

# HSD

Hochschule Düsseldorf  
University of Applied Sciences



Fachbereich Medien  
Faculty of Media

# **Automatisierte Analyse und Bewertung der Usability von Graphical User Interfaces**

**Untersuchung sowie Konzeption und Implementierung einer prototypischen Umsetzung**

Wissenschaftliche Arbeit zur Erlangung des Grades Master of Science im Studiengang

**M.Sc. Medieninformatik (PO18)**

im Fachbereich Medien der Hochschule Düsseldorf

**Lara Bertram**

**736057**

**Mai 2022**

Betreuer\*innen:

Prof. Dr.-Ing., M.Sc. Markus Dahm

Prof. Dr. Alina Huldtgren

## **Eidesstattliche Erklärung**

Ich, Lara Bertram, Matrikel-Nr. 736057, versichere hiermit, dass ich meine Masterarbeit mit dem Thema

*Automatisierte Analyse und Bewertung der Usability von Graphical User Interfaces*

- *Untersuchung sowie Konzeption und Implementierung einer prototypischen  
Umsetzung*

selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Die Stellen der Arbeit, die anderen Werken dem Wortlaut oder Sinn nach entnommen sind, wurden in jedem einzelnen Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht. Das gleiche gilt auch für beigegebene Skizzen und Darstellungen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

---

Datum, Unterschrift

## **Kontaktinformationen**

Lara Bertram

Elfriede-Bial-Straße 1

lara.bertram@outlook.de

## Zusammenfassung

In dieser Arbeit wird untersucht, wie UI-Gestalter\*innen bei der Erstellung einer grafischen Benutzungsoberfläche durch ein automatisiertes Werkzeug zeit- und kostensparend unterstützt werden können, um die Usability ihres Produkts zu verbessern. Dazu wird eine prototypische Lösung eines solchen Werkzeugs implementiert.

Zu Beginn wird eine Literaturrecherche durchgeführt, in der klassische Methoden und automatisierte Herangehensweisen zur Usability-Evaluation untersucht werden und bereits vorhandene Werkzeuge zur automatisierten Evaluation von UI-Prototypen betrachtet werden. Im Anschluss an diese Literaturrecherche wird der Nutzungskontext des zu entwickelnden Werkzeugs untersucht, indem das optimale Vorgehen bei der Arbeit von UI-Gestalter\*innen im Rahmen einer menschenzentrierten Entwicklung untersucht wird. Auf Basis der daraus gewonnenen Erkenntnisse und einigen Erfahrungsberichten aus dem Bereich der UI-Gestaltung werden Anforderungen für das zu entwickelnde Werkzeug ermittelt.

Auf Basis der zuvor ermittelten Anforderungen wird das Evaluationswerkzeug konzipiert und implementiert. Dabei werden mehrere Herangehensweisen an eine automatische Usability-Evaluation kombiniert. Auf Basis einer Metriken-Sammlung werden visuelle und (einfache) navigatorische Usability-Aspekte untersucht. Ergänzend dazu wird eine Modell-basierte Evaluation genutzt, die auf Basis selbst erstellter Interaktionsfolgen durch Nutzung eines Modells der Mensch-Computer-Interaktion die Dauer einer Interaktion vorhersagt und eine Auswahl von Hinweismustern auf strukturelle Usability-Probleme finden kann.

Die Durchführung einiger Tests zeigt, dass das entwickelte Konzept Gestalter\*innen nach einer Einarbeitung zeit- und kostensparend dabei unterstützen kann, die Usability einer grafischen Benutzungsoberfläche zu verbessern. Fragwürdig bleibt allerdings, ob ein Teil der funktionalen Evaluationen im Rahmen von Prototypen überhaupt nützlich ist. Die tatsächliche Einsetzbarkeit, Akzeptanz und Effektivität des Werkzeugs, sowie der verwendeten Metriken und Hinweismuster sollte durch weitere Untersuchungen evaluiert werden.

## **Abstract**

This thesis investigates how UI designers can be supported in the creation of a graphical user interface by an automated tool in a time- and cost-saving way in order to improve the usability of their product. For this purpose, a prototypical implementation of such a tool is carried out.

To begin with, a literature review is conducted in which classical methods and automated approaches to usability evaluation are examined and existing tools for the automated evaluation of UI prototypes are considered. Following this literature review, the context of use of the tool to be developed is investigated by examining the optimal approach for UI designers working in the context of human-centered development. Based on the findings and some experience reports from the field of UI design, requirements for the tool to be developed are determined.

Based on the previously determined requirements, the evaluation tool is designed and implemented. Several approaches to an automatic usability evaluation are combined. Visual and (simple) navigational usability aspects are examined on the basis of a metrics collection. In addition, a model-based evaluation is used, which predicts the duration of an interaction on the basis of self-generated interaction sequences by using a model of human-computer interaction and can find a selection of hint patterns for structural usability problems.

The execution of some tests has shown that the developed concept can support designers in improving the usability of a graphical user interface in a time- and cost-saving way. However, it remains questionable whether some of the functional evaluations are useful at all in the context of prototypes. The actual usability, acceptance, and effectiveness of the tool, as well as the metrics and hint patterns used, should be evaluated through further investigation.

## **Danksagung**

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Masterarbeit unterstützt haben.

Zuerst gebührt mein Dank Herrn Prof. Dr. Ing., M.Sc. Markus Dahm, der meine Arbeit betreut und begutachtet hat. Für die umfangreiche Betreuung und hilfreichen Anregungen möchte ich mich herzlich bedanken.

Auch möchte ich mich bei meinen Interviewpartnern und den Teilnehmer\*innen meiner praktischen Tests bedanken, die mir mit ihren Erfahrungen hilfreiche Anregungen gegeben haben und mich durch das Testen meiner entwickelten Anwendung zielführend unterstützt haben.

Ein besonderer Dank gilt selbstverständlich meiner Familie und meinem Freund, die mich während meines Studiums unterstützt haben und stets ein offenes Ohr für mich hatten.

## Inhaltsverzeichnis

<b>Abbildungsverzeichnis .....</b>	<b>vii</b>
<b>Abkürzungsverzeichnis.....</b>	<b>viii</b>
<b>1 Einleitung .....</b>	<b>1</b>
1.1 Motivation und Kontext .....	1
1.2 Ziele und Abgrenzung .....	2
1.3 Vorgehensweise und Gliederung .....	3
<b>2 Usability.....</b>	<b>4</b>
2.1 Definition .....	4
2.2 Usability-Prinzipien, -Normen und -Richtlinien .....	5
2.3 Usability-Evaluation.....	7
<b>3 Automatisierte Usability-Evaluation von GUIs.....</b>	<b>9</b>
3.1 Automatisierungsgrad .....	9
3.2 Kategorisierung .....	9
3.3 Automatisierte Evaluation von UI-Prototypen.....	12
<b>4 Anforderungsermittlung.....</b>	<b>14</b>
4.1 Nutzungskontext.....	14
4.2 Nutzer*innen .....	16
4.3 Zusammenfassung .....	19
<b>5 Plattform .....</b>	<b>20</b>
5.1 Auswahl der Plattform .....	20
5.2 Figma.....	20
<b>6 Konzeptionierung.....</b>	<b>24</b>
6.1 Metriken und Evaluationsprofile.....	24
6.2 Modell zur Evaluation von Interaktionsabläufen .....	26
6.3 Formatierung von Interaktionsabläufen .....	27
6.4 Hinweismuster für Usability-Probleme.....	28
6.5 Ablaufmodelle .....	30

6.6	Benutzungsoberfläche .....	35
6.7	Darstellung der Evaluationsergebnisse .....	38
<b>7</b>	<b>Implementierung .....</b>	<b>41</b>
7.1	Implementierungsrahmen.....	41
7.2	Aufbau des Plugins.....	42
7.3	Benutzungsoberflächen .....	43
7.4	Plugin-Code und Evaluation .....	44
<b>8</b>	<b>Praktische Tests.....</b>	<b>46</b>
8.1	Aufgaben und Entwurf .....	46
8.2	Fragebogen .....	48
8.3	Ergebnisse .....	48
<b>9</b>	<b>Fazit .....</b>	<b>53</b>
<b>10</b>	<b>Ausblick.....</b>	<b>55</b>
<b>A.</b>	<b>Anhang .....</b>	<b>63</b>
A.1.	Werkzeuge zur automatisierten Usability-Evaluation.....	63
A.2.	Leitfaden für die Befragungen .....	68
A.3.	Usability-Metriken .....	69
A.4.	Vordefinierte Evaluationsprofile.....	75
A.5.	Aufgabenstellung Test.....	76
A.6.	Fragebogen Test .....	78

## Abbildungsverzeichnis

Abbildung 1: Menschenzentrierte Gestaltung nach DIN EN ISO 9241 .....	14
Abbildung 2: Figma-Editor Benutzungsoberfläche .....	21
Abbildung 3: Figma-Dokumentenstruktur .....	22
Abbildung 4: Stark-Annotation .....	27
Abbildung 5: Ablaufmodell Metriken-Evaluation .....	31
Abbildung 6: Ablaufmodell Aufgabendefinition .....	32
Abbildung 7: Ablaufmodell Szenarien-Definition .....	33
Abbildung 8: Ablaufmodell Aufgaben- bzw. Szenarien-Evaluation .....	34
Abbildung 9: Plugin-Benutzungsoberfläche Metriken-Auswahl .....	35
Abbildung 10: Plugin-Benutzungsoberfläche - Frames-Auswahl .....	36
Abbildung 11: Plugin-Benutzungsoberfläche - Aufgabenliste Bearbeitungsmodus.....	36
Abbildung 12: Plugin-Benutzungsoberfläche - Szenarien-Definition .....	37
Abbildung 13: Plugin-Benutzungsoberfläche - Aufgaben- bzw. Szenarien-Evaluation.....	37
Abbildung 14: Ergebnis Metriken-Evaluation .....	38
Abbildung 15: Ergebnis Aufgaben-Evaluation Vergleich .....	39
Abbildung 16: Ergebnis Aufgaben-Evaluation .....	39
Abbildung 17: Ergebnis Szenarien-Evaluation Vergleich .....	39
Abbildung 18: Ergebnis Szenarien-Evaluation .....	40
Abbildung 19: Code-Struktur Benutzungsoberfläche Metriken-Evaluation.....	43
Abbildung 20: Code-Struktur Benutzungsoberfläche Aufgaben-Evaluation.....	44
Abbildung 21: Entwurf zur Aufgaben-Evaluation .....	46
Abbildung 22: Entwurf zur Metriken-Evaluation .....	47
Abbildung 23: Auswertung Fragebogen Gefühl der Unterstützung .....	51
Abbildung 24: Auswertung Fragebogen Gefühl der Nützlichkeit der Ergebnisse.....	51
Abbildung 25: Auswertung Fragebogen Wiederverwendung und Weiterempfehlung.....	52

## Abkürzungsverzeichnis

(G)UI	(Graphical) User Interface
DIN	Deutsches Institut für Normung
EN	Europäische Norm
ISO	International Organization for Standardization
WCAG	Web Content Accessibility Guidelines
W3C	World Wide Web Consortium
GOMS	Goals, Operators, Method, Selection Rules
KLM	Keystroke Level Model
TLM	Touch Level Model
UX	User Experience
API	Application Programming Interface
NPM	Node Package Manager

# 1 Einleitung

## 1.1 Motivation und Kontext

Für den Erfolg einer (grafischen) Benutzungsoberfläche spielt ihre Usability eine entscheidende Rolle. Um in der heutzutage vorherrschenden Masse an Software-, Web- oder Mobil-Anwendungen hervorstechen, bedarf es einer hohen Usability. Die Nutzer\*innen wünschen sich Produkte, die einfach zu bedienen und leicht zu erlernen sind und ihnen dabei helfen können, ihre Ziele schnell und problemlos zu erreichen. Bietet eine Anwendung diese Möglichkeiten nicht, greifen Nutzer\*innen in der breiten Masse an Alternativen schnell zu anderen Angeboten.

Vielen Entwicklerinnen und Entwicklern digitaler Produkte scheint dieser Zusammenhang bekannt zu sein, da eigene Prozessbeschreibungen zur Entwicklung gebrauchstauglicher Produkte (u.a. (DIN e.V., 2020) und (Sarodnik, et al., 2016 S. 91-103)) entwickelt wurden. In all diesen Prozessbeschreibungen steht der/die Nutzer\*in im Zentrum der Entwicklung und die Usability des Produkts wird im gesamten Verlauf der Entwicklung regelmäßig untersucht und falls nötig Verbesserungen zur Steigerung der Usability vorgenommen. Dafür haben sich verschiedene Evaluationsmethoden herausgestellt. Diese Vorgehensweisen bedürfen meist dem Einbezug der tatsächlichen Nutzer\*innen und der Expertise spezieller Usability-Fachpersonen.

Aufgrund eng getakteter Projektzeiten und fehlender Mittel während der Entwicklung werden Usability-Evaluationen oft nur am Ende der Entwicklung durchgeführt. Werden hier erhebliche Mängel gefunden, können sie meist nur durch einen hohen Aufwand, sowohl zeitlich als auch finanziell und personell, behoben werden. (Sarodnik, et al., 2016 S. 24) Die Entwicklung einer hohen Usability sollte also bereits bei der Anforderungsanalyse des Produkts und der ersten Erstellung von Entwürfen durch Gestalter\*innen erfolgen und hier auch bereits einen großen Stellenwert innehaben. Dadurch können Kosten reduziert werden, die am Ende der Entwicklung entstehen können und das Risiko für nicht behebbare Usability-Probleme reduziert werden. (Sarodnik, et al., 2016 S. 25) Viele Design-Werkzeuge zur frühen Entwicklung von Gestaltungsentwürfen fokussieren sich allerdings stärker auf die reine Visualisierung von Ideen und betrachten Prototypen bislang nicht als sich entwickelndes Produkt im menschenzentrierten Entwicklungsprozess, welches stets den Bedürfnissen der Nutzer\*innen entsprechen sollte. (Hak, et al., 2016) Dadurch ist die Evaluation solcher

Entwürfe mit einem Mehraufwand verbunden, der wiederum dazu führen kann, dass neue Kosten entstehen und die Hürde zur frühzeitigen Evaluation hoch gehalten wird.

Die Automatisierung der Usability-Evaluation bzw. einzelner Aspekte der Evaluation kann hier Abhilfe schaffen und die Effizienz der Evaluation erhöhen. (Khasnis, et al., 2019) Um nicht nur dem Problem hoher Kosten durch Usability-Evaluationen im Allgemeinen entgegenzuwirken, sondern auch eine frühe Evaluation effektiver gestalten zu können, erweist es sich als interessant, frühe Gestaltungsentwürfe automatisiert zu evaluieren.

## 1.2 Ziele und Abgrenzung

Eine automatisierte Evaluation von UI-Prototypen kann dazu führen, dass Kosten reduziert werden und dennoch Produkte mit einer hohen Usability entstehen. Dabei dürfen Gestalter\*innen allerdings nicht in ihrer eigentlichen Arbeit gestört werden, weshalb sich ein Werkzeug zur automatisierten Evaluation möglichst störungsfrei einbinden lassen sollte und im Idealfall leicht zu erlernen und effektiv zu nutzen sein sollte.

Diese Arbeit dient in erster Linie dazu, herauszufinden, wie Gestalter\*innen bei der Entwicklung von Gestaltungslösungen durch ein automatisiertes Werkzeug im Hinblick auf die Gewährleistung bzw. Verbesserung der Usability zeit- und kostensparend unterstützt werden können. Ein solches Werkzeug richtet sich vor allem an professionell tätige UI-Gestalter\*innen, weshalb zur Beantwortung der zuvor beschriebenen Fragestellung ein Verständnis ihrer Arbeitsweise geschaffen werden muss, aus dem Anforderungen an ein solches Werkzeug ermittelt werden können.

Prototypen beschreiben frühe Entwürfe geplanter Umsetzungen und umfassen im Rahmen dieser Arbeit Gestaltungsentwürfe von Webanwendungen mit grundlegenden funktionalen Aspekten zur Beschreibung einiger Interaktionen zur Nutzung des Produkts. Sie werden sowohl dazu genutzt visuelle Vorstellungen einer Anwendung zu schaffen als auch einzelne Funktionalitäten und Abläufe zu erproben (Richter, et al., 2016 S. 72f.). Ein Werkzeug zur automatisierten Usability-Evaluation von Prototypen sollte idealerweise verschiedene Aspekte untersuchen können, um umfangreiche Evaluationen unterstützen zu können.

Um eine tatsächliche Zeitersparnis und Kostenreduktion zu gewährleisten, soll das in dieser Arbeit entwickelte Konzept und Werkzeug keine Beteiligung durch Nutzer\*innen involvieren. An dieser Stelle sei aber angemerkt, dass dadurch keine Evaluation mit tatsächlichen

Nutzer\*innen ersetzt werden soll. Im Rahmen eines Usability Engineerings, der Entwicklung digitaler gebrauchstauglicher Produkte, besitzt der aktive und passive Einbezug der Nutzer\*innen trotzdem eine gewisse Relevanz (DIN e.V., 2020 S. 13). Das Werkzeug soll eine Ergänzung dazu darstellen und zur Untersuchung und Erprobung einiger Möglichkeiten zur Einbeziehung verschiedener Usability-Aspekte in den Arbeitsalltag der Gestalter\*innen genutzt werden.

### **1.3 Vorgehensweise und Gliederung**

Zur Untersuchung der Möglichkeiten einer angemessenen und zielführenden Integration einer automatisierten Usability-Evaluation in die Arbeit von UI-Gestalterinnen und -Gestaltern wird im Zuge dieser Arbeit ein prototypisches Werkzeug entwickelt und evaluiert. Die Arbeit gliedert sich dazu in vier Hauptbereiche.

Zu Beginn wird eine Einführung in die Thematik der Usability gegeben und auf Basis einer Recherche sowohl klassische Methoden der Usability-Evaluation als auch verschiedene Herangehensweisen automatisierter Usability-Evaluationen digitaler Produkte und automatisierter Evaluationen von Prototypen untersucht.

Im zweiten Abschnitt werden einige Anforderungen zur Entwicklung des Werkzeugs ermittelt, indem die Arbeit von UI-Gestalter\*innen, die Eigenschaften von Prototypen und das Vorgehen bei der Entwicklung dieser untersucht wird. Im Anschluss daran wird die zur Entwicklung verwendete Plattform beschrieben und das Konzept für das Werkzeug dargestellt, sowie einige ausgewählte Aspekte der Implementierung erläutert.

Der dritte Abschnitt thematisiert die Durchführung einiger Tests mit dem zuvor konzipierten und implementierten Werkzeug. Dabei werden der Durchführungsort, die Testpersonen und ihr Hintergrund, die durchgeführten Aufgaben und zur Einschätzung der Nützlichkeit gestellte Fragen thematisiert. Außerdem werden verschiedene Beobachtungen und Einschätzungen der Testpersonen analysiert und erörtert.

Der letzte Abschnitt umfasst eine abschließende Bewertung, Evaluation und Diskussion der Ergebnisse. Dabei wird sowohl auf die gewonnenen Erkenntnisse eingegangen als auch ein Ausblick geschaffen.

## 2 Usability

Der Begriff Usability stammt aus der Ergonomie bzw. Software-Ergonomie und besitzt in verschiedenen Bereichen Relevanz. Er findet unter anderem bei der Beschreibung von interaktiven Systemen wie Software-, Web- und Mobil-Anwendungen Einsatz. Der Begriff der Ergonomie beschreibt die wissenschaftliche Disziplin der Arbeitswelt. Die Software-Ergonomie erweitert dieses Konzept zur Beschreibung der Arbeit mit Computersystemen (Sarodnik, et al., 2016 S. 19) und beschäftigt sich mit der Untersuchung der Beziehung zwischen dem Menschen und seinem Arbeitssystem und zielt auf eine Anpassung der Computersysteme an die Anforderungen und Bedürfnisse der nutzenden Person ab (Richter, et al., 2016 S. 213).

### 2.1 Definition

Im allgemeinen Sprachgebrauch werden verschiedene Übersetzungen des englischen Begriffs Usability verwendet. Dazu gehören unter anderem die Gebrauchstauglichkeit, die Nutzbarkeit bzw. Verwendbarkeit und die (Be-) Nutzungsfreundlichkeit (DeepL, 2021). Ebenso vielfältig wie diese Übersetzungen sind auch die verwendeten Definitionen (Richter, et al., 2016 S. 10). Um ein gemeinsames Verständnis dieses Begriffs zu ermöglichen, haben sich in den 1990er Jahren internationale Fachpersonen auf eine Norm zur einheitlichen Definition geeinigt (Sarodnik, et al., 2016 S. 35). In der deutschen Fassung dieser Norm wird die Usability mit Gebrauchstauglichkeit übersetzt, weshalb dieser Begriff, vor allem in wissenschaftlichen Zusammenhängen, häufig als korrekte Übersetzung genannt und verwendet wird.

Besagte Norm, die DIN EN ISO 9241 *Ergonomie der Mensch-System-Interaktion* beschreibt die Gebrauchstauglichkeit als „das Ausmaß, in dem ein System, ein Produkt oder eine Dienstleistung durch bestimmte Benutzer[\*innen] in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen“ (DIN e.V., 2018 S. 9). Somit beschreibt die Usability eine Qualität, die das Zusammenspiel von System, Nutzer\*in und Aufgabe aus dem Blickwinkel des durch den/die Nutzer\*in wahrgenommenen Ausmaßes und der wahrgenommenen Qualität der Zielerreichung betrachtet (Sarodnik, et al., 2016 S. 19f.).

## 2.2 Usability-Prinzipien, -Normen und -Richtlinien

Bereits vor einigen Jahren wurden auf Basis verschiedener Eigenschaften der menschlichen Wahrnehmungen und verschiedener Erkenntnisse über die Nutzung von interaktiven Systemen durch menschliche Nutzer\*innen einige Usability-Grundsätze und -Prinzipien entwickelt. Solche Grundsätze und Prinzipien sind eher abstrakter Natur, weshalb ihre direkte Anwendung ohne tiefgreifendes Fachwissen erschwert sein kann. Usability-Richtlinien können da Abhilfe schaffen. Sie konkretisieren diese Prinzipien, indem sie sie auf bestimmte Anwendungsdomänen beziehen und Hinweise zur Entwicklung entsprechender Produkte mit hoher Usability geben (Richter, et al., 2016 S. 97).

### 2.2.1 Usability-Prinzipien

Sowohl der US-amerikanische Informatiker und Professor Ben Shneiderman als auch der dänische Informatiker Jakob Nielsen stellten auf Basis ihrer Erfahrungen einige Usability-Prinzipien auf. Shneidermans *8 goldene Regeln der Gestaltung von Benutzungsoberflächen* (Shneiderman) und Niensens *10 Heuristiken für die Gestaltung von Benutzungsoberflächen* (Nielsen, 2020) besitzen einige Überschneidungen. Sowohl Shneiderman als auch Nielsen sehen das Streben nach Konsistenz, die minimale mentale Belastung des/der Nutzers/Nutzerin, informative Rückmeldungen und Umkehrmöglichkeiten, sowie die Unterstützung erfahrener Nutzer\*innen, bspw. durch Tastenkürzel, als wesentliche Aspekte einer guten Benutzungsschnittstelle an. Des Weiteren gehen die Autoren auf die Behandlung von Fehlern ein und heben sowohl die Wichtigkeit von informativen und hilfreichen Fehlermeldungen als auch die Unterstützung der Nutzer\*innen durch Maßnahmen zur Fehlervermeidung hervor. Shneiderman geht außerdem darauf ein, dass Aktionssequenzen, bestehend aus mehreren aufeinander folgenden Teilaktionen, als solche kenntlich gemacht werden sollten und mit einer Rückmeldung über ihren Erfolg oder Misserfolg abgeschlossen werden sollten und der/die Benutzer\*in bei der Interaktion mit einer Benutzungsschnittstelle stets das Gefühl der Kontrolle haben sollte. (Dahm, 2006 S. 151-153) (Sonntag, 2015) Nielsen geht dagegen noch etwas intensiver auf die Gestaltung der Benutzungsoberfläche ein und nennt einfache und natürliche Dialoge, sowie ein ästhetisches und minimalistisches Design als grundlegende Aspekte gebrauchstauglicher Benutzungsoberflächen. Zur Unterstützung der Nutzer\*innen sollten nach Nielsen die Ausdrucksweisen des/der Anwenders/Anwenderin verwendet und Hilfen und Dokumentationen angeboten werden. (Dahm, 2006 S. 154-156) (Sarodnik, et al., 2016 S. 147f.)

### 2.2.2 Usability-Normen

Usability-Normen basieren auf der Beobachtung realer Systeme und bilden daraus gesicherte Erfahrungen und Wissensstände ab, die sich auf die Entwicklungsprozesse von Benutzungsschnittstellen und die Eigenschaften beziehen, die gebrauchstaugliche Schnittstellen besitzen sollten. (Preim, et al., 2010 S. 237) Solche Normen sind in ihrer Formulierung bewusst recht allgemein gehalten, um über längere Zeit angemessen zu sein und eine höhere Unabhängigkeit von technischen Änderungen zu bieten (Jacobsen, et al., 2018 S. 59). Sie werden in ausführlicher Abstimmung verschiedener Fachpersonen erstellt und unter anderem vom Deutschen Institut für Normung (DIN) und der Internationalen Organisation für Normung (ISO) angenommen und veröffentlicht.

Ein prominentes Beispiel im Bereich der Usability ist die DIN EN ISO 9241 *Ergonomie der Mensch-System-Interaktion*. Die Normenreihe beinhaltet ergonomische Anforderungen an die Arbeitsplatzgestaltung (Stapelkamp, 2010 S. 328) und zielt auf die Vermeidung gesundheitlicher Schäden von Arbeitnehmenden bei der Arbeit mit Bildschirmgeräten ab (Jacobsen, et al., 2018 S. 61). Innerhalb der Normenreihe sind unter anderem eine Definition der Usability (s. Abschnitt 2.1) und einiger grundlegender Begriffe (Teil 11), einige Grundsätze zur Gestaltung gebrauchstauglicher Dialoge (Teil 110) und eine Prozessdefinition zur Entwicklung gebrauchstauglicher interaktiver Systeme (Teil 210) zu finden. (DIN e.V., 2020)

### 2.2.3 Usability-Richtlinien

Richtlinien können bei der Entwicklung digitaler Systeme unterstützen, um konsistente und gebrauchstaugliche Benutzungsschnittstellen zu erschaffen (Sarodnik, et al., 2016 S. 123). Sie enthalten Empfehlungen zur Lösung wiederkehrender Aufgaben und Probleme und sind praktischer formuliert als die zuvor beschriebenen Usability-Prinzipien und -Normen. Nach Preim und Dachsel sollten Richtlinien klar strukturiert und plausibel begründet sein und auf Erfahrungen bei der Auswahl angemessener Bedienelemente in Hinblick auf ihren konkreten Einsatz beruhen. (Preim, et al., 2010 S. 239) Hersteller- und plattformspezifische Richtlinien bspw. dienen der Aufrechterhaltung der Konsistenz innerhalb einer Anwendung oder über mehrere Anwendungen hinweg und tragen somit zur Gewährleistung eines bestimmten Look-and-Feels bei. (Richter, et al., 2016 S. 97). Preim und Dachsel sehen solche Richtlinien insofern kritisch, da das Interesse der Firma eine wesentliche Rolle bei der Entwicklung darstellt und gewisse Dialogelemente ohne ersichtlichen Vorteil vorzüglich behandelt

werden könnten, um die firmenspezifische Gestaltung zu gewährleisten (Preim, et al., 2010 S. 241). Da heutzutage ein Großteil der Informationssuche und Firmenpräsenz über das Internet erfolgt, haben sich außerdem einige Richtlinien für den Webbereich und die Entwicklung, sowie die Gestaltung von Webseiten etabliert. Hierbei sind vor allem die *Research-based Web Design & Usability Guidelines* und die *Web Content Accessibility Guidelines* (WCAG) zu erwähnen, die von vielen Autoren und Autorinnen zitiert, verwendet oder empfohlen werden. Preim und Dachselst heben die WCAG-Richtlinien positiv hervor, da sie in mehreren Iterationen von Fachpersonen diskutiert wurden und anschließend als offizielles Dokument des W3-Konsortiums (W3C) veröffentlicht wurden (Preim, et al., 2010 S. 241).

