Automatische Prüfung von Websites auf Barrierefreiheit - Ein Vergleich von Monitoring-Tools

Bachelorarbeit

im Studiengang
Mobile Medien

vorgelegt von

Mona Steininger
Matr.-Nr.: 39159

am 24. August 2023
an der Hochschule der Medien Stuttgart

Erstprüfer/in: Prof. Dr. Gottfried Zimmermann
Zweitprüfer/in: Christoph Strauß

Ehrenwörtliche Erklärung

„Hiermit versichere ich, Mona Steininger, ehrenwörtlich, dass ich die vorliegende Bachelorarbeit mit dem Titel: „Automatische Prüfung von Websites auf Barrierefreiheit - Ein Vergleich von Monitoring-Tools“ selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Ebenso sind alle Stellen, die mit Hilfe eines KI-basierten Schreibwerkzeugs erstellt oder überarbeitet wurden, kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ich habe die Bedeutung der ehrenwörtlichen Versicherung und die prüfungsrechtlichen Folgen (§ 24 Abs. 2 Bachelor-SPO, § 23 Abs. 2 Master-SPO (Vollzeit)) einer unrichtigen oder Mai 2023 Prüfungsverwaltung 6 unvollständigen ehrenwörtlichen Versicherung zur Kenntnis genommen.“

Stuttgart, 24.08. 2023

Kurzfassung

Behinderte Personen können durch digitale Angebote vorhandene Barrieren ausgleichen und Zugang zu neuen Bereichen erhalten. Hierbei ist es wichtig, dass möglichst viele Personen gleichberechtigt auf die Angebote ohne neue Barrieren zugreifen. Dies spiegelt sich auch in Gesetzen und Richtlinien für Informationsangebote öffentlicher Stellen und ab 2025 auch für bestimmte private Unternehmen wider.

Für die Überprüfung der Barrierefreiheit von Websites können Automatische Monitoring Systeme (AMS) eingesetzt werden, die Websites immer wieder automatisch nach Barrieren durchsuchen. Gegenstand dieser Bachelorarbeit ist ein Vergleich von drei AMS: Pa11y, ARC Monitoring und Siteimprove.

Die AMS werden zuerst nach qualitativen Gesichtspunkten gegenübergestellt, wie zum Beispiel Preismodell, Hosting, unterstützte Richtlinien oder Einsatz von Gamification.

Um quantitative Ergebnisse zu erhalten, verglichen wir die Scanergebnisse einer selbst konstruierten Test-Website, die Verstöße für alle Success Criteria der Web Content Accessibility Guidelines (WCAG) 2.1 enthält. Die Ergebnisse wurden nach den Evaluationskriterien Coverage (Abdeckung) und Correctness (Richtigkeit) ausgewertet. Coverage umfasste die Abdeckung der WCAG Success Criteria, Principles, Guidelines und der in der EN 301549 zugeordneten betroffenen Benutzendengruppen.

Um die Usability der AMS zu analysieren, wurde eine Heuristische Evaluation nach Nielsens 10 Heuristiken durchgeführt.

Pa11y hat von allen AMS die beste Coverage, jedoch die schlechteste Correctness aufgrund vieler Falsch Positive durch unspezifische Meldungen. Eine Berechnung ohne diese verschlechterte die Coverage und verbesserte die Correctness. ARC Monitoring konnte aufgrund der schlechten Usability und mittelmäßigen quantitativen Werte am wenigsten überzeugen. Siteimprove ist das AMS mit dem größten Funktionsumfang und der besten Usability. Die quantitativen Ergebnisse sind ohne Einberechnung der Notices ähnlich gut wie die von Pa11y.

Die Coverage aller drei AMS zusammengefasst zeigt, dass sie eine manuelle Prüfung nicht ersetzen können. Die erzielten Werte zeigen selten eine Abdeckung von mehr als 50 %.

Schlagwörter: Digitale Barrierefreiheit, Automatische Monitoring Systeme, Vergleich, Usability Evaluation, Bachelorarbeit

Abstract

Disabled persons can compensate existing barriers through digital products and gain access to new spaces. It is important that as many people as possible can access these products equally and without new barriers. This is also represented in legislation and guidelines for information services of public offices and from 2025 on also for certain private companies.

