. Abbildung 2.2 zeigt einen Ausschnitt aus einer solchen Umfrage.



The screenshot shows a survey interface for Burger King. At the top left is the Burger King logo. To its right is the title "My BURGER KING® Experience" with "Satisfaction Survey" in smaller text below it. The main content area contains the question: "Bitte bewerten Sie Ihre Gesamtzufriedenheit mit Ihrer Erfahrung in diesem BURGER KING® Restaurant." Below the question is a horizontal scale with five radio buttons. Above the scale are the labels: "Sehr zufrieden", "Zufrieden", "Weder noch", "Unzufrieden", and "Sehr unzufrieden". A blue "Weiter" button is centered below the scale. At the bottom, a progress bar labeled "Fortschritt" shows a small red segment and the text "3%".

Abbildung 2.2: Screenshot der Burger-King Feedbacksammlung [23]

2.4.2 Microworkers.com

Microworkers stellt als dienstleistende Plattform eine Verbindung zwischen Auftraggebern und -nehmern dar. Dort können Jobs in Form von sogenannten Microtasks angeboten und bearbeitet werden. Die Plattform selbst wirbt mit 756.000 Nutzern weltweit. Ein Nutzer der Plattform übernimmt dabei die Rolle von Auftraggeber (Employer) und/ oder Auftragnehmer (Worker). Der Ablauf einer Kampagne bei Microworkers wird in Abbildung 2.3 veranschaulicht.

Auftraggeber können sogenannte Campaigns erstellen (1.), welche über die Plattform an die Nutzer verteilt werden (2.). Erledigte Jobs (3.-4.) erreichen den Auftraggeber, der die Kampagne auf der Plattform bezahlen muss (5.). Der Worker erhält durch einen Verifizierungscode seine Vergütung (6.).

2.4.3 Crowd als Forschungsgruppe

Die Crowd als Forschungsgruppe lässt sich für verschiedene QoE Studien, wie am Beispiel vom Videostreaming in [22] beschrieben wird, einsetzen. Für die hier durchgeführten Untersuchungen wird die QoE für Webapplikationen basierend auf Verzögerungen erfasst. In „Best Practices for QoE Crowdttesting“ [48] werden Grundpraktiken sowie Tipps und Tricks für erfolgreiche Untersuchungen mit der Crowd, z.B. die Teilnehmer von Microworkers.com, vorgestellt und analysiert.

Als erstes sollte sich die technische Implementierung der Untersuchung an der Verteilung der aktuellen Technologien orientieren. Für Video-Studien wird z.B. Flash empfohlen. Als nächstes ist beim Design der Aufgaben bzw. dem der Kampagnen

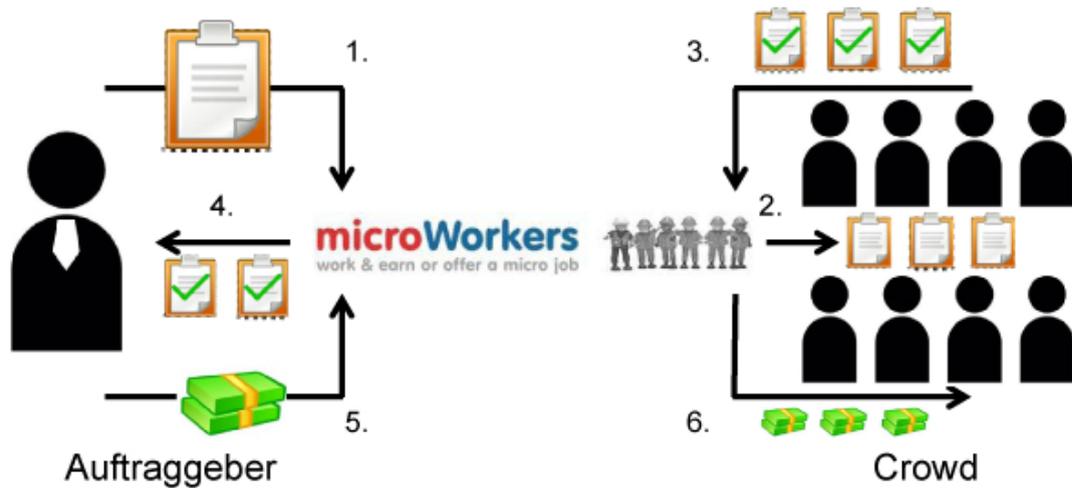


Abbildung 2.3: Ablauf bei einer Kampagne über die Microworkers-Plattform im Zusammenspiel von Auftraggeber und Auftragnehmer

darauf zu achten, dass dieses einem möglichen Mogeln zuvorkommt. Eine Möglichkeit ist, die Kampagne in mehrere Schritte aufzuteilen. So können z.B. im ersten Schritt verlässliche Teilnehmer festgestellt und anschließend zur eigentlichen QoE Studie weiter zugelassen werden.

Innerhalb der Aufgaben sollten mehrere Punkte berücksichtigt werden. Ein Ansatz gegen ein automatisiertes Ausfüllen von Formularen sind z.B. Captchas. Weiter können sogenannte Konsistenztests eingebaut werden. Am Anfang des Tasks wird beispielsweise der Nutzer aufgefordert seinen Kontinent anzugeben und zum Ende wird er nach dem Land gefragt. Weiterhin können Fragen zu den Inhalten der Umfrage gestellt werden.

Neben einer Bewertung z.B. von Verzögerungen kann auch grundlegend gefragt werden, ob eine Verzögerung festgestellt wurde. Ein Indiz für eine willkürlich ausgefüllte Umfrage ist dann eine nicht festgestellte Verzögerung bei gleichzeitig schlechter Bewertung.

Zusätzlich können die Zeiträume der jeweiligen Prozessschritte aufgezeichnet werden, um den Fokus des Nutzers während der Untersuchung einschätzen zu können.

2.5 Verwandte Arbeiten

Im folgenden Abschnitt werden verwandte wissenschaftliche Arbeiten zum Thema dieser Arbeit angeschnitten und inhaltlich wiedergegeben. Bei den Themen der Arbeiten geht es um Verzögerungen im Web, bei Audio-/ Video-Streaming, in Applikationen und bei der Nutzung von Diensten. Dabei wird die QoE und deren Einflussfaktoren untersucht. Die Resultate und Kernaussagen der Arbeiten dienen als Referenz für die hier durchgeführten Untersuchungen.

2.5.1 Einfluss von Wartezeiten und Verzögerungen im psychologischen Zusammenhang

Nielson erklärt 1993 in [24], dass 0.1 Sekunden die Grenze ist, bis zu welcher wir etwas als unterbrechungsfrei empfinden. Feedback ist dabei nicht relevant. Eine Sekunde entspricht dem Wert, bei dem der Gedankengang des Nutzers nicht unterbrochen wird, obgleich eine Verzögerung festgestellt wird. Ab zehn Sekunden geht die Aufmerksamkeit des Nutzers zum Kontext verloren und der Nutzer neigt dazu andere Aufgaben o.Ä. während der Wartezeit zu erledigen.

Weitere Studien zu neuronalen Prozessen und Antwortzeiten untersuchen die Toleranz für Verzögerungen. Die „2-second rule“ (zwei Sekunden Regel) beschreibt Miller 1968 in [25] im Bezug auf das Kurzzeitgedächtnis, welches eine große Rolle bei der Informationsverarbeitung spielt. Für ihn sind zwei Sekunden ausschlaggebend dafür, ob die Informationsverarbeitung im Gehirn ununterbrochen bleibt. Im Gegensatz dazu beziffert Nielson 1995 [26] die Antwortzeit bei Prozessen mit Computern auf eine Sekunde, während er für andere Gebiete 10 Sekunden veranschlagt. Zusätzlich wurden zwei Sekunden von Shneiderman 1984 [27] als Referenzantwortzeit für Computer empfohlen. Er untersucht die Antwortzeiten im Zusammenhang mit der Produktivität des Nutzers bei der Arbeit mit einem Computersystem. Neben den Antwortzeiten spielen hier Faktoren wie Fehlerraten, Arbeitsstil und Zufriedenheit eine Rolle. In den meisten Situationen erzielen sinkende Antwortzeiten eine höhere Produktivität und erhöhen gleichzeitig die Zufriedenheit. Damit steigt das Tempo des Arbeitsflusses und die Fehlerrate. Solange diese klein und schnell zu korrigieren sind, ist generell eine Erhöhung der Produktivität erreicht. Wenn sie jedoch schwer zu beheben sind, ist ein langsames Tempo möglicherweise ertragreicher.

In [1] wird untersucht, wie die qualitative Wahrnehmung eines Nutzers im Zusammenhang mit interaktiven Systemen von Wartezeiten dominiert wird. Abbildung 2.4 zeigt ein Beispiel für die diskutierten grundlegenden psychologischen Aspekte bei der Zeitwahrnehmung. Es wird gezeigt, dass diese so wie auch die QoE maßgebend subjektiv und kontextabhängig ist.

Anhand eines Beispiels aus dem täglichen Leben untersucht [28] die Wartezeiten in Fast-Food-Restaurants. Die Wartezeit wird mit der Kundenzufriedenheit bei einer Dienstonutzung in Relation zueinander gestellt. Diesbezüglich wurde ein Framework entwickelt und eine Teststudie durchgeführt, bei der gemessene Wartezeiten mit zugehöriger Umfrage ausgewertet wurden. Es wird verdeutlicht, dass aus Sicht des Managements von Diensten nur der Zusammenhang von Wartezeit zur Anzahl an Mitarbeitern von Interesse ist. Für das Marketing ist jedoch die Kundenzufriedenheit, basierend auf den Meinungen der Kunden über den Dienst, direkt im Fokus. Das Ergebnis brachte hervor, dass eine Reihe verschiedener personenbezogener Aspekte, zusätzlich zur Wartezeit, in Verbindung mit der Kundenzufriedenheit steht. Abbildung 2.5 zeigt, wie sich das Verhalten des Nutzers basierend auf Erwartungen und Erlebnissen ausbildet.

So sind z.B. verschiedene Tageszeiten (Mittag/ Abend), Werktag oder Wochenende und Umgebung/ Lage des Restaurants mitunter entscheidend, wie sich die Wartezeit auf die Bewertung auswirkt. Als besonderes Merkmal wurde auch die Tä-

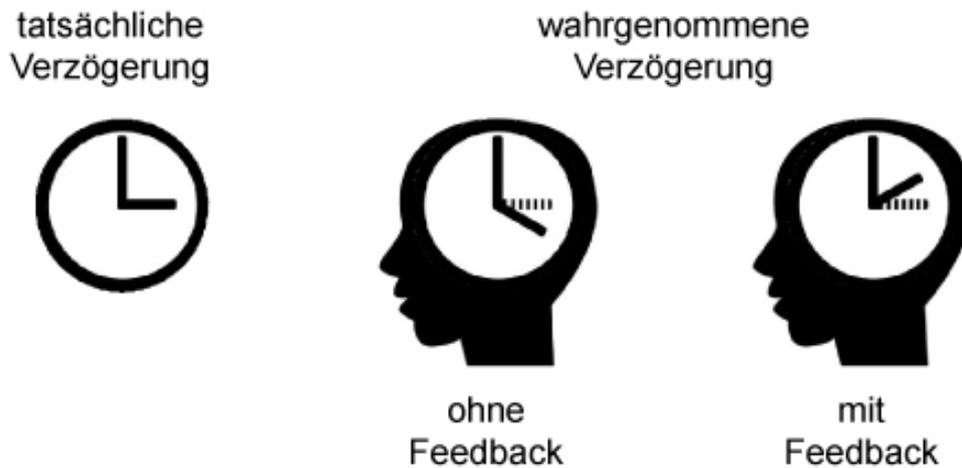


Abbildung 2.4: Unterschiedliche Wahrnehmung von Verzögerungen im Verhaltensmodell eines Nutzers nach [1] beispielsweise durch Manipulation der Erwartungshaltung.

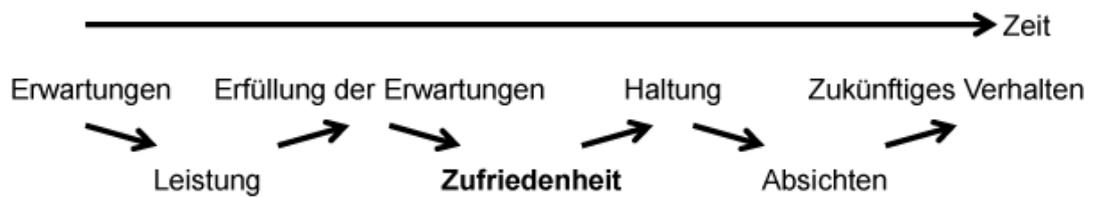


Abbildung 2.5: Rolle der Zufriedenheit im Verhaltensmodell der Nutzer nach [28]

tigkeit, die vor oder nach dem Besuch stattgefunden hat oder stattfinden wird, als Faktor festgestellt. Hier sind Arbeit und Schule insbesondere aufgefallen. Aus diesem Beispiel lassen sich Rückschlüsse darüber ziehen, dass auch für die Web-QoE viele Einflussfaktoren eine Rolle spielen.

[2] beschreibt die Verständnislücke zwischen subjektiver Wahrnehmung und QoE. Es wird untersucht, ob die Reduzierung auf den Aspekt des Wartens auf Fertigstellung einer Aufgabe in typischen Szenarien für die Bewertung von webbasierten Diensten ausreichend ist und dadurch Rückschlüsse auf logarithmische Gesetze in der menschlichen Zeitwahrnehmung möglich sind. Weiter wird gezeigt, dass diese Zuteilung für einfache Szenarien, wie dem Herunterladen einer Datei, zulässig ist, während die Betrachtung von interaktiven Webseiten sehr viel komplexer ist.

Die Ergebnisse der genannten Arbeiten können als Richtlinien für die Entwicklung von Benutzeroberfläche und der Interaktion mit dem Nutzer verwendet werden.

2.5.2 Einfluss von Wartezeiten und Verzögerungen in technischen Systemen und webbasierten Diensten

Weitere Forschungsarbeiten konzentrieren sich auf Verzögerungen beim Laden von Webseiten. Die Beziehung von QoE und der Seitenaufbauzeit (page-loading-time, PLT) wird in [2] und [1] in logarithmischen Zusammenhang gestellt.

Die Toleranz eines Nutzers gegenüber der Wartezeit in webbasierten Diensten, z.B. beim Laden von Webseiten zur Informationssuche, wird in [29] untersucht. Ziel der Studie war eine quantitative Bestimmung des Schwellwertes für die Dauer der Verzögerung, welche beim Nutzer zum Abbruch führt. Weiter werden die Auswirkungen von Feedback, z.B. in Form eines Ladebalkens, auf die Toleranz analysiert. Konsistent zur Literatur wurde gezeigt, dass die Nutzer eine Antwortzeit von 2 Sekunden bei der Informationsbeschaffung im Web erwarten. 15 Sekunden sind für Nutzer als Abbruchkriterium für einen Download festgestellt worden, wenn keinerlei Informationen oder Anzeichen über einen Status bereitgestellt werden. Man stellte fest, dass das Feedback in Form einer Ladeanimation die Toleranz um 2 Sekunden erhöht. Zusätzlich wird gezeigt, dass Faktoren wie der Inhalt der Seite (Multimedia, Grafiken), Ziele des Nutzers, demografische Daten (Alter, Geschlecht), Zeitdruck oder eine Alternative zur Webseite die Toleranz beeinflussen können. Ähnliche Studien wurden in [30] durchgeführt. Die Arbeit zeigt, dass Faktoren, die die erfahrene Dienstgüte potentiell reduzieren können, durch effektive Verwaltung der Wartezeiten neutralisiert werden können. Dazu zählen Informationen über die Wartezeiten (Nutzer-Feedback), die Dauer und die Position der Verzögerung. Untersucht wird im Rahmen speziell angefertigter Konzepte für neue Internetseiten, die zur Bewertung stehen.

Eine Reihe von Untersuchungen behandeln die QoE im Kontext von E-Commerce. In [31] werden die Auswirkungen von Downloadzeiten auf die Haltung des Verbrauchers zum Internethändler betrachtet. Experimentelle Untersuchungen finden beispielsweise in [32] statt, wobei hier die Gesamtqualität der Seite zur Bewertung im Vordergrund steht. Die Seite wird speziell für die Untersuchung programmiert. In [33] werden die Auswirkungen der Seitenaufbauzeiten im E-Commerce Kontext beschrieben. Es werden Aspekte wie z.B. Design, Funktionalität und Alternativshops

zusätzlich zu Wartezeiten als QoE-Einflussfaktoren untersucht. Durch einfache und günstig zu implementierende psychologische Aspekte auf einer speziell für den Test gefertigten Seite wird die vom Nutzer wahrgenommene Verzögerung reduziert. Längere Verzögerungen verursachen kontinuierlich schlechtere Bewertungen. Bei fünf Sekunden Verzögerung ist es wahrscheinlicher, dass der Nutzer sich noch keine Alternative sucht. In [34] wird gezeigt, dass die vom Nutzer wahrgenommene Dienstgüte in das Design der Webseite eingearbeitet werden kann. Bei der Durchführung von QoS-Experimenten im E-Commerce Kontext stellte man fest, dass die Nutzer bei einer Verzögerung von null bis fünf Sekunden eine hohe Toleranz zeigen. Ab fünf Sekunden weisen sie eine mittlere und ab elf Sekunden keine Toleranz mehr auf. Aus den Resultaten kam hervor, dass die Nutzer sehr stark von ihrer Erwartungshaltung beeinflusst werden. Bei zufälligem Browsen wird die Beeinträchtigung durch Verzögerung nicht so schlecht bewertet wie bei gezieltem Surfen.

Weiter existieren Arbeiten zum Thema außerhalb des E-Commerce Kontexts. Zusätzlich zu einem formellen Teil in [35] wird die QoE für Video, Audio und das Web untersucht. Die Auswertung zeigt, dass für 90% der Nutzer bereits eine mittlere Qualität mit einem MOS-Rating von drei ausreichend ist. Es wird gezeigt, dass die Akzeptanz ein weiteres Maß für die vom Nutzer wahrgenommene Dienstgüte ist. Statt wie beim MOS nach der Bewertung über eine Seite zu fragen, kann man über die Akzeptanz überprüfen, wann ein Nutzer aufhört einen Dienst zu verwenden. Eine Untersuchung des Unterschieds zwischen einer anfänglichen Verzögerung und einer Unterbrechung ist in [36] gegeben. Die Resultate zeigen, dass die QoE für die untersuchten Wartezeiten stark von der vorliegenden Applikation abhängig ist. Speziell aufgefallen ist, dass die Nutzer sehr sensibel auf Unterbrechungen reagieren. Dies impliziert, dass z.B. bei Videostreaming eine größere anfängliche Verzögerung zum Vorladen des Videos angepeilt werden sollte um damit die Unterbrechungen während des Videos zu reduzieren.

2.5.3 Analyse des Einflusses von Verzögerungen in Web- und Business Applikationen

Der Einfluss von Verzögerungen auf die QoE wurde in der Masterarbeit „Analyse des Einflusses von Verzögerungen in Web- und Business Applikationen auf die erfahrene Dienstgüte“ [37] von Johann Scherer untersucht. Diese diente als Basis und Anstoß für die hier vorgestellte Methode zum Generieren von Tests.

Im Rahmen der Arbeit wurden spezifische Prozessabläufe, wie z.B. aus der SAP-basierten Business-Applikation Oscare, als Webapplikation nachgebaut. Verzögerungen wurden mittels einer speziell implementierten Javaskriptbibliothek in den Prozess eingebunden. Für die Untersuchung wurden die nachgebauten Tests mittels Crowdsourcing auf der Microworkersplattform durchgeführt und anhand von Umfragen, jeweils vor und nach dem Test, ausgewertet. Ziel der Untersuchung war einerseits der Einfluss von Eingabeverzögerungen und andererseits Ladeverzögerungen. Das Ergebnis zeigt, dass kleine Eingabeverzögerungen im Bereich von 2 Sekunden bereits sehr störend wirken, da hier ein hoher Grad von Interaktivität mitspielt. Bei Ladeverzögerungen lies sich kein aussagekräftiger Schwellwert bestimmen, ab welcher die QoE nicht mehr als akzeptabel bewertet wurde.

Bei der Auswertung der erhobenen demografischen und qualitativen Daten traten Schwierigkeiten auf. Aus reiner Sicht auf die Daten ließ sich nicht feststellen, ob diese gewissenhaft eingegeben oder die Umfragen einfach durchgeklickt wurden. Es wurden verschiedene Filter angewendet um Falschangaben und Cheater aus dem Datensatz zu entfernen. Das Ergebnis war die Reduktion des Datensatzes um die Hälfte und es verlor an Aussagekraft. Die Realisierung der Tests als Webapplikationen brachte ein weiteres Problem mit sich. Die tatsächliche Ladezeit für das Aufbauen der Testwebseiten kam in jedem Prozessschritt zusätzlich zur eingestellten Verzögerungszeit hinzu. Das Nachbauen der Applikationen als Webseiten ist mit viel Aufwand verbunden. Ein großer Teil der Untersuchungszeit wurde für die Programmierung investiert, was den Vorteil eines hohen Detailgrads der Anwendung mit sich bringt. Zusätzlich ist viel fachliches Wissen im Bereich Webprogrammierung notwendig.

Es wurde versucht möglichst viele der diskutierten Untersuchungsaspekte in das Testdesign sowie die Entwicklung des Editors einfließen zu lassen, sodass ein annähernder Vergleich durchgeführt werden konnte. Aus den oben genannten Arbeiten lassen sich folgende Schlüsse ziehen:

- Von den psychologischen Aspekten halten wir fest, dass die Wahrnehmung der Verzögerung ab etwa 2 Sekunden mehr in eine Unterbrechung übergeht. Man kann davon ausgehen, dass die Bewertung bis zu diesem Punkt gut ausfällt. Ab 10 Sekunden Wartezeit wird von einer sehr schlechten Bewertung ausgegangen.
- Die Forschung konzentriert sich weitestgehend auf die Auswertung von Webapplikationen. Unternehmenssoftware wird durch die schlechten Möglichkeiten zum Testen her gemieden und sollten mit in zukünftige Untersuchungen aufgenommen werden.
- Für die durchgeführten Tests wurde im Normalfall eine speziell programmierte Seite verwendet. Dafür ist ein hohes Maß an fachlichem Wissen Voraussetzung. Testen von Live-Systemen resultiert in nicht reproduzierbaren Analysen, da sich diese während der Laufzeit verändern können. Dies stellt weiterhin eine Herausforderung dar.
- Eine vereinheitlichte Methode zur Untersuchung des Einflusses von Verzögerungen in Applikationen, welche kein zusätzliches Expertenwissen erfordert, steht nicht zur Verfügung.
- Aufwändige Nachbauten bringen den Vorteil eines hohen Detailgrads mit sich. Durch die Realisierung als reine Webapplikation können zwischen den Prozessschritten unkontrollierbare Wartezeiten beim Laden der Seiten entstehen, was zur Verfälschung der Auswertung führen kann.