### 2.3 Usability-Evaluation

Für die menschenzentrierte Gestaltung und Entwicklung von Benutzungsoberflächen gibt es keine allgemeingültigen Lösungen, die immer anwendbar sind. Deshalb müssen Gestaltungsideen, -vorschläge und -umsetzungen bei der Entwicklung gebrauchstauglicher Produkte regelmäßigen Evaluationen unterzogen werden. (Sarodnik, et al., 2016 S. 24) Usability-Evaluationen zielen dabei meist auf eins von zwei Zielen ab. Eine formative Evaluation wird während des Gestaltungsprozesses durchgeführt und hat die Verbesserung der Schnittstelle zum Ziel. Summative Evaluationen, die im Anschluss an den Gestaltungs- und Entwicklungsprozess vollzogen werden, dienen der Prüfung einer Benutzungsschnittstelle im Rahmen einer Gesamt-Qualitätsbewertung. (Richter, et al., 2016 S. 103f.) (Sarodnik, et al., 2016 S. 24) Einige Methoden der formativen Usability-Evaluation zielen auf die Identifikation von Usability-Problemen ab. Sarodnik und Brau definieren ein Usability-Problem als Herausforderung bei der Erreichung von Zielen innerhalb eines bestimmten Nutzungskontextes durch bestimmte Nutzer\*innen mit hinreichender Erfahrung der Systemdomäne. (Sarodnik, et al., 2016 S. 26) Die Methoden der Usability-Evaluation lassen sich auf verschiedene Art und Weise kategorisieren. Sarodnik und Brau teilen sie in empirische und analytische Methoden ein. Empirische Evaluationsverfahren beinhalten Befragungen und Beobachtungen der tatsächlichen Nutzer\*innen. Im Gegensatz dazu umfassen analytische Evaluationsverfahren die Beurteilung der Usability durch Fachpersonen mit Hilfe formal-analytischer Verfahren und Inspektionsmethoden. (Sarodnik, et al., 2016 S. 119)

### 2.3.1 Empirische Methoden

Empirische Evaluationsmethoden basieren, wie zuvor beschrieben, auf der Teilhabe der tatsächlichen Nutzer\*innen. Usability-Tests dienen dazu, ein interaktives System anhand realistischer Aufgaben zu beurteilen. Erkenntnisse über die Usability des Systems werden dadurch gewonnen, dass Testpersonen bei der Ausführung ihrer Aktionen von Fachleuten beobachtet werden, anschließend Interviews geführt werden oder ggf. Messungen verschiedener Zeit- und Fehlerdaten vorgenommen werden. (Sarodnik, et al., 2016 S. 163) Teil 11 der DIN EN ISO 9241 enthält einen Abschnitt über die Messung der Gebrauchstauglichkeit auf Basis der Eigenschaften Effektivität, Effizienz und Zufriedenstellung. Als Maß für die Effektivität werden Benutzungsfehler oder -schwierigkeiten, unnötige Systemausgaben, Verhältnisse erfolgreicher Ergebnisse, Zielerreichungsgrade und Lernerfolge genannt. Die Effizienz kann durch die Betrachtung verschiedener Ressourcen wie der Zeit, des menschlichen Aufwands, der Kosten und der Materialien eingeschätzt werden. Maße der Zufriedenheit sind eher subjektiven Charakters, weshalb sie vor allem durch Befragungen ermittelt werden. (DIN e.V., 2018 S. 37-39)

### 2.3.2 Analytische Methoden

Die analytischen Evaluationsmethoden umfassen verschiedene formal-analytische Verfahren und Inspektionsmethoden. Zu den formal-analytischen Verfahren gehören sog. Expertenleitfäden und aufgabenanalytische Vorgehensweisen (Sarodnik, et al., 2016 S. 131). Expertenleitfäden werden von Usability-Fachpersonen genutzt, um gestalterische Mängel aufzudecken (Sarodnik, et al., 2016 S. 134). Aufgabenanalytische Vorgehensweisen werden verwendet, um Handlungsschritte zur Erfüllung einer Aufgabe abzubilden und anhand verschiedener Maße zu bewerten. Dazu können verschiedene Modelle bzw. Modellsprachen genutzt werden. (Sarodnik, et al., 2016 S. 131) Ein prominentes Beispiel eines solchen Modells ist das sog. GOMS-Modell (Goals, Operators, Methods, Selection rules). Es beinhaltet Aufgabenstrukturen, bestehend aus Zielen, Operatoren, Handlungsschemata und Auswahlregeln. (Sarodnik, et al., 2016 S. 132) Inspektionsmethoden werden dazu verwendet, potenzielle Usability-Probleme anhand von Designprinzipien oder potenziell kritische Interaktionsaspekte anhand von Design-Aufgabenanalysen zu erkennen (Lavery, et al., 1997). Eine Design-Aufgabenanalyse kann bspw. in Form eines Cognitive Walkthroughs (dt.: kognitiver Durchgang) erfolgen, in dem eine Fachperson sich in die Lage des/der Nutzenden versetzt und einige typische Nutzungsaufgaben durchgeht (Jacobsen, et al., 2018 S. 213).

### 3 Automatisierte Usability-Evaluation von GUIs

Ein häufig genanntes Problem klassischer Methoden zur Usability-Evaluation grafischer Benutzungsoberflächen sind hohe Zeit- und Ressourcenkosten (Khasnis, et al., 2019 S. 85). Zur Untersuchung der Schnittstellen müssen Usability-Fachkräfte und häufig auch Testpersonen herangezogen werden (s. Abschnitt 2.3). Deshalb arbeiten Wissenschaftler\*innen bereits seit einigen Jahren an der Entwicklung von Lösungen, die Teile der Usability-Evaluation automatisieren sollen. Ivory et al. nennen in einer 2001 erschienenen und häufig zitierten<sup>1</sup> Veröffentlichung, neben reduzierten Kosten, als Vorteile einer Automatisierung unter anderem, dass Zeit- und Fehlerkosten besser vorhergesagt werden können, sich die Konsistenz der aufgedeckten Fehler erhöhe, wodurch auch der Vergleich zwischen alternativen Entwürfen möglich werde, und die Bewertung besser in die Designphase der Entwicklung einbezogen werden könne (Ivory, et al., 2001 S. 471).

#### 3.1 Automatisierungsgrad

Ivory et al. beschreiben vier Ansätze zur Automatisierung von Usability-Evaluationsmethoden. Neben der Usability-Evaluation ohne eine Automatisierung gibt es drei verschiedene Automatisierungsgrade. Die automatische Aufzeichnung beschreibt die Erfassung relevanter Informationen über den/die Benutzer\*in und das System. Dies umfasst bspw. die Aufzeichnung von Maus- und Tastaturaktionen oder visuellen Daten durch eine Software. Mit Hilfe einer automatisierten Analyse können potenzielle Usability-Probleme von Benutzungsoberflächen anhand aufgezeichneter Interaktionen oder anderer Informationsgrundlagen automatisch identifiziert werden. Anschließend können Ansätze zur automatisierten Kritik, aufbauend auf einer Analyse, Verbesserungen zur Steigerung der Usability des untersuchten Systems vorschlagen. (Ivory, et al., 2001 S. 473)

#### 3.2 Kategorisierung

In einer 2016 veröffentlichten Arbeit von Bakaev et al. (Bakaev, et al., 2016) über aktuelle Entwicklungen im Bereich der automatisierten Usability-Evaluation werden drei Kategorien

---

<sup>1</sup> Die betrachtete Arbeit liegt sowohl als Papier (*The State of the Art in Automating Usability Evaluation of User Interfaces*) von Melody Y. Ivory und Marti A. Hearst vor als auch in Form eines Kapitels, in dem von Melody Y. Ivory veröffentlichten Buch *An empirical foundation for automated web interface evaluation*. Eine Suche auf Google Scholar hat gezeigt, dass die beiden zusammen 1306 (125 x Buch, 1181 x Papier; Datum der Anfrage: 11.01.2022) Mal zitiert wurden.

genannt, anhand derer sich die Ansätze einteilen lassen. Im Folgenden werden diese Kategorien untersucht. Im Anhang sind Beschreibungen einiger implementierter Ansätze zu jeder Kategorie zu finden (s. A.1).

Interaktionsbasierte Ansätze gewinnen Aussagen über die Usability von Benutzungsoberflächen aus aufgezeichneten Nutzungsaktionen. Dazu werden reale oder inszenierte Interaktionen (bspw. Mausbewegungen und -klicks, Tastatureingaben, aufgerufenen Links; z.B. (Vargas, et al., 2010)) und ggf. ergänzende Informationen (bspw. formale Eigenschaften der Benutzungsschnittstelle wie CSS-Attribute oder HTML-Tags; z.B. (Tiedtke, et al., 2001)) bei der Bearbeitung zuvor definierter Aufgaben aufgezeichnet. Zur Analyse der Usability werden diese aufgezeichnete Interaktionsprotokolle anschließend auf Nutzungsmuster hin untersucht, die auf Usability-Probleme hinweisen (z.B. (Harms, et al., 2014)) oder mit der geplanten Interaktionsfolge zur Bearbeitung der Aufgabe verglichen (z.B. (Vargas, et al., 2010)). Die Referenz-Aufgabenbearbeitung beschreibt die von den gestaltenden oder entwickelnden Personen geplanten Schritte zur Bearbeitung bestimmter Aufgaben, die durch eine Fachperson durchgeführt oder in Form eines (Aufgaben-) Modells definiert werden. Abweichungen von dieser „optimalen“ Interaktion können auf Usability-Probleme hinweisen (Guarino de Vasconcelos, et al., 2012 S. 709). Diese beiden Varianten der interaktionsbasierten automatisierten Usability-Evaluation können kombiniert werden, indem bspw. im Anschluss an den Vergleich mit einer Referenz eine Untersuchung auf Usability-Problemuster hin durchgeführt wird (z.B. (Vargas, et al., 2010)). Bakaev et al. weisen darauf hin, dass bei interaktionsbasierten Verfahren genügend reale Interaktionen benötigt würden, bevor eine Bewertung vorgenommen werden könne. Sowohl Modell- als auch Metriken-basierte Verfahren benötigen im Gegensatz zu den interaktionsbasierten Ansätzen keine Teilhabe durch den/die Nutzer\*in. (Bakaev, et al., 2016 S. 513)

Metriken-basierte Methoden streben eine Evaluation der Benutzungsoberfläche anhand von Kennzahlen an. Diese Kennzahlen sollen bspw. die Übereinstimmung mit Usability-Richtlinien erkennen oder die visuelle Komplexität der Benutzungsoberfläche sowie verschiedene übergeordnete Gestaltungsfaktoren, die im Einklang mit Aspekten der Usability stehen, untersuchen. (Bakaev, et al., 2016 S. 513) Dazu werden relevante Richtlinien oder andere Usability-Kennzahlen ausgewählt und anschließend in Form von automatisch auswertbaren Regeln formuliert (Hamacher, 2006 S. 25-26). Zur eigentlichen Evaluation werden diese Regeln in maschinen-lesbarer Form hinterlegt, anhand derer die gesamte Benutzungsoberfläche

oder einzelne UI-Elemente analysiert und bewertet werden können (z.B. (Marenkov, et al., 2016)). In seiner Arbeit *Automatische kriterienorientierte Bewertung der Gebrauchstauglichkeit interaktiver Systeme* (Hamacher, 2006) vergleicht Hamacher verschiedene Kriterien-Systeme und ordnet diese anhand ihres Interpretationsbedarfs und ihres Automatisierungsgrades an. (Parametrisierte) Designregeln beschreiben dabei „ausformulierte Regeln für konkrete Systemobjekte oder -bereiche“ (Hamacher, 2006 S. 25) und sind bspw. in Richtlinien-Sammlungen zu finden, wie sie in Abschnitt 2.2.3 beschrieben wurden. Im Gegensatz zu Usability-Normen und theoretisch begründeten Gestaltungsgrundsätzen, die bewusst offener formuliert sind, lässt sich ihre Evaluation laut Hamacher besser automatisieren. Er merkt allerdings auch an, dass darin ein Nachteil liege, da Designregeln und Styleguides eine höhere Distanz zu den konkreten Nutzer\*innen und ihren Aufgaben besäßen. (Hamacher, 2006 S. 24f.) Unter anderem deshalb seien laut Bakaev et al. bei der Anwendung Metriken-basierter Methoden die Auswirkungen unterschiedlicher Nutzungsaufgaben und -kontexte auf die Signifikanz der Metriken zu berücksichtigen. (Bakaev, et al., 2016 S. 513)

Modell-basierte Evaluationsmethoden verwenden Nutzer\*innen- und Schnittstellenmodelle, um durch analytische Modellierung oder Simulation Vorhersagen über die Usability einer Benutzungsoberfläche treffen zu können. Wie erfolgreich verschiedene Nutzer\*innen innerhalb einer Anwendung navigieren können, hat einen Einfluss auf ihre Gebrauchstauglichkeit. Deshalb werden Modell-basierte Methoden häufig zur Evaluation der Navigationsfähigkeit verwendet, indem bspw. die Aussagekraft von Navigationselementen (z.B. (Chi, et al., 2003)) und Seitenüberschriften (z.B. (Blackmon, et al., 2002)) untersucht wird. Dazu werden bspw. Nutzer\*innen-Modelle, Modelle der zu untersuchenden Schnittstellen und ein Suchbegriff bzw. eine Aufgabe genutzt. Anhand unterschiedlicher Aspekte wird das Vorgehen verschiedener Nutzer\*innen bei der Navigation simuliert und anschließend evaluiert. Solche Ansätze untersuchen unter anderem die Wahrscheinlichkeit, mit der unterschiedliche Nutzer\*innen verschiedene Navigationselemente verwenden (z.B. (Ivory, et al., 2000)). Sie können aber auch dazu genutzt werden die Reihenfolge zu untersuchen, in der einzelne UI-Elemente wahrgenommen werden und wie hoch die Wahrscheinlichkeit der Interaktion mit diesen, auf Basis einer hinterlegten Aufgabe, ist (z.B. (Feuerstack, et al., 2008)). Bakaev et al. weisen darauf hin, dass die Modellierungs- und Berechnungskomplexität bei einigen solcher Methoden unerschwinglich hoch sein könne und die Anpassung der Modelle an sich ändernde Trends im Usability Engineering recht kostspielig sei. (Bakaev, et al., 2016 S. 513)

### 3.3 Automatisierte Evaluation von UI-Prototypen

Im Bereich der automatisierten Evaluation von UI-Prototypen gibt es einige kommerziell verfügbare Werkzeuge zur Evaluation der Barrierefreiheit. Da die Barrierefreiheit eng mit der Usability von Benutzungsoberflächen zusammenhängt, werden diese Ansätze im Folgenden intensiver betrachtet. Die drei Prototyping-Werkzeuge Figma (Figma Inc.), Adobe XD (Adobe Inc.) und Sketch (Sketch B.V.) bieten die Möglichkeit (teil-) automatisierte Evaluationen mit Hilfe von integrierten Programmen, sog. Plugins, durchzuführen.

Eine Recherche verschiedener Plugins hat gezeigt, dass im Bereich der Barrierefreiheit vor allem Farb- und Kontrast-Evaluationen, die unter anderen auf den WCAG-Richtlinien basieren, automatisiert werden. Stark, eine Suite von Evaluationswerkzeugen, die in Figma, Adobe XD und Sketch zur Verfügung steht, bietet außerdem Vorschläge für WCAG-konforme Farbalternativen an (Stark Lab Inc., 2022). Das Evaluationswerkzeug Adeo (Adeo App Inc.) nimmt eine automatisierte Überprüfung der Größe von Touch-Elementen auf Basis von Apple- und Android-Touchgeräten, den Heuristiken nach Nielsen und den WCAG-Richtlinien vor. Zur Untersuchung eines Entwurfs unter Betrachtung verschiedener Sehbehinderungen, bieten sowohl Stark als auch Adeo verschiedene Simulatoren an, die die Darstellung des zu untersuchenden Entwurfs entsprechend simulieren.

Im Bereich der Usability-Evaluation bieten Plugins von Prototyping-Programmen bislang vor allem Unterstützung in der Durchführung von entfernten (Remote-) Usability-Tests, die automatisch aufgezeichnet und teilweise evaluiert werden. Die Evaluation basiert dabei auf der Integration separater Plattformen und ermöglicht die Nutzung ihrer Funktionalitäten. Für eine automatische Usability-Evaluation ohne Testpersonen gibt es dagegen bislang nur wenige Ansätze in Form solcher Plugins. VisualEyes (Loceye, 2020) und Attention Insight (Attention Insight, 2021), zwei Figma-Plugins, verwenden KI-Methoden zur Vorhersage des wahrscheinlichen Blickverlaufs der Nutzer\*innen.

In ihrer Masterarbeit aus dem Jahr 2020 beschäftigt sich Franziska Trojahn mit der automatischen Usability-Evaluation von Prototypen anhand von Bild- und Kompositions-basierten sowie grundlegend navigatorischen Metriken. Dabei orientiert sie sich an der sogenannten *Quality in Use Integrated Map* (QUIM), einigen Kennzahlen verschiedener Autoren und den WCAG-Richtlinien. Aus den verschiedenen Anforderungen erstellt sie eine Metriken-Sammlung und implementiert die Messung eines dieser Kennwerte, der Schriftkonsistenz,

sowie die Rückgabe des Ergebnisses prototypisch in Form eines Adobe XD Plugins. Durch Zuordnung der Metriken zu einigen Usability-Aspekten (Effizienz, Effektivität, Produktivität, Zufriedenstellung, Lernfähigkeit, Sicherheit, Vertraulichkeit, Barrierefreiheit, Universalität und Nutzbarkeit) begründet sie die Fähigkeit der Metriken-Sammlung zur Evaluation der Usability von Benutzungsoberflächen. (Trojahn, 2020)

Ein an der University of Canterbury entwickeltes Werkzeug zur empirischen Modellierung der Performance von Interaktionen mit Touch-basierten Benutzungsoberflächen soll Gestalter\*innen bei dem Entwurf berührungsempfindlicher Schnittstellen unterstützen, indem es seine Performance vorhersagt. Die Nutzer\*innen des Werkzeugs platzieren und kombinieren einzelne Berührungskaktionen auf Abbildungen einer Benutzungsoberfläche, welche anschließend mit Hilfe des sog. Touch-Level-Models<sup>2</sup> (Rice, et al., 2014) ausgewertet werden. Da es bislang keine empirisch ermittelten Zeitwerte für dieses Modell gibt, werden Daten einer Datenbank verwendet, die aus laufenden webbasierten Feldexperimenten gespeist wird. Auf Basis der Zeitwerte werden Balkendiagramme erstellt, die die Bearbeitungsdauer der geplanten Interaktion darstellen. (Goguey, et al., 2018)

---

<sup>2</sup> Eine für Touch-basierte Benutzungsoberflächen angepasste Variante des Keystroke-Level-Models der GOMS-Familie, welches einzelnen Interaktionen Operatoren mit empirisch ermittelten Zeitwerten zuordnet, um Vorhersagen über die Dauer einer Interaktionsfolge zu machen.

## 4 Anforderungsermittlung

In dieser Arbeit soll ein Werkzeug entwickelt werden, das eine automatisierte Evaluation der Usability von UI-Entwürfen vornehmen und Gestalter\*innen bei der Entwicklung gebrauchstauglicher Benutzungsoberflächen unterstützen soll. Die Entwicklung dieses Plugins erfolgt orientiert an der Prozessdefinition des Usability Engineerings im Rahmen einer menschen- bzw. nutzerzentrierten Entwicklung. Abbildung 1 stellt den Prozess der menschenzentrierten Gestaltung nach DIN EN ISO 9241 Teil 210 (DIN e.V., 2020) dar. Beginnend werden der Nutzungskontext und die Anforderungen der Nutzer\*innen ermittelt. Dazu wird in diesem Abschnitt eine Einordnung des Prozesses der UI-Prototypen-Entwicklung in den Gesamtprozess der Entwicklung interaktiver Systeme vorgenommen und die Ergebnisse einiger Leitfaden-gestützter Interviews zusammengefasst, die zuvor durchgeführt wurden.

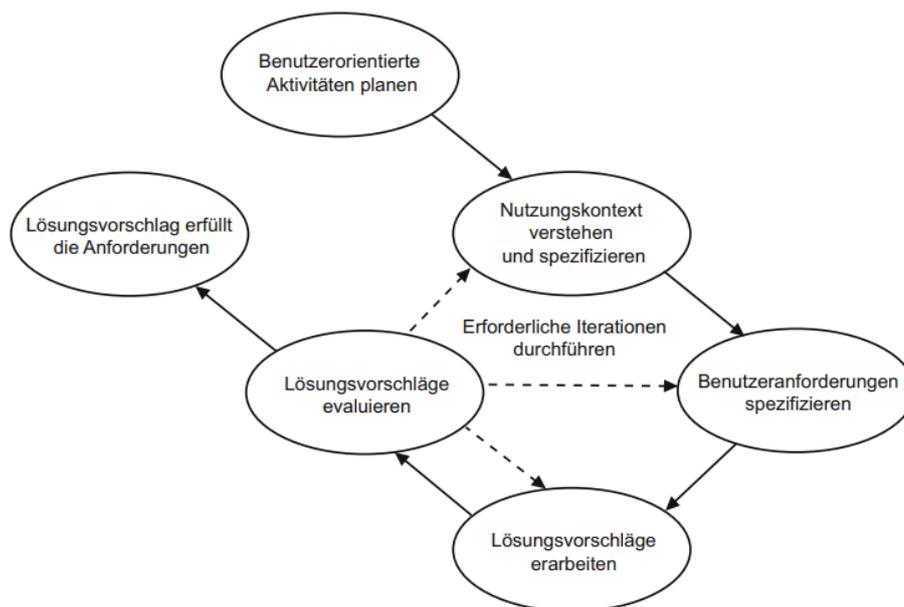


Abbildung 1: Menschzentrierte Gestaltung nach DIN EN ISO 9241 (Richter, et al., 2013 S. 17)

### 4.1 Nutzungskontext

Der sogenannte Nutzungskontext beschreibt die „Kombination von Benutzern [und Benutzerinnen], Zielen, Aufgaben, Ressourcen [sowie der] technische[n], physische[n], soziale[n], kulturelle[n] und organisationsbezogene[n] Umgebungen“ (DIN e.V., 2018 S. 11). Im Folgenden wird der Nutzungskontext für die automatisierte Usability-Evaluation von UI-Entwürfen spezifiziert.

### 4.1.1 Einordnung in den Entwicklungsprozess

Innerhalb verschiedener Prozessbeschreibungen des Usability Engineerings wird Prototyping als relevanter Teil angesehen. Für diese Arbeit wurden die Prozessbeschreibungen des Usability-Engineerings der DIN EN ISO 9241, zu finden in Teil 210 (DIN e.V., 2020), und der Autoren Sarodnik und Brau, zu finden im Buch *Methoden der Usability Evaluation* (Sarodnik, et al., 2016), betrachtet.

Die Prozessdefinitionen sehen die Entwicklung von Prototypen im Anschluss an eine Analyse- und Konzeptphase vor, in der der Nutzungskontext und die Anforderungen der Nutzer\*innen spezifiziert werden. Das Prototyping wird dazu genutzt, visuelle und funktionale Aspekte geplanter Benutzungsschnittstellen zu entwerfen und zu erproben, sodass sie noch vor der tatsächlichen Implementierung evaluiert und ggf. verbessert werden können (Richter, et al., 2016 S. 72). Prototypen lassen sich anhand des Funktionsumfangs, der Funktionstiefe, der Darstellungstreue, der Interaktivität, des Datengehalts und der technischen Reife unterscheiden. Abhängig davon, welche Eigenschaften des späteren Systems untersucht werden sollen, wird der Fokus verteilt. (Richter, et al., 2016 S. 73) Des Weiteren unterscheiden Sarodnik und Brau horizontal und vertikal ausgeprägte Prototypen. Ein horizontaler Prototyp bildet grob den gesamten Funktionsumfang des gewünschten Systems ab, besitzt dabei aber eine geringe Funktionstiefe. Ein vertikaler Prototypen dient stattdessen dazu einzelne Funktionalitäten genauer zu betrachten, die in ihrer Funktionstiefe sehr ausgeprägt sind und realistisch genutzt werden können. Oft bietet sich eine Kombination dieser beiden Typen an, sodass ein Überblick über das Gesamtsystem geschaffen und einzelne Aspekte realistisch untersucht werden können. (Sarodnik, et al., 2016 S. 166) Im Anschluss an die Entwicklung der Prototypen folgt in beiden Prozessbeschreibungen die Evaluation der Lösungsvorschläge und ggf. anschließende Anpassungen der Anforderungen und der Nutzungskontextbeschreibung zur Überarbeitung und Verbesserung der Gestaltungslösung. Wenn ein Lösungsvorschlag die Anforderungen erfüllt, wird er an die Entwickler\*innen zur Implementierung weitergegeben. (DIN e.V., 2020) (Sarodnik, et al., 2016 S. 91-103)

An der Entwicklung von Prototypen können sowohl UX- (User Experience<sup>3</sup>) als auch UI-Designer\*innen beteiligt sein. Die Aufgaben von UX-Designer\*innen konzentrieren sich vor allem auf die Optimierung von Produkten für eine effektive, effiziente und zufriedenstellende Nutzung. Die Aufgaben von UI-Designer\*innen liegen dagegen vor allem in der visuellen Gestaltung, sowie der Präsentation und der Interaktivität des Produkts. Die Weitergabe der Erkenntnisse von UX- an UI-Designer\*innen kann durch Styleguides erfolgen. Styleguides können Branding-Richtlinien, Farben, Schriftarten, Layouts und Raster, Buttons, Icons, grundlegenden Kodierungsrichtlinien, sowie Informationen zu globalem und Formularverhalten beinhalten. (Canolli, 2021)

#### **4.1.2 Evaluation von Prototypen**

Abhängig vom Detailgrad und dem Funktionsumfang des UI-Prototypen können Unterschiede in der Evaluation entstehen. Werden Prototypen mit Hilfe technischer Werkzeuge entwickelt, repräsentieren sie ein funktionierendes Teilsystem und können im Rahmen von klassischen Nutzer\*innen-Tests realistisch erprobt werden. Allerdings werden Prototypen, vor allem in früheren Projektphasen, häufig papierbasiert erstellt. Dabei bietet sich vor allem die Evaluation mit analytischen Methoden an. (Sarodnik, et al., 2016 S. 166)

## **4.2 Nutzer\*innen**

Das in dieser Arbeit entwickelte Werkzeug richtet sich vorwiegend an UI- und UX-Gestalter\*innen. Um einige Erfahrungsberichte und ein erstes Meinungsbild zu einem solchen Werkzeug zu erhalten, wurden Leitfaden-gestützte Interviews mit einem Masterstudenten der Medieninformatik, einem Frontend-Entwickler und einem Mediengestalter geführt. Ein Leitfaden-gestütztes Interview beschreibt eine semi-strukturierte Befragungsmethode der qualitativen Sozialforschung. Die Basis des Interviews bildet ein Leitfaden, der dazu dienen soll, sowohl eine gewisse Strukturiertheit als auch Offenheit im Gespräch zu ermöglichen. Somit kann der/die Interviewer\*in zuvor bereits eine Reihe von Themen und Fragerichtungen festlegen, es wird aber auch eine dynamische Gesprächsentwicklung ermöglicht. (Stübling, 2018 S. 102)

---

<sup>3</sup> Die User Experience bezieht sich per Definition auf das Gesamterlebnis der Nutzer\*innen vor, während und nach der Verwendung eines Systems (Richter, et al., 2016 S. 12). Die Usability ist ein Teilaspekt der User Experience. (Sarodnik, et al., 2016 S. 22)

### 4.2.1 Erfahrungen

Der befragte Masterstudent absolvierte im Rahmen seines Bachelorstudium ein fünf-monatiges Praktikum im Bereich des UX-Designs. Dabei entwickelte er Prototypen aufbauend auf Workshops mit verschiedenen Interessensvertretern. Bei der Entwicklung verwendete er das Prototyping-Werkzeug Sketch (Sketch B.V.) und ging auf Basis der Workshop-Ergebnisse und Styleguides der Kundinnen und Kunden und/oder der eigenen Firma vor. Die Evaluation der Prototypen erfolgte meist auf Basis von Rückmeldungen der Nutzer\*innen und der Kontrolle bzw. Kritik durch Kolleginnen und Kollegen mit langjähriger Erfahrung.

Der interviewte Frontend-Entwickler besitzt bislang selbst keine professionelle Erfahrung mit der Entwicklung von UI-Prototypen. Zur Programmierung von Anwendungen nutzt er allerdings von UI-Designerinnen und Designern entwickelte Gestaltungsentwürfe. Diese erhält er entweder in Form von PDF-Dateien oder als Projektdateien entsprechender Prototyping-Werkzeuge.