Automated Monitoring Systems (AMS) can be used to check the accessibility of websites. AMS scan websites automatically and periodically. The main subject matter of this bachelor's thesis is a comparison of three AMS: Pa11y, ARC Monitoring und Siteimprove.

First, we compared the AMS in qualitative aspects such as pricing, hosting, supported guidelines or usage of gamification.

To gain quantitative data we then compared the scan results of a constructed test website that includes failures for all Web Content Accessibility Guidelines (WCAG) 2.1 success criteria. The findings were evaluated by the evaluation criteria coverage and correctness. Coverage includes coverage of WCAG success criteria, principles, guidelines and affected user groups as established in EN 301549.

To analyse the AMS' usability we conducted a Heuristic Evaluation after Nielsen's 10 Heuristics.

Pa11y has the best coverage of all AMS but the worst correctness due to many false positives from unspecific notices. Without incorporating these the calculated coverage was worse, but the correctness was better. ARC Monitoring could convince the least as its usability was bad and the quantitative outcome was mediocre. Siteimprove has the biggest range of functions and the best usability. The quantitative results are similar to Pa11y's when accounted without notices.

The coverage of all three AMS combined shows that they cannot replace a manual check. Most of the results do not exceed a coverage of 50 %.

Keywords: digital accessibility, automated monitoring systems, comparison, usability evaluation, bachelor thesis

Inhaltsverzeichnis

Ehrenwörtliche Erklärung ..... **II**

Kurzfassung ..... **III**

Abstract ..... **IV**

Inhaltsverzeichnis ..... **V**

Abbildungsverzeichnis ..... **VIII**

Tabellenverzeichnis ..... **X**

Abkürzungsverzeichnis ..... **XI**

1 Einleitung **1**

1.1 Motivation 1

1.2 Zielsetzung 2

2 Theoretische Grundlagen **3**

2.1 Gesetzliche Vorschriften in Deutschland bzgl. Barrierefreiheit 3

2.1.1 Gesetze 3

2.1.2 Standards 4

2.2 Automatische Monitoring-Systeme 5

2.2.1 Vorteile 6

2.2.2 Grenzen 7

3 Methodik **8**

3.1 Erstellung der Test-Website 8

3.1.1 Auswahl der Methode 8

3.1.2 Zusammenstellung des Fehlerkatalogs 8

3.1.3 Gestaltung der Fehler 9

3.1.4 Fehlerbeispiel 1: BITV-Prüfschritt 9.1.3.1e Datentabellen richtig aufgebaut 10

3.1.5 Fehlerbeispiel 2: Success Criterion 1.4.3 Kontraste (Minimum) 12

3.1.6 Durchführung der Scans 13

3.2 Quantitativer Vergleich der AMS 14

3.2.1 Evaluationskriterien 14

3.2.2 Coverage der WCAG 2.1 Success Criteria 15

3.2.3 Coverage der WCAG 2.1 Principles 15

3.2.4 Coverage der WCAG 2.1 Guidelines 16

3.2.5 Coverage der betroffenen Gruppen 16

3.2.6 Correctness 17

3.3 Heuristische Evaluation 18

4 Vorstellung der Tools 20
4.1 Pa11y 20
4.2 Aufbau des Pa11y Dashboards 20
4.3 ARC Monitoring 22
4.4 Aufbau von ARC Monitoring 22
4.5 Siteimprove 26
4.6 Aufbau von Siteimprove 27

5 Qualitativer Vergleich 32
5.1 Open Source 33
5.2 Hosting 34
5.3 Preismodell 34
5.4 Unterstützung ganzer Websites 35
5.5 Angebotene Sprachen 35
5.6 Unterstützte Richtlinien 35
5.7 Unterstützung zur manuellen Prüfung 36
5.8 Einsatz von Gamification 37
5.9 Angebot ergänzender Tools 37
5.10 Eingabeformate 38
5.11 Ausgabeformate 39

6 Auswertung 40
6.1 Quantitativer Vergleich der AMS 40
6.1.1 Coverage hinsichtlich der WCAG Success Criteria 40
6.1.2 Coverage hinsichtlich der WCAG Principles 42
6.1.3 Coverage hinsichtlich der WCAG 2.1 Guidelines 44
6.1.4 Coverage hinsichtlich der betroffenen Benutzendengruppen 46
6.1.5 Correctness 48
6.2 Heuristische Evaluation 50
6.2.1 Report zur Heuristischen Evaluation von Pa11y 50
6.2.2 Report zur Heuristischen Evaluation von ARC Monitoring 51
6.2.3 Report zur Heuristischen Evaluation von Siteimprove 55
6.2.4 Fazit zur Heuristischen Evaluation 55

