

# **Softwareergonomie und Barrierefreiheit - Modus für Farbblindheit/Sehschwäche am Beispiel des Produktes yeet der Firma vectorsoft**

Studiengang Medieninformatik

## **Bachelorarbeit**

vorgelegt von

**Felix Wegener**

geb. in Hanau

durchgeführt bei der  
Vectorsoft AG

Referent der Arbeit: Lisa Ranold M.Sc.  
Korreferent der Arbeit: Prof. Dr. Monika Maria Möhring, Dipl.-Chem., MBA  
Betreuer im Unternehmen: Miqueas Cramer

Friedberg, 2021



*Für Diana und Michael Wegener*



# Abstract

*The aim of the present work is to carry out a comprehensive analysis of competing products of the low-code platform yeet, to examine them with regard to their perceptibility, to define a catalog of requirements and thus to implement a prototype in yeet. Named prototype should deliver a mode for colorblindness/poor eyesight and is to be tested in a user-test. For the analysis, official criteria such as the WCAG2.1 were used in combination with a tool, to investigate the surface of each competing product. Based on these results, a requirements catalog could be created, on which base, a prototype was implemented. Thanks to the collected data, competing products were rated, and a ranking was created. Finally, the user testing was carried out in order to assess the quality on the one hand and to question whether a general solution is possible on the other. On the one hand, the analysis was able to show the weaknesses and strengths of the competition in terms of their perceptibility and, on the other hand, testing the prototype was able to positively confirm its quality.*



# Danksagung

Dieses Kapitel dient dazu, sich bei all den Beteiligten zu bedanken, die tatkräftig den Prozess der Erarbeitung dieser wissenschaftlichen Arbeit entweder passiv oder aktiv unterstützt haben.

In erster Linie wäre da die Firma Vectorsoft AG zu nennen, ohne deren Unterstützung in Form eines Projektes und Rahmenbedingungen diese Arbeit so nicht hätte realisiert werden können. Durch die Bereitstellung von Hard- und Software sowie fachlichen Kompetenzen konnte diese Arbeit erst so realisiert werden, wie sie hier zu lesen ist. Dabei sind insbesondere die Kollegen und Kolleginnen zu nennen, die mich über den gesamten Schaffensprozess unterstützt haben und von der Ideenfindung über die Ausarbeitung bis hin zur Realisierung der wissenschaftlichen Arbeit geholfen haben.

Des Weiteren möchte ich mich bei allen Helfern bedanken, die diese Arbeit Korrektur gelesen haben und damit dafür gesorgt haben, grammatikalische Probleme oder Rechtschreibfehler festzustellen, welche ich selbst nicht in der Lage war, zu erkennen. Nicht nur die Grammatik, sondern vielmehr auch die Lesbarkeit und die Sprache wurden durch die Vorschläge stetig verbessert. Erst durch deren Zutun konnte der Lesefluss optimiert werden, sodass ein flüssiges Lesen möglich ist.

Zuletzt möchte ich mich noch bei meiner Familie und meinen Freunden bedanken, die mich sowohl moralisch, aber auch ressourcentechnisch während der Erarbeitung unterstützt haben.



# Selbstständigkeitserklärung

Ich erkläre, dass ich die eingereichte Bachelorarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die von mir angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Hanau, den 11.10.2021

Felix 

Felix Wegener



# Inhaltsverzeichnis

<b>Abstract</b>	<b>i</b>
<b>Danksagung</b>	<b>iii</b>
<b>Selbstständigkeitserklärung</b>	<b>v</b>
<b>Inhaltsverzeichnis</b>	<b>vii</b>
<b>Abbildungsverzeichnis</b>	<b>xi</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Vorbemerkung . . . . .	1
1.2 Motivation . . . . .	1
1.3 Problemstellung . . . . .	3
1.4 Zielsetzung . . . . .	4
1.5 Aufbau der Arbeit . . . . .	4
<b>2 Stand der Technologien</b>	<b>7</b>
2.1 Einleitende Worte . . . . .	7
2.2 Bereits existierende Arbeiten . . . . .	7
2.2.1 Arbeit: "Barrierefreiheit betriebswirtschaftlicher Anwendungen - Diskussion spezieller Anforderungen und Lösungsansätze" . . . . .	8
2.2.2 Arbeit "Barrierefreie Web-Nutzung durch blinde und sehbehinderte Menschen" . . . . .	9
2.2.3 Arbeit "Web Accessibility for Visually Impaired People: Requirements and Design Issues" . . . . .	10
2.3 Wie weit ist der Entwicklungsstand der Technologien? . . . . .	11
2.4 Ansatzpunkt dieser wissenschaftlichen Arbeit . . . . .	12
<b>3 Grundlagenkapitel</b>	<b>15</b>
3.1 Intro . . . . .	15
3.2 Softwareergonomie . . . . .	15
3.2.1 Computersysteme im Alltag . . . . .	15
3.2.2 Interaktivität von Computersystemen . . . . .	16

3.2.3	Teildisziplin Softwareergonomie . . . . .	16
3.2.4	Computer-Ergonomie . . . . .	16
3.2.5	Wirtschaftlicher Nutzen . . . . .	17
3.2.6	Verschwimmende Grenzen zwischen Arbeit und Freizeit . . . . .	17
3.2.7	Interaktionsdesign . . . . .	17
3.2.8	Gebrauchstauglichkeit . . . . .	17
3.3	Barrierefreiheit . . . . .	18
3.3.1	Der Begriff der Behinderung . . . . .	18
3.3.2	Unternehmerische Sicht . . . . .	19
3.3.3	Barrierefreiheit im Web . . . . .	19
3.3.4	W3C . . . . .	19
	WCAG . . . . .	20
3.3.5	BITV . . . . .	20
3.4	Farbblindheit / Sehschwäche . . . . .	21
3.4.1	Menschliche Farbwahrnehmung . . . . .	21
3.4.2	Farbblindheit . . . . .	22
3.5	Low-Code . . . . .	24
3.5.1	Begriffserklärung . . . . .	24
3.5.2	Erleichterung durch Low-Code . . . . .	25
3.6	yeet . . . . .	25
3.6.1	Unternehmen: Vectorsoft AG . . . . .	25
3.6.2	Produkterklärung . . . . .	25
3.6.3	Einsatzgebiet der Abschlussarbeit . . . . .	27
<b>4</b>	<b>Analyse von Konkurrenzprodukten</b>	<b>29</b>
4.1	Umfrage bei Vectorsoft AG . . . . .	29
4.2	Einleitende Worte . . . . .	29
4.3	Vorgehensweise . . . . .	30
4.4	Kriterien . . . . .	30
4.4.1	Erfolgskriterium 1.4.1 Verwendung von Farbe . . . . .	30
4.4.2	Erfolgskriterium 1.4.3 Kontrast(Minimum) . . . . .	31
4.4.3	Erfolgskriterium 1.4.4 Textgröße ändern . . . . .	31
4.4.4	Erfolgskriterium 1.4.5 Bilder von Text . . . . .	31
4.4.5	Erfolgskriterium 1.4.6 Kontrast(Erweitert) . . . . .	31
4.4.6	Erfolgskriterium 1.4.8 Visuelle Präsentation . . . . .	31
4.4.7	Erfolgskriterium 1.4.9 Bilder von Text(ohne Ausnahme) . . . . .	31
4.4.8	Erfolgskriterium 1.4.10 Reflow . . . . .	32
4.4.9	Erfolgskriterium 1.4.11 Kontrast ohne Text . . . . .	32
4.4.10	Erfolgskriterium 1.4.12 Textabstand . . . . .	32
4.4.11	Erfolgskriterium 1.4.13 Inhalt auf Hover oder Fokus . . . . .	32
4.5	Verwendete Hilfswerkzeuge . . . . .	32
4.6	Plattform 1: Airtable . . . . .	35
4.6.1	Produktbeschreibung . . . . .	35
4.7	Plattform 2: Quickbase . . . . .	35

4.7.1	Produktbeschreibung . . . . .	35
4.8	Plattform 3: Simplifier . . . . .	36
4.8.1	Produktbeschreibung . . . . .	36
4.9	Analyse der Plattformen . . . . .	37
4.10	Auswertung der Analyseergebnisse . . . . .	40
4.10.1	Vorwort . . . . .	40
4.10.2	Auswertung der Excel-Daten . . . . .	40
4.10.3	Auswertung Bewertungsbogen . . . . .	43
4.10.4	Stellenwert von Barrierefreiheit . . . . .	45
<b>5</b>	<b>Anforderungskatalog</b>	<b>47</b>
5.1	Anforderungen des Prototyps und der Bedienoberfläche . . . . .	47
5.1.1	Farbe . . . . .	47
5.1.2	Schrift . . . . .	48
5.1.3	Kontrast . . . . .	49
5.1.4	Skalierbarkeit . . . . .	49
5.1.5	Theming . . . . .	50
5.1.6	Checkliste . . . . .	51
<b>6</b>	<b>Implementierung eines Prototyps</b>	<b>53</b>
6.1	Verwendete Hilfswerkzeuge . . . . .	53
6.2	Implementierung . . . . .	54
6.2.1	Farbe . . . . .	54
6.2.2	Schrift . . . . .	55
6.2.3	Kontrast . . . . .	56
6.2.4	Skalierbarkeit . . . . .	56
6.2.5	Theming . . . . .	57
6.2.6	Erstellen eines neuen Themes . . . . .	57
6.3	Vorstellung des implementierten Prototyps . . . . .	58
<b>7</b>	<b>Durchführung eines Usability-Tests</b>	<b>59</b>
7.1	Sinnhaftigkeit des Usability-Tests . . . . .	59
7.2	Aufbau des Usability-Tests . . . . .	59
7.3	Ergebnisse des Usability-Tests . . . . .	61
7.3.1	Ergebnisse der aufgestellten Thesen . . . . .	63
	"Eine Person mit Farbsehschwäche kann nach der Überarbeitung besser mit yeet arbeiten als vorher." . . . . .	63
	"Eine Person ohne Einschränkungen kann nach der Überarbeitung mindestens genauso gut mit yeet arbeiten wie vorher." . . . . .	63
<b>8</b>	<b>Ergebnisse</b>	<b>65</b>
8.1	Vorwort zum Prototyp . . . . .	65
8.2	Beurteilung der Qualität des Prototyps . . . . .	65
8.3	Interpretation des Usability-Tests . . . . .	66

8.3.1	“Eine Person mit Farbsehschwäche kann nach der Überarbeitung besser mit yeet arbeiten als vorher.“ . . . . .	66
8.3.2	“Eine Person ohne Einschränkungen kann nach der Überarbeitung mindestens genauso gut mit yeet arbeiten wie vorher.“ . . . . .	66
<b>9</b>	<b>Zusammenfassung und Ausblick</b>	<b>67</b>
9.1	Zusammenfassung der erarbeiteten Ergebnisse . . . . .	67
9.2	Die Zukunft von yeet . . . . .	68
9.2.1	Ausrichtung hinsichtlich Barrierefreiheit . . . . .	68
9.2.2	Weiterentwicklung des Prototyps . . . . .	68
9.3	Entwicklung von Softwareergonomie und Barrierefreiheit in Zukunft . . . . .	69
9.3.1	Hilfswerkzeuge mit künstlicher Intelligenz . . . . .	69
9.3.2	Schulungen und Testing durch eigenes Personal . . . . .	70
9.3.3	Fazit zu beiden Ansätzen . . . . .	70
<b>A</b>	<b>Anhang</b>	<b>71</b>
A.1	Dokumente . . . . .	71
	<b>Glossar</b>	<b>73</b>
	<b>Literaturverzeichnis</b>	<b>75</b>

# Abbildungsverzeichnis

2.1	Low-Code Marktgröße . . . . .	11
3.1	Farbblindheit . . . . .	22
3.2	Arten der Farbblindheit . . . . .	23
3.3	Beispielanwendung . . . . .	26
3.4	yeet-Designer . . . . .	26
4.1	Untersuchen eines Elements im Browser . . . . .	33
4.2	Color Contrast Analyzer . . . . .	34
4.3	Simulation von Farbblindheit - CCA . . . . .	34
4.4	Airtable . . . . .	35
4.5	Quickbase . . . . .	36
4.6	Simplifier . . . . .	37
4.7	Durchschnittlicher Kontrast . . . . .	41
4.8	Anzahl bestandener/durchgefallener Proben . . . . .	42
5.1	Theming der Plattformen . . . . .	50
5.2	Checkliste - Prototyp . . . . .	51
6.1	Leonardo . . . . .	53
6.2	Prototyp - Farbe . . . . .	55
6.3	Prototyp - Schrift . . . . .	55
6.4	Prototyp - Kontrast . . . . .	56
6.5	Prototyp - Skalierbarkeit . . . . .	56
6.6	Prototyp - Theming . . . . .	57
6.7	Überblick über den Prototyp . . . . .	58
7.1	Bewertung Wahrnehmbarkeit . . . . .	62
7.2	Bewertung Elementgröße . . . . .	62



# Kapitel 1

## Einleitung

Die erbrachte wissenschaftliche Arbeit wurde in gemeinsamer Zusammenarbeit mit der Firma Vectorsoft AG <sup>1</sup> erarbeitet und befasst sich mit der Analyse und dem Vergleich von Low-Code Plattformen hinsichtlich ihrer Barrierefreiheit und dem Entwurf eines Prototyps für einen Modus für Farbblindheit/Sehschwäche zur Integration in das neue Produkt *yeet* (Siehe Kapitel 3.6), welches die Firma aktuell entwickelt.

### 1.1 Vorbemerkung

Im Sinne eines guten Leseflusses und aufgrund der fehlenden Bewandnis zum Thema Gendern wird bewusst in dieser Arbeit darauf verzichtet und mit dem generischen Maskulinum geschrieben. Allerdings soll dies niemanden in seiner Person und seinem Geschlecht diskriminieren und jeder interessierte Leser soll sich dementsprechend angesprochen fühlen.

### 1.2 Motivation

**“Behinderung ist eine schwere Last, die sich erleichtern lässt, wenn es uns gelingt [sic] zu lernen, wie wir uns auf Verschiedenheit einstellen können.”** [Wei93]

Dieses Kapitel beginnt ganz bewusst mit einem Zitat, da dieses den Zweck dieser Arbeit nahezu identisch widerspiegelt. In der heutigen Zeit ist kaum ein Thema so relevant und in den Fokus gerückt wie die Gleichstellung und Gleichberechtigung von Menschen jeglichen Geschlechts, jeglicher Herkunft und körperlicher Verfassung. Eines der Themen, das gerade im letzten Jahrzehnt die Gesellschaft enorm beschäftigt hat, ist die Gleichstellung von Menschen mit Behinderungen. Ziel dieser Bemühungen ist es, die Einschränkungen jeglicher Art zu minimieren und im Idealfall vollständig zu beseitigen. Ab diesem Zeitpunkt muss dann nicht mehr von Menschen mit einer Behinderung gesprochen werden, da eine vorhandene

---

<sup>1</sup><https://www.vectorsoft.de/>

Einschränkung sei sie körperlicher oder geistiger Natur, sie nicht mehr im alltäglichen oder im beruflichen Leben behindert. Bereits im Kindesalter werden junge Menschen mit diesem Thema konfrontiert und es wird versucht, mit Aufklärungsarbeit Bewusstsein für die Einschränkungen dieser Menschen zu schaffen.

Meist beginnt die Aufklärungsarbeit bereits im Biologieunterricht der Schule und wird fortgeführt bis hin zu Vorträgen an der Universität und Aufklärungsmaterial im alltäglichen Leben, welches dem Menschen in unterschiedlicher Form begegnet. Auch an der Technischen Hochschule Mittelhessen wird das Thema ernsthaft angegangen und so gibt es unter anderem ein Zentrum für blinde und sehbehinderte Studierende (BliZ). Im Rahmen des Bachelorstudiengangs Medieninformatik erhalten die Studierenden schon früh Einblick in die Arbeit des BliZ<sup>2</sup> und werden damit konfrontiert. Bereits zu Beginn des Studiums nehmen die Studierenden an einem Vortrag eines blinden Menschen teil, welcher versucht, zu vermitteln, welche Mittel und Wege er verwendet, um die alltäglichen Hürden des Lebens mit einer Behinderung zu meistern und die Hürden der Technik und des Internets zu bewältigen, um möglichst problemfrei durch dieses zu navigieren. Dabei wird neben einer klassischen Fragerunde auch das Werkzeug seiner Wahl vorgestellt und gezeigt, wie der Mann mithilfe einer Braillezeile, also einer "Tastatur" für sehbehinderte Menschen, welche Textausgaben unmittelbar in Blindenschrift übersetzt [Hä16, vgl.], durch das Internet und insbesondere Webseiten und Webanwendungen navigiert. Damit soll ein Verständnis dafür geschaffen werden, worauf die Studierenden als zukünftige Entwickler Rücksicht nehmen sollten. Denn die immer weiter fortschreitende Digitalisierung sorgt dafür, dass immer mehr Bereiche im alltäglichen Leben mit Software durchdrungen werden und angesichts der wachsenden Population auch immer mehr Menschen und damit auch potenzielle Nutzer mit Behinderungen in Berührung mit Software kommen.

Angesichts dieser Tatsachen wird es immer wichtiger, Software einer breiten Masse an Nutzern zugänglich zu machen und dafür Sorge zu tragen, dass möglichst viele unterschiedliche Menschen die Software ordnungsgemäß bedienen können. Als Softwareentwickler kann man heutzutage nicht mehr wissen, unter welchen körperlichen, geistigen aber auch technischen Einschränkungen ein Nutzer leidet. Denn unter einer klassischen Behinderung muss nicht zwangsläufig eine körperliche oder geistige Einschränkung verstanden werden. Behinderungen können in unterschiedlichster Form auftreten und so kann auch eine unpassende oder veraltete technische Ausstattung des Nutzers eine Behinderung und damit Minderung seiner Effizienz und Arbeitsleistung darstellen. Aber schlichtweg auch das Alter des Nutzers kann diesen in der Arbeit mit einer Software einschränken. Dies kann sich von Hemmungen im Umgang mit neuer Technik bis hin zu altersbedingten Schwächen wie einer Sehschwäche erstrecken. Durch einfache Grafiken und Bilder, Testmöglichkeiten sowie diversen Publikationen kann man sich umfassend über das Thema informieren.

Das Produkt der Firma Vectorsoft AG, welches sich aktuell noch im Entwicklungsstadium befindet, aber den Anspruch hat, später mit den Marktführenden im Bereich *Low-Code* (siehe Kapitel 3.5) zu konkurrieren, soll später unter anderem auch bereits existierenden Bestandskunden angeboten werden. Besonders interessant ist, dass unter den Kunden der Firma auch ein Anwender existiert, welcher unter einer Form der Farbblindheit leidet und ab

---

<sup>2</sup><https://www.thm.de/bliz/>

dem Zeitpunkt darüber philosophiert werden muss, wie solche Menschen später die neu entwickelte Low-Code Plattform nutzen werden. Hauptaugenmerk liegt darauf, zu überlegen, ob diese Nutzer das Produkt auch in voller Funktionalität nutzen können.

Primär geht es also zunächst darum, den Nutzern mit einer Sehschwäche die Arbeit mit yeet zu erleichtern und eine gleichwertige Effizienz wie bei uneingeschränkten Nutzern zu gewährleisten. Um sich also einerseits von der Konkurrenz abzuheben oder in diesem Bereich mit ihr zu konkurrieren und andererseits sehbehinderten Nutzern die Chance zu bieten, das Produkt in vollem Umfang und voller Funktionalität zu nutzen, hat diese wissenschaftliche Arbeit den Zweck, die Barrierefreiheit des Produktes yeet nachhaltig zu verbessern.

### 1.3 Problemstellung

Da das Thema *Barrierefreiheit* (Siehe Kapitel 3.3) allerdings noch nicht besonders in den Fokus geraten ist bei der Entwicklung von yeet, kam der Gedanke auf, an dieser Stelle anzusetzen und dieses sehr komplexe und anspruchsvolle Thema anzugehen. Weil eine umfassende Barrierefreiheit ein enorm aufwändiges Unterfangen ist und deren vollständige Umsetzung vermutlich den Rahmen einer Bachelorarbeit und vermutlich auch einer Masterarbeit übersteigt, beschäftigt sich diese wissenschaftliche Arbeit hauptsächlich mit der Analyse von Konkurrenzprodukten und deren Umsetzung von Barrierefreiheit sowie der Integration eines Prototyps für einen Modus für Farbblindheit/Sehschwäche.

Über die Thesis hinaus soll durchaus auch das Augenmerk auf die restlichen Anforderungen ausreichender *Softwareergonomie* ( Siehe Kapitel 3.2) und Barrierefreiheit gelegt werden. Nach allgemeinem Verständnis ist dies heutzutage ein sehr relevantes Thema in der Öffentlichkeit und für ein Produkt, welches die Ambitionen hat, mit den Marktführenden zu konkurrieren, nahezu unumgänglich. Macht man seine Software für eine möglichst breite Masse an Nutzern zugänglich, so kann dies auch als *USP* dienen, um sein Produkt besser vermarkten zu können.

Wie bei nahezu allen Behinderungen, gibt es auch im Bereich der Sehbehinderungen und speziell der Farbblindheit, verschiedene Arten der Sehschwäche, die berücksichtigt werden müssen, um keine Gruppe an Menschen auszuschließen. Um dafür ein besseres Verständnis zu bekommen, muss sich zu Beginn einer solchen Arbeit intensiv mit den verschiedenen Formen der *Farbblindheit* ( Siehe Kapitel 3.4) und vor allem der Farbwahrnehmung des menschlichen Auges befasst werden.

Anhand dieser Grundlagen kann erst verstanden werden, welche verschiedenen Formen es gibt, wie sie sich unterscheiden und welche Gesichtspunkte die Entwickler berücksichtigen müssen. Neben den zahlreichen Formen der Farbblindheit gilt es, bestehende Normen und Möglichkeiten der barrierefreien Gestaltung von Software zu studieren. Zu beachten ist unter anderem, welche Normen vom W3C empfohlen werden und außerdem, welche *ISO*-Normen im Bereich Softwareergonomie und Barrierefreiheit bestehen. Anhand der gesammelten Informationen, welche berücksichtigt werden müssen, kann ein Katalog an Richtlinien zusammengetragen werden, welcher es dem Entwickler und interessierten Leser erleichtern soll, in Zukunft einen solchen Prototyp in Software zu integrieren. Aktuell wird der Entwickler mit

Informationen von vielen Anlaufstellen versorgt und kann in dieser Arbeit ein gesammeltes Werk sehen, welches versucht, alle unter ein Dach zu bringen und Informationen dazu zu liefern.

### 1.4 Zielsetzung

Dem interessierten Leser sollen Themen wie Barrierefreiheit und Softwareergonomie sowie Farbblindheit näher gebracht werden, um ein Verständnis dafür zu schaffen, wie wichtig solche Themen heutzutage sind und welche Vorteile es bringt, Software anhand dieser Kriterien für mehr Nutzer zugänglich und effizient nutzbar zu gestalten. Ziel dieser Arbeit ist es auch, anhand der gesammelten Informationen eine Analyse von Konkurrenzprodukten durchzuführen, die dazu dient, ein Grundgerüst an Anforderungen zu komplettieren. Anhand des selbst erarbeiteten Konzeptes kann dann auch ein Prototyp für die Low-Code Plattform entwickelt und ausgearbeitet werden, der das Produkt nachhaltig hinsichtlich Barrierefreiheit und Softwareergonomie verbessern soll. Die Intention ist, nicht nur für den Leser einen Mehrwert zu liefern, sondern auch die Qualität des Produktes zu verbessern und anhand des Prototyps ein beispielhaftes Vorgehen zur Erstellung eines solchen Modus auszuarbeiten. So können sich zukünftige Entwickler daran orientieren und dieses Vorhaben ähnlich umsetzen.

### 1.5 Aufbau der Arbeit

Zu Beginn der Arbeit wird in einem eigenen Kapitel erläutert, welche Arbeiten im Feld der Barrierefreiheit für Software und Softwareergonomie bereits existieren, ob es Arbeiten bezüglich eines Modus für Farbblindheit in Software bereits gibt und ob auch Arbeiten bestehen, die sich dies im Kontext von Low-Code Plattformen anschauen (Siehe Kapitel 2). Daraufhin soll Grundlagenwissen vermittelt werden, welches dem Leser das Verständnis der nachfolgenden Kapitel erleichtern soll. Dies kann separat von den wissenschaftlich erarbeiteten Resultaten der Analyse von Konkurrenzprodukten und des Modus für Farbblindheit/Sehschwäche betrachtet werden und dient lediglich als Hilfe für Leser, die kaum Wissen in diesen Teilgebieten mitbringen oder vielleicht eine Auffrischung ihrer Kenntnisse benötigen. Konkret soll dort eine Definition und Erklärung von *Barrierefreiheit* (Siehe Kapitel 3.3) geliefert werden und im Kontext von Software verdeutlicht werden, was dies für Softwareprodukte bedeutet und welche Anforderungen bestehen. Gleiches gilt für den Begriff der *Softwareergonomie* (Siehe Kapitel 3.2). Auch dort soll grundlegendes Wissen vermittelt werden, um zu erklären, was diesen Begriff kennzeichnet und was Entwickler dabei berücksichtigen sollten. In einem weiteren Abschnitt wird dann näher auf das Thema *Farbsehen* (Siehe Kapitel 3.4.1) und *Farbblindheit* (Siehe Kapitel 3.4) eingegangen, um Verständnis zu schaffen, wie der Mensch Farbe sieht und verarbeitet und insbesondere, was Farbblindheit für den Nutzer einer Software bedeutet. In einem letzten Abschnitt soll dann außerdem beleuchtet werden, für welche *Firma* (Siehe Kapitel 3.6.1) der spätere Prototyp eingesetzt und vor allem in welchem *Produkt* (Siehe Kapitel 3.6) dieser verwendet wird. Dem Grundlagenkapitel folgen die einzelnen Kapitel zu den Hauptzielen dieser wissenschaft-

lichen Arbeit. Begonnen wird mit der Analyse der Konkurrenzprodukte hinsichtlich deren Umsetzung von Barrierefreiheit (Siehe Kapitel 4). Diese dient dazu, Informationen und Richtlinien für den Prototyp zu sammeln und anschließend eine kritische Beurteilung der Konkurrenten durchzuführen und eine Einschätzung abzugeben im Kontext der jeweiligen Barrierefreiheit.

Im darauffolgenden Abschnitt wird dann anhand der Analyseergebnisse im Verbund mit den Anforderungen, die sich aus dem Grundlagenkapitel ergeben, ein Anforderungskatalog zusammengetragen. Dieser dient als Orientierung für Entwickler und kann schließlich dazu verwendet werden, Bedienoberflächen hinsichtlich ihrer Wahrnehmbarkeit nachhaltig zu optimieren (Siehe Kapitel 5).

Nachdem der Katalog erarbeitet wurde, kann mit der Implementierung eines Prototyps begonnen werden. Dieser soll Einzug finden in die Low-Code Plattform yeet und wird auf Basis des erarbeiteten Anforderungskataloges entwickelt (Siehe Kapitel 6).

Den Abschluss der Hauptziele dieser Arbeit bildet das Kapitel zur Durchführung eines Usability-Testings. Dieses wird im unmittelbaren Anschluss an die Implementierung durchgeführt und dient der nachhaltigen Qualitätssicherung und Optimierung des Prototyps (Siehe Kapitel 7).

An jenes Kapitel schließt sich ein abschließend Stellung nehmendes Kapitel an (Siehe Kapitel 8), das eine Einschätzung zum erarbeiteten Prototypen abgibt und die aufgestellten Thesen beantwortet.

Zum Schluss folgt noch eine Stellungnahme und Einschätzung zur Ausrichtung von yeet hinsichtlich Barrierefreiheit in der Zukunft (Siehe Kapitel 9). Dabei soll diskutiert werden, welche nächsten Schritte auf den umgesetzten Prototyp folgen und wie sich das Produkt am Markt in Richtung Barrierefreiheit positioniert (Siehe Kapitel 9.2). Nach dem doch sehr produktspezifischen Ausblick folgt noch eine generelle Einschätzung zur Entwicklung der Barrierefreiheit und Softwareergonomie in Zukunft. Dabei soll spekuliert werden, welche Entwicklungsarbeit auf diesem Gebiet geschehen kann und wohin sich das Ganze entwickelt (Siehe Kapitel 9.3).



## Kapitel 2

# Stand der Technologien

### 2.1 Einleitende Worte

Barrierefreiheit hat sich in der Softwareentwicklung längst zu einem sehr wichtigen Thema entwickelt und beschäftigt Entwickler aus aller Welt. Die Integration und Inklusion von Menschen mit Einschränkung in den betriebswirtschaftlichen Prozess ist längst sehr bedeutend geworden und versucht, Betroffene in den Wertschöpfungsprozess zu einzubeziehen und nachhaltig zu unterstützen.

### 2.2 Bereits existierende Arbeiten

Trotz der Relevanz von Barrierefreiheit gibt es bisher wenige Arbeiten, die sich konkret im Kontext der Softwareentwicklung bewegen, sondern eher allgemeiner Natur sind oder sich mit der Barrierefreiheit in der physischen Welt beschäftigen. Dennoch gibt es vereinzelt Arbeiten, die sich mit der Barrierefreiheit von Software befassen und aufzeigen, an welche Richtlinien sich Entwickler halten sollten. Da das Thema Barrierefreiheit allerdings erst in den letzten Jahrzehnten so enorm an Wichtigkeit gewonnen hat, ist noch sehr viel Forschungsarbeit auf diesem Gebiet zu leisten. Auch angesichts der Tatsache, dass die Integrationsquote von Menschen mit Behinderungen in den Arbeitsmarkt signifikant ansteigt, ist es immer wichtiger, in diesem Gebiet zu forschen. So stieg die Erwerbsquote von behinderten Menschen laut Statistischem Bundesamt im Vergleich zum Jahr 2009 auf 52% an [sta21, vgl.]. Der Anteil an erwerbstätigen Menschen mit Behinderung nimmt stetig zu, was auch der Tatsache geschuldet ist, dass die Gesellschaft sich ihrer Barrieren immer bewusster wird und versucht, diese zu minimieren. Trotzdem entspricht längst nicht jede Seite oder Anwendung im Web den gedachten Standards und erschwert somit das Leben für einige Menschen.