Der befragte Mediengestalter besitzt Gestaltungserfahrungen im Bereich des UX- und UI-, sowie Web- und App-Designs. Da er innerhalb seiner Firma bereits an verschiedenen Projekten teilgenommen hat, konnte er ausführliche Einsichten in seinen Arbeitsalltag geben. Zur Gestaltung verwendet er das Prototyping-Werkzeug Figma (Figma Inc.). Das Vorgehen beschreibt er wie folgt. Zu Beginn setzen sich Gestalter\*innen, Entwickler\*innen und Projektmanager\*innen zusammen und besprechen das zu entwickelnde Produkt und die gewünschten Funktionalitäten. Dabei fassen sie die Nutzungsanforderungen aus Sicht der Nutzer\*innen zusammen. Optimalerweise, sagt er, wäre damit auch eine vertiefende Recherche in Form von Nutzer\*innen-Befragungen verbunden, die allerdings aus verschiedenen Gründen nicht immer möglich sei. Im Anschluss an die Definition der Nutzungsanforderungen beginnen die Gestalter\*innen mit dem Entwurf von Lösungen in Zweier-Teams. Dabei orientieren er und seine Kolleginnen und Kollegen sich unter anderem an bereits bestehenden Anwendungen, Vorgehensweisen wie dem 8-Pixel-Grid<sup>4</sup>, Styleguides und bewährten Mustern, die die Nutzer\*innen in ihren erlernten Vorgehensweisen unterstützen. Nach der ersten Gestaltung setzen sie sich mit den Entwickler\*innen zusammen, stellen die Entwürfe und

---

<sup>4</sup> Die Einteilung des Entwurfs in acht Pixel Abschnitte. Daran orientiert werden die Elemente positioniert, sodass immer ein Abstand von acht Pixeln oder einem Vielfachen davon verwendet wird. Damit kann die Anordnung von UI-Elementen schnell und konsistent vorgenommen werden. (aus der Erklärung des Befragten)

eventuelle Alternativen vor, und besprechen, wie die Ideen technisch umgesetzt werden können. In Bezug auf die Betrachtung der Barrierefreiheit und Usability bemerkt er, er selbst verwende vor allem das Wissen, das er in der Ausbildung gelernt habe. Innerhalb des Unternehmens gebe es aber außerdem ein Design-System, welches von einigen Gestalterinnen und Gestaltern sowie einer Gruppe von Entwicklerinnen und Entwicklern verwaltet werde. Die Inhalte des Design Systems seien Farben und Komponenten wie Buttons, die ausgiebig getestet und auf die Konformität mit Richtlinien (vor allem in Bezug auf die Farbgebung und Sehschwächen) evaluiert und angepasst würden. Er als Designer könne diese Komponenten anschließend importieren und verwenden. In Bezug auf die Verwendung von Plugins sagt er, dass es innerhalb seiner Firma keine Vorgaben gäbe und die Nutzung eher persönlicher Natur wäre. Seiner Erfahrung nach erleichtern gute Plugins notwendige und regelmäßig durchgeführte Tätigkeiten, sind performant, beeinflussen nicht die Leistung des Gestaltungsprogramms und sind einfach und schnell zu bedienen. Plugins, in denen zu viele Konfigurationsschritte notwendig sind, empfand er in der Vergangenheit als kontraproduktiv.

#### **4.2.2 Vorstellungen zur Evaluation**

Um Wiederholungen vorzubeugen, werden die Anregungen, Wünsche und Vorstellungen der befragten Personen zu einem Werkzeug zur automatisierten Usability-Evaluation zusammengefasst beschrieben.

Einer der Befragten sagte, dass er sich vorstellen könne, dass das Werkzeug anfänglich die grundlegende Gestaltung der Benutzungsschnittstelle auf Basis gewünschter Elemente übernehmen könne. Anschließend sollte die Kontrolle über alles Weitere allerdings bei dem/der Gestalter\*in liegen. Im Hinblick auf die Evaluation wurden verschiedene Dinge angemerkt. Einer der Befragten merkte an, dass eine willkürliche Auswahl und Kombination mehrerer Elemente zur Evaluation weniger sinnvoll sein könnte, da diese eventuell in keinem geeigneten Zusammenhang miteinander stehen könnten und eine Evaluation, seiner Ansicht nach, besser auf Basis einzelner Elemente oder Elementtypen erfolgen sollte. Außerdem wurde, aufgrund des Vorgehens bei der Gestaltung, eine Evaluation auf Basis verschiedener Stufen (Evaluationsgrade) vorgeschlagen. Dazu könnten zuerst grundlegende Aspekte wie Farben und Größenverhältnisse evaluiert und anschließend dynamische Objekte wie Buttons oder Formate und Inhalte von Texten betrachtet werden. Auf Basis seiner Erfahrungen in der Zusammenarbeit mit Gestalterinnen und Gestaltern merkte der befragte Frontend-

Entwickler außerdem an, dass die Prüfung der Umsetzbarkeit von Entwürfen auf dem später genutzten System sinnvoll sein könne.

Auf die Frage nach den erwünschten Rückmeldungen des Systems wurden Begründungen für die Hervorhebung problematischer Aspekte und Hinweise für die Korrektur dieser genannt. Einer der Befragten schlug vor, dass die Darstellung der Evaluationsergebnisse in Form einer visuellen Gegenüberstellung des eigenen und des vorgeschlagenen Entwurfes erfolgen könnte. Änderungen könnten dann in den eigenen Entwurf übernommen oder abgelehnt werden. Eine Alternative dazu, die sich in einem anderen Interview herausgestellt hat, könnte die Markierung von Problemen innerhalb des Entwurfes sein, die den/die Gestalter\*in bei der Identifikation problematischer Bereiche unterstützen könnte. Außerdem wurde die Rückgabe einer Liste von Problemen mit verschiedenen Schweregraden, ähnlich der Rückmeldung einer Konsole bei der Programmierung, genannt.

### **4.3 Zusammenfassung**

Die automatisierte Usability-Evaluation mit Hilfe des zu entwickelnden Werkzeugs sollte sich angenehm in den Arbeitsablauf des/der UI-Designers/-Designerin einbinden lassen und als Grundlage für Gespräche mit anderen entwickelnden Personen genutzt werden können. Die Nutzung und Steuerung des Werkzeugs sollte keinen negativen Einfluss auf die Effektivität, Effizienz und Zufriedenheit der Gestalter\*innen haben. Da UI-Designer\*innen, wie unter anderem aus den Erzählungen des interviewten Mediengestalters hervorging, meist bestimmten Abläufen in der Erstellung eines Entwurfes folgen, sollte die Durchführung sich wiederholender Arbeitsschritte unterstützt werden und einzelne Aspekte der Evaluation konfigurierbar sein. Außerdem sollte die Untersuchung verschiedener Gestaltungsalternativen unterstützt werden und Grundlagen zur Entscheidungsfindung geboten werden. Die Evaluation sollte insofern konfigurierbar sein, dass unterschiedliche Aspekte des Entwurfs, wie verschiedene Elementtypen, betrachtet werden können. Zur Unterstützung der Gestalter\*innen bei der Behebung eventueller Usability-Probleme sollte die zu entwickelnde Anwendung Hinweise innerhalb des Entwurfes bieten und Erläuterungen zu Problemen geben, so dass die Gestalter\*innen verstehen können, wie sie das Problem beheben können. Weitere Anforderungen an die gebrauchstaugliche Gestaltung des Werkzeugs sind unter anderem den Usability-Prinzipien aus Abschnitt 2.2.1 zu entnehmen.

## 5 Plattform

### 5.1 Auswahl der Plattform

Eine kleine Auswahl kommerzieller Prototyping-Werkzeuge bietet die Möglichkeit, Plugins zu entwickeln, die direkt auf Entwürfe, einzelne UI-Elemente und deren Attribute zugreifen können. Eine Untersuchung verschiedener Werkzeuge hat gezeigt, dass die Entwicklung solcher Plugins aktuell durch die vier Tools Adobe XD (Adobe Inc.), Figma (Figma Inc.), Justinmind (Justinmind) und Sketch (Sketch B.V.) unterstützt wird. Das Werkzeug Sketch, welches lediglich für das Betriebssystem macOS angeboten wird, kommt im Rahmen dieser Arbeit nicht in Frage, da das Plugin auf einem Windows-Betriebssystem entwickelt wird. Da die Entwicklung einer individualisierbaren Benutzungsoberfläche nicht möglich zu sein scheint, fällt auch Justinmind aus dieser Wahl heraus. Die beiden Werkzeuge Adobe XD und Figma ähneln sich sowohl in der Plugin-Entwicklung als auch bei ihren Funktionalitäten.

Eine seit 2017 jährlich durchgeführte Online-Umfrage der UI-Designer Jordan Bowman und Taylor Palmer untersucht die Beliebtheit verschiedener Design-Werkzeuge anhand unterschiedlicher Kategorien. Die Umfrage wird durch Firmen wie UsabilityHub (UsabilityHub Pty. Ltd.) und Useberry (Useberry) unterstützt und richtet sich an Personen, die an der Entwicklung digitaler Produkte beteiligt sind. 2021 nahmen über 3.000 Personen aus 111 Ländern teil. Dabei belegte Figma in den Kategorien *UI-Design*, *Prototyping*, *Developer Handoff*, *Design Systems* und *Versioning and file management* den ersten Platz. (Bowman, et al., 2021) Laut dieser Umfrage scheint Figma folglich eine große Beliebtheit unter Gestalterinnen und Gestaltern digitaler Systeme zu besitzen, die ebenso in Online-Artikeln wie bspw. der IT-Nachrichten-Plattform Heise (Berger, 2022) unterstrichen wird. Unter anderem aufgrund dieser Beliebtheit, einer ausführlichen Dokumentation und Hilfestellung, sowie der bereits zu diesem Zeitpunkt umfangreichen Plugin-Bibliothek wird Figma als Plattform für die Entwicklung einer automatisierten Usability-Evaluation gewählt.

### 5.2 Figma

Bei Figma (Figma Inc.) handelt es sich um ein webbasiertes Prototyping-Werkzeug der Firma Figma Inc., das vor allem die Unterstützung der Kollaboration während der Entwicklung digitaler Produkte zum Ziel hat. Figma ist grundsätzlich kostenlos. Durch Hochstufung auf eine der zwei kostenpflichtigen Varianten (Professional, Organization) sind allerdings

eine größere Anzahl an Entwürfen und weitere ergänzende Funktionalitäten möglich (Figma Inc.). Neben der eigentlichen Design-Plattform können mittlerweile außerdem Online-Whiteboards mit dem Dienst FigJam (Figma Inc.) erstellt werden. Da FigJam für die Thematik dieser Arbeit nicht relevant ist, wird im weiteren Vorgehen nicht darauf eingegangen. Wenn von Figma gesprochen wird, ist im Folgenden stets die Design-Plattform gemeint.

### 5.2.1 Benutzungsoberfläche

Die Gestaltung von UI-Entwürfen in Figma erfolgt innerhalb eines Editors, bestehend aus vier Hauptbereichen (s. Abbildung 2). An der oberen Seite befindet sich die Symbolleiste mit dem Dateimenü und verschiedenen Werkzeugen zur Gestaltung des Entwurfs. An der rechten Seite der Symbolleiste befinden sich einige Avatare, die Auskunft darüber geben, welche Nutzer\*innen den Entwurf geöffnet haben. Daneben gibt es Möglichkeiten zum Teilen des Entwurfs und zur interaktiven Darstellung in Form eines klickbaren Prototypen. Unter der Symbolleiste befindet sich die Leinwand, innerhalb der der Gestaltungsentwurf dargestellt wird. An der linken und rechten Seite befindet sich jeweils eine Seitenleiste. Die linke Seitenleiste ermöglicht den Zugriff auf die Seiten (Pages), Ebenen (Frames) und Elemente des Entwurfs sowie auf interne und externe Komponenten Projekt-interner Design Systeme (Libraries). Die rechte Seitenleiste bietet Möglichkeiten zur Bearbeitung einzelner Entwurfselemente und den Zugriff auf Interaktionen und Animationen, sowie einen Bereich zur Ansicht der Elemente in Codeform. (Figma Inc.)

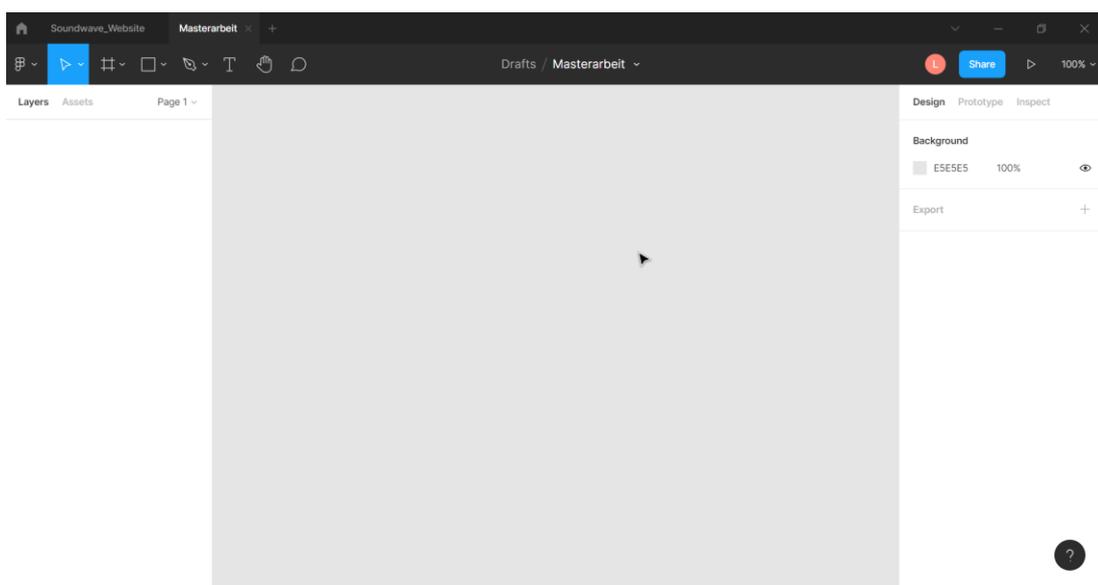


Abbildung 2: Figma-Editor Benutzungsoberfläche

### 5.2.2 Dokumentenstruktur

Entwürfe, die in Figma-Dokumenten gespeichert werden, sind in Form eines Knotenbaums (s. Abbildung 3) aufgebaut. Die Wurzel jedes Figma-Entwurfs ist ein Dokumenten-Knoten, über den auf den Inhalt der Entwurfsdatei zugegriffen werden kann. Ein Dokumenten-Knoten kann mehrere Kinder besitzen. Bei diesen Kindern handelt es sich um Seiten-Knoten, die einzelne Seiten des Entwurfs repräsentieren. Ein Seiten-Knoten kann wiederum eigene Kinder besitzen.

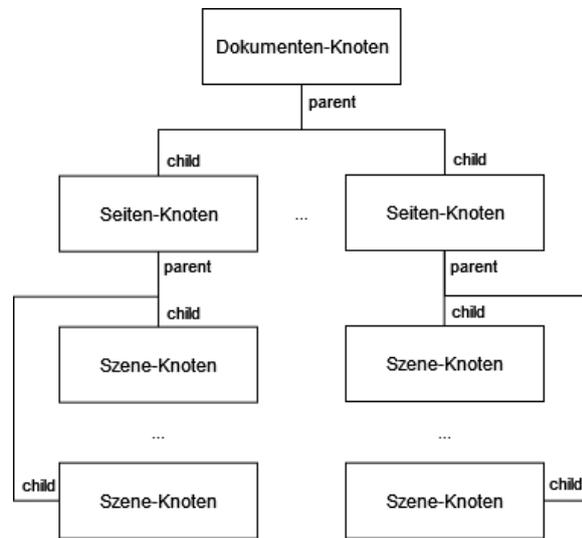


Abbildung 3: Figma-Dokumentenstruktur

Dabei handelt es sich um Szene-Knoten, die

verschiedenen Typs sein können. Dazu gehören Knoten, die geometrische Formen repräsentieren (wie Rechtecke oder Ellipsen); Text-Knoten; Komponenten-Knoten; Gruppen-Knoten; Knoten, die eine Kombination von Formebenen (Vereinigung, Subtraktion, Überschneidung und Ausschluss) darstellen und Knoten mit Quellcode. (Figma Inc.)

### 5.2.3 Plugins

Im Figma-Editor können Plugins über das Dateimenü, per Rechtsklick oder über das Tastenkürzel STRG + P geöffnet werden. Die ersten beiden Varianten nutzen eine Menüstruktur innerhalb der alle installierten Plugins ausgewählt werden können. Der Schnellzugang über STRG + P erwartet die Eingabe des Plugin-Namens.

Figma-Plugins basieren auf einem Plugin-Code, der in JavaScript (bzw. TypeScript) verfasst wird. Der Plugin-Code kann direkt mit Figma-Entwürfen interagieren, ihre Inhalte modifizieren oder neue Inhalte generieren. Wird der Plugin-Code in TypeScript geschrieben, stellt Figma eine Typisierungsdatei bereit, anhand derer Fehler in Bezug auf Knotentypunterschiede erkannt werden können. Zur Interaktion mit dem Plugin ist es außerdem möglich, eine optionale Benutzungsoberfläche zu erstellen. Die Kommunikation zwischen dem Plugin-Code und der Benutzungsoberfläche erfolgt mittels eines Nachrichtensystems. Zur Erstellung der Benutzungsoberfläche wird standardmäßig HTML verwendet. Alternativ können aber auch Bibliotheken wie ReactJS (Meta Platforms) oder VueJS (You) genutzt

werden. Komplexe Plugins, die aus einer großen Anzahl von Dateien bestehen, können mit Hilfe von Webpack (Koppers) gebündelt werden. (Figma Inc.)

#### **5.2.4 APIs und Zugriff auf Gestaltungselemente**

Bei der Entwicklung von Figma-Plugins können Application Programming Interfaces (APIs) verwendet werden, die den Zugriff auf das Dokument, auf den sichtbaren Bereich der Leinwand, auf den lokalen Speicher der Nutzer\*innen und auf Plugin-Parameter ermöglichen. Parameter ermöglichen die Weitergabe von Informationen an das Plugin, ohne eine Benutzungsoberfläche erstellen zu müssen. Über `figma.root` kann auf den Dokumentenknoten und alle seine Kinder sowie deren Kinder zugegriffen werden. (Figma Inc.)

#### **5.2.5 Prototyping System**

Das Prototyping-System von Figma kann dazu genutzt werden, interaktive Abläufe zu erstellen, die anschließend innerhalb des Entwurfs visualisiert und ausprobiert werden können. Innerhalb eines Entwurfs können mehrere sog. Flows erstellt werden, die ein Netzwerk von Frames und Verbindungen zwischen diesen Frames darstellen. Dabei sind verschiedene Interaktionen möglich, die durch einen Trigger ausgelöst werden. Zu diesen Auslösern gehören Mausklicks, Tastenanschläge, das Ziehen bzw. Verschieben eines Elements, das Schweben mit der Maus über einem Element, und einzelne Mausektionen wie Auf- und Abbewegungen. Die Aktionen, die auf solche Auslöser folgen können, sind die Navigation zu anderen Frames, der Wechsel zu einem anderen Element, das Öffnen, Wechseln und Schließen von Overlays, Scroll-Bewegungen und das Öffnen von Hyperlinks. (Figma Inc.)

## 6 Konzeptionierung

In diesem Abschnitt wird aufbauend auf den zuvor gesammelten Erkenntnissen und Anforderungen ein Konzept zur automatisierten Usability-Evaluation in Form eines Figma-Plugins entwickelt. Im Anschluss daran wird dieses Plugin implementiert und getestet.

Für eine automatisierte Usability-Evaluation ohne die Beteiligung von Nutzerinnen und Nutzern eignen sich Metriken- und Modell-basierte Vorgehensweisen (s. Abschnitt 3.2). Durch das Plugin sollen Gestalter\*innen bei der Entwicklung von Gestaltungsentwürfen zielführend unterstützt werden. In Abschnitt 4.2.1 haben sich allein durch die Gespräche mit zwei UI-Gestaltern Unterschiede in der Herangehensweise bei der Entwurfsgestaltung, sowie in Abschnitt 4.2.2 unterschiedliche Vorstellungen für ein solches Plugin feststellen lassen. Dabei wurde eine mehrstufige Evaluation bzw. eine Evaluation verschiedener Aspekte des Prototypen als sinnvoll erachtet. Um dies zu unterstützen, wird für das Plugin sowohl eine Methode der Metriken- als auch der Modell-basierten Evaluation verwendet, ohne aber eine Reihenfolge bei der Evaluation vorzuschreiben. Im Rahmen der Interviews sprach eine der befragten Personen davon, dass ihr Unternehmen ein Design System verwende, innerhalb dessen Komponenten vorlägen, die auf die Konformität mit einigen Richtlinien geprüft worden wären. Da diese Überprüfung sich laut dem Befragten vor allem auf die Farbgebung und eventuelle Sehschwächen fokussiert, ist es interessant, sie im Rahmen einer Metriken-basierten Evaluation durch weitere Aspekte zu ergänzen. Die Modell-basierte Evaluation kann zur Evaluation anderer, vor allem funktionaler Aspekte der Usability genutzt werden. Die Verwendung eines Modells, wie dem GOMS (s. Abschnitt 2.3.2), bietet außerdem die Möglichkeit, Gestaltungsalternativen auf benutzungsorientierte Aspekte hin zu vergleichen (John, et al., 1994 S. 13).

### 6.1 Metriken und Evaluationsprofile

Die Basis einer Metriken-basierten Evaluation bilden i.d.R. formatierte Richtlinien oder andere Usability-Kennzahlen (s. Abschnitt 3.2). In Abschnitt 2.2.3 wurde bereits auf zwei bekannte Richtlinien-Sammlungen eingegangen. Die *Research-Based Web Design and Usability Guidelines* (U.S. Department of Health & Human Services, 2006) und *Web Content Accessibility Guidelines* (W3C, 2018) beinhalten einige Richtlinien, die nicht direkt zur Evaluation von Prototypen verwendet werden können, da sie relativ allgemein formuliert sind, mit der Performance der zugrunde liegenden Hardware in Verbindung stehen, visuell

zusammenhängende Elemente adressieren oder abhängig von inhaltlichen Aspekten sind. Lediglich die Metriken *Enable Access to Homepage*, *Use at Least 12-Point Font*, *Use Appropriate Text Link Lengths* (U.S. Department of Health & Human Services, 2006), *Target Size* und *(Non-Text) Contrast* (W3C, 2018) eignen sich zur Evaluation von Prototypen.

Die in Abschnitt 3.3 vorgestellte Masterarbeit (Trojahn, 2020) thematisierte die Erstellung einer Metriken-Sammlung zur Usability-Evaluation von UI-Prototypen. Eine Evaluation des Prozentsatzes alternativer Texte multimedialer Inhalte, die in dieser Sammlung vorgesehen ist, kann in Figma nicht evaluiert werden, da Grafiken oder anderen Inhalten keine Alternativtexte hinterlegt werden können. Die komplette Liste der verwendeten Metriken ist im Anhang dieser Arbeit zu finden (s. A.3). Aus den zuvor genannten Richtlinien-Sammlungen wird lediglich das Maß der Schriftgröße (*Use at Least 12-Point Font*) übernommen, da die anderen zur Evaluation von Prototypen angemessenen Kennzahlen bereits in der Metriken-Sammlung nach Trojahn vorliegen.

Um wiederholte Herangehensweisen zur Evaluation zu unterstützen, werden Evaluationsprofile eingeführt. Diese Profile können durch die Nutzer\*innen selbst angelegt oder ein vordefiniertes Profil gewählt werden. Einige der in Abschnitt 3.2 untersuchten Herangehensweisen nutzen die Quellen der Richtlinien oder verschiedene Kategorien zur Gruppierung der Metriken. Marenkov et al. (s. A.1) verwenden bspw. die Kategorien *Mobile Usability Guidelines*, *Common Usability Guidelines* und *Accessibility Guidelines* (Marenkov, et al., 2018). Die hier verwendete Metriken-Sammlung anhand ihrer Quelle zu gruppieren, erscheint allerdings wenig sinnvoll, da die einzelnen Metriken nicht explizit einer Quelle zugeordnet werden können. Ebenso erscheint eine Kategorisierung nach Gerätetyp (Mobil, Desktop) nur wenig nützlich, da es sich größtenteils um Metriken handelt, die sich nicht auf einen bestimmten Gerätetyp beziehen. Stattdessen können die Usability-Kriterien der QUIM-Metriken-Sammlung zur Kategorisierung verwendet werden. Ein Kriterium beschreibt eine Charakteristik eines Softwareprodukts und kann durch mehrere Metriken gemessen werden (Padda, 2003 S. 34). Da die Schriftgröße separat zur Metriken-Sammlung hinzugefügt wird, muss eine Zuordnung zu den Usability-Kriterien erfolgen. Anhand der Definition der Kriterien (Padda, 2003) und dem Vergleich mit anderen Metriken werden ihr die Kriterien Präsentation und Lesbarkeit zugeordnet. Die daraus erzeugten Evaluationsprofile sind im Anhang (s. A.4) zu finden.

## 6.2 Modell zur Evaluation von Interaktionsabläufen

Das Keystroke Level Model (KLM) der GOMS-Familie bietet die Möglichkeit, schnelle Vorhersagen über die Dauer der Bearbeitung zuvor definierter Aufgaben zu erstellen und wird häufig zum Vergleich verschiedener Gestaltungsalternativen genutzt (John, et al., 1994 S. 13). Dazu verwendet das KLM einen Satz von Operatoren und empirisch ermittelten Zeitwerten. Direkte Interaktionen in Form von Tastatureingaben und Mausclicks werden durch die Operatoren K (Keystroke, durchschnittlich 0,2s) und P (Pointing) beschrieben. P beschreibt die Dauer für den Vorgang des Anvisierens (AD) eines Ziels mit der Maus und kann durch das Fitts'sche Gesetz (s. Gleichung 1) ermittelt werden. Das Gesetz beschreibt den „Zusammenhang zwischen der Zeit zur Positionierung und der Entfernung des Cursors vom Ziel“ (D), sowie der Breite des Zielobjekts (S) (Dahm, 2006 S. 106).

$$AD = a + b * \log_2 \left( \frac{D}{S} + c \right) \quad (1)$$

Muss zwischen aufeinander folgenden Aktionen zwischen der Maus und der Tastatur gewechselt werden, wird zur Beschreibung dieses Wechsels der H-Operator (Homing, 0,4s) verwendet. Die Reaktionszeit R(t) beschreibt Momente, in denen auf eine Rückmeldung des Systems gewartet werden muss und ist abhängig von dem zugrunde liegenden System. Der Operator M (1,35s) beschreibt die mentale Vorbereitung auf eine Handlung. (John, et al., 1994 S. 13) Die Positionierung des mentalen Operators erfolgt unter Verwendung einiger Regeln. Jedem K und jedem P, das einen Befehl auswählt, wird ein M voran gesetzt. Anschließend werden antizipierte Ms (Vorbereitung bereits in einem vorherigen M), Ms in kognitiven Einheiten (zusammenhängende Aktivitäten wie die Eingabe von Zeichenketten), Ms vor aufeinander folgenden Begrenzern (Begrenzer markieren das Ende eines Wortes oder Satzes), Ms, die Begrenzer von Befehlen sind (Begrenzer, die zur Ausführung eingegeben werden müssen) und Ms, die von einem R überlagert werden, entfernt. (Dahm, 2006 S. 105) Da sich das klassische KLM vor allem auf die Untersuchung von Desktop-basierten Anwendungen konzentriert, heutzutage aber vermehrt Mobilanwendungen zum Einsatz kommen, haben sich einige Wissenschaftler damit beschäftigt, Modelle zu entwickeln, die die Evaluation von touchbasierten Anwendungen mit dem KLM ermöglichen (u.a. (Rice, et al., 2014) und (Goguey, et al., 2018)). Solche angepassten Modelle können im Plugin dazu verwendet werden, die Evaluation auf Basis verschiedener Plattformen zu ermöglichen.

### 6.3 Formatierung von Interaktionsabläufen

Die Modell-basierte Evaluation soll eine funktionale und nutzungsorientierte Evaluation ermöglichen. Dafür bietet sich die Untersuchung einzelner Interaktionsabläufe an. Da solche Abläufe meist mit einem bestimmten Ziel durchgeführt werden, werden sie als Aufgaben beschrieben. Eine Aufgabe besteht, im Rahmen dieser Arbeit, aus einem eindeutigen Namen, einer zufällig ausgewählten Farbe und mindestens einem Bearbeitungsschritt. Ein Bearbeitungsschritt beschreibt eine einzelne Interaktion wie einen Klick auf einen Button oder die Eingabe von Text in ein Eingabefeld. Da sich die Komplexität von Aufgaben ggf. unterscheidet und um der Erzeugung zu großer Aufgaben entgegenzuwirken, werden außerdem sog. Szenarien definiert. Ein Szenario kombiniert mehrere Aufgaben und besitzt aus Gründen der Zuordnung und Unterscheidbarkeit einen eindeutigen Namen.