Aufbauend auf den Methoden und den Ergebnissen der verwandten Arbeiten wird klar, dass speziell Businessapplikationen weitere Untersuchungen erfordern. Insbesondere durch die zeitaufwändigen und komplexen Nachbauten aus [37] wird deut-

lich, dass eine einheitliche Methode zur Erstellung reproduzierbarer Tests ohne Expertenwissen in Webprogrammierung notwendig ist. Aus diesem Fakt entstand die Motivation für diese Arbeit.

Im folgenden Kapitel wird ein Editor zum schnellen und einfachen Generieren solcher Tests vorgestellt.

3 Framework

Das im Rahmen dieser Masterarbeit entstehende Framework entspricht einer Umgebung zur Erstellung von Simulationen (im Weiteren als Editor bezeichnet) basierend auf Vorlagen existierender Web- oder Businessapplikationen. Die mit dem Editor erstellten Emulationen werden als einzelne HTML-Datei zur Verfügung gestellt. Der nächste Abschnitt beschäftigt sich vorab mit technischen Details zur Implementierung des Editors und der Untersuchungsumgebung.

3.1 Verwendete Webtechnologie

Im Folgenden werden die Technologien vorgestellt, die für eine Realisierung des Editors als Webseite und der generierten Emulationen als Webapplikationen verwendet werden. Dies sind PHP, HTML, JavaScript und CSS. Durch diese Umsetzung sind sowohl Editor als auch die Emulationen zu einem Großteil der Geräte kompatibel.

3.1.1 PHP

Hypertext Preprocessor (PHP) ist die verbreitetste der serverseitigen Programmiersprachen und findet zu 81.1% Einsatz im Web. Auf Platz zwei liegt ASP.NET mit 18.8% und auf Platz drei Java mit 2.7% [38]. Perl oder Ruby sind beispielsweise exklusivere Sprachen und noch nicht so weit verbreitet.

PHP dient hauptsächlich der Erstellung von dynamischen Webseiten oder Webapplikationen und wird als kostenlose Software zur Verfügung gestellt. Merkmale von PHP sind eine gute Datenbankunterstützung sowie eine Vielfalt von Bibliotheken für allerlei Einsatzgebiete.

Der Ablauf beim Aufrufen einer Webseite, die auf PHP basiert, wird in Abbildung 3.1 dargestellt.

Der Benutzer stellt eine Anfrage für eine PHP Datei. Über das Internet wird nun die Datei auf dem Server geladen und an einen Interpreter weitergegeben. Der Code einer PHP Datei wird somit nicht beim Benutzer geladen sondern serverseitig vom Interpreter verarbeitet. Die Antwort auf den Aufruf ist die Verarbeitung der Daten, die wieder an den Benutzer übermittelt wird. Wie diese Antwort aussieht, hängt vom jeweiligen Skript ab. In den meisten Fällen handelt es sich um ein HTML-Dokument zum Darstellen einer Webseite. Es können jedoch auch z.B. PDF oder Excel Dateien generiert und zum Download angeboten werden. Dynamischer Inhalt kann innerhalb der HTML-Seite mit den Tags „<?php Befehl; ?>“ eingebaut werden.

Für die Ausführung der Dateien benötigt man ein System, welches die beinhalteten PHP Anweisungen interpretieren kann. Der dafür zuständige Interpreter wird durch einen Server-Daemon oder Server-Dienst (z.B. Apache) ausgeführt und läuft

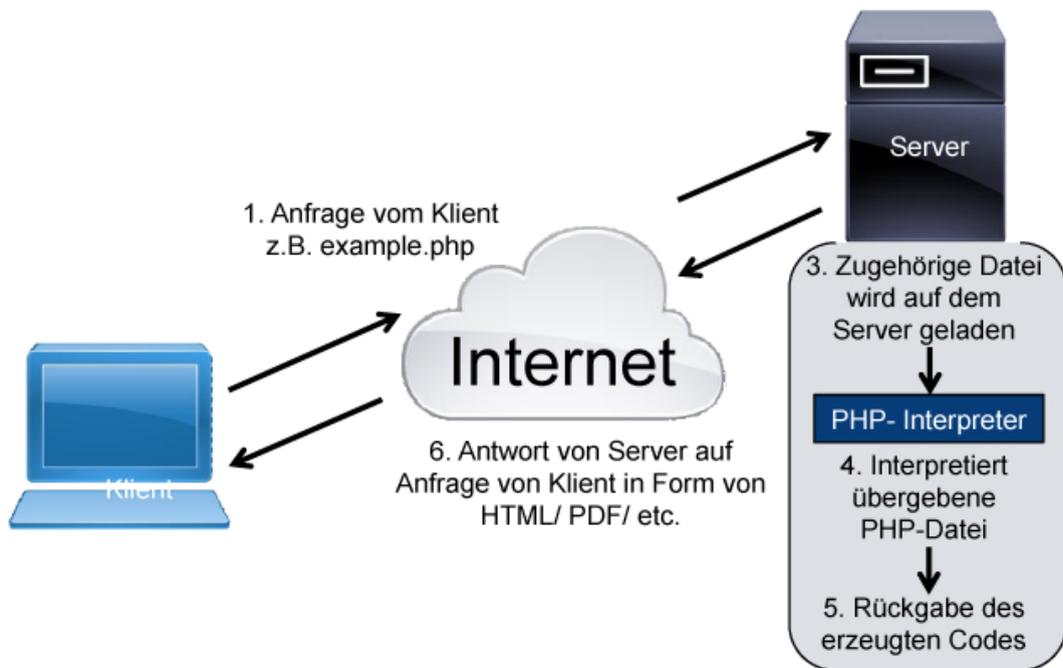


Abbildung 3.1: Funktionsweise von PHP nach [39]

als sogenannter Webserver. Als Datenbanksystem wird MySQL verwendet. Die Kombination der Elemente PHP, Datenbank und Webserver mit einem Betriebssystem wird als WAMP (Windows)/ LAMP (Linux)/ MAMP (Mac) bezeichnet. Als fertige Lösungen gibt es z.B. XAMPP, welches jedoch nur als Entwicklungssystem empfohlen wird.

PHP findet häufig Anwendung bei Foren, Content-Management-Systemen, E-Commerce-Systemen und vielen mehr. Bekannte Frameworks wie z.B. TYPO3 und Wordpress arbeiten mit PHP.

3.1.2 HTML

Mit „Hypertext Markup Language“ (HTML) wird eine Auszeichnungssprache zur Strukturierung digitaler Inhalte bezeichnet. 1989 erfand Tim Berners-Lee das Konzept für die Sprache und 1990 wurde der erste Prototyp für einen Webbrowser vorgestellt [40], womit die Grundlage für das World Wide Web geschaffen war. Die aktuelle Version ist seit dem 28. Oktober 2014 HTML5 [39].

HTML entspricht keiner Programmiersprache im eigentlichen Sinn. Durch die Sprache wird beschrieben, welcher Inhalt wie strukturiert dargestellt wird. Die Webbrowser interpretieren die HTML-Dokumente und stellen dadurch Inhalte, wie ursprünglich Texte mit Hyperlinks und Bilder, semantisch strukturiert, jedoch nicht formatiert, dar. Im Laufe der Zeit entwickelte sich die Technologie und mittlerweile sind Videos ein Standard und interaktive Inhalte wie z.B. Spiele sind immer mehr im Trend.

3.1.3 JavaScript und JQuery

JavaScript ist die populärste Programmiersprache und wurde 1995 von Brendan Eich entwickelt [41]. Die statische Darstellung der HTML-Seite ist oft nicht ausreichend. Viele Aspekte machen die Abhängigkeit zu JavaScript notwendig.

Eine Webseite kann beispielsweise mittels JavaScript die aktuelle Breite und Höhe des Browsers in Echtzeit ermitteln und anhand dieser Informationen die Elemente in Relation skalieren und positionieren. In Formularen ist meist eine Validierung der Eingabefelder, wie z.B. nach einer korrekten Email, notwendig. Es können Inhalte nachgeladen werden, ohne die gesamte Seite zu erneuern. Durch JavaScript kann eine Webseite in vielerlei Hinsicht interaktiver gestaltet werden.

Der Skriptcode kann z.B. direkt innerhalb des jeweiligen HTML-Dokuments mit umschließenden „<script>“-Tags eingebunden werden. Eine höhere Lesbarkeit und Wiederverwendbarkeit von Funktionen erreicht man jedoch durch referenzieren einer JavaScript-Datei (.js) im „<head>“-Bereich des HTML-Dokuments.

JQuery ist eine JavaScript-Bibliothek, die das Ansteuern und Manipulieren von HTML-Elementen, das Verwalten von Events, Animationen und asynchrone Aufrufe erleichtert und eine leicht verständliche API mitbringt. Weiterhin ist JQuery mit seiner Flexibilität und Erweiterbarkeit in Form von Plugins eine beliebte Art um JavaScript zu schreiben [42].

3.1.4 CSS

Zur Gestaltung der HTML-Elemente werden sogenannte „Cascading Style Sheets“ (CSS) verwendet. Dazu wird in einer CSS-Regel beschrieben, wie ein Element auszusehen hat und wie es positioniert werden soll. Die Sammlung solcher Regeln wird in einer CSS-Datei (.css) gespeichert. Die Zugehörigkeit zu einem HTML-Dokument lässt sich innerhalb des „<head>“-Tags definieren. Dabei wird mittels des „<link rel=stylesheet" ...>“-Tags eine CSS-Datei referenziert. Optional können CSS-Regeln direkt innerhalb der „<style>“-Tags definiert werden.

HTML war ursprünglich nicht dafür gedacht, Tags, die das Aussehen definieren, zu beinhalten. Die Einführung der Schrift- und Farbtags in HTML 3.2 bedeutete einen enormen Aufwand für Entwickler, wenn diese auf großen Seiten vielen Inhalten durch extra Tags Gestaltungsdetails hinzufügen mussten [43]. Mit CSS werden diese Informationen aus der Strukturseite in eine Darstellungsdatei ausgelagert, was eine höhere Lesbarkeit des Codes mit sich bringt. Zusätzlich wird eine Menge an Code gespart, da mittels einer Klassendefinition direkt eine ganze Art von HTML-Elementen, z.B. die „<p>“-Elemente, ein Aussehen bekommen kann.

3.2 Framework Übersicht

Das Ziel der Arbeit ist eine Umgebung zur Verfügung zu stellen, in welcher beliebige Applikationen nachgebaut werden können. Die Nachbauten werden in eine erweiterte Untersuchungsumgebung integriert. Um dies so einfach wie möglich zu gestalten,

sind mehrere Schritte nötig. Abbildung 3.2 zeigt eine Gesamtübersicht über das Framework.

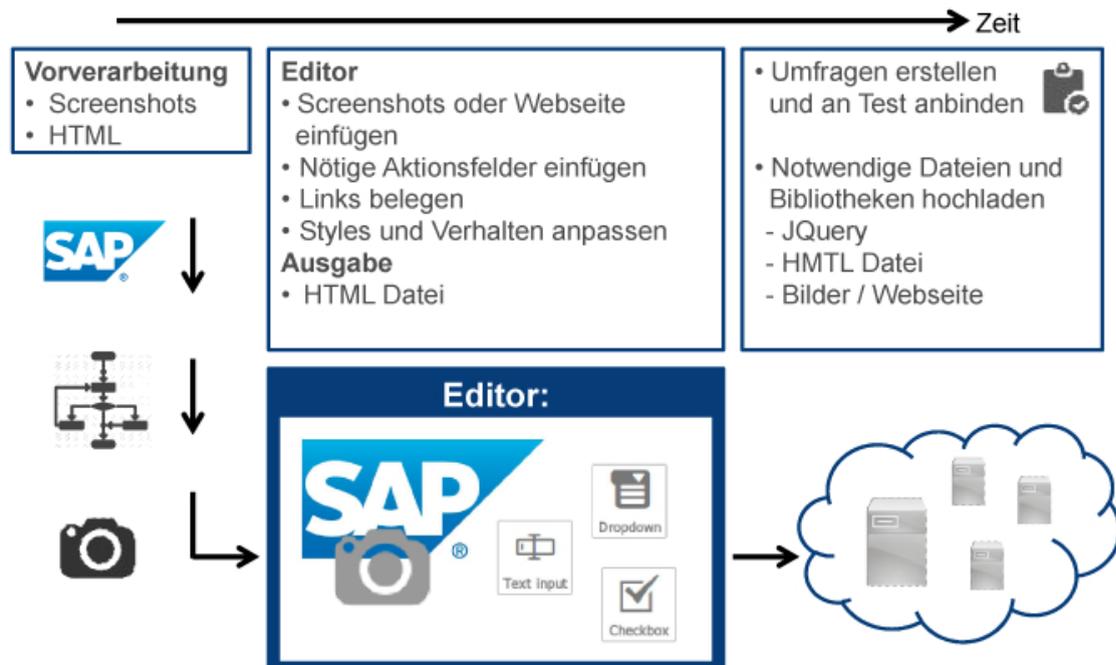


Abbildung 3.2: Übersicht über das Framework

Zum Erstellen einer Untersuchung ist als erstes eine Vorverarbeitung durchzuführen. Hier werden die Vorlagen für die Emulationen erstellt. Anschließend können die Vorlagen im hier implementierten Editor eingebaut und zu Emulationen gestaltet werden. Die Emulationen werden dann mit zugehörigem Vorlagenmaterial und Bibliotheken auf einem Webserver bereitgestellt und an Umfragen angebinden, was die Untersuchung vervollständigt. Die einzelnen Schritte werden in den nächsten Abschnitten detailliert erklärt.

3.3 Vorverarbeitung

Um einen Prozess nachbauen und simulieren zu können muss dieser zunächst kristallisiert werden. Ist ein zu testender Prozessablauf gefunden, so müssen die Vorlagen für den Editor generiert werden. Hier sind zwei verschiedene Möglichkeiten gegeben. Dazu zählen Screenshots und HTML Webseiten. Im Weiteren wird dies als Vorlagenart differenziert.

1. Screenshots als Vorlagen

Wenn der Prozess mithilfe von Screenshots nachgebaut werden soll, wird von jedem Schritt im Prozess eine Momentaufnahme benötigt. Die Aufnahmen sollten sehr genau ausgeschnitten und nachbearbeitet werden, sodass später nicht der Eindruck einer Attrappe aufgrund von unsauberem Arbeiten entsteht. Ränder müssen beispielsweise ebenfalls entfernt werden, damit keine

doppelten Rahmen im Browser entstehen. Für Businessapplikationen muss dies immer manuell erfolgen. Webseiten kann man sich über online verfügbare Tools als Bilder ausgeben lassen. Weiter ist darauf zu achten, dass man gegebenenfalls die Daten anonymisiert oder Einträge entfernt.

2. HTML-Seiten als Vorlagen

Bei HTML-Seiten gibt es mehrere Möglichkeiten die notwendigen Dateien herunterzuladen. Die einfachste Variante stellt die in den Browsern eingebaute „Seite speichern unter...“-Funktion dar. Um eine möglichst vollständige Offlinekopie zu erhalten sollte hier nicht die einzelne HTML-Datei gewählt werden, sondern die Variante mit den Inhalten (Webseite komplett).

Eine weitere Möglichkeit sind Tools wie „wget“, „curl“ und „httrack“. Diese Tools sind wesentlich umfangreicher und bieten bis in den Expertenmodus Freiheitsgrade. Es kann beispielsweise die angegebene Seite inklusive der Elternknoten und aller referenzierten Knoten heruntergeladen werden. Zusätzlich ist eine automatische Anpassung der Links möglich. Werden HTML-Seiten als Muster eingebettet ist zu beachten, dass evtl. Inkompatibilitäten durch die externen Skriptinhalte entstehen können.

3.4 Editor

Mit dem Begriff „mockup“ wurde 1919 eine Simulation bzw. ein Modell bezeichnet [44]. Nach [45] entspricht die Definition von „mockup“ einem 1:1 Modell eines Gegenstandes, welches zu Vorführzwecken oder Forschung verwendet werden kann. Des Weiteren kann es ein funktionsfähiges Muster darstellen und beispielsweise in der Designforschung eingesetzt werden. Mit dem Begriff wird im Deutschen auch eine Attrappe bezeichnet [46].

Im Bereich des Internet findet man „mockup“ häufig im Zusammenhang mit „Tools“ bzw. Werkzeugen, in denen man Attrappen von Webseiten erstellen kann. Diese sind ohne Funktionalität und werden nur zu Demonstrationszwecken oder als Studien für Entwicklungen verwendet. So können z.B. Designvorlagen von einem Designer erstellt werden, die von einem Programmierer umgesetzt werden.

Die wichtigste Komponente des hier vorgestellten Frameworks ist der Editor. Er basiert auf dieser Idee und ist vom Design her dem Tool „moqups“ [47] angelehnt. Die Elemente und Funktionen des Editors wurden auf die Notwendigen reduziert, angepasst und umgestaltet. Darunter finden sich ein Panel für Webseitenelemente, ein Panel zum Hochladen und Einbinden von Vorlagen, Status- bzw. Werkzeugleisten sowie die eigentliche Arbeitsfläche.

3.4.1 Implementierung

Generell ist jede Sprache für die Implementierung geeignet. Für die Auswahl der Nützlichsten gilt es die Vorteile abzuwägen sowie die Voraussetzungen und Anforderungen zu untersuchen und dementsprechend auszuwählen.

Eine Realisierung als Webseite bietet sich dabei in vielerlei Hinsicht an. So lässt sich diese sehr einfach verbreiten und die elementare Voraussetzung ist ein Browser,

welcher auf allen Computern, Smartphones und Tablets, standardmäßig vorinstalliert ist. Da der Editor öffentlich zugänglich sein wird, stellt dies den Hauptgrund dar. Weiter müssen im Normalfall keine zusätzlichen Bibliotheken o.ä. installiert werden. Eine Entwicklung mit Flash kommt daher beispielsweise nicht in Frage, da dies als externes Plugin installiert werden muss. Lediglich JavaScript muss in diesem Fall im Browser aktiviert sein. Es ist für die Grundfunktionalität notwendig und ist Voraussetzung für die JQuery-Bibliothek. Diese Punkte erhöhen die Plattformunabhängigkeit.

Die Umsetzung des Editors im Web setzt jedoch keine Grenzen beim möglichen Funktionsumfang, so existiert beispielsweise ein in JavaScript nachgebautes Linux (<https://osjsv2.0o.no/>). Einen weiteren Vorteil stellt die Updateverwaltung dar. Updates werden vom Entwickler auf den Server hochgeladen und der Nutzer erhält beim Zugriff auf die Seite ohne zusätzlichen Aufwand die neueste Version.

Die Anforderung an den Editor war, aus Screenshots von Prozessen, diese mit Modifikationen und der Grundfunktionalität Nachbauen und Emulieren zu können. Basierend auf dieser Problemstellung ergaben sich die notwendigen Bestandteile. Abbildung 3.3 zeigt eine Momentaufnahme des Editors.

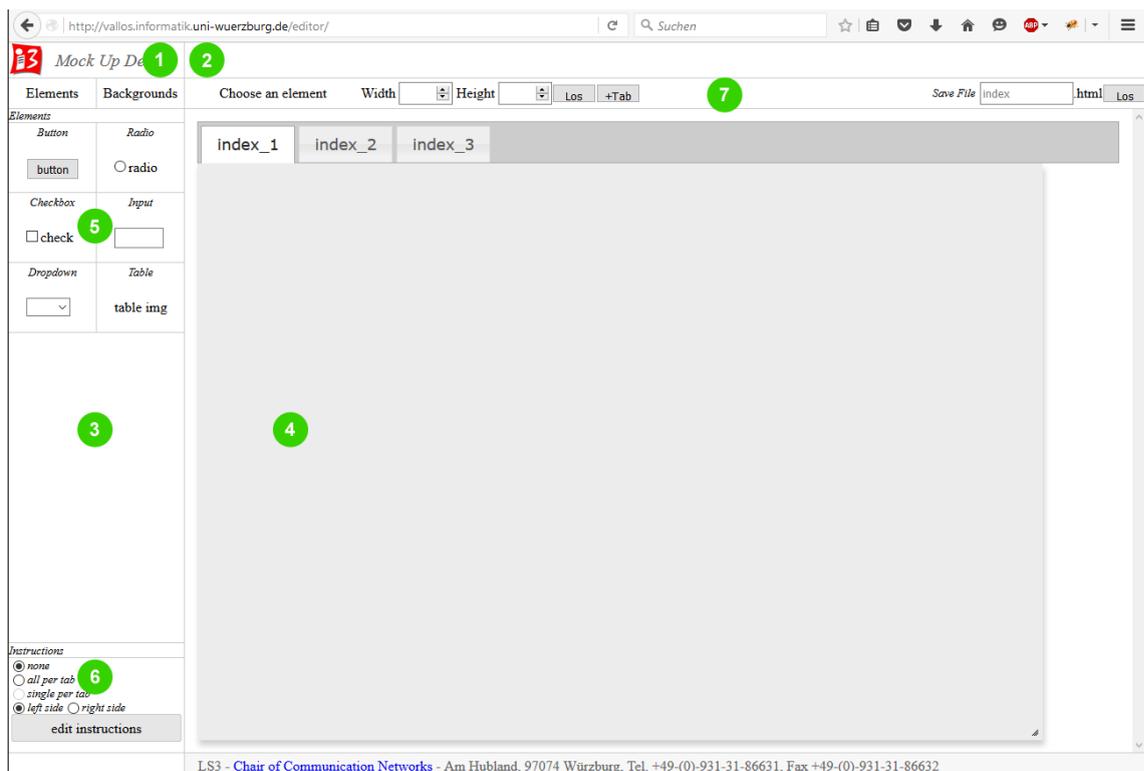


Abbildung 3.3: Überblick über den Editor

So stellt sich der grundlegende statische Aufbau der Webseite wie folgt aus einfachen HTML-Elementen zusammen: In der obersten Leiste befindet sich links ein Feld für ein Logo oder eine Überschrift (1) und rechts eine breite Leiste z.B. für eine Navigation, weitere Werkzeuge o.ä. (2). Weiter teilt sich der Editor in einen Bereich für die funktionalen Elemente auf der linken Seite (3) und einen Arbeitsbereich auf

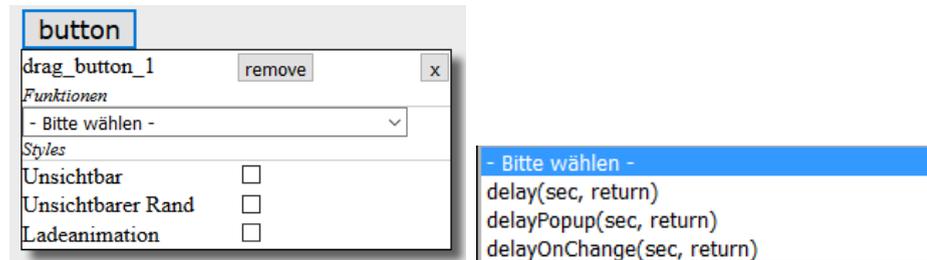
der rechten Seite (4) auf. Als Auswahl für die „Elements“ wurden die grundlegenden Eingabefelder, wie sie z.B. in Formularen zu finden sind, implementiert. Dazu gehören Buttons, Radiobuttons, Checkboxes, Inputfelder und Aufklappmenüs (5). Unter „Backgrounds“ stehen zwei Optionen für Hintergründe zur Verfügung, Bilder und HTML-Seiten. Ein weiteres Feature auf dieser Seite sind die Anweisungen (6). Die Auswahl für die Ausrichtung wird mittels Radiobuttons getroffen und die Anweisungen selbst werden in einem separaten Dialogfenster bearbeitet. Den Hauptbestandteil des Editors bildet die Arbeitsfläche für das Zusammenstellen der Prozesse mit einer darüber liegenden Statusleiste (7), die sowohl informative als auch funktionale Elemente beinhaltet. Dies sind unter anderem ein Debugfeld, welches das zuletzt angeklickte Element im Arbeitsbereich anzeigt und zwei Eingabefelder für Höhe und Breite dieses Elements. Standardmäßig sind drei Reiter für drei Prozessschritte offen. Mit „+Tab“ in der Statusleiste können beliebig viele Reiter hinzugefügt werden. Zuletzt lässt sich die erstellte Emulation mit variablem Dateinamen als HTML Datei speichern. Die Arbeitsfläche kann wie ein normales Fenster am Rand gepackt und in der Größe verändert werden.