7 Diskussion 57
7.1 Diskussion der Ergebnisse 57
7.2 Limitationen 59

8 Fazit und Ausblick 60

Anhang A: Test-Website 62
A.1 Konzeption der Test-Website 62

A.2 Code der Test-Website 62

Anhang B: Auswertung 63

B.1 Auswertungstabelle 63

Literaturverzeichnis 64

Abbildungsverzeichnis

Abbildung 1 *Zeitlicher Verlauf der gefundenen Probleme im AMS Siteimprove (eigener Screenshot)* 6

Abbildung 2 *Prüfung des Kontrastverhältnisses im WebAIM Contrast Checker (eigener Screenshot)* 13

Abbildung 3 *Ansicht der erstellten Test-Website (eigener Screenshot)* 14

Abbildung 4 *Startseite des Pa11y Dashboards (eigener Screenshot)* 20

Abbildung 5 *Übersicht einer angelegten Website in Pa11y (eigener Screenshot)* 21

Abbildung 6 *Ansicht der Fehlerbeschreibung in Pa11y (eigener Screenshot)* 22

Abbildung 7 *Ansicht des Monitoring Dashboards in ARC Monitoring (eigener Screenshot)* 23

Abbildung 8 *Visualisierung des Fortschritts in ARC Monitoring (eigener Screenshot)* 23

Abbildung 9 *Gefundene Fehler und Handlungsempfehlung in ARC Monitoring (eigener Screenshot)* 24

Abbildung 10 *Auflistung der Ergebnisse eines Scans in ARC Monitoring (eigener Screenshot)* 25

Abbildung 11 *Ansicht eines einzelnen Fehlers in ARC Monitoring (eigener Screenshot)* 26

Abbildung 12 *Accessibility Übersicht in Siteimprove (eigener Screenshot)* 28

Abbildung 13 *Auflistung von gefundenen Barrierefreiheits-Problemen in Siteimprove (eigener Screenshot)* 29

Abbildung 14 *Detailseite des Problems „Die Seitenausrichtung ist unveränderbar“ in Siteimprove (eigener Screenshot)* 30

Abbildung 15 *Ansicht des Problems "Die Seitenausrichtung ist unveränderbar" im Seitenbericht von Siteimprove (eigener Screenshot)* 31

Abbildung 16 *Anleitung zur manuellen Überprüfung durch Pa11y (eigener Screenshot)* 36

Abbildung 17 *Accessibility Score in Siteimprove (eigener Screenshot)* 37

Abbildung 18 *Coverage der WCAG Success Criteria* 40

Abbildung 19 *Meldungen, die Fehler anzeigen, die Success Criteria verletzen* 41

Abbildung 20 *Gesamte Coverage der Succes Criteria ohne Einbezug der Notices von Pa11y* 42

Abbildung 21 *Coverage nach WCAG Principles* 43

Abbildung 22 *Durchschnittliche Coverage der WCAG 2.1 Principles über alle Tools hinweg* 44

Abbildung 23 *Coverage der WCAG Guidelines der AMS* 44

Abbildung 24 *Coverage der WCAG Guidelines 1-7* 45

Abbildung 25 *Coverage der WCAG Guidelines 8-13* 45

Abbildung 26 *Durchschnittliche Coverage aller Tools nach WCAG Guidelines* 46

Abbildung 27 *Coverage hinsichtlich der betroffenen Benutzendengruppen 1-5* 46

Abbildung 28 *Coverage hinsichtlich der betroffenen Benutzendengruppen 6-9* 47

Abbildung 29 *Durchschnittliche Coverage der betroffenen Benutzendengruppen aller Tools* 48

Abbildung 30 *Correctness: Verhältnis von RP und FP der AMS* 49

Abbildung 31 *Correctness: Verhältnis von RP und FP der AMS (ohne Einbezug der Pa11y Notices)* 50

Abbildung 32 *Fehlermeldung im Pa11y Dashboard (eigener Screenshot)* 51

Abbildung 33 *Ansicht von ARC Monitoring während Inhalte geladen werden (eigener Screenshot)* 52