Um auch innerhalb des Gestaltungsentwurfs einen Überblick über die Aufgaben erhalten zu können, werden Annotationen verwendet. In Figma gibt es bereits einige Plugins, die Annotationen verwenden, um die Reihenfolge verschiedener UI-Elemente festzulegen. Abbildung

4 zeigt die Annotation der Stark-Suite (Stark Lab Inc., 2022), die dazu verwendet wird, die Reihenfolge bei einer Tabulator-Steuerung zu visualisieren. Die grundlegende Gestaltung dieser Annotationen eignet sich gut für die Annotation von Aufgaben. Durch die Nutzung der bei der Definition der Aufgabe zufällig gewählten Farbe kann die mentale Zuordnung gewährleistet werden. Da es vorkommen kann, dass ein einzelnes Interaktionselement innerhalb mehrerer Aufgaben verwendet wird, müssen mehrere Annotationen

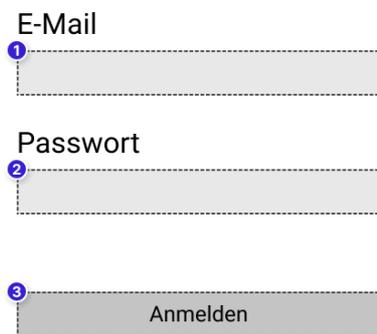


Abbildung 4: Stark-Annotation

an einem Element angezeigt werden können. Für die korrekte Evaluation von Eingabefeldern mit Hilfe des KLM werden durch den/die Nutzer\*in typische Eingabebeispiele angegeben, welche innerhalb des Eingabefelds dargestellt werden. Wenn die Annotationen direkt mit dem entsprechenden UI-Element verbunden werden, lassen sich durch einfaches Verschieben bereits unterschiedliche Gestaltungsideen untersuchen und evaluieren.

Durch die Verwendung von Aufgaben-Annotationen werden einzelne Funktionsabläufe direkt innerhalb der geplanten Anwendung visualisiert. In Abschnitt 4.2.1 sprach der interviewte Mediengestalter an, dass Gestaltungsalternativen entwickelt und anschließend mit

Entwickler\*innen etc. besprochen würden. Hier könnten Annotationen helfen, allen Beteiligten die Funktionsabläufe klar darzustellen und somit eine Basis zur Diskussion liefern.

## 6.4 Hinweismuster für Usability-Probleme

Das KLM ist vor allem dazu geeignet, mehrere Gestaltungsalternativen zu vergleichen (John, et al., 1994 S. 13). Der Wert der Bearbeitungsdauer für eine Aufgabe kann allein nur wenig Auskunft über die Gebrauchstauglichkeit einer Benutzungsoberfläche geben. In Abschnitt 3.2 wurden im Rahmen interaktionsbasierter Usability-Evaluationsmethoden bereits Hinweismuster für Usability-Probleme angesprochen, die in Interaktionsabläufen erkannt werden können. Ein Teil dieser Muster bezieht sich lediglich auf fehlerhafte Ausführungen oder ist mit Fehlern seitens des Systems verbunden. Diese Hinweise können bei der Untersuchung von Prototypen nicht evaluiert werden. Manche dieser Muster betreffen allerdings wesentlich grundlegendere Gestaltungsprobleme und können dadurch zur Evaluation von Prototypen verwendet werden. Indem somit auch Usability-Aspekte einzelner Abläufe evaluiert werden, wird die Untersuchung ohne Vergleichswert unterstützt und ein weiterer Anwendungsfall für die Evaluation ermöglicht. Die in dieser Arbeit verwendeten Hinweismuster werden im Folgenden kurz vorgestellt.

**Zu viele Schichten:** Die Anwendung besitzt viele Fenster und der/die Benutzer\*in muss fünf oder mehr Fenster durchlaufen, um eine Aufgabe zu erledigen (Almeida, et al., 2015). Dieses Hinweismuster beeinflusst vor allem die Performance der Nutzer\*innen und erhöht ihre mentale Belastung, da sich vergangene und zukünftige Schritte gemerkt werden müssen (Ribeiro, et al., 2019).

**Entfernter Inhalt:** Auf einem Navigationsweg wird min. ein Fenster schnell überschritten, ohne dass eine tatsächliche Interaktion mit den Inhalten des Fensters stattfindet. Das Hinweismuster weist darauf hin, dass ein Navigationsweg zu lang sein könnte und beeinflusst sowohl die Performance der Nutzer\*innen und ihre mentale Belastung, als auch ihre Effizienz bei der Aufgabenbearbeitung. (Grigera, et al., 2016)

**Langes Anvisieren:** Bei der Bearbeitung einer Aufgabe müssen die Nutzer\*innen mit der Maus lange Wege zwischen Interaktionselementen zurücklegen. Dies äußert sich durch große Pointing-Zeiten und deutet auf eine aufwändige Bedienung hin. (Dahm, 2006)

**Viele Maus-Tastatur Wechsel:** Bei der Bearbeitung einer Aufgabe müssen die Nutzer\*innen häufig zwischen der Maus und der Tastatur wechseln. Dies äußert sich durch viele Homing-Operatoren, die die Effizienz der Nutzer\*innen beeinflussen. (Dahm, 2006)

**Hohe Website-Element-Abstände:** Die Elemente einer Webseite, die zur Erledigung einer Aufgabe notwendig sind, sollten möglichst nah aneinander positioniert werden. Wenn Nutzer\*innen weite Abstände zwischen Website-Elementen zurücklegen müssen, beeinflusst das die Effizienz der Aufgabenbearbeitung. Für die Evaluation dieses Hinweismusters wurden von Harms und Grabowski verschiedene Abstände ermittelt, die in Tabelle 1 aufgelistet sind. Mit Gleichung 2 wird ein Schweregrad berechnet. Liegt dieser Schweregrad über 0,5 gilt der Hinweis als gefunden. (Harms, et al., 2014)

Abstand $d(e_1, e_2)$	Beschreibung
0.0	$e_1$ und $e_2$ sind identisch oder haben denselben Zweck
0.2	$e_1$ und $e_2$ gehören zum selben Panel (div-Tag)
0.5	$e_1$ und $e_2$ gehören zur selben Seite
0.75	$e_1$ und $e_2$ gehören zur selben Webseite
1.0	$e_1$ und $e_2$ gehören zu verschiedenen Webseiten

Tabelle 1: Abstände Hinweismuster Hohe Website-Element-Abstände

Zur Ermittlung des Schweregrades wird die Gleichung 2 genutzt. Dazu werden die Website-Element-Abstände zwischen allen aufeinander folgenden Elementen einer Aufgabensequenz  $t'$  addiert und durch die Summe aller verwendeten Elemente minus Eins geteilt. (Harms, et al., 2014)

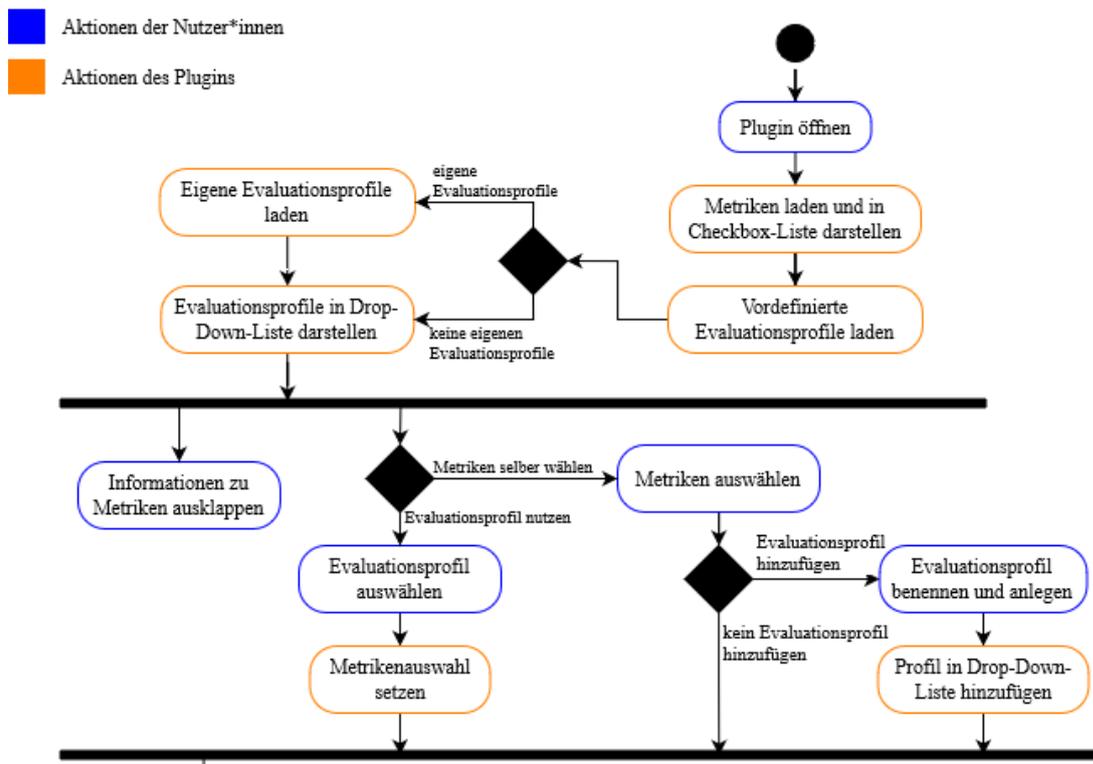
$$gd(t) = \frac{\sum_{t' \in t} \sum_{i=2}^{|E(t')|} d(e_{i-1}, e_i)}{\sum_{t' \in t} (|E(t')| - 1)} \quad (2)$$

## 6.5 Ablaufmodelle

Das Vorgehen bei der Nutzung des Plugins kann mit Hilfe einiger Ablaufmodelle erläutert werden.

### 6.5.1 Metriken-basierte Evaluation

Nachdem der/die Nutzer\*in das Plugin zur Metriken-Evaluation geöffnet hat, werden die hinterlegten Metriken und Evaluationsprofile geladen. Der/die Nutzer\*in kann sich weitere Informationen zu den einzelnen Metriken anzeigen lassen, eine Metriken-Auswahl treffen oder ein Evaluationsprofil zur Auswahl nutzen, wodurch das System die entsprechenden Metriken auswählt. Wird eine eigene Metriken-Auswahl getroffen, kann daraus optional ein Evaluationsprofil erstellt werden. Die Evaluation wird auf Frame-Basis durchgeführt, da sich ein Teil der Metriken (z.B. Vertikales Gleichgewicht) auf komplette Seiten bezieht oder die Verknüpfungen mehrerer Seiten evaluiert. Die Evaluation einzelner Elemente wird dadurch nicht beeinflusst. Nach der Metriken-Auswahl muss der/die Nutzer\*in deshalb eine Seite und anschließend mindestens einen der Frames dieser Seite auswählen. Nachdem die Evaluation gestartet wird, werden die zuvor ausgewählten Frames anhand der ebenfalls vorher ausgewählten Metriken evaluiert und das Evaluationsergebnis auf der Ergebnisseite dargestellt (mehr zur Ergebnisdarstellung s. Abschnitt 6.7.1).



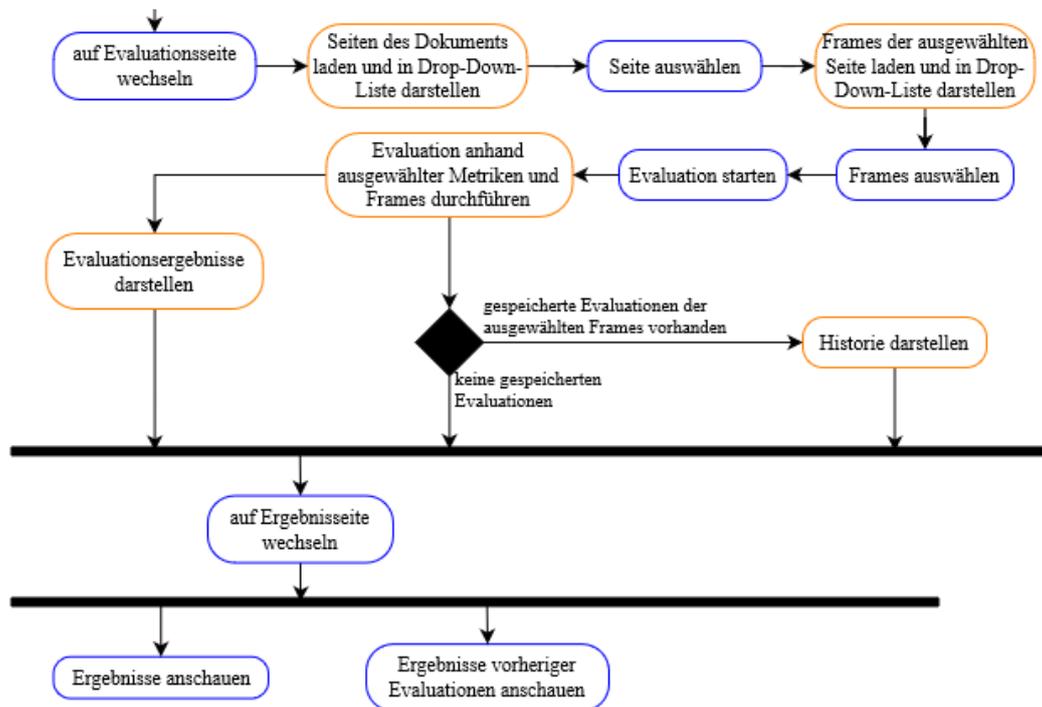


Abbildung 5: Ablaufmodell Metriken-Evaluation

### 6.5.2 Modell-basierte Evaluation

Bei der Modell-basierten Evaluation können Aufgaben und Szenarien definiert und evaluiert werden. Die Vorgehensweise bei der Aufgabendefinition wird in Abbildung 6 dargestellt. Wird das Plugin geöffnet und es wurden zuvor bereits Aufgaben definiert, werden sie direkt dargestellt. Anschließend kann der/die Nutzer\*in einen Aufgabennamen und die Plattform des Systems wählen und eine neue Aufgabe hinzufügen. Dabei wird kontrolliert, ob es bereits eine Aufgabe mit dem angegebenen Namen gibt. Ist dies der Fall, fordert eine Fehlermeldung den/die Nutzer\*in zur Eingabe eines anderen Namens auf. Wenn es keine Aufgabe mit dem Namen gibt, taucht eine neue Aufgabe in der Aufgabenliste auf. Anschließend können Bearbeitungsschritte hinzugefügt werden. Dazu wählt der/die Nutzer\*in ein UI-Element innerhalb des Entwurfs und einen Interaktionstypen aus einer Auswahlliste. Handelt es sich bei dem Typen um ein Eingabefeld, wird außerdem die Eingabe eines Beispiels verlangt. Wenn bereits Bearbeitungsschritte innerhalb der Aufgabe vorliegen, wird überprüft, ob das neue Element mit dem vorherigen Schritt kompatibel ist. Bearbeitungsschritte können auch in der Mitte oder am Anfang einer Aufgabe hinzugefügt werden, dann wird die Kompatibilität mit dem nachfolgenden bzw. dem nachfolgenden und dem vorherigen Schritt überprüft. Kompatible Schritte befinden sich entweder auf dem gleichen Frame oder können sich gegenseitig durch eine Interaktion erreichen. Wenn die Kompatibilität vorliegt wird der neue

Schritt hinzugefügt. Dabei wird kontrolliert, ob das Element einigen Usability-Vorgaben entspricht. Buttons und grafische Links müssen eine bestimmte Größe haben, Text-links sollten nicht über mehrere Zeilen gehen und ein Eingabefeld sollte groß genug für eine typische Eingabe sein (s. Richtlinien Abschnitt 6.1). Widerspricht das Element den Vorgaben, wird eine Warnung angezeigt, das Element aber trotzdem hinzugefügt. Dadurch soll der Blick der Gestalter\*innen für grundlegende Aspekte gebrauchstauglicher Interaktionselemente geschärft werden.

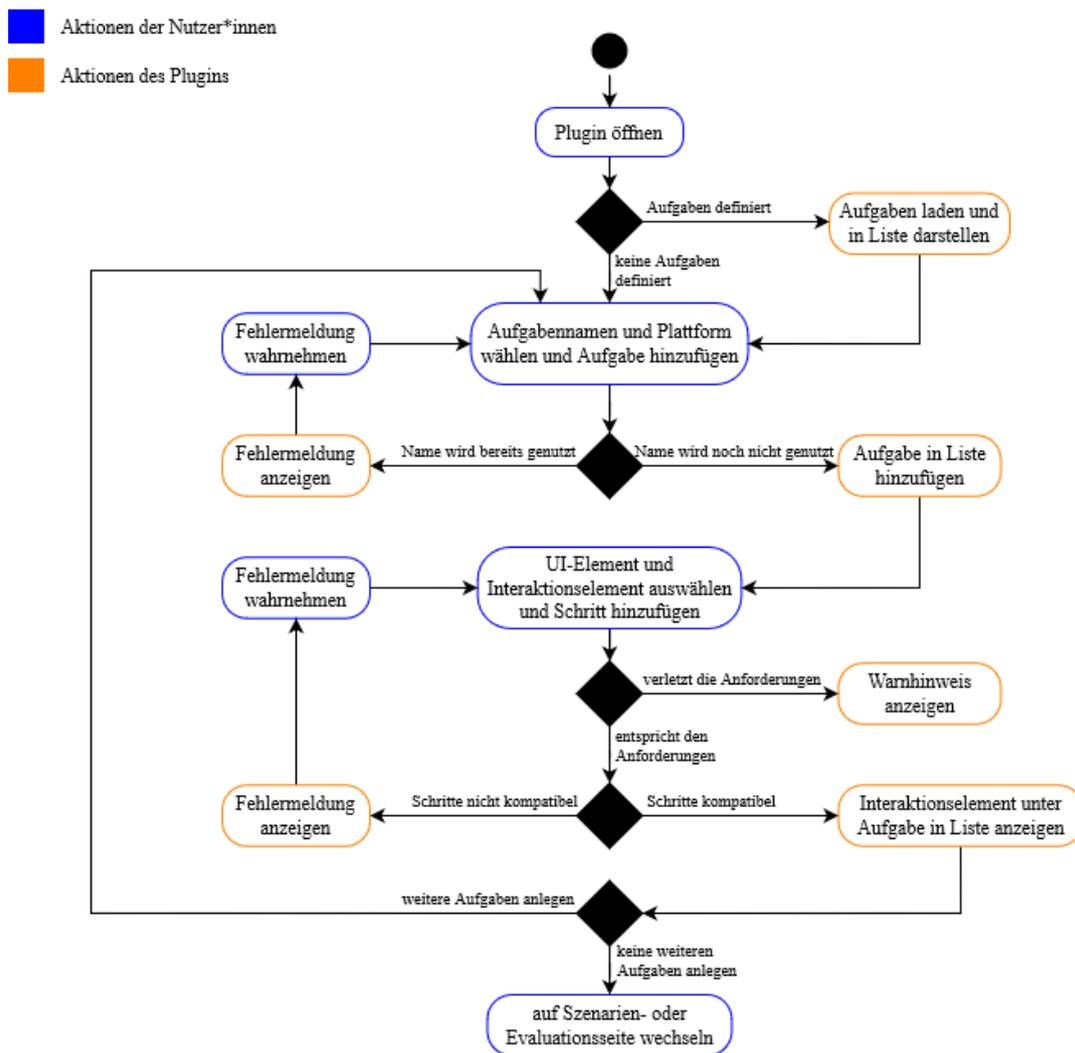


Abbildung 6: Ablaufmodell Aufgabendefinition

Die Szenarien-Definition (s. Abbildung 7) wird in der Regel dann durchgeführt, wenn zuvor mehrere Aufgaben definiert wurden. Falls bereits Szenarien angelegt wurden, werden diese geladen und in einer Liste dargestellt. Anschließend kann der/die Nutzer\*in einen Namen wählen und ein neues Szenario anlegen, welches einer Liste hinzugefügt wird, wenn noch

kein Szenario mit diesem Namen vorliegt. Anschließend können Aufgaben hinzugefügt werden. Hierbei erfolgt angelehnt an die Aufgabendefinition ebenfalls eine Kompatibilitätsprüfung, die anhand des letzten Schrittes der aktuell letzten Aufgabe und des ersten Schrittes der neuen Aufgabe durchgeführt wird. Falls die Aufgaben nicht direkt aufeinander folgen können, wird der/die Nutzer\*in aufgefordert, eine Übergangsaufgabe zu erstellen.

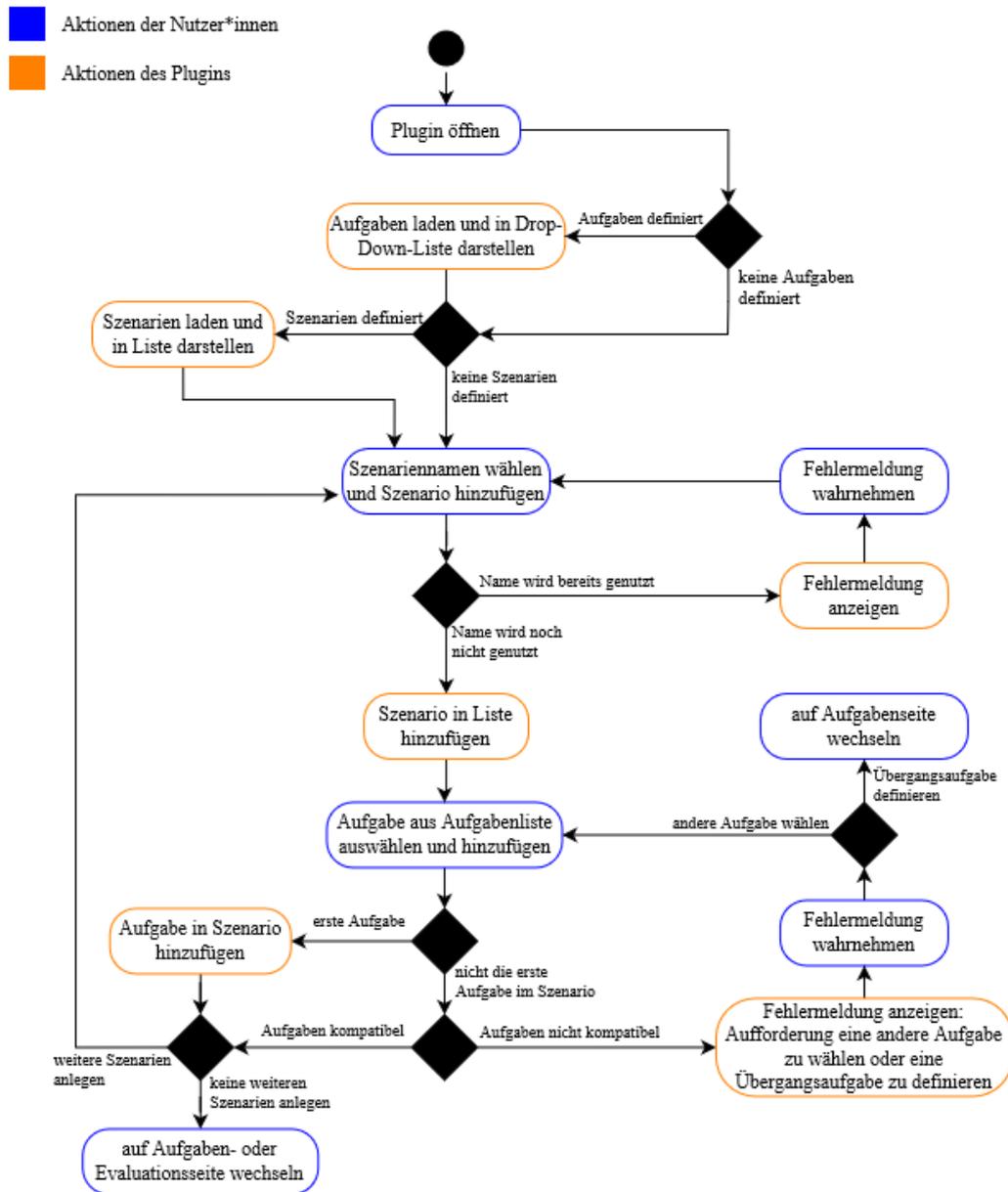


Abbildung 7: Ablaufmodell Szenarien-Definition

Die Evaluation (s. Abbildung 8) kann auf Aufgaben- oder Szenarien-Basis durchgeführt werden. Dazu können mindestens eine Aufgabe bzw. ein Szenario und maximal zwei Aufgaben bzw. zwei Szenarien gewählt werden. Bei einem Szenario werden die Aufgaben

separat evaluiert. Da das Hinweismuster *Entfernter Inhalt* übersehen werden kann, wenn es in einem Übergangsbereich zwischen zwei Aufgaben vorliegt, aber nur die Aufgaben an sich evaluiert werden, werden die Übergangsbereiche ebenfalls daraufhin untersucht. Falls zuvor bereits Evaluationen der aktuell evaluierten Frames, Aufgaben oder Szenarien vorliegen, werden diese in einer Historie dargestellt (mehr zur Ergebnisdarstellung s. Abschnitt 6.7.2).

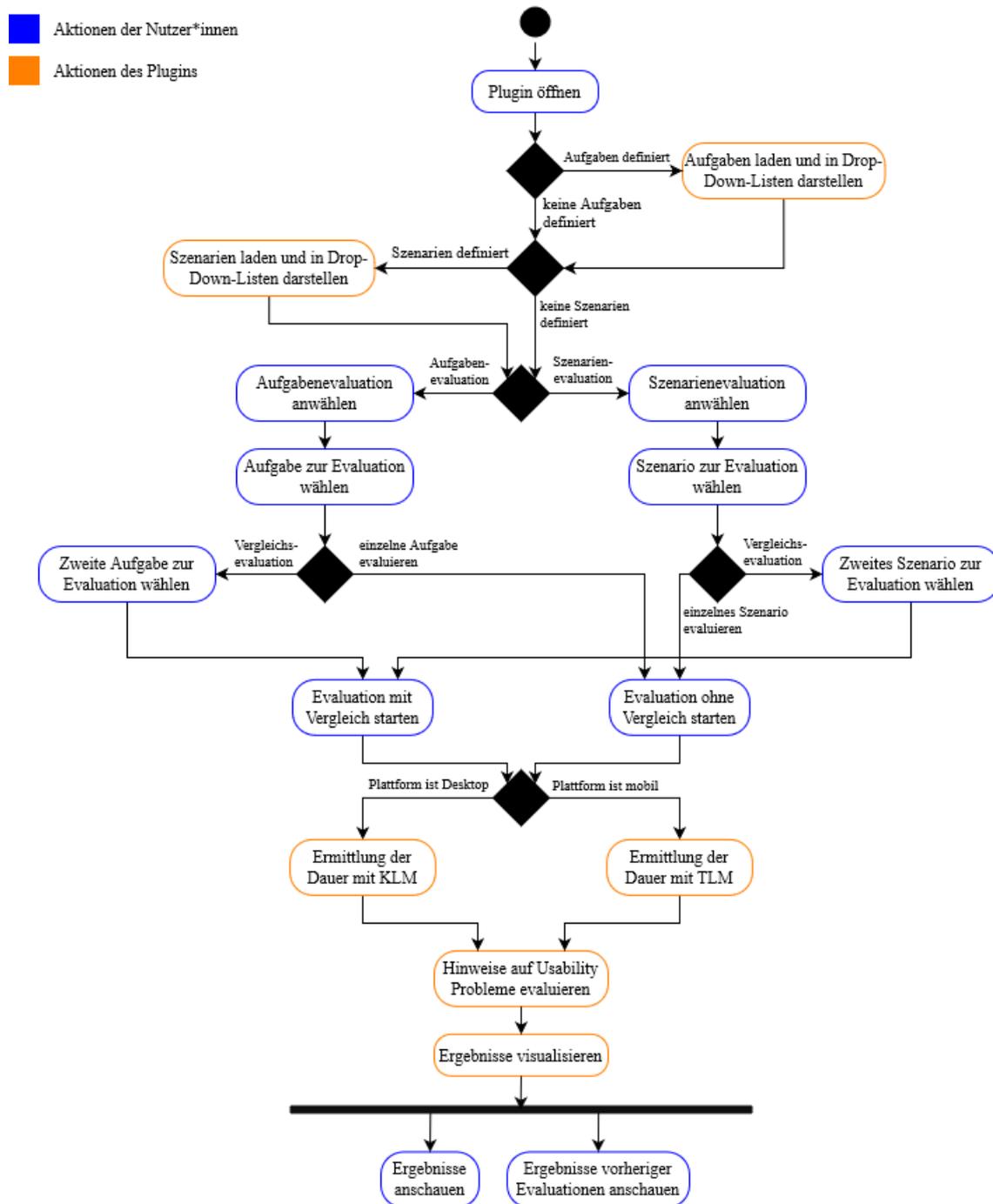


Abbildung 8: Ablaufmodell Aufgaben- bzw. Szenarien-Evaluation

## 6.6 Benutzungsoberfläche

Der grundlegende Aufbau der Benutzungsoberflächen der Metriken- und Modell-basierten Evaluation ist gleich. An der oberen Seite des Fensters befinden sich mehrere Reiter zur Navigation. Dabei werden die Metriken-Auswahl und die Aufgaben- bzw. Szenarien-Definition von der Evaluation und der Darstellung der Ergebnisse getrennt. Außerdem kann jeweils über ein Buch-Icon auf eine Dokumentation des Plugins zugegriffen werden. Die Benutzungsoberflächen können über eine gemeinsame Startseite des Plugins oder über einen eigenen Menüpunkt erreicht werden (s. Abschnitt 7.4).