Damit den dynamisch hinzugefügten Elementen, z.B. einem Button, Funktionen zugewiesen werden können, öffnet sich beim Klick auf diesen ein Dialogfenster mit einer Reihe von Einstellungsmöglichkeiten. In Abbildung 3.4 werden die Einstellungsmöglichkeiten für die Funktion „delay(sec, return)“ dargestellt.

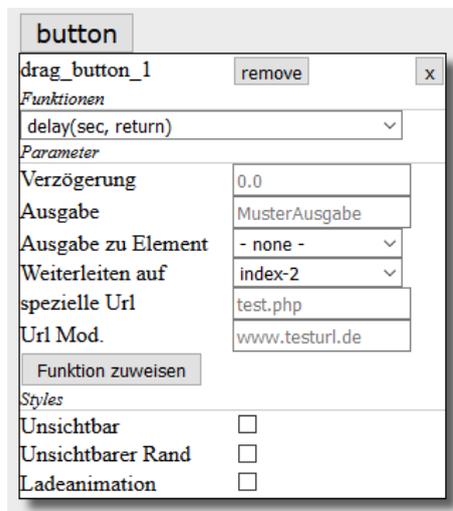
Die erste sichtbare Auswahl besteht aus einem Aufklappmenü mit Funktionen und drei Optionen für Styles (Element unsichtbar, Rand unsichtbar, Ladeanimation). Vorab stehen drei verschiedene Funktionen zur Verfügung, die den Übergang zum nächsten Prozessschritt definieren. Man kann zwischen einem normalen Übergang, einem Übergang zu einem Dialogfenster oder einem Übergang nach einem Eingabetrigger wählen.

Durch Auswahl einer der jeweiligen Funktionen werden weitere Einstellungsmöglichkeiten angezeigt. Der wichtigste Parameter ist hierbei die frei einstellbare Verzögerung. Weiter kann man definieren, welche Ausgabe in welches Element beim Prozessübergang erfolgen soll. Dies ist z.B. für Formular- oder Suchfelder notwendig. Damit keine Probleme mit der Benennung und dem Ansprechen der hinzugefügten Elemente entstehen, werden diese beim „Drophen“ in die Arbeitsfläche automatisch mit eindeutigen Namen im „id“-Attribut versehen. Dadurch kann eine vorgefertigte Auswahl auf die verfügbaren Elemente, welche die Ausgabe erhalten sollen, gegeben und eine falsche Referenzierung verhindert werden. Ähnlich ist dies für das Weiterleiten auf die nächste Seite geregelt. So werden die im Editor dargestellten Reiter als verfügbare Ziele angezeigt. Durch Auswahl von „custom“ wird stattdessen auf den in „spezielle Url“ angegebenen Link weitergeleitet. Es war vorgesehen, dass die Url in der Adressleiste modifiziert werden sollte. Dies ist aus Sicherheitsgründen nicht trivial realisierbar und bisher nur als Platzhalter verbaut.

Neben diesen Einstellungsmöglichkeiten weisen die einzelnen Funktionen zusätzlich noch spezifische Parameter auf. So kann man beim Laden als Dialogfenster einstellen, ob das Fenster an einem gewissen Punkt x,y fest erscheinen soll oder beweglich bleibt. Für die an die Eingabe gebundene Funktion ist noch die Zieleingabe, die das Event schließlich abfeuert, zu spezifizieren. Im nächsten Abschnitt werden



(a) Element: Dialogfenster für Einstellungen (b) Dialogfenster: Auswahl der Funktionen



(c) Dialogfenster: Beispiel für Funktion

Abbildung 3.4: Dialogfenster für Einstellungen im Detail

die Funktionsweise und der Zusammenhang der Elemente erklärt und anschließend an einem Anwendungsbeispiel erläutert.

3.4.2 Erstellung eines emulierten Prozesses

1. Vorlagen einfügen

Stehen die Vorlagen zur Verfügung, können diese im Editor unter „Backgrounds“ hochgeladen werden. Hierfür ist jeweils die zugehörige Auswahl (Bilder/ HTML) zu treffen. Per Doppelklick werden Bild- oder HTML-Dateien im Arbeitsbereich dem aktuell aktiven Tab hinzugefügt. Es ist empfehlenswert zunächst alle Vorlagen einzufügen.

2. Elemente einfügen

Als nächstes sollten alle nötigen Aktions-, Eingabe- und Auswahlfelder eingefügt werden, die innerhalb des Prozesses angesprochen oder verwendet werden. Hier ist darauf zu achten, dass alle mit dem Nutzer in Interaktion stehenden Elemente beachtet werden, sodass der höchstmögliche Detailgrad der originalen Anwendung erreicht wird.

3. Funktionen einfügen

Alle vorher hinzugefügten Elemente werden beim Zuweisen von Funktionen in der Auswahl bereitgestellt. Die Namensgebung wird vom Editor übernommen. Namen können durch klick auf das jeweilige Element in der Statusleiste oder dem Dialogfenster direkt abgelesen werden. Jedem Element, welches eine Aufgabe erfüllen soll, muss die entsprechende Funktion zugewiesen werden.

4. Aussehen anpassen

Die Elemente müssen so angepasst werden, dass sie entweder die Elemente der Vorlage überdecken oder als unsichtbar deklariert werden. Wird ein Feld als unsichtbar deklariert, so wird im Editor nur noch dessen Umriss in Rot dargestellt, sodass dieses gegebenenfalls wiedergefunden werden kann. Eingabefeldern kann auch nur der Rand entfernt werden. Per Klick auf ein Element gilt dieses als ausgewählt, wenn in der Statusleiste über der Arbeitsfläche sein Name angezeigt wird. Weiter kann hier die Größe angepasst werden.

5. Anweisungen einfügen

Sollen während der Emulation Anweisungen für den Nutzer zu sehen sein, können diese unter „Anweisungen bearbeiten“ hinzugefügt werden. Für jeden Reiter steht ein Eingabefeld zur Verfügung. Eine Konfiguration der Anzahl und Position kann über der Schaltfläche vorgenommen werden.

6. Speichern

Als letztes kann die Emulation optional benannt und anschließend gespeichert werden.

Überprüfung der Funktionalität

Die während der Erstellung hochgeladenen Vorlagen müssen zusammen mit der erstellten Emulation und der Funktionsbibliothek JQuery in einem Ordner liegen. Für eine individuelle Ordnerstruktur muss man die Emulation im Code dementsprechend manuell anpassen. Sind diese Voraussetzungen erfüllt kann die Emulation einfach in den Browser gezogen und der Prozess überprüft werden. Eine Übersicht in die Implementierung der generierten Emulationen folgt im nächsten Abschnitt.

3.4.3 Aufbau der generierten Emulationen

Ein Problem bei [37] bestand darin, dass die Ladezeiten der Webseiten zusätzlich zu den eingestellten Verzögerungen hinzukamen. Um diese unkontrollierbare Variable zu umgehen werden alle Emulationsschritte direkt am Anfang geladen. Dadurch muss von den Funktionen, die die nächsten Schritte aufrufen, nur noch die Sichtbarkeit der Emulationsschritte manipuliert werden.

Die während des Editors zugewiesenen Funktionen werden beim Herunterladen der Datei im oberen Skriptbereich mit eingebunden und können so verwendet werden. Dies gilt auch für den zur korrekten Darstellung benötigten CSS Inhalt. Für das Einstellen der Verzögerung braucht anders als bei [37] keine spezielle Bibliothek implementiert werden. Die von der eingestellten Funktion beinhaltete Funktionalität

wird mittels der JavaScript „setTimeout“-Funktion mit der zugewiesenen Verzögerung und den anderen Parametern ausgeführt.

```
setTimeout(function () {
  if (linkTo != -1) {
    // —von aktionen
    $('#' + linkFr + '-instructions').hide();
    $('#' + linkFr).hide();
    // Ausgabe zu Element
    if (returnTo != -1) $('#' + returnTo).val(returnValue);
    // —nach aktionen
    $('#' + linkTo + '-instructions').show();
    $('#' + linkTo).show();
    // falls IFrame-BG im naechsten Schritt, dann scrollen entfernen,
    // sonst wieder hinzufuegen
    ($('#' + linkTo).hasClass('bg-iframe')) ? $('body').addClass('
      overflow-visibility') : $('body').removeClass('overflow-
      visibility');
  } else {
    document.location.href = customUrl;
  }
  if (loadAnim == 1) $('#Test, .tab>input').css("cursor", "default");
}, delayTime * 1000);
```

Die Emulation wird als konforme HTML Datei zum Herunterladen bereitgestellt. Der im Editor zusammengestellte Prozess befindet sich im sogenannten „Test“-Container und hier sind die einzelnen Emulationsschritte jeweils in einer Gruppe aus zwei Containern aneinandergereiht. Lediglich die Container „tabs-1-instructions“ (sofern Anweisungen mit erstellt wurden) und „tabs-1“ (der eigentliche Arbeitsbereich aus dem Editor) sind als sichtbar deklariert. Alle anderen Container sind unsichtbar, werden jedoch direkt am Anfang mit geladen. Nach diesem Prinzip wird der Inhalt aller im Editor zusammengestellten Reiter eingebaut.

Nach dem „Test“-Container folgt noch einmal ein Skriptabschnitt. Sind HTML Inhalte als Vorlagen eingebunden, so wird für das IFrame, welches den Inhalt lädt, hier definiert, dass es nach Abschluss des Ladevorgangs Verweise und Links in verschiedenen Formen aus seinem Inhalt deaktiviert oder entfernt. Dadurch soll sichergestellt werden, dass ein Nutzer nicht versehentlich von der eigentlichen Emulation weggeleitet wird.

3.5 Nachbearbeitung

Im Fall der Screenshots als Vorlagen ist im Allgemeinen keine Nachbearbeitung des Codes nötig. Bei HTML Vorlagen kann es sein, dass noch eine separate Initialisierungszeit zur Emulationsinitialisierung eingebaut werden sollte. Je nach Menge und Komplexität der Skriptinhalte der Vorlagen kann das Aufbauen des IFrames den Browser auslasten. Damit das keinen Einfluss auf die Bewertung hat, sollte der Nutzer beim Beginn der Emulation darüber informiert werden. Die Style Regeln für dieses Dialogfenster sind im CSS Inhalt der Datei enthalten. Der zugehörige Container befindet sich auskommentiert vor dem „Test“-Container. Im unteren Skriptbereich muss noch eine „load()“-Funktion für den ersten Reiter eingefügt werden, die nach

einer eingestellten Pause das Popup verschwinden lässt.

Wie in den hier durchgeführten Studien sind Elemente von PHP nötig, in etwa um Zeitpunkte in der Datenbank festzuhalten. Dafür muss der nötige Code am Anfang der Testdatei eingefügt und diese anschließend als PHP Datei gespeichert werden, da sie als HTML Datei vom Editor kommt.

Um nicht viele verschiedene Dateien für alle Parameterkonfigurationen (z.B. 5 Sekunden Verzögerung, Ladeanimation / 8 Sekunden Verzögerung, keine Ladeanimation usw.) generieren zu müssen, können die durch den Editor fest eingetragenen Parameter in der Funktion durch PHP Sessionvariable ersetzt werden. Dadurch kann man automatisiert verschiedene Konfigurationen erhalten.

```
LoadNextSite(<?php print $_SESSION['test_delay']; ?>, 'scooter', +
             'drag_input_5', 'tabs-4', 'tabs-5', +
             '-1', '-1', '-1', '-1', '-1', +
             '<?php print $_SESSION['test_loader']; ?>')
```

Spezialfälle von Ereignissen oder Abläufen in der Emulation sind ebenfalls im unteren Skriptbereich zu implementieren. Am Beispiel der Amazonemulation mit HTML Vorlagen überlagert ein unsichtbarer Knopf die Suchvorschläge. Damit dieser nicht bereits ohne eine Eingabe in das Suchfeld ausgelöst werden kann, wird über eine Funktion abgefragt, ob das Suchfeld nicht leer ist. Gegebenenfalls kann hier auch nach dem eingegeben Wort verglichen werden, bevor der Knopf auslösbar gemacht wird.

Die entstehenden Emulationen können in eine erweiterte Umgebung eingebaut werden und stellen den Kontext für Untersuchungen dar. Im nächsten Abschnitt wird die Reproduktion eines Anwendungsfalls beschrieben.

3.6 Anwendungsbeispiel: Amazon.com Emulation

In den hier durchgeführten Untersuchungen wurden Emulationen verwendet, die mit dem Editor erstellt wurden. Der Einsatz des Editors in der Praxis wird im Folgenden anhand der Punkte „Vorverarbeitung“ und „Erstellung eines emulierten Prozesses“ im Kontext eines Einkaufs bei Amazon beschrieben.

3.6.1 Vorverarbeitung

Die Emulation wird sowohl auf Basis von Screenshots als auch von HTML Dateien als Vorlagen erstellt. Als erstes wird „Amazon.com“ „inPrivate“ in einem Browser aufgerufen, damit evtl. Cookies und vorherige Suchergebnisse die Seite nicht beeinflussen. Nun wird die Seite mittels der Funktion des jeweiligen Browsers gespeichert (mit der Einstellung „Webseite - komplett“) und der Link aus der Adressleiste kopiert. Dieses Szenario wird für alle weiteren notwendigen Schritte des Einkaufs durchgeführt. Damit sind die Vorlagen für die HTML Version bereits fertig.

Auf der Webseite <http://web-capture.net/> werden die einzelnen Webseiten als Bilder heruntergeladen. Zusätzlich muss manuell ein Screenshot der Suchvorschläge erstellt werden. Um das Ganze zu erleichtern wurde der Screenshot der Startseite kopiert und das Bild der Suchvorschläge an dem Suchfeld positioniert. Damit die

Größe der Suchvorschläge passt, muss man den eigenen Browser auf die Breite der von der Seite generierten Screenshots ziehen oder wie z.B. in Firefox die Entwicklertools verwenden, um die Seite in speziellen Größen anzuzeigen. Der Untergrund der Amazon Webseite wird beim Auslösen der Suche dunkel hinterlegt. Dies lässt sich bei diesem Vorgehen nicht reproduzieren. Nachdem die Vorlagen fertig sind, kann man mit der Erstellung der Emulation fortfahren.

3.6.2 Erstellung der Emulation

Der folgende Abschnitt beschreibt die Erstellung einer auf Screenshots basierenden Emulation.

1. Als erstes werden die fünf Screenshots unter „Backgrounds - Pictures“ hochgeladen. Die Anzahl der Reiter wird mittels „+Tab“ in der Statusleiste über der Arbeitsfläche auf fünf erhöht. Anschließend werden die Vorlagen nacheinander zu den Reitern per Doppelklick in der Auswahl als Hintergrund eingefügt.
2. Nun wird zurück auf „Elements“ gewechselt und die für die Interaktion nötigen Elemente eingefügt. In allen Reitern wird zunächst ein „Inputfeld“ auf die Suchleiste der Amazon Webseite gezogen. Auf „index-2“ wird ein „Button“ auf die Suchvorschläge gelegt. In „index-3“ bis „index-5“ kommt jeweils ein Button auf den „Razor A Kick Scooter“, „Add to Cart“ und „Proceed to checkout (1 item)“.
3. Die hinzugefügten Felder werden nun mit Funktionen belegt. Das Feld „index-1“ erhält eine „onChange“-Funktion und die anderen eine „delay“-Funktion. Nach Eingabe und Auswahl der Parameter (z.B. Delay = 1, Output = scooter, OutputTo = drag input 2, LinkTo = index-2, TargetKeyword = scooter) wird die Funktion mittels der Schaltfläche dem Element zugewiesen.
4. Anschließend sollten die Elemente nun noch so gestaltet werden, dass sie entweder unsichtbar oder integriert erscheinen und die richtige Größe haben. Die Eingabefelder auf dem Suchfeld werden nacheinander zuerst auf die Größe des Originals eingestellt. Dafür wird das jeweilige Feld per Klick ausgewählt und in der Statusleiste über der Arbeitsfläche mit Eingabe in „width“ und „height“ angepasst. Damit kein doppelter Rand sichtbar ist, werden die Elemente in die Mitte des Suchfelds gesetzt und der Rand als unsichtbar deklariert. Nun müssen noch die „Buttons“ angepasst werden. Die Größe ist so zu wählen, dass sie den Raum, der das Ereignis auslösen soll, abdeckt (z.B. das gesamte Suchvorschlagpopup). Danach werden die „Buttons“ auf unsichtbar gesetzt.
5. Da der Nutzer während der Emulation die nächsten Schritte sehen soll, werden noch Anweisungen hinzugefügt (z.B. „index-1“: 1. Type „scooter“ into the search field.).
6. Die Emulation kann als nächstes gespeichert werden. Es ist empfehlenswert, die möglichen Konfigurationen für die Anweisungen (links / rechts / keine und alle / eine) direkt mit zu generieren, falls diese später benötigt werden.

3.6.3 Nachbearbeitung

Zur Nachbearbeitung gehört zum einen die Implementierung von Umfragen, die den Kontext betreffen. Sie werden genutzt um die vom Nutzer wahrgenommene Dienstgüte zu erfassen. Anschließend können sie in den Gesamtablauf der Studie eingebaut werden, indem entsprechend die Weiterleitung angepasst wird.

Nach Fertigstellung wird die Studie, das Vorlagenmaterial und die Funktionsbibliothek JQuery auf einen Server hochgeladen und als Webseite bereit gestellt.

Weitere Details, wie die Speicherung der erfassten Daten und zusätzliche Schritte, wie eine Einführungsseite, werden am Beispiel der durchgeführten Studie in Kapitel 4 genauer beschrieben.

3.7 Erweiterungsmöglichkeiten

Abschließend zum Framework sind die Erweiterungsmöglichkeiten für den Editor zu erwähnen. Für die hier entwickelte Version sind im Grunde nur die elementaren Funktionselemente eingebaut worden. Viele Applikationen benötigen Felder wie Tabellen und verschiedene Auswahlelemente wie Aufklappmenüs, die hinzugefügt werden können.

Neben den Ladeverzögerungen ist es auch möglich eine Eingabeverzögerung, wie sie auch in [37] untersucht wurde, in Funktionen zu setzen und für die Eingabeelemente verfügbar zu machen. Dies würde eine weitere Art von Applikationen in die Reihe der Studien mit aufnehmen, wie z.B. Applikationen für den Fernzugriff.

Ein weiterer Aspekt um den Detailgrad des Nachbaus zu erhöhen sind Tastenevents. Da sich die meisten Applikationen nicht nur über die Maus steuern lassen, sollte ein Suchfeld z.B. per Enter ausgelöst werden können.

Der Editor ist ohne spezielles technisches Grundwissen zu bedienen. Ein zusätzliches Feature könnte eine Umschaltmöglichkeit in einen Expertenmodus sein, in welchem beispielsweise auch eigenes JavaScript integriert werden kann.

4 Methodik und Umsetzung

Um vergleichbare Untersuchungen zu den in Kapitel 2.5 vorgestellten Arbeiten durchführen zu können ist eine vergleichbare Methodik nötig. Die Erfassung der vom Nutzer erfahrene Dienstgüte erfolgt über Umfragen. Die Nutzerbewertungen werden anschließend ausgewertet und der MOS berechnet, sodass Aussagen zu den Ergebnissen getroffen werden können. Abbildung 4.1 zeigt eine Übersicht der hier durchgeführten Schritte. Nachdem die Implementierung des Editors und das Generieren von Emulationen im vorherigen Kapitel erläutert wurden, beschäftigt sich das folgende Kapitel mit der Durchführung der Studien mittels Crowdsourcing. Anschließend werden in Kapitel 5 die Methoden zur Eingrenzung der Datensätze vorgestellt und die Auswertung vorgenommen.