Abbildung 34 *Breadcrumb-Navigation in ARC Monitoring (eigener Screenshot)* 52

Abbildung 35 *Hierarchieebenen der Navigation in ARC Monitoring (eigener Screenshot)* 53

Abbildung 36 *Darstellung der Vorkommnisse eines Fehlers in den Components von ARC Monitoring* 54

Abbildung 37 *Hinweis auf Elemente im Source Code in einer Fehlermeldung in ARC Monitoring* 54

Abbildung 38 *Darstellung von gelösten Problemen und nicht relevanten Problemen in Siteimprove (eigener Screenshot)* 55

Tabellenverzeichnis

Tabelle 1 *Vergebene Wertung für die unterschiedlichen Meldungen der Tools* 15

Tabelle 2 *Verwendete Abkürzungen für die Aussagen zur funktionalen Leistungsfähigkeit* 16

Tabelle 3 *Zusammenfassende Vergleichstabelle über die Tools* 32

Tabelle 4 *Ergebnisse der Auswertung der Coverage der Principles* 43

Tabelle 5 *Ergebnisse der Auswertung der Correctness* 48

Abkürzungsverzeichnis

AMS	Automatisches Monitoring Tool für Barrierefreiheit
BGG	Bundesgleichstellungsgesetz
BFSG	Barrierefreiheitsstärkungsgesetz
CI	Continuous Integration
EAA	European Accessibility Act
FN	Falsch Negative
FP	Falsch Positive
npm	Node Package Manager
RP	Richtig Positive
SaaS	Software as a Service
UI	User Interface
WAI	Web Accessibility Initiative
WAD	Web Accessibility Directive
WCAG	Web Content Accessibility Guidelines
W3C	World Wide Web Consortium

1 Einleitung

1.1 Motivation

Nach einer Umfrage des Sinus Instituts aus dem Jahr 2019 bietet die Digitalisierung vielfältige Chancen, um die gesellschaftliche Teilhabe von behinderten Menschen zu unterstützen. Die digitalen Mittel und Technologien können Beeinträchtigungen kompensieren und Zugang zu bisher nicht erreichbaren gesellschaftlichen Bereichen ermöglichen. Dies vergrößert die Autonomie, Vernetzung und Teilhabe von behinderten personen und gibt ihnen die Möglichkeit, neue Kompetenzen und Wissen zu erwerben (*Digitale Teilhabe von Menschen mit Behinderung*, 2019).

Um dieses Potenzial auszuschöpfen, ist es unbedingt erforderlich, dass die digitalen Angebote für möglichst viele gleichberechtigt zugänglich sind und dabei keine neuen Barrieren schaffen (*Digitale Teilhabe von Menschen mit Behinderung*, 2019).

Ende 2021 waren 9,4 % der Menschen in Deutschland schwerstbehindert, in der EU hat einer oder eine von vier europäischen Erwachsenen eine Form von Behinderung (*Behinderungen in der EU*, 2022). Ein Großteil der Behinderungen tritt erst im Alter auf. Aufgrund der älter werdenden Bevölkerung ist also zusätzlich davon auszugehen, dass sich die Zahl der behinderten Menschen vergrößert.

Auch von gesetzlicher Seite gibt es immer mehr Vorgaben, die öffentliche Stellen und in Zukunft auch private Dienstleistende dazu verpflichten, ihre Informationsangebote und Produkte barrierefrei zugänglich zu machen. Ab 2025 gelten durch den European Accessibility Act (EAA) in der EU neue Regelungen auch für Dienstleistungen wie Bankterminals oder Onlineshops und nicht mehr nur wie bisher für die Informationsangebote öffentlicher Stellen (*Der „European Accessibility Act“*, o. J.).

Aufgrund der gesellschaftlichen und gesetzlichen Relevanz wird es immer wichtiger, barrierefreie digitale Angebote zu erstellen und vorhandene Angebote auf Barrierefreiheit zu prüfen. Hierbei kann gerade bei umfangreichen Websites der Einsatz von Automatischen Monitoring Systemen (AMS) sinnvoll sein.

AMS sind Softwareprogramme, die Websites automatisch in festgelegten Abständen auf Barrieren untersuchen und den Verlauf der gefundenen Barrieren über einen Zeitraum darstellen. Viele der angebotenen Tools erklären zudem zugrunde liegende gesetzliche Regelungen oder nutzen Gamification, um die Motivation der Nutzenden zu steigern (Burkard et al., 2021).