### 6.6.1 Metriken-basierte Evaluation

Die erste Seite (s. Abbildung 9) dient der Metriken-Auswahl und kann über den Reiter *Metriken* erreicht werden. Den Hauptteil der Seite bildet eine Check-box-Liste zur Auswahl der Metriken. Über der Liste kann ein Evaluationsprofil gewählt werden, welches dann die Auswahl der Metriken übernimmt. Um die Benutzung zu erleichtern, kann über der Liste eine Checkbox geklickt werden, die alle Metriken an- bzw. abwählt. Unter der Metriken-Liste kann ein Evaluationsprofil aus der aktuellen Auswahl erstellt und bestehende Evaluationsprofile gelöscht werden.

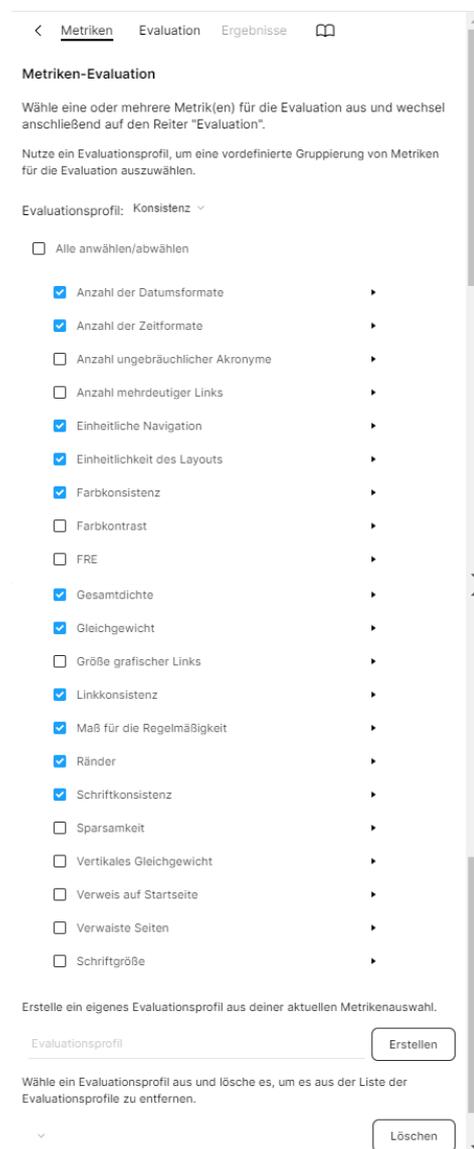


Abbildung 9: Plugin-Benutzungsoberfläche  
Metriken-Auswahl

Auf der Seite *Evaluation* (s. Abbildung 10) werden die Frames zur Evaluation gewählt. Die dritte Seite dient der Darstellung der Evaluationsergebnisse und kann über den Reiter *Ergebnisse* erreicht werden. Wenn neue Evaluationsergebnisse vorliegen, wird durch eine Notifikation an dem Reiter *Ergebnisse* darauf hingewiesen (mehr zur Darstellung der Ergebnisse s. Abschnitt 6.7.1).

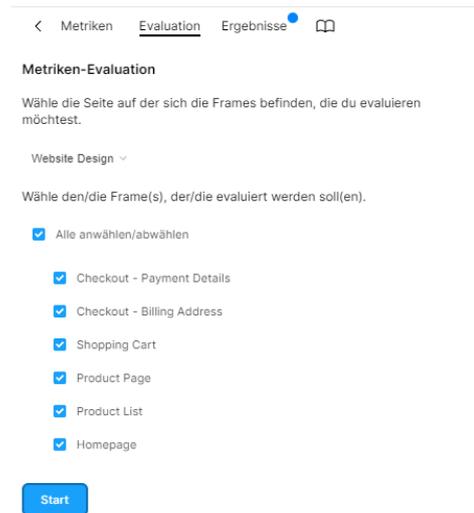


Abbildung 10: Plugin-Benutzeroberfläche - Frames-Auswahl

## 6.6.2 Modell-basierte Evaluation

Die erste Seite (s. Abbildung 11) dient der Definition von Aufgaben und kann über den Reiter *Aufgaben* erreicht werden. In der oberen Hälfte befinden sich die Funktionalitäten zum Anlegen einer Aufgabe, darunter werden bereits definierte Aufgaben mitsamt ihren Bearbeitungsschritten aufgelistet. Ein Bearbeitungsschritt beinhaltet eine Nummer und eine Beschreibung, bestehend aus dem Interaktionstypen und dem Namen des Interaktionselements. Im Bearbeitungsmodus (s. Abbildung 11) können Schritte gelöscht oder verschoben werden. Beim Verschieben wird stets eine Kompatibilitätskontrolle durchgeführt. Wenn Schritte in der neuen Reihenfolge nicht kompatibel sind, wird eine Fehlermeldung zurückgegeben. Des Weiteren können im Bearbeitungsmodus Schritte über das Plus innerhalb der Aufgabe hinzugefügt werden. Auch hier werden Kompatibilitätsprüfungen vorgenommen.

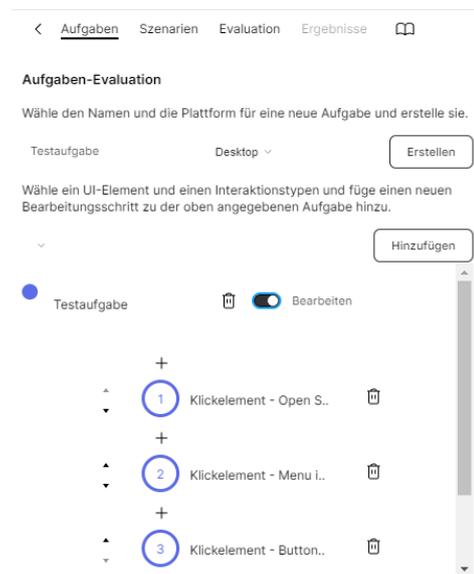


Abbildung 11: Plugin-Benutzeroberfläche - Aufgabenliste Bearbeitungsmodus

Auf der nächsten Seite (s. Abbildung 12) können Szenarien definiert werden. Diese Seite ist ähnlich aufgebaut wie die Seite zur Aufgabenerstellung. Im oberen Bereich befinden sich die Funktionalitäten zur Definition von Szenarien, die dann im unteren Bereich angezeigt werden.

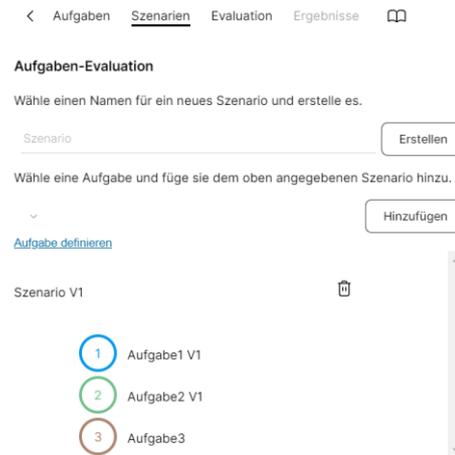


Abbildung 12: Plugin-Benutzeroberfläche - Szenarien-Definition

Die dritte Seite (s. Abbildung 13) dient der Auswahl der Aufgaben bzw. Szenarien zur Evaluation. Am unteren Ende der Seite kann die Evaluation gestartet werden. Wenn neue Evaluationsergebnisse vorliegen, wird durch eine Notifikation an dem Reiter *Ergebnisse* darauf hingewiesen (mehr zur Darstellung der Ergebnisse s. Abschnitt 6.7.2).

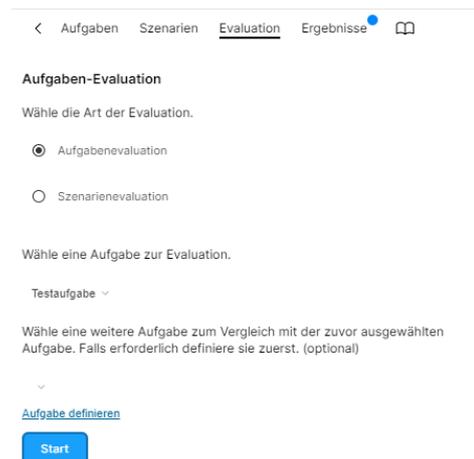


Abbildung 13: Plugin-Benutzeroberfläche - Aufgaben- bzw. Szenarien-Evaluation

## 6.7 Darstellung der Evaluationsergebnisse

### 6.7.1 Metriken-basierte Evaluation

Die Ergebnisse der Metriken-Evaluation werden in einer Auflistung der evaluierten Metriken dargestellt (s. Abbildung 14). Durch das Icon vor dem Namen einer Metrik wird dargestellt, ob Verstöße gefunden wurden und wie schwerwiegend diese sind. Die Farbe (gelb oder rot) wird dabei durch verschiedene Faktoren beeinflusst. Wenn eine Metrik einen Schwellenwert oder einen optimalen Bereich besitzt (s. A.3), wird ab der Überschreitung dieses Werts ein rotes Warnsymbol angezeigt. Wenn Verstöße vorliegen, aber das Ergebnis der Metrik noch nicht den Schwellenwert erreicht oder über- bzw. unterschritten hat, wird ein gelbes Warnsymbol dargestellt. Bei Metriken, die vorgeben, dass keine Verstöße vorliegen dürfen, wird bereits ab dem ersten Verstoß der rote Warnhinweis angezeigt. Unter den Namen der Metriken, zu denen Verstöße gefunden wurden, werden die problematischen Attribute bzw. die problematischen Elemente aufgelistet. Durch einen Klick wird das jeweilige Element oder alle Elemente mit dem problematischen Attribut innerhalb des Entwurfes ausgewählt. Falls eine Historie vorhanden ist, wird diese unter den Evaluationsergebnissen angezeigt. Die Darstellung entspricht dabei der Darstellung der aktuellen Evaluationsergebnisse. Da die einzelnen Ergebnisse allerdings bereits behoben sein könnten, sind die aufgelisteten Elemente und Attribute hier nicht klickbar.

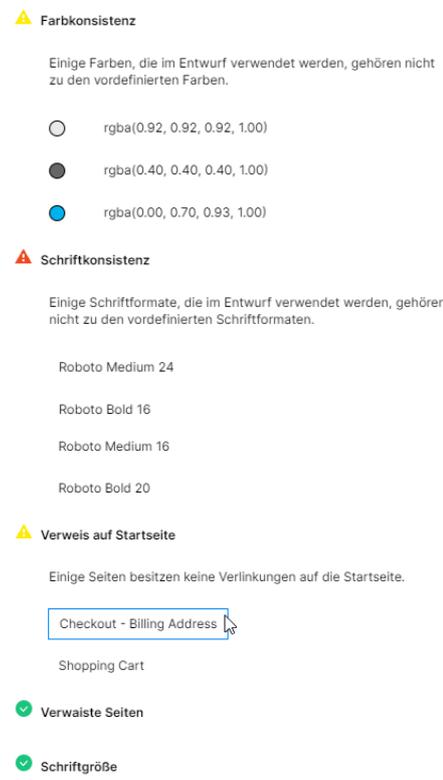


Abbildung 14: Ergebnis Metriken-Evaluation

### 6.7.2 Modell-basierte Evaluation

In Abschnitt 3.3 wurde ein Werkzeug zur empirischen Modellierung touchbasierter Benutzungsinteraktionen vorgestellt. Die im Rahmen einer visuellen Auswertung beschriebenen Balkendiagramme können auf die Ergebnisdarstellung einer Modell-basierten Evaluation anhand von Aufgaben und Szenarien übertragen werden. Diese Balkendiagramme dienen

vor allem als Orientierung für die visuelle Auswertung des Vergleichs zweier Aufgaben oder Szenarien und die Kombination von Aufgaben in einem Szenario.

Bei der Vergleichsevaluation zweier Aufgaben (s. Abbildung 15) wird ein Balkendiagramm dargestellt, innerhalb dessen die Ausführungszeiten der Aufgaben gegenübergestellt werden. Darunter werden diese Zeiten noch einmal aufgelistet, sodass der genaue Betrag erkannt werden kann. Der Klick auf einen der Balken öffnet die Detailansicht einer Aufgabe (s. nächster Absatz).

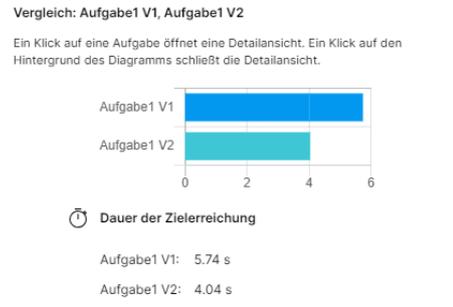


Abbildung 15: Ergebnis Aufgaben-Evaluation Vergleich

Bei der Evaluation einer einzelnen Aufgabe wird eine zeitliche Übersicht über die verschiedenen zur Bearbeitung der Aufgabe notwendigen Aktionen gegeben, die sich an den Operatoren des KLM orientiert (s. Abbildung 16). Die Operatoren werden hier unter anderem aufgrund des Platzmangels innerhalb des Graphens und der Zugehörigkeit zu dem hinterlegten Modell verwendet. Durch mehrfache Benutzung des Plugins und Einblick in die Dokumentation sollte ihre Bedeutung erlernt werden können. Darunter wird die Zeit zur Bearbeitung der Aufgabe dargestellt. Falls Hinweismuster auf Usability-Probleme gefunden wurden, werden sie unter der Darstellung der Bearbeitungsdauer aufgelistet. Falls sich das Hinweismuster auf einen bestimmten Bearbeitungsschritt bezieht, wird dieser neben einer Beschreibung des Problems und einem Hinweis zur Behebung hier angegeben.

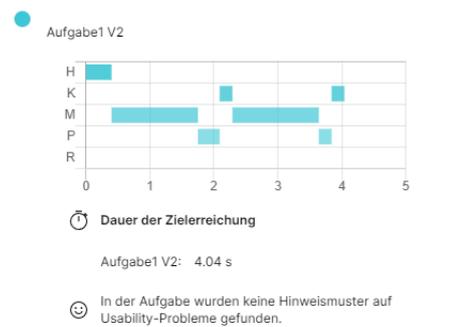


Abbildung 16: Ergebnis Aufgaben-Evaluation

Bei der Vergleichsevaluation zweier Szenarien wird ein Balkendiagramm dargestellt, welches die Ausführungsdauer der Szenarien direkt gegenüberstellt (s. Abbildung 17). Wie auch bei der Vergleichsevaluation zweier Aufgaben werden darunter die genauen Beträge der Dauern dargestellt. Innerhalb des Balkens eines Szenarios werden die Zeiten der einzelnen Aufgaben farblich, anhand der Aufgabenfarben

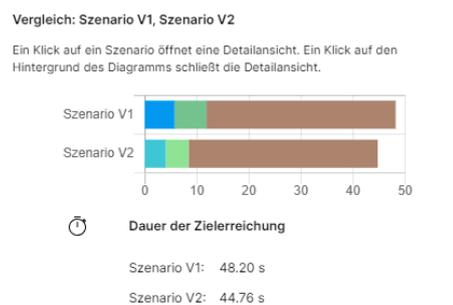


Abbildung 17: Ergebnis Szenarien-Evaluation Vergleich

unterschieden. Der Klick auf einen der Balken öffnet die Detailansicht eines Szenarios (s. nächster Absatz).

Bei der Evaluation eines Szenarios wird ein Balkendiagramm dargestellt, in dem die Bearbeitungsdauer des gesamten Szenarios visualisiert wird (s. Abbildung 18). Das Diagramm ähnelt dabei dem Diagramm der Vergleichsevaluation (s. Abbildung 17). Auch hier werden die Aufgaben farblich unterschieden. Der Klick auf den Balken einer Aufgabe öffnet darunter die Detailansicht der Aufgabenevaluation (s. Abbildung 16). Wenn keine Aufgabe ausgewählt ist, wird unter dem Balkendiagramm die Gesamtbearbeitungsdauer und falls vorhanden die in den Übergängen gefundenen Hinweismuster dargestellt.

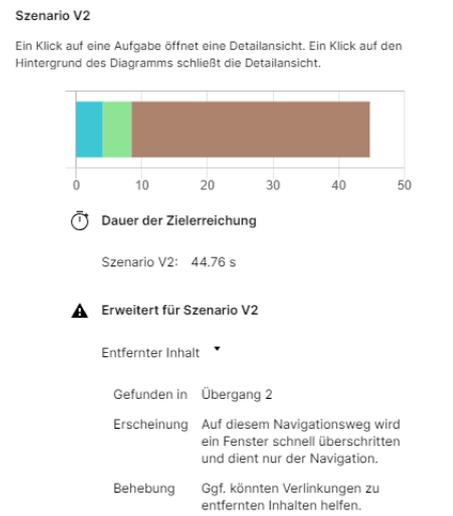


Abbildung 18: Ergebnis Szenarien-Evaluation

## 7 Implementierung

Im Folgenden werden einige ausgewählte Aspekte der Implementierung beschrieben. Dabei wird auf den Implementierungsrahmen eingegangen und der Aufbau des Plugins erläutert. Zur Implementierung der Benutzungsoberfläche wird die JavaScript-Bibliothek VueJS (You) verwendet. Um das Plugin möglichst einwandfrei in die Benutzungsoberfläche von Figma einbinden und eine konsistente Gestaltung fortführen zu können, wird das Design-System von Figma (Lowry) in Form einer VueJS-Bibliothek (Widua) über NPM (NPM Inc.) in das Projekt eingebunden und genutzt. Da das Plugin aus einer Vielzahl einzelnen Dateien besteht, wird es mit Hilfe von Webpack (Koppers) gebündelt.

### 7.1 Implementierungsrahmen

Aus der in Abschnitt 6.1 vorgestellten Metriken-Sammlung werden einige Maße zur Implementierung ausgewählt (*Farbkonsistenz*, *Schriftkonsistenz*, *Schriftgröße*, *Verwaiste Seiten*, *Verweis auf Startseite*). Die Auswahl beruht unter anderem darauf, dass die Implementierung einiger Metriken mit einem Mehraufwand verbunden ist, der im Hinblick auf die Beantwortung der Forschungsfrage nicht angemessen ist. Dazu zählen visuelle Metriken und Metriken, die die Inhalte von Textfeldern betreffen. Vor der Evaluation visueller Metriken muss, wie Trojahn es in ihrer Arbeit bereits beschrieben hat, eine Segmentierung der Benutzungsoberfläche erfolgen. Die Segmentierung kann entweder strukturiert, bildbasiert oder in einer Kombination der beiden Techniken erfolgen. Die strukturierte Segmentierung ist mit einigen Schwierigkeiten verbunden, da eine Abhängigkeit davon besteht, wie Gestalter\*innen die UI-Elemente des Entwurfs strukturieren. Aufgrund dessen schlägt Trojahn die Verwendung von bildbasierten Techniken mit Screenshots des Entwurfs oder eine Kombination der beiden Methoden vor. (Trojahn, 2020 S. 30f.) Des Weiteren wurden vor allem visuelle Metriken bereits zur Evaluation anderer Anwendungen implementiert. Die Evaluation des Farbkontrastes insbesondere findet sich bereits in Evaluationswerkzeugen für UI-Prototypen wie Stark oder A11y (s. Abschnitt 3.3).

Das Konzept der Modell-basierten Evaluation wird grundlegend implementiert. Es werden sowohl Funktionalitäten für Aufgaben als auch für Szenarien umgesetzt und jeweils die notwendigen Funktionen für eine separate und eine vergleichende Evaluation implementiert. Da keine empirisch ermittelten Zeitwerte für die KLM-ähnliche Repräsentation von Touch-Interaktionen gefunden werden konnten, wird dieser Aspekt nicht implementiert. Der

Implementierungsrahmen konzentriert sich auf die Evaluation von Entwürfen desktopbasierter Webanwendungen. Wie aus Abschnitt 5.2.5 entnommen werden kann, bietet Figma die Möglichkeit, Overlays zu erstellen, die über Interaktionen erreicht werden können. Aufgrund des mit der Evaluation solcher Overlays verbundenen Aufwands wird dieser Aspekt ebenfalls nicht implementiert. Zur Beantwortung der Forschungsfrage sollte die Evaluation einer Ebene vorerst ausreichen.

## 7.2 Aufbau des Plugins

In Abschnitt 5.2.3 wurde der grundlegende Aufbau von Figma-Plugins mit Benutzungsoberflächen bereits erläutert. Wie aus dem zuvor vorgestellten Konzept (s. Abschnitt 6) entnommen werden kann, besteht das in dieser Arbeit entwickelte Plugin aus zwei Teilen. Das Plugin wird dementsprechend aufgeteilt, indem innerhalb des Manifests, einer Datei zur Beschreibung des Plugins, zwei Menüpunkte definiert werden, die jeweils einem Befehl zugeordnet werden (s. Listing 1). Ein dritter Menüpunkt wählt eine gemeinsame Startseite aus.

```
"menu": [  
  {"name": "Startseite", "command": "start"},  
  {"name": "Metriken-Evaluation", "command": "metricEval"},  
  {"name": "Aufgaben-Evaluation", "command": "taskEval"}  
],
```

*Listing 1: Menüpunkte in Manifest*

Innerhalb des Figma-Editors werden die Menüpunkte unter dem Plugin-Namen dargestellt. Wird einer dieser Punkte ausgewählt, wird der entsprechende Befehl an den Plugin-Code weitergegeben. Dort erfolgt die Auswahl der entsprechenden Benutzungsoberfläche, indem auf die Definition innerhalb des Manifests (s. Listing 2) zugegriffen wird.

```
"ui": {  
  "start": "dist/start_ui.html",  
  "metricEval": "dist/metricEval_ui.html",  
  "taskEval": "dist/taskEval_ui.html"  
},
```

*Listing 2: Verweis auf Benutzungsoberflächen in Manifest*

Zur Separation der Funktionalitäten innerhalb des Plugin-Codes wurden Handler definiert. Ausgehend von der Benutzungsoberfläche werden Nachrichten an den Plugin-Code gesendet, die von dem entsprechenden Handler verarbeitet werden. Auf diese Weise werden unter

anderem das Laden verschiedener Speicherinhalte, die Evaluation der Entwurfsinhalte, Aufgaben und Szenarien, die Validierungen der UI-Elemente, der aufeinander folgenden Schritte und die Darstellung der Annotationen, sowie ihre Aktualisierung nach einer Änderung der Schritt-Reihenfolge angestoßen.

### 7.3 Benutzungsoberflächen

Die Struktur der Benutzungsoberfläche zur Metriken-Evaluation ist in Abbildung 19 dargestellt. Die Arrays *metricsArray* und *evaluationProfilesArray* beinhalten alle vordefinierten Metriken und Evaluationsprofile. Jedem Reiter der Benutzungsoberfläche (s. Abschnitt 6.6.1) wird eine Vue-Komponente zugeordnet (s. Abbildung 19 zweite Ebene). Ergänzend werden weitere Komponenten genutzt, um einzelne Bereiche wiederverwendbar und die Benutzungsoberfläche skalierbar zu machen. Für die Erstellung der Metriken-Checkbox-Liste wird bspw. das Metriken-Array durchgegangen und jeder Metrik eine Instanz der Komponente *Metric* zugeordnet. Neue Metriken müssen nur dem entsprechenden Array hinzugefügt werden und tauchen bei der nächsten Verwendung des Plugins automatisch in der Checkbox-Liste auf. Da die vordefinierten Metriken an unterschiedlichen Stellen dargestellt oder verwendet werden, ermöglicht die Nutzung eines separaten Arrays die Aufrechterhaltung der Aktualität an jeder dieser Stellen.

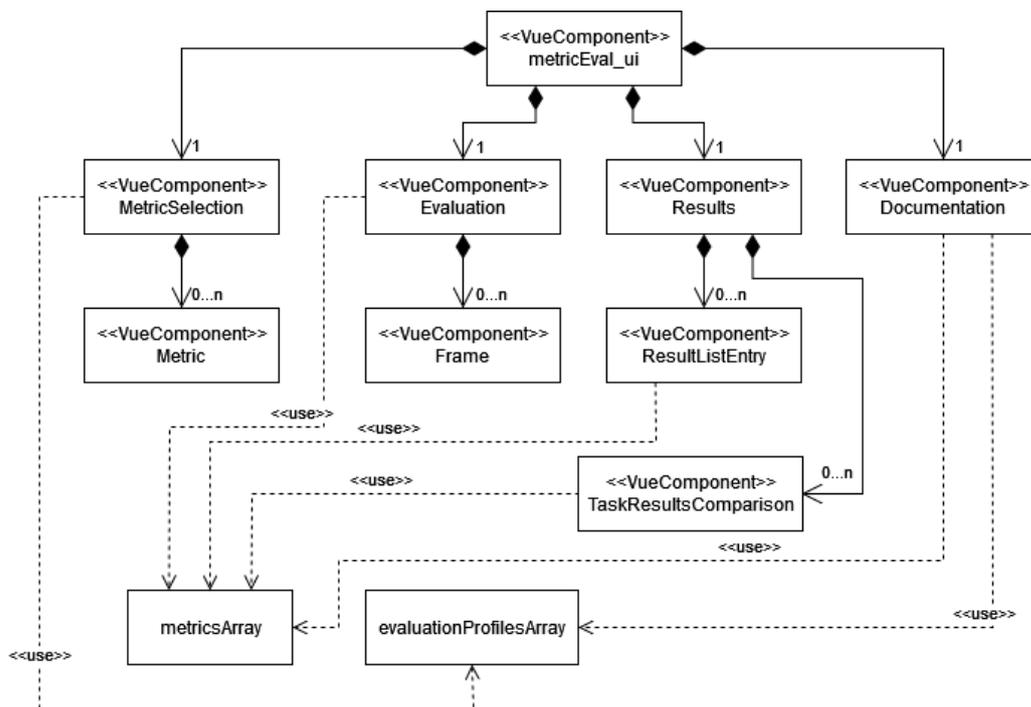


Abbildung 19: Code-Struktur Benutzungsoberfläche Metriken-Evaluation

Die Struktur der Benutzungsoberfläche der Modell-basierten Evaluation ist in Abbildung 20 dargestellt. Auch hier wird jeder Reiter des Plugins durch eine Komponente repräsentiert. Weitere Komponenten dienen hier vor allem der Wiederverwendbarkeit. Da die Hinweismuster für Usability-Probleme bspw. sowohl für Aufgaben als auch für Szenarien evaluiert werden, müssen sie jeweils in den entsprechenden Ergebnissen dargestellt werden können. Um dabei eine konsistente und dynamisch generierte Darstellung zu ermöglichen, die davon abhängt, ob ein Hinweismuster gefunden wurde, wird jeweils die Komponente *UsabilitySmellResult* verwendet. Bei der Ergebnisdarstellung werden Balkendiagramme verwendet, die auf der JavaScript Bibliothek Chart.js (Perkins, et al.) basieren.