Abbildung 4.1: Ablaufschritte der Masterarbeit von Implementierung des Editors bis zur Auswertung der Daten

4.1 Durchführung der Studie mittels Crowdsourcing

Aus der Implementierung des Editors und den damit erstellten Emulationen als Webanwendungen sind für eine erfolgreiche Durchführung ein paar Grundvoraussetzungen zu erfüllen. Es ist empfehlenswert die neueste Version der gängigen Browser zu verwenden (getestet Internet Explorer 11, Firefox 43 / 44, Chrome 48). Die Browser müssen JavaScript unterstützen und aktiviert haben.

Für eine Online-Verfügbarkeit und den damit hohen Verbreitungsgrad wurde ein Webserver in Form eines Apache2-Linux Servers für das Hosten der verschiedenen Emulationen aufgesetzt. Darauf läuft zusätzlich ein Datenbankserver zur persistenten Speicherung der Daten, was das Exportieren und Weiterverarbeiten erleichtert.

Der grundlegende Ablauf ist bei den hier durchgeführten Studien identisch und gliedert sich in fünf Teilabschnitte, wie in Abbildung 4.2 zu sehen ist. Dies sind in der Reihenfolge eine „Umfrage“, eine sogenannte „Store-Page“, die „Emulation“, eine zweite „Umfrage“ und eine zweite „Store-Page“.

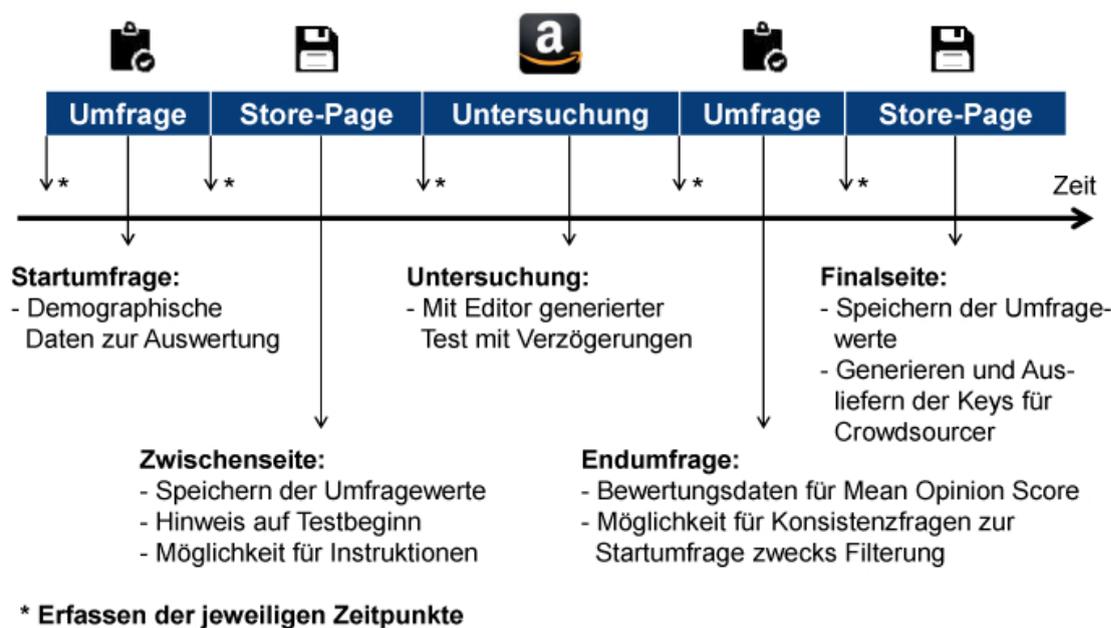


Abbildung 4.2: Ablauf bei der Durchführung der Tests

Über einen Link, der vorher an die Teilnehmer verteilt wurde, werden diese auf eine Einführungsseite geleitet. Sie beinhaltet eine grundlegende Beschreibung der Untersuchung sowie vom weiteren Vorgehen. Ist JavaScript deaktiviert, wird dem Nutzer eine Meldung ausgegeben und der Startknopf bleibt ausgeblendet. Beim Zugriff auf die Einführungsseite werden bereits die Untersuchungsparameter aus dem verfügbaren Set per Zufall ausgewürfelt und in der PHP Sessionvariablen gespeichert. So kann z.B. die Verzögerung, die während der Emulation bestehen soll, an den dafür vorgesehen Stellen über die Session abgerufen werden.

In der ersten Umfrage werden demografische Daten erhoben. Dabei handelt es sich um anonyme Angaben zur Person wie z.B. das Geschlecht und den Altersbereich sowie um andere Kriterien wie Internetzugang und -nutzung. Aufgrund dieser Daten können die Testgruppen detaillierter klassifiziert werden.

Auf den Store-Pages werden aus technischer Sicht die Ergebnisse der Umfragen in der Datenbank gespeichert. Sie können jedoch genutzt werden um den Nutzer durch den Prozess zu leiten oder um zusätzliche Informationen bereitzustellen. In der Store-Page nach der ersten Umfrage können dem Nutzer einführende Anweisungen für die Emulation gegeben werden. Der Umfang der Anweisungen sollte sich an

der Prozesslänge und an den möglicherweise aktivierten Anweisungen während der Emulation orientieren. In unserem Fall wird zu Beginn nur eine kleine Einführung gegeben, da die Anweisungen während der Emulation aktiviert sind.

Die Emulation entspricht dem Nachbau, dessen Erstellung in Kapitel 3.6 mit dem vorgestellten Editor, beschrieben wurde. Die Aufgabe für den Nutzer während der Emulation wird in Abschnitt 4.5.2 erläutert.

Mit dem Abschluss der Emulation wird der Nutzer auf eine Umfrage über die erfahrene Dienstgüte geleitet. Hier sollen hauptsächlich die Verzögerungen, die gegebenenfalls während der Emulation festgestellt wurden, bewertet werden. Dafür stehen wie in Kapitel 2.2 fünf Auswahlmöglichkeiten zur Verfügung und wurden nach ITU-Standard implementiert. Hinzu kommen ein paar Fragen die unter anderem zur Filterung der Daten bei der Auswertung notwendig sind.

Abschließend wird auf die zweite Store-Page geleitet. Hier wird der Teilnehmer über den Abschluss der Umfrage informiert und es können zusätzliche Inhalte dargestellt werden. Dazu gehört z.B. das Generieren und Ausgeben von Verifizierungscodes, wie sie für Teilnehmer der Microworkers Crowdsourcing Plattform nötig sind.

4.2 Untersuchungsparameter

Die möglichen Untersuchungsparameter ergeben sich einerseits aus den Einstellungsmöglichkeiten des Editors und andererseits aus den an den Nutzer weitergegebenen Informationen während der Durchführung eines Tests. Der wichtigste im Editor einstellbare Parameter ist die Verzögerungszeit. Weitere Parameter sind die Ladeanimation, Anweisungen und die Vorlagenart. Tabelle 4.1 zeigt einen Ausschnitt aus den möglichen Zusammenstellungen der Untersuchungsparameter.

Test	Ladeanimation	Anweisungen L	Anweisungen R	Verzögerung (sek)
T1	-	-	-	3
T2	x	-	-	3
T3	x	x	-	3
T4	x	-	x	3
T5	-	x	-	3
T6	-	-	x	3

Tabelle 4.1: Mögliche Zusammenstellung für verschiedene Parameter bei einem Test mit 3 Sekunden Verzögerung.

In der Tabelle werden die Konkatenationen für die Untersuchungsparameter „Ladeanimation“, „Anweisungen Links“, „Anweisungen Rechts“ und „Verzögerung“ dargestellt. Für die Verzögerung von drei Sekunden variieren die restlichen Parameter.

Zusätzliche Verzögerungsdauern und weitere Untersuchungsparameter wie z.B. die Vorlagenart lassen eine Vielzahl an Parameterzusammenstellungen zu.

4.3 Nutzergruppen

Die Realisierung der Emulationen als Webanwendungen ermöglicht ein flexibles Aufsetzen und Verteilen von Untersuchungen. Die leichte Erreichbarkeit über das Internet ermöglicht es mittels Weitergabe eines Links einen Teilnehmer zu erreichen.

Die Untersuchungen wurden für zwei unterschiedliche Nutzergruppen durchgeführt. Die erste besteht aus Freunden und Verwandten und stellt die Referenzgruppe für die Untersuchungen dar. Durch die enge soziale Bindung zu diesen Teilnehmern wird eine hohe Verlässlichkeit beim Bearbeiten der Aufgabe erwartet. Alle Teilnehmer wurden dabei ermutigt, die Umfrage ebenfalls an Freunde und Verwandte weiterzuleiten.

Die zweite Gruppe der Teilnehmer (Crowd) besteht aus den Mitgliedern der internationalen Plattform Microworkers.com, die gegen eine Vergütung bei der Untersuchung mitmachen.

4.4 Filterkriterien

Die Ergebnisse der Crowd, in Form der Teilnehmer von Microworkers.com, sind im Vergleich zur Referenzgruppe genauer zu betrachten. Es kann nicht ausgeschlossen werden, dass die Aufgaben schnell und schlampig bearbeitet wurden. Um den Einfluss dieser Bewertungen auf die Ergebnisse zu minimieren sind verschiedene Ansätze möglich. Basierend auf den „Best Practices“ in [48] wurden einige Mechanismen zur Konsistenzprüfung implementiert. Abbildung 4.3 zeigt die bei der Untersuchung verwendeten Umfragen für die Referenzgruppe.

Die Teilnehmer der Untersuchung mussten im Rahmen der Umfragen eine Konsistenzfrage bezüglich dem Kontinent und Land, in welchen sie leben, beantworten. Die Landesangabe bietet zusätzliche Möglichkeiten zur Nachkontrolle. So lässt sich bei der Microworkersgruppe die Angabe bei der Umfrage mit der Angabe im Microworkers-Profil abgleichen. Die Profile sind anhand der automatisch übermittelten und anschließend gespeicherten IDs der Nutzer erreichbar. Für alle Nutzergruppen lässt sich im Normalfall über die IP-Adresse die Ländereingabe überprüfen, sofern z.B. keine VPN-Zugänge benutzt werden oder der Nutzer die Umfrage z.B. aus dem Urlaub oder einer Geschäftsreise und damit aus einem anderen Land macht, jedoch seine tatsächliche Adresse zur Angabe verwendet. Für die Auswahl bestimmter Antworten war eine spezielle Folgeantwort zu wählen. So sollte basierend auf der Angabe zur Verzögerung mit „nicht wahrnehmbar“, die Auswahl beim Surfverhalten mit „ja“ erfolgen.

Da für jede Untersuchung die Möglichkeit besteht die eingestellten Verzögerungen auszulesen, kann die Antwort auf die Frage, ob eine Verzögerung festgestellt wurde, überprüft werden. Wird z.B. ein Test mit acht Sekunden Verzögerung durchgeführt und es wird angegeben das keine Verzögerung festgestellt wurde, ist die Bewertung

4.4. Filterkriterien

The image shows a screenshot of a web browser with two tabs open: 'Personal Test' and 'Exprience Survey'. The 'Personal Test' tab is active and displays a survey form with the following questions and options:

- Welches Geschlecht haben Sie?
 Männlich Weiblich
- Wählen sie Ihren Altersbereich:
Kategorie
- Auf welchem Kontinent leben Sie?
 Afrika
 Asien
 Australien
 Europa
 Nord Amerika
 Süd Amerika
- Was für ein Internet benutzen Sie?
 Modem
 DSL oder höher
 Kabel
 Satellit
 Mobiles Internet (Smartphone, Laptop, o.Ä.)
- Wie oft surfen Sie im Internet?
 Ein paar mal am Tag
 Einmal am Tag
 Ein paar mal in der Woche
 Einmal in der Woche
 Ein paar mal im Monat
 Einmal im Monat
 weniger oft
 nie
- Besitzen Sie irgendwelche der folgenden Geräte?
 Smartphone Tablet Notebook PC keines davon

A blue 'Weiter' button is located at the bottom of the 'Personal Test' column.

The 'Exprience Survey' tab is also visible and contains the following questions and options:

- Was mussten Sie während der Simulation kaufen?
Produkt..
- Haben Sie während der Simulation Verzögerungen festgestellt?
 Ja Nein
- Wie haben Sie diese Verzögerungen empfunden?
 Sehr störend
 Störend
 Ein wenig störend
 Wahrnehmbar, aber nicht störend
 Nicht wahrnehmbar
(falls sie keine Verzögerungen festgestellt haben, wählen sie bitte "nicht wahrnehmbar")
- Würden Sie auf einer Seite mit solchen Verzögerungen surfen?
 Ja
 Nein
 Nur wenn ich muss
(falls sie keine Verzögerungen festgestellt haben, wählen sie bitte "Ja")
- In welchem Land leben Sie?
Land..
- Optionale Kommentare:

A blue 'Weiter' button is located at the bottom of the 'Exprience Survey' column.

Abbildung 4.3: Erfassung persönlicher Daten und der Bewertung der Applikation anhand von Umfragen bei den durchgeführten Untersuchungen.

inkonsistent. Entweder hat der Nutzer nicht verstanden was er machen soll oder er hat die Umfrage schnell durchgeklickt. Ähnlich ist die Konsistenzfrage welches Produkt während der Emulation gekauft wurde. Wird nicht der Scooter angegeben lässt sich nicht sagen, ob die restliche Umfrage gewissenhaft ausgefüllt wurde.

Ein weiteres Indiz zur Überprüfung sind die Zeiträume für die Umfragen und die Emulation. Die Zeiträume können anhand der festgestellten Zeitpunkte des jeweiligen Starts eines Testschritts berechnet werden. Ist die Dauer für eine Umfrage extrem kurz oder extrem lang kann man davon ausgehen, dass der Test nur durchgeklickt oder nebenher ohne Nachdenken abgearbeitet wurde. Für diese Überprüfung können Referenzzeiten erstellt werden. Dafür lässt man Referenzpersonen unter Aufsicht die Umfragen durchführen und berechnet die Zeiten für die einzelnen Prozessschritte.

4.5 Experimente

Der folgende Abschnitt beschreibt die Studie anhand der einzelnen Experimente. Dabei werden Teilnehmeranzahl, Laufzeit, Parameter und der Kontext der Emulation erläutert. Nach einer Zusammenfassung der Experimente folgt in Kapitel 5 die Auswertung der Daten.

4.5.1 Amazon Funktionstest

Für das geplante Szenario wurde eine Pilotuntersuchung für die eigentliche Kampagne durchgeführt. Sie sollte mögliche Unstimmigkeiten und Fehler darlegen und gegebenenfalls auf notwendige Korrekturen und Strukturanpassungen aufmerksam machen.

So wurde die IP-Ermittlung korrigiert und ein zusätzliches Feld als „benötigt“ deklariert. Bei der hier durchgeführten Untersuchung waren alle Felder außer dem Feedback Pflichtfelder. Zusätzlich wurde noch die Prüfung für JavaScript eingebaut, sodass die Untersuchung anders gar nicht gestartet werden konnte. Zur Ergebnisverifikation wurde noch die Frage nach dem während der Emulation gekauften Artikel aufgenommen. Da der nächste Schritt immer angezeigt wird, wurden die Anweisungen vor der Emulation entfernt.

Funktional und inhaltlich verlief die Untersuchung mit 32 Teilnehmern ohne Probleme. Aufgrund dieser Daten konnte die umfangreichere Kampagne durchgeführt werden.

4.5.2 Amazon Emulation

Nach den oben genannten Anpassungen wurde die Kampagne zwei mal für je 256 Teilnehmer auf Microworkers erstellt und freigegeben. In Abbildung 4.4 ist die Startseite mit Anweisungen zu sehen. Auf der linken Seite im Browser sieht man die Anweisungen für den Nutzer und rechts davon die nachgebaute Amazon Webseite anhand von Screenshots.

Die Tasks wurden mit einer „time-to-finish“ (TTF, Zeit für die gesamte Umfrage) von 5 Minuten angesetzt. Diese kleine Einsatzzeit wird mit 10 Cent pro Task

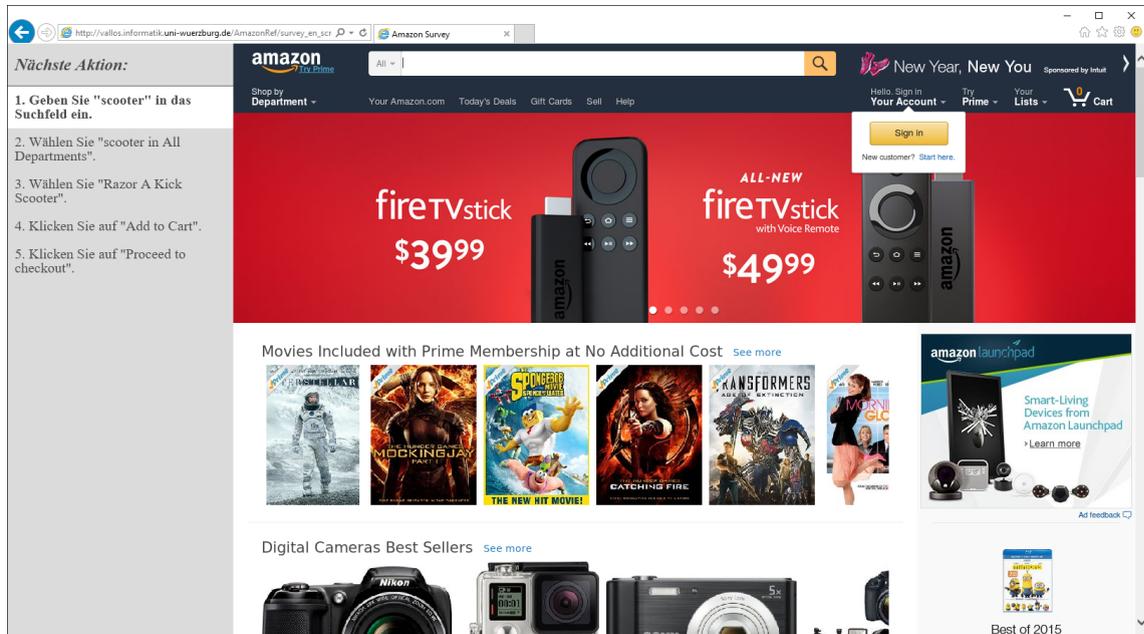


Abbildung 4.4: Momentaufnahme der Startseite der Emulation mit Anweisungen auf der linken Seite für die Teilnehmer der Referenzgruppe.

vergütet. Es wird eine langsame Verteilzeit über die Plattform eingestellt. Die zwei Läufe der Untersuchung mit den Microworkers liefen jeweils über zwei Tage und insgesamt ca. 24 Stunden. So konnten Nutzer aus verschiedenen Ländern erreicht werden. Zusätzlich konnte ein Worker nur einmalig mit seiner ID an der Untersuchung teilnehmen.

Parallel zur Microworkersgruppe wurde die Untersuchung mit der Referenzgruppe durchgeführt und lief über ca. 25 Tage. Die Teilnahme war komplett freiwillig.

Für die Referenzgruppe wurden ein paar Anpassungen vorgenommen. Hierfür wurde eine separate Untersuchung als Webseite mit plattformunabhängigem Zugang und eigenständigen Tabellen in der Datenbank angelegt. Sie wurde in deutscher Sprache gestaltet und der für die Microworkers benötigte Verifikationscode am Ende wurde entfernt. Der Kontext der Emulation war für alle Experimente identisch und wird im Folgenden genauer erläutert.

Ablauf der Emulation

Der Gesamtprozess der Untersuchungen für die Microworkers und die Referenzgruppe, wie in Kapitel 4.1 erläutert, ist grundlegend identisch. Nach einer Einführung über den Zweck der Untersuchung folgen die Fragen zur Person. Anschließend wird der Inhalt der Emulation beschrieben.

Ähnlich zu den E-Commerce Untersuchungen der verwandten Arbeiten müssen die Teilnehmer einen Einkauf tätigen. Der detaillierte Ablauf ist in Abbildung 4.5 zu sehen.

Als E-Commerce-Vorlage wurde Amazon.com gewählt. Im ersten Schritt muss der Nutzer einen „Scooter“ suchen und aus den Suchvorschlägen die Oberkategorie



Abbildung 4.5: Die einzelnen Ablaufschritte der Emulation eines Einkaufs bei Amazon.

wählen. Daraufhin erhält er eine Liste von Scootern und soll den ersten aus der Liste auswählen. Auf der damit aufgerufenen Artikelbeschreibung wird der Scooter in den Warenkorb gelegt und anschließend soll man „auschecken“.

Daraufhin folgt die Umfrage über die QoE und abschließend die Finalseite mit der Auslieferung des Verifikationscodes für Microworkers. Die Teilnehmer der Referenzgruppe erhalten eine Benachrichtigung über den Abschluss der Umfrage.

Untersuchungsparameter

Aus den in Kapitel 4.2 beschriebenen Variationen für Untersuchungsparameter wurden die wichtigsten und die im Zusammenhang als für sinnvoll empfundenen Faktoren ausgewählt und in die Untersuchungsumgebung eingebaut.

1. Screenshot- vs. HTML-Vorlage

Die Emulation des Einkaufs wurde sowohl mit Bildvorlagen als auch mit Webseitenvorlagen der einzelnen Ablaufschritte nachgebaut. Zum einen um die Funktionen des Editors zu zeigen und zum anderen um eine mögliche Beeinflussung des Nutzers durch die interaktive HTML Seite gegenüber der statischen Seite bestehend aus Bildern festzustellen. Beide Varianten sind im Ablauf identisch.

2. Ladeanimation

Wie in den verwandten Arbeiten thematisiert wurde, ist das Feedback für den Nutzer ein wichtiger Faktor bei der Wahrnehmung von Störungen. So wurden die Tests mit allen Verzögerungen sowohl mit als auch ohne Ladeanimation durchgeführt, um zu sehen ob sich die Bewertungen dadurch verändern.

3. Anweisungen, alle Testschritte vs. Aktuellen Testschritt

Die Anweisungen sollen nicht nur den korrekten Ablauf der Emulation gewährleisten. Es wurden zwei Varianten von Anweisungen eingebaut, alle Anweisungen mit Hervorhebung des aktuellen Schritts und eine einzelne Anweisung, die die nächste Aktion beschreibt. Es soll festgestellt werden, ob der Nutzer durch die Anweisungen abgelenkt wird und evtl. besser bewertet.