1.2 Zielsetzung

Die Auswahl eines passenden AMS kann sich aufgrund des großen Angebots und unterschiedlichen Funktionsumfangs als herausfordernd gestalten.

Das Ziel dieser Thesis ist es daher, existierende AMS zu analysieren und zu vergleichen. Die AMS im Vergleich wählten wir aufgrund ihrer unterschiedlichen Preismodelle: Pa11y ist ein kostenfreies Open Source Tool, ARC Monitoring besitzt ein flexibles Preismodell, bei dem nach einzelnen Scans der Website abgerechnet werden kann, und Siteimprove richtet sich vor allem an Enterprise-Kund:innen.

Im Zentrum des Vergleichs steht folgende Forschungsfragen:

Welches AMS schneidet im Vergleich am besten ab?

a) Welches AMS weist die beste Abdeckung der Web Content Accessibility Guidelines vor?
b) Wie gut ist die Usability der Fehlerberichte der AMS?

Darüber hinaus wollen wir einen Einblick in die Fragen erlangen, ob und wie weit ein automatisches Tool generell einen manuellen Test ersetzen kann und welche Bereiche der WCAG abgedeckt werden.

Um die Forschungsfragen zu beantworten, vergleichen wir die AMS qualitativ und quantitativ.

Zuerst stellen wir die AMS in verschiedenen Aspekten qualitativ gegenüber. Um quantitative Ergebnisse zu erhalten, wenden wir die AMS anschließend auf eine selbst erstellte Test-Website mit konstruierten Fehlern an, die bewusst Regelungen zur digitalen Barrierefreiheit verletzen. Danach vergleichen und analysieren wir die erzielten Prüfungsergebnisse. Die Usability der AMS bewerten wir mit einer Heuristischen Evaluation anhand Nielsens 10 Heuristiken. Unser Vorgehen beschreiben wir in Kapitel 3 Methodik.

2 Theoretische Grundlagen

Bevor wir die AMS vergleichen, schaffen wir in diesem Kapitel notwendige theoretischen Grundlagen über in Deutschland geltende gesetzliche Vorschriften und die Funktionsweise von AMS, sowie ihre Vor- und Nachteile.

2.1 Gesetzliche Vorschriften in Deutschland bzgl. Barrierefreiheit

Verpflichtungen zu digitaler Barrierefreiheit sind auf verschiedenen Ebenen gesetzlich geregelt. Das ist eine bewusste Position der Gesetzgebenden für mehr Inklusion von behinderten Personen. In Deutschland sind verschiedene Gesetze und Standards maßgebend bei der Gestaltung von barrierefreien digitalen Angeboten.

2.1.1 Gesetze

Die Web Accessibility Directive (WAD) bzw. Richtlinie 2016/ 2102 des Europäischen Parlaments und Rates aus dem Jahr 2016 regelt die Anforderung an digitale Angebote wie Websites und mobile Anwendungen von öffentlichen Stellen. Deren Informationen und Dienstleistungen sind für die Allgemeinheit von grundlegender Bedeutung und sollen für alle Nutzenden besser zugänglich sein (Richtlinie (EU) 2016/ 2102 des Europäischen Parlaments und des Rates über den barrierefreien Zugang zu den Websites und mobilen Anwendungen öffentlicher Stellen, 2016).

Das Behindertengleichstellungsgesetz (BGG) setzt die WAD auf Bundesebene um. Es gilt seit 2002 in Deutschland und schützt Menschen mit Behinderungen vor Diskriminierung und soll sicherstellen, dass sie die gleichen Rechte und Möglichkeiten wie alle anderen haben. In § 12 sind Bestimmungen für barrierefreie Informationstechnik geregelt (BGG - Gesetz zur Gleichstellung von Menschen mit Behinderungen, 2002).

Aufgrund des BGG wurde 2002 die Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz (Barrierefreie Informationstechnik-Verordnung – BITV) erlassen, welche 2011 durch die BITV 2.0 abgelöst wurde (*BITV 2.0 - Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz*, o. J.). Die BITV 2.0 wurde im Mai 2019 novelliert. Seither beschreibt sie Anforderungen nicht mehr im Detail, sondern verweist auf die Europäische Norm 301 549. Die BITV 2.0 wird mit dem BITV-Test geprüft (BITV-Test, o. J.).