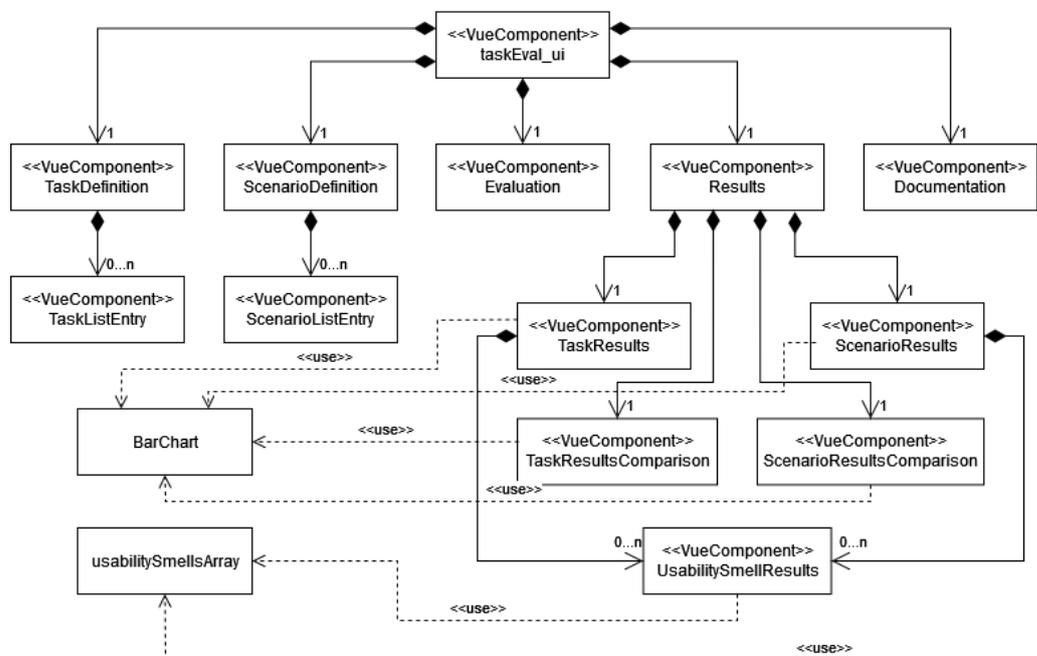


Abbildung 20: Code-Struktur Benutzungsoberfläche Aufgaben-Evaluation

## 7.4 Plugin-Code und Evaluation

Wie in Abschnitt 7.2 bereits angesprochen, kümmern sich einige Handler um die Entgegennahme und Ausführung unterschiedlicher Befehle der Benutzungsoberfläche. Die Nachrichten werden durch Message-Handler übermittelt. Das Senden einer Nachricht erfolgt über die Funktion *dispatch(<action>, <data>)*, das Empfangen über *handleEvent(<type>, <callback>)*. Um bspw. die Evaluation einer Aufgabe anzustoßen, wird eine Nachricht mit der Aktion *evaluateTask* und der zu evaluierenden Aufgabe als Data-Objekt übermittelt. Die Evaluation einer Vergleichsaufgabe wird durch die Aktion *evaluateTaskComparison* beschrieben. Auf Seiten des Handlers wird dann die Vorhersage zur Dauer der

Aufgabenbearbeitung per KLM ermittelt und die Hinweismuster für Usability-Probleme evaluiert. Die Metriken-Evaluation wird durch Übermittlung der Aktion *evaluate* und die ausgewählten Metriken und Frames gestartet. Auf Seiten des entsprechenden Handlers werden die ausgewählten Frames anhand der zur Evaluation gewählten Metriken untersucht. Die Funktionalitäten für die Evaluation und den Zugriff auf den Entwurf werden strukturiert, indem sie auf mehrere Helper aufgeteilt werden. Die Aufgaben-Evaluation benutzt Helper für die KLM-Berechnung, die Suche nach Hinweismustern für Usability-Probleme und die Validierung beim Hinzufügen neuer Schritte zu einer Aufgabe oder neuer Aufgaben zu einem Szenario (s. Abschnitt 6.5.2). Die Metriken-Evaluation verwendet einen Helper, in dem die Evaluation der Metriken implementiert ist. Außerdem werden komplexere Funktionalitäten für den Zugriff auf den Entwurf und die Erzeugung neuer UI-Elemente getrennt implementiert, sodass sie an unterschiedlichsten Stellen wiederverwendet werden können.

Unter anderem für die Darstellung vergangener Evaluationen und das Laden bereits definierter Aufgaben und Szenarien nach dem Öffnen des Plugins werden Daten auf dem Rechner des/der Nutzers/Nutzerin gespeichert. Listing 3 stellt bspw. den von der Benutzungsoberfläche ausgelösten Zugriff auf den Speicher vergangener Aufgaben-Evaluationen dar.

```
handleEvent('getTaskEvaluationStorage', async () => {
  var evaluationStorage = undefined;
  await figma.clientStorage.getAsync('taskEvaluation').then((value) => {
    evaluationStorage = value;
  });
  dispatch('currentTaskEvaluationStorage', evaluationStorage);
});
```

*Listing 3: Zugriff auf Speicher des/der Nutzers/Nutzerin*

## 8 Praktische Tests

Zur Untersuchung des Plugins wurden Tests im Labor *Software Ergonomie und interaktive Systeme* der Hochschule Düsseldorf (HSD) durchgeführt. Zur Durchführung der Tests haben sich sechs Personen bereit erklärt, die allesamt einen mediengestalterischen, medieninformatischen oder medientechnologischen Hintergrund haben. Da eine umfangreiche Erfahrung mit Figma nicht vorausgesetzt werden konnte, konnten sich die Testpersonen zu Beginn mit einer Anleitung (s. A.5) kurz in die Software einarbeiten. Den Test führten sie an einem mit einer Maus, einer Tastatur und einem externen Bildschirm ausgestatteten Rechner durch. Auf dem Rechner läuft der Desktop-Client von Figma. Das Plugin zur Usability-Evaluation ist vorinstalliert.

### 8.1 Aufgaben und Entwurf

Der Test beinhaltet zwei Aufgaben, die jeweils eine Herangehensweise zur Usability-Evaluation thematisieren (s. A.5). Die erste Aufgabe behandelt die Modell-basierte Evaluation und zielt auf eine evaluationsgestützte Entscheidung für einen von zwei Gestaltungsentwürfen sowie die Verbesserung des Entwurfs ab. Der dabei untersuchte Entwurf, der einen Teil eines Onlineshops darstellt, ist in Abbildung 21 dargestellt. Die Seiten *Product Page V1* und *Product Page V2* stellen die zu vergleichenden Gestaltungsalternativen dar. Daneben gibt es außerdem die Seite *Shopping Cart*, die einen Warenkorb repräsentiert und die Seite *Checkout* zur Eingabe der Adressinformationen im Bezahlvorgang. Innerhalb des Entwurfes können zwei der implementierten Hinweismuster erkannt werden. Die Seite *Shopping Cart* wird schnell überschritten, bei der Eingabe der Adressinformationen handelt es sich also um einen entfernten Inhalt und die Eingabe selbst benötigt verhältnismäßig viele Wechsel zwischen der Maus und der Tastatur. Vor dem Test wurden bereits drei Aufgaben für diesen Entwurf angelegt. Zwei davon beziehen sich auf die erste Variante des Entwurfs und beinhalten das

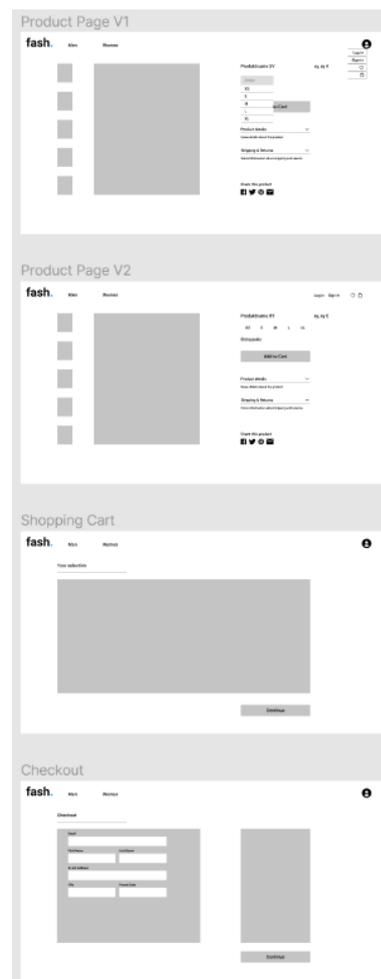


Abbildung 21: Entwurf zur Aufgaben-Evaluation

Hinzufügen des Produkts zum Warenkorb in der Größe M und die Navigation zur Seite *Checkout*. Die dritte Aufgabe bezieht sich auf die Eingabe der Adressinformationen. Die Test-Teilnehmer\*innen sollen nun zwei weitere Aufgaben hinzufügen, die das Hinzufügen des Produkts zum Warenkorb und die Navigation zum *Checkout* für den Gestaltungsentwurf V2 beinhalten. Aus diesen zwei Aufgaben und der Eingabe der Adressinformationen sollen sie anschließend ein Szenario erstellen. Im Anschluss daran sollen die Entwürfe verglichen werden. In der Art der Evaluation sind die Testpersonen freigestellt. Sie können einzelne Aufgaben oder Szenarien evaluieren, Aufgaben vergleichen oder ein weiteres Szenario erstellen und die beiden Szenarien vergleichen. Auf Basis der Evaluationsergebnisse sollen sie sich für den besseren Entwurf entscheiden und diesen falls nötig verbessern oder ihre Erkenntnisse über eventuelle Verbesserungen verbal beschreiben.

In der zweiten Aufgabe sollen die Testpersonen das Plugin zur Metriken-basierten Usability-Evaluation verwenden und den zweiten Beispielentwurf, dargestellt in Abbildung 22, evaluieren. Der Entwurf stellt eine ausgereiftere Version des zuvor bereits evaluierten Onlineshops dar. Die verwendeten Fotos stammen aus dem Figma-Plugin *Photos* (Sinani) und von der Webseite *Pixabay* (Pixabay GmbH). Innerhalb des Entwurfs

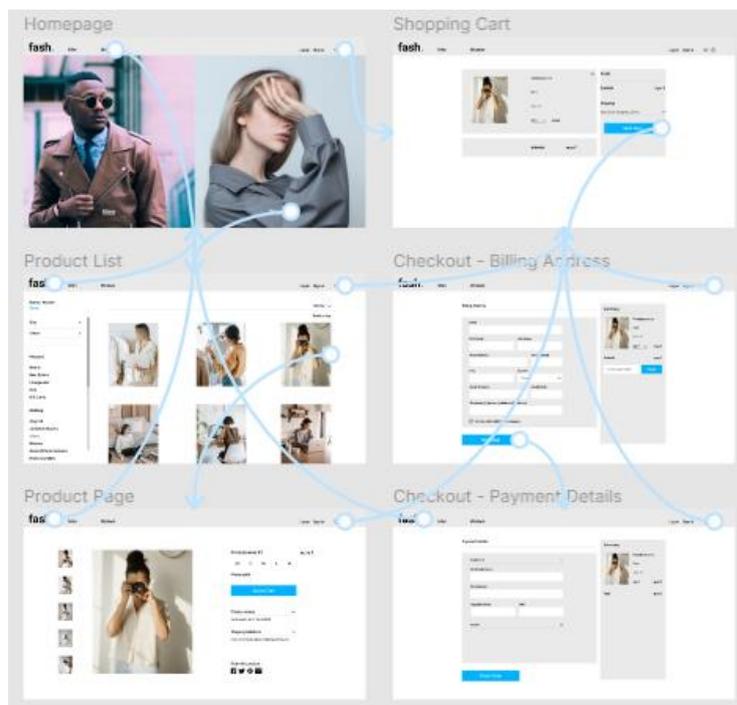


Abbildung 22: Entwurf zur Metriken-Evaluation

sind einige Fehler in Bezug auf die Farbkonsistenz und die Schriftkonsistenz eingebaut, sowie einige fehlende Referenzen auf die Startseite zu finden. Bei der Aufgabenbearbeitung sollte durch die Testpersonen erkannt werden können, dass der erste Entwurf hier auf den Seiten *Product Page*, *Shopping Cart* und (teilweise) *Checkout Billing Address* wiederzufinden ist. Durch diesen Zusammenhang der Entwürfe sollte verdeutlicht werden, dass und inwiefern die beiden Teile des Plugins unterschiedliche Bereiche der Gestaltung ansprechen und zusammen verwendet

werden können. Bei der Evaluation sind die Testpersonen auch hier freigestellt. Sie werden durch die Aufgabenstellung dazu aufgefordert, das entsprechende Plugin zu öffnen, anschließend aber dazu motiviert, verschiedene Dinge auszuprobieren. Auf Basis der Evaluationsergebnisse soll der Entwurf verbessert werden.

## **8.2 Fragebogen**

Im Anschluss an die Bearbeitung der Aufgaben sollen die Testpersonen einen Fragebogen (s. A.6) ausfüllen. Dadurch soll ein erster Überblick über die Akzeptanz und die Nützlichkeit des Plugins geschaffen werden. Der Fragebogen beginnt mit einer Frage dazu, ob die Testpersonen das Gefühl hatten, dass das Plugin sie bei der Evaluation zielführend unterstützen konnte. Anschließend wird eine Einschätzung der Testpersonen erfragt, inwiefern sie sich in der Evaluation und in der Gestaltung unterstützt gefühlt haben. Auf diese Aspekte wird separat eingegangen, da das Plugin zwar vorrangig zur Evaluation erstellt wurde, sich aber auch hilfreich, durch Verbesserungsempfehlungen o.ä., in den Gestaltungsprozess einbinden lassen soll. Im Anschluss daran wird separat erfragt, wie hilfreich die Testpersonen die Ergebnisse der Metriken- und der Aufgaben-Evaluation jeweils fanden. Die Nützlichkeit der beiden Teile wird separat erfragt, um zu verhindern, dass der Eindruck der einen Methode die Bewertung der anderen Evaluation beeinflusst. Abschließend wird nach der Wahrscheinlichkeit der Wiederverwendung und Weiterempfehlung des Plugins gefragt. Diese Fragen sollen bei der Einschätzung unterstützen, inwiefern die Testpersonen den Ergebnissen des Plugins vertrauen und denken, dass das Plugin die Arbeit eines/einer Gestalters/Gestalterin unterstützen kann.

## **8.3 Ergebnisse**

### **8.3.1 Auswertung Beobachtungen**

Bei der Beobachtung der Testpersonen sind verschiedene Dinge aufgefallen. Es wurden Aspekte und Probleme notiert, die aufgrund fehlender Figma-Kenntnisse oder der Formulierung der Aufgaben sowie einzelner Implementierungsfehler des Plugins entstanden sind. Außerdem wurden Aspekte und Probleme beobachtet, die mit der Benutzungsoberfläche und den Funktionalitäten des Plugins zusammenhängen und im Folgenden genauer betrachtet werden.

Bei der Aufgaben-Evaluation konnte beobachtet werden, dass die vorherige Auseinandersetzung mit der Dokumentation und den Beschriftungen des Plugins zu weniger Problemen führte. Außerdem schienen Fehlermeldungen größtenteils verständlich zu sein und das Mapping der Aufgabenfarben über das Plugin und den Entwurf hinweg bei der Identifikation und Wiederfindung einzelner Aufgaben und dadurch als Orientierung bei der Aufgabenerstellung zu unterstützen. Durch die selbstständige Behebung einiger Probleme der Teilnehmer\*innen konnten bereits erste Lernerfolge und Effizienzsteigerungen in der Nutzung beobachtet werden.

Im Anschluss an die erste Aufgabendefinition versuchten einige Testpersonen durch Doppelklick auf den Namen oder die Farbe die Aufgabe zu bearbeiten oder begannen damit, sich sowohl die Benutzungsoberfläche als auch das Plugin genauer anzuschauen, was darauf hindeutet, dass ihnen nicht direkt klar war, wie an dieser Stelle fortgefahren wird. Das Hinzufügen des ersten Arbeitsschrittes war bei manchen Testpersonen mit Problemen verbunden. Sie wählten weder ein Interaktionselement noch einen Interaktionstypen, nur ein Interaktionselement oder nur einen Interaktionstypen aus. Meist schien das Lesen der UI-Beschriftungen und der Fehlermeldungen dieses Problem zu beheben. Zum Hinzufügen eines Schrittes versetzten die Testpersonen die entsprechende Aufgabe fast ausnahmslos in den Bearbeitungsmodus. Eine Aufgabe beinhaltete das Hinzufügen eines Schrittes an den Anfang einer Interaktionsfolge (Aufgabe). Fast alle Teilnehmer\*innen nutzten dafür den Button *Hinzufügen*, der den Schritt ans untere Ende der Aufgabe anhängt. Als ihnen dies auffiel, verschoben sie den Schritt mit den Pfeiltasten nach oben. Das Plus zum Hinzufügen eines Schrittes wurde nur durch eine Testperson verwendet. Allerdings nutzte diese Person das Plus zum Anhängen eines Schrittes bei der Definition einer eigenen Aufgabe. Da der Schritt dort folglich falsch herum hinzugefügt wurde, zeigte die Benutzungsoberfläche eine Fehlermeldung an, die die fehlende Kompatibilität beschrieb. Diese Meldung konnte nicht bei der Behebung des Problems helfen. Beim Anlegen der Szenarien kam es zu keinen Problemen.

Im Anschluss an die Definition der Aufgaben und des Szenarios haben fast alle Teilnehmer\*innen mit der Evaluation eines Szenarios oder dem Vergleich zweier Szenarien begonnen. Die Evaluation eines einzelnen Szenarios schien meist wenig hilfreich zu sein. Durch die Hinzunahme eines Vergleichswerts wurde das Verständnis für die Ergebnisse sichtlich verbessert. Da die Testpersonen die Klickbarkeit der Ergebnisbalken zuerst nicht wahrnahmen, schienen sie bei der Betrachtung der Ergebnisse etwas orientierungslos. Die Interaktion

mit den Balken schien die Begeisterung für die Auswertung und Untersuchung der Ergebnisse sichtlich zu steigern. Grundsätzlich wurde die Vorhersage der Dauer zur Bearbeitung und die Darstellung der Balkendiagramme akzeptiert, die Herkunft der geschätzten Zeit und der Zweck der Operatoren schienen aber nicht direkt verständlich.

Nahezu alle Testpersonen schauten sich die Rückmeldungen des Systems auf die Hinweismuster für Usability-Probleme zwar an und erkannten, dass die Eingabe der Adressinformationen viele Wechsel zwischen der Maus und der Tastatur benötigt, nahmen aber keine Veränderung der Benutzungsoberfläche vor. Das Hinweismuster, welches innerhalb der Szenarien gefunden werden konnte und die Warenkorb-Seite bei der Navigation als ggf. hinderlich herausstellt, wurde nur von einer kleinen Gruppe der Teilnehmer\*innen genauer untersucht. Zwei der Testpersonen schlugen eine Verbesserung in Form eines Schnellzugriffs auf die Seite *Checkout* vor. Bei der Betrachtung der Ergebnisse merkte eine der Testpersonen an, dass sie sich wundere, dass ein Szenario länger dauere, aber diesem Unterschied scheinbar kein Usability-Problem zugrunde liege.

Bei der Verwendung des Plugins zur Metriken-Evaluation schien die Betrachtung der Erläuterungen der Metriken dazu zu führen, dass die Ergebnisse besser verstanden wurden. Eine der Personen, die sich diese Erläuterungen nicht anschaute, wünschte sich bei der Ergebnisdarstellung ebenso eine Rückmeldung für die erfolgreich evaluierten Metriken, wie sie auch bei den gefunden Verstößen gegeben wurde. Ein Teil der Testpersonen schien nach der Auswahl der Metriken zu denken, dass sie ein Evaluationsprofil anlegen müssten, welches anschließend bei der Evaluation gewählt werden könne. Zur Evaluation wählte ein Großteil der Teilnehmer\*innen alle Metriken und Frames aus. Alle Testpersonen nutzten die Auswahl der Elemente durch Klicken auf die Ergebnisse und untersuchten damit den Entwurf. Ein Testteilnehmer merkte an, dass die gemeinsame Auswahl vieler einzelner Elemente unübersichtlich werden könne und wünschte sich eine visuelle Separation. Ein anderer äußerte, dass er es hilfreich fände bei der Farb- und Schriftkonsistenz direkt in den Ergebnissen erkennen zu können, wie viele Elemente das entsprechende Attribut besitzen. Nach der Untersuchung der Ergebnisse haben einige der Testpersonen den Entwurf verbessert und erneut evaluiert, um ihre Verbesserung zu bestätigen. Dabei wurden sie nach eigenen Angaben dadurch motiviert, dass sie bei allen Metriken einen grünen Haken haben wollten.

Allgemein konnte bei der Arbeit mit dem Plugin beobachtet werden, dass die Historie bei der Aufgaben-Evaluation weniger benutzt wurde. Sie wurde meist geöffnet, kurz zur

Kenntnis genommen und anschließend wieder geschlossen. Bei der Metriken-basierten Evaluation wurde sie dagegen etwas genauer betrachtet und die aktuelle Evaluation mit vergangenen Evaluationen verglichen. Vermutlich ist die geringe Nutzung der Historie bei der Aufgaben-Evaluation in der Aufgabenstellung begründet. Der Fokus lag dabei mehr auf dem Erkennen problematischer Aspekte. Wäre eine umfangreiche Verbesserung vieler Aspekte des Entwurfs vorgenommen worden, wie es in der zweiten Aufgabe möglich war, wäre eventuell auch die Historie verstärkt genutzt worden.

### 8.3.2 Auswertung Fragebogen

Im Folgenden wird eine Auswertung der Fragebögen vorgenommen. Da lediglich eine kleine Anzahl an Testpersonen teilgenommen hat und darunter Personen waren, die keine oder nur geringe Erfahrungen im Bereich des UI- bzw. UX-Designs haben, sind diese Ergebnisse nicht aussagekräftig für die Zielgruppe, sie können aber eine erste Einschätzung des Plugins unterstützen. Auf die Frage, ob die Testpersonen das Gefühl hatten, dass sie das Plugin bei der Evaluation der Entwürfe zielführend unterstützen konnte, haben alle Testpersonen einstimmig mit *Ja* geantwortet.

Bei der Auswertung der Fragen 2 und 3 ist zu erkennen, dass sich die Teilnehmer\*innen bei der Evaluation stärker unterstützt gefühlt haben als bei der Gestaltung. Da das Plugin vor allem auf eine Evaluation abzielt, ist dies verständlich. Die trotzdem größtenteils positive Rückmeldung in Bezug auf die Gestaltung lässt darauf schließen, dass die Empfehlungen zur Verbesserung anregend waren.

Generell wurden die Ergebnisse beider Evaluationen größtenteils als hilfreich angesehen. Eine der Testpersonen schrieb allerdings in Bezug auf die Aufgaben-Evaluation, dass ihm nicht klar war, wie die Werte helfen sollten und er sich fragte, welchen Wert er im Idealfall erreichen wollte.

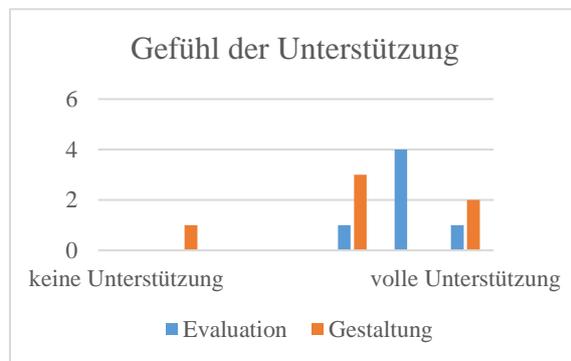


Abbildung 23: Auswertung Fragebogen Gefühl der Unterstützung



Abbildung 24: Auswertung Fragebogen Gefühl der Nützlichkeit der Ergebnisse

Bei allen Teilnehmer\*innen ist die Wahrscheinlichkeit der Weiterempfehlung des Plugins relativ hoch, was für ein Vertrauen in die Evaluationsergebnisse spricht. Auch die Bereitschaft zur erneuten Verwendung des Plugins schien hoch. Eine der Testpersonen, die einen geringeren Wert ausgewählt hat, begründete dies damit, dass sie sonst nicht im UI- bzw. UX-Bereich tätig ist.

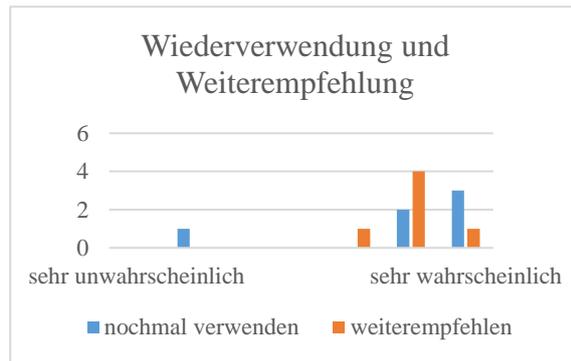


Abbildung 25: Auswertung Fragebogen Wiederverwendung und Weiterempfehlung

## 9 Fazit

Das Ziel dieser Masterarbeit war es, zu untersuchen, wie Gestalter\*innen digitaler Benutzungsoberflächen bei der Entwicklung von Gestaltungslösungen im Hinblick auf die Gewährleistung bzw. Verbesserung der Usability durch ein automatisiertes Werkzeug zeit- und kostensparend unterstützt werden können. Dazu wurde eine Literaturrecherche bestehender Ansätze zur automatisierten Usability-Evaluation durchgeführt, sowie die Aufgaben und Vorgehensweisen von UI-Gestalter\*innen im Rahmen einer Nutzungskontextanalyse untersucht und Interviews zur Ermittlung der Anforderungen an ein automatisiertes Werkzeug geführt, welches anschließend konzeptioniert, implementiert und mit einer kleinen Personengruppe getestet wurde. Aufgrund der geringen Anzahl der Teilnehmenden können keine allgemeingültigen Schlussfolgerungen gezogen werden. Da es sich aber um eine qualitative Untersuchung handelte, konnte die Beobachtung der Testpersonen dennoch wertvolle Ergebnisse liefern.

Die Literaturrecherche hat gezeigt, dass im Bereich automatisierter Usability-Evaluationen grundsätzlich drei Herangehensweisen unterschieden werden: interaktionsbasiert, Metrikenbasiert, Modellbasiert. Durch eine Untersuchung der verschiedenen Charakteristiken von Prototypen unter dem Gesichtspunkt des Forschungsziels konnte ermittelt werden, dass sowohl eine Metriken- als auch eine Modellbasierte Evaluation geeignet sind. Beide wurden für das Konzept des Werkzeugs genutzt.

Für die Modellbasierte Evaluation von Prototypen wurde das Keystroke Level Model der GOMS-Familie gewählt, welches interaktionsnahe Untersuchungen ermöglicht. Durch einige vereinfachende Annahmen ist das Modell nur bedingt aussagekräftig über die tatsächliche Performance und die komplexen Gedankengänge menschlicher Nutzer\*innen (John, et al., 1994). Die Durchführung der Tests hat allerdings gezeigt, dass die Vorhersage der Bearbeitungsdauer mit dem KLM beim Vergleich mehrerer Alternativen eine Grundlage zur Entscheidungshilfe bieten kann. Die Vorhersage der Bearbeitungszeiten schien bei der Evaluation einzelner Aufgaben/Szenarien weniger zielführend zu sein und die Verwendung verschiedener Hinweismuster für Usability-Probleme dabei nur bedingt Abhilfe zu schaffen. Hierbei stellt sich die Frage, ob die Untersuchung der Hinweismuster generell einen eher kleinen Nutzen bietet oder die Auswahl der Muster und die Formulierung der Empfehlungen

zur Verbesserung, die sich ursprünglich auf voll funktionsfähige Anwendungen bezogen, nur bedingt für die Evaluation von Prototypen geeignet sind.

Neben der Modell-basierten wurde ebenfalls eine Metriken-basierte Evaluation implementiert. Die Untersuchung weit verbreiteter Usability-Richtlinien aus dem Webbereich hat ergeben, dass diese sich nur bedingt für die Evaluation von UI-Prototypen eignen, weshalb nur eine sehr kleine Auswahl dieser Richtlinien genutzt werden konnte. Stattdessen wurde größtenteils auf einer bestehenden Metriken-Sammlung aufgebaut. Im Unterschied zu bisher kommerziell eingesetzten Lösungen (Plugins) zur Prüfung verschiedener Aspekte der Barrierefreiheit von Prototypen, beinhaltet diese Metriken-Sammlung navigationsbezogene Aspekte wie einen Startseitenverweis und die Untersuchung verwaister Seiten. Nach dem Testen des entwickelten Plugins stellt sich allerdings die Frage, wie groß der Mehrwert dieser Metriken im Prototyping-Bereich tatsächlich ist. Bei der Entwicklung von Prototypen geht es zu einem großen Teil vor allem um die visuelle Darstellung. Aspekte wie die Verlinkungen auf die Startseite sind allerdings von der tatsächlichen Implementierung abhängig und sollten spätestens zu diesem Zeitpunkt sowieso getestet werden.