4. Ladezeiten

Um die Toleranz gegenüber den Verzögerungen feststellen zu können wurde je eine Dauer von 0, 2, 5 und 8 Sekunden ausgewählt. Diese Wahl beruht sowohl auf dem Szenario der Emulation als auch den literarischen Werten und wurde in Konstellation mit den anderen Parametern zu 32 verschiedenen Varianten der Untersuchung gestaltet. Tabelle 4.2 zeigt die finale Zusammenstellung der Untersuchungen anhand der Parameterkonfigurationen.

ID	Vz.	Lade.	Ins.	# Nutzer	ID	Vz.	Lade.	Ins.	# Nutzer
1	0	x	x	8	9	0	x	-	8
2	2	x	x	8	10	2	x	-	8
3	5	x	x	8	11	5	x	-	8
4	8	x	x	8	12	8	x	-	8
5	0	-	x	8	13	0	-	-	8
6	2	-	x	8	14	2	-	-	8
7	5	-	x	8	15	5	-	-	8
8	8	-	x	8	16	8	-	-	8

Tabelle 4.2: Parameterzusammenstellung für die Studie basierend auf einem Einkauf bei Amazon.com mit Screenshots als Vorlagen. Die komplette Palette wurde ebenfalls mit HTML-Vorlagen durchgeführt.

4.5.3 Zusammenfassung der durchgeführten Untersuchungen

Im Rahmen der Arbeit wurden vier Untersuchungen durchgeführt. Ein Funktionstest, eine Untersuchung mit einer Referenzgruppe und zwei Untersuchungen mit den Teilnehmern von Microworkers.com. Tabelle 4.3 zeigt eine Übersicht über die durchgeführten Experimente.

Anhand der Parameter lassen sich grundlegende Unterschiede erkennen. Der Funktionstest hat insgesamt einen kleinen Umfang. Es wurden nur 32 Teilnehmer zugelassen und nur die Verzögerungszeiten von 0, 5 und 8 Sekunden getestet. Insgesamt hatte die Untersuchung 4 verschiedene Varianten. Da hier nur technische Ziele verfolgt wurden, werden die Daten auch nicht ausgewertet.

Zwischen der Untersuchung der Referenzgruppe und der Läufe mit den Microworkers ist die Gesamtlaufzeit ein großer Unterschied. Während die Verteilung zu den sozialen Kontakten viel Zeit in Anspruch nimmt wird im Gegenzug eine hohe Verlässlichkeit der Daten aufgrund der sozialen Bindung erwartet. Von den 128 angepeilten Slots wurden 80 ausgefüllt, während bei den Microworkers beide Läufe mit

Untersuchung	Dauer	# Nutzer	Verzögerungen	# verschiedener Varianten
Funktionstest	< 4h	32/32	2s, 5s, 8s	4
Referenzgruppe	ca. 25T	80/128	0s, 2s, 5s, 8s	32
Microworkers 1. Lauf	ca. 24h	256/256	0s, 2s, 5s, 8s	32
Microworkers 2. Lauf	ca. 24h	256/256	0s, 2s, 5s, 8s	32

Tabelle 4.3: Übersicht über die durchgeführten Experimente anhand der Konfigurationen.

256 Teilnehmern erfolgten. Die Verzögerungszeiten von 0, 2, 5 und 8 Sekunden sowie die Anzahl verschiedener Varianten waren für Microworkers und die Referenzgruppe identisch, damit die erfassten Daten miteinander verglichen werden können.

5 Auswertung

Im folgenden Kapitel werden alle Daten und Ergebnisse der durchgeführten QoE Studien vorgestellt und analysiert. Begonnen wird mit der Charakterisierung der Nutzergruppen. Anschließend werden die Filterkriterien zum Eingrenzen der Datensätze erläutert. Daraufhin wird die aus den Bewertungen gewonnene Aussage vorgestellt und diskutiert.

5.1 Charakterisierung der Nutzergruppen

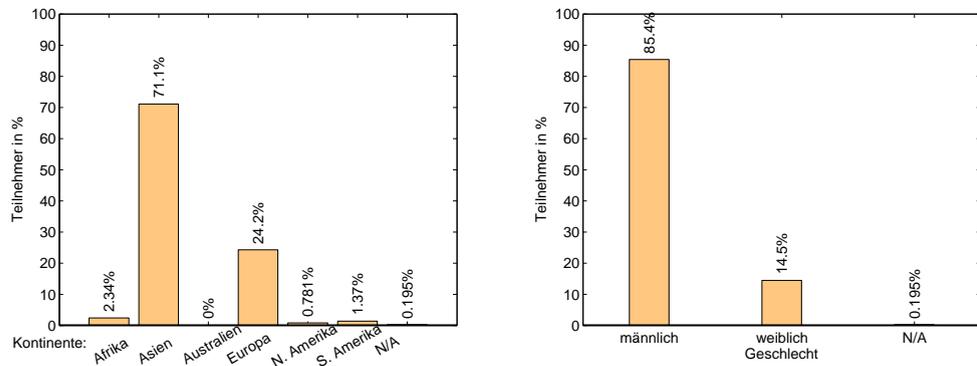
Einige Angaben im Rahmen der Umfragen dienten der Analyse der Teilnehmer. Dabei handelt es sich um anonyme Daten, die keine Rückschlüsse auf einzelne Personen zulassen. Wie in Kapitel 4 erläutert, wurden Untersuchungen für unterschiedliche Nutzergruppen durchgeführt. Das sind zum einen Nutzer der Microworkers Plattform und zum anderen eine Referenzgruppe bestehend aus Familie, Freunden und Bekannten.

5.1.1 Struktur der Teilnehmer von Microworkers.com

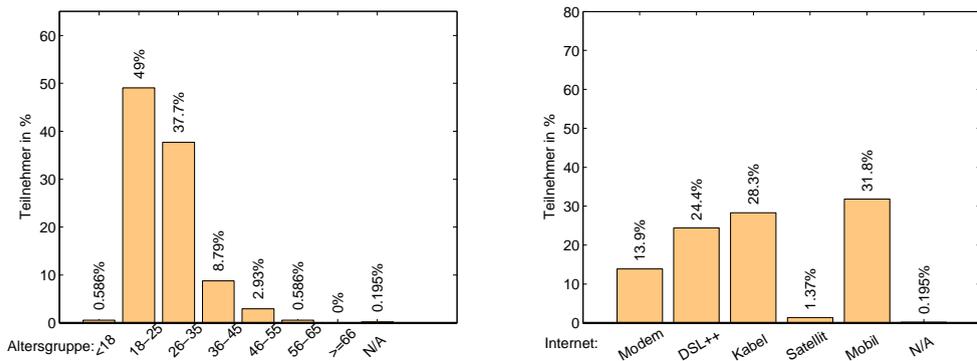
Für den emulierten Einkauf bei Amazon wurden zwei Kampagnen für je 256 Teilnehmer auf der Plattform eingestellt. Sie liefen je ca. 24 Stunden. Insgesamt wurde die Untersuchung während der verfügbaren Zeit 1176 mal gestartet bis alle 512 Aufgaben erledigt waren. Im Weiteren beschränkt sich die Klassifizierung auf die Teilnehmer, welche die Umfragen vollständig abgeschlossen haben. Abbildung 5.1 zeigt die Nutzerstatistiken für die Microworkers.

Diese kommen zu knapp drei Vierteln aus Asien und zu ca. einem Viertel aus Europa. 2.34% gaben an aus Afrika zu kommen und jeweils unter 2% aus Nord und Süd Amerika bzw. machten keine Angabe. Schaubild 5.1a zeigt die Verteilung der Teilnehmer auf die Kontinente. Für die recht eindeutige Verteilung spricht zum einen die geringe Vergütung der Aufgaben, welche für westliche Länder weniger ansprechend wirkt. Zum Startzeitpunkt der Kampagne war es ca. 14 Uhr mitteleuropäische Zeit. Entsprechend waren zu diesem Zeitpunkt vor allem Nutzer der Microworkers Plattform in den westlichen Ländern aktiv. Der technologische Fortschritt ist ebenfalls ein Faktor. So sind z.B. große Teile Afrikas bezüglich eines flächendeckenden Internetzugangs nicht so weit entwickelt, was eine Teilnahme weniger wahrscheinlich macht.

Die Geschlechterverteilung ist in Abbildung 5.1b gegeben. Der Großteil der Nutzergruppe setzt sich mit 85.4% aus männlichen Teilnehmern zusammen. 14.5% waren weibliche Teilnehmer und 1 Teilnehmer gab kein Geschlecht an.



(a) Verteilung der Nutzer auf die Kontinente (b) Verteilung der Nutzer nach Geschlecht



(c) Verteilung der Nutzer auf verschiedene Altersgruppen (d) Verteilung der Internetanbindung der Nutzer

Abbildung 5.1: Charakterisierung der Microworkers-Nutzergruppe im Detail

Abbildung 5.1c zeigt, dass von den 512 Personen 49% im Alter von 18-25 Jahren waren. Mit 37.7% folgen die Teilnehmer im Alter von 26-35. Weiter sind 8.79% der Personen 36-45 und 2.93% 46-55 Jahre alt. 3 Teilnehmer gaben ein Alter von unter 18 Jahren an. Dies ist laut den Nutzungsbedingungen der Plattform allerdings nicht erlaubt. Entsprechend bestehen Zweifel daran, in wie weit diese Teilnehmer Sinn und Ziel der Umfrage verstanden haben. Deshalb werden die Umfrageergebnisse dieser Teilnehmer für die Auswertung nicht in Betracht gezogen. Insgesamt sind 86% der Teilnehmer im Alter von 18-35 Jahren. Sie sollten somit einigermaßen technikaffin sein und ausreichend Kenntnisse im Umgang mit dem Web besitzen.

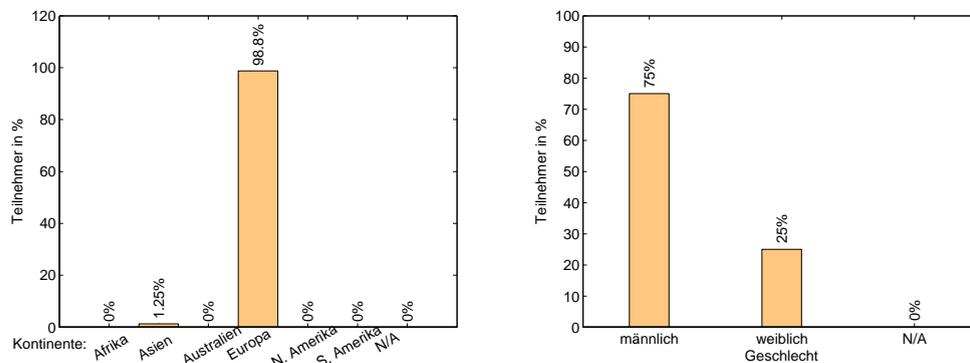
Die Internetanbindung der Nutzer wird in 5.1d abgebildet. Die Mehrheit verteilt sich mit 31.8% auf mobiles Internet und 28.3% auf Kabel. 24.4% der Nutzer gaben DSL als Anschluss an und 13.9% Modem. Satellit ist mit 1.37% am wenigsten vertreten. Das Surfverhalten ist recht eindeutig verteilt. 87.5% der Nutzer gaben an täglich mehrmals online zu sein und 6.25% einmal täglich. Der Rest verteilt sich mit je 0–3% auf die anderen Optionen. Damit sollte gut die Hälfte der Nutzer durch eine schnelle Anbindung in Form von Kabel und DSL recht anfällig gegen Verzögerungen sein. Im Gegensatz dazu steht die große Gruppe mit mobilem Internet. Hier könnten die Bewertungen wesentlich toleranter gegenüber den Verzögerungen ausfallen, da

hier im Normalfall längere Ladezeiten üblich sind.

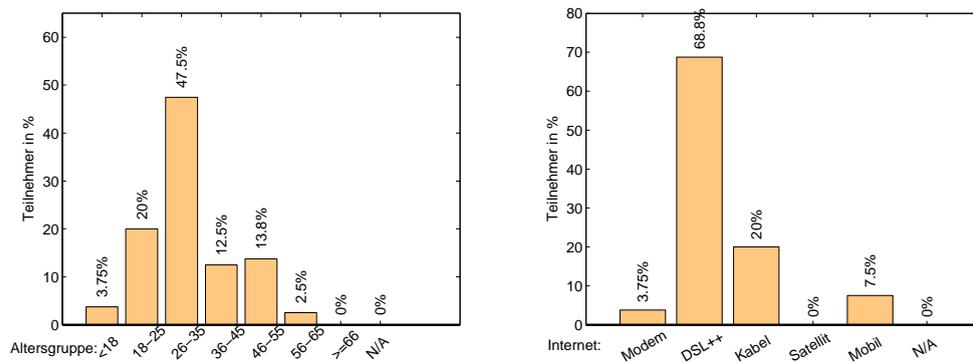
5.1.2 Struktur der Referenzgruppe

Um eine Einschätzung dafür zu bekommen, wie aussagekräftig die Daten der Microworkers sind, wurde eine Referenzgruppe hinzugezogen. Ihre Bewertung soll als Vergleichswert dienen.

Insgesamt nahmen 80 Personen an der Untersuchung teil. Dies waren Verwandte, Freunde und Bekannte. Der Großteil wurde persönlich, per Email oder Messenger angesprochen und ermutigt die Anfrage weiterzuleiten. Zusätzlich kamen noch ein paar Kommilitonen und wissenschaftliche Mitarbeiter aus dem bekannten Umfeld hinzu. Von dieser Nutzergruppe wird erwartet, dass die Aufgabe gewissenhaft erledigt wurde. Abbildung 5.2 zeigt die Statistiken für die Referenzgruppe.



(a) Verteilung der Nutzer auf die Kontinente (b) Verteilung der Nutzer nach Geschlecht



(c) Verteilung der Nutzer auf verschiedene Altersgruppen (d) Verteilung der Internetanbindung der Nutzer

Abbildung 5.2: Charakterisierung der Referenzgruppe im Detail

Dabei kamen 98.75% der Befragten aus Europa und 1.25% aus Asien. Alle europäischen Teilnehmer kamen aus Deutschland und der Teilnehmer aus Asien kam aus Singapur.

47.5% gaben an im Alter von 26-35 Jahren zu sein. Darauf folgen 20% im Alter von 18-25 Jahren sowie 13.8% von 46-55 Jahren und 12.5% von 36-45 Jahren. Unter

18 Jahren zu sein war in der Referenzgruppe kein Ausschlusskriterium. Dies waren 3.75%. Im Alter von 56-65 Jahren waren 2.5%. Die Internetnutzung in Deutschland reicht im Alltag mittlerweile von der Schule bis in den Beruf. Damit sollten die Nutzer bis 45 Jahre im Normalfall sehr technikaffin sein und keine Probleme mit der Webnutzung haben.

Da die große Mehrheit der Nutzer aus Deutschland kommt und das Land zu einem technologisch fortschrittlichen Land zählt, ist die Verteilung der Internetanbindung recht einseitig. Dabei gaben 88.8% der Nutzer an eine sehr schnelle Internetverbindung zu haben, verteilt auf 68.8% DSL und 20% Kabel. 7.5% gehen mit mobilem Internet online und 3.75% mit Modem. Durch die überwiegend schnelle Internetanbindung der Referenzgruppe sollten hier deutliche Unterschiede bei der Bewertung der erfahrenen Verzögerungen entstehen. Es ist eine niedrige Toleranz gegenüber großen Verzögerungen zu erwarten.

87.5% der Personen benutzen das Internet mehrmals täglich. 5% der Teilnehmer sind täglich ein Mal im Netz. 7.5% beschränken sich auf eine wöchentliche Nutzung. Diese Angaben sprechen für einen recht ausgeprägten Internetkonsum. Damit sollten genügend Erfahrungswerte vorhanden sein um verlässliche Daten zur Untersuchung beizutragen.

5.2 Filterung der Daten

Nachdem die Charakteristiken der Nutzergruppen erläutert wurden, geht es im folgenden Abschnitt um die von den Nutzern erhaltenen Daten. Ein Problem bei der Evaluierung ist die Zuverlässigkeit der Ergebnisse. In der Regel kann man davon ausgehen, dass die Ergebnisse der Referenzgruppe verlässlicher sind als die der Microworkers.

Mehrere Faktoren können hierbei einen Einfluss haben. Eine komplexe Struktur der Untersuchung, komplizierter Inhalt oder schwer verständliche Beschreibungen können ein Problem für die Teilnehmer darstellen, sodass evtl. nicht mehr nach tatsächlichen Fakten bewertet wird. Trotz internationaler Sprache können zwischen den Teilnehmern aus verschiedenen Ländern Defizite beim Sprachverstehen vorliegen. Weiter kann die verwendete bzw. verfügbare Technologie Probleme verursachen. Dazu zählen sowohl Soft- als auch Hardware und die Internetanbindung.

Ein zusätzlicher Punkt ist die Vergütung für das Absolvieren einer Aufgabe. Nutzer können dazu neigen Aufgaben hinsichtlich der Maximierung ihrer Vergütung zu beenden. Dabei werden Aufgaben so schnell wie möglich durchgeklickt ohne auf den Kontext zu achten. Bei einer parallelen Abarbeitung mehrerer Aufgaben muss man davon ausgehen, dass die Bewertung unabhängig von den eingestellten Parametern getroffen wird. So können beispielsweise bei einem Wechsel auf einen anderen Tab relevante Informationen zur Bewertung der Emulation nicht wahrgenommen werden. Dies mindert die Qualität der erhobenen Daten und verfälscht die Resultate der Studie.

Um aussagekräftige Ergebnisse zu erhalten müssen verfälschte Resultate entfernt werden. Dazu werden verschiedene Filter auf den kompletten Datensatz angewendet.

5.2.1 Filter

Für die Anwendung der Filter, wie in Kapitel 4.4 beschrieben, werden verschiedene Praktiken in die Umfrage und das Design der Untersuchung eingebaut, um anschließend eine Separierung der Daten vornehmen zu können. Folgende Filter lassen sich in dem Szenario definieren und anwenden.

Überprüfung auf Null-Werte

Bis auf das Feedback waren alle Angaben als Pflichtangaben gefordert. In Ausnahmefällen kann es vorkommen, dass ein Nutzer ohne Angaben durch die Untersuchung kommt. Grund dafür sind z.B. mobile und/ oder ältere Browser, die ein paar Mechanismen nicht unterstützen. In diesem Fall werden „Null“-Werte in die Datenbank geschrieben. Bei der Auswertung können diese Einträge ermittelt und außer Acht gelassen werden. In dieser Studie war dies bei den Läufen mit Microworkers eine sehr geringe Zahl mit 1-3 Teilnehmern. Bei der Referenzgruppe kamen keine Nulleinträge vor.

Überprüfung auf Kontinent-Land Übereinstimmung

Die Nutzer sollten in den Umfragen vor und nach der Emulation jeweils das Land bzw. den Kontinent, in welchem sie leben, angeben. In den Auswertungen werden diese Informationen auf Korrektheit überprüft. Wenn das Kontinent-Land-Paar nicht valide ist, kann dieser Datensatz für die Auswertung ausgeschlossen werden.

Überprüfung von Kontextfragen

Es wurden zwei Kontextfragen in die Umfrage integriert. Zum einen sollten die Teilnehmer am Ende angeben, was sie während der Emulation eingekauft haben. Wird aus den drei Möglichkeiten nicht der Scooter ausgewählt, ist dies ein Indiz dafür, dass der Nutzer sich keine Gedanken über die Umfrage macht.

Ähnlich ist es bei der Frage über die festgestellte Verzögerung. Sofern keine Verzögerung festgestellt wurde, sollte die Frage, ob man auf einer Seite mit solchen Verzögerungen surfen würde, mit „Ja“ beantwortet werden. Dadurch kann man überprüfen ob der Nutzer die Anweisungen genau liest oder die Umfragen und die Emulation nur überflogen hat.

Feststellung von Verzögerungen

Zusätzlich zur Überprüfung des Kontextes aus der Umfrage kann die Angabe über die festgestellte Verzögerung mit den Untersuchungsparametern in Zusammenhang gebracht werden. Bei Verzögerungen von 5 oder 8 Sekunden ist es abstrus, wenn der Nutzer keine Verzögerung bemerkt. Gleiches gilt für Verzögerungen von 0 Sekunden, wenn der Teilnehmer Verzögerungen bemerkt. Also kann die erfahrene Verzögerung mit der Frage ob Verzögerungen festgestellt wurden korreliert werden. Sind diese Zusammenhänge nicht schlüssig, können diese Datensätze entfernt werden.

Diese Art von Falschangaben muss nicht mutwillig geschehen. Nutzer mit Modem als Internetzugang sind z.B. lange Ladezeiten gewohnt. Des Weiteren können

unvorhersehbare Fehler während der Untersuchung auftreten, die entgegen der eingestellten Verzögerung Wartezeiten verursachen. Für eine saubere Auswertung werden Ergebnisse basierend auf diesem Szenario, wie oben genannt, entfernt. Abbildung 5.3 zeigt die Angaben über festgestellte Verzögerungen für die Studien. Die X-Achse zeigt die unterschiedlichen Verzögerungszeiten und die Y-Achse die dazu abgegebenen Stimmen für „Verzögerung festgestellt“ und „keine Verzögerung festgestellt“.

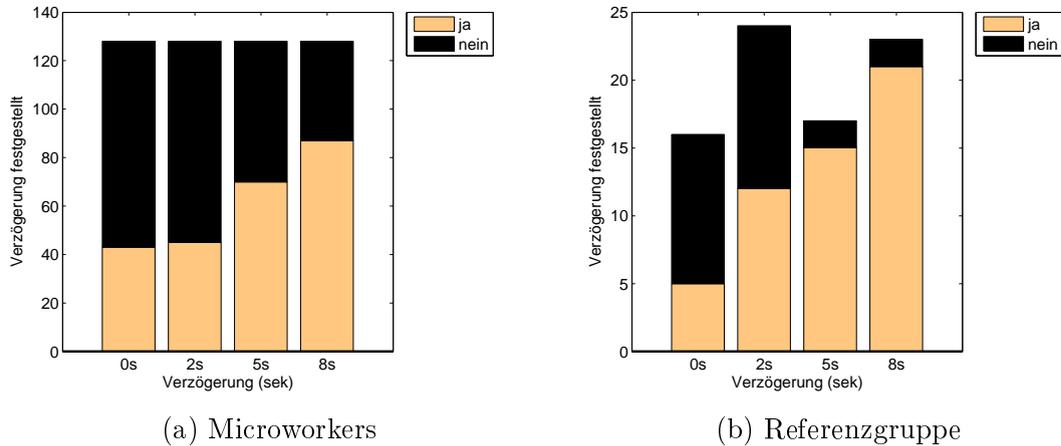


Abbildung 5.3: Feststellung einer Verzögerung im Vergleich von Microworkers und der Referenzgruppe, jeweils für alle getesteten Verzögerungen von 0, 2, 5 und 8 Sekunden.