Der European Accessibility Act (EAA) wurde 2019 erlassen (*European Accessibility Act*, o. J.). Der EAA hat zum Ziel, bestimmte alltägliche Gegenstände und Services für

behinderte Personen zugänglicher zu machen. Grundlage ist hier die EN 301549. Betroffen sind hiervon zum Beispiel öffentliche Selbstbedienungsterminals, Messenger-Dienste und Bankdienstleistungen. Hiermit findet erstmals auch eine Regelung zur Barrierefreiheit ihren Weg in die Privatwirtschaft. Es verpflichtet nicht der EAA, sondern das jeweilige Landesgesetz die privaten Unternehmen (*Der „European Accessibility Act“*, o. J.).

In Deutschland wird der EAA in Form des Barrierefreiheitsstärkungsgesetz (BFSG) und in den Medien durch den Medienänderungsstaatsvertrag umgesetzt. Das BFSG ist ab dem 28. Juni 2025 anzuwenden und bezieht sich stark auf die Inhalte des EAA (*BFSG - Gesetz zur Umsetzung der Richtlinie (EU) 2019/882 des Europäischen Parlaments und des Rates über die Barrierefreiheitsanforderungen für Produkte und Dienstleistungen 1*, o. J.).

Um zu bestimmen, ob und welche gesetzlichen Bestimmungen zur Barrierefreiheit beachtet werden müssen, ist entscheidend, ob es sich um ein Informationsangebot einer öffentlichen Stelle handelt. Falls ja, ist in Deutschland die BITV 2.0 maßgeblich. Wenn nicht, ist genau zu prüfen, ob es sich um ein Angebot handelt, für das ab 2025 das BFSG gültig ist. Dies ist von verschiedenen Faktoren wie zum Beispiel der Größe des Unternehmens abhängig.

2.1.2 Standards

Die Web Content Accessibility Guidelines (WCAG) sind ein Standard, der vom World Wide Web Consortium (W3C) herausgegeben wurde, einem Gremium für die Standardisierung von Techniken im World Wide Web. Zum Zeitpunkt dieser Bachelorthesis ist WCAG 2.1 die neueste gültige Version, wobei auch Version 2.0 noch verwendet wird und gültig ist. Es gibt nur eine nicht vom W3C autorisierte Übersetzung in deutscher Sprache, im Zweifelsfall ist daher die englische Version maßgebend (*Richtlinien für barrierefreie Webinhalte WCAG 2.1*, o. J.).

Die WCAG 2.1 umfassen unterschiedliche Schichten. Die gröbste Einteilung stellen die vier Principles dar: Perceivable, Operable, Understandable und Robust. Die Principles enthalten wiederum insgesamt 13 Guidelines, welche grundlegende Ziele beschreiben, wenn Inhalte barrierefrei gestaltet werden sollen. Die Guidelines enthalten insgesamt 74 Success Criteria. Um auf die Bedürfnisse unterschiedlicher Zielgruppen und Situationen einzugehen gibt es drei Konformitätsstufen: A (die Niedrigste), AA und AAA (die Höchste).

Die Success Criteria werden durch hinreichende und beratende Techniques ergänzt, welche einen informativen Charakter haben. Die beratenden Techniken gehen teilweise

über die Anforderungen hinaus, die zur bloßen Einhaltung der Success Criteria notwendig ist. Teil der Techniken sind auch die Failures, also häufig gemachte Fehler (*Web Content Accessibility Guidelines (WCAG) 2.1*, o. J.).

Seit dem 20.07.2023 sind die WCAG 2.2 eine Proposed Recommendation des W3C. Dies bedeutet, dass sie als qualitativ ausreichend eingeschätzt werden, um zukünftig ein Standard zu werden und die WCAG 2.1 als neueste Empfehlung des W3C abzulösen (*Web Content Accessibility Guidelines (WCAG) 2.2*, o. J.). Da die Versionen der WCAG 2 im Vergleich zur vorherigen lediglich eine Erweiterung darstellen, herrscht eine Abwärtskompatibilität.

Die bereits im Zusammenhang mit der BITV 2.0 und dem EAA erwähnte Europäische Norm 301 549 verweist in Kapitel 9 auf die Anforderungen der WCAG und enthält noch darüberhinausgehende Anforderungen für