### 2.2.1 Arbeit: “Barrierefreiheit betriebswirtschaftlicher Anwendungen - Diskussion spezieller Anforderungen und Lösungsansätze“

Unter anderem haben Beschäftigte der Firma *SAP AG*<sup>1</sup> ein Paper veröffentlicht, welches die speziellen Anforderungen betriebswirtschaftlicher Software, hinsichtlich Barrierefreiheit, diskutiert. Dabei soll dies helfen, bestimmte Punkte bei der Konzeption von betriebswirtschaftlicher Software zu berücksichtigen und eine Art Katalog bieten, der Informationen liefert und die Umsetzung erleichtert. Ähnlichen Zweck soll auch diese Arbeit haben. Anhand des Vergleiches bestehender Produkte auf dem Markt und den Informationen zu Barrierefreiheit aus offiziellen Quellen wie dem W3C soll zum einen die Umsetzung auf dem Low-Code Markt analysiert werden und zum anderen ein Prototyp für einen Modus für Farbblindheit/Sehschwäche entwickelt werden.

Auch die Produkte der Firma Vectorsoft AG sind de facto betriebswirtschaftliche Software, welche ausschließlich in einem Unternehmenskontext zum Einsatz kommt. Daher hat jenes Paper durchaus auch Relevanz für yeet, obwohl der programmatische Low-Code Ansatz, welcher bei yeet verfolgt wird, nicht von Hardt und Schrepp [HS10] berücksichtigt wird. Die beiden Autoren untersuchen anhand einer CRM-Software “ spezielle Probleme und gewählte Lösungsansätze zur barrierefreien Gestaltung der Benutzerschnittstelle“ [HS10, :86].

Dabei stellen sie unter anderem die zu berücksichtigenden Benutzergruppen dar, erklären, wie eine effiziente Nutzung für die relevanten Nutzergruppen möglich ist und erläutern, welche Herausforderungen sich beim Design gerade im Kontext betriebswirtschaftlicher Anwendungen ergeben. Hardt und Schrepp erläutern, dass Computer-Arbeitsplätze behinderten Menschen eine Möglichkeit zu beruflicher Tätigkeit bieten, die nicht im Konflikt zu ihren körperlichen oder geistigen Einschränkungen stehen. Ein Kennzeichen für gelungene Inklusion konstatieren die beiden Autoren als die erfolgreiche Erfüllung spezieller Anforderungen hinsichtlich der Barrierefreiheit der Software. Konkret sollen dabei Hindernisse gemieden werden, welche dem Nutzer der betriebswirtschaftlichen Software die Arbeit erschweren und seine Effizienz mindern. So müssen Nutzer solcher Software diese meist über einen längeren Zeitraum nutzen, was es unerlässlich macht, einerseits alle Funktionalitäten zugänglich zu machen, andererseits aber auch den vollen Funktionsumfang effizient nutzen zu können [HS10, vgl.:86].

Daraus ersichtlich wird, dass es dem Nutzer nichts bringt, wenn dieser zwar alle Funktionen nutzen kann, in der Verwendung dieser aber stark eingeschränkt ist und seine Aufgaben nicht zufriedenstellend beziehungsweise effizient erledigen kann. Um dies zu gewährleisten, sollte das Design der Benutzeroberfläche betriebswirtschaftlicher Anwendungen immer dem universellen Design angepasst werden [HS10, vgl.:87]. Unter diesem Begriff versteht man, Benutzeroberflächen für eine möglichst breite Masse an Nutzern zugänglich zu machen, ob direkt oder mit Hilfswerkzeugen technischer Natur [vn208, vgl.:10].

In ihrem Paper schreiben die beiden Autoren, es sei von großer Wichtigkeit, Barrierefreiheit bereits zu Beginn der Entwicklung von Software zu berücksichtigen, da es sehr schwer sei, diese nachträglich in eine bereits existierende Software einzupflegen. Dies wird als Inspiration genommen, das Thema Barrierefreiheit und die Integration in yeet frühzeitig anzugehen und unter anderem mit einem Modus für Farbblindheit/Sehschwäche zu beginnen. Um die

<sup>1</sup><https://www.sap.com/germany/index.html>

Barrierefreiheit vollständig zu berücksichtigen, solle man entsprechende Anforderungen in der Spezifikation der Benutzerschnittstelle festhalten, um diese später nicht zu vergessen. Dies helfe nicht nur für die Usability, relevante Punkte im Designprozess nicht zu vergessen, sondern erleichtere auch die Arbeit der Designer, da komplexe Designlösungen frühzeitig erarbeitet und erkannt werden können [HS10, vgl.:90]. Die Tatsache, dass die beiden Autoren diesen Prozess anhand einer CRM-Software beschreiben, rechtfertigt den Entwurf dieser Thesis, da signifikante Unterschiede zwischen solch einer Software und einer klassischen Low-Code Plattform bestehen. So sind Low-Code Plattformen in der Regel deutlich komplexer, was zur Folge hat, dass auch die Umsetzung der Barrierefreiheit divergiert und damit eine wissenschaftliche Arbeit benötigt wird, die eine ähnliche Vorgehensweise im Kontext Low-Code erarbeitet.

### 2.2.2 Arbeit “Barrierefreie Web-Nutzung durch blinde und sehbehinderte Menschen“

Da das Themengebiet rund um Low-Code noch recht jung ist, gibt es im Allgemeinen bisher sehr wenige wissenschaftliche Publikationen, welche sich in diesem Rahmen bewegen. Schaut man hingegen nach dem Thema Barrierefreiheit im Kontext Software, so findet man schon ein größeres Angebot an wissenschaftlichen Arbeiten. Da ein Ziel dieser Arbeit ist, einen Prototyp für einen Modus für Farbblindheit/Sehschwäche zu implementieren, sind insbesondere Arbeiten relevant, welche sich mit Sehbehinderungen befassen. Unter anderem wurde im Jahre 2005 eine Professur veröffentlicht, in der sich die beiden Autoren mit der barrierefreien Web-Nutzung durch blinde und sehbehinderte Menschen befassen. Ziel dieser Professur ist es, neben einer klassischen Situationsanalyse Kriterien für die Umgestaltung von geschäftlichen Webseiten hinsichtlich Barrierefreiheit herauszuarbeiten und schließlich einen Leitfaden für die barrierefreie Umgestaltung von Webseiten zu geben [PS05, vgl.:1]. Zu Beginn ihrer Arbeit legen die beiden Autoren die Barrieren der Web-Nutzung durch behinderte Menschen offen und schließen daran eine Erklärung der Motive und Rechtfertigung einer barrierefreien Umsetzung von Webseiten an. Thematisiert werden dabei soziale, politische, gesetzliche und unternehmerische Faktoren, die es rechtfertigen, das Thema Barrierefreiheit oben auf die Agenda zu schreiben. Daraufhin wird im nächsten Kapitel über die beiden unterschiedlichen Möglichkeiten zur Umsetzung geforderter Barrierefreiheit diskutiert und versucht, diese zu erklären. Insbesondere das nächste Kapitel hat große Gewichtung in dieser Arbeit, da dort systematisch versucht wird, einen klassischen Leitfaden zu konzipieren, an dem sich Entwickler orientieren können, die planen, Barrierefreiheit umzusetzen.

Wie ersichtlich wird, wurde die Arbeit zwar in einem etwas divergierenden Kontext erarbeitet, verfolgt allerdings ähnliche Ziele, wie es diese Arbeit tut. Allerdings ist gerade im Bereich der Informationstechnologie eine Arbeit aus dem Jahre 2005 nicht unbedingt mehr der “State of the Art“, da sich dieses Wissensgebiet enorm schnell weiterentwickelt. Gerade deshalb ist diese Arbeit ein guter Anschluss an die Arbeit von Puhl und Schwickert [PS05], da sie sich einerseits im sehr populären Sektor der Low-Code Anwendungen bewegt und andererseits insbesondere spezieller auf die Umsetzung eines Modus für Farbblindheit/Sehschwäche eingeht. Dem liegt eine reichhaltige Grundlage an Informationen über die gängigen Methoden auf dem Markt zugrunde, welche erarbeitet und anhand dessen in Kombination mit Infor-

mationen aus offiziellen Quellen, wie beispielsweise der WCAG (Siehe Kapitel 3.3.4), angereichert und zu einem Katalog an Anforderungen modelliert wird. Anhand der gesammelten Informationen kann dann ein Prototyp entwickelt werden, der den modernen Anforderungen hinsichtlich visueller Wahrnehmung und ihrer Barrieren nachkommt.

### 2.2.3 Arbeit “Web Accessibility for Visually Impaired People: Requirements and Design Issues“

Eine Arbeit, die hingegen recht aktuell ist und sich mit dem Thema der Barrierefreiheit für sehbehinderte Menschen im Web und deren Anforderungen beschäftigt, stammt aus dem Jahre 2016. Diese veröffentlichte Arbeit beschäftigt sich mit den Designanforderungen und den Problematiken, welche mit Sehbehinderungen einhergehen und welche Schwierigkeiten resultieren, wenn sich Webangebote nicht an den Richtlinien der WCAG orientieren und diese verfehlen [FVK<sup>+</sup>16, vgl.:79].

Um die Anforderungen aufzustellen und besser zu verstehen, wurden drei verschiedene Workshops organisiert, deren Ergebnisse darauf hindeuten, dass es keine allgemeingültige barrierefreie Lösung gibt, ohne den Grad der jeweiligen Sehbehinderung zu berücksichtigen und gezielter darauf eingehen zu können. Der erste Workshop diente dazu, Anforderungen der sehbehinderten Nutzer festzustellen, zu sammeln und zu dokumentieren. Die nachfolgenden Workshops hingegen divergierten vom Aufbau und hatten den Zweck, die aus Workshop 1 erarbeiteten Anforderungen zu validieren und anhand von Expertisen zu ergründen und zu spezifizieren [FVK<sup>+</sup>16, vgl.:79ff].

Ergebnisse der Workshops sind unter anderem Anforderungen an die Barrierefreiheit für Webangebote. Aus der Analyse der Daten ergaben sich im Allgemeinen zwei Hauptfaktoren für die Anforderungen an die Oberflächengestaltung in Hinsicht auf Barrierefreiheit und Bedienbarkeit für sehbehinderte Nutzer. Zum einen beeinflussen kontextuelle Faktoren die Wahrnehmbarkeit und Barrierefreiheit von Webinhalten enorm. Dabei spielen Faktoren wie technische Ausstattung, Witterung, Ort, persönliche Technikaffinität und die allgemeine Barrierefreiheit wichtige Rollen und beeinflussen die Wahrnehmung enorm. Als zweiter wichtiger Faktor hat sich ergeben, dass es von großer Wichtigkeit ist, Menschen mit einer Sehbehinderung gezielt zu schulen und ihnen Möglichkeiten im Umgang mit ihrer Behinderung zu offenbaren. Wichtig sei es, diesen Menschen Techniken zu vermitteln und beim Designen der Bedienoberflächen deren besondere Bedürfnisse zu verstehen [FVK<sup>+</sup>16, vgl.:84].

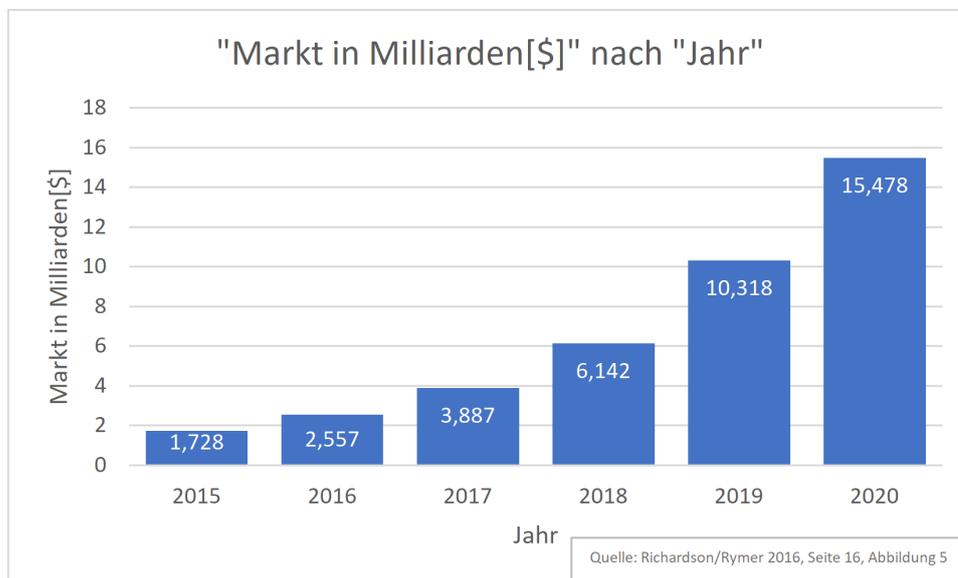
So konnte festgestellt werden, dass es, wie bereits oben erwähnt, kein allgemeingültiges Modell gibt, da die Bedürfnisse der Nutzer divergieren, was bedingt wird durch unterschiedliche Grade einer Behinderung. Deshalb ist es wichtig, die Bedürfnisprofile der Nutzer zu identifizieren und so eine gezielte Lösung für jeden Nutzertyp zu finden. Diese Lösung soll über die Grenzen der klassischen WCAG hinausgehen und den Horizont erweitern können [FVK<sup>+</sup>16, vgl.:85].

Ähnliches Ziel verfolgt auch diese Arbeit, allerdings im speziellen Kontext von Low-Code Plattformen, da dieser Kontext von klassischen Webangeboten divergiert und seine ganz eigenen Charakteristiken aufweist. Zwar können die Ergebnisse auch für den zu erstellenden Prototyp in yeet hilfreich sein, sollen aber speziell durch die Konkurrenzanalyse erweitert und verfeinert werden. Die Schwäche bzw. der Unterschied zu dieser Arbeit ist, dass klassische

### 2.3. Wie weit ist der Entwicklungsstand der Technologien?

Webangebote meist eher statischer Natur sind und nicht eine solch hohe Interaktivität wie Low-Code Plattformen aufweisen. Somit sind die Ziele der unterschiedlichen Webangebote genauso verschieden und damit auch die Anforderungen an die Oberfläche, die es in dieser Arbeit festzustellen gilt.

Anhand der erarbeiteten Ergebnisse aus dem Workshop konnte ein Prototyp als "Middleware" entwickelt werden, der sich acht adaptiver Techniken bedient, um die Nutzung für verschiedene Grade der Sehbehinderung zu verbessern. Eine vollzogene Bewertung der "Middleware" legt nahe, dass die adaptiven Techniken vor allem bei Webangeboten besser funktionieren, die den Anforderungen der WCAG nicht gerecht werden, als bei Webangeboten, die diese bereits erfüllen [FVK<sup>+</sup>16, vgl.:93].



**Abbildung 2.1:** In der folgenden Abbildung sieht man das prognostizierte Wachstum des Low-Code Markts in Milliarden \$. Quelle: In Anlehnung an [RR16, vgl.:16]

### 2.3 Wie weit ist der Entwicklungsstand der Technologien?

Angesichts der steigenden Nachfrage an Software, auch bedingt durch die immer weiter fortschreitende Digitalisierung von Geschäftsprozessen, muss dieser hohen Nachfrage nachgekommen werden. Und so hat das Marktforschungsinstitut Forrester 2016 eine Grafik veröffentlicht, welche den stetig wachsenden Markt von Low-Code Plattformen darstellt (Siehe Abbildung 2.1). Die veröffentlichte Grafik prognostiziert ein enormes Wachstum im Zeitraum von 2015 bis 2020, in der der Markt um fast das 9-Fache anwachsen soll.

Wie heutzutage jede Webseite, Webanwendung und Software sollte auch bei Low-Code Plattformen auf ausreichende Barrierefreiheit und Softwareergonomie gesetzt werden. Denn Low-Code besitzt den Anspruch, die Entwicklungsdauer und Kosten der Softwareentwicklung zu

reduzieren, was allerdings nicht gewährleistet werden kann, wenn potenziell eingeschränkte Nutzer die Anwendungen nicht effizient nutzen können. Eine fehlende Barrierefreiheit könnte daher dem Nutzen von Low-Code potenziell im Wege stehen und die Vorteile des Konzeptes zum Erlöschen bringen. Doch gibt es tatsächlich noch kaum beziehungsweise tatsächlich keine wissenschaftlichen Arbeiten zum Thema der Umsetzung von Barrierefreiheit in Low-Code Plattformen. Zwar werben viele Produkte auf diesem Markt damit, ihren Nutzern eine barrierefreie Oberfläche zu bieten, doch gibt es keine allgemeine Analyse, die es auch neuen Mitbewerbern ermöglicht, sich bei der Integration von ausreichender Barrierefreiheit an den bereits etablierten Produkten der Branche zu orientieren. An dieser Stelle soll diese Arbeit ansetzen und Produkte auf dem Markt vergleichen, um zu ermitteln, wie stark das Thema Barrierefreiheit im Bereich Low-Code berücksichtigt wird und vor allem, welchen Stellenwert sie innehat. Anhand der zusammengetragenen Ergebnisse kann dann mit der Implementierung eines Prototyps für einen Modus für Farbblindheit/Sehschwäche begonnen werden und über die Thesis hinaus auch die restlichen erarbeiteten Punkte der Recherche und Analyse umgesetzt werden.

### **2.4 Ansatzpunkt dieser wissenschaftlichen Arbeit**

Wie bereits aus den vorangegangenen Sektionen ersichtlich wurde, sind sowohl Barrierefreiheit als auch Low-Code zwei Themengebiete, die aktuell von recht großer Relevanz in der Informatik sind und es vermutlich auch noch für einen längeren Zeitraum bleiben werden. Allerdings gibt es bisher keine wissenschaftlichen Publikationen, welche beide Themen in einer gemeinsamen Arbeit zusammen betrachten, also von der Barrierefreiheit in Low-Code Plattformen handeln und sich mit dieser beschäftigen. Man könnte argumentieren, dass es für die Barrierefreiheit von Webseiten und Webanwendungen genug Arbeiten gibt, die sich mit dieser Thematik befassen, wie es auch Puhl und Schwickert getan haben, allerdings divergieren der Aufbau und die Komplexität von Low-Code Plattformen doch sehr stark von klassischen Webseiten und sollten somit in eigenen Arbeiten näher untersucht werden.

Durch den schnell wachsenden Markt arbeiten zukünftig immer mehr Menschen mit solchen Anwendungen, was im Umkehrschluss bedeutet, dass auch immer mehr eingeschränkte und behinderte Nutzer potenzielle Anwender solcher Anwendungen sein werden. Daher ist es umso wichtiger, mithilfe einer solchen Forschungsarbeit das Gebiet im Kontext Low-Code zu analysieren und Hilfestellung zu leisten. Insbesondere zum Thema Sehbehinderung gibt es außerdem noch sehr wenige Arbeiten, die dies im Kontext von Software oder Webseiten behandeln und auch daher ist der zu entwickelnde Prototyp als Output dieser Arbeit ein beispielhaftes Vorgehen, an dem sich andere Entwickler zukünftig orientieren oder lediglich inspirieren lassen können. Denn wie es bereits viele Webseiten oder Videospiele vormachen, ist ein Modus für Farbblindheit/Sehschwäche, aber auch allgemein Barrierefreiheit, sehr wichtig und nützlich für das Produkt und seine Anwender. Auch kann die vorausgegangene Analyse verschiedener Low-Code Plattformen auf dem Markt Erkenntnis darüber liefern, inwiefern andere Produkte dies bereits berücksichtigen, ob Barrierefreiheit in Low-Code Plattformen ausreichend gewährleistet wird oder ob in diesem Sektor vielleicht Aufholarbeit zu leisten ist

und diese Arbeit einen beispielhaften Ansatz bieten kann.

Anders als die vorgestellten verwandten Arbeiten bewegt sich diese Arbeit also vollständig im Kontext von Low-Code und soll nachhaltig nicht nur Informationen für andere Entwickler liefern, sondern auch das Produkt der Firma Vectorsoft AG positiv verändern und damit die spätere Zufriedenheit der Nutzer sichern. So soll dank der Analyse gezeigt werden, was Konkurrenzprodukte auf dem Gebiet der Barrierefreiheit leisten, damit dies entweder adaptiert oder optimiert umgesetzt werden kann. Dies dient auch dazu, später eine echte Konkurrenz darzustellen oder möglicherweise eine Vorreiterrolle einnehmen zu können und das Ganze als USP zu verkaufen.



# Kapitel 3

## Grundlagenkapitel

### 3.1 Intro

Als Grundlage für die Erarbeitung eines Prototyps gilt es, im folgenden Kapitel Begrifflichkeiten und Konzepte zu erklären, welche relevant für die späteren Kapitel der Arbeit sind und für Verständnis auf Seite des Lesers sorgen sollen. Falls der Leser im Laufe des Leseflusses also Informationen über ein Thema benötigt, können diese in diesem Kapitel abgerufen werden.

### 3.2 Softwareergonomie

#### 3.2.1 Computersysteme im Alltag

Im Rahmen der Digitalisierung werden immer mehr Bereiche unseres alltäglichen Lebens von Technik durchzogen. So begegnen wir in allgemeinen Lebensbereichen, wie Arbeit, Bildung und Freizeit, einer Vielzahl an Computersystemen und Computeranwendungen [Her18, vgl.:1].

Hertzeg spricht in seinem Werk zu Softwareergonomie von einer Wechselwirkung von Computersystemen und dem Menschen und bezeichnet diese deshalb auch als "interaktive Computersysteme" [Her18, :1]. Diese tiefgehende Vernetzung des Menschen und weitgehende Durchdringung unseres alltäglichen Lebens erweitert die Lebensgrundlage vieler Menschen in unserer heutigen Gesellschaft. Damit geht eine sich immer weiter entwickelnde Abhängigkeit einer hürdenfreien Nutzung solcher interaktiver Computersysteme einher. Dies soll garantieren, dass Menschen unter jeglicher Voraussetzung diese Computersysteme sicher bedienen können und gegebenenfalls über geeignete Schulungen in der Nutzung unterrichtet werden. Damit wird sichergestellt, dass eine größere Menge an Menschen die Software und die Computersysteme effizient und gewinnbringend nutzen kann. Heutzutage sind die berufliche Funktion, die persönlichen Entfaltungsmöglichkeiten, die soziale Einbindung sowie das alltägliche Wohlbefinden zunehmend abhängig vom problemlosen und wirkungsvollen Umgang mit Computersystemen [Her18, vgl.:1ff].

#### 3.2.2 Interaktivität von Computersystemen

Wie bei einem klassischen Schweizer Taschenmesser, welches einen großen Katalog an Funktionen bietet, vergleicht Herzeg diese mit einer Bedienoberfläche eine Computeranwendung. Denn mit einem sich erweiternden Funktionsumfang muss außerdem die Handhabung gewährleistet werden, damit das Werkzeug effizient genutzt werden kann. So muss die Bedienoberfläche hinsichtlich seiner Formgebung an seine Funktionen angepasst werden, um nicht bei seiner Wirksamkeit und damit der Nutzungsqualität einzubüßen [Her18, vgl.:4]. Eine "Bedienoberfläche bildet die Form oder Gestalt eines computerbasierten Werkzeugs" [Her18, :5] und kann in Kombination mit den Nutzungsregeln als Benutzungsschnittstelle definiert werden. Dabei sichert diese Schnittstelle die Verbindung von Mensch und Computer in ähnlicher Weise, wie technische Schnittstellen die Verbindung zwischen technischen Systemen [Her18, vgl.:5]. Durch die zunehmende Integration von Technik in den menschlichen Alltag erhöht sich die Frequenz der Nutzung und die Menge an Wechselwirkung zwischen Mensch und Computer. Dies deutet Herzeg mit dem Begriff der "Interaktivität" [Her18, :5].

#### 3.2.3 Teildisziplin Softwareergonomie

Computerbasierte Systeme zeichnen sich als Werkzeug durch eine hohe Reaktivität aus. Sie reagieren auf Input und erzeugen einen entsprechenden Output, der bedingt wird durch innere Prozesse und Vernetzungen. Die Besonderheit des Zusammenspiels der Interaktivität mit der Multimedialität dieser Werkzeuge bedingen eigene damit verbundene Herstellungs- und Entwicklungsmethoden sowie spezielle Nutzungsbedingungen. All dies rechtfertigt die Entstehung einer eigenen Teildisziplin mit dem Namen Softwareergonomie. Diese Teildisziplin der klassischen Ergonomie liefert Entwicklern Theorien und Methoden, die dabei unterstützen, computerbasierte Werkzeuge in benutzer- und anwendungsgerechter Weise zu konzipieren, zu realisieren und zu testen. Entspricht eine Software den grundlegenden ergonomischen Anforderungen, so kann diese als gebrauchstauglich beschrieben werden. Daraus schließt Herzeg, dass Softwareergonomie sich der Gebrauchstauglichkeit interaktiver Computersysteme widmet. Wie eingangs erwähnt, reicht es allerdings nicht nur, wenn die Software den ergonomischen Kriterien entspricht, während die Hardware die Nutzung dieser wiederum erschwert. Und so geht Softwareergonomie immer auch mit einer benutzer- und anwendungsgerechten Gestaltung der Hardware sowie des Arbeitsplatzes einher. Gemeinsames Ziel ist dann, einen angemessenen räumlichen und zeitlichen Nutzungskontext für den Nutzer zu schaffen [Her18, vgl.:7].

#### 3.2.4 Computer-Ergonomie

Insofern gilt es die sogenannte Hardwareergonomie zu berücksichtigen, welche dem Nutzer die "technischen Rahmenbedingungen und Optionen der Gestaltung eines interaktiven Systems liefert" [Her18, :7]. Herzeg fasst die beiden Begriffe der Software- und Hardwareergonomie unter dem gemeinsamen Begriff der "Computer-Ergonomie" [Her18, :7] zusammen.

### 3.2.5 Wirtschaftlicher Nutzen

Mangelt es an Ergonomie, kann dies verschiedene Ursachen haben und beeinträchtigt die Effizienz und Effektivität der Arbeit der Nutzer erheblich. Simpel formuliert kann daraus geschlossen werden, dass schlechte Arbeitsmittel auch schlechte Ergebnisse nach sich ziehen. Aus betriebswirtschaftlicher Sicht macht Herzeg die Wirtschaftlichkeit von Software auch abhängig von der Gebrauchstauglichkeit [Her18, vgl.:10].

### 3.2.6 Verschwimmende Grenzen zwischen Arbeit und Freizeit

Wie bereits in den oberen Abschnitten verdeutlicht, beschreibt die Softwareergonomie also, wie gebrauchstauglich eine Software gestaltet ist. Dabei lässt sich der Begriff der Gebrauchstauglichkeit noch einmal unterteilen in drei Unterbegriffe, die diesen ausmachen. Effektivität, Effizienz und Zufriedenstellung wurden dabei in einer internationalen ISO-Norm DIN EN ISO 9241-118 festgehalten und "zu den Hauptmerkmalen gebrauchstauglich gestalteter Computeranwendungen erhoben" [Her18, :10]. Doch nicht nur am Arbeitsplatz macht es Sinn, sich mit Softwareergonomie zu beschäftigen. Auch im Alltag verwendet der Mensch immer mehr Software und die Grenzen zwischen Arbeitsplatz und Freizeit beginnen immer weiter zu verschwimmen, sodass kaum noch strikt unterschieden werden kann. Somit gewinnt die Softwareergonomie über die Grenzen des Arbeitsplatzes hinweg immer weiter an Bedeutung und Relevanz. Sie findet insbesondere immer öfter in Anwendungen für mobile Systeme, Webseiten und Haushaltsgeräten oder Computerspielen einen Platz. Durch das Verschwimmen der Grenzen zwischen Arbeit und Freizeit kann der Kontext nicht mehr rein auf die Arbeit bezogen werden und tritt in unterschiedlichsten Situationen auf [Her18, vgl.:11].

### 3.2.7 Interaktionsdesign

Die Umsetzung der Gebrauchstauglichkeit definiert Herzeg mit dem Begriff des "[sic] Interaktionsdesign" [Her18, :12]. Ziel dieser Arbeit ist es, Computersysteme an ihre Benutzer, die Aufgaben und den sich wechselnden Nutzungskontext gezielt anzupassen. So beschreibt Herzeg, dass sich nicht die Benutzer an die Computersysteme anpassen sollen, sondern es das Ziel ist, dass Computersysteme benutzer- und anwendungsgerecht gestaltet werden, um für vordefinierte Aufgaben der Benutzer, optimalen Komfort und Effizienz zu bieten. Herzeg definiert Interaktionsdesign als eine eigene Disziplin, die mit Hilfe der von der Softwareergonomie gelieferten Kriterien qualitativ gute Gestaltungslösungen bietet [Her18, vgl.12ff].