Bei den Microworkers wurden 128 Umfragen pro Verzögerung ausgefüllt. Bei der Referenzgruppe waren 32 pro Verzögerung angepeilt und je 16 bis 24 Ergebnisse wurden erfasst. Die hellen Balken bei 0 Sekunden Verzögerung und die schwarzen Balken bei 5 und 8 Sekunden zeigen die Datensätze, die gefiltert werden. Prozentual sind das bei 0 Sekunden für beide Gruppen annähernd gleich viele. Für 5 und 8 Sekunden sieht man jedoch deutlich, dass bei den Microworkers mehr Datensätze gefiltert werden, was für eine höhere Verlässlichkeit der Daten der Referenzgruppe spricht.

Verweilzeiten

Die Berechnung der Verweilzeiten erfolgt durch die gespeicherten Startzeitpunkte der jeweiligen Schritte der Umfrage. So lassen sich die Zeiten, die für Umfragen, die Emulation oder für die gesamte Untersuchung benötigt wurden, mit den Referenzzeiten vergleichen. Die Referenzwerte werden erstellt um sinnvolle Bearbeitungszeiten für die einzelnen Schritte angeben zu können. Dazu haben Personen aus privatem Umfeld die Umfrage unter Anweisung durchgeführt. Diese Zeiten wurden um Faktoren und eine Konstante erhöht, um ein Intervall für konforme Werte zu erhalten. Dies ist notwendig, da nicht jeder Nutzer gleich schnell mit den Aufgaben fertig ist. Das Lesen und Verstehen einer Fremdsprache kann, je nach Sprachverständnis des Nutzers, ein Vielfaches an Zeit kosten. Weiter können die Nutzer schneller sein, die bereits mit der Amazon Umgebung vertraut sind.

Internetanbindung

Teilnehmer, die darauf angewiesen sind mit einer langsamen Internetverbindung zu surfen, sind im Normalfall wesentlich längeren Ladezeiten ausgesetzt. Ähnlich wie bei den festgestellten Verzögerungen sind auch diese Angaben, die aus Gewohnheit zu besserer Bewertung führen, zu ignorieren. Deshalb kann nach der Internetanbindung gefiltert werden.

Für die Auswertung gilt es die Filter, im Hinblick auf den abgefragten Kontext und den Komplexitätsgrad der Untersuchung, sinnvoll auszuwählen. Im folgenden Teilabschnitt wird deren Anwendung und Auswirkung auf den Datensatz genauer erläutert.

5.2.2 Anwendung der Filter

Für die Analyse der Daten werden die Filter zu Filterstufen zusammengefasst. Dadurch müssen sie nicht nach jedem Filter einzeln, sondern nach sinnvoll zusammengefassten Kombinationen ausgewertet werden. In Abbildung 5.4 ist eine Übersicht über die angewandten Filterstufen gegeben.

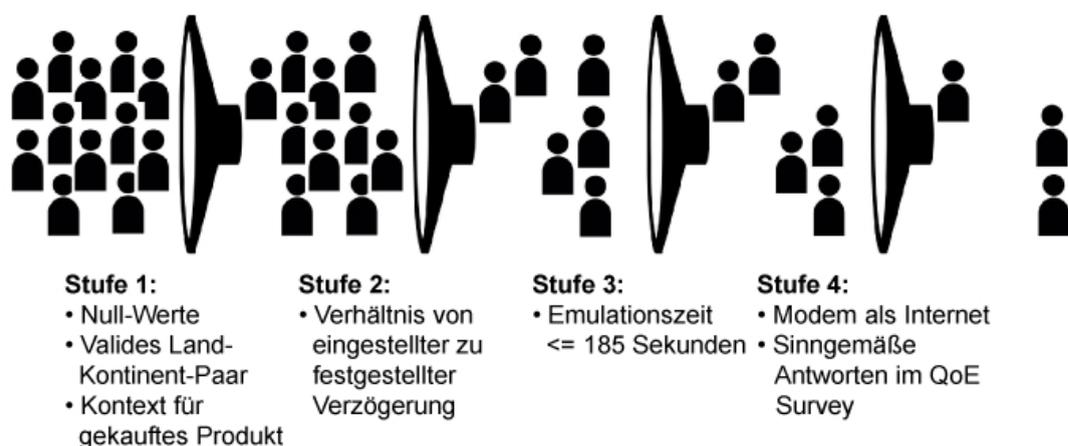


Abbildung 5.4: Übersicht über die Filter zusammengefasst in Stufen. Folgestufen beinhalten immer alle vorgegangenen Filter.

In der ersten Stufe werden drei Filter zusammengefasst. Als erstes lassen wir die Null-Werte außer Acht. Weiter wird die Konsistenzfrage nach dem Kontinent und Land berücksichtigt. Die Datensätze mit nicht validem Paar werden ausgeschlossen. Der dritte Filter entspricht der Frage nach dem eingekauften Produkt. Hier werden die Einträge entfernt, bei denen nicht der Scooter ausgewählt wurde.

Stufe 2 beinhaltet die vorherigen Filter und zusätzlich wird die Korrelation zur festgestellten Verzögerung geprüft. Dabei werden die Datensätze nach den oben beschriebenen Kriterien entfernt.

Durch das Design der Untersuchung können die Verweilzeiten der jeweiligen Schritte einzeln betrachtet werden. Unter Berücksichtigung der internationalen Teilnehmer

und der damit möglicherweise verbundenen Defizite im Sprachverständnis werden die Zeiten für die Umfragen nicht eingeschränkt. Für die Emulation jedoch lassen sich sinnvolle Referenzwerte erstellen. In Abbildung 5.5 werden die Auswirkungen der eingeschränkten Emulationszeit auf die Anzahl der Teilnehmer dargestellt. Die X-Achse zeigt die eingestellten Verzögerungen und die Y-Achse zugehörigen MOS.

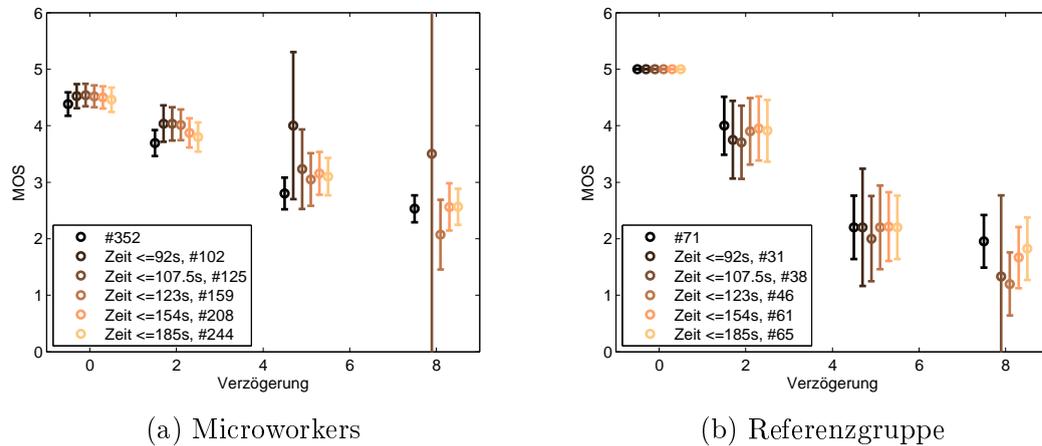


Abbildung 5.5: Vergleich der Abnahme der Nutzerzahlen nach Einschränkung der Zeit für die Emulation mit Darstellung der Auswirkungen auf den MOS.

Nach Anwendung der Filterstufe 2 sind bei den Microworkers noch 352 von 512 und bei der Referenzgruppe 71 von 80 Teilnehmern übrig. Man erkennt die stetige Abnahme der Teilnehmer anhand von kleiner werdenden Zeiten für die Emulation. Werte unter 90 Sekunden sind im Zusammenhang mit den eingestellten Verzögerungszeiten teilweise nicht zu erreichen, wofür die geringe Anzahl an Nutzern spricht. Bei 185 Sekunden bleibt knapp die Hälfte der Nutzer bei den Microworkers übrig und bei der Referenzgruppe werden 3 Personen gefiltert. Ab dieser Grenze ist eine hohe Wahrscheinlichkeit für eine Parallelabfertigung von Aufgaben bzw. eine höhere Unaufmerksamkeit bei der Emulation zu vermuten. Mit 185 Sekunden wurde die Emulationszeit damit als Stufe 3 ausgewählt.

Zu Versuchszwecken wurde eine 4. Stufe angewendet. Dabei wurden die vorherigen Filter eingesetzt und zusätzlich nach der Angabe mit Modem als Internetanbindung gefiltert. Dazu kam auch der Zusammenhang der Kontextfrage, ob eine Verzögerung festgestellt wurde und wie das zukünftige Surfverhalten aussieht. Nur wenn der Eintrag nicht Modem als Internet ausgewählt und im Kontext korrekt geantwortet hat, bleibt er im Datensatz für die Auswertung.

Abbildung 5.6 zeigt die Abnahme der Teilnehmerzahlen nach Anwendung der einzelnen Filterstufen, getrennt für Microworkers und die Referenzgruppe. Dafür werden auf der X-Achse die einzelnen Filterstufen und auf der Y-Achse die übrigbleibenden Teilnehmerzahlen in Prozent dargestellt.

Als erstes fällt der Einfluss der Filterstufe 2 auf die Nutzerzahlen beider Gruppen auf. Im Vergleich zur Stufe 1 werden bei den Microworkers 26.5% und bei der

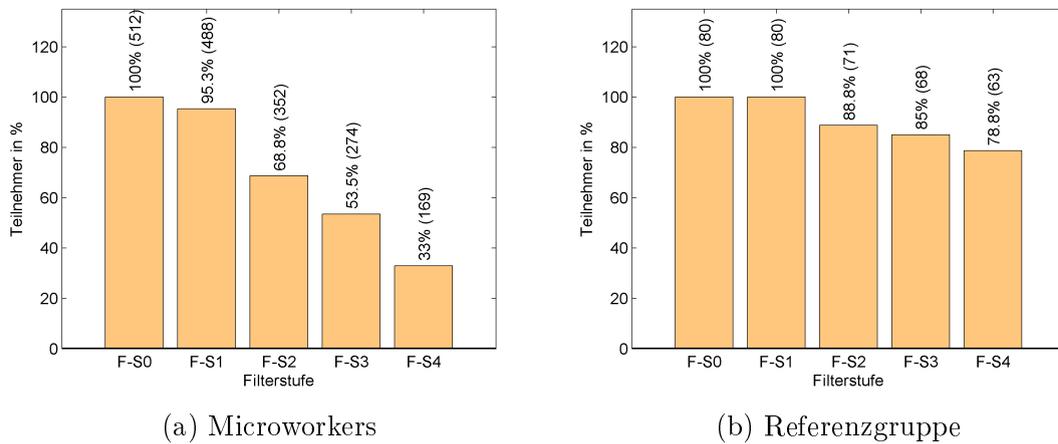


Abbildung 5.6: Vergleich der Anzahl der Teilnehmer

Referenzgruppe 11.2% der Teilnehmer gefiltert. Bei den Microworkers setzt sich die Zahl im Schnitt zu gleichen Teilen aus Nutzern zusammen, die bei 5 und 8 Sekunden keine und bei 0 Sekunden eine Verzögerung festgestellt haben. Man kann davon ausgehen, dass der Großteil davon die Umfrage nicht gewissenhaft durchführt und ein kleiner Teil z.B. sehr tolerant gegenüber 5 Sekunden Verzögerungen ist. Die 9 Personen der Referenzgruppe werden überwiegend auf gewohnten Verhältnissen bewerten. Die folgenden Filterstufen reduzieren den Datensatz bei den Microworkers um jeweils weitere 15 – 20%. Bei der Referenzgruppe erfolgt eine Reduktion um 3 – 7%.

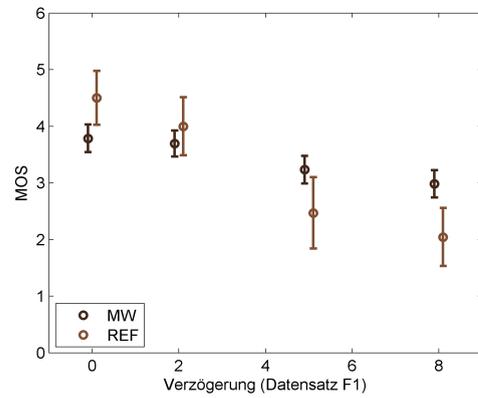
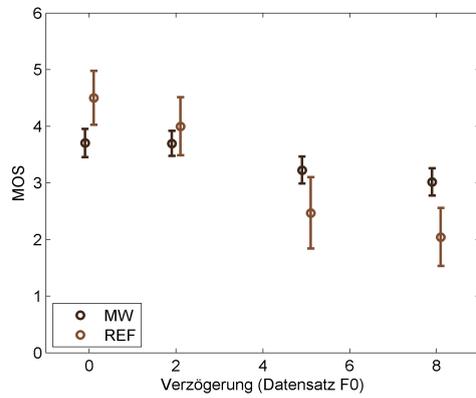
Die Reduktion des Datensatzes pro Filterstufe zeigt, dass in Anbetracht von FS3 46% der Microworkers keine verlässlichen Daten liefert. Im Vergleich dazu sind dies 15% bei der Referenzgruppe, was weiterhin für die gewissenhafte Durchführung der Untersuchung spricht.

5.3 Auswertung

Der folgende Abschnitt beschäftigt sich mit der Auswertung der Daten. Die Abbildungen 5.7, 5.8, 5.10 und 5.12 zeigen jeweils die Ergebnisse nach den einzelnen Filterstufen für die wichtigsten Einflussfaktoren im Vergleich von Microworkers und der Referenzgruppe.

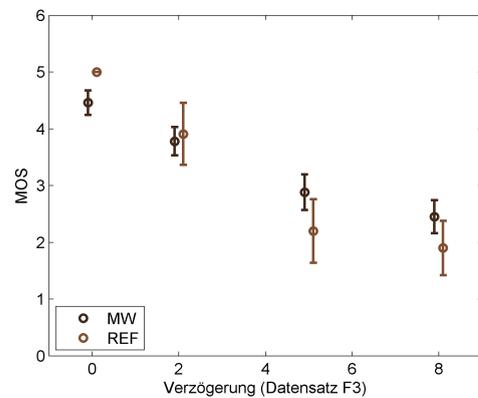
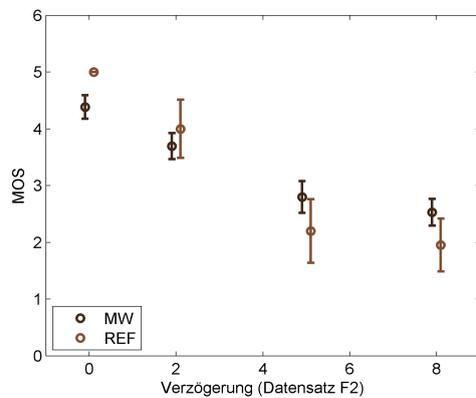
5.3.1 Einfluss der eingestellten Verzögerung auf die QoE

Als erstes wurde die Gesamtbewertung ohne Einteilung in die untersuchten Designparameter (Ladeanimation, Hintergrundart und Anweisungsart) betrachtet. Dabei wurde die Entwicklung des MOS nach der Filterung beobachtet. Die Ergebnisse werden in Abbildung 5.7 zusammengefasst. 5.7a zeigt die Bewertung für die Verzögerungen basierend auf den ungefilterten Daten. Dabei zeigt die X-Achse die jeweilige Verzögerung und die Y-Achse den zugehörigen MOS. Abbildungen 5.7b bis 5.7e zeigen die Entwicklung des MOS nach Anwendung der Filterstufen.



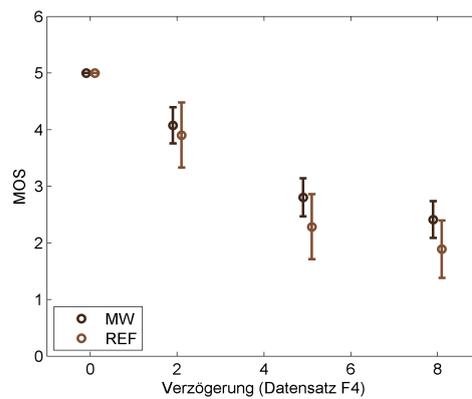
(a) Vergleich der Bewertungen auf den un- gefilterten Daten, Filterstufe 0

(b) Vergleich der Bewertungen mit Filterstu- fe 1



(c) Vergleich der Bewertungen mit Filterstu- fe 2

(d) Vergleich der Bewertungen mit Filterstu- fe 3



(e) Vergleich der Bewertungen mit Filterstu- fe 4

Abbildung 5.7: Vergleich der Bewertungen von Microworkern und Referenzgruppe für die jeweils eingestellten Verzögerungen anhand der angewandten Filterstufen

Von den ungefilterten Daten zur Filterung Stufe 1 ist kaum ein Unterschied fest-

zustellen. Der Datensatz weist nur wenige Null-Werte aufgrund der Pflichtangaben auf. Weiterhin haben die meisten Teilnehmer ein valides Land-Kontinent-Paar angegeben und waren sich bewusst, welches Produkt sie während der Emulation gekauft haben. Im Vergleich zur Referenzgruppe scheiden bei den Microworkers etwas mehr Teilnehmer aus, bei welchen zusätzlich die unter 18-Jährigen gefiltert wurden.

Der MOS liegt bei den Microworkers für alle Verzögerungen im Schnitt zwischen 3 und 4. Bei der Referenzgruppe ist ein Sprung des MOS zwischen 2 und 5 Sekunden zu erkennen. Nach Anwendung des Filters bezüglich der festgestellten Verzögerung (F2) verändert sich der MOS der Microworkers in die Richtung des MOS der Referenzgruppe, wobei die Microworkers dazu neigen die großen Verzögerungen durchschnittlich besser zu bewerten.

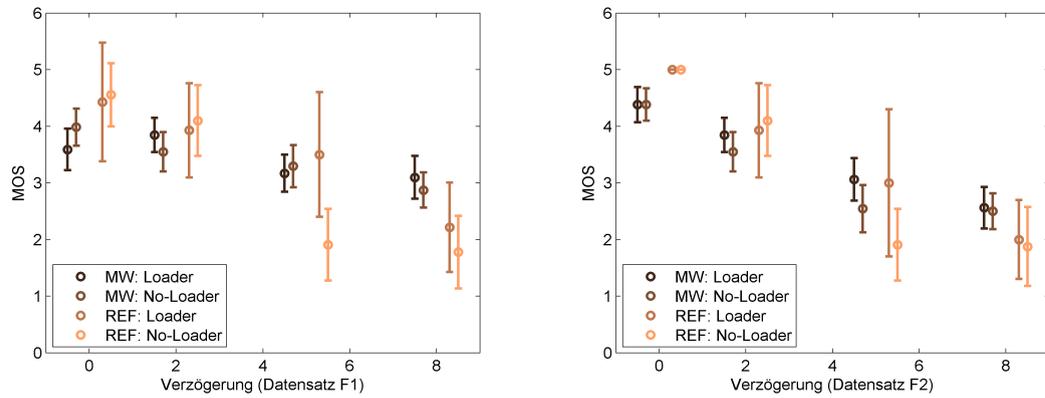
Eine zusätzliche Filterung der Daten nach Emulationszeiten von ≤ 185 Sekunden verändert die Daten nur noch geringfügig. Für die Microworkers lässt sich sagen, dass die Bewertung mit zunehmender Verzögerung schlechter wird. Bei der Referenzgruppe ist ein klares Verhältnis erkennbar. 0 und 2 Sekunden werden sehr gut bewertet und 5 und 8 Sekunden hingegen schlecht. Die Anwendung der Filterstufen lässt den mit steigenden Verzögerungen stetig fallenden MOS nach und nach in einen sprunghaftigen Übergang von 0-2 und 5-8 Sekunden verwandeln.

Filterstufe 4 hat einen starken Einfluss auf die Bewertungen, speziell bei den Microworkers. Bei den Verzögerungen 0 und 2 Sekunden wird die Bewertung aufgrund der damit zusammenhängenden Vorgabe auf die Maximalbewertung gedrückt. Bei der Referenzgruppe ist hier nur eine kleine Korrektur zu bemerken, was für eine höhere Aufmerksamkeit beim Lesen des Kontextes spricht. Für die Microworkers reduziert sich hier der Datensatz auf 169 von 512 und bei der Referenzgruppe auf 63 von 80 Teilnehmern. Dies entspricht noch einer ausreichenden Anzahl an Stichproben. Die Veränderung der Ergebnisse folgt dem Trend der vorherigen Filterstufen und zeigt die Annäherung der Bewertungen der Microworkers zu denen der Referenzgruppe.

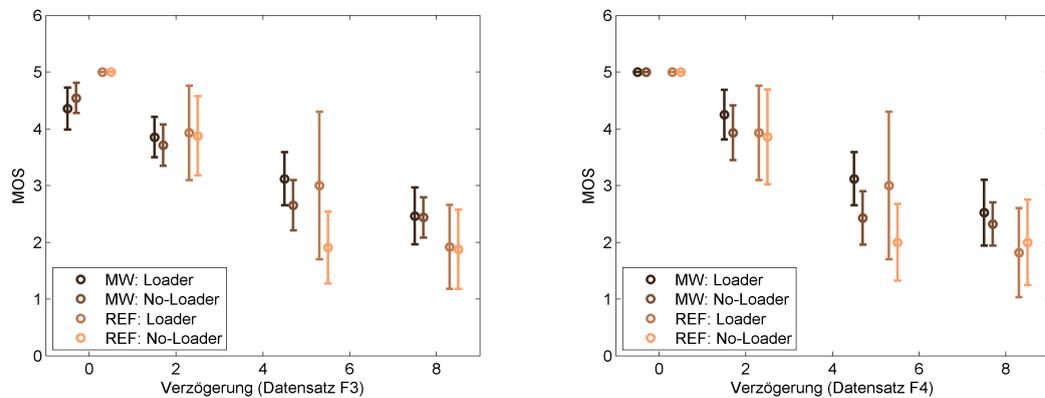
5.3.2 Einfluss der Ladeanimation auf die QoE

Als nächstes wurde der Einfluss der Designaspekte zusätzlich zur eingestellten Verzögerung beobachtet. Abbildung 5.8 zeigt eine Zusammenfassung des Einflusses der Ladeanimation auf den MOS nach den einzelnen Filterstufen. In Abbildung 5.8a wird dies für die Filterstufe 1 dargestellt. Die X-Achse zeigt die Verzögerungen und die Y-Achse den zugehörigen MOS. Es wird jeweils der Wert mit und ohne Ladeverzögerung für die jeweilige Nutzergruppe abgebildet. In den Abbildungen 5.8b bis 5.8d folgen die Auswirkungen der Filterstufen 2 bis 4.