Zusammenfassend lässt sich sagen, dass die Ergebnisse dieser Arbeit und die Evaluation des entwickelten Plugins im Rahmen praktischer Tests zeigen konnten, dass eine automatisierte Evaluation mit Metriken- und Modell-basierten Herangehensweisen die Verbesserung der Usability von Prototypen unterstützen kann. Mit der Modell-basierten Evaluation wurde ein Untersuchungsaspekt genutzt, der in diesem Rahmen von Prototypen bislang weniger Beachtung fand. Hinsichtlich der durchgeführten Tests schien das Werkzeug sowohl in der Evaluation als auch der Gestaltung unterstützen zu können. Die Arbeit mit der Modell-basierten Evaluation schien dabei etwas aufwändiger zu sein, eine Verbesserung der Effizienz bei mehrmaliger Nutzung scheint aber realistisch. Von diesem Standpunkt aus kann von einer Einsparung der Zeit- und Ressourcenkosten ausgegangen werden, vor allem im Hinblick auf die gesamte Produktentwicklung. Die Kombination einer Metriken- und Modell-basierten Evaluation kann bereits einen größeren Bereich der Aspekte abdecken, die einen UI-Prototypen ausmachen, indem sowohl visuelle als auch grundlegend funktionale Usability-Probleme erkannt werden können und Interaktionsabläufe, die maßgeblich die Gestaltung der Benutzungsoberfläche beeinflussen, früh zeit- und kosteneffizient evaluiert werden können. Die Bewertung der tatsächlichen Einsetzbarkeit, Akzeptanz und Effektivität dieses Konzeptes bedarf allerdings weiterer Untersuchungen.

## 10 Ausblick

Um tatsächlich herauszufinden, inwiefern sich das Werkzeug angenehm in den Arbeitsablauf einbinden lässt und dabei keinen negativen Einfluss auf die Effektivität, Effizienz und Zufriedenheit der Nutzer\*innen hat, sowie die Ergebnisse als Grundlage für Gespräche mit anderen entwickelnden Personen dienen können, müssten im nächsten Schritt Feldversuche durchgeführt werden, in denen Testpersonen das Plugin über eine längere Zeit hinweg verwenden. Dabei wäre es interessant zu beobachten, inwiefern die Nutzer\*innen tatsächlich beide Teile des Plugins verwenden.

In Bezug auf die Modell-basierte Evaluation wäre ein nächster Schritt die Unterstützung für Mobilgeräte zu implementieren, bspw. durch Nutzung des Touch Level Models. Dafür müssten allerdings vorerst die entsprechenden Zeitwerte der Operatoren empirisch ermittelt werden. Die Ermittlung weiterer Hinweismuster, die sich besonders auf die Usability von Prototypen beziehen, könnte die Modell-basierte Evaluation einzelner Aufgaben oder Szenarien aussagekräftiger machen und eine angemessenere Gestaltung der Verbesserungsvorschläge ermöglichen. Eventuell könnte eine tiefgreifendere Analyse, der mit dem aufgefundenen Hinweismuster in Zusammenhang stehenden UI-Elemente die Ermittlung maßgeschneiderter Empfehlungen zur Behebung der Usability-Probleme ermöglichen.

Auch im Bereich der Metriken-basierten Evaluation bieten sich Weiterentwicklungen an. Zuallererst könnten die bislang fehlenden Metriken implementiert werden und ihr Nutzen in Bezug auf Prototypen evaluiert werden. Dabei sollte außerdem durch Tests oder Feldversuche untersucht werden, inwiefern eine größere Menge von Metriken die Nutzung des Plugins und vor allem die Arbeit der Gestalter\*innen beeinflusst und welche Metriken tatsächlich zur Evaluation genutzt werden. Eine große scrollbare Liste könnte insbesondere bei der Erstbenutzung des Plugins erschlagend wirken und die Bereitschaft zur Verwendung des Plugins reduzieren.

Eine der interviewten Personen beschrieb die Verwendung von Design-Systemen zur Bereitstellung geprüfter Komponenten. Da an dieser Stelle sowieso eine gewisse Evaluation durchgeführt wird, erscheint die Verwendung des Plugins hier ebenfalls interessant. Dazu könnte in einem nächsten Schritt untersucht werden, ob das Plugin dazu in der aktuellen Version geeignet ist oder ob Anpassungen vorgenommen werden müssen und wie diese aussehen.

## Quellenverzeichnis

- Adee App Inc.** Adee. [Online] [Zitat vom: 9. Februar 2022.] <https://www.adeeapp.com/>.
- Adobe Inc.** Adobe XD. *Adobe*. [Online] [Zitat vom: 28. Januar 2022.] <https://www.adobe.com/de/products/xd.html>.
- Almeida, Diogo, et al. 2015.** Towards a Catalog of Usability Smells. *SAC '15: Proceedings of the 30th Annual ACM Symposium on Applied Computing*. April 2015. S. 175-181.
- Attention Insight. 2021.** Attention Insight. [Online] 2021. [Zitat vom: 9. Februar 2022.] <https://attentioninsight.com/>.
- Bakaev, Maxim, Mamysheva, Tamara und Gaedke, Martin. 2016.** Current Trends in Automating Usability Evaluation of Websites. *2016 11th International Forum on Strategic Technology (IFOST)*. 2016. S. 510-514.
- Balbo, Sandrine, et al. 2005.** Leading Web Usability Evaluations to WAUTER. *AusWeb05: Australasian World Wide Web Conference*. 2005.
- Berger, Daniel. 2022.** Figma: Das beste Design-Tool für Website-Prototypen in der Praxis. *heise online*. [Online] 26. Januar 2022. [Zitat vom: 20. Februar 2022.] <https://www.heise.de/ratgeber/Figma-Das-beste-Design-Tool-fuer-Website-Prototypen-6337596.html>.
- Blackmon, Marilyn Hughes, et al. 2002.** Cognitive Walkthrough for the Web. *CHI '02: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. April 2002. S. 463-470.
- Bowman, Jordan und Palmer, Taylor. 2021.** 2021 Design Tools Survey. *UX Tools*. [Online] 2021. [Zitat vom: 20. Februar 2022.] <https://uxtools.co/survey-2021/>.
- Canolli, Chava. 2021.** How Do UX and UI Designers Work Together. *CareerFoundry*. [Online] 5. August 2021. [Zitat vom: 23. Mai 2022.] <https://careerfoundry.com/en/blog/ux-design/how-ux-and-ui-designers-use-style-and-functional-guidelines-to-collaborate/>.
- Chi, Ed H., et al. 2003.** The Bloodhound Project: Automating Discovery of Web Usability Issues using the InfoScent Simulator. *CHI '03: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. April 2003. S. 505-512.

**Dahm, Markus. 2006.** *Grundlagen der Mensch-Computer-Interaktion*. München : Pearson Studium, 2006. ISBN 3-8273-7175-9.

**DeepL. 2021.** DeepL Übersetzer. [Online] 7. Dezember 2021. Ergebnis der Übersetzung des Wortes "Usability" ins Deutsche. <https://www.deepl.com/translator#en/de/usability>.

**DIN e.V. 2020.** DIN EN ISO 9241. *Ergonomie der Mensch-System-Interaktion*. 2020. Deutsche Fassung.

—. **2018.** DIN EN ISO 9241-11. *Ergonomie der Mensch-System-Interaktion - Teil 11: Gebrauchstauglichkeit: Begriffe und Konzepte*. 2018. Deutsche Fassung.

—. **2020.** DIN EN ISO 9241-210. *Ergonomie der Mensch-System-Interaktion - Teil 210: Menschzentrierte Gestaltung interaktiver Systeme*. 2020. Deutsche Fassung.

**Dingli, Alexiei und Cassar, Sarah. 2014.** An Intelligent Framework for Website Usability. *Advances in Human-Computer Interaction*. 2014. S. 1-13.

**Dingli, Alexiei und Mifsud, Justin. 2011.** USEful: A Framework to Mainstream Web Site Usability through Automated Evaluation. *International Journal of Recent Trends in Human Computer Interaction (IJHCI)*. März/April 2011. S. 10-30.

**Feuerstack, Sebastian, et al. 2008.** Automated Usability Evaluation during Model-Based Interactive System Development. [Hrsg.] Springer Berlin Heidelberg. *Engineering Interactive Systems*. 2008. 5247, S. 134-141.

**Figma Inc.** API Reference. *Figma Plugins*. [Online] [Zitat vom: 22. Mai 2022.] <https://www.figma.com/plugin-docs/api/api-reference/>.

—. Explore Design files. *Figma Help Center*. [Online] [Zitat vom: 27. Februar 2022.] <https://help.figma.com/hc/en-us/sections/4403928559639-Explore-Design-files>.

—. FigJam. *Figma*. [Online] [Zitat vom: 18. Mai 2022.] <https://www.figma.com/figjam/>.

—. Figma. *Figma*. [Online] [Zitat vom: 18. Mai 2022.] <https://www.figma.com/>.

—. Guide to prototyping in Figma. *Figma Help Center*. [Online] [Zitat vom: 22. Mai 2022.] <https://help.figma.com/hc/en-us/articles/360040314193-Guide-to-prototyping-in-Figma>.

—. Introduction. *Figma Plugins*. [Online] [Zitat vom: 22. Mai 2022.] <https://www.figma.com/plugin-docs/intro/>.

—. Pricing. *Figma*. [Online] [Zitat vom: 18. Mai 2022.] <https://www.figma.com/pricing/>.

**Goguey, Alix, et al. 2018.** Storyboard-Based Empirical Modelling of Touch Interface Performance. *CHI '18: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. April 2018. S. 1-12.

**Grigera, Julián, et al. 2016.** Automatic Detection of Usability Smells in Web Applications. *International Journal of Human-Computer Studies*. 2016. S. 129-148.

**Guarino de Vasconcelos, Leandro und Baldochi Jr., Laércio Augusto. 2012.** Towards an automatic evaluation of Web applications. *SAC '12: Proceedings of the 27th Annual ACM Symposium on Applied Computing*. 2012. S. 709-716.

**Hak, Jean-Luc, Winckler, Marco und Navarre, David. 2016.** PANDA: Prototyping using ANnotations and Decision Analysis. *EICS '16: Proceedings of the 8th ACM SIGCHI Symposium of Engineering Interactive Computing Systems*. Juni 2016. S. 171-176.

**Hamacher, Nicolas Alexander. 2006.** Automatische kriterienorientierte Bewertung der Gebrauchstauglichkeit interaktiver Systeme. Aachen : s.n., 2006. ISBN 3-89963-432-2.

**Harms, Patrick und Grabowski, Jens. 2014.** Usage-Based Automatic Detection of Usability Smells. [Hrsg.] Springer Berlin Heidelberg. *Human-Centered Software Engineering*. 2014. S. 217-234.

**Ivory, Melody Y. und Hearst, Marti A. 2001.** The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys*. Dezember 2001. Bd. 33, S. 470-516.

**Ivory, Melody Y. und Hearst, Marti A. 2000.** Web TANGO: towards automated comparison of information-centric web site designs. *CHI EA '00: CHI '00 Extended Abstracts on Human Factors in Computing Systems*. April 2000. S. 329-330.

**Jacobsen, Jens und Meyer, Lorena. 2018.** *Praxisbuch Usability und UX*. Bonn : Rheinwerk Verlag, 2018. ISBN 978-3-8362-4423-7.

**John, Bonnie E. und Kieras, David E. 1994.** The GOMS Family of Analysis Techniques: Tools for Design and Evaluation. 1994.

**Justinmind.** *Justinmind*. [Online] [Zitat vom: 28. Januar 2022.] <https://www.justinmind.com/>.

**Khasnis, Shubhangi S., et al. 2019.** Analysis of automation in the field of Usability Evaluation. *2019 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE)*. 2019. S. 85-91.

**Koppers, Tobias.** Webpack. *Webpack*. [Online] [Zitat vom: 24. Mai 2022.] <https://webpack.js.org/>.

**Lavery, D., Cockton, G. und Atkinson, M. P. 1997.** Comparison of evaluation methods using structured usability problem reports. [Hrsg.] Taylor & Francis. *Behaviour & Information Technology*. 1997. Bd. 16, S. 246-266.

**Loceye. 2020.** VisualEyes. [Online] 2020. [Zitat vom: 9. Februar 2022.] <https://www.visualeyes.design/>.

**Lowry, Thomas.** Figma Plugin DS - A lightweight UI library for creating Figma plugins. *GitHub*. [Online] [Zitat vom: 6. Mai 2022.] <https://github.com/thomas-lowry/figma-plugin-ds>.

**Mahajan, Rohit und Shneiderman, Ben. 1997.** Visual and Textual Consistency Checking Tools for Graphical User Interfaces. *IEEE Transactions on Software Engineering*. November 1997. Bd. 23, S. 722-735.

**Marenkov, Jevgeni, Robal, Tarmo und Kalja, Ahto. 2016.** A Framework for Improving Web Application User Interfaces Through Immediate Evaluation. *Databases and Information Systems IX*. 2016. Bd. 291.

—. **2018.** Guideliner - a Tool to Improve Web UI Development for Better Usability. *WIMS '18: Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*. Juni 2018. S. 1-9.

**Meta Platforms.** ReactJS. *ReactJS*. [Online] [Zitat vom: 24. Mai 2022.] <https://reactjs.org/>.

**Ngo, David Chek Ling und Byrne, John G. 2001.** Another look at a model for evaluating interface aesthetics. *International Journal of Applied Mathematics and Computer Science*. 2001. Bd. 11, S. 515-535.

**Nielsen, Jakob. 2020.** 10 Usability Heuristics for User Interface Design. *Nielsen Norman Group*. [Online] 15. November 2020. [Zitat vom: 27. November 2021.] <https://www.nngroup.com/articles/ten-usability-heuristics/>.

**NPM Inc.** NPM. *NPM*. [Online] [Zitat vom: 21. Mai 2022.] <https://www.npmjs.com/>.

**Padda, Harkirat Kaur. 2003.** QUIM map: a repository for usability/quality in use measurement. 2003.

**Perkins, Derek, et al.** Chart.js. *Chart.js*. [Online] [Zitat vom: 19. Mai 2022.] <https://www.chartjs.org/>.

**Pixabay GmbH.** Pixabay. *Pixabay*. [Online] [Zitat vom: 19. Mai 2022.] <https://pixabay.com/de/>.

**Preim, Bernhard und Dachsel, Raimund. 2010.** *Interaktive Systeme Band 1 Grundlagen, Graphical User Interfaces, Informationsvisualisierung*. Berlin Heidelberg : Springer-Verlag, 2010. ISBN 978-3-642-05401-3.

**Ribeiro, Rafael Fontinele, et al. 2019.** Usability Problems Discovery Based on the Automatic Detection of Usability Smells. *SAC '19: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. April 2019. S. 2328-2335.

**Rice, Andrew D. und Lartigue, Jonathan W. 2014.** Touch-Level Model (TLM): Evolving KLM-GOMS for Touchscreen and Mobile Devices. *ACM SE '14: Proceedings of the 2014 ACM Southeast Regional Conference*. März 2014. S. 1-6.

**Richter, Michael und Flückiger, Markus. 2013.** *Usability Engineering kompakt*. Berlin Heidelberg : Springer-Verlag GmbH, 2013. ISBN 978-3-642-34831-0.

—. **2016.** *Usability und UX kompakt*. Berlin Heidelberg : Springer-Verlag GmbH, 2016.

**Robal, Tarmo, Marenkov, Jevgeni und Kalja, Ahto. 2017.** Ontology Design for Automatic Evaluation of Web User Interface Usability. *2017 Portland International Conference on Management of Engineering and Technology (PICMET)*. 2017.

**Sarodnik, Florian und Brau, Henning. 2016.** *Methoden der Usability Evaluation: wissenschaftliche Grundlagen und praktische Anwendung*. Bern : Hogrefe Verlag, 2016. ISBN 978-3-456-85597-4.

**Shneiderman, Ben.** The Eight Golden Rules of Interface Design. *University of Maryland - Ben Shneiderman*. [Online] [Zitat vom: 16. Februar 2022.] <https://www.cs.umd.edu/users/ben/goldenrules.html>.

**Sinani, Emin.** Photos. *Figma Community*. [Online] [Zitat vom: 19. Mai 2022.] <https://www.figma.com/community/plugin/745428728979155171/Photos>.

**Sketch B.V.** Sketch. *Sketch*. [Online] [Zitat vom: 18. Mai 2022.] <https://www.sketch.com/>.

**Sonntag, Martin. 2015.** Shneiderman-Regeln - Die 8 goldenen Regeln des Interface Designs. *Erwartungskonforme Webseiten*. [Online] 2015. [Zitat vom: 28. Januar 2022.] <http://www.informatik.uni-oldenburg.de/~iug15/ew/usability/shneiderman-regeln/>.

**Stapelkamp, Torsten. 2010.** *Interaction- und Interfacedesign*. Berlin Heidelberg : Springer-Verlag GmbH, 2010.

**Stark Lab Inc. 2022.** Introducing Stark. [Online] 2022. [Zitat vom: 9. Februar 2022.] <https://www.getstark.co/>.

**Stübling, Jörg. 2018.** *Qualitative Sozialforschung: Eine komprimierte Einführung*. 2. Berlin/Boston : De Gruyter, 2018. ISBN 978-3-11-052991-3.

**The Webby Awards. 2022.** *The Webby Awards - Honoring the Best of the Internet*. [Online] 2022. [Zitat vom: 4. April 2022.] <https://www.webbyawards.com/>.

**Tiedtke, Thomas, Martin, Christian und Gerth, Norbert. 2001.** AWUSA - A Tool for Automated Website Usability Analysis. *PreProceedings of the 9th International Workshop on the Design, Specification and Verification of Interactive Systems*. 2001.

**Trojahn, Franziska. 2020.** Improving the Usability of UI Prototypes by Incorporating an Automated Evaluation in the Design Process. November 2020.

**U.S. Department of Health & Human Services. 2006.** Research-Based Web Design & Usability Guidelines. 2006.

**UsabilityHub Pty. Ltd.** UsabilityHub. *UsabilityHub*. [Online] [Zitat vom: 18. Mai 2022.] <https://usabilityhub.com/>.

**Useberry.** Useberry. *Useberry*. [Online] [Zitat vom: 18. Mai 2022.] <https://www.useberry.com/>.

**Vanderdonckt, Jean, Beirekdar, Abdo und Noirhomme-Fraiture, Monique. 2004.** Automated Evaluation of Web Usability and Accessibility by Guideline Review. [Hrsg.] Springer Berlin Heidelberg. *Web Engineering*. 2004. S. 17-30.

**Vargas, Ariel, Weffers, Harold und Vieira da Rocha, Heloísa. 2010.** A Method for Remote and Semi-Automatic Usability Evaluation of Web-based Applications Through Users Behavior Analysis. *MB '10: Proceedings of the 7th International Conference on Methods and Techniques in Behavioral*. August 2010. S. 1-5.

**W3C. 2018.** Web Content Accessibility Guidelines (WCAG) 2.1. *W3C Recommendation*. [Online] 05. Juni 2018. [Zitat vom: 27. Januar 2022.] <https://www.w3.org/TR/WCAG21/>.

**Widua, Alexander.** figma-plugin-ds-vue. *NPM*. [Online] [Zitat vom: 6. Mai 2022.] <https://www.npmjs.com/package/figma-plugin-ds-vue>.

**You, Evan.** The Progressiv JavaScript Framework. *Vue.js*. [Online] [Zitat vom: 23. Mai 2022.] <https://vuejs.org/>.

## A. Anhang

### A.1. Werkzeuge zur automatisierten Usability-Evaluation

#### Interaktionsbasierte Werkzeuge

Das Werkzeug *Awusa* beschreibt einen Aufgaben-basierten Ansatz, der eine statische und eine dynamische Analyse durchführt, um Usability-Probleme zu erkennen. Die statische Analyse, bestehend aus einer Code- und einer Objektstrukturanalyse, soll Syntaxfehler und Bereiche erkennen, die nicht mit Codierungsstandards übereinstimmen, und anhand der Anzahl unterschiedlicher UI-Elemente allgemeine visuelle Metriken ermitteln. Bei der dynamischen Analyse werden reale Nutzungsdaten aus Protokolldateien analysiert, mit Nutzungsmustern verglichen, die auf Usability-Probleme hinweisen, und auf erfolgreiche oder fehlerhafte Bearbeitung einer Aufgabe hin untersucht. Abschließend werden die zuvor definierten Strukturen und die Analyseergebnisse interaktiv visualisiert, um die evaluierenden Personen bei der Interpretation der Ergebnisse zu unterstützen. (Tiedtke, et al., 2001)

Bei *Wauter* handelt es sich um eine, an der University of Melbourne entwickelte Werkzeug-Suite zur automatisierten Usability-Analyse. Die Usability-Analyse mit *Wauter* basiert auf dem Vergleich der beabsichtigten Nutzung mit der tatsächlichen Nutzung einer Website. Dazu werden sowohl ein Modell der beabsichtigten Nutzung in Form eines Aufgabenmodells erstellt, indem eine Fachperson die Schritte aller möglichen Pfade zur Erledigung einer Aufgabe durchführt, und die tatsächliche Nutzung durch Testpersonen oder Endbenutzer aufgezeichnet. Der Vergleich erfolgt in Übereinstimmung mit vordefinierten Heuristiken, die potenzielle Usability-Probleme identifizieren. Die Heuristiken beinhalten Richtungswechsel, Sofortiges Abbrechen einer Handlung, Wiederholungen von Handlungen, irrelevante Handlungen und zeitliche Aspekte. Abschließend unterstützt die *Wauter*-Suite die Analyse der Ergebnisse durch Visualisierungen und erzeugt interaktive Sitemaps mit Videoaufnahmen der Aufgabenbearbeitung. (Balbo, et al., 2005)

*WebHint* beschreibt eine teilweise implementierte Methode zur automatischen Usability-Evaluation webbasierter Anwendungen, bestehend aus drei Schritten. Die Evaluation der Gebrauchstauglichkeit basiert dabei auf dem Vergleich der beabsichtigten und der tatsächlichen Nutzung. Dazu werden sowohl die Nutzungsinteraktion einer Fachperson als auch die Interaktionen von Testpersonen oder Endbenutzern aufgezeichnet und in Logdateien gespeichert. Für die Evaluation werden die Protokolldateien der Nutzer vorverarbeitet, indem die

Bearbeitungsschritte der einzelnen Testpersonen extrahiert und anschließend für jede Aufgabe geclustert zusammengefasst werden. Dadurch werden die häufigsten Ausführungsmuster identifiziert. Diese Muster werden im nächsten Schritt mit den Sequenzen der beabsichtigten Nutzung verglichen und abschließend durch Hinzunahme von Mustern konkreter Usability-Probleme analysiert. (Vargas, et al., 2010)

*Usabilics*, ein an der Universidade Federal de Itajubá entwickeltes Werkzeug zur automatischen Usability-Evaluation, nutzt den Vergleich der beabsichtigten mit der tatsächlichen Nutzung von Web-Anwendungen. Die Aufzeichnung der Referenz erfolgt, indem eine Fachperson die Nutzungsschnittstelle zur Bearbeitung einer Aufgabe nutzt. Die Autoren definieren ein sog. COP-Schnittstellenmodell, welches Benutzungsschnittstellen in Container, Objekte und Seiten (Pages) strukturiert und bei der Aufzeichnung der beabsichtigten Nutzung verwendet wird, um Spezialisierungen oder Generalisierungen der Seiten, Container oder Objekte zu ermöglichen. Bei der Analyse orientieren sie sich an der Definition der Usability und verwenden einen gewichteten Ähnlichkeitswert (Gewichtung anhand korrekter Seite, korrektem Container und korrektem Objekt) zwischen der erwarteten und der tatsächlichen Nutzung als Maß für die Effizienz, den Prozentsatz der Vollständigkeit der Aufgabe als Maß für die Effektivität und Daten über das Betriebssystem, den Browser und die Verbindungsgeschwindigkeit für den Nutzungskontext. Auf Basis des gewichteten Ähnlichkeitswertes bilden sie einen Usability Index, der als Maß für die Gebrauchstauglichkeit der Website dient. Anschließend schlägt das System je nach Art des Fehlers Verbesserungsmöglichkeiten vor. (Guarino de Vasconcelos, et al., 2012)

### **Metriken-basierte Werkzeuge**

*Sherlock* ist ein automatisches Analysewerkzeug zur Evaluation von Windows-Benutzungsoberflächen, das eine aufgabenunabhängige Konsistenzprüfung innerhalb der Benutzungsoberfläche oder über mehrere Benutzungsoberflächen hinweg durchführt. Zur Auswertung werden Windows-Benutzungsoberflächen in spezielle kanonische Formatdateien übersetzt, die einen organisierten Satz von GUI-Objektbeschreibungen enthalten. Dazu wurden Übersetzer entwickelt, die mit Visual Basic 3.0 und Visual C++ 4.0 verwendet werden können. *Sherlock* ist sequenziell modular aufgebaut und besitzt Subsysteme, die visuelle Eigenschaften von Dialogfeldern, die Terminologie sowie die Größe und Beschriftung von Schaltflächen bewerten. Nach Abschluss der Analyse erhalten die Anwender Rückgaben in Form von Tabellen, die einen Überblick über das Design der Dialogboxen, fehlerhafte Ränder und

Vorschläge zur Verbesserung dieser, verwendete Wörter in Beschriftungen und Nachrichten, das Design von Buttons, falsch geschriebene Begriffe und verwendete Terminologien beinhalten. (Ivory, et al., 2001) (Mahajan, et al., 1997)

Bei *Kwaresmi* handelt es sich um ein Werkzeug zur automatisierten Usability-Evaluation anhand von Richtlinien. Das Werkzeug basiert auf der Trennung der Logik zur Bewertung der Leitlinien von der Bewertungsmaschine, weshalb der Evaluierungsprozess in die zwei Phasen Spezifikation formaler Richtlinien und Bewertung von Webseiten unterteilt wird. Die Spezifikation der Richtlinien erfolgt in Form einer XML-konformen Sprache, die von den Autoren als Guideline Definitions Language (GDL) bezeichnet wird. Zur Evaluation wird eine Webseite heruntergeladen und durch einen Parser erfasst, wodurch ihr Inhalt und ihre Struktur identifiziert werden können. Anschließend werden die Bewertungsbedingungen der Richtlinien durch eine Prüfung der Eigenschaften der entsprechenden Instanzen evaluiert. (Vanderdonckt, et al., 2004)

*USEful* ist ein Werkzeug zur automatisierten Evaluation der Usability von Webseiten anhand von Richtlinien. Zur Evaluation wird eine Sammlung von 240 Richtlinien verschiedener Kategorien (u.a. UX-Optimierung, Navigation, Texterscheinung und Grafiken, Bilder and Multimedia) verwendet und innerhalb einer Datenbank gespeichert. Dazu werden Datenbanktabellen verwendet, die die Usability-Kategorien, die Implementationsstufen, die Richtlinien-Definitionen und die Evaluationsregeln beinhalten. Zur Evaluation werden die Richtlinien aus der Datenbank geladen und nacheinander untersucht. Dazu werden die Bedingungen der Evaluationsregeln, mit der zuvor durch einen Parser erfassten und in HTML- und CSS-Dateien hinterlegten Webseite geprüft, indem die zu untersuchenden Attribute der verschiedenen UI-Elemente betrachtet werden. (Dingli, et al., 2011)

Das Werkzeug *Guideliner* wurde entwickelt, um die Evaluation von Webseiten bereits während der Erstellung durch einen Entwickler durchführen zu können. Dazu wird für die Entwicklungsumgebung ein spezieller Hook gesetzt, der das Evaluationswerkzeug aufruft, wenn Änderungen an der Benutzungsoberfläche vorgenommen wurden. Beim Start des Werkzeugs wird eine Konfigurationsdatei geladen, die den Zugriff auf ein Repository bestimmt, in welchem die zur Evaluation verwendbaren Richtlinien gespeichert sind. In einer früheren Version von *Guideliner* wurden die Richtlinien mit Hilfe einer Metasprache definiert. Da die Metasprache allerdings laut den Autoren nur bedingt dazu in der Lage war, die Richtlinien auszudrücken, wurde die Definition der Richtlinien auf Ontologien umgestellt.