### 3.2.8 Gebrauchstauglichkeit

Der Begriff der sogenannten Gebrauchstauglichkeit ist für das Konzept der Softwareergonomie unerlässlich und wird in der ISO 9241-11 definiert, als "[d]as Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen" [Her18, :210]. Die drei Teilkriterien, welche den Begriff der Gebrauchstauglichkeit ausmachen, lassen

sich jeweils eigens definieren. Unter der bereits erwähnten Effektivität wird die Genauigkeit und Vollständigkeit verstanden, mit welcher der Benutzer ein bestimmtes Ziel erreichen kann. Der zweite charakteristische Unterbegriff ist die Effizienz, welche definiert wird aus der Menge der Aktionen und Artikulationen und vorgibt, diese auf ein mögliches Minimum zu reduzieren [Her18, vgl.:212], also faktisch Software oder Computersysteme so effizient wie möglich nutzen zu können. Das Trio komplettiert die sogenannte Zufriedenstellung, welche laut der ISO 9241-11 als "Freiheit von Beeinträchtigungen und positive Einstellungen gegenüber der Nutzung des Produkts" verstanden und definiert wird [Her18, :214]. Neben diesen drei Teilkriterien, gibt es diverse wichtige Aspekte, die dazu führen, eine Software gebrauchstauglich für den Nutzer zu gestalten. Als unabdingbare Voraussetzung bei der Gestaltung von Software kann die Wahrnehmbarkeit von Informationen definiert werden, welche befähigt, höhere Verarbeitungsleistungen zu vollziehen [Her18, vgl.:240]. Um eine gute Wahrnehmbarkeit zu erreichen, sind Prinzipien wie Lesbarkeit, Differenzierung von verschiedenen Informationen und Übersichtlichkeit unabdingbar. Außerdem ist die Orientierungsförderlichkeit sehr relevant, welche sich damit beschäftigt, "inwieweit eine Systemgestaltung Benutzer befähigt, den Überblick über Funktions- und Objektstrukturen zu gewinnen und zu erhalten" [Her18, :242]. Der gelungenen Gestaltung der Informationen und einheitlichen Struktur des Aufbaus der Benutzeroberfläche, schließt sich die Lenkbarkeit der Aufmerksamkeit an. Diese beschäftigt sich damit, wie mithilfe von bestimmten stilistischen Mitteln die Aufmerksamkeit des Nutzers gelenkt werden kann. Das letzte Prinzip, welches den Katalog an Anforderungen komplettiert, ist die Handhabbarkeit, die schlichtweg dafür sorgt, dass Funktionalitäten der Software auch optimal genutzt werden können und die Arbeit effizient, effektiv und zufriedenstellend vollzogen werden kann [Her18, vgl.:241-244].

## 3.3 Barrierefreiheit

### 3.3.1 Der Begriff der Behinderung

Seit jeher sind Menschen mit gewissen Einschränkungen Teil unserer Gesellschaft, die sie auf unterschiedliche Arten im alltäglichen Leben behindern und einschränken. Diverse Bewegungen und gesetzliche Richtlinien versuchen, Sorge zu tragen, dass dagegen etwas unternommen wird und den Betroffenen eine Hilfestellung geleistet wird, um die Einschränkungen in gewisser Hinsicht zu überkommen. Dabei kommt es sehr häufig dazu, dass man im alltäglichen Sprachgebrauch bei dem Begriff der Behinderung davon ausgeht, dass diese von körperlicher oder geistiger Natur ist. Dabei kann eine Behinderung in vielerlei Hinsicht auftreten und beschränkt sich keinesfalls nur auf geistige oder physische Fähigkeiten. Ziel gesetzlicher Richtlinien und menschlicher Bemühungen ist es, Barrieren für Menschen mit einer Behinderung zu minimieren. Im deutschen Sprachgebrauch wird häufig der Begriff "barrierefrei" mit dem Begriff "behindertengerecht" synonymisiert, was die Ursache vermutlich darin trägt, "dass die maßgebliche Barrierefreie Informationstechnikverordnung (BITV) eine Rechtsverordnung zum Gesetz zur Gleichstellung behinderter Menschen, kurz Behindertengleichstellungsgesetz (BGG), ist " [The16, :43]. Trotz der Tatsache, dass Menschen mit unterschiedlichsten Einschränkungen von einer barrierefreien Gestaltung profitieren, ist der Grundsatz dieser beiden Begriffe am Ende doch ein unterschiedlicher. Wie bereits zu Ab-

schnittsbeginn erwähnt, beschränken sich Behinderungen keinesfalls nur auf physische und geistige Einschränkungen. So können sie in diverser Gestalt auftreten und so auch als technische Hürden einen gesunden Menschen in seinen Fähigkeiten limitieren und seine Leistung minimieren [The16, vgl.:43ff].

#### 3.3.2 Unternehmerische Sicht

Ziel eines jeden betriebswirtschaftlichen Unternehmens sollte es sein, gewinnorientiert Ergebnisse zu erzielen, die nachweislich einen Mehrwert für das Unternehmen bieten. Sei dies finanzieller, personeller, sozialer, ökonomischer oder ökologischer Natur, um nur ein paar Begriffe zu nennen, steht der Mehrwert im Fokus. Auch das Thema Barrierefreiheit ist aus unternehmerischer Sicht ein sehr relevantes und spannendes Thema. Aus Sicht eines Unternehmens ermöglicht eine umfassende Barrierefreiheit, dass zwangsläufig mit dem Produkt deutlich mehr potenzielle Kunden angesprochen und überzeugt werden können [The16, vgl.:44].

#### 3.3.3 Barrierefreiheit im Web

Prinzipiell kann jeder Webauftritt enorm von Barrierefreiheit profitieren und eine große Zahl an neuen Kunden für das Unternehmen und seine Produkte begeistern. Zu diesen zählen dann nicht ausschließlich "Menschen mit schweren und permanenten Einschränkungen" [The16, :44], sondern auch Menschen, die etwa unter einer temporären Behinderung leiden, Analphabeten, Legastheniker oder Senioren, können so besser erreicht und für die Produkte des Unternehmens gewonnen werden. Aber auch die Minimierung von Problematiken bedingt durch technische Voraussetzungen sollte berücksichtigt werden, da gerade die Technik sehr stark limitieren kann. Ebenfalls relevant sind umgebungstechnische Einschränkungen, die den Menschen in seiner Handlungsfähigkeit limitieren und so sollten diese daher berücksichtigt werden, um die Zielgruppe zu erweitern. Daraus lässt sich allgemein fassen, dass es das Ziel aus unternehmerischer Sicht ist, das eigene Produkt für eine möglichst breite Masse an Kunden attraktiv zu machen und mehr Kunden zu gewinnen. Der Erfolg der Umsetzung dieses Vorhabens und damit der Grad der Barrierefreiheit lassen sich über diverse Plug-ins und Tools genauestens prüfen. Erzielt das Produkt dabei ein möglichst gutes Ergebnis, so kann sich dies nachhaltig positiv auf die *SEO* auswirken und dafür sorgen, dass beispielsweise der Webauftritt ein höheres Ranking für eine Suche in populären Suchmaschinen hat. Außerdem ist barrierefreie Gestaltung der Benutzerschnittstelle bei gewerblichen Webauftritten verpflichtend und wird innerhalb von gesetzlichen Richtlinien näher definiert und kann Hilfestellung liefern, an was sich das Unternehmen zu halten hat [The16, vgl.:44]. Dabei muss auf diverse Einschränkungen geachtet werden, die möglicherweise bei potenziellen Nutzern anzutreffen sind. Besonders interessant sind für den zu entwerfenden Prototyp vor allem visuelle Wahrnehmungsstörungen wie beispielsweise Farbfehlsichtigkeit (Siehe Kapitel 3.4).

#### 3.3.4 W3C

Eine der großen Vereinigungen, welcher die Standardisierung der Techniken im Internet obliegt, ist das World Wide Web-Consortium. Ziel dieser Bemühungen ist es, im Allgemeinen,

“die Zugänglichkeit von Software und Internetseiten zu erhöhen“ [The16, :52] und durch die Standards so einen gemeinsamen Konsens zu schaffen. Dabei hat die Vereinigung diverse Richtlinienkataloge erstellt, welche Hinweise liefern, wie Inhalte barrierefrei gestaltet werden sollen [The16, vgl.:52].

#### **WCAG**

Die sogenannten WCAG, also Web Content Accessibility Guidelines, haben das Ziel, “einen gemeinsamen Standard für die Barrierefreiheit von Webinhalten zur Verfügung zu stellen, der die Bedürfnisse von Einzelpersonen, Organisationen und Regierungen auf internationaler Ebene erfüllt“ [wca08]. Bereits im Jahre 1991 wurde eine erste Fassung veröffentlicht und später zur Version 2.0 aktualisiert, welche seit Dezember 2008 existiert und im Jahre 2018 mit der bis dato gültigen Version 2.1 aktualisiert wurde. Die Richtlinien stützen sich dabei auf vier Grundprinzipien, die den Kern bilden [The16, vgl.:53].

1. Wahrnehmbarkeit
2. Bedienbarkeit
3. Verständlichkeit
4. Robustheit

Diese Grundprinzipien werden jeweils noch einmal durch 12 Richtlinien genauer spezifiziert, wobei pro Richtlinie testbare Erfolgskriterien mitgegeben werden. Geordnet nach Priorität, lassen sich diese Erfolgskriterien nach den Konformitätsstufen A, AA und AAA einteilen [The16, vgl.:53]. Jene Erfolgskriterien lauten wie folgt:

1. “Muss-Kriterien“ - Anforderungen, die immer erfüllt sein müssen (A)
2. “Soll-Kriterien“ - Anforderungen, welche signifikante Hindernisse beseitigen (AA)
3. “Kann Kriterien“ - Anforderungen, welche weitere Erleichterungen für den Nutzer bringen (AAA)

Um den Entwicklern Hilfestellung zu leisten und die Standardisierung zu erleichtern, formuliert die WCAG etwa 200 Vorschläge zur Realisierung, um nachhaltig die Erfolgskriterien umzusetzen. Aufgrund dieser Struktur der WCAG kann diese als technikübergreifend begriffen werden und ist somit auch in Zukunft noch eine hilfreiche Anleitung zur nachhaltigen Verbesserung der Barrierefreiheit im Web [The16, vgl.:53].

#### **3.3.5 BITV**

Während die WCAG Vorgaben für staatliche Informationsangebote der Mitgliedsstaaten der EU enthalten, wurde 2002 in Deutschland ein bedeutender Schritt gemacht hinsichtlich der Umsetzung dieser Vorgaben in nationales Recht in Form der BITV. Diese Rechtsverordnung

zu barrierefreier Informationstechnik (BITV) spezifiziert in § 11 die "Barrierefreiheit in der Informationstechnologie" als Teil des Gesetzes zur Gleichstellung behinderter Menschen, genauer [The16, vgl.:54]. Bedingt durch die föderale Struktur der Bundesrepublik Deutschland gilt die BITV allerdings nicht auf Landesebene und wird dort durch " die jeweiligen Landesgleichstellungsgesetze(LGG) " [The16, :54] ersetzt, die sich aber in der Regel damit gleichen und kaum divergieren.

## 3.4 Farbblindheit / Sehschwäche

### 3.4.1 Menschliche Farbwahrnehmung

Möchte der Mensch einen Gegenstand wahrnehmen, muss dieser Licht eines bestimmten Spektrums an Wellenlängen reflektieren. Das menschliche Auge kann nur Dinge wahrnehmen, von denen Photonen der richtigen Energie ausgehen, welche schließlich in unserem Auge zu neuronalen Signalen verarbeitet werden. Dabei ist es irrelevant, ob die Gegenstände als Quelle fungieren oder lediglich durch Reflexion oder Transparenz Lichtteilchen aussenden. Als wichtigste Quelle für das menschliche Sehen fungiert die Sonne in Form der Lichtquelle, welche ein Gemisch aus unterschiedlichen Wellenlängen des gesamten sichtbaren Spektrums emittiert, welches wir als neutrales Weiß empfinden. Dass wir nicht alles als weißen Gegenstand wahrnehmen, liegt der Tatsache zugrunde, dass Objekte Licht unterschiedlicher Wellenlänge verschieden stark reflektieren bzw. absorbieren, wodurch wir diese als bunt wahrnehmen können [WL12, vgl.:233].

Um dieses Licht überhaupt wahrnehmen zu können, absorbieren spezielle Fotorezeptoren die Photonen, die in unser Auge fallen und lösen in Folge dessen ein Nervensignal aus. Diesen Prozess der Wandlung von Energie zu einem Reiz in einer Rezeptorzelle nennt man auch Transduktion. Von den verantwortlichen Fotorezeptoren, welche für diesen Prozess zuständig sind, besitzt der Mensch 2 Arten, welche jeweils andere Funktionen ausüben. Die sogenannten Stäbchen und Zapfen sitzen auf der Netzhaut und sind unterschiedlich in ihrer Anzahl verteilt [WL12, vgl.:239].

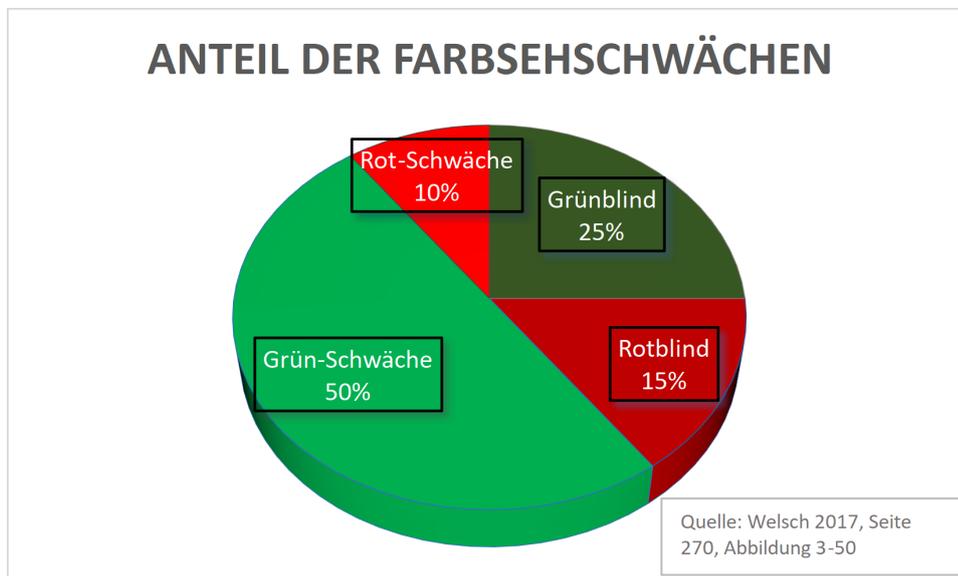
Von den Stäbchen besitzt der Mensch in etwa 120 Millionen pro Auge und damit stellen sie "den häufigsten Sehzellentypus" [WL12, :240]. Diese Sehzellen sind extrem lichtempfindlich und ermöglichen dem Menschen skotopisches Sehen, also das Wahrnehmen von Lichtreizen, was das Hell-/Dunkel-Sehen ermöglicht. Sie enthalten Moleküle des Sehfärbstoffs Rodopsin, der durch die Absorption von Photonen zerfällt und infolgedessen ein elektrisches Signal sendet. Vom anderen Typus der Sehzellen, den sogenannten Zapfen, besitzt der Mensch nur etwa 6 Millionen pro Auge, wobei diese für das sogenannte photopische Sehen verantwortlich sind, also das Wahrnehmen von Farben. Aus der Kombination dieser beiden Sehzellen und den chemischen Prozessen ergibt sich für den Menschen die Fähigkeit, zu sehen. Tatsächlich können Zapfen und Stäbchen allerdings nicht zwischen Licht verschiedener Wellenlängen unterscheiden. Dies wird erst möglich dadurch, dass es unterschiedliche Typen gibt, welche sich hinsichtlich ihrer Absorptionsspektren, ihrer lichtempfindlichen Farbstoffe und der Verarbeitung dieser Nervensignale unterscheiden [WL12, vgl.:240]. Aufgrund ähnlicher Absorptionskurven der Rot- und Grünzapfen ist der höchstmögliche Kontrast im Rot-/Grün-Kanal sehr gering. Limitierungen in unserem Sehapparat führen dementsprechend dazu, dass eine

Farbe ein gewisses Gesichtsfeld einnehmen muss, um vom Menschen beurteilt zu werden [WL12, vgl.:261].

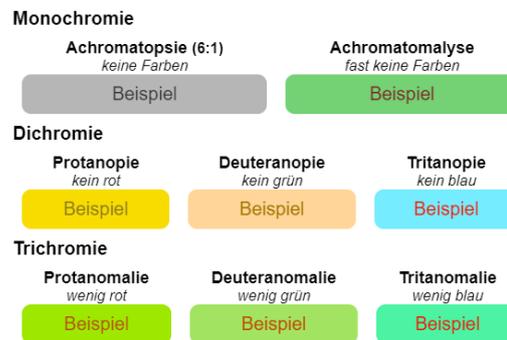
### 3.4.2 Farbblindheit

Ein gesunder Mensch kann unzählig viele Farben des Spektrums wahrnehmen und verarbeiten. Allerdings sind schätzungsweise etwa 0.4% aller Frauen und 8% aller Männer in ihrer Farbwahrnehmung gestört und nehmen Farben nicht so wahr, wie 95% aller anderen Menschen dies tun. Diese Störung der Farbwahrnehmung kann aufgrund einer Krankheit oder angeborenen Schwäche hervorgerufen werden und tritt in diversen Formen auf, die genetisch rezessiv und geschlechtsgebunden von Generation zu Generation vererbt werden. Das für diese Anomalie verantwortliche X-Chromosom wird von 15% aller Frauen an ihre Kinder übertragen. Und während dieses fehlerhafte Chromosom bei Jungen nicht korrigiert wird, passiert dies bei Mädchen in aller Regel schon. Aufgrund dessen kann jene Anomalie bei Frauen nur entstehen, wenn beide Elternteile diese Anomalie ebenfalls besitzen. Eine Form der Farbblindheit ist die totale Farbblindheit, also das rein monochromatische Sehen, bei dem der Betroffene nur Schwarz und Weiß wahrnimmt, also lediglich die Helligkeitsunterschiede der unterschiedlichen Farben. Ist hingegen nur die Wahrnehmung einer der drei Grundfarben Rot, Grün und Blau gestört, was bei etwa 60% der Farbfehlsichtigen auftritt, so wird im Fachjargon von Dichromaten gesprochen, also sogenannten Zweifarbsichtigen.

Diese lassen sich dabei im Allgemeinen in folgende Gruppen einteilen, die auch in Abbil-



**Abbildung 3.1:** In der folgenden Abbildung werden die unterschiedlichen Formen von Farbschwächen in einem Tortendiagramm dargestellt, wobei die Blau-Schwäche und Blau-Blindheit sowie die totale Farbenblindheit nicht dargestellt werden, da ihr Vorkommen extrem selten ist. Quelle: In Anlehnung an [WL12, vgl.:270]



**Abbildung 3.2:** Überblick über eine Demonstration der Sicht mit den unterschiedlichen Arten der Farbfehlsichtigkeit bei rotem Vordergrund auf grünem Hintergrund. Quelle: Eigene Darstellung

Abbildung 3.2 simuliert, präsentiert werden:

- Protanopie - Rotblindheit
- Protanomalie - Störung der Wahrnehmung von Rottönen
- Deuteranopie - Grünblindheit
- Deuteranomalie - Störung der Wahrnehmung von Grüntönen
- Tritanopie - Blaublindheit
- Tritanomalie - Störung der Wahrnehmung von Blautönen

Dabei kann unterschieden werden zwischen Protanopen und Deuteranopen, Rot-Grün-Blinden "wobei Protanopen hohe intensive langwellige Strahlung zur Erkennung benötigen, sowie Tritanopen, Gelb-Blau-Blinde" [WL12, :269]. Protanopen und Deuteranopen verwechseln Farben wie Rot, Gelb, Braun und Grün, können keinen Unterschied zwischen Violett und Blau feststellen und speziell Protanopen sehen dunkelrot lediglich Schwarz. Farbblinde, welche der Gruppe der Tritanopen angehören, haben hingegen Schwierigkeiten, Blau von Grün und Gelbgrün von Grau zu unterscheiden. Menschen, die hingegen unter keinerlei Beeinträchtigung ihrer Farbwahrnehmung leiden, bezeichnet man als sogenannte Trichromaten, also "Menschen ohne Farbsinnesstörung und mit normaler Spektralempfindlichkeit" [WL12, :270], wobei auch bei diesen Anomalien auftreten können, die sie Farben etwas anders wahrnehmen lassen als die Mehrheit. In Abbildung 3.1 stellt Welsch in einem Tortendiagramm die Vorkommenswahrscheinlichkeit der häufigsten Formen der Farbsehschwächen dar, wobei Störungen in der Wahrnehmung von Blau, eine Blau-Blindheit oder die totale Farbblindheit nicht dargestellt werden, da ihr Vorkommen unfassbar selten ist [WL12, :269ff].

## 3.5 Low-Code

### 3.5.1 Begriffserklärung

Im Laufe der Zeit, haben sich viele Ansätze entwickelt, mit denen Programmierer Webseiten und Webanwendungen entwerfen. Es ist ein sich ständig weiterentwickelnder Prozess, der die Programmierarbeiten von Entwicklern erleichtern und verbessern soll. Seit jeher gibt es das Bestreben, sich die Arbeit zu erleichtern, um effizienter und besser einer Tätigkeit nachgehen zu können. Einer dieser Ansätze ist der sogenannte Low-Code Entwicklungsansatz, bei dem man in der Regel mit wenigen Programmierkenntnissen auskommt und welcher die Entwicklung komplexer Anwendungen vereinfachen soll. Die fortschreitende Digitalisierung bringt nicht ausschließlich Vorteile mit sich und so stehen kleine bis mittlere Unternehmen häufig vor der Schwierigkeit, mit großen Konzernen mithalten zu können. Allerdings mangelt es jenen Unternehmen meist an einer guten IT-technischen Infrastruktur bedingt durch einen Fachkräftemangel in der IT-Abteilung. An dieser Stelle soll der Low-Code Ansatz Hilfestellung leisten und gewinnt im Zusammenhang mit der Digitalisierung und *Industrie 4.0* immer mehr an Bedeutung. Dabei gelten Low-Code Plattformen "als wesentliche Bausteine für die flächendeckende digitale Transformation von Unternehmen" [Sch20]. Mit einer Art Baukastenprinzip lassen sich beispielsweise per simplem Drag and Drop Softwarelösungen nach eigenen Anforderungen produzieren, ohne dass die Arbeiter größere Programmierkenntnisse benötigen. Auch die Low-Code Plattformen unterscheiden sich jeweils noch einmal untereinander und setzen von Produkt zu Produkt andere Schwerpunkte [Sch20, vgl.]. Allerdings ist der Grundaufbau trotzdem meist identisch und so bestehen Low-Code Plattformen in der Regel aus folgenden 3 Schichten:

1. Präsentationsschicht
2. Schicht für Geschäftslogik
3. Datenschicht

Diese bilden die Grundbausteine einer klassischen Low-Code Plattform. Während die Präsentationsschicht und die Geschäftslogik auf graphische Editoren setzen, liefert die Datenschicht den beiden darüberliegenden Schichten die Daten. Die sogenannte Präsentationsschicht ermöglicht es dem Nutzer dabei, das User-Interface zu gestalten, während in der Geschäftslogik Geschäftsprozesse und geschäftliche Entitäten angelegt werden. Ein großes Erkennungsmerkmal von Low-Code ist das Prinzip *WYSIWYG*, bei dem der Nutzer im Vordergrund gestaltet und Quellcode im Hintergrund erzeugt wird, um diesem unmittelbares visuelles Feedback zu geben. "Die visuellen Editoren erstellen den Code und übernehmen ebenso die Definitionen für die Oberfläche" [Sch20], während die Datenschicht Daten managed und Tabellen befüllt und dies oftmals auch über einfachere visuelle Editoren realisiert, um Zeit und Kosten zu senken [Sch20, vgl.].

### 3.5.2 Erleichterung durch Low-Code

Das Grundprinzip von Low-Code ist es also, dem Nutzer die komplexen Hintergrundprozesse zu verbergen und die Erstellung von Webanwendungen so simpel wie möglich zu halten. Die Erstellung des Quellcodes erfolgt automatisch, während der Nutzer über eine graphische Oberfläche seine Aktionen durchführt. Dies dient einerseits dazu, die Programmierarbeit für den Endnutzer zu ersparen, die Erstellung der Anwendungen so einfach wie möglich zu machen und andererseits dazu, dem Nutzer trotzdem die Fähigkeit zu lassen, alle wichtigen Anwendungen für die Herausforderungen des Unternehmens zu entwickeln [Sch20, vgl.].

## 3.6 yeet

### 3.6.1 Unternehmen: Vectorsoft AG

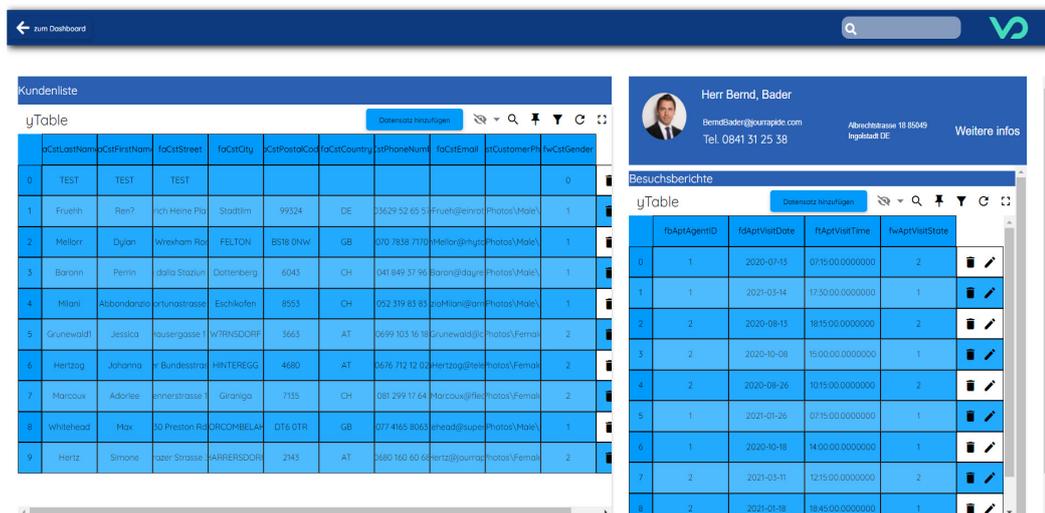
Das mittelständische und international erfolgreiche IT- Unternehmen wurde vor über 35 Jahren gegründet. Der Firmensitz befindet sich in Heusenstamm bei Frankfurt am Main. Vectorsoft entwickelt und vermarktet als Hersteller die Entwicklungsumgebung concept 16. Mit integrierter Datenbank und Frontend-Designer ist eine effiziente Anwendungsentwicklung per *Rapid Application Development (RAD)* einschließlich *SaaS (Software as a Service)* realisiert. Mit yeet entsteht derzeit eine innovative Plattform für den Low-Code-Entwickler. Das Team von vectorsoft unterstützt von Deutschland und der Schweiz aus Anwender in aller Welt. Die hochwertige und flexible Betreuung ihrer Kunden erfährt seit jeher besondere Aufmerksamkeit.

### 3.6.2 Produkterklärung

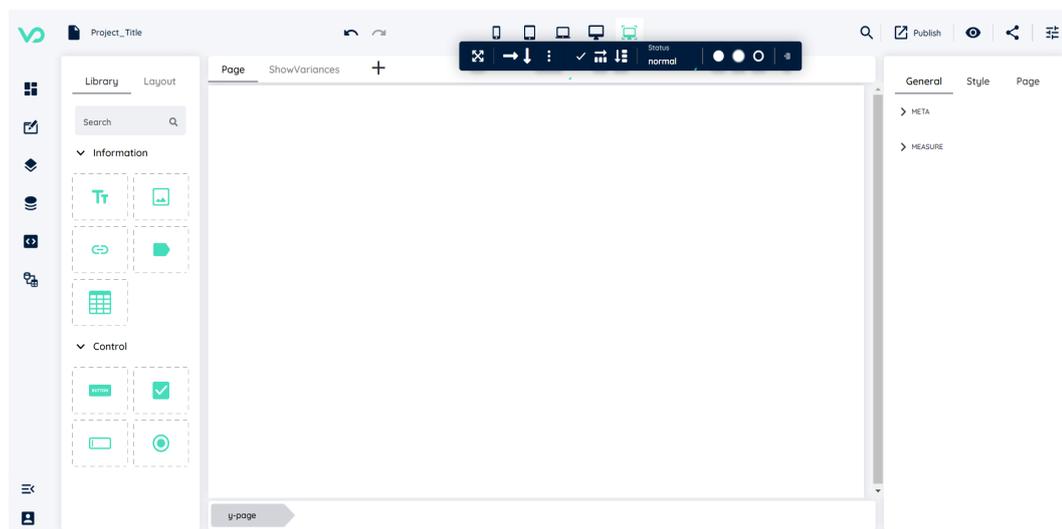
Neben dem bereits etablierten Produkt concept 16, welches bereits seit sehr langer Zeit auf dem Markt besteht, entwickelt die Firma Vectorsoft AG aktuell ein neues Softwarepaket, mit dem sie ihre Produktpalette erweitern möchte. Mit der Low-Code Plattform yeet soll ein salonfähiges Produkt in den Markt der Low-Code Plattformen etabliert werden. Es besteht der Anspruch, mit den Marktführern in diesem Gebiet mithalten und sich an der Spitze festzusetzen. Bis man jedoch von yeet als ernsthafte Konkurrenz für Produkte aus dem Hause SAP und Oracle sprechen kann, muss viel Entwicklungsaufwand in die Entwicklung dieser neuen Software gesteckt werden.