Bei der Trennung der Daten auf Emulation mit und ohne Ladeanimation sieht die Gesamtbewertung sehr ähnlich aus. Es lässt sich kein statistisch signifikanter Unterschied zwischen mit und ohne Ladeanimation sowohl bei den Microworkers als auch der Referenzgruppe feststellen, da sich die Konfidenzintervalle überschneiden. Die Referenzgruppe bewertete im Schnitt bei einer Verzögerung von 5 Sekunden und angezeigter Ladeanimation um 1 Punkt besser in der MOS-Skala als die Teilnehmer ohne angezeigte Ladeanimation.



(a) Vergleich der Bewertungen mit Filterstufe 1 mit und ohne Ladeanimation (b) Vergleich der Bewertungen mit Filterstufe 2 mit und ohne Ladeanimation



(c) Vergleich der Bewertungen mit Filterstufe 3 mit und ohne Ladeanimation (d) Vergleich der Bewertungen mit Filterstufe 4 mit und ohne Ladeanimation

Abbildung 5.8: Vergleich der Bewertungen von Microworkern und Referenzgruppe für die jeweils eingestellten Verzögerungen anhand der angewandten Filterstufen getrennt nach angezeigter Ladeanimation

Abbildung 5.9 zeigt den Gesamtvergleich der Bewertungen der Teilnehmergruppen bezüglich der Ladeanimation. Auf der X-Achse werden die Verzögerungen zusammengefasst und als Gruppe dargestellt. Die Y-Achse zeigt den zugehörigen MOS. Abgebildet sind jeweils die Ergebnisse nach Filterstufe 3 und 4.

Es ist lediglich ein geringer Unterschied in der gemittelten Bewertung der Referenzgruppe zu erkennen. Es lässt sich mit einem Signifikanzniveau von 95% sagen, dass keine statistischen Unterschiede beim MOS der jeweiligen Verzögerung vorliegen. Zu diesem Zweck müssten weitere Durchläufe zur Datenerfassung erfolgen, bis sich die Konfidenzintervalle nicht mehr überschneiden. Nach Anwendung der Filterstufen lässt sich der Trend wie im Gesamtvergleich erkennen.

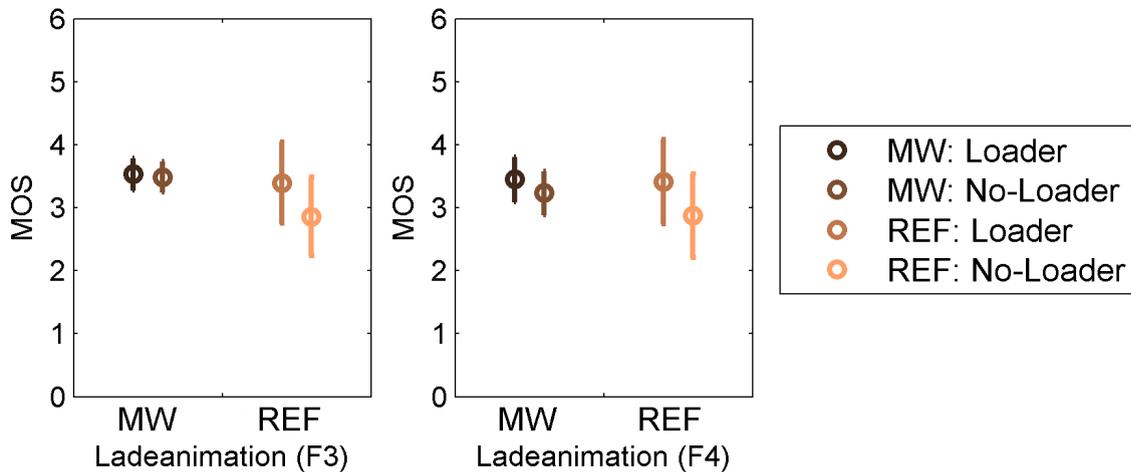


Abbildung 5.9: Vergleich der Daten mit und ohne Ladeanimation zusammengefasst über alle Verzögerungen

5.3.3 Einfluss der Vorlagenart der Emulation auf die QoE

Im Editor lassen sich verschiedene Vorlagenarten bei der Emulationserstellung verwenden. Der Detailgrad einer eingefügten HTML Seite ist dabei höher als der eines statischen Screenshots. In Abbildung 5.10 werden die Bewertungen getrennt nach Vorlagenart nach Anwendung der Filterstufen 1-4 dargestellt. Auf der X-Achse liegen die Verzögerungen und die Y-Achse zeigt den MOS getrennt für die Nutzergruppen und nach der Vorlagenart.

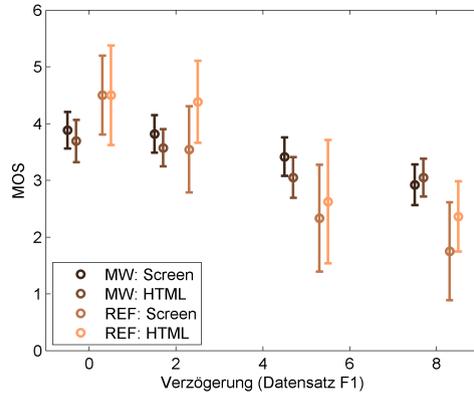
Alle Untersuchungen wurden in gleicher Zusammenstellung jeweils mit Screenshots und HTML Seiten als Vorlagen durchgeführt. Die Ergebnisse zeigen, dass die Microworkers sowohl Screenshots als auch HTML Seiten in diesem Szenario gleich bewerten. Abgesehen von einer geringen Abweichung bei den Mittelwerten der Referenzgruppe können auch hier keine signifikanten Unterschiede belegt werden.

Für dieses Szenario der Untersuchung lässt sich sagen, dass die Vorlagenart keinen Einfluss auf die Bewertung hat. Für eindeutige Ergebnisse müssen mehr Daten erhoben werden um eine klare Trennung der Konfidenzintervalle zu erhalten.

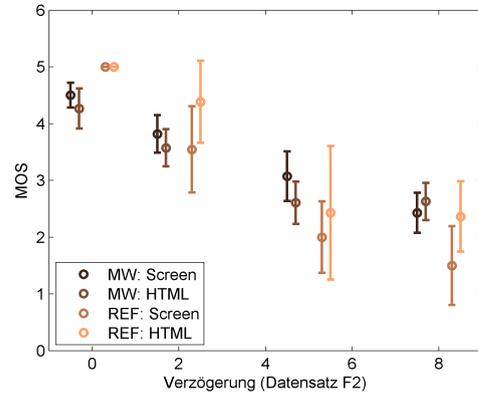
Für das Design kann somit in einem ähnlichen Umfeld auf die schnellere und einfachere Variante mit Screenshots zurückgegriffen werden. Dies sollte speziell zur konventionellen Methode mit aufwändigem und komplexem Nachbau von Prozessen berücksichtigt werden. Die Vorlagenart spielt damit eine größere Rolle für die Ersteller der Emulationen.

Die Ergebnisse für zusammengefasste Verzögerungszeiten werden in Abbildung 5.11 dargestellt und anhand der Nutzergruppen verglichen. Auf der X-Achse sind wieder die Gruppen abgebildet und auf der Y-Achse die zusammengefassten Bewertungen, getrennt nach Vorlagenart.

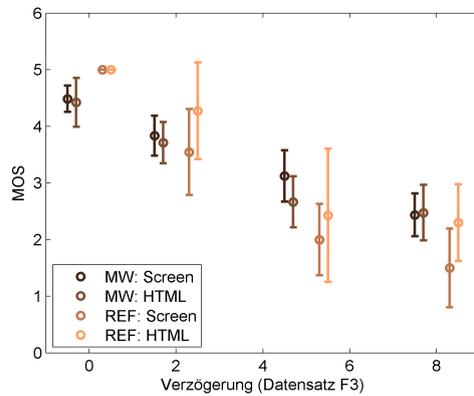
Auch hier ist zu sehen, dass die Art des Hintergrunds mit einem Signifikanzniveau von 95% keinen relevanten Unterschied darstellt. Damit können Emulationen mit der schnellen Variante mit Screenshots als Vorlagen erstellt werden, ohne dadurch



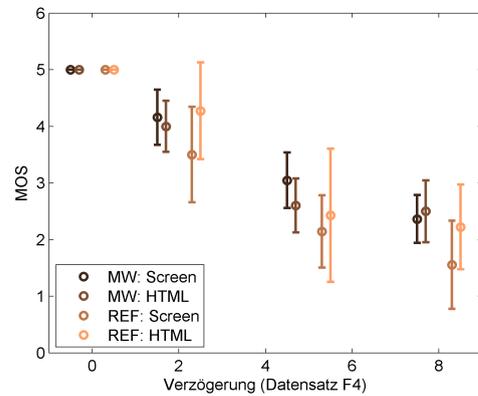
(a) Vergleich der Bewertungen mit Filterstufe 1 getrennt nach Screenshots und HTML



(b) Vergleich der Bewertungen mit Filterstufe 2 getrennt nach Screenshots und HTML



(c) Vergleich der Bewertungen mit Filterstufe 3 getrennt nach Screenshots und HTML



(d) Vergleich der Bewertungen mit Filterstufe 4 getrennt nach Screenshots und HTML

Abbildung 5.10: Vergleich der Bewertungen von Microworkern und Referenzgruppe für die jeweils eingestellten Verzögerungen anhand der angewandten Filterstufen getrennt nach Hintergrundvorlage

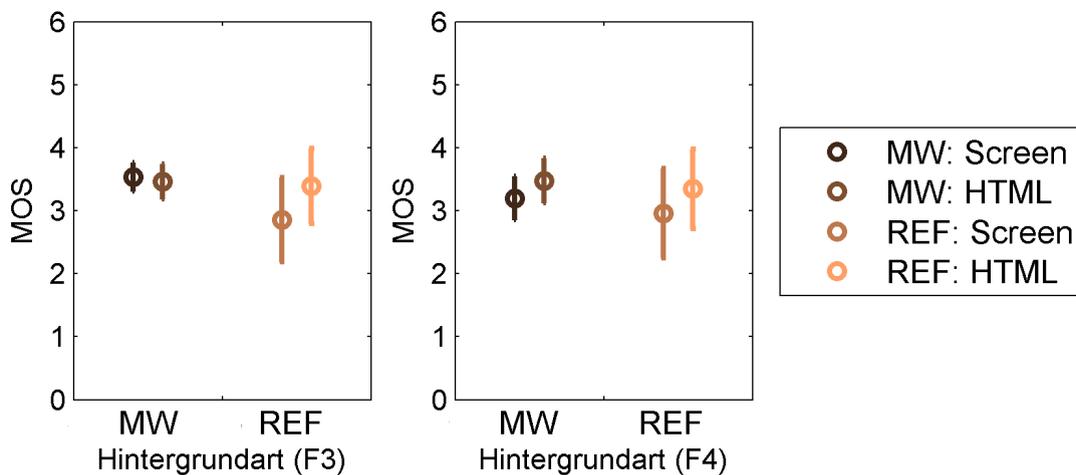


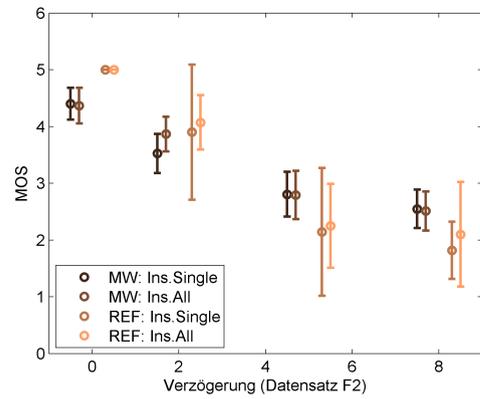
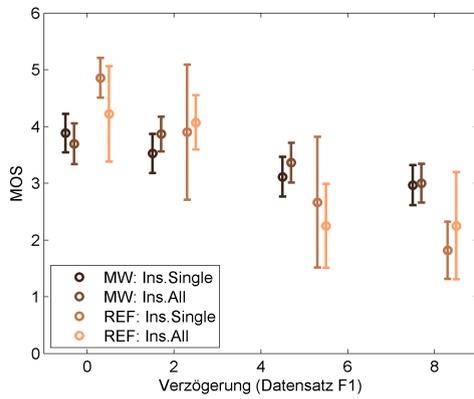
Abbildung 5.11: Vergleich der Daten nach Hintergrundart zusammengefasst über alle Verzögerungen

eine Beeinträchtigung des MOS zu erhalten. Im Fall einer Untersuchung wird der Nutzer im Normalfall nicht auf die Idee kommen das System auszuprobieren und wird sich mehr an die notwendigen Schritte halten. Dadurch erhält er zudem keine Gelegenheit, das darunter liegende Design zu inspizieren.

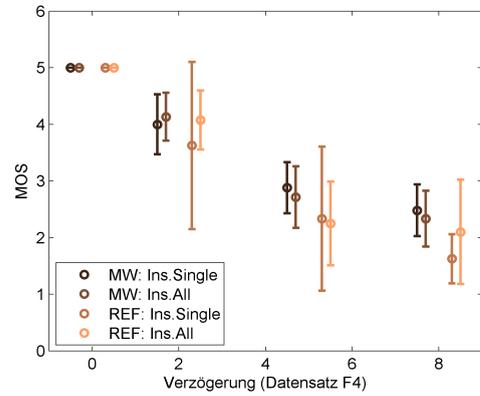
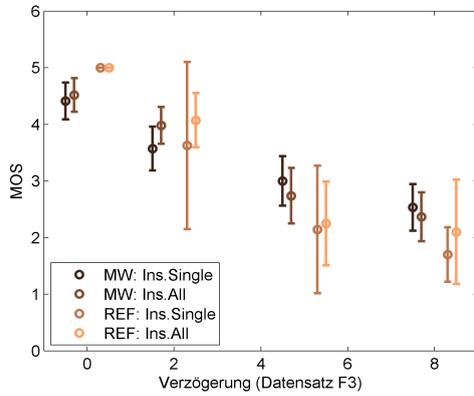
5.3.4 Einfluss der Anweisungsart auf die QoE

Als letzter Designpunkt wurden verschiedene Darstellungsformen von Anweisungen getestet. Ein Teil der Nutzer erhielt dabei immer genau eine Anweisung mit dem nächsten Schritt. Der andere Teil erhielt in jedem Prozessabschnitt immer alle Anweisungen. Anweisungen sollen dem Nutzer als Übersicht und gegebenenfalls als Ablenkung dienen. Die Ergebnisse werden in Abbildung 5.12 dargestellt. Die Y-Achse zeigt den MOS zu den Verzögerungszeiten auf der X-Achse. Die Werte sind getrennt nach Anweisungsart und Nutzergruppe. Jede Abbildung von 5.12a bis 5.12d zeigt die Auswirkung einer Filterstufe.

Die Ergebnisse zeigen, dass keine großen Unterschiede vorhanden sind. Dies hängt zum einen mit dem recht einfachen und kurzen Design der Emulation zusammen. Zum anderen wurde das Amazon Beispiel basierend auf der Einschätzung ausgewählt, dass es den meisten Nutzern bekannt ist. Dadurch konnte zwar ein reibungsloser Untersuchungsverlauf gewährleistet werden, jedoch könnte das dazu führen, dass die Anweisungen aufgrund der vertrauten Umgebung keinen Einfluss haben. Ein weiterer im Zusammenhang stehender Faktor sind die gewählten Verzögerungszeiten. In den Bereichen von 0-2 Sekunden tilgt das Lesen der Anweisung bereits die ganze Dauer der Verzögerung, was sich positiv auf die Bewertung auswirken kann. Im Gegensatz dazu ist der Bereich von 5-8 Sekunden möglicherweise noch nicht groß genug um einen positiven Effekt durch die Anweisungen zu erzielen. Speziell bei der Referenzgruppe sind die Intervalle sehr groß, da die Anzahl der Stichproben im Vergleich zu den Microworkers kleiner ist. Auch hier ist die Entwicklung des MOS in Richtung einer starken Akzeptanz des Systems im Bereich von 0 bis 2 Sekunden



(a) Vergleich der Bewertungen mit Filterstufe 1 getrennt nach Anweisungsart (b) Vergleich der Bewertungen mit Filterstufe 2 getrennt nach Anweisungsart



(c) Vergleich der Bewertungen mit Filterstufe 3 getrennt nach Anweisungsart (d) Vergleich der Bewertungen mit Filterstufe 4 getrennt nach Anweisungsart

Abbildung 5.12: Vergleich der Bewertungen von Microworkern und Referenzgruppe für die jeweils eingestellten Verzögerungen anhand der angewandten Filterstufen getrennt nach der Art der dargestellten Anweisungen

zu erkennen, während die Toleranz ab 5 Sekunden dazu neigt, sprunghaft zu fallen.

Abbildung 5.13 zeigt noch einmal die zusammengefassten Bewertungen für die Anweisungsarten getrennt nach Gruppen.

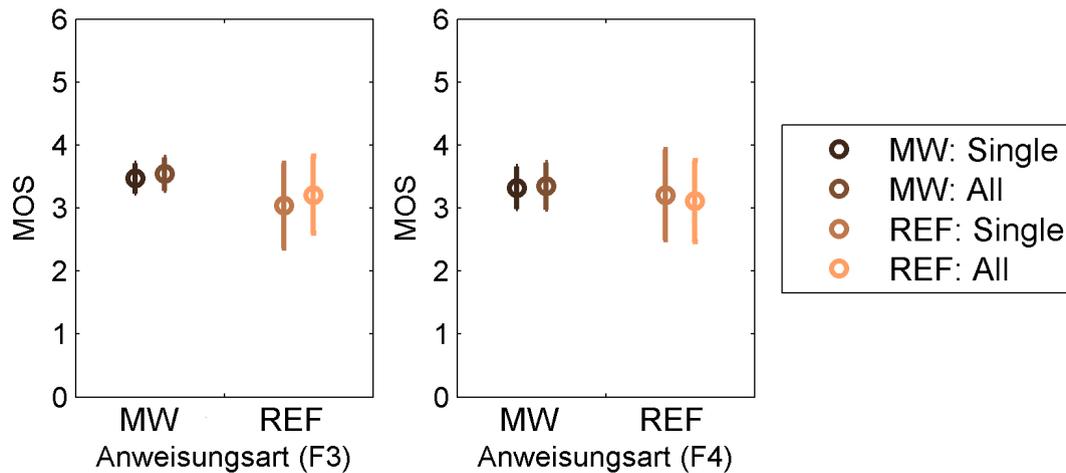


Abbildung 5.13: Vergleich nach Anweisungsart, alle Verzögerungen zusammengefasst

Wie auch bei den vorherigen Designaspekten sind hier mit einem Signifikanzniveau von 95% keine Unterschiede festzustellen. Bei der kurzen Emulation mit den geringen Verzögerungsdauern lassen sich somit durch die Anweisungen keine Effekte erzielen.

5.4 Vergleich der Daten der Nutzergruppen

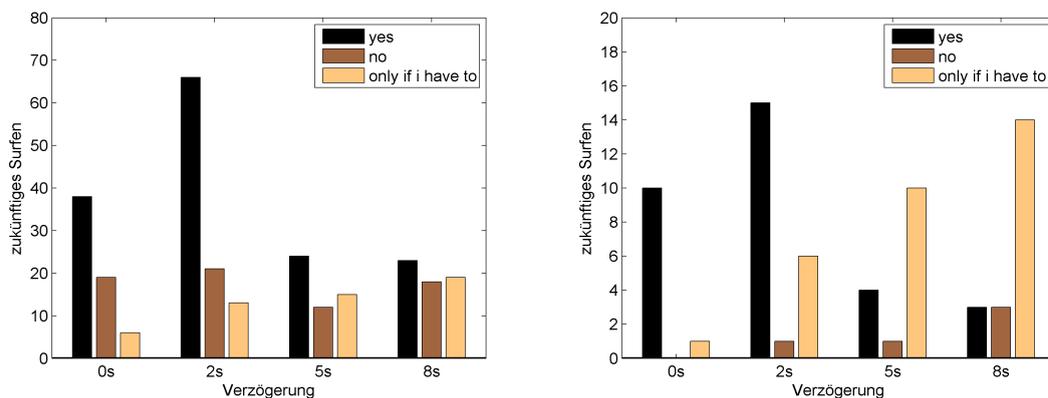
Nach Anwendung der Filterstufen 1-3 tendieren die Daten der Microworkers dazu, denen der Referenzgruppe zu ähneln. Jede Filterstufe macht dabei den Sprung in der Bewertung zwischen 2 und 5 Sekunden deutlicher. Von der Gesamtwertung geht hervor, dass das getestete System mit 0-2 Sekunden Verzögerung einheitlich sehr gut bis gut bewertet wird. Ab 5 Sekunden Verzögerung fällt die Bewertung abrupt schlecht aus. Dies deutet darauf hin, dass sich der Schwellwert für die Toleranz eines Nutzers gegenüber der Wartezeit im Fenster von 2-5 Sekunden befinden muss. In [37] wurden ähnliche Resultate erzielt.