Mit Hilfe von Verarbeitungsadaptern erfolgt der Zugriff auf die UI-Elemente der zu bewertenden Benutzungsoberfläche und die Evaluation anhand der Richtlinien-Metriken. Die Ergebnisse der Evaluation werden in Form eines Berichts an den Entwickler zurückgegeben. (Robal, et al., 2017) (Marenkov, et al., 2018)

### **Modell-basierte Werkzeuge**

Bei *WebTango* handelt es sich um einen, an der University of California entwickelten Prototypen zur automatisierten Usability-Evaluation von Benutzungsoberflächen. WebTango verwendet die zufallsgesteuerte Simulationstechnik der Monte-Carlo-Simulation, um das Verhalten von menschlichen Nutzer\*innen bei der Informationssuche anzunähern und die daraus entstehenden Navigationspfade vorherzusagen, zusammen mit quantitativen Usability-Kennzahlen, die aus der Analyse einer großen Anzahl von Webseiten abgeleitet wurden, um Aussagen über die Usability der Informationsarchitekturen von Webseiten zu generieren. Die Ergebnisse der Evaluation werden anschließend mit angemessenen Gegenstücken erfolgreicher Webseiten verglichen, die hohe Bewertungen bei den Webby Awards (The Webby Awards, 2022) erhielten. (Ivory, et al., 2000) (Dingli, et al., 2014)

Der akademische Prototyp *InfoScent Bloodhound Simulator* beschreibt ein Navigationsanalyzesystem, das die Navigationsfähigkeit von Webseiten misst, indem es die Wahrscheinlichkeit berechnet, mit der Nutzer ihre gewünschten Ziele erreichen und einen Bericht darüber erstellt. Die Ziele werden dazu in Form von Schlüsselwörtern angegeben, die in Zusammenhang mit dem Ziel stehen. Die Methode zur Messung der Wahrscheinlichkeit wird von den Autoren als „Information Scent Absorption Rate“ bezeichnet und basiert auf dem Konzept des „Information Scent“. Die Idee dahinter ist, dass Nutzer\*innen anhand der Hinweise, die einen Hyperlink umgeben, entscheiden, ob es sich lohnt, die Informationen auf der anderen Seite des Links zu erkunden. (Chi, et al., 2003)

Im DAI-Labor der Technischen Universität Berlin wurde ein System zur Simulation von Nutzungsinteraktionen, die MeMo-Workbench, entwickelt. Auf Basis zuvor definierter Nutzer\*innen-Attribute sowie Eigenschaften und Beschriftungen von Objekten einer Benutzungsschnittstelle werden eine Wahrnehmungs-Reihenfolge der Objekte und die Interaktionswahrscheinlichkeit mit jedem dieser Objekte bei der Bearbeitung einer Aufgabe ermittelt. Basierend auf kognitiven Eigenschaften der simulierten Benutzer werden UI-Objekte zur Interaktion gewählt und anschließend mit den Koordinationsfähigkeiten des

Nutzer\*innen-Modells und den visuellen Eigenschaften der Objekte ermittelt, ob die Aktion erfolgreich sein wird. Während der Simulation werden verschiedene Informationen, unter anderem über das Interaktionselement, mit der Auswahl in Zusammenhang stehende Eigenschaften der nutzenden Person, der Wahrscheinlichkeitsverteilung und weiterer Statistiken gesammelt und anschließend teilweise visualisiert. Dies soll die bewertende Person bei der Identifikation problematischer Bereiche unterstützen. (Feuerstack, et al., 2008)

### A.2. Leitfaden für die Befragungen

<b>Leitfragen, Erzählaufrorderungen</b>	<b>Checkliste – Wurde das erwähnt?</b>  <i>Nachfragen, wenn nicht von allein angesprochen</i>	<b>Konkrete Fragen</b>  <i>An passender Stelle</i>	<b>Aufrechterhaltungs-, Steuerungsfragen</b>
Du hast erzählt, dass du im UX-Bereich schon gearbeitet hast.  Was hast du da gemacht?  Welchen Bezug hattest du da zu Usability?		Hast du Usability-Evaluationen durchgeführt?	
Stell dir vor, du gestaltest mit einem Prototyping-Werkzeug (wie Figma oder AdobeXD) einen Entwurf für ein UI.  Wie gehst du dabei vor?  Was machst du dabei?	Oberfläche von Prototyping-Werkzeugen  Gestaltungsablauf  Evaluation des UIs	Wie kannst du dir vorstellen, eine (Usability-) Evaluation in den Gestaltungsprozess einzubeziehen?	Wie wäre das z.B. bei der Entwicklung einer Oberfläche für eine App oder Webseite?
Stell dir vor, es gäbe ein wundersames (Blackbox-) Werkzeug, das die Usability-Evaluation für dich übernehmen kann.  Wie stellst du dir dieses Werkzeug vor?	Visuelle Aspekte  Funktionale Aspekte	Was macht das Werkzeug?  Wie bekommst du von Aktionen des Werkzeugs mit?  Welche Rückmeldungen erwartest du?	Wie könnte das Werkzeug dir z.B. bei der Erstellung eines Menüs helfen?
Wie stellst du dir konkret deine Interaktion mit diesem Werkzeug vor?	Rückmeldungen  Eingaben	Wie kann die Interaktion mit dem Werkzeug, deiner Vorstellung nach, den Evaluationsvorgang unterstützen?  Wie würdest du auf Rückmeldungen reagieren?	
		Letzte Frage: Hast du Punkte, die du gerne noch ansprechen würdest?	

### **A.3. Usability-Metriken**

#### **Anzahl der Datumsformate**

Aus Gründen der Konsistenz sollte auf einer Website nur ein einziges Datumsformat verwendet werden. (Trojahn, 2020)

#### **Anzahl der Zeitformate**

Aus Gründen der Konsistenz sollte auf einer Website nur ein einziges Zeitformat verwendet werden. (Trojahn, 2020)

#### **Anzahl ungebräuchlicher Akronyme**

Im Sinne einer guten Lesbarkeit sollten Akronyme, die den Benutzer\*innen möglicherweise unbekannt sein könnten, vermieden werden. (Trojahn, 2020)

#### **Anzahl mehrdeutiger Links**

Mehrdeutige Links können für alle Benutzer\*innen irreführend sein. Nutzer\*innen ohne Einschränkungen können die Bedeutung ggf. aus dem Kontext ableiten. Nutzer\*innen dagegen, die assistive Technologien nutzen, könnte es schwerfallen, den zusätzliche Kontext eines Links wahrzunehmen. Aus Gründen der Barrierefreiheit sollten Linktexte folglich ohne Kontext zugänglich sein und keine mehrdeutigen Links verwendet werden. (Trojahn, 2020)

#### **Einheitliche Navigation**

Die Darstellung, das Layout und die Reihenfolge von Navigationsmenüs sollte über mehrere Webseiten hinweg gleich bleiben, es sei denn, der Benutzer veranlasst eine Änderung. Diese Metrik zielt in erster Linie auf eine konsistente Darstellung von sich wiederholenden Inhalten ab und gilt nicht für inhaltliche Unterschiede sekundärer Navigationselemente. (Trojahn, 2020)

#### **Einheitlichkeit des Layouts**

Die Einheitlichkeit des Layouts wird aus den Höhen, den Breiten und den Ausrichtungen der visuellen Komponenten ermittelt und gibt in Bezug auf die Attraktivität der Benutzungsoberfläche Auskunft darüber, wie gut die visuellen Komponenten angeordnet sind. (Padda, 2003) Der Wert sollte zwischen 50 und 85 liegen. (Trojahn, 2020)

$$LU = 100 * \frac{1 - (N_h + N_b + N_o + N_l + N_u + N_r) - M}{6 * N_k - M} \in [0, 100]$$

$$M = 2 + 2 * |2\sqrt{N_c}|$$

$N_k$ : Anzahl Komponenten

$N_h$ : Anzahl unterschiedlicher Höhen<sup>5</sup>

$N_b$ : Anzahl unterschiedlicher Breiten<sup>5</sup>

$N_o$ : Anzahl verschiedener Ausrichtungen der oberen Kante<sup>5</sup>

$N_u$ : Anzahl verschiedener Ausrichtungen der unteren Kante<sup>5</sup>

$N_l$ : Anzahl verschiedener Ausrichtungen der linken Kante<sup>5</sup>

$N_r$ : Anzahl verschiedener Ausrichtungen der rechten Kante<sup>5</sup>

### Farbkonsistenz

Die Farbkonsistenz umfasst den Prozentsatz der vordefinierten Farben im Verhältnis zur Gesamtzahl der verwendeten Farben. Im Idealfall sind nur die vordefinierten Farben enthalten. (Trojahn, 2020)

$$\text{Farbkonsistenz} = \frac{\text{Anzahl der vordefinierten Farben}}{\text{Gesamtzahl der verwendeten Farben}}$$

### Farbkontrast

Das W3-Konsortium legt für den Farbkontrast eine detaillierte Metrik fest, die sich auf die relative Leuchtdichte konzentriert und sowohl die verminderte Sehschärfe als auch Farbsehschwächen berücksichtigt. Die Leuchtdichte von Vordergrundelementen und dem Hintergrund wird mit folgender Formel berechnet. (Trojahn, 2020)

$$L = 0.2126 * R + 0.57152 * G + 0.0722 * B$$

Die Werte für R, G und B werden mit Hilfe einer Gleichung berechnet. Beispielhaft wird hier einmal die Gleichung für R betrachtet, wobei die sRGB-Daten eine Skala von 0 bis 1 verwenden. (Trojahn, 2020)

$$R = \begin{cases} \frac{R \text{ sRGB}}{12.92}, & \text{falls } R \text{ sRGB} \leq 0.03928 \\ \left(\frac{R \text{ sRGB} + 0.055}{12.92}\right), & \text{sonst} \end{cases}$$

Die relative Gesamtleuchtdichte wird mit der folgenden Gleichung bestimmt.

---

<sup>5</sup> von visuellen Komponenten

$$\text{Relative Leuchtdichte} = \frac{L_1 + 0.05}{L_2 + 0.05} \in [1: 1, 21: 1]$$

$L_1$ : Relative Leuchtdichte der helleren Vorder- und Hintergrundfarben

$L_2$ : Relative Leuchtdichte der dunkleren Vorder- und Hintergrundfarben

Die W3C-Richtlinie betrachtet Textkontrast und den Kontrast zwischen aktiven Komponenten der Benutzungsoberfläche oder grafischen Objekten und benachbarten Farben. Dabei gelten die folgenden Schwellenwerte. Für Fließtexte sollte ein Kontrast zwischen 4.5:1 und 7:1 vorliegen. Für große Texte, Komponenten und grafische Objekte wird ein Kontrast von 3:1 bis 4.5:1 empfohlen. Die Schwellenwertbereiche ergeben sich aus verschiedenen Sehschärfestufen. Die untere Grenze ist für Nutzer\*innen mit einer Sehschärfe von 20/40 erforderlich, die obere für Nutzer\*innen mit einer Sehschärfe von 20/80. (Trojahn, 2020)

### **Flesh-Reading-Ease (FRE)**

Der FRE beschreibt die Lesbarkeit bzw. die Verständlichkeit eines Textes und bildet sich aus der durchschnittlichen Anzahl der Wörter pro Satz und der durchschnittlichen Silbenanzahl pro Wort. (Padda, 2003) Optimalerweise sollte der FRE einen Wert zwischen 70 und 80 betragen. (Trojahn, 2020)

$$FRE_{\text{Deutsch}} = 180 - DSL - (58.5 * DSW) \in [0, 100]$$

$DSL$ : Durchschnittliche Satzlänge

$DSW$ : Durchschnittliche Silbenanzahl pro Wort

### **Gesamtdichte**

Die Gesamtdichte drückt aus, wie groß das Verhältnis der Objektbereiche zur gesamten Bildschirmgröße ist. Um ein überladenes Layout zu verhindern, sollten maximal 40% des Bildschirms von Objektbereichen bedeckt sein. (Trojahn, 2020) (Ngo, et al., 2001)

$$\text{Gesamtdichte} = \frac{\sum_i^n a_i}{a_{\text{screen}}} * 100\% \in [0\%, 100\%]$$

$a_i$ : Fläche des Objekts  $i$

$a_{\text{screen}}$ : Fläche des Bildschirms oder Rahmens

### Gleichgewicht

Das Maß beschreibt, wie zentriert das Layout in Bezug auf den Rahmen ist. Wie beim vertikalen Gleichgewicht zuvor wird auch hier aufgrund der Scrolling-Eigenschaften von Webseiten nur die vertikale Dimension betrachtet. (Trojahn, 2020)

### Größe grafischer Links

Laut QUIM sollten Grafiklinks mindestens 1,27 cm x 1,27 cm groß sein. Für den Webbereich ist die Definition der Größe in der Einheit Zentimeter ungeeignet, weshalb das W3-Konsortium einen Schwellenwert von 44 CSS-Pixel x 44 CSS-Pixel dafür vorschlägt. (Trojahn, 2020)

### Linkkonsistenz

Links mit demselben Ziel sollten die gleiche Linkbeschreibung haben. Im Einklang mit diesem Ziel sollten Links mit unterschiedlichen Zielen nicht denselben Linktext verwenden. Dies gilt ebenso für nicht-textliche Verlinkungen. (Trojahn, 2020)

### Maß für die Regelmäßigkeit

Die Regelmäßigkeit für die Bildschirmgestaltung ist ein Maß für die Ausrichtung der visuellen Objekte. Die Anzahl der Ausrichtungspunkte sollte minimiert und horizontale und vertikale Ausrichtungspunkte möglichst gleichmäßig verteilt werden. (Ngo, et al., 2001) Optimalerweise sollte der Wert für die Regelmäßigkeit bei 1 liegen. (Trojahn, 2020)

$$R = \frac{|R_{Ausrichtung}| + |R_{Abstände}|}{2} \in [0, 1]$$

$$R_{Ausrichtung} = 1 - \frac{n_{vap} + n_{hap}}{2n}$$

$$R_{Abstände} = \begin{cases} 1, & \text{falls } n = 1 \\ 1 - \frac{n_{Abstände} - 1}{2 * (n - 1)}, & \text{sonst} \end{cases}$$

*R<sub>Ausrichtung</sub>*: Ausmaß, in dem die Ausrichtungspunkte minimiert werden

*n<sub>vap</sub>*: Anzahl vertikaler Ausrichtungspunkte

*n<sub>hap</sub>*: Anzahl horizontaler Ausrichtungspunkte

*R<sub>Abstände</sub>*: Ausmaß, in dem die Ausrichtungspunkte gleichmäßige Abstände haben

*n<sub>Abstände</sub>*: Anzahl verschiedener Abstände zw. Spalten- und Zeilenanfangspunkten

*n*: Anzahl der Objekte

## Ränder

Aus Gründen der Konsistenz sollten die Ränder zwischen dem oberen, unteren, linken und rechten Rahmen des Browserfensters und dem nächstgelegenen Objekt (Textfeld, Bild etc.) über mehrere Seiten hinweg gleich sein. (Trojahn, 2020)

## Schriftgröße

Laut U.S. Department of Health & Human Services rufen Schriftarten mit weniger als 12 Punkt eine geringere Leseleistung bei den Benutzer\*innen hervor. Für Benutzer\*innen über 65 Jahren werden Schriftarten mit mindestens 14 Punkt empfohlen. Ggf. kann die Schriftgröße durch benutzerdefinierte Browsereinstellungen oder die Größe der Pixel des verwendeten Monitors beeinflusst werden. (U.S. Department of Health & Human Services, 2006)

## Schriftkonsistenz

Die Schriftkonsistenz umfasst den Prozentsatz der vordefinierten Schriftformate im Verhältnis zur Gesamtzahl der verwendeten Schriftformate. Ein Schriftformat kann die Schriftfamilie, die Größe, den Stil, die Farbe und andere Schriftmerkmale wie den Zeichenabstand umfassen. Es sollten nur (einige wenige) vordefinierte Schriftformate verwendet werden. (Trojahn, 2020)

$$\text{Schriftkonsistenz} = \frac{\text{Anzahl der vordefinierten Schriftformate}}{\text{Gesamtzahl der verwendeten Schriftformate}}$$

## Sparsamkeit

Der Einfachheit halber sollte die Anzahl der verschiedenen Größen, Farben und Formen minimiert werden. (Trojahn, 2020)

$$\text{Sparsamkeit} = \frac{3}{(N_{\text{Größe}} + N_{\text{Farbe}} + N_{\text{Form}})} \in [0, 1]$$

$N_{\text{Größe}}$ : Anzahl verschiedener Größen

$N_{\text{Farbe}}$ : Anzahl verschiedener Farben

$N_{\text{Form}}$ : Anzahl verschiedener Formen

## Vertikales Gleichgewicht

Das Gleichgewicht beschreibt die Verteilung des optischen Gewichts in einem Bild. Größere Objekte sind dabei schwerer gewichtet als kleinere. Eine ausgewogene Gestaltung einer Benutzungsoberfläche kann durch eine gleichmäßige Gewichtung der Bildelemente

erreicht werden. (Ngo, et al., 2001) Die ursprüngliche Metrik von Ngo und Byrne kombiniert die horizontale und vertikale Balance und berücksichtigt keine scrollbaren und größenveränderbaren Bildschirme. Bei scrollbaren Designs ist es weniger sinnvoll, die horizontale Balance für alle möglichen Bildlaufpositionen zu berechnen, da es für Designer\*innen unmöglich ist, die horizontale Balance für alle Größen und Positionen des Fensters zu verbessern. (Trojahn, 2020)

$$B_{\text{vertikal}} = \frac{w_L - w_R}{\max(|w_L|, |w_R|)} \in [-1,1], \text{ mit } w_j = \sum_i^{n_j} a_{ij} * d_{ij}, j = L, R$$

$a_{ij}$ : Fläche des Objekts  $i$  auf der Seite  $j$

$d_{ij}$ : Abstand zwischen den Mittellinien des Objekts und dem Rahmen

$n_j$ : Gesamtzahl der Objekte auf Seite  $j$

### **Verwaiste Seiten**

Verwaiste Seiten sind Teil einer Website, aber keine der anderen Seiten der Website verweist auf sie, so dass sie nicht über einen internen Link erreicht werden können. Sie verschlechtern die Navigierbarkeit einer Webseite und sollten daher vermieden werden. (Trojahn, 2020)

### **Verweis auf Startseite**

Bei einer Webseite mit mehr als einer Seite, sollten alle Seiten einen Link zurück zur Homepage/Startseite haben. Lediglich die Startseite/Homepage selbst muss nicht mit sich selbst verlinkt sein. (Trojahn, 2020)

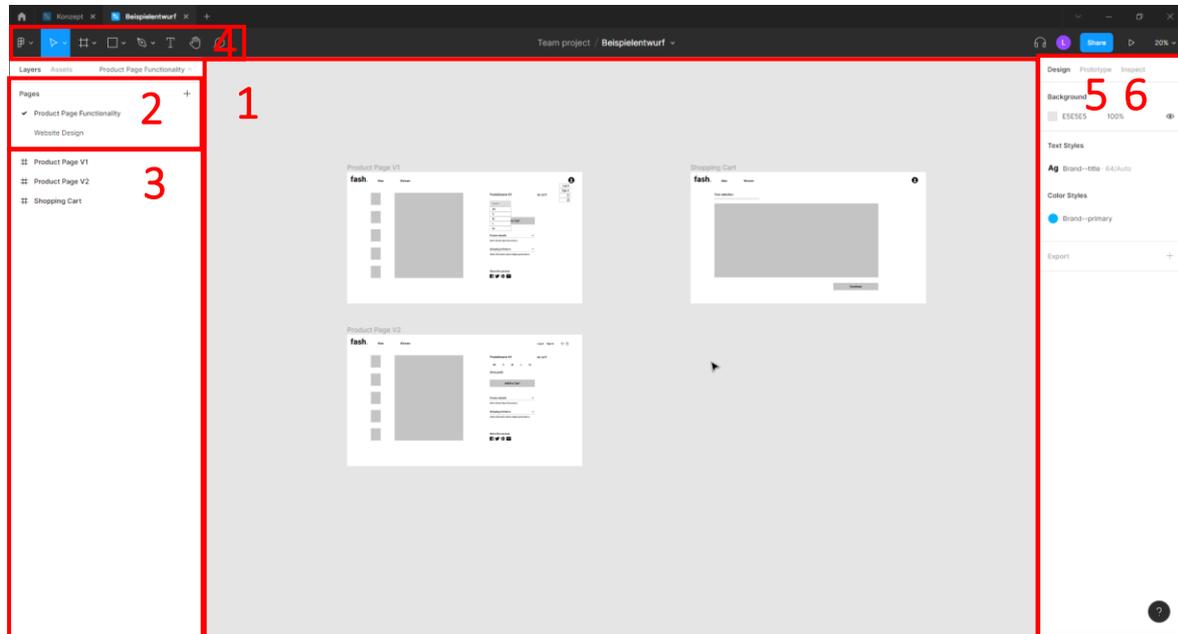
$$\text{Relative Referenz} = \frac{\text{Anzahl der Seiten mit Links zurück zur Homepage}}{\text{Gesamtzahl der Seiten einer Website}}$$

#### A.4. Vordefinierte Evaluationsprofile

Metriken	Kriterien							
	Attraktivität	Konsistenz	Min. mentale Belastung	Präsentation	Verständlichkeit	Lesbarkeit	Navigierbarkeit	Einfachheit
Anzahl der Datumsformate		x						
Anzahl der Zeitformate		x						
Anzahl ungebräuchlicher Akronyme			x		x			
Anzahl mehrdeutiger Links							x	
Einheitliche Navigation		x		x			x	
Einheitlichkeit des Layouts	x	x	x	x				
Farbkonsistenz	x	x		x				
Farbkontrast						x		
FRE					x	x		
Gesamtdichte	x	x			x			
Gleichgewicht	x	x						
Größe grafischer Links							x	
Linkkonsistenz		x	x				x	
Maß für die Regelmäßigkeit	x	x						
Ränder	x	x						
Schriftgröße				x		x		
Schriftkonsistenz	x	x		x		x		
Sparsamkeit								x
Vertikales Gleichgewicht	x			x				
Verwaiste Seiten							x	
Verweis auf Startseite							x	

## A.5. Aufgabenstellung Test

### Erläuterung der Benutzungsoberfläche:



In der Mitte des Fensters befindet sich die Leinwand (1). Hier erkennst du einen groben Entwurf eines Teils einer Website.

Unter dem Punkt „Pages“ auf der linken Seite (2), kannst du zwischen Figma-Seiten wechseln (bleibe aber vorerst auf dieser Seite).

Darunter befindet sich eine Auflistung deiner Frames (3), die auf der Leinwand jeweils eine Seite der Website darstellen. Hier kannst du auf alle Elemente des Entwurfs zugreifen.

An der oberen Seite befindet sich eine Symbolleiste (4) unter der du verschiedene Werkzeuge auswählen kannst.

Auf der rechten Seite kannst du UI-Elemente bearbeiten (Reiter „Design“) (5) und Interaktionen hinzufügen (Reiter „Prototype“) (6). Wenn kein Element ausgewählt wird und der Reiter „Prototype“ geöffnet wird, werden alle vorhandenen Interaktionen angezeigt.

### Aufgabe 1

- 1) Starte auf der Seite „Product Page Functionality“. Hier solltest du zwei alternative Gestaltungsentwürfe der „Product Page“ erkennen (V1 und V2). Diese beiden Entwürfe sollen mit Hilfe einer Usability Evaluation verglichen werden.

- 2) Öffne dazu über den Shortcut STRG + P das Plugin „Usability Evaluation – Aufgabenevaluation“. Über das Buch-Icon kannst du auf eine Dokumentation für das Plugin zugreifen.
- 3) Unter dem Reiter „Aufgaben“ sollten bereits drei Aufgaben angelegt sein. Zwei Aufgaben gehören dabei zum Gestaltungsentwurf V1. Die dritte Aufgabe beinhaltet die Eingabe der Adressinformationen auf der Seite „Checkout“. Erstelle nun zwei weitere Aufgaben.
  - a. In der ersten Aufgabe soll das auf der „Product Page V2“ angezeigte Produkt in Größe M zum Warenkorb hinzugefügt werden.
  - b. In der zweiten Aufgabe soll ausgehend von der „Product Page V2“ zur Eingabe der Adressinformationen („Checkout“) navigiert werden.
- 4) Erzeuge ein Szenario für „Product Page V2“, welches die Auswahl des Produkts, die Navigation auf die Seite „Checkout“ und die Eingabe der Adressinformationen (Aufgabe „Adressinformationen eingeben“) beinhaltet.
- 5) Wie du vielleicht gesehen hast, fehlt bei der Eingabe der Adressinformationen eine E-Mail. Bitte füge die Eingabe der E-Mail-Adresse an den Anfang der entsprechenden Aufgabe hinzu.
- 6) Evaluiere dieses Szenario bzw. die Aufgaben nun und vergleiche sie mit der Aufgabenausführung für den Gestaltungsentwurf V1. Probiere hier ruhig einige Dinge mit dem Plugin aus.
- 7) Entscheide dich für die bessere Version der beiden Alternativen und verbessere diese, falls nötig, noch weiter.

## **Aufgabe 2**

- 1) Wechsel nun auf die Seite „Website Design“. Hier solltest du nun einen etwas ausgereiften Website Design“. Hier solltest du nun einen etwas ausgereiften Website-Entwurf erkennen können. Diesen Entwurf sollst du nun Metriken-basiert evaluieren. Wähle dazu über den Shortcut STRG + P das Plugin „Usability Evaluation – Metriken-Evaluation aus.
- 2) Verbessere den Entwurf mit Hilfe des Plugins. Probiere hier ruhig einige Dinge aus.

### A.6. Fragebogen Test

1. Hattest du das Gefühl, dass dich das Plugin bei der Evaluation der Entwürfe zielführend unterstützen konnte?

- Ja
- Nein
- Weiß nicht

2. Inwiefern fühltest du dich durch das Plugin in der Evaluation unterstützt?

Keine Unterstützung						Volle Unterstützung

3. Inwiefern fühltest du dich durch das Plugin in der Gestaltung unterstützt?

Keine Unterstützung						Volle Unterstützung

4. Wie hilfreich empfandest du die Ergebnisse der Metriken-Evaluation zur Verbesserung des Entwurfs?

Gar nicht hilfreich						Sehr hilfreich

5. Falls du auf die vorherige Frage mit 1 oder 2 geantwortet hast: Welche Informationen hättest du dir bei der Darstellung der Ergebnisse gewünscht? Was hat dich gestört?

6. Wie hilfreich empfandest du die Ergebnisse der Aufgaben-Evaluation zur Verbesserung des Entwurfes?

Gar nicht hilfreich							Sehr hilf- reich

7. Falls du auf die vorherige Frage mit 1 oder 2 geantwortet hast: Welche Informationen hättest du dir bei der Darstellung der Ergebnisse gewünscht? Was hat dich gestört?

8. Wie wahrscheinlich wäre es, dass du das Plugin nochmal verwenden würdest?

Sehr un- wahrschein- lich							Sehr wahr- scheinlich

9. Falls du die Frage zuvor mit 1, 2 oder 3 beantwortet hast: Was würde dich davon abhalten?

10. Wie wahrscheinlich wäre es, dass du das Plugin weiterempfehlen würdest?

Sehr un- wahrschein- lich							Sehr wahr- scheinlich

11. Falls du die Frage zuvor mit 1, 2 oder 3 beantwortet hast: Was würde dich davon abhalten?

## Index

<b>A</b>	
Annotation .....	27
Aufgabe .....	27
Automatisierungsgrad.....	9
<b>B</b>	
Barrierefreiheit.....	12
<b>D</b>	
Design-System.....	17
DIN EN ISO 9241.....	6
<b>E</b>	
Evaluationsprofil .....	25
<b>F</b>	
Figma.....	20
<b>G</b>	
Gebrauchstauglichkeit .....	4
GOMS-Modell .....	8
<b>H</b>	
Hinweismuster .....	28
<b>K</b>	
Keystroke Level Model .....	26
<b>M</b>	
menschenzentrierten Gestaltung .....	14
<b>N</b>	
Nutzungskontext.....	14
Nutzungsmuster.....	10
<b>P</b>	
Plugin .....	20
Prototyp	
horizontal.....	16
vertikal .....	16
Prototyping.....	15
Prototyping-System .....	23
<b>Q</b>	
Quality in Use Integrated Map .....	12
<b>R</b>	
Research-based Web Design & Usability Guidelines .....	7
<b>S</b>	
Szenario .....	27
<b>T</b>	
Touch-Level-Model.....	13
<b>U</b>	
Usability .....	4
Usability-Engineering.....	14
Usability-Evaluation .....	7
analytische Methoden .....	8
empirische Methoden.....	8
formativ .....	7
summativ .....	7
Usability-Evaluation, automatisiert	
interaktionsbasiert.....	10
Metriken-basiert .....	10
Modell-basiert.....	11
Usability-Normen.....	6
Usability-Prinzipien .....	5
Usability-Problem .....	7
Usability-Richtlinien .....	6
<b>W</b>	
Web Content Accessibility Guidelines .....	7

## Inhalt der CD

- Masterarbeit
- Programmcode des entwickelten Plugins
- Beispielentwürfe als PDF
- Aufgabenstellung für den Test
- Fragebogen für den Test
- Ausgefüllte Fragebögen der Testpersonen
- Interviewleitfaden

## Installationsanleitung Plugin

1. Den Figma Desktop-Client öffnen.
2. Auf der Startseite über die Profileinstellungen den Menüpunkt *Plugins* wählen.
3. Unter *In development* > *New Import from Manifest* wählen.
4. Die Datei *manifest.json* im Ordner *Plugin* > *Usability Evaluation* auswählen.
5. Das Plugin kann nun im Figma-Editor verwendet werden.