Das Produkt yeet ist eine klassische Low-Code Plattform und soll die Arbeit nicht nur für Bestandskunden, sondern auch für zahlreiche neue Interessenten erleichtern. Mithilfe der Software lassen sich unzählige Geschäftsanwendungen in kurzer Zeit realisieren, was einen großen Vorteil darstellt. Von einem einfachen Kontaktformular über komplexe Kundenverwaltung über Tabellen lassen sich mit yeet unzählige Anwendungsszenarien abbilden (Siehe Abbildung 3.3). Dabei gleicht yeet vom Aufbau her dem klassischen Low-Code Prinzip und bietet dem Anwender neben einem einfachen Designer (Siehe Abbildung 3.4) außerdem eine Datenbank, die dazu dient, die im Designer realisierten Anwendungen mit relevanten Daten zu versorgen. So können mit yeet neben Webauftritten, Formularen und vielem mehr bei-

### 3. GRUNDLAGENKAPITEL



**Abbildung 3.3:** Überblick über eine beispielhaft konzipierte Anwendung im yeet-Designer, Quelle: Eigene Darstellung



**Abbildung 3.4:** Überblick über die Designeranwendung von yeet, Quelle: Eigene Darstellung

spielsweise auch Prozesse wie die Personalverwaltung oder eine Terminverwaltung realisiert werden.

### **3.6.3 Einsatzgebiet der Abschlussarbeit**

Bisher wurde dem Thema Barrierefreiheit noch keine sehr starke Beachtung geschenkt während der Entwicklung und genau dort soll diese Arbeit Hilfestellung bieten. Mithilfe des entwickelten Prototyps soll die Software ein Stück näher Richtung Barrierefreiheit gebracht werden und die Arbeit für potenziell behinderte und eingeschränkte Nutzer erleichtern. Der Prototyp ist jedoch lediglich der Anfang in Richtung vollständiger Barrierefreiheit von yeet und nur ein Teilelement auf diesem Weg. Durch den erarbeiteten Katalog an Kriterien für einen Modus für Farbblindheit/Sehschwäche sowie das gesammelte Wissen zu Barrierefreiheit und Softwareergonomie soll nachhaltig das Verständnis dieser Themen für die Entwickler der Firma gestärkt werden. Dadurch soll die Minimierung bestehender Hürden und Barrieren für Nutzer mit Behinderungen erleichtert werden. Ziel ist es, mit dieser Arbeit also einen Grundstein zu legen und darauf aufbauend das Produkt yeet nachhaltig positiv zu beeinflussen und stetig zu verbessern.



## Kapitel 4

# Analyse von Konkurrenzprodukten

### 4.1 Umfrage bei Vectorsoft AG

Dem Beginn der Erarbeitung dieses wissenschaftlichen Konstrukts wurde eine unternehmensinterne Umfrage vorangestellt. Ziel dieser Umfrage war es, zu erfahren, inwiefern die Beschäftigten im Unternehmen sich mit dem Thema Barrierefreiheit auskennen, ob sie selbst oder Bekannte an einer Farbsehschwäche leiden und wie hoch sie den allgemeinen Nutzen für yeet einstufen.

Dabei ergab sich, dass tatsächlich jeder der Teilnehmer den Begriff der Barrierefreiheit kennt und grundlegende Kenntnisse in diesem Bereich besitzt, obwohl keiner der Befragten an einer Form der Farbsehschwäche leidet. Trotz eines Mangels an Fachwissen zu Barrierefreiheit, stuft lediglich eine Person dies als unwichtig ein, während die anderen großen bis sehr großen Wert darauf legen. Inwiefern diese Ergebnisse anders aussehen würden, wenn sich die Beschäftigten besser damit auskennen würden, bleibt reine Spekulation.

Ganze 45,5% der Befragten haben Menschen in ihrem Umfeld, die an einer Form der Farbsehschwäche leiden, also Menschen, welche sehr wahrscheinlich von barrierefreien Inhalten profitieren würden. In einer abschließenden Frage ergab sich zudem, dass 36,4% den Nutzen für yeet als "hoch" einstufen, 54,5% den Profit eher "neutral" empfinden und lediglich eine Person einen "geringen" Vorteil für yeet darin sieht.

### 4.2 Einleitende Worte

Das folgende Kapitel beschäftigt sich damit, zufällig ausgewählte Konkurrenzprodukte von yeet anhand ihrer Umsetzung der in der nächsten Sektion vorgestellten Richtlinie (Siehe Sektion 4.4) der WCAG zu analysieren. Anhand dieser gilt es, Merkmale und Schwächen festzustellen und mithilfe der Analyse und der Richtlinien, Anforderungen an einen Modus für Farbblindheit/Sehschwäche herauszuarbeiten. Dadurch soll dann ein Prototyp in yeet integriert und schließlich in einem vorbereiteten Usability-Test überprüft und bewertet werden.

### 4.3 Vorgehensweise

Anhand der zuvor erwähnten Richtlinie und ihrer Prinzipien soll untersucht werden, inwiefern Faktoren wie Farbe und Kontrast sowie die Darstellungsweise und Größe von Text bei den ausgewählten Low-Code Plattformen umgesetzt sind. Ziel ist es festzustellen ob die Erfolgskriterien der Richtlinie 1.4 eingehalten oder verfehlt werden. Mithilfe eines Tools zur Analyse von Kontrastverhältnissen wird die Konformität hinsichtlich der Erfüllung eines ausreichenden Kontrastverhältnisses geprüft. Durch die gesammelten Daten kann auch erschlossen werden, ob Ähnlichkeiten bei den Plattformen bestehen, ob es gewisse Standards gibt und ob sich gewisse Techniken für yeet angeeignet werden können. Dafür wurden zu Beginn drei verschiedene Plattformen als Konkurrenzprodukte von yeet identifiziert, die anhand der WCAG Richtlinie untersucht und bewertet werden. Dafür wurde ein Bewertungsbogen angelegt der zur Bewertung der Umsetzung dieser Richtlinie dient. Die Daten werden mit einem Hilfswerkzeug entnommen und mit Hilfe von Excel ausgewertet, um anschließend den Bewertungsbogen ausfüllen zu können. Dabei werden exemplarisch jeweils 15 Datenproben aus den Oberflächen der drei Low-Code Plattformen erhoben, in das Excel Dokument eingepflegt und schließlich analysiert.

### 4.4 Kriterien

Die Web Content Accessibility Guidelines(WCAG) beinhalten diverse Richtlinien und Vorschriften für Entwickler, anhand welcher Webangebote zugänglicher und barrierefrei gestaltet werden können. Das erste Prinzip beschäftigt sich mit der Wahrnehmbarkeit von Inhalten und definiert konkrete Richtlinien dafür, um diese zu gewährleisten. Ein Bestandteil dieser ist die Richtlinie 1.4, welche sich insbesondere mit der Unterscheidbarkeit von Inhalten beschäftigt, also wie gut ein möglicher Nutzer der Anwendung den Inhalt vom Hintergrund differenzieren und wahrnehmen kann [wca18l, vgl.]. Insbesondere relevant für die Analyse sind dabei die Erfolgskriterien der Richtlinie exklusiver zweier, welche sich mit der Wahrnehmbarkeit von Video- und Musikinhalten befassen und insofern bei Low-Code-Plattformen keine Anwendung finden. Für die Analyse werden viel mehr die Kriterien 1.4.1 bis 1.4.13 exklusive der zuvor erwähnten Kriterien 1.4.2 und 1.4.7 geprüft. Zur Analyse einiger Kriterien wurde ein externes Hilfswerkzeug herangezogen, welches im nächsten Abschnitt näher beschrieben wird.

#### 4.4.1 Erfolgskriterium 1.4.1 Verwendung von Farbe

Durch Erfolgskriterium 1.4.1 wird geprüft, ob Informationen der Bedienoberfläche auch ohne Farbe wahrgenommen werden können und diese nicht als einziges Transportmedium gebraucht wird. Insbesondere ist relevant, dass beispielsweise Feedback oder nähere Informationen auch ohne Farbe verstanden werden können [wca18a, vgl.].

#### **4.4.2 Erfolgskriterium 1.4.3 Kontrast(Minimum)**

Bei diesem Erfolgskriterium ist es relevant, welches Kontrastverhältnis zwischen Vorder- und Hintergrund herrscht. Es soll also geprüft werden, ob Bedienelemente die Voraussetzungen erfüllen und großer Text zum einen ein Kontrastverhältnis von mindestens 3:1 besitzt und zum anderen normaler Text ein Minimum an 4.5:1 nicht unterschreitet. Außerdem muss klassischer Text eine Mindestgröße von 24px haben und im fett gedruckten Stil eine Größe von 18.5px besitzen [wca18f, vgl.].

#### **4.4.3 Erfolgskriterium 1.4.4 Textgröße ändern**

Jenes Erfolgskriterium fordert, dass die Bedienoberfläche auf mindestens 200% vergrößert werden kann, ohne dass die Nutzbarkeit und Darstellung der Bedienoberfläche bei größerem Zoomfaktor unhandlich wird [wca18g, vgl.]. Dafür kann im Browser der Zoomfaktor ganz einfach erhöht werden und so kann geprüft werden, ob das Kriterium erfüllt wird.

#### **4.4.4 Erfolgskriterium 1.4.5 Bilder von Text**

Hierbei soll überprüft werden, ob vorhandene Bildelemente und darin enthaltener informationsbehafteter Text signifikanten Einfluss auf die Nutzung haben und anpassbar sind durch den Nutzer. Insgesamt soll außerdem klassischer Text dem Text in Bildern bevorzugt verwendet werden [wca18h, vgl.].

#### **4.4.5 Erfolgskriterium 1.4.6 Kontrast(Erweitert)**

Dieses Erfolgskriterium erweitert die Anforderungen aus Erfolgskriterium 1.4.3 und erhebt striktere Anforderungen an das Kontrastverhältnis von Texten. Hierbei soll großer Text mindestens ein Kontrastverhältnis von 4.5:1 erfüllen und normaler Text eines von 7.5:1. Die geforderten Schriftgrößen für Texte sind hierbei identisch zu Kriterium 1.4.3 [wca18i, vgl.].

#### **4.4.6 Erfolgskriterium 1.4.8 Visuelle Präsentation**

Für Erfolgskriterium 1.4.8 sind diverse kleinere Anforderungen erhoben. Unter anderem ist es gewünscht, dass Nutzer die Vorder- und Hintergrundfarben selbst bestimmen können, also eine Art der Individualisierung angeboten wird. Außerdem sollen Textzeilen eine Breite von 80 Zeichen(inklusive Glyphen) nicht überschreiten und dabei immer links- und rechtsbündig ausgerichtet sein. Der Abstand zwischen Zeilen soll innerhalb von Paragraphen einen Wert von 1.5 nicht unterschreiten, während der Abstand von Paragraphen selbst das 1.5-fache dieses Abstandes haben soll [wca18j, vgl.].

#### **4.4.7 Erfolgskriterium 1.4.9 Bilder von Text(ohne Ausnahme)**

Auch dieses Erfolgskriterium erweitert ein bereits vorausgegangenes Kriterium um weitere Anforderungen. Kriterium 1.4.5 wird erweitert und ergänzend dazu gefordert, dass die Verwendung von Text in Bildern einen notwendigen Grund haben muss. Dabei muss erkennbar

sein, warum er bevorzugt gegenüber klassischem Text verwendet wird und nur so die Informationen passend vermittelt werden können [wca18k, vgl.]. Dabei gilt es zu prüfen, ob zunächst andere Bilder als Firmenlogos verwendet werden und falls diese gefunden werden, die Existenz von informationsbehaftetem Text notwendig ist.

### **4.4.8 Erfolgskriterium 1.4.10 Reflow**

Erfolgskriterium 1.4.10 beschäftigt sich mit der Frage, ob Elemente und insbesondere Inhalt auch ohne vertikales und horizontales Scrollen in einem festgelegten Rahmen vollkommen funktionstüchtig sind und ausreichend wahrgenommen werden können [wca18b, vgl.].

### **4.4.9 Erfolgskriterium 1.4.11 Kontrast ohne Text**

Nach diesem Erfolgskriterium müssen Bedienelemente einer Benutzeroberfläche mindestens ein Kontrastverhältnis von 3:1 erfüllen, um den Anforderungen gerecht zu werden [wca18c, vgl.]. Insbesondere relevant ist allerdings der Text oder das Icon auf dem Bedienelement, welche die Funktionalität beschreiben und daher auch auf Kriterium 1.4.3 und 1.4.5 zu prüfen sind.

### **4.4.10 Erfolgskriterium 1.4.12 Textabstand**

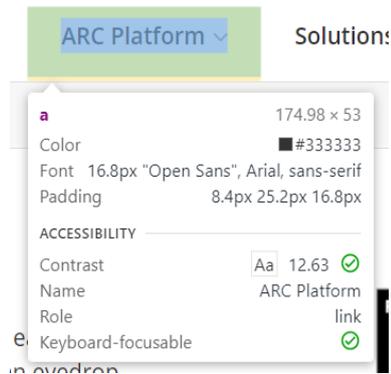
Das vorletzte der betrachteten Erfolgskriterien beschäftigt sich erneut mit Texten. Dabei soll Text auf die in der Richtlinie definierten Werte angepasst werden können, ohne an Funktionalität und Wahrnehmbarkeit zu verlieren [wca18d, vgl.].

### **4.4.11 Erfolgskriterium 1.4.13 Inhalt auf Hover oder Fokus**

Die Anliegen des letzten Erfolgskriteriums beschäftigen sich vollkommen mit extra Inhalten. Dies sind klassische Inhalte, die bei Elementen auftauchen, wenn diese sich im Hover- oder Focus-State befinden. Unter anderem sollen Mechanismen für Nutzer verfügbar sein, um ohne Mausbewegung diese Inhalte wieder zu schließen und gleichzeitig per Maus über den extra Inhalt zu hovern. Außerdem soll der Inhalt nur verschwinden, wenn Gebrauch vom eingangs erwähnten Mechanismus gemacht wird oder das Element den Hover-/Focus-State verlässt [wca18e, vgl.].

## **4.5 Verwendete Hilfswerkzeuge**

Für die Analyse der Kontrastverhältnisse von Vorder- und Hintergrund gibt es viele verschiedene Ansätze. Bereits über die Entwicklertools der gängigen Internetbrowser kann durch Untersuchen eines Elements der Oberfläche festgestellt werden, welches Kontrastverhältnis gegeben ist (Siehe Abbildung 4.1). Dies funktioniert allerdings nicht für jedes Element, das man untersuchen möchte und daher macht es Sinn, weitere Hilfswerkzeuge heranzuziehen, die die Arbeit ein wenig erleichtern und darüber hinaus noch mehr Auskunft geben können. Um den Kontrast von Farben zu untersuchen, gibt es unzählige Werkzeuge auf dem Markt,



**Abbildung 4.1:** Über die Entwicklertools können Elemente untersucht und Informationen über das Element erlangt werden (Beispielhaft hier in Chrome anhand der Webseite <https://www.tpgi.com/color-contrast-checker/> demonstriert).

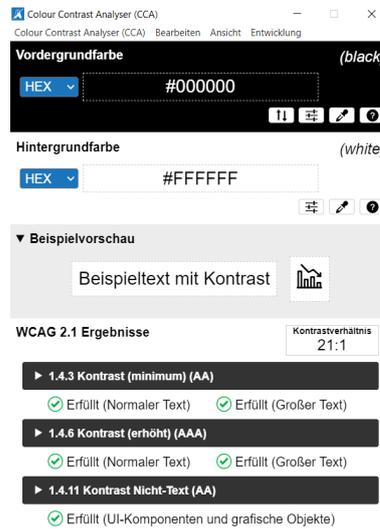
die dafür infrage kommen. Für das gewollte Vorhaben müssen diese Werkzeuge allerdings mehr können, als nur das Kontrastverhältnis wiederzugeben. Wie eingangs erwähnt, ist für die Analyse die Richtlinie 1.4 im Fokus, anhand welcher die Wahrnehmbarkeit von Inhalten bemessen wird. Dazu geben die Autoren der Richtlinien neben Empfehlungen und Querverweisen zu interessanten Artikeln unter anderem auch Empfehlungen für Werkzeuge zur Bestimmung von Kontrastverhältnissen heraus. Eine der Empfehlungen ist dabei das Produkt der Firma TPGi – a Vispero™ Company, welche ein webbasiertes Werkzeug zum kostenlosen Download anbietet, das dazu dienen soll, die Kontrastverhältnisse zwischen zwei verschiedenen Farben zu analysieren (Siehe Abbildung 4.2).

Das Werkzeug erlaubt es, den Inhalt zu analysieren und mithilfe der Daten zugänglicher für Menschen mit einer möglichen Sehbeeinträchtigung zu gestalten. Dabei bietet die Anwendung ganz charakteristische Features an, die sie als geeignet für das Vorhaben dieser Arbeit erscheinen lassen. Neben dem Hinweis durch Indikatoren, ob das errechnete Kontrastverhältnis mit WCAG Anforderungen vereinbar ist, bietet die Anwendung die Möglichkeit für den Nutzer, über eine bestimmte Funktion Farbblindheit zu simulieren und dabei über einen Beispieltext darzustellen, wie das jeweilige Kontrastverhältnis von Menschen mit Farbschwächen wahrgenommen wird (Siehe Abbildung 4.3). Die Bestimmung der Farbwerte kann dabei auf mehrere Arten geschehen und außerdem in den gängigsten Formaten angegeben werden[tpg, vgl.]. Jenes Werkzeug wurde daher gewählt, um die Kontrastverhältnisse der drei ausgewählten Low-Code Plattformen zu analysieren. Die Ergebnisse, welche der Color Contrast Analyzer liefert, kann der Anwender außerdem kopieren und kann diese so anderweitig weiterverwenden. Insbesondere werden mit diesem Werkzeug die Kriterien 1.4.3, 1.4.6 und 1.4.11 der WCAG näher untersucht und vom Tool analysiert.

Der Rest der Kriterien wird über das Untersuchen der Elemente über die Entwickleroptionen des verwendeten Browsers durchgeführt. Dabei können insbesondere Texte und deren Eigenschaften im DOM-Baum betrachtet und anhand der Kriterien genau analysiert werden.

#### 4. ANALYSE VON KONKURRENZPRODUKTEN

---

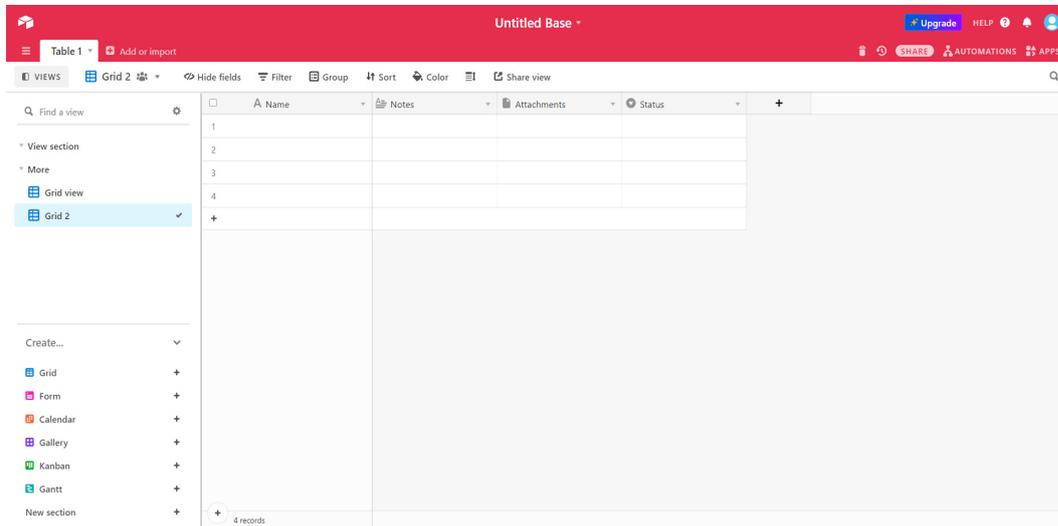


**Abbildung 4.2:** Beispielhaft ist hier die Oberfläche des Color Contrast Checker dargestellt. Quelle: In Anlehnung an [tpg, Abbildung 1]



**Abbildung 4.3:** Beispielhaft wird hier simuliert, wie bestimmte Kontrastverhältnisse unter verschiedenen Farbsehschwächen aussehen. Quelle: Eigene Darstellung

## 4.6 Plattform 1: Airtable



**Abbildung 4.4:** Überblick über die Anwendungsoberfläche von Airtable, Quelle: Eigene Darstellung

### 4.6.1 Produktbeschreibung

Das erste ausgewählte Produkt stammt von der Firma Airtable mit der gleichnamigen Low-Code Plattform (Siehe Abbildung 4.4).

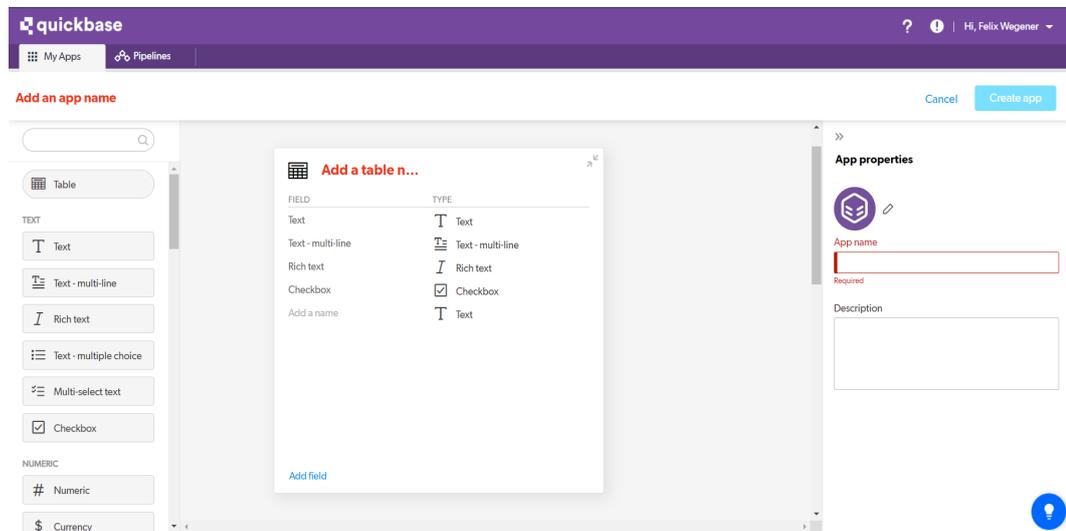
Die klassische Low-Code Plattform versucht, ihren Nutzer mit einfachen Mitteln bei der Erstellung von Tabellen, Anwendungen und vielem mehr zu unterstützen. Über eine minimalistisch gestaltete Oberfläche verfolgt Airtable das bekannte Low-Code Muster und versucht, die Arbeit für seine Anwender zu erleichtern. Dabei können Anwendungen in kurzer Zeit realisiert und konzipiert werden. Diese erstellten Applikationen lassen sich dabei optional einfach mit vom Nutzer favorisierten Apps verbinden und es besteht die Möglichkeit, dass mehrere Nutzer gemeinsam an einem Projekt kollaborieren und die Ergebnisse mit anderen teilen. Anwendungen werden dabei in Echtzeit synchronisiert, was ein problemfreies Arbeiten ermöglichen soll und dafür Sorge trägt, dass mehrere Nutzer gleichzeitig Projekte gestalten können. Das Endgerät spielt keine große Rolle, da Airtable sowohl auf mobilen Endgeräten als auch als Desktopanwendung abrufbar ist [air, vgl.].

## 4.7 Plattform 2: Quickbase

### 4.7.1 Produktbeschreibung

Das Produkt der gleichnamigen Firma Quickbase ist das zweite Konkurrenzprodukt und definiert sich selbst als Low-Code Plattform für Business-Developer (Siehe Abbildung 4.5).

## 4. ANALYSE VON KONKURRENZPRODUKTEN



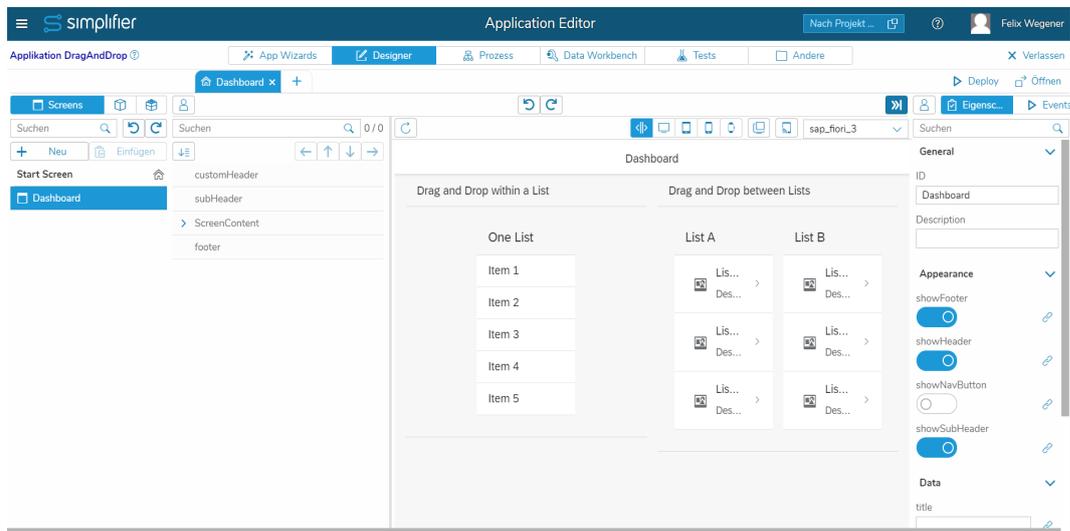
**Abbildung 4.5:** Überblick über die Anwendungsoberfläche von Quickbase, Quelle: Eigene Darstellung

Das Produkt Quickbase ist eine Entwicklungsplattform, die Geschäfts- und IT-Teams vereint, indem sie Lösungsansätze bietet, die jegliche technische Probleme lösen sollen. Dabei können gemeinsam sicher und nachhaltig Lösungsansätze mit Quickbase entwickelt werden, die bestehende Probleme mit dem Low-Code Ansatz lösen können. Wie für Low-Code üblich soll auch Quickbase die Prozesse eines Unternehmens beschleunigen und durch seine Einfachheit in der Bedienung dafür sorgen, dass schneller Lösungen bereit sind. Dabei betitelt sich Quickbase selbst als die einzig vollständige Plattform, die nachweislich die hauseigene Entwicklung auf Unternehmensebene unterstützt. Ziel der Plattform ist es, die Entwicklung von Anwendungen stetig zu verbessern und beschleunigen, indem vom klassisch programmatischen Ansatz Abstand genommen wird. Quickbase Anwendungen ermöglichen es, Daten schnell zu verbinden, die Systeme in Echtzeit zu integrieren und automatisierte Workflows mit einfacher Geschäftslogik zu orchestrieren. Außerdem wird eine einfache Integration von Drittanbieter Applikationen gewährleistet, um diese problemfrei in erstellte Anwendungen zu integrieren. Als weiterer USP erklärt Quickbase, dass Anpassungen an bestehenden und laufenden Applikationen in Echtzeit vorgenommen werden können, ohne dass die Entwickler die Anwendung offline nehmen müssen. Dies soll die Erfahrung noch einmal um ein Vielfaches verbessern und den Kunden des Unternehmens einen großen Komfort bieten [qui, vgl.].

## 4.8 Plattform 3: Simplifier

### 4.8.1 Produktbeschreibung

Das Produkt, das das Trio komplettiert, ist die Low-Code Plattform Simplifier. Das Produkt des deutschen Unternehmens Simplifier AG ist ein unmittelbares Konkurrenzprodukt von yeet und daher sehr relevant für die Analyse (Siehe Abbildung 4.6).



**Abbildung 4.6:** Überblick über die Anwendungsoberfläche von Simplifier, Quelle: Eigene Darstellung

Die Anwendung wirbt mit einer bis zu zehnmals schnelleren App-Erstellung durch User und kann auf jedem beliebigen Endgerät verwendet werden, ist also geräteunabhängig. Dabei können Nutzer Simplifier über beliebig viele Geräte verwenden und gemeinsam mit beliebig vielen Nutzern an einem Projekt arbeiten und interagieren. Außerdem ist es dem Anwender möglich, SAP-, non-SAP- und IoT-Systeme simpel in erstellte Applikationen zu integrieren und dabei mit modernen Technologien aus den Bereichen 2D, 3D, VR und AR zu interagieren [sim, vgl.].

## 4.9 Analyse der Plattformen

Voraussetzung für die Analyse ist die Erstellung eines Benutzerkontos bei jeder Plattform, um auf die kostenlose Version des Produktes zugreifen zu können. Um die Analyse zu beginnen, wurde beispielhaft ein Testprojekt für jede der Plattformen erstellt, anhand dessen die Benutzeroberfläche untersucht wurde. Zunächst galt es, signifikante Bedienelemente festzustellen, die die Bedienung der Benutzeroberfläche und die Funktionalität bestimmen. Denn diese sind die Elemente, über die der Nutzer später seine Anwendung erstellen und entwerfen kann und damit auch die Elemente, die bei einer unzureichenden Beachtung der Barrierefreiheit die Bedienung für potenziell eingeschränkte Nutzer erschweren oder gar verhindern. Erfüllen diese die Anforderungen der Richtlinie, so ist dies ein Schritt mehr in Richtung gelungener Inklusion [HS10, vgl.:86].