Für die hier getesteten Designaspekte konnten mit einem Signifikanzniveau von 95% keine statistisch signifikanten Unterschiede festgestellt werden. Zum einen hängt dies mit dem Design der Emulation und zum anderen mit den gewählten Verzögerungszeiten zusammen. Am Beispiel der Ladeanimation kann man davon ausgehen, dass die Wartezeiten zu gering sind, um den Nutzer dementsprechend stark zu beeinflussen. Dies müsste z.B. für Ladezeiten von >15 Sekunden überprüft werden. Bei noch größeren Wartezeiten (Minutenbereich) könnte das Feedback einen Abbruch bei der Durchführung einer Untersuchung verhindern, da der Nutzer einfach eine Bestätigung hat, dass das System noch arbeitet. Um statistisch signifikante Unterschiede festzustellen, müssen weitere Läufe für die Untersuchung durchgeführt

werden, bis sich die Konfidenzintervalle für die einzelnen Verzögerungsdauern nicht mehr überschneiden.

5.5 Akzeptanz der Webapplikation und Einzelfeedback

Neben den Angaben, die für die Auswertung wichtig sind, wurden die Nutzer außerdem nach ihrer Haltung für die Wiederkehr auf eine Seite mit den jeweils erfahrenen Verzögerungen gefragt. Die Ergebnisse dieser Angabe in Zusammenhang mit den eingestellten Verzögerungen werden in Abbildung 5.14 dargestellt. Auf der X-Achse werden die untersuchten Verzögerungen abgebildet. Die Anzahl der Antworten für die Optionen „ja“, „nein“ und „nur wenn ich muss“ für die jeweilige Verzögerung wird auf der Y-Achse gezeigt. Abbildung 5.14a zeigt die Antworten der Microworkers und 5.14b die der Referenzgruppe.



(a) Zukünftiges Surfverhalten der Microworkers basierend auf der Verzögerung (b) Zukünftiges Surfverhalten der Referenzgruppe basierend auf der Verzögerung

Abbildung 5.14: Vergleich des zukünftigen Surfverhaltens der Teilnehmer (F3), basierend auf der Frage ob die Seite mit den erfahrenen Verzögerungen wieder besucht wird.

Aus diesen Daten kann die Akzeptanz des Nutzers gegenüber dem System beobachtet werden. Bei den Verzögerungen von 5 und 8 Sekunden antworteten mehr Microworkers mit „ja“ bzw. „nur wenn ich muss“ als mit „nein“. Das heißt, dass die Verzögerungen eher toleriert werden als dass die Seite gemieden wird. Bei der Referenzgruppe ist ein deutlicherer Trend zu sehen. So antworteten die meisten bei 5 und 8 Sekunden Verzögerung mit „nur wenn ich muss“. „nein“ wird erst ab 5 Sekunden in Betracht gezogen. Bei 8 Sekunden steigt zwar die Anzahl der „neins“, ist jedoch gleich zu den Antworten mit „ja“.

Das Ergebnis zeigt, dass die Nutzer dazu neigen auch hohe Verzögerungen eher zu tolerieren als die Seite nicht mehr zu besuchen. Hier spielt die Möglichkeit von Alternativen eine Rolle wie auch die persönliche Haltung der Einzelpersonen.

Einzelfeedback

Für die Auswertung des optionalen Feedbacks wurde jeder Eintrag manuell überprüft. Bei den Microworkers wurden viele Kommentare einfach nur zum Ausfüllen des Feldes abgegeben. Bei der Referenzgruppe enthielten die abgegebenen Kommentare auch konstruktives Feedback.

Unter den Microworkers wurde ein paar mal geäußert, dass geringe Verzögerungen normal seien und so auch tolerierbar sind. Es wurde auch kommentiert, dass größere Verzögerungen normal sind, wenn man mobiles Internet verwendet. Im Sekundenbereich galt die Verzögerung als innerhalb der Schmerzgrenze.

Ein paar Teilnehmer haben sich darüber beschwert, keine Reaktion des Systems festzustellen und haben daraufhin mehrfaches Klicken beschrieben. Dies schien mehr Frustration zu verursachen, als die eingestellte Verzögerung. Andere schreiben, dass die Absicht etwas zu kaufen, die Verzögerung als irrelevant erscheinen lässt. Zwei Personen beschreiben ihre Erfahrung mit Amazon und sagen die Ladezeiten wären auf ihrem System bzw. aufgrund der Fülle an Bildern normal.

Die Referenzgruppe beschreibt zum Hauptteil den Mangel an Feedback, den sie nach dem Klicken erwarten. Viele hatten ein Problem damit, wenn sich die Maus nach dem Klicken nicht verändert. Dabei würden sie direkt anfangen Probleme im Browser o.ä. zu vermuten und - ebenfalls wie die Microworkers - mehrfach zu klicken. Einige ergänzen ihren Kommentar damit, dass eine lange Ladezeit nicht schlimm ist, sofern man die Bestätigung hat, dass sich etwas tut.

Ein Teilnehmer hat festgestellt, dass die Tastatur nicht geht und dass es sich um Bilder handelt. Einen anderen hat gestört, dass sich die Maus nicht anpasst, wenn man über den Button zum Auschecken fährt.

Ein Weiterer bestätigt die Toleranz gegenüber den Wartezeiten, wenn man mit der Einstellung etwas zu kaufen auf die entsprechende Webseite geht. Er beschreibt auch, dass man sich im Notfall eine Alternative suchen kann, sieht dazu jedoch keine Möglichkeit im Fall von Amazon.

6 Fazit und Ausblick

Verzögerungen sind im Alltag überall präsent. Sie stören während der Freizeit, wie z.B. beim Surfen im Internet oder bei der Betrachtung von Videos und auch im Beruf, z.B. bei der Interaktion mit technischen Systemen. Je nach Dauer verärgern sie, unterbrechen Gedankengänge und lenken ab. Damit haben sie einen starken Einfluss auf das Anwendungsverhalten und die vom Nutzer erfahrene Anwendungsqualität, die Quality of Experience.

Ziel dieser Arbeit war es eine schnelle und einfache Methode zur Durchführung von QoE Studien für verschiedene Systeme zu entwickeln und zu evaluieren. Zu diesem Zweck wurde ein Editor zur Reproduktion von beliebigen Web- und Businessapplikationen implementiert. Um diese verallgemeinerte Testmöglichkeit einer großen Gruppe von Nutzern zugänglich zu machen wurde der Editor als open-source Projekt gestaltet. Der Editor ermöglicht es, verschiedene Arten von Anwendungen, wie z.B. SAP Enterprise, als Attrappen mit verminderter Funktionalität, sog. Mock-Ups, nachzubauen. Dazu wird auf Basis von Vorlagen, in Form von Screenshots einer Applikation oder HTML Seiten, die grafische Oberfläche imitiert und als Webapplikation repräsentiert. Sowohl Webseiten als auch Businesssoftware können dadurch als Webapplikationen zu Testzwecken verwendet werden. Dies ermöglicht eine Evaluierung z.B. mittels Tests über Crowdsourcing, was ansonsten aus Kosten- und Zeitgründen für die Erstellung aufwändiger Tests, wie in [37], teilweise nicht in Frage kommen würde. Der Editor ist so gestaltet, dass er ohne viel technisches Hintergrundwissen, für eine große Zahl von Nutzern, anwendbar ist. Die direkte Einbaumöglichkeit der generierten Emulationen in Form von HTML-Dateien ermöglicht eine einfache Integration in Testumgebungen auf einem Webserver oder Crowdsourcing Plattformen. Im Vergleich zur konventionellen Methode mit aufwändigen Nachbauten und großem Ressourceneinsatz wurde mit dieser Methode eine einfache und schnelle Variante zur Erstellung von Tests mit geringem Ressourceneinsatz vorgestellt.

Wir haben die Methode für weit verbreitete Anwendungsfälle für Web-QoE im Bereich von E-Commerce angewandt. Dafür wurde der Einkauf eines Produktes bei Amazon.com mit dem Editor nachgebaut, um die allgemeine Anwendbarkeit vorzuführen. Anschließend wurde der Nachbau in eine Testumgebung eingebunden und online verfügbar gemacht. Mittels Crowdsourcing sowie einer Referenzgruppe wurden Untersuchungen über den Einfluss von Verzögerungen auf die QoE durchgeführt.

Untersucht wurde im Speziellen der Einfluss von Verzögerungen auf die vom Nutzer erfahrene Dienstgüte. Durch den Editor und das Szenario der Untersuchung ließen sich weitere Einflussfaktoren für die QoE bestimmen. Dazu zählen Feedback in Form einer Ladeanimation, verschiedene Darstellungen von Anweisungen und die Vorlagenart der Emulation.

Die Auswertung der Resultate ergab, dass diese Variante zum Testen von Systeme-

men mit kurzen Prozessabläufen im E-Commerce Kontext - im Vergleich zu Tests mit der konventionellen Methode für QoE Bewertungen - keine zusätzlichen Auswirkungen auf den MOS hat. Weiterhin zeigen unsere Auswertungen, dass die Designaspekte, wie Ladeanimation, Anweisungsart und Vorlagenart, für den untersuchten Prozess keine relevante Rolle spielten. Die Auswirkungen auf die QoE hängen dabei im Grunde von der Dauer der Verzögerungen ab. Bei 0 und 2 Sekunden wurde das System stark akzeptiert und durchweg gut bewertet. Ab 5 Sekunden Verzögerung scheint die Toleranz bezüglich der Wartezeit in diesem Kontext ausgereizt zu sein. Hier bewerteten die Nutzer sprunghaft schlechter.

Das zeigt, dass die Akzeptanz und Toleranz im Wesentlichen durch die Erwartungshaltung und Absicht des Nutzers bei der Interaktion mit einem System geprägt ist. Beim Einzelfeedback wurde dies teilweise im Speziellen genannt.

Wie in [37] haben wir eine sprunghafte Lücke des MOS zwischen 2 und 5 Sekunden Verzögerung erhalten. Weitere Untersuchungen könnten einerseits dazu genutzt werden, um die QoE für diese Verzögerungsdauer zu bestimmen. Hier stellt sich die Frage ob der Übergang tatsächlich sprunghaft oder fließend verläuft. Andererseits sollten Tests für Prozesse mit einer größeren Anzahl an Einzelschritten und auch Verzögerungen >10 Sekunden und je nach Kontext bis in den Minutenbereich durchgeführt werden.

Die gleichen Parameter, wie sie in der E-Commerce Studie verwendet wurden, können weiterhin in einem anderen Kontext überprüft werden, um hier einen möglichen Zusammenhang zur Verzögerung und Bewertung festzustellen.

Abbildungsverzeichnis

2.1	Crowdsourcing Granularität der Tasks im Zusammenspiel von Arbeitgeber und Arbeitnehmer nach [19].	8
2.2	Screenshot der Burger-King Feedbacksammlung [23]	9
2.3	Ablauf bei einer Kampagne über die Microworkers-Plattform im Zusammenspiel von Auftraggeber und Auftragnehmer	10
2.4	Unterschiedliche Wahrnehmung von Verzögerungen im Verhaltensmodell eines Nutzers nach [1] beispielsweise durch Manipulation der Erwartungshaltung.	12
2.5	Rolle der Zufriedenheit im Verhaltensmodell der Nutzer nach [28]	12
3.1	Funktionsweise von PHP nach [39]	18
3.2	Übersicht über das Framework	20
3.3	Überblick über den Editor	22
3.4	Dialogfenster für Einstellungen im Detail	24
4.1	Ablaufschritte der Masterarbeit von Implementierung des Editors bis zur Auswertung der Daten	30
4.2	Ablauf bei der Durchführung der Tests	31
4.3	Erfassung persönlicher Daten und der Bewertung der Applikation anhand von Umfragen bei den durchgeführten Untersuchungen.	34
4.4	Momentaufnahme der Startseite der Emulation mit Anweisungen auf der linken Seite für die Teilnehmer der Referenzgruppe.	36
4.5	Die einzelnen Ablaufschritte der Emulation eines Einkaufs bei Amazon.	37
5.1	Charakterisierung der Microworkers-Nutzergruppe im Detail	41
5.2	Charakterisierung der Referenzgruppe im Detail	42
5.3	Feststellung einer Verzögerung im Vergleich von Microworkers und der Referenzgruppe, jeweils für alle getesteten Verzögerungen von 0, 2, 5 und 8 Sekunden.	45
5.4	Übersicht über die Filter zusammengefasst in Stufen. Folgestufen beinhalten immer alle vorgegangenen Filter.	46
5.5	Vergleich der Abnahme der Nutzerzahlen nach Einschränkung der Zeit für die Emulation mit Darstellung der Auswirkungen auf den MOS.	47
5.6	Vergleich der Anzahl der Teilnehmer	48
5.7	Vergleich der Bewertungen von Microworkers und Referenzgruppe für die jeweils eingestellten Verzögerungen anhand der angewandten Filterstufen	49

5.8	Vergleich der Bewertungen von Microworkers und Referenzgruppe für die jeweils eingestellten Verzögerungen anhand der angewandten Filterstufen getrennt nach angezeigter Ladeanimation	51
5.9	Vergleich der Daten mit und ohne Ladeanimation zusammengefasst über alle Verzögerungen	52
5.10	Vergleich der Bewertungen von Microworkers und Referenzgruppe für die jeweils eingestellten Verzögerungen anhand der angewandten Filterstufen getrennt nach Hintergrundvorlage	53
5.11	Vergleich der Daten nach Hintergrundart zusammengefasst über alle Verzögerungen	54
5.12	Vergleich der Bewertungen von Microworkers und Referenzgruppe für die jeweils eingestellten Verzögerungen anhand der angewandten Filterstufen getrennt nach der Art der dargestellten Anweisungen	55
5.13	Vergleich nach Anweisungsart, alle Verzögerungen zusammengefasst .	56
5.14	Vergleich des zukünftigen Surfverhaltens der Teilnehmer (F3), basierend auf der Frage ob die Seite mit den erfahrenen Verzögerungen wieder besucht wird.	57

Tabellenverzeichnis

2.1	Verschiedene Bewertungsskalen für die Erfassung der subjektiven Wahrnehmung	5
4.1	Mögliche Zusammenstellung für verschiedene Parameter bei einem Test mit 3 Sekunden Verzögerung.	32
4.2	Parameterzusammenstellung für die Studie basierend auf einem Einkauf bei Amazon.com mit Screenshots als Vorlagen. Die komplette Palette wurde ebenfalls mit HTML-Vorlagen durchgeführt.	38
4.3	Übersicht über die durchgeführten Experimente anhand der Konfigurationen.	39

Literaturverzeichnis

- [1] S. Egger, T. Hossfeld, R. Schatz, and M. Fiedler, "Waiting times in quality of experience for web based services," in *Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on*. IEEE, 2012, pp. 86–96.
- [2] S. Egger, P. Reichl, T. Hosfeld, and R. Schatz, "'time is bandwidth'? narrowing the gap between subjective time perception and quality of experience," in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1325–1330.
- [3] D. Strohmeier, S. Jumisko-Pyykko, and A. Raake, "Toward task-dependent evaluation of web-QoE: Free exploration vs. "who ate what?," in *IEEE Globecom Workshops*, 2012.
- [4] D. Strohmeier, M. Mikkola, and A. Raake, "The importance of task completion times for modeling web-QoE of consecutive web page requests," 2013.
- [5] I. Arapakis, X. Bai, and B. B. Cambazoglu, "Impact of response latency on user behavior in web search," in *ACM SIGIR Conference on Research & Development in Information Retrieval*, 2014.
- [6] M. S. Jake D. Brutlag, Hilary Hutchinson, "User preference and search engine latency," in *JSM Proceedings, Quality and Productivity Research Section*, 2008.
- [7] S. Stefanov, "Yslow 2.0," in *China Software Developers Network*, 2008.
- [8] E. Schurman and J. Brutlag, "Performance related changes and their user impact," in *Velocity – Web Performance and Operations Conference*, 2009, <http://www.youtube.com/watch?v=bQSE51-gr2s>.
- [9] T. Hosfeld and M. Fiedler, "The unexpected qoe killer: When the network emulator misshapes traffic and qoe," in *Quality of Multimedia Experience (QoMEX), 2015 Seventh International Workshop on*. IEEE, 2015, pp. 1–6.
- [10] "Wikipedia: Quality," visited on 2016-02-04. [Online]. Available: <https://de.wikipedia.org/wiki/Qualit%C3%A4t>
- [11] ITU-T, "E. 800," *Quality of service and dependability vocabulary*, p. 122, 1994.
- [12] "Wikipedia: Quality of Service," visited on 2016-02-04. [Online]. Available: https://en.wikipedia.org/wiki/Quality_of_service
- [13] P. Le Callet, S. Möller, A. Perkis *et al.*, "Qualinet white paper on definitions of quality of experience," *European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003)*, 2012.

- [14] ITU-T, “Bt. 500-13,“,” *Methodology for the subjective assessment of the quality of television pictures*,” Jan, 2012.
- [15] —, “P. 800: Methods for subjective determination of transmission quality,” *International Telecommunication Union, Geneva*, 1996.
- [16] D. Schlosser, A. Binzenhöfer, , and B. Staehle, “Performance Comparison of Windows-based Thin-Client Architectures,” in *Proc. of the Australasian Telecommunication Networks and Applications Conference 2007*, Christchurch, New Zealand, Dec. 2007.
- [17] B. Staehle, A. Binzenhöfer, D. Schlosser, , and B. Boder, “Quantifying the Influence of Network Conditions on the Service Quality Experienced by a Thin Client User,” in *14. GI/ITG Konferenz Messung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen (MMB 2008)*, Dortmund, Germany, Apr. 2008.
- [18] J. Howe, “The rise of crowdsourcing,” *Wired magazine*, vol. 14, no. 6, pp. 1–4, 2006.
- [19] T. Hoffeld, M. Hirth, and P. Tran-Gia, “Aktuelles Schlagwort: Crowdsourcing,” *Informatik Spektrum*, vol. 35, Apr. 2012.
- [20] “Wikipedia: Wikipedia,” visited on 2016-02-10. [Online]. Available: <https://de.wikipedia.org/wiki/Wikipedia>
- [21] “Open Street Map,” visited on 2016-02-10. [Online]. Available: <https://www.openstreetmap.org/#map=17/49.78084/9.97301>
- [22] T. Hoffeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz, “Quantification of youtube qoe via crowdsourcing,” in *Multimedia (ISM), 2011 IEEE International Symposium on.* IEEE, 2011, pp. 494–499.
- [23] “Burger King: Burger King Experience Survey, Feedback Collection,” visited on 2016-02-01. [Online]. Available: <https://www.bk-feedback-de.com/>
- [24] J. Nielsen, “Response times: The 3 important limits,” *Usability Engineering*, 1993.
- [25] R. B. Miller, “Response time in man-computer conversational transactions,” in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I.* ACM, 1968, pp. 267–277.
- [26] J. Nielsen, “Guidelines for multimedia on the web,” *World Wide Web Journal*, vol. 2, no. 1, pp. 157–162, 1997.
- [27] B. Shneiderman, “Response time and display rate in human performance with computers,” *ACM Computing Surveys (CSUR)*, vol. 16, no. 3, pp. 265–285, 1984.

- [28] M. M. Davis and T. E. Vollmann, “A framework for relating waiting time and customer satisfaction in a service operation,” *Journal of Services Marketing*, vol. 4, no. 1, pp. 61–69, 1990.
- [29] F. F.-H. Nah, “A study on tolerable waiting time: how long are web users willing to wait?” *Behaviour & Information Technology*, vol. 23, no. 3, pp. 153–163, 2004.
- [30] B. G. Dellaert and B. E. Kahn, “How tolerable is delay?: Consumers’ evaluations of internet web sites after waiting,” *Journal of interactive marketing*, vol. 13, no. 1, pp. 41–54, 1999.
- [31] G. M. Rose and D. W. Straub, “The effect of download time on consumer attitude toward the e-service retailer,” *E-service Journal*, vol. 1, no. 1, pp. 55–76, 2001.
- [32] M. Cao, Q. Zhang, and J. Seydel, “B2c e-commerce web site quality: an empirical examination,” *Industrial Management & Data Systems*, vol. 105, no. 5, pp. 645–661, 2005.
- [33] B. D. Weinberg, “Don’t keep your internet customers waiting too long at the (virtual) front door,” *Journal of Interactive Marketing*, vol. 14, no. 1, pp. 30–39, 2000.
- [34] N. Bhatti, A. Bouch, and A. Kuchinsky, “Integrating user-perceived quality into web server design,” *Computer Networks*, vol. 33, no. 1, pp. 1–16, 2000.
- [35] R. Schatz, T. Hofffeld, L. Janowski, and S. Egger, “From packets to people: quality of experience as a new measurement challenge,” in *Data traffic monitoring and analysis*. Springer, 2013, pp. 219–263.
- [36] T. Hofffeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, “Initial delay vs. interruptions: between the devil and the deep blue sea,” in *Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on*. IEEE, 2012, pp. 1–6.
- [37] J. Scherer, “Analyse des einflusses von verzögerungen in web- und business applikationen auf die erfahrene dienstgüte,” Deutschland, Sep. 2014.
- [38] “Entwickler: PHP-Marktanteil,” visited on 2016-02-11. [Online]. Available: <https://entwickler.de/online/php/php-marktanteil-auf-811-prozent-geklettert-137114.html>
- [39] “Wikipedia: PHP,” visited on 2016-02-16. [Online]. Available: <https://de.wikipedia.org/wiki/PHP>
- [40] “HTML History,” visited on 2016-02-16. [Online]. Available: <https://www.w3.org/People/Raggett/book4/ch02.html>
- [41] “w3schools: Javascript Intro,” visited on 2016-02-02. [Online]. Available: http://www.w3schools.com/js/js_intro.asp

- [42] “jQuery: What ist JQuery?” visited on 2016-02-02. [Online]. Available: <https://jquery.com/>
- [43] “w3schools: CSS Intro,” visited on 2016-02-14. [Online]. Available: http://www.w3schools.com/css/css_intro.asp
- [44] “etymonline: mockup,” visited on 2016-02-13. [Online]. Available: <http://www.etymonline.com/index.php?term=mockup>
- [45] “merriam-webster: mockup,” visited on 2016-02-13. [Online]. Available: <http://www.merriam-webster.com/dictionary/mock-up>
- [46] “Wikipedia: Mock-Up,” visited on 2016-02-13. [Online]. Available: <https://de.wikipedia.org/wiki/Mock-up>
- [47] “Moqups,” visited on 2016-02-14. [Online]. Available: <https://moqups.com/>
- [48] T. Hoffeld, C. Keimel, M. Hirth, B. Gardlo, J. Habigt, K. Diepold, and P. Tran-Gia, “Best practices for qoe crowdtesting: Qoe assessment with crowdsourcing,” *Multimedia, IEEE Transactions on*, vol. 16, no. 2, pp. 541–558, 2014.