Zur korrekten Identifikation der signifikanten Bedienelemente galt es, sich mit den Funktionen der Anwendung auseinanderzusetzen und erste Aktionen auszuführen für ein besseres Bild vom Workflow. Dies dient dazu, einen ersten Eindruck zu jeder Plattform zu erhalten und möglichst gut differenzieren zu können, welche Bedienelemente essenziell für die Bedie-

nung der jeweiligen Low-Code Plattform sind. Zwar ist es im Nachhinein sinnvoll, dass die gesamte Oberfläche den barrierefreien Anforderungen entspricht, allerdings werden manche Elemente weniger frequent verwendet als andere und sind somit in der Rangliste nach Wichtigkeit eher hinten anzuordnen.

Nachdem die Elemente identifiziert wurden, galt es, diese mit Bildmaterial zu dokumentieren, sodass zu den später erhobenen Datenproben auch Beispielbilder vorhanden sind. Das ist insbesondere wichtig, wenn Außenstehende sich ein Bild der analysierten Elemente machen möchten, aber keinen Account bei einer der Low-Code Plattformen besitzen und somit auch keinen Zugriff auf diese Anwendungen haben.

Nach Abschluss der Dokumentation der ausgewählten Bedienelemente wurde mit der Analyse der Kriterien 1.4.3, 1.4.6 und 1.4.11 begonnen und dafür das verwendete Hilfswerkzeug gestartet. Über ein Pipetten-Tool innerhalb der Anwendung kann die Farbe der Bedienelemente festgestellt werden und das Kontrastverhältnis zum Hintergrund wird errechnet. Die Besonderheit des Color Contrast Analyzer ist einerseits die integrierte Analyse, ob das errechnete Kontrastverhältnis den Vorschriften der WCAG entspricht und andererseits auch die Möglichkeit, die erhobenen Daten aus dem Werkzeug zu exportieren und damit in das Worddokument dem entsprechenden Bild anzufügen. Das Hilfswerkzeug prüft dabei die Konformität hinsichtlich der zuvor erwähnten drei Erfolgskriterien und teilt dem Anwender mit, ob diese erfüllt werden. Dies wurde für jedes ausgewählte Bedienelement durchgeführt und entsprechend im angelegten Dokument vermerkt. So findet eine konsistente Erhebung der Daten statt und man kann die Werte dem entsprechenden Bedienelement sehr einfach zuordnen.

Neben diesen Kriterien, wurde auch Erfolgskriterium 1.4.1 mithilfe der Anwendung analysiert. Über die Möglichkeit Farbblindheit zu simulieren, wurde für jedes Element festgestellt, ob einerseits der Vordergrund auf dem Hintergrund wahrgenommen werden kann und im weiteren Vorgehen ohne das Werkzeug bestimmt, ob andere Faktoren außer Farbe Informationen vermitteln, also beispielsweise Texte oder Icons ein gleichwertiges Transportmedium sind.

Daraufhin wurde mit der Analyse der restlichen Kriterien fortgefahren, die nicht über das verwendete Tool analysiert werden können. Durch die Arbeit mit dem Inspektor und den Entwickleroptionen des Browsers können nahezu alle Eigenschaften des jeweiligen HTML-Elementes im DOM-Baum untersucht werden. Dies ist insbesondere hilfreich für die Analyse der Schriftgröße und der Kriterien 1.4.8 und 1.4.12, wobei durch DOM-Manipulation die Elemente an die Forderungen aus Kriterium 1.4.12 angepasst werden können, um die Validität der Erfüllung zu überprüfen.

Anders als die vorausgegangenen Erfolgskriterien wurde für die Analyse der verbleibenden kein weiteres Hilfswerkzeug benötigt. So wurde beispielsweise für Kriterium 1.4.4 der Zoomfaktor im Browserfenster auf 200% erhöht und ein normaler Workflow simuliert, um die Funktionalität zu überprüfen. Dies ist insbesondere relevant für Menschen, die eine sehr geringe allgemeine Sehkraft besitzen oder für Menschen, die auf sehr kleinen Endgeräten arbeiten, da dies oftmals erfordert, dass der Inhalt vergrößert wird, um diesen richtig wahrnehmen zu können. So sollte ein Inhalt um einen gewissen Faktor vergrößert werden können, ohne an Funktionalität, Komfort oder Ästhetik einzubüßen.

Für die Erfolgskriterien 1.4.5 und 1.4.9 wurden die wenigen Bilder der Bedienoberflächen ana-

lysiert, wobei sich diese meist auf die jeweiligen Produktlogos beschränken. Hierbei fordert die WCAG, dass hauptsächlich Textinhalte verwendet werden und möglichst ein Verzicht von Bildern mit relevantem Text stattfindet, da dieser bei schlechter Auflösung möglicherweise nicht mehr wahrgenommen werden kann. Deshalb werden dafür zwei Kriterien definiert, die sich mit der Frage beschäftigen, ob Bildinhalte nur in äußersten Fällen verwendet werden, keine signifikant relevanten Informationen enthalten und überwiegend nur gestalterischer Natur sind. Dies hat den Zweck, wichtige Informationen über klassische Texte zu übermitteln, die auch bei einer höheren Skalierung nicht an Schärfe verlieren, also wenig von der Pixeldichte oder der Größe eines Endgerätes abhängen.

Das letzte übrig gebliebene Erfolgskriterium 1.4.10 galt es, im letzten Schritt für die Elemente zu analysieren. Da diese allerdings keinen großen Inhalt enthalten, wurde die Analyse auf die gesamte Oberfläche ausgelagert, um ähnlich wie bei Kriterium 1.4.4 die Oberfläche als Ganzes bewerten zu können. Dafür wurden, wo möglich, scrollbare Komponenten untersucht, die eine größere Menge an relevantem Inhalt tragen. Auch hierfür konnte ähnlich wie bei den Bildern wenig analysiert werden, da die Bedienoberflächen von Low-Code Plattformen in der Regel wenige große Inhaltsblöcke enthalten und eher auf eine minimalistische Ausprägung der Bedienelemente und deren Inhalt setzen.

Nach der Analyse der Elemente wurde damit fortgefahren, den Teil der Daten für Kontrast und Schrift in ein Verhältnis zu setzen und Durchschnittswerte für die Bewertung zu bestimmen. Zu diesem Zweck wurde zuvor in Excel ein Dokument angelegt, in dem die in Word erhobenen Daten sauber in einen tabellarischen Zusammenhang gesetzt werden können.

In dem für diesen Zweck bestimmten Dokument wurden insbesondere die Werte zu den Kriterien 1.4.3, 1.4.6 und 1.4.11 dokumentiert. Neben dem Kontrastverhältnis und der Schriftgröße der Elemente wurde außerdem das Bestehen und Durchfallen jedes Kriteriums vermerkt. Entscheidend ist, anhand der eingetragenen Datensätze weiterführende Berechnungen durchzuführen, um einen Zusammenhang herzustellen.

Dafür wurde jeder Datensatz anhand der zuvor erwähnten drei Kriterien bewertet und dahingehend festgehalten, ob diese erfüllt oder missachtet werden. Darüber hinaus wurde anhand dieser Zahlen das prozentuale Verhältnis zwischen Bestehen und Durchfallen jedes Kriteriums bestimmt. Denn auch anhand dieser Werte kann festgestellt werden, wie viele Proben den gewünschten Anforderungen entsprechen und ob das Verhältnis eher zugunsten von erfüllten Erfolgskriterien oder eher hin zu unzureichender Wahrnehmbarkeit tendiert. Zwar sind im Idealfall alle Bedienelemente richtlinienkonform, doch zeigen bereits die Zahlen für die erhobene Anzahl an Proben eine klare Tendenz für die restlichen Bedienelemente einer Oberfläche, die nicht analysiert wurden.

Aus den dokumentierten Kontrastverhältnissen wurde außerdem das durchschnittliche Kontrastverhältnis für jede Plattform ermittelt, um eine Rangliste der Plattformen zu erstellen und in Kombination mit dem Verhältnis des Bestehens und Durchfallens von Proben ein Fazit ziehen zu können. Damit kann neben der klassischen Rangliste auch festgestellt werden, ob die Plattformen im Allgemeinen überhaupt genug Wahrnehmbarkeit gewährleisten oder ob möglicherweise alle drei ein unzureichendes Ergebnis erreichen. Denn nur weil eine der drei Plattformen das beste Ergebnis erreicht, heißt es im Umkehrschluss nicht, dass diese die Wahrnehmbarkeit ausreichend gewährleistet und als gebrauchstauglich bezeichnet werden kann.

Neben dem Kontrastverhältnis und Faktoren wie der Farbe und der Schriftgröße spielen bei der Richtlinie 1.4 noch weitere Faktoren eine Rolle, damit diese als erfüllt gilt. Da keine der Low-Code Plattformen auf Audio- oder Videoinhalte setzt, können diese beiden Erfolgskriterien, die sich an solche Inhalte richten, in dieser Analyse außen vor gelassen werden. Um auch alle anderen analysierten Erfolgskriterien der Richtlinie zu beantworten, wurde ein Bewertungsbogen erstellt, der dazu dient, die Ergebnisse aller analysierten Kriterien zusammenhängend zu dokumentieren. Der Bewertungsbogen enthält mindestens eine Frage zu jedem der Erfolgskriterien dieser Richtlinie und deckt dabei einen Rahmen ab zwischen Erfüllung und Durchfallen.

Nach und nach wurden die einzelnen Erfolgskriterien für sich bewertet und so bot sich am Ende der Bewertung ein Gesamtbild, wie viele erfüllt und wie viele verfehlt wurden. Nach einer vollzogenen Bewertung für jede der drei Low-Code Plattformen konnten die Einzelergebnisse in einen Zusammenhang gesetzt werden. Dieser dient einerseits dazu, eine Rangliste zu erstellen, hat aber andererseits auch den Sinn und Zweck zu bewerten, inwiefern sich Low-Code Plattformen mit der Wahrnehmbarkeit ihrer Bedienoberflächen beschäftigen.

Als nützlicher Output dieser Analyse können auch Charakteristiken festgestellt werden, die besonders empfehlenswert sind oder als vermeidbar eingestuft werden, um schließlich den Anforderungskatalog für den Prototyp und eine wahrnehmbare Bedienoberfläche von yeet zu vervollständigen. Damit können die Richtlinien der WCAG ergänzt werden um Features, die andere Low-Code Plattformen ihren Nutzern anbieten, falls diese signifikant positive Einwirkungen auf Barrierefreiheit und Wahrnehmbarkeit der Benutzeroberfläche haben.

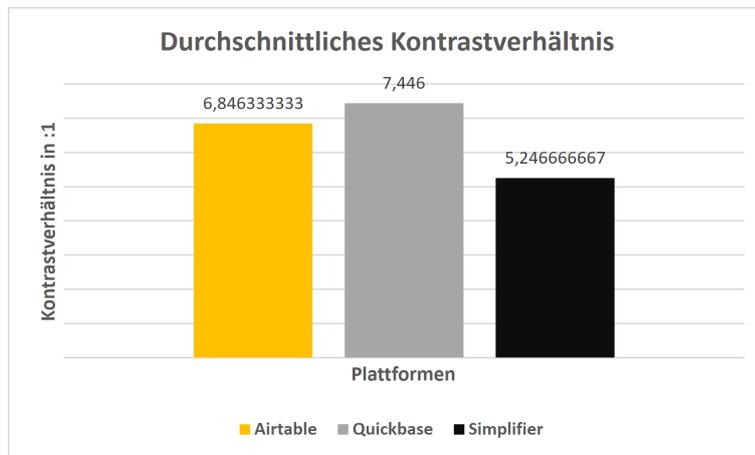
### **4.10 Auswertung der Analyseergebnisse**

#### **4.10.1 Vorwort**

Im folgenden Abschnitt werden nun die Ergebnisse der durchgeführten Analyse dargelegt und differenziert betrachtet. Ziel der Analyse war es, festzustellen, inwiefern die unmittelbare Konkurrenz auf dem Markt das Thema Barrierefreiheit behandelt und insbesondere, ob die Anforderungen der WCAG, bzw. der Richtlinie 1.4 erfüllt oder verfehlt werden. Dafür wurden exemplarisch die Produkte Airtable, Quickbase und Simplifier analysiert und mithilfe des Color Contrast Analyzer 15 signifikant relevante Bedienelemente der jeweiligen Oberfläche untersucht.

#### **4.10.2 Auswertung der Excel-Daten**

Zu Beginn der Auswertung werden die gesammelten Daten aus dem angelegten Excel-Dokument betrachtet und ausgewertet. Dieses enthält die gesammelten Daten aus dem Color Contrast Analyzer und gibt Auskunft über Datensätze wie Kontrast und bestandene/durchgefallene Proben. Vorweg kann angemerkt werden, dass auch der Faktor Schriftgröße eine wichtige Rolle spielt, allerdings jede der drei analysierten Plattformen in dieser Hinsicht die Kriterien der WCAG missachtet und dadurch durchgefallen ist. Selten haben Elemente der Plattformen eine Schriftgröße, die überhaupt dem geforderten Mindestmaß der



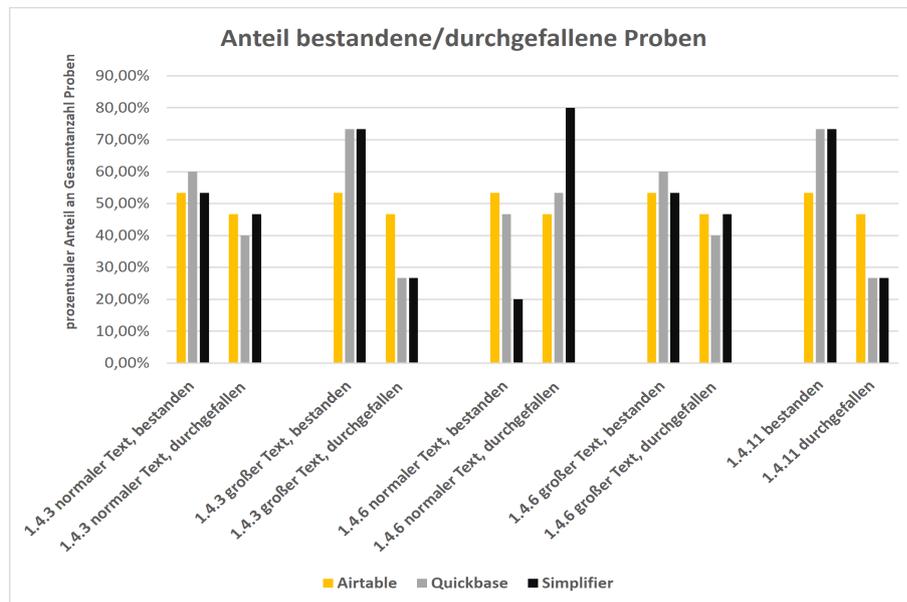
**Abbildung 4.7:** In diesem Balkendiagramm werden die durchschnittlichen Kontrastverhältnisse der untersuchten Plattformen dargestellt von Airtable(links), Quickbase(Mitte) und Simplifier(rechts). Quelle: Eigene Darstellung

WCAG entspricht. Dies ist ein großes Manko hinsichtlich der ausreichenden Wahrnehmbarkeit und Lesbarkeit von Textinhalten.

Wie im bereits im vorherigen Kapitel thematisiert, spielt das Kontrastverhältnis zwischen Vorder- und Hintergrund eines Bedienelements eine entscheidende Rolle für dessen Wahrnehmbarkeit. Zu diesem Zweck wurde für jede Probe das Kontrastverhältnis notiert und in einem nächsten Schritt ein Durchschnittswert aus den erhobenen Daten gebildet. Dies kann in Abbildung 4.7 betrachtet werden.

Die Abbildung zeigt in einem Balkendiagramm den Überblick über das durchschnittliche Kontrastverhältnis jeder Plattform. Ruft man sich in Erinnerung, dass die Richtlinie 1.4 als Erfolgskriterium der Anforderungsstufe AA definiert, dass normale Texte ein Kontrastverhältnis von 4,5:1 und große Texte eins von 3:1 haben, ist dies ein positives Ergebnis für jede der drei Plattformen. Auch das Kriterium 1.4.11, das sich mit dem impliziten Kontrast von Bedienelementen beschäftigt, kann bei jeder Plattform positiv hervorgehoben werden, da die Anforderung eines Kontrastverhältnisses von 3:1 mit dem durchschnittlichen Kontrastverhältnis von jeder Plattform erfüllt wird. Dabei ist es nicht signifikant wichtig, dass beispielsweise ein Button ein gutes Kontrastverhältnis zwischen seiner Hintergrundfarbe und dem Hintergrund, auf dem er eingebunden ist, hat, um als Button erkannt zu werden. Vielmehr ist wichtig, dass der Text und die Kennzeichnung des Buttons ein gutes Kontrastverhältnis besitzen und dies wird bei jeder Plattform gewährleistet, wenn man das durchschnittliche Kontrastverhältnis als Richtwert betrachtet. Jede Low-Code Plattform wird der Anforderungsstufe AA gerecht und erfüllt damit die Erfolgskriterien 1.4.3. und 1.4.11. Schlusslicht bildet dabei die Plattform von Simplifier mit einem Wert von 5,247:1, die im Durchschnitt etwas weiter hinter den beiden anderen Produkten zurückliegt. An Platz 2 der Rangliste ordnet sich das Produkt aus dem Hause Airtable an, das mit einem durchschnittlichen Wert von 6,8643:1 noch einmal ein etwas höheres durchschnittliches Kontrastverhältnis bietet. Den höchsten durchschnittlichen Wert bietet die Plattform Quickbase seinen Anwen-

#### 4. ANALYSE VON KONKURRENZPRODUKTEN



**Abbildung 4.8:** In diesem Balkendiagramm werden die bestandenen/durchgefallenen Proben der untersuchten Plattformen dargestellt von Airtable(links), Quickbase(Mitte) und Simplifier(rechts) für jedes Kriterium. Quelle: Eigene Darstellung

dern und steht damit an Platz 1 der Rangliste. Mit einem Wert von 7,446:1 bietet sie ein durchschnittliches Kontrastverhältnis, welches weit über den Forderungen der Anforderungsstufe AA liegt.

Anders als die Anforderungsstufe AA legt die Stufe AAA schärfere Kriterien in Form von höher geforderten Kontrastverhältnissen fest. Normale Texte müssen hierbei ein Kontrastverhältnis von mindestens 7,5:1 bieten und große Texte eines von 4,5:1. Schaut man sich die analysierten Durchschnittswerte erneut an, so kann festgestellt werden, dass keine der Plattformen mit ihrem durchschnittlichen Kontrastverhältnis den Anforderungen für normale Texte gerecht wird und somit bereits in dieser Anforderungsstufe durchgefallen ist. Für große Texte bieten aber auch hier die Plattformen ein zufriedenstellendes durchschnittliches Kontrastverhältnis, das dem Erfüllen der Anforderungsstufe AAA trotzdem nicht gerecht werden kann, bedingt durch die Anforderung für normalen Text. Die Kriterien für das Kontrastverhältnis von Texten sind insofern sehr relevant, als das die Bedienelemente mit einem Label versehen sind und dessen Wahrnehmung zwingend notwendig ist.

Neben dem Kontrastverhältnis wurde auch die Anzahl durchgefallener und bestandener Proben festgehalten (für die Anforderungen 1.4.3, 1.4.6, 1.4.11). Diese Kennzahlen geben einen noch genaueren Überblick darüber, wie die drei Plattformen die einzelnen Anforderungen erfüllen oder verfehlen (Siehe Abbildung 4.8). Ersichtlich wird, dass bis auf wenige Ausnahmen das Verhältnis zwischen bestandenen und durchgefallenen Anforderungen etwa in Waage ist und tendenziell eher zu Gunsten bestandener Proben ausgeht. Lediglich die Anforderungsstufe AAA für normalen Text zeigt einen etwas stärkeren Bruch in der Darstellung und offenbart das bereits weiter oben erwähnte häufige Durchfallen dieses Erfolgskriteri-

ums. Während die Plattformen Airtable und Quickbase allerdings noch recht ausgeglichen bei dieser Anforderung abschneiden, zeigt insbesondere die Plattform Simplifier dort ihre Schwächen. Gerade die Balken dieser Anforderungsstufe zeigen, dass das Verhältnis zwischen bestandenen und durchgefallenen Proben deutlich ins Negative ausschlägt bzw. deutlich mehr Proben durchfallen bei Simplifier.

### 4.10.3 Auswertung Bewertungsbogen

Zur weiteren Beurteilung der Erfüllung von Richtlinie 1.4 wurde wie bereits erwähnt ein Bewertungsbogen erstellt, um die Ergebnisse zu sammeln. Begonnen wird mit Erfolgskriterium 1.4.1, das die Benutzung von Farbe behandelt. So ist positiv hervorzuheben, dass jede analysierte Plattform diesem gerecht werden kann und die Informationen über diverse Mittel unabhängig der Farbe mitteilt. Dabei reichen textuelle Beschreibungen und Icons vollkommen aus, um alles verstehen zu können. Gutes Beispiel ist dabei ein Ansatz der Plattform Quickbase, bei der das untersuchte Inputfeld neben Farbe auch textuell mitteilt, wenn ein benötigtes Inputfeld nicht ausgefüllt wurde.

Aus der Analyse der Kontrastverhältnisse geht bereits hervor, dass ein ausgewogenes, tendenziell eher positives Verhältnis zwischen erfüllten und durchgefallenen Proben für das Erfolgskriterium 1.4.3 bezüglich des Kontrastminimums herrscht. Dieses wird für normalen sowie großen Text im Durchschnitt öfter gewährleistet, als dass es verfehlt wird. Lediglich der Faktor der Schriftgröße wird von keiner Plattform erfüllt und bei nahezu jeder Probe verfehlt, kann also daher als Schwachstelle identifiziert werden. Gefordert sind Schriftgrößen von 24px für normalen und 18,5px für fett-gedruckten Text, wobei keine einzige entnommene Probe diesen Wert erfüllt.

Das nächste Erfolgskriterium 1.4.4 beschäftigt sich mit der Skalierbarkeit von Texten, wonach diese ohne Informationsverlust auf bis zu 200% vergrößert werden können sollen. In dieser Hinsicht funktioniert jede der drei Plattformen auch bei einer starken Vergrößerung sehr gut und Texte, Informationen und Bedienelemente können auch bei 200% Zoomfaktor gut wahrgenommen werden.

Daraufhin wurde Erfolgskriterium 1.4.5, dessen Erfüllung daran bemessen wird, dass klassischer Text gegenüber Text in Bildern bevorzugt wird, im Bewertungsbogen bewertet. Aus der Analyse geht hervor, dass die Low-Code Plattformen im Allgemeinen sehr wenige statische Elemente wie Bilder verwenden und diese meistens nur als Produktlogo auftreten. Teilkriterium ist dabei auch, dass Text in Bildern anpassbar ist und die Notwendigkeit gegeben sein soll, wenn dieser in Bildern dem normalen, klassischen Text bevorzugt verwendet wird. Da, wie bereits erwähnt, lediglich Produktlogos die Bedienoberfläche zieren, kann der Text dieser zwar nicht angepasst werden, allerdings ist mit diesem Kriterium auch informationsbehalteter Text gemeint, also wird dieses Kriterium faktisch nicht verfehlt und die Verwendung des Textes in Bildern findet damit ausschließlich für sinnvolle, notwendige Zwecke statt und erfüllt im gleichen Zug damit auch Erfolgskriterium 1.4.9.

Anschließend folgt das Erfolgskriterium 1.4.6, welches die Weiterführung des Erfolgskriteriums 1.4.3 ist und sich mit dem Kontrast von Inhalten auf der Anforderungsstufe AAA beschäftigt. Wie bereits weiter oben erwähnt, tendiert lediglich die Plattform Quickbase bei normalem Text zu einem ausgewogen positiven Verhältnis von durchgefallenen und be-

standenen Proben. Die beiden anderen Plattformen werden den Kontrastanforderungen der Stufe AAA für normalen Text nicht bzw. kaum gerecht. Anders hingegen sind die Ergebnisse für großen Text, der nach vorausgegangener Analyse bei jeder Plattform mindestens ein ausgewogenes Verhältnis zwischen Erfüllung und Missachtung hat und bei Quickbase wieder positiv tendiert.

Daraufhin wird Erfolgskriterium 1.4.8 bewertet, das sich der visuellen Präsentation von Inhalten annimmt. Kriterien, die sich mit der Platzierung, mit dem Zeilenabstand und dem Zeichenabstand von Textinhalten beschäftigen, werden sehr gut von den Bedienoberflächen der Plattformen erfüllt und werden den Anforderungen gerecht. Dabei gilt es, trotzdem anzumerken, dass dies selten auf die Probe gestellt wird, da Low-Code Plattformen häufig keine Blocksätze wie klassische Webseiten enthalten, sondern eher kurze prägnante Begrifflichkeiten verwenden. Lediglich über Hilfen und Beschreibungen wird Text etwas ausführlicher, wird aber in diesen Fällen allen Kriterien gerecht. Das einzige Teilkriterium, welches keine positiven Ergebnisse liefert, ist das Teilkriterium, welches sich damit beschäftigt, dass Nutzer Vorder- und Hintergrundfarben der Bedienoberfläche selbst bestimmen können, also eine Art des Themings anbieten. Lediglich Airtable und Simplifier bieten kleinere Anpassungsmöglichkeiten an, die allerdings meist sporadischer Natur sind und nicht erlauben, einzelne Elemente anzupassen. Simplifier bietet beispielsweise die Chance, zwischen einem Light- und einem Dark-Theme zu wechseln und Airtable, dass die Hintergrundfarbe des Projektes verändert werden kann. Dies hat zumindest bei Airtable aber kaum Einfluss auf die Bedienoberfläche selbst und auch bei Simplifier besteht außer dem Themewechsel keine weitere Möglichkeit für die Nutzer, speziellere Anpassungen von Farben vorzunehmen.

Nach diesem Kriterium wurde mit der Bewertung von Erfolgskriterium 1.4.10 weitergemacht. So lässt sich feststellen, dass alle Plattformen diesem Kriterium gerecht werden können und Inhalt auch ohne horizontales und vertikales Scrollen in dem dafür festgelegten Rahmen von 320px zu 256px wahrgenommen werden kann und darüber hinaus seine volle Funktionalität behält.

Auch das Kriterium 1.4.11 zum Kontrast für Bedienelemente erhält eine durchweg positive Bewertung und wird im Durchschnitt betrachtet deutlich öfter gewährleistet, als dass es von den untersuchten Elementen verfehlt wird. Lediglich Airtable hat ein ausgeglichenes Verhältnis aus bestandenen und durchgefallenen Proben, das nur leicht ins Positive tendiert. Damit bildet die Plattform in diesem Fall das Schlusslicht für jenes Kriterium.

Bei der Bewertung des Kriteriums 1.4.12 können alle analysierten Produkte die Kriterien vollständig erfüllen. Die Manipulation über den Inspektor der Entwicklertools erlaubte es, die geforderten Werte anzupassen und offenbarte, dass jedes untersuchte Element dem Erfolgskriterium gerecht wird.

Das letzte untersuchte Kriterium 1.4.13 hingegen erreicht generell eher schlechte Bewertungen bei der Analyse. Lediglich eine Plattform kann zwei von drei der geforderten Kriterien erfüllen und erreicht damit eine tendenziell positive Bewertung. Während alle drei Plattformen über den Hover- und Focus-State verschiedener Elemente persistenten extra Inhalt über ein Pop-Up anbieten, bietet lediglich Simplifier die Möglichkeit an, dass der Nutzer diesen externen Inhalt über die klassische Mausbewegung hinaus mit der ESC-Taste der Tastatur schließen kann. Enttäuschend hingegen ist die Tatsache, dass bei keiner Plattform der Inhalt mit der Maus angesteuert werden kann, ohne dass er verschwindet.

#### 4.10.4 Stellenwert von Barrierefreiheit

Anhand der Analyseergebnisse kann der Stellenwert der Barrierefreiheit in den analysierten Low-Code Plattformen gedeutet werden. Zwar ist die Analyse nicht repräsentativ für den gesamten Markt, doch vermittelt sie eine Tendenz, wie Low-Code Plattformen das Thema Barrierefreiheit behandeln und wie diese sich mit dem Thema auseinandersetzen.

Die Ergebnisse zeigen, dass die Oberflächen durchaus vielen Kriterien der WCAG gerecht werden können und diese zum Großteil gewährleisten. Zwar offenbaren sich einige Schwächen, gerade im Bereich der Schriftgröße zum Beispiel oder der generellen Anpassbarkeit an individuelle Anforderungen. Allerdings überwiegen die positiven Ergebnisse der Analyse im Vergleich zu den verfehlten Kriterien.

Dank der Analyse konnten einige Charakteristiken festgestellt werden, die für die Implementation in yeet durchaus berücksichtigt werden müssen. Faktoren wie das Kennzeichnen von unterschiedlichen Status über ergänzende Texte oder den Stellenwert eines guten Kontrastes haben bleibenden Eindruck hinterlassen und sind insbesondere Punkte, die nach der Analyse positive und negative Eindrücke hinterlassen haben. Das Vorgehen konnte offenbaren, wie unumgänglich es ist, dass die Bedienelemente optimal wahrgenommen werden können und inwiefern dies die Differenzierbarkeit der Anwendung verbessert. Durch das verwendete Hilfswerkzeug konnte simuliert werden, inwiefern beispielsweise schlechter Kontrast bei einer Sehschwäche die Arbeit erschweren kann. Durch die Simulation der Farbblindheit konnte sich selbst ein Bild der Lage gemacht werden, was nachhaltig im Gedächtnis geblieben ist. Zum jetzigen Zeitpunkt machen die Plattformen vieles richtig, sollten trotzdem aber noch akribisch an den verfehlten Kriterien arbeiten und ihre Oberfläche damit besser zugänglich machen. Nach der durchgeführten Analyse bildet sich jetzt ein besseres Gesamtbild für das Entwicklerteam von yeet, da nun beurteilt werden kann, welchen Stellenwert Barrierefreiheit bei Konkurrenzprodukten hat und insbesondere, was es umzusetzen gilt und ob gewisse Standards vertreten sind. Nebenbei können die Ergebnisse dazu verwendet werden, auch yeet anhand dieser Kriterien stetig zu analysieren, zu verbessern und die Barrierefreiheit der Plattform zu einem USP zu machen.



# Kapitel 5

## Anforderungskatalog

### 5.1 Anforderungen des Prototyps und der Bedienoberfläche

Voraussetzung für die Implementierung des Prototyps sind die Ergebnisse aus der Analyse der Konkurrenzprodukte, welche den Anforderungskatalog komplettieren sollen. Grundlage bilden aber zunächst die Anforderungen aus den Richtlinien der WCAG, insbesondere aus der Richtlinie 1.4, die die Differenzierbarkeit von Inhalten thematisiert, also genau den Zweck des Prototyps und einer gelungenen Bedienoberfläche. Dieser soll die Differenzierbarkeit und Wahrnehmbarkeit der Benutzeroberfläche nachhaltig verbessern und so für Nutzer mit einer Farbblindheit/Sehschwäche optimieren. Inspiration der Anforderungen ist des Weiteren gegeben durch die verwandte Arbeit [FVK<sup>+</sup>16], die durch ihre Analyse weitere Anforderungspunkte für Webangebote feststellen konnte. Für den Prototyp und die Bedienoberfläche von Low-Code Plattformen lassen sich die Anforderungen in die nachfolgenden Richtlinien untergliedern:

- Farbe
- Schrift
- Kontrast
- Skalierbarkeit
- Theming

#### 5.1.1 Farbe

Als nahezu wichtigstes Merkmal bei einem Modus für Farbblindheit/Sehschwäche kann vor allem der Faktor Farbe betitelt werden. Erfolgskriterium für eine barrierefreie Nutzung ist, dass Informationen nicht ausschließlich über Farben vermittelt werden. Dies meint, dass Alternativen der Informationsvermittlung existieren, also Farbe nicht als ausschließliches Transportmedium dient. Möglichkeiten wie die Verwendung von Icons oder die textuelle Darstellung von Informationen sind sinnvoll und sollten in Erwägung gezogen werden. Außerdem sollte auf bestimmte Farben prinzipiell verzichtet werden beziehungsweise innerhalb

der Bedienoberfläche vermieden werden, besonders Farben wie Grün, Rot oder Blau zu verwenden, da dafür bekannte Farbsehschwächen existieren [wca18a, vgl.].

Als positives Beispiel kann dabei Quickbase gesehen werden, wo beispielsweise notwendige Eingabefelder mit einem beschreibenden Text gekennzeichnet werden. Zwar sind auch diese mit roten Farbakzenten markiert, doch bietet die textuelle Hilfe die gleichen Informationen und macht die Signalfarbe dementsprechend obsolet. Um einen erfolgreichen Prototyp zu gestalten, gilt es, Nutzen, Ästhetik und Zweck der Farbgebung abzuwägen, damit ein sinnvolles Gesamtkonstrukt konzipiert werden kann. Gerade die Texteingabe über Formularfelder kann zur Herausforderung für Betroffene werden, wenn unter anderem eine misslungene Eingabe ausschließlich über rote Farbakzente signalisiert wird. So können Missverständnisse bei der Bedienung von Low-Code-Plattformen entstehen, die Nutzer negativ werten und vielleicht abschrecken. Es gilt, verschiedene Status der Bedienelemente nicht abhängig von Farbe zu machen und Alternativen in Erwägung zu ziehen. Geeignet sind Kennzeichnung, die klar und verständlich die benötigten Informationen vermitteln und in ihrer Einfachheit für sich sprechen. Ein Ausrufezeichen hat etwa den gleichen Wirkungscharakter wie die Signalfarbe Rot und kann daher genauso stark vermitteln, dass bestimmte Eingaben nötig sind.

Werden diese Anforderungen eingehalten, so stellt man eine ausreichende Informationsvermittlung bereit. Interessant sind dabei auch die Ergebnisse aus der Arbeit [FVK<sup>+</sup>16], da die Ergebnisse der dort durchgeführten Workshops und die Expertisen zeigen, dass sich einige sehbehinderte Nutzer wohler fühlen, wenn auf höhere Kontraste mit Farben anstelle von klassischen Schwarz und Weiß gesetzt wird [FVK<sup>+</sup>16, vgl.:85].

### 5.1.2 Schrift

Neben der Farbe von Inhalten ist auch die Schriftgröße ein entscheidender Faktor, der Inhalte und vor allem Texte in ihrer Lesbarkeit signifikant beeinflusst. Dies bringt nicht exklusiv Vorteile für Menschen mit einer Sehbehinderung, sondern macht Inhalte klarer und erleichtert die Identifikation der Funktionalität von Elementen. Da Bedienelemente essenziell für die Nutzung von Low-Code Plattformen sind, gilt es diese klar und lesbar zu gestalten, sodass Nutzer die Anwendung wie gewünscht verwenden können. Für Elemente sollte daher eine Textgröße von 18.5px nicht unterschritten werden, da dies die Qualität der Differenzierbarkeit verschlechtern kann [wca18f, vgl.][wca18i, vgl.].

Die Bedienelemente der untersuchten Plattformen habe diesen Richtwert bei nahezu jedem Element verfehlt und laufen daher Gefahr, dass Menschen mit einer Sehbeeinträchtigung die Elemente nicht mehr richtig wahrnehmen können. Können Buttons oder informationsbehaltete Texte nicht gelesen werden, verunsichert dies den Nutzer in seiner Handlungsweise und kann ebenfalls abschreckend wirken.

Charakteristikum der klassischen Bedienoberfläche von Low-Code Plattformen ist, dass wenig Text verwendet wird und die Elemente meist für sich sprechen. Doch gerade innerhalb von Tooltips oder Beschreibungen können längere Texte durchaus auftreten. Ist dies der Fall, so muss gewährleistet werden, dass der Abstand der Buchstaben, der Wörter, der Zeilen und der Absätze groß genug ist und die Texte klar strukturiert und lesbar sind. Die Übersicht und eine gute Lesbarkeit fördern die Wahrnehmbarkeit und erhöhen die Differenzierbarkeit der Inhalte. Texte sollten so vorbereitet sein, dass, wenn sie vergrößert werden, trotzdem

noch lesbar sind und die Formatierung nicht darunter leidet und sich verschlechtert. Gerade wenn Nutzer beispielsweise ergänzende assistierende Technologien verwenden, sollten die Texte dieses Kriterium erfüllen. All dies garantiert, dass Menschen mit Sehschwächen dem Text besser folgen können und dieser im Allgemeinen klarer zu erkennen ist [wca18j, vgl.].

### 5.1.3 Kontrast

Neben den beiden vorausgegangenen Faktoren ist auch der Kontrast von signifikanter Bedeutung für differenzierbare Inhalte auf Bedienoberflächen. So kann weißer Text, der allen Kriterien für lesbare Schrift entspricht, kaum auf hellgrauem Hintergrund wahrgenommen werden, einfach weil das Kontrastverhältnis so gering ist, dass der Vordergrund kaum vom Hintergrund differenziert werden kann. Ein schlechter Kontrast macht Bedienelemente nahezu unnutzbar und damit auch die Low-Code Plattform für effiziente Arbeit ungeeignet. Vor allem in Bedienoberflächen, die aus unzähligen Bedienelementen bestehen, ist es unerlässlich, dass diese entsprechend beschriftet und gekennzeichnet sind hinsichtlich ihrer Funktion. Herrscht dann ein unzureichendes Kontrastverhältnis zwischen Vorder- und Hintergrund, so kann die Beschriftung nicht mehr wahrgenommen werden und der Nutzer kann nicht mehr verstehen, welche Funktion das Bedienelement besitzt [wca18l, vgl.].

Dies impliziert, dass ein gutes Kontrastverhältnis unerlässlich für die Oberfläche und ihre Bedienelemente ist. Da dieser Anforderungskatalog eine möglichst hohe Differenzierbarkeit ermöglichen soll, kann sich hierbei auf die Anforderungsstufe AAA der WCAG bezogen werden. Diese fordert mindestens ein Kontrastverhältnis von 7.5:1 für normalen Text und Text innerhalb von Bildern, damit dieser optimal wahrnehmbar ist. Texte mit großer Schriftgröße hingegen benötigen lediglich ein Kontrastverhältnis von mindestens 4.5:1, um die Anforderungen zu erfüllen [wca18i, vgl.].

Da es unerlässlich ist, dass die Beschreibungen der Bedienelemente und damit deren Zweck erkannt werden können, wird ein so hohes Kontrastverhältnis in diesem Katalog gefordert. Gerade die Bedienoberflächen der analysierten Low-Code-Plattformen haben in diesem Segment Schwächen gezeigt und so bieten nicht alle Bedienelemente ein gutes Kontrastverhältnis und erschweren damit die Arbeit für Betroffene. Erkenntnis der durchgeführten Analyse ist unter anderem, dass gerade Hell-/Dunkel-Kontraste das Kontrastverhältnis enorm verbessern. Die Verwendung eines Komplementärkontrastes zwischen Rot und Grün hingegen ist ein absolut vermeidbares Szenario, dass keinesfalls Einzug in eine Bedienoberfläche finden sollte.

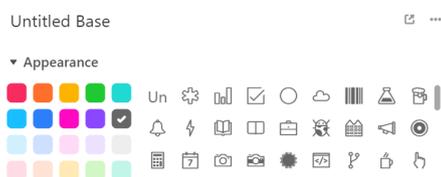
### 5.1.4 Skalierbarkeit

Eine weitere zu beachtende Anforderung ist die Skalierbarkeit von Inhalten. Damit ist gemeint, dass die Oberfläche auf bis zu 200% der eigentlichen Darstellungsgröße vergrößert werden kann. Dies trägt zum einen dazu bei, dass Menschen mit einer schwächeren Sehkraft die Inhalte vergrößern können, um diese besser wahrnehmen und differenzieren zu können. Andererseits macht diese Anforderung es möglich, dass Nutzer mit Endgeräten, die beispielsweise eine geringere Pixeldichte haben oder kleinere Bildschirme besitzen, die Inhalte vergrößern können. Dies sichert vor allem ergonomische Vorteile, da Inhalte somit nicht

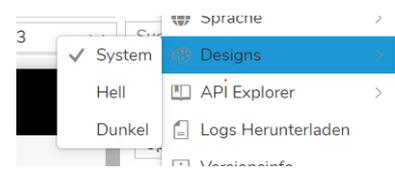
nur für jeden Nutzer wahrnehmbar sind, sondern auch geräteunabhängig abgerufen werden können. Dabei ist wichtig, dass Inhalte auch bei einem größeren Zoomfaktor noch ihre volle Funktionalität behalten und lesbar sind [wca18g, vgl.]. Die Ergebnisse der Analyse der gesammelten Daten und der Expertisen aus [FVK<sup>+</sup>16] bestätigen außerdem die Annahme, dass es wünschenswert für Nutzer ist, wenn Bedienoberflächen die Möglichkeit bieten, die Größe von Texten anzupassen und dabei nicht die Qualität der Navigation leidet [FVK<sup>+</sup>16, vgl.:85]. Für Low-Code Plattformen heißt dies insbesondere, dass signifikante Bedienelemente diesen Anforderungen nachkommen und dabei gut lesbar und übersichtlich sind. Idealfall ist dabei die Verwendung vom Flexbox-System, da dieses von Haus aus eine automatische Skalierung bietet und sich somit die Elemente unmittelbar an den Viewport anpassen. Somit ist Flexbox eine responsive Möglichkeit, die der Anforderung der Skalierbarkeit von Inhalten sehr genau nachkommt.

### 5.1.5 Theming

Als letzte Anforderung wird das Theming definiert. Unter Theming wird in diesem Anforderungskatalog verstanden, dass Nutzern die Möglichkeit gegeben wird, die Benutzeroberfläche an die eigenen Anforderungen anpassen zu können. Damit ist gemeint, dass beispielsweise Farben oder Schriftgrößen einstellbar sind, also der Nutzer die Chance hat, die Oberfläche zu beeinflussen. Dies gewährleistet, dass Nutzer ihre Bedienoberfläche an ihre jeweiligen, doch meist sehr speziellen Bedürfnisse, anpassen können und somit ein gewisser Gestaltungsfreiraum besteht, der die eigene Arbeit effizienter machen kann. Manche Nutzer können bestimmte Farben besser als andere differenzieren und manche Texte selbst mit 24px als Schriftgröße schlecht wahrnehmen. Durch die Möglichkeit der Editierbarkeit wird dies umgangen und der Nutzer hat seine Nutzungserfahrung ein Stück weit selbst in der Hand [wca18j, vgl.]. Inspiration liefern Plattformen wie Airtable und Simplifier, wo der Nutzer beispielsweise zwischen verschiedenen Themes wechseln kann oder die Möglichkeit hat, die Grundfarbe des Projektes aus einer vordefinierten Palette einzustellen. Dieser Gedankengang wird mit dieser Anforderung aufgegriffen und soll weitergeführt werden. So sind die Möglichkeiten bei den analysierten Plattformen doch sehr limitiert und dies soll im zu entwickelnden Prototyp umfangreicher angeboten werden.



(a) Theming bei Airtable



(b) Theming bei Simplifier

**Abbildung 5.1:** Überblick über das Theming der Plattformen, Quelle: Eigene Darstellung

## 5.1. Anforderungen des Prototyps und der Bedienoberfläche

Checkliste zur Umsetzung des Anforderungskataloges		
	Do's - für eine gelungene Umsetzung	Dont's - für eine misslungene Umsetzung
Farbe	Icons oder Texte als Alternative zu reiner Farbe	Verwendung von Rot-/Grün-/Blau-Tönen
	Tooltips bei Hover-Animation einbauen -> mit prägnanten Informationen	Informationen nur über Farben vermitteln
Schrift	normaler Text mindestens 24px groß, fett gedruckter Text mindestens 18.5px groß	enge und unübersichtliche Textblöcke
	ausreichender Zeilen- und Wort- und Buchstabenabstand	zu lange Textblöcke
	kurze und prägnante Informationstexte -> links- oder rechtsbündig ausgerichtet	
Kontrast	bei großen Texten mindestens ein Kontrastverhältnis von 4.5:1 oder höher	Hinter- und Vordergrund mit gleicher Farbe, aber anderer Sättigung
	bei normalen Texten ein Kontrastverhältnis von 7.5:1 oder höher	den Komplementärkontrast aus Rot und Grün verwenden
	bei Bedienelement und dessen Beschriftung ein Kontrastverhältnis von 7.5:1 oder höher	
Skalierbarkeit	Arbeit mit dem Flexbox-System -> automatisches Skalieren und Anpassen	Elemente static oder sticky machen
	Breakpoints einrichten	feste Pixelwerte bei Elementen unabhängig der Schriftgröße
	vergeben von Größenwerten in Einheiten wie bspw. %	
Theming	Gestaltungsfreiraum bieten -> verändern von CSS-Werten ist möglich	feste und unveränderbare Themes

**Abbildung 5.2:** Die Abbildung zeigt die Checkliste zur Umsetzung des Anforderungskataloges. Quelle: Eigene Darstellung

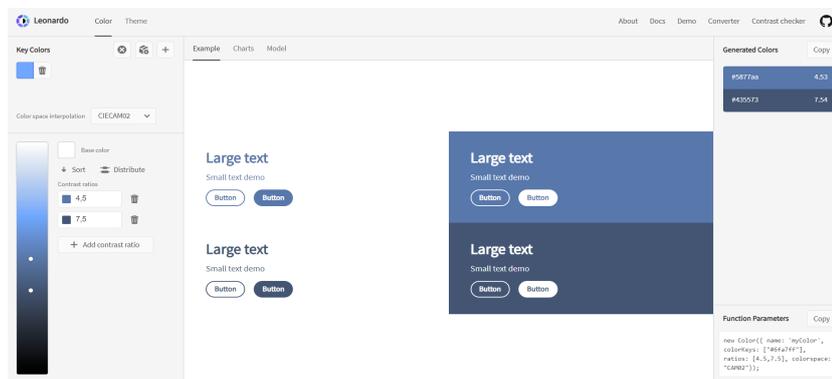
### 5.1.6 Checkliste

Zur Überprüfung der Umsetzung der Anforderungen wurde eine Checkliste erstellt, die dazu dient, sich an guten und schlechten Praktiken zu orientieren und das Ganze als Hilfestellung zu verwenden (Siehe Abbildung 5.2).



# Kapitel 6

## Implementierung eines Prototyps



**Abbildung 6.1:** Die Abbildung zeigt die Bedienoberfläche der Open-Source Anwendung Leonardo aus dem Hause Adobe. Quelle: Eigene Darstellung

### 6.1 Verwendete Hilfswerkzeuge

Für die Erstellung des Prototyps wurde ein Hilfswerkzeug verwendet, das dazu dient, anhand der bereits vorhandenen Farben von yeet, Kontrastverhältnisse zu bestimmen und Anpassungen dort vorzunehmen, wo die vorhandenen Farben den Anforderungen der WCAG nicht gerecht werden. Dafür wurde das Open-Source Projekt Leonardo der Firma Adobe verwendet, welches im Folgenden näher betrachtet wird.

Bei Leonardo handelt es sich um ein Tool, mit dem Nutzer adaptive Farbpaletten generieren und anhand gewünschter Kontrastverhältnisse Farben generieren lassen können (Siehe Abbildung 6.1) [Nat19, vgl.]. Dabei sortiert das Tool die Farben anhand von Helligkeit und dem Benutzer stehen die Möglichkeiten offen, aus einer Vielzahl an Farbräumen zu wählen. Die Besonderheit, die dieses Tool, so behauptet es von sich selbst von anderen unterscheidet, ist, dass der Nutzer die Kontrastverhältnisse nach eigenem Ermessen bestimmen kann, was insbesondere dazu dient, besonderen Fokus auf Barrierefreiheit zu legen. Damit ist dieses

Tool sehr geeignet für das Vorhaben des Prototyps, da anhand der Kontrastverhältnisse der WCAG, die entsprechenden Farben generiert werden können und dies den Aufwand enorm erleichtert und den Workflow und die Effizienz verbessert [Nat19, vgl.].

Außerdem erlaubt Leonardo, dass mehrere Entwickler an einem Projekt arbeiten, da die URL für jedes Projekt zum einen eindeutig ist und zum anderen mit Mitarbeitenden geteilt werden kann. So können Arbeiten innerhalb der Anwendung simpel exportiert und so wiederverwendet werden, um sie beispielsweise auch an weitere Beteiligte zu übermitteln, die nicht direkt an dem Leonardo-Projekt mitarbeiten. Besonders relevant für den Prototyp ist insbesondere die Möglichkeit, bereits vorhandene Farbpaletten in die Anwendung zu migrieren und Anpassungen daran vorzunehmen, was exakt dem Vorhaben des Prototyps entsprechen soll, da im Grunde das Material Design bestehen bleiben soll, allerdings in einer leicht optimierten Version hinsichtlich der Erfüllung der Anforderungen der WCAG [Nat19, vgl.].

### 6.2 Implementierung

Die Low-Code Plattform yeet besitzt aktuell drei verschiedene Themes, welche der Bedienoberfläche über Designtokens bestimmte Werte zuweisen und damit beispielsweise Farben und Schriftgrößen festlegen. Ziel des Themings ist es dabei, langfristig jede Komponente der Oberfläche über das Design-Theme zu steuern.

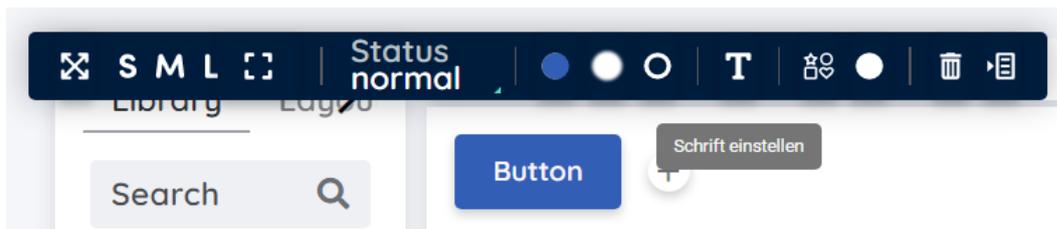
Die Optik von yeet wird durch die Designer an die Entwickler mitgeteilt, die schließlich die Werte für die Tokens in die einzelnen Themes eintragen. Das Standarddesign ist das aktuelle Light-Theme der Anwendung und darin sind die Farben enthalten, die als Corporate Design für das Produkt yeet verstanden werden können. Während im Light-Theme die Farbe Weiß als Primärfarbe zugrunde liegt, wird im Dark-Theme Schwarz als grundlegende Primärfarbe verstanden.

Für den Prototyp wurde sich als Basis dem aktuellen Light-Theme angenommen, um dieses WCAG konform zu optimieren und darüber hinaus an die Anforderungen aus dem erarbeiteten Anforderungskatalog anzupassen. Zu Beginn der Arbeit am Prototyp galt es, zu identifizieren, welche Bedienelemente auf der Benutzeroberfläche nicht konform gegenüber den definierten Anforderungen sind. Ähnlich wie bei der Analyse der Konkurrenzprodukte wurden dabei die Bedienelemente und die CSS-Eigenschaften untersucht. Für den Optimierungsprozess sind hierbei allerdings nur Elemente relevant, die die Anforderungen verfehlen, weshalb auch nur diese dokumentiert und ihre Schwächen festgehalten wurden. Mit Hilfe von Leonardo, dem ausgewählten Tool, wurde nach fertiger Dokumentation der durchgefallenen Bedienelemente mit der eigentlichen Arbeit am Prototyp begonnen.

#### 6.2.1 Farbe

Da auf Basis des aktuellen Light-Themes gearbeitet wurde, bestand der Anspruch, das Corporate Design von yeet beizubehalten und lediglich Nuancen anzupassen. Dafür wurden die Elemente identifiziert, bei denen die primäre Informationsvermittlung lediglich über Farben geschieht. Da yeet bereits darauf setzt, möglichst viel über Icons oder beschreibende Texte zu vermitteln, musste in dieser Hinsicht keine Veränderung am Basis-Theme vorgenommen werden. So sind außerdem die Bedienelemente und insbesondere die vielen Icons mit Tooltips

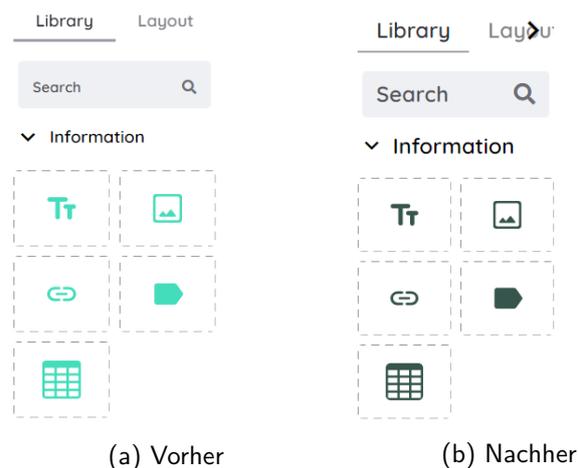
ausgestattet, die dem Benutzer Informationen über die Funktionalität des Bedienelementes verraten (Siehe Abbildung 6.2). Das Design setzt dabei auf die Verwendung von einem Hell-/Dunkel-Kontrast und vermeidet Farben, die problematisch für farbenblinde Nutzer sind. Mit fortschreitender Weiterentwicklung der Bedienoberfläche gilt es, dies beizubehalten und im Ernstfall eine Anpassung für betroffene Elemente vorzunehmen, falls diese das Kriterium verletzen.



**Abbildung 6.2:** Überblick über die Anforderung Farbe, Quelle: Eigene Darstellung

### 6.2.2 Schrift

Neben der Farbe und dem Kontrast wurden die Bedienelemente von yeet auch auf ihre Schriftgröße untersucht, da diese ebenfalls wesentlich für eine gute Wahrnehmung und Differenzierbarkeit ist. Für die Elemente, bei denen die Schriftgröße nonkonform bezüglich der Anforderungen ist, wurde auch diese an die Anforderungen (Siehe Kapitel 5.1.2) angepasst, beispielhaft dargestellt in Abbildung 6.3.



**Abbildung 6.3:** Überblick über die Veränderung der Schrift, Quelle: Eigene Darstellung

### 6.2.3 Kontrast

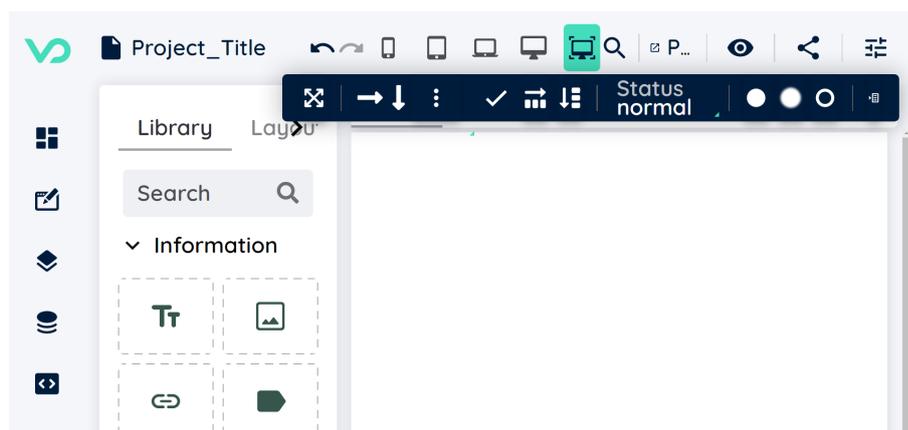
Für jedes Bedienelement, dessen Kontrastverhältnis die Anforderungen verfehlt (Siehe Kapitel 5.1.1 und Kapitel 5.1.3), wurden Vorder- und Hintergrundfarbe in das Tool Leonardo händisch eingetragen und das gewünschte Kontrastverhältnis über einen Regler mitgeteilt. So konnte Leonardo eine neue Primärfarbe auf Basis der zuvor mitgeteilten Vordergrundfarbe bestimmen. Die neu berechnete Primärfarbe steht nun im gewünschten Kontrastverhältnis zur eingespeisten Hintergrundfarbe und wurde unmittelbar als Hex-Wert aus Leonardo kopiert und dokumentiert. Dieser Prozess wurde für jedes Element repetitiv durchgeführt und kann exemplarisch in Abbildung 6.4 betrachtet werden.



**Abbildung 6.4:** Überblick über die Veränderung des Kontraste, Quelle: Eigene Darstellung

### 6.2.4 Skalierbarkeit

Die Anforderung der Skalierbarkeit (Siehe Kapitel 5.1.4) musste nun im nächsten Schritt auch für yeet überprüft werden. Dafür wurde der Zoomfaktor auf bis zu 200% erhöht und getestet, inwiefern die Bedienbarkeit und Ästhetik der Oberfläche noch gegeben ist. Ähnlich wie beim Kriterium der Farbe konnte zum aktuellen Entwicklungsstand keine Schwäche festgestellt werden, sodass auch hier keine weitere Anpassung vorgenommen werden musste. Da yeet bereits mit dem Flexbox-System arbeitet, skalieren sich die Elemente aktuell automatisch und sind vollkommen responsiv (Siehe Abbildung 6.5).

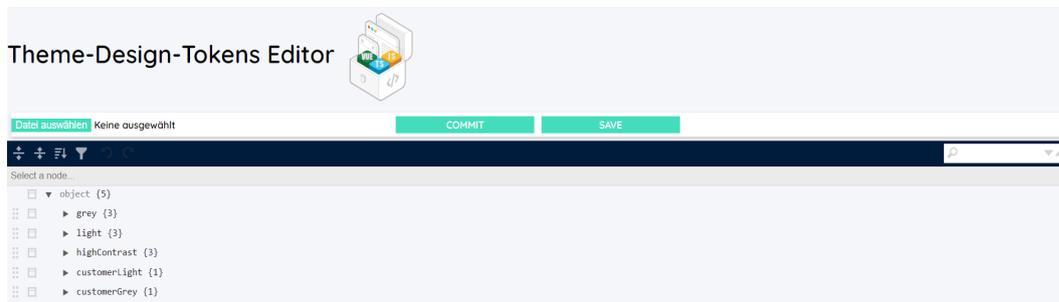


**Abbildung 6.5:** Überblick über die Oberfläche bei einem Zoomfaktor von 200%, Quelle: Eigene Darstellung

### 6.2.5 Theming

Die letzte Anforderung ist das sogenannte Theming (Siehe Kapitel 5.1.5), also die Möglichkeit für Nutzer, Parameter wie Farbe zu beeinflussen und ggf. zu verändern. Tatsächlich bietet yeet eine solche Möglichkeit bereits für seine Nutzer an in Form eines Theme-Editors. Dabei können Nutzer die Werte des aktuellen Themes in einer eigens dafür gestalteten Bedienoberfläche verändern und speichern (Siehe Abbildung 6.6).

Somit mussten dafür keine Vorkehrungen im Theme selbst getroffen werden, da diese Veränderungen von außerhalb geschehen und durch die Verwendung von Designtokens die Werte dieser von den Benutzern einfach verändert werden können und das jeweilige Theme unmittelbar darauf reagiert.



**Abbildung 6.6:** Überblick über den Theme-Editor, Quelle: Eigene Darstellung

### 6.2.6 Erstellen eines neuen Themes

Nachdem die Anforderungen überprüft wurden, wurde eine neue Theme-Basis kreiert, die sich den Werten aus dem Light-Theme bedient. Daraufhin wurden die neu definierten Werte an entsprechender Stelle eingetragen und übernommen, sodass das neue Theme den definierten Anforderungen entspricht. Wechselt der Nutzer nun zum Prototyp, so bekommt die Bedienoberfläche die neu definierten Werte in die Designtokens eingespeist und die Bedienoberfläche passt sich an die veränderten Werte an. Bestehen über den neuen Prototyp hinaus Schwierigkeiten in der Differenzierbarkeit der Oberflächenelemente kann der jeweilige Nutzer das Theme kontinuierlich an seine ganz eigenen Bedürfnisse anpassen. Wie bereits [FVK<sup>+</sup>16] erwähnen, gibt es unzählige individuelle Anforderungen, weshalb es nahezu unmöglich ist, eine allgemeingültige Lösung zu kreieren, die allen Anforderungen jedes einzelnen Nutzers gerecht wird. Der Prototyp versucht durch seine Existenz, nahezu alle Anforderungen zu erfüllen und kann durch Theming, also die Bearbeitung des aktuellen Themes, persönlich zugeschnitten werden bei Bedarf.

## 6. IMPLEMENTIERUNG EINES PROTOTYPS

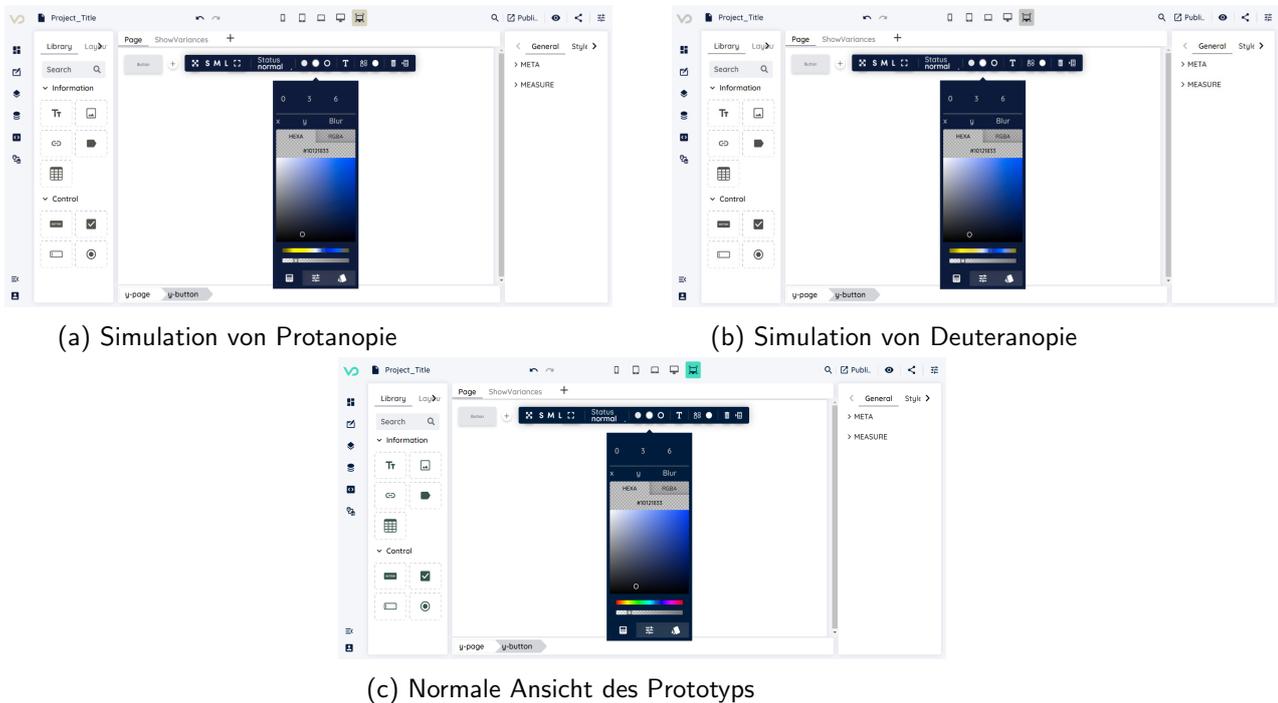


Abbildung 6.7: Überblick über den Prototyp. Quelle: Eigene Darstellung

### 6.3 Vorstellung des implementierten Prototyps

Wie in Abbildung 6.7 zu sehen, wurde ein Prototyp für yeet umgesetzt und dort integriert. Die Charakteristiken des Prototyps belaufen sich dabei auf die Anforderungen, welche zuvor erarbeitet wurden.

Auf Basis des Light-Themes wurden die nicht konformen Komponenten der Bedienoberfläche optimiert und so angepasst, dass sie den Anforderungen gerecht werden. In der fertigen Anwendung kann der Nutzer ein Theme seiner Wahl wählen und dabei auch den entwickelten Prototyp zu einem beliebigen Zeitpunkt auswählen.

Besteht Bedarf zu Individualisierung, so können über den Theme-Editor Anpassungen vorgenommen werden, um den eigenen Bedürfnissen entgegenzukommen. Dies gewährleistet einerseits mehr Komfort, andererseits wird dadurch auch die Anforderung Theming (Siehe Kapitel 5.1.5) des Anforderungskataloges erfüllt. Der Theme-Editor war bereits Bestandteil der Anwendung und so musste dies nicht ergänzend entwickelt werden, sondern erweitert den erarbeiteten Prototyp. Mithilfe von User-Tests und dem Sammeln von Analyseergebnissen wird das Ergebnis noch weiter verfeinert und mit der Zeit immer mehr optimiert.

## Kapitel 7

# Durchführung eines Usability-Tests

### 7.1 Sinnhaftigkeit des Usability-Tests

Nach der Analyse von Konkurrenzprodukten, der Definition der Anforderungen an einen Prototyp und der Implementation in yeet fehlt nur noch ein letzter Schritt. Die erreichten Resultate sind wenig aussagekräftig, wenn keine Menschen mit Farbsehschwäche den Prototyp im Einsatz testen. Sinn dieser Arbeit ist es, die Arbeit für die Betroffenen zu optimieren und nachhaltig zu erleichtern. Denn ein theoretisches Konstrukt, auf dem der Prototyp basiert, kann immer noch Fehler enthalten und so erscheint es sinnvoll, die Wirkung und die Qualität zu überprüfen. Sinn der benutzerorientierten Arbeit ist es nämlich auch, die erreichten Resultate mit Betroffenen zu überprüfen und durch das Feedback zu optimieren. Eine Methode, dies durchzuführen, ist ein sogenannter Usability-Test, durch welchen Benutzer dabei beobachtet werden, wie sie bestimmte Aufgaben mit dem realisierten Prototyp erledigen. Durch die Observation und anschließendes Feedback können Qualität und Funktionsweise des Prototyps besser beurteilt und Qualitäten sowie Probleme dokumentiert werden [RF13, vgl.:26].

### 7.2 Aufbau des Usability-Tests

Zu Beginn eines jeden User-Testings werden Ziele definiert, die man mithilfe des Testens erreichen möchte, um einen Mehrwert durch diese Prozedur zu gewinnen. In diesem Fall stehen zwei Ziele ganz klar im Fokus des Testens. Mithilfe des Feedbacks kann die Praktikabilität und Qualität des entwickelten Prototyps und der Bedienoberfläche bewertet werden. Es soll getestet werden, ob das theoretisch erarbeitete und praktisch umgesetzte Konstrukt in der realen Anwendung die Nutzbarkeit für Menschen mit einer Farbsehschwäche erleichtert oder möglicherweise nicht dem eigenen Anspruch an Barrierefreiheit gerecht wird.

Da der Prototyp sich zunächst in einem frühen Entwicklungsstand befindet, die Oberfläche von yeet regelmäßig durch Usability Labs geprüft wird und das Auffinden von Menschen mit Farbsehschwächen durchaus diffizil ist, wurden für den User Test lediglich zwei Personen getestet. Dies kann nicht als quantitative Studie gesehen werden, da dafür schlichtweg die Anzahl der Nutzer zu gering ist, doch kann der User-Test trotzdem Aufschluss über die Qua-

lität des Prototyps geben und vermittelt zunächst eher einen ersten Eindruck. Während die erste Testperson an einer Form der Rot-/Grün-Sehchwäche leidet, ist die zweite Testperson vollkommen gesund und bildet die Kontrollgruppe. Bewusst wird auch eine Kontrollgruppe untersucht, um eine weitere These beantworten zu können. Dabei ist der Gedanke festzustellen, ob Nutzer ohne Sehbehinderung, also eine Vergleichsgruppe, ebenfalls produktiv mit dem entwickelten Prototyp arbeiten können oder möglicherweise Gegenteiliges eintritt.

Nachdem die Ziele festgelegt wurden, wurde im nächsten Schritt ein Aufgabenkatalog erarbeitet, den die Testpersonen im späteren Versuch abarbeiten sollen. Dabei sind dies Standardaufgaben, die unabhängig der Anzahl an Testnutzern definiert werden und sinnvollerweise für jeden gleich sind. Denn nur bei gleichem Workflow können auch die Daten in ein Verhältnis zueinander gesetzt und verglichen werden. Kriterium für die definierten Aufgaben ist dabei, dass diese einem realistischen Szenario entspringen, also klassischen Anwendungsszenarien entsprechen und so auch in der Zukunft häufig auftreten werden. Für diesen Zweck wurde ein Dokument angelegt, welches mit drei trivialen Aufgaben bestückt wurde, die nicht länger als 5-6 Minuten in Anspruch nehmen sollten, also nicht zu komplex, aber doch fordernd für die Testpersonen sind. Dieses Dokument wurde zu Beginn der Testdurchführung an die Testnutzer übergeben und eine Lesedauer von 5 Minuten terminiert, die dazu dienen sollte, die Aufgaben zu studieren und gegebenenfalls Fragen dazu zu beantworten [RF13, vgl.:80]. Da yeet auf sogenanntes Theming setzt und der Benutzer über eine Dropdown-Liste in der Kopfleiste auswählen kann, welches Design die Oberfläche hat, wurde dort als alternatives Design der entwickelte Prototyp ergänzt, der die Oberfläche für Menschen mit einer Farbblindheit/Sehchwäche optimiert. So erhielten die Nutzer zu Beginn einen Eindruck über die klassische Oberfläche und konnten daraufhin zum eigentlichen Prototyp wechseln. Während der Bearbeitung der Aufgaben wurden die Testperson darum gebeten, ihre Gedanken laut mitzuteilen und möglichst rege zu kommunizieren. Als Durchführungszeit wurde eine Gesamtdauer von ca. 30-40 Minuten angesetzt, um dem Nutzer die Möglichkeit zu geben, zwischendurch pausieren zu können und zu Beginn die Regeln der Testung gemeinsam zu besprechen [RF13, vgl.:81].

Nach Beendigung der Testung, folgte auf diese ein abschließendes Evaluationsgespräch. Unter der Prämisse, Feedback über das durchgeführte Testen und den Prototyp zu sammeln, wurde sich verbal ausgetauscht. Für dieses Gespräch wurden weitere 10 Minuten terminiert, um Fragen ausführlich beantworten zu können und ein möglichst breites Meinungsbild einzufangen. Mit dem Hintergedanken zwei zuvor definierte Thesen zu beantworten, wurde dem jeweiligen Testnutzer eine auf ihn zugeschnittene Frage gestellt.

So galt es für den Testnutzer mit der Rot-/Grün-Schwäche die Qualität des Prototyps zu beurteilen und diesbezüglich zu kommentieren, inwiefern die Oberfläche seinen Anforderungen gerecht werden kann und ob der Prototyp bzw. die Bedienoberfläche der Plattform yeet seine Arbeit erleichtert oder möglicherweise erschwert. Ziel war es zu beurteilen, ob eingeschränkte Nutzer das Produkt besser nutzen können, als es zuvor der Fall war. Für die Vergleichsgruppe wurde eine ähnliche These vorbereitet, bei der zu beurteilen galt, inwiefern die Testnutzerin mit dem Prototyp zurecht käme und ob er die Arbeit für sie ebenfalls erleichtern könne oder erschwere. Da diese Testperson außerdem Entwicklerin im Team der Vectorsoft AG ist und an yeet mitarbeitet, kennt sie den aktuellen Workflow und kann dies daher besser vergleichen.

Über die mündliche Diskussion hinaus wurde ebenfalls ein kurzer Fragebogen mit Google Forms vorbereitet, über den die Testnutzer eine abschließende Beurteilung durchführen konnten. Diese Maßnahme ist eine weitere Qualitätssicherungsmaßnahme und hat die spätere Auswertung der Daten erleichtert. Innerhalb des Fragebogens wurden Fragen beantwortet, die sich mit der Darstellungsweise der Oberfläche befassen und vor allem auf Erfolgskriterien der Richtlinie 1.4 der WCAG eingehen. Insbesondere die Differenzierbarkeit von Inhalten gegenüber dem Hintergrund und die Größe von Elementen spielen hierbei eine besondere Rolle. Mithilfe der Diskussion und des Fragebogens konnte herausgefunden werden, ob die Testperson die Informationen der Oberfläche ausreichend wahrnehmen konnte und die Nutzbarkeit der Anwendung gegeben ist [RF13, vgl.:83].

### 7.3 Ergebnisse des Usability-Tests

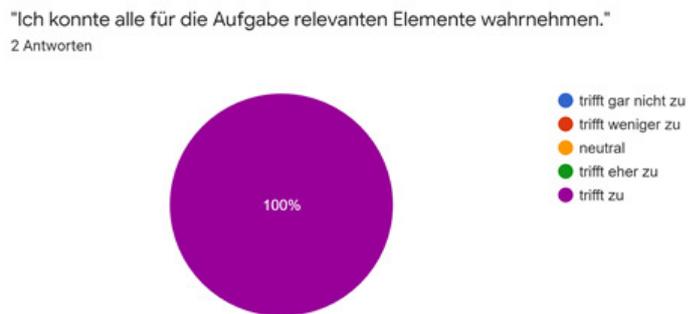
Zur Qualitätssicherung und nachhaltigen Verbesserung des angefertigten Prototyps wurde, wie bereits erwähnt, ein Usability-Test durchgeführt. Dafür wurde einerseits ein Testnutzer eingeladen, der unter einer klassischen Rot-/Grün-Schwäche leidet und daher genau in die Zielgruppe hineinpasst und andererseits eine Testnutzerin, die unter keinerlei Sehbehinderung leidet und damit die Vergleichsgruppe bildet. Die Nutzer wurden mit der Anwendung vertraut gemacht und schließlich in das Nutzungsszenario eingeführt. Die Testung wurde dabei direkt am Arbeitsplatz der Nutzer durchgeführt, um ein möglichst detailgetreues und natürliches Umfeld zu haben.

Nach der kurzen Einführung in das Produkt und die Idee des Prototyps wurde den Testnutzern das Aufgabenblatt überreicht und es konnte begonnen werden. Zur besseren Dokumentation und späteren Verwertung wurde das Szenario auditiv aufgezeichnet und zuvor die entsprechende Einwilligung der Nutzer dafür eingeholt.

Durch eine gewisse Technikaffinität konnten beide Nutzer jede einzelne Aufgabe in der dafür vorgesehenen Zeit zufriedenstellend erarbeiten. Dies war zwar nicht Ziel der Testung, ist allerdings ein Indikator für eine gute Praktikabilität des Prototyps und der Bedienoberfläche von yeet. Der Nutzer mit einer Farbsehschwäche in Form einer Rot-/Grün-Schwäche, nachfolgend Testnutzer 1 genannt, gab im anschließenden Bewertungsbogen an, dass die ihm gestellten Aufgaben fordernd, aber durchaus einfach zu lösen waren und vor allem die Bedienbarkeit und Wahrnehmbarkeit der Oberfläche dabei geholfen habe. So gab er an, dass er alle für die Aufgabe relevanten Elemente wahrnehmen konnte (Siehe Abbildung 7.1). Die Übersichtlichkeit der Oberfläche sei laut Angaben von Testnutzer 1 gegeben und man könne jedes Element gut erkennen und Vordergrund von Hintergrund differenzieren. Insbesondere sei die Größe der Bedienelemente sehr zufriedenstellend, wobei die sehr große Schriftgröße des Anforderungskataloges zwar positiv sei, aber eine kleinere Schriftgröße bei der vorhandenen Sehschwäche auch ausreichend sei, da der hohe Kontrast die Wahrnehmbarkeit enorm erleichtere. Zuletzt gab Testnutzer 1 an, dass Inhalte auch durchaus ohne Farbe wahrgenommen werden können und seiner Meinung nach die Farbe nicht das ausschließliche Transportmedium für Informationen sei. Die Mischung aus hohem Kontrast und einer großen Schriftgröße mache die Oberfläche sehr übersichtlich und leicht durchschaubar. Die Vergleichsgruppe, also die Testnutzerin ohne Sehbehinderung, gab ähnliches Feedback

## 7. DURCHFÜHRUNG EINES USABILITY-TESTS

---



**Abbildung 7.1:** Diagramm aus dem Bewertungsbogen des Prototyps, Quelle: Eigene Darstellung

wie Testnutzer 1. So seien auch für sie alle Elemente der Bedienoberfläche gut wahrnehmbar und man könne den Vordergrund optimal vom Hintergrund differenzieren. Ähnlich wie Testnutzer 1 hob sie außerdem die große Schriftgröße hervor, die zwar nicht unbedingt benötigt sei, die Lesbarkeit an gewissen Stellen aber enorm verbessere. Die Elemente seien insgesamt ausreichend groß dargestellt, was ebenfalls die Wahrnehmbarkeit verbessere (Siehe Abbildung 7.2). Kritikpunkt sei laut Testnutzerin 2 lediglich, dass noch nicht alle Elemente optimiert wären. Dies liegt allerdings daran, dass sich die Anwendung aktuell in der Entwicklungsphase befindet und noch nicht jedes Element der Oberfläche fertig ausgearbeitet und programmiert ist.



**Abbildung 7.2:** Diagramm aus dem Bewertungsbogen des Prototyps, Quelle: Eigene Darstellung

### 7.3.1 Ergebnisse der aufgestellten Thesen

**“Eine Person mit Farbsehschwäche kann nach der Überarbeitung besser mit yeet arbeiten als vorher.“**

Durch das zuvor durchweg positiv geäußerte Feedback und die Zufriedenheit und Sicherheit bei der Bewältigung der Arbeit, konnte zusammen festgestellt werden, dass die veränderte Bedienoberfläche die Arbeit für Testnutzer 1 durchaus erleichtert. Der Faktor Schriftgröße und die hohen Kontrastverhältnisse machen die Oberfläche gut differenzierbar und damit besser wahrnehmbar. Einerseits fördert die bessere Differenzierbarkeit eine höhere Effizienz, da Elemente schneller erkannt werden können, andererseits kann bedingt dadurch auch effektiver und souveräner mit der Anwendung umgegangen werden, da Komfort auch eine gewisse Sicherheit im Umgang mit der Plattform suggeriert.

Um die Stellungnahme zur These zu festigen gilt es, dieser in folgenden Testungen immer wieder nachzugehen und anhand des geäußerten Feedbacks weitere Schlüsse dazu zu ziehen. Stand jetzt konnte der entwickelte Prototyp allerdings die Bedienbarkeit für Testnutzer 1 verbessern und so die gewünschten Auswirkungen bestätigen.

**“Eine Person ohne Einschränkungen kann nach der Überarbeitung mindestens genauso gut mit yeet arbeiten wie vorher.“**

Der Testnutzerin der Vergleichsgruppe wurde eine zweite These vorgestellt, wozu sie Feedback geben sollte. Ihr wurde die Frage gestellt, ob sie nach der Überarbeitung der Bedienoberfläche und ihrer Optimierung hinsichtlich der Wahrnehmbarkeit mindestens genauso gut mit yeet arbeiten könne, wie es zuvor möglich war. Da sie Teil des Entwicklerteams von yeet ist und somit die Anwendung und die Bedienbarkeit kennt, ist sie qualifiziert, solch eine Frage beantworten zu können. Insgesamt sei für sie die Nutzungserfahrung vor und nach Implementierung des Prototyps sehr gut und sie könne effizient mit der Oberfläche von yeet arbeiten. Insbesondere die erhöhte Schriftgröße erleichtere die Wahrnehmbarkeit in vielen Fällen und verbessere die Lesbarkeit einiger Bedienelemente.

Die Veränderung einiger Farben mache die Oberfläche zwar weniger “bunt“ und dadurch weniger “hübsch“, allerdings sei dies kein Kriterium, welches die Arbeit mit yeet erschwere. Solange die Veränderung des Prototyps die Formatierung nicht zerstöre und die Übersicht gegeben sei, könne sie die Aufgaben genauso effizient und schnell wie davor erledigen.

Als erste Schlussfolgerung lässt sich daraus ziehen, dass der entwickelte Prototyp eine mindestens genauso gute, wenn nicht sogar bessere Nutzungserfahrung für Nutzer ohne Sehhinderung bietet und damit die These bestätigt werden kann. In weiteren User-Testings in Zukunft gilt es, neben betroffenen Nutzern, eine Vergleichsgruppe aufrecht zu erhalten und die These weiter zu beobachten.



# Kapitel 8

## Ergebnisse

### 8.1 Vorwort zum Prototyp

Der für yeet entwickelte Prototyp basiert auf den Ergebnissen der Analyse von Konkurrenzprodukten und dem Anforderungskatalog der WCAG. Eine Kombination der Ergebnisse und der WCAG hat den erstellten Anforderungskatalog hervorgebracht, welcher nicht nur Basis für diesen Prototyp ist, sondern auch für andere Entwickler von Relevanz sein soll.

### 8.2 Beurteilung der Qualität des Prototyps

Durch den durchgeführten User-Test und das damit verbundene Feedback der Testnutzer kann eine abschließende Beurteilung der Qualität und der Nutzbarkeit des Prototyps vollzogen werden.

Demnach erleichtert und verbessert er die Nutzungserfahrung nicht nur für die zu Beginn definierte Zielgruppe, sondern auch für Nutzer ohne jegliche Sehbeeinträchtigung. Neben diesen positiven Argumenten sticht jedoch ein negatives insbesondere heraus. Für yeet wurde im Rahmen dieser wissenschaftlichen Arbeit ein einzelnes Theme entwickelt, dass alle Anforderungen aus dem Katalog erfüllt. Ein Gedanke wäre, dies weiterführend zu individualisieren und damit passender für kleinere Gruppen an Ziel-Nutzern zu machen. Die Testnutzerin der Vergleichsgruppe merkte an, dass es für sie von Vorteil sei, wenn beispielsweise der Nutzer seine besonderen Kriterien an die Oberfläche bestimmen könne und sich dahingehend die Charakteristik der Bedienoberfläche anpassen würde. Denn nicht alle Anforderungen aus dem erarbeiteten Katalog sind für jeden Nutzer nötig und über diesen zuvor erwähnten Ansatz könnte eine noch passendere Nutzungserfahrung erreicht werden. So kann beispielsweise ein Nutzer mit einer Rot-/Grün-Schwäche und einer sehr geringen Sehkraft festlegen, dass diese beiden Faktoren beim Aufbau der Bedienoberfläche berücksichtigt werden.

Zwar kann im Nachhinein eine Anpassung über den integrierten Theme-Editor vorgenommen werden, doch stellt dies einen Mehraufwand für den Nutzer dar, den vermutlich nicht jeder bereit ist, zu leisten. Über die Möglichkeit, bestimmte Beeinträchtigungen im Vorhinein auszuwählen, könnte das Theming bereits im Aufbau der Bedienoberfläche greifen und der

Nutzer müsste im Nachhinein vermutlich weniger über den Editor anpassen. So hätte man de facto immer noch ein einziges Theme, dass die Charakteristiken der WCAG hinsichtlich der Wahrnehmbarkeit umsetzen kann, allerdings hat der Nutzer selbst in der Hand, welche umgesetzt und angewendet werden.

### 8.3 Interpretation des Usability-Tests

Im Allgemeinen war die Durchführung eines Usability-Testings sehr hilfreich, um die Qualität des Prototyps beurteilen zu können und nebenbei die zuvor aufgestellten Thesen beantworten zu können. Zwar war die Anzahl der Teilnehmer sehr begrenzt, doch sind auch wenige Daten schon ausreichend, um eine momentane Beurteilung durchzuführen und einen ersten Eindruck zum Produkt zu erhalten. Dabei konnten die in der vorherigen Sektion festgestellten Erkenntnisse zum Prototyp gewonnen werden, die in zukünftigen Testungen um noch mehr Daten ergänzt werden können.

Zu beiden aufgestellten Thesen konnten Erkenntnisse gewonnen werden, die die Bestätigung oder Widerlegung dieser ermöglichen.

#### 8.3.1 **“Eine Person mit Farbsehschwäche kann nach der Überarbeitung besser mit yeet arbeiten als vorher.“**

Da Testnutzer 1, also der Nutzer mit einer Farbsehschwäche signifikante Verbesserungen der Differenzierbarkeit feststellen konnte, kann diese These durchaus bestätigt werden. Für ihn ist die Bedienoberfläche passend gestaltet und ermöglicht eine effektive und effiziente Arbeit mit der Low-Code-Plattform yeet.

#### 8.3.2 **“Eine Person ohne Einschränkungen kann nach der Überarbeitung mindestens genauso gut mit yeet arbeiten wie vorher.“**

Auch die zweite aufgestellte These kann nach der Testung bestätigt werden. So kann die Testnutzerin der Vergleichsgruppe genauso gut und effizient mit yeet arbeiten, wie sie es ohne den Prototyp kann. Dabei bietet der WCAG konforme Prototyp durchaus auch Vorteile für nicht sehbehinderte Nutzer und schränkt diese daher nicht in ihrer Effizienz ein und verschlechtert keineswegs die Funktionalität der Bedienoberfläche von yeet, sondern beeinflusst diese positiv.

# Kapitel 9

## Zusammenfassung und Ausblick

### 9.1 Zusammenfassung der erarbeiteten Ergebnisse

Das Ziel der vorliegenden Bachelorarbeit war es, einen Prototyp für einen Modus für Farbblindheit/Sehschwäche zu entwickeln und in das Produkt der Firma Vectorsoft AG zu integrieren. Dafür wurde zunächst mit einer Analyse von Konkurrenzprodukten auf dem Markt der Low-Code-Plattformen begonnen und exemplarisch drei Produkte anhand ihrer Barrierefreiheit bzw. konkret ihrer Wahrnehmbarkeit anhand der Richtlinie 1.4 der WCAG untersucht. Auf Grundlage dieser Ergebnisse konnte im Verbund mit den Erfolgskriterien der WCAG und den empfehlenswerten Techniken der untersuchten Plattformen ein Katalog an Anforderungen erarbeitet werden, der anderen Entwicklern eine Hilfe bei der Implementierung von Wahrnehmbarkeit bieten kann.

Auf Basis des erarbeiteten Kataloges konnte ein Prototyp in das Produkt yeet implementiert werden, der die Wahrnehmbarkeit der Oberfläche nachhaltig für Nutzer verbessern soll. Dafür wurde das Corporate Design des Produktes als Basis genommen und hinsichtlich der Forderungen aus dem erarbeiteten Katalog angepasst und kann schließlich von Nutzern verwendet werden. Abschließend wurde ein Usability-Test durchgeführt, der der Qualitätssicherung und der Beantwortung der aufgestellten Thesen diente. Im Rahmen der durchgeführten Testung wurden ein farbenblinder Nutzer und eine Vergleichsgruppe mit einer vollkommen gesunden Nutzerin befragt. Diese wurden einem klassischen Testszenario ausgesetzt und haben im Anschluss in einem Bewertungsbogen und einer Nachbesprechung Feedback gegeben.

Aus der Analyse der Konkurrenzprodukte hat sich dabei ergeben, dass jedes untersuchte Produkt Schwächen hinsichtlich der Wahrnehmbarkeit der eigenen Bedienoberfläche aufweist und diese die Arbeit für Nutzer mit einer Sehbeeinträchtigung erschweren können. Außerdem konnte festgestellt werden, welche Charakteristiken insbesondere für Low-Code-Plattformen wichtig sind und berücksichtigt werden müssen.

Durch die durchgeführte Analyse konnte der Katalog an Anforderungen definiert werden, anhand dessen der Prototyp entwickelt wurde. Im Rahmen der Implementierung konnten auch die Schwachstellen von yeet festgestellt werden, welche durch den Prototyp verbessert und optimiert werden. Gerade im Bereich Kontrast und Schriftgröße weisen alle untersuchten Konkurrenzprodukte und yeet signifikante Schwächen auf, die der Prototyp aufgreift und

zumindest bei yeet beseitigt.

Der zum Schluss durchgeführte Usability-Test hat darüber hinaus bestätigt, dass der entwickelte Prototyp die Arbeit mit yeet für Nutzer mit einer Sehbeeinträchtigung erleichtert und darüber hinaus auch für gesunde Benutzer Vorteile in der Nutzungserfahrung bringt. Faktoren wie eine ausreichend große Schriftgröße und ein gutes Kontrastverhältnis machen Elemente für Nutzer leichter erkennbar und steigern die Effizienz, da keine Zeit mit der Identifikation der Bedienelemente verloren geht.

Durch diese wissenschaftliche Arbeit konnte mithilfe des entwickelten Prototyps die Wahrnehmbarkeit der Bedienoberfläche von yeet nachhaltig optimiert werden. Außerdem wurden Charakteristiken festgestellt, die bei Low-Code-Plattformen zu berücksichtigen sind, wenn diese für eine möglichst breite Masse an Nutzern wahrnehmbar sein sollen. Dabei bietet der angefertigte Anforderungskatalog eine Orientierung für andere Entwickler, die ein ähnliches Vorhaben umsetzen möchten. Insgesamt konnte damit der Stellenwert von Barrierefreiheit und insbesondere Wahrnehmbarkeit von Oberflächen festgestellt werden, die die Nutzung für jeden Nutzertyp erleichtern kann.

## 9.2 Die Zukunft von yeet

### 9.2.1 Ausrichtung hinsichtlich Barrierefreiheit

Mit dem erarbeiteten Prototyp wurde lediglich ein Grundstein hinsichtlich einer umfassenden Barrierefreiheit gelegt. Natürlich ist es langfristig das Ziel, aus der durchgeführten Analyse von Konkurrenzprodukten und den offiziellen Vorschriften eine vollständige Barrierefreiheit in das neu entwickelte Produkt zu integrieren. Dies macht das Produkt yeet nicht nur attraktiver für eine breitere Masse an potenziellen Nutzern, sondern positioniert das Produkt auch in einer Art Vorreiterrolle. Es ist aus ökonomischer Sicht also durchaus sinnvoll, wenn yeet eine umfassend barrierefrei gestaltete Oberfläche anbietet, da langfristig attraktiver für neue Kunden geworben werden kann und Kunden mit einer Behinderung jeglicher Art effizienter arbeiten können. Die gesteigerte Effizienz und verbesserte Bedienbarkeit kann außerdem darin münden, dass Kunden das Produkt viel häufiger weiterempfehlen und weitere neue Kunden für das Unternehmen anwerben. Wichtig ist es nur, dass bereits sehr früh das Thema Barrierefreiheit beachtet wird, da eine nachträgliche Integration einen unnötigen Mehraufwand bedeutet, wie dies bereits [HS10, vgl.:90] in ihrer Arbeit erläutern. Ziel ist und soll es sein, bereits jetzt im Entwicklungsstadium möglichst achtsam mit barrierefreien Richtlinien umzugehen und einen vermeidbaren Mehraufwand zu umgehen.

### 9.2.2 Weiterentwicklung des Prototyps

Neben dem User-Test, der bereits im Rahmen dieser wissenschaftlichen Arbeit durchgeführt wurde, sollen noch weitere Tests folgen, um den Prototyp nachträglich noch weiter zu optimieren. Mithilfe der Ergebnisse können Schwachstellen der Bedienoberfläche identifiziert werden und es findet ein Optimierungsprozess statt. Der Rahmen für spätere Testprozeduren wird außerdem um eine breitere Masse an Nutzern erweitert, um noch mehr Daten zu erfassen und ein breiteres Meinungsbild zu erhalten. Anders als in dem bereits durchgeführten

Test macht es durchaus Sinn, neben der Anzahl der Nutzer auch die Diversität in der Art der Sehbehinderung zu erhöhen. Je mehr mögliche Nutzerszenarien getestet werden können, desto stichhaltiger kann der Prototyp optimiert werden, was wiederum den betroffenen Nutzern zu Gute kommt. Möglich wäre es, über eine Agentur an neue Tester zu kommen, wobei die effizientere Variante wäre, sich diese Nutzer selbst zu suchen und eine Kartei damit anzulegen. Langfristig ist das Kosten-/Nutzen Verhältnis dieser Variante optimaler und attraktiver für das Unternehmen.

### 9.3 Entwicklung von Softwareergonomie und Barrierefreiheit in Zukunft

#### 9.3.1 Hilfswerkzeuge mit künstlicher Intelligenz

Tatsächlich gibt es bereits aktuell schon sehr viele Möglichkeiten für Entwickler und Unternehmen, ihre Webangebote hinsichtlich Barrierefreiheit zu überprüfen und beispielsweise anhand eines Tools oder eines Kataloges an Anforderungen zu untersuchen, inwieweit diese gewährleistet ist. Diverse Anbieter bieten dafür ihre Lösungen in diesem Marktsegment an und liefern eine breite Auswahl an Möglichkeiten für potenzielle Neukunden.

Dabei zu nennen wären einige Produkte wie das der Firma *accessiBe LTD*, welches unter anderem auf eine künstliche Intelligenz setzt, um Webangebote barrierefreier zu gestalten. Dabei besteht das Tool aus zwei Hauptkomponenten, die in gemeinsamer Arbeit den Optimierungsprozess einleiten. Zum einen gibt es ein graphisches Interface, welches für designbezogene Prozesse zuständig ist, die die zweite Komponente der Anwendung analysieren. Der zweite Bestandteil des Tools ist eine künstliche Intelligenz, die das Webangebot des Kunden analysiert und durch diese Analyse versteht, welche Sinn- und Zweckmäßigkeit bestimmte Komponenten besitzen, welche Funktionen sie ausüben und außerdem, wie man diese noch hinsichtlich der Barrierefreiheit optimieren kann. Dabei werden ganz unterschiedliche Prozesse gestartet, die unter anderem Optimierungen für Screen-Reader vornehmen, die klassische Navigation per Tastatur optimieren oder beispielsweise Bildern anhand eines Bilderkennungsalgorithmus korrekte und passende Alternativtexte hinzufügen [acc21, vgl.]. Die Anpassungen über das Interface sind dabei vollkommen konform mit gängigen Richtlinien wie der WCAG und sorgen so dafür, dass die Oberfläche diesen Anforderungen gerecht werden kann [acc21, vgl.]. Der Ansatz, eine künstliche Intelligenz anzulernen und für sehr komplexe Prozesse zu verwenden, ist sehr vielversprechend und keinesfalls ein neuer Gedanke. So werden künstliche Intelligenzen immer ausgereifter und sind mittlerweile ein elementarer Bestandteil vieler Webangebote. Die Möglichkeiten sind schier unerschöpflich und bieten Entwicklern die Chance, den Prozess der Prüfung hinsichtlich Barrierefreiheit immer weiter zu optimieren und auf ein neues Level zu erheben. Von daher ist zu erwarten, dass die Menge an solchen Testing-Tools stetig steigen wird, da es auch die Nachfrage an Software und insbesondere an barrierefreien Webangeboten tut. Inwiefern diese Tools bereits auf komplexere Software optimiert sind und ob solche Angebote aktuell von Low-Code Plattformen wahrgenommen werden können, ergibt sich aus dem Inserat der Webseite leider nicht. Da der Markt an Low-Code Plattformen allerdings stetig wächst, liegt es nahe, solche

Tools an komplexere Anwendungen anzupassen. Dies dürfte durch die Arbeit mit künstlicher Intelligenz um einiges leichter geschehen, da diese stetig lernen kann und sich so an immer komplexere Anwendungen anpassen kann, um zu verstehen.

### 9.3.2 Schulungen und Testing durch eigenes Personal

Eine weitere Alternative sind Schulungen, welche durch Experten im Gebiet der Barrierefreiheit von Software durchgeführt werden und dazu dienen, die Entwickler eines Unternehmens in diesem Gebiet weiterzubilden. Dadurch wird das Bewusstsein geschärft und das Fachwissen der Entwickler geschult, sodass man Kosten für externe Tools spart. Eine Firma, die ein solches Angebot für andere Unternehmen zur Verfügung stellt, ist die Firma TWIN CUBES GmbH, die Schulungen für Unternehmensbeschäftigte anbietet, um diese im Thema Barrierefreiheit zu unterrichten. Das Unternehmen gehört als Prüfstelle dem BITV-Test-Prüfverband an und bietet neben dieser Dienstleistung auch Testmöglichkeiten an. Die Prüfer sind dabei zertifiziert und können so ihr Fachwissen in die Schulungen und die Tests einbringen [twi20, vgl.]. Resultat solcher Schulungen sind Beschäftigte, die sich zum Thema Barrierefreiheit weiterbilden und in Zukunft bewusster und vermutlich effizienter in solchen Bereichen arbeiten und beispielsweise ein Webangebot besser optimieren können, um das Konsultieren von externen Experten zu reduzieren oder gar obsolet zu machen.

### 9.3.3 Fazit zu beiden Ansätzen

Anders als die automatisierten Testungen durch Tools sind Schulungen vermutlich auf Dauer gesehen kostengünstiger, allerdings von der Qualität der eigenen Beschäftigten abhängig und wie diese die Informationen aus dem Unterricht verinnerlichen. Zusätzlich zu Beratungsangeboten durch Experten oder Hilfswerkzeugen ist es eine gute Sache, seine Beschäftigten weiterzubilden und beispielsweise auch zum Thema Barrierefreiheit zu schulen.

Als Ergänzung zu solchen Schulungen ist es durchaus sinnvoll, als weitere Möglichkeit ein Hilfswerkzeug in Betracht zu ziehen, da Maschinen zuverlässiger sind, was die Fehlerquote angeht. So ist es eine gute Unterstützung, wenn die Arbeit der Beschäftigten durch Werkzeuge erleichtert und verbessert werden kann. Klarer Vorteil von Tools, die auf künstliche Intelligenz setzen, ist die Genauigkeit und die Möglichkeit, Dinge zu entdecken, die dem menschlichen Auge unter gewissen Umständen verborgen bleiben.

Zusammenfassend lässt sich also feststellen, dass eine Kombination aus Schulungen und externen Hilfswerkzeugen die vermutlich beste und gründlichste Vorgehensweise ist, um die Webangebote von Firmen langfristig und nachhaltig barrierefreier zu optimieren. Gerade das Gebiet der künstlichen Intelligenz ist unfassbar vielversprechend und durchläuft einen enormen Entwicklungsprozess, auch bedingt durch das vorhandene Potenzial und den Popularitätsgewinn. Dies macht diese Technologie langfristig sehr vielversprechend und kann vielen Bereichen zu Gute kommen.

# Anhang A

## Anhang

### A.1 Dokumente

Unter folgendem Link können alle relevanten PDF-Dokumente abgerufen werden, auf die im Laufe dieser Arbeit verwiesen wird: <https://drive.google.com/drive/folders/1QWFNon5wvXlgiwVlS3zRhqSvud7tH6AF?usp=sharing>.

Alternativ befinden sich alle relevanten Dokumente noch einmal auf den hinterlegten USB-Sticks.



# Glossar

CRM	Customer Relationship Management, also “die ganzheitliche Bearbeitung der Beziehung eines Unternehmens zu seinen Kunden“ [HUD17, :7]
USP	Unique Selling Proposition, ein rationales Produktmerkmal [Kun00, vgl.:90]
ISO	ein globales Netzwerk aus den weltführenden Standardisierern [int, vgl.]
RAD	Rapid Application Development - eine Methode, die auf iterative Entwicklung und Erstellung von Prototypen basiert [idg10, vgl.]
SaaS	Software as a System - Kunden können über das auf Angebote zugreifen, die von einem Service-Provider gehostet werden. [idg20, vgl.]
SEO	Suchmaschinenoptimierung, “beschreibt Techniken und Strategien, mit denen die Positionierung einer Website zu bestimmten Suchbegriffen (Keywords) in einer Suchmaschine wie Google erhöht werden sollen“ [Dol21, vgl.]
Industrie 4.0	“International steht Industrie 4.0 heute für die Digitalisierung der Industrie“ [bmb16]
WYSIWYG	“What you see is what you get“, der Text wird dargestellt, wie er auch im Editor erscheint [ion18, vgl.]



# Literaturverzeichnis

- [acc21] accessiBe LTD: *The Industry-Leading Web Accessibility Technology*. Online. <https://accessibe.com/product>. Version: 2021. – [abgerufen 17-August-2021, Übersetzt von Felix Wegener]
- [air] Airtable: *The power of a database with the familiarity of a spreadsheet*. Online. <https://www.airtable.com/product>. – [abgerufen 25-August-2021, übersetzt von Felix Wegener]
- [bmb16] Bundesministerium für Forschung und Bildung: *Industrie 4.0*. Online. <https://www.bmbf.de/bmbf/de/forschung/digitale-wirtschaft-und-gesellschaft/industrie-4-0/industrie-4-0>. Version: Januar 2016. – [abgerufen 4-August-2021]
- [Dol21] DOLL, Mareike: *Was ist SEO? Suchmaschinenoptimierung einfach erklärt*. Online. <https://www.luna-park.de/blog/35752-was-ist-seo/>. Version: Januar 2021. – [abgerufen 15-August-2021]
- [FVK<sup>+</sup>16] FERATI, Mexhid ; VOGEL, Bahtijar ; KURTI, Arianit ; RAUFI, Bujar ; ASTALS, David S.: *Web Accessibility for Visually Impaired People: Requirements and Design Issues*. In: EBERT, Achim (Hrsg.) ; HUMAYOUN, Shah R. (Hrsg.) ; SEYFF, Norbert (Hrsg.) ; PERINI, Anna (Hrsg.) ; BARBOSA, Simone D. (Hrsg.): *Usability- and Accessibility-Focused Requirements Engineering*. Cham : Springer International Publishing, 2016. – ISBN 978–3–319–45916–5, S. 79–96. – [übersetzt von Felix Wegener]
- [Her18] HERCZEG, Michael: *Software-Ergonomie - Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme*. 4. Berlin : Walter de Gruyter GmbH & Co KG, 2018. – ISBN 978–3–110–4468–69
- [HS10] HARDT, Anett ; SCHREPP, Martin: *Barrierefreiheit betriebswirtschaftlicher Anwendungen - Diskussion spezieller Anforderungen und Lösungsansätze*. In: BRAU, Henning (Hrsg.) ; DIEFENBACH, Sarah (Hrsg.) ; GÖRING, Katharina (Hrsg.) ; PEISSNER, Matthias (Hrsg.) ; PETROVIC, Kostanija (Hrsg.): *Tagungsband UP10*. Stuttgart : Fraunhofer Verlag, 2010, S. 86–91

- [HUD17] *Kapitel 1.* In: HELMKE, Stefan ; UEBEL, Matthias ; DANGELMAIER, Wilhelm: *Grundlagen und Ziele des CRM-Ansatzes.* Wiesbaden : Springer Fachmedien Wiesbaden, 2017. – ISBN 978–3–658–06624–6, 3–20
- [Hä16] HÄNEL, Matthias: *Mit Blindenschrift und Computer ins Internet.* Online. <https://www.matthias-haenel.de/wazeile.html#>. Version: Juli 2016. – [abgerufen 16-August-2021]
- [idg10] IDG Business Media GmbH: *Agile Softwareentwicklung - Rapid Application Development.* Online. <https://www.computerwoche.de/a/rapid-application-development,2352552>. Version: August 2010. – [abgerufen 30-Juli-2021]
- [idg20] IDG Business Media GmbH: *SaaS - Was ist Software as a Service?* Online. <https://www.computerwoche.de/a/was-ist-software-as-a-service,3332266>. Version: April 2020. – [abgerufen 30-Juli-2021]
- [int] International Organization for Standardization: *What We Do.* Online. <https://www.iso.org/what-we-do.html>. – [abgerufen 28-Juli-2021, Übersetzt von Felix Wegener]
- [ion18] 1&1 IONOS SE: *WYSIWYG: Ich sehe was, was du auch siehst.* Online. <https://www.ionos.de/digitalguide/websites/webseiten-erstellen/was-bedeutet-wysiwyg/>. Version: September 2018. – [abgerufen 8-August-2021]
- [Kun00] KUNDE, Jesper: *Corporate Religion - Bindung schaffen durch starke Marken.* 1. Wiesbaden : Gabler Verlag, 2000. – ISBN 978–3–322–82284–0
- [Nat19] NATE, Baldwin: *Leonardo: an open source contrast-based color generator.* Online. <https://medium.com/@NateBaldwin/leonardo-an-open-source-contrast-based-color-generator-92d61b6521d2>. Version: Dezember 2019. – [abgerufen 03-September-2021, übersetzt von Felix Wegener]
- [PS05] PUHL, Steffen ; SCHWICKERT, Axel C.: *Barrierefreie Web-Nutzung durch blinde und sehbehinderte Menschen.* 2. Gießen : Professur BWL – Wirtschaftsinformatik, Justus-Liebig-Universität Gießen, 2005. ISSN 1613–6667
- [qui] Quickbase: *What is Quickbase?* Online. <https://www.quickbase.com/product/product-overview>. – [abgerufen 25-August-2021, übersetzt von Felix Wegener]
- [RF13] RICHTER, Michael ; FLÜCKIGER, Markus D.: *Usability Engineering kompakt: benutzbare Produkte gezielt entwickeln.* 3. Heidelberg : Springer-Verlag Berlin, 2013. – ISBN 978–3–642–34832–7

- [RR16] RICHARDSON, Clay ; RYMER, John R.: *Vendor Landscape: The Fractured, Fertile Terrain Of Low-Code Application Platforms*. [https://informationsecurity.report/Resources/Whitepapers/0eb07c59-b01c-4399-9022-dfc297487060\\_Forrester%20Vendor%20Landscape%20The%20Fractured,%20Fertile%20Terrain.pdf](https://informationsecurity.report/Resources/Whitepapers/0eb07c59-b01c-4399-9022-dfc297487060_Forrester%20Vendor%20Landscape%20The%20Fractured,%20Fertile%20Terrain.pdf). Version: Januar 2016. – [Übersetzt von Felix Wegener]
- [Sch20] SCHENK, Philip ; GMBH, 02100 D. (Hrsg.): *Low-Code vs. No-Code - für Unternehmen*. Online. <https://www.02100.io/ratgeber/low-code-vs-no-code>. Version: Oktober 2020. – [abgerufen 28-Juli-2021]
- [sim] Simplifier AG: *Enterprise Apps made simple*. Online. <https://simplifier.io/>. – [abgerufen 25-August-2021, übersetzt von Felix Wegener]
- [sta21] *Zahl der Woche Nr. 20 vom 18. Mai 2021*. Online. [https://www.destatis.de/DE/Presse/Pressemitteilungen/Zahl-der-Woche/2021/PD21\\_20\\_p002.html](https://www.destatis.de/DE/Presse/Pressemitteilungen/Zahl-der-Woche/2021/PD21_20_p002.html). Version: Mai 2021. – [abgerufen 07-September-2021]
- [The16] THESMANN, Stephan: *Interface Design - Usability, User Experience und Accessibility im Web gestalten*. 2. Wiesbaden : Springer Vieweg, Wiesbaden, 2016. – 43–125 S. – ISBN 978–3–658–03857–1
- [tpg] TPGi – a Vispero™ Company: *Colour Contrast Analyser (CCA)*. Online. <https://www.tpgi.com/color-contrast-checker/>. – [abgerufen 25-August-2021, übersetzt von Felix Wegener]
- [twi20] TWIN CUBES GmbH: *Barrierefreiheit*. Online. <https://twin-cubes.com/>. Version: 2020. – [abgerufen 17-August-2021]
- [vn208] Vereinte Nationen: *Übereinkommen der Vereinten Nationen über Rechte von Menschen mit Behinderungen Erster Staatenbericht der Bundesrepublik Deutschland*. Version: 05 2008. [https://www.bmas.de/SharedDocs/Downloads/DE/staatenbericht-2011.pdf%3F\\_\\_blob%3DpublicationFile](https://www.bmas.de/SharedDocs/Downloads/DE/staatenbericht-2011.pdf%3F__blob%3DpublicationFile) [abgerufen 9-August-2021]
- [wca08] BEN, Caldwell (Hrsg.) ; MICHAEL, Cooper (Hrsg.) ; G., Reid L. (Hrsg.) ; GREGG, Vanderheiden (Hrsg.) ; WENDY, Crisholm (Hrsg.) ; JOSH, Slatin (Hrsg.) ; JASON, White (Hrsg.): *Richtlinien für barrierefreie Webinhalte (WCAG) 2.0*. Online. <https://www.w3.org/Translations/WCAG20-de/>. Version: Dezember 2008. – [abgerufen 2-August-2021]
- [wca18a] ANDREW, Kirkpatrick (Hrsg.) ; JOSHUE, O C. (Hrsg.) ; ALASTAIR, Campbell (Hrsg.) ; MICHAEL, Cooper (Hrsg.) ; BEN, Caldwell (Hrsg.) ; G., Reid L. (Hrsg.) ; GREGG, Vanderheiden (Hrsg.) ; WENDY, Crisholm (Hrsg.) ; JOSH, Slatin (Hrsg.) ; JASON, White (Hrsg.): *Understanding Success Criterion 1.4.1: Use of Color*. Online. <https://www.w3.org/WAI/WCAG21/Understanding/>

- [use-of-color.html](#). Version: Juni 2018. – [abgerufen 29-August-2021, Übersetzt von Felix Wegener]
- [wca18b] ANDREW, Kirkpatrick (Hrsg.) ; JOSHUE, O C. (Hrsg.) ; ALASTAIR, Campbell (Hrsg.) ; MICHAEL, Cooper (Hrsg.) ; BEN, Caldwell (Hrsg.) ; G., Reid L. (Hrsg.) ; GREGG, Vanderheiden (Hrsg.) ; WENDY, Crisholm (Hrsg.) ; JOSH, Slatin (Hrsg.) ; JASON, White (Hrsg.): *Understanding Success Criterion 1.4.10: Reflow*. Online. <https://www.w3.org/WAI/WCAG21/Understanding/reflow.html>. Version: Juni 2018. – [abgerufen 29-August-2021, Übersetzt von Felix Wegener]
- [wca18c] ANDREW, Kirkpatrick (Hrsg.) ; JOSHUE, O C. (Hrsg.) ; ALASTAIR, Campbell (Hrsg.) ; MICHAEL, Cooper (Hrsg.) ; BEN, Caldwell (Hrsg.) ; G., Reid L. (Hrsg.) ; GREGG, Vanderheiden (Hrsg.) ; WENDY, Crisholm (Hrsg.) ; JOSH, Slatin (Hrsg.) ; JASON, White (Hrsg.): *Understanding Success Criterion 1.4.11: Non-text Contrast*. Online. <https://www.w3.org/WAI/WCAG21/Understanding/non-text-contrast.html>. Version: Juni 2018. – [abgerufen 29-August-2021, Übersetzt von Felix Wegener]
- [wca18d] ANDREW, Kirkpatrick (Hrsg.) ; JOSHUE, O C. (Hrsg.) ; ALASTAIR, Campbell (Hrsg.) ; MICHAEL, Cooper (Hrsg.) ; BEN, Caldwell (Hrsg.) ; G., Reid L. (Hrsg.) ; GREGG, Vanderheiden (Hrsg.) ; WENDY, Crisholm (Hrsg.) ; JOSH, Slatin (Hrsg.) ; JASON, White (Hrsg.): *Understanding Success Criterion 1.4.12: Text Spacing*. Online. <https://www.w3.org/WAI/WCAG21/Understanding/text-spacing.html>. Version: Juni 2018. – [abgerufen 29-August-2021, Übersetzt von Felix Wegener]
- [wca18e] ANDREW, Kirkpatrick (Hrsg.) ; JOSHUE, O C. (Hrsg.) ; ALASTAIR, Campbell (Hrsg.) ; MICHAEL, Cooper (Hrsg.) ; BEN, Caldwell (Hrsg.) ; G., Reid L. (Hrsg.) ; GREGG, Vanderheiden (Hrsg.) ; WENDY, Crisholm (Hrsg.) ; JOSH, Slatin (Hrsg.) ; JASON, White (Hrsg.): *Understanding Success Criterion 1.4.13: Content on Hover or Focus*. Online. <https://www.w3.org/WAI/WCAG21/Understanding/content-on-hover-or-focus.html>. Version: Juni 2018. – [abgerufen 29-August-2021, Übersetzt von Felix Wegener]
- [wca18f] ANDREW, Kirkpatrick (Hrsg.) ; JOSHUE, O C. (Hrsg.) ; ALASTAIR, Campbell (Hrsg.) ; MICHAEL, Cooper (Hrsg.) ; BEN, Caldwell (Hrsg.) ; G., Reid L. (Hrsg.) ; GREGG, Vanderheiden (Hrsg.) ; WENDY, Crisholm (Hrsg.) ; JOSH, Slatin (Hrsg.) ; JASON, White (Hrsg.): *Understanding Success Criterion 1.4.3: Contrast (Minimum)*. Online. <https://www.w3.org/WAI/WCAG21/Understanding/contrast-minimum.html>. Version: Juni 2018. – [abgerufen 29-August-2021, Übersetzt von Felix Wegener]
- [wca18g] ANDREW, Kirkpatrick (Hrsg.) ; JOSHUE, O C. (Hrsg.) ; ALASTAIR, Campbell (Hrsg.) ; MICHAEL, Cooper (Hrsg.) ; BEN, Caldwell (Hrsg.) ; G., Reid L. (Hrsg.) ; GREGG, Vanderheiden (Hrsg.) ; WENDY, Crisholm (Hrsg.) ; JOSH, Slatin (Hrsg.) ; JASON, White (Hrsg.): *Understanding Success Criterion 1.4.4:*

- Resize text*. Online. <https://www.w3.org/WAI/WCAG21/Understanding/resize-text.html>. Version: Juni 2018. – [abgerufen 29-August-2021, Übersetzt von Felix Wegener]
- [wca18h] ANDREW, Kirkpatrick (Hrsg.) ; JOSHUE, O C. (Hrsg.) ; ALASTAIR, Campbell (Hrsg.) ; MICHAEL, Cooper (Hrsg.) ; BEN, Caldwell (Hrsg.) ; G., Reid L. (Hrsg.) ; GREGG, Vanderheiden (Hrsg.) ; WENDY, Crisholm (Hrsg.) ; JOSH, Slatin (Hrsg.) ; JASON, White (Hrsg.): *Understanding Success Criterion 1.4.5: Images of Text*. Online. <https://www.w3.org/WAI/WCAG21/Understanding/images-of-text.html>. Version: Juni 2018. – [abgerufen 29-August-2021, Übersetzt von Felix Wegener]
- [wca18i] ANDREW, Kirkpatrick (Hrsg.) ; JOSHUE, O C. (Hrsg.) ; ALASTAIR, Campbell (Hrsg.) ; MICHAEL, Cooper (Hrsg.) ; BEN, Caldwell (Hrsg.) ; G., Reid L. (Hrsg.) ; GREGG, Vanderheiden (Hrsg.) ; WENDY, Crisholm (Hrsg.) ; JOSH, Slatin (Hrsg.) ; JASON, White (Hrsg.): *Understanding Success Criterion 1.4.6: Contrast (Enhanced)*. Online. <https://www.w3.org/WAI/WCAG21/Understanding/contrast-enhanced.html>. Version: Juni 2018. – [abgerufen 29-August-2021, Übersetzt von Felix Wegener]
- [wca18j] ANDREW, Kirkpatrick (Hrsg.) ; JOSHUE, O C. (Hrsg.) ; ALASTAIR, Campbell (Hrsg.) ; MICHAEL, Cooper (Hrsg.) ; BEN, Caldwell (Hrsg.) ; G., Reid L. (Hrsg.) ; GREGG, Vanderheiden (Hrsg.) ; WENDY, Crisholm (Hrsg.) ; JOSH, Slatin (Hrsg.) ; JASON, White (Hrsg.): *Understanding Success Criterion 1.4.8: Visual Presentation*. Online. <https://www.w3.org/WAI/WCAG21/Understanding/visual-presentation.html>. Version: Juni 2018. – [abgerufen 29-August-2021, Übersetzt von Felix Wegener]
- [wca18k] ANDREW, Kirkpatrick (Hrsg.) ; JOSHUE, O C. (Hrsg.) ; ALASTAIR, Campbell (Hrsg.) ; MICHAEL, Cooper (Hrsg.) ; BEN, Caldwell (Hrsg.) ; G., Reid L. (Hrsg.) ; GREGG, Vanderheiden (Hrsg.) ; WENDY, Crisholm (Hrsg.) ; JOSH, Slatin (Hrsg.) ; JASON, White (Hrsg.): *Understanding Success Criterion 1.4.9: Images of Text (No Exception)*. Online. <https://www.w3.org/WAI/WCAG21/Understanding/images-of-text-no-exception.html>. Version: Juni 2018. – [abgerufen 29-August-2021, Übersetzt von Felix Wegener]
- [wca18l] ANDREW, Kirkpatrick (Hrsg.) ; JOSHUE, O C. (Hrsg.) ; ALASTAIR, Campbell (Hrsg.) ; MICHAEL, Cooper (Hrsg.) ; BEN, Caldwell (Hrsg.) ; G., Reid L. (Hrsg.) ; GREGG, Vanderheiden (Hrsg.) ; WENDY, Crisholm (Hrsg.) ; JOSH, Slatin (Hrsg.) ; JASON, White (Hrsg.): *Web Content Accessibility Guidelines (WCAG) 2.1*. Online. <https://www.w3.org/TR/WCAG21/>. Version: Juni 2018. – [abgerufen 2-August-2021, übersetzt von Felix Wegener]
- [Wei93] WEISZÄCKER, Richard v.: *Ansprache von Bundespräsident Richard von Weizsäcker bei der Eröffnungsveranstaltung der Tagung der Bundesarbeitsgemeinschaft Hilfe für Behinderte*. Online. <https://www.bundespraesident>.

[de/SharedDocs/Reden/DE/Richard-von-Weizsaecker/Reden/1993/07/19930701\\_Rede.html](https://www.sharedocs.de/SharedDocs/Reden/DE/Richard-von-Weizsaecker/Reden/1993/07/19930701_Rede.html). Version: Juli 1993. – [abgerufen 07-September-2021]

- [WL12] WELSCH, Norbert ; LIEBMANN, Claus C.: *Farben - Natur, Technik, Kunst*. 3. Berlin Heidelberg New York : Springer-Verlag, 2012. – ISBN 978-3-662-56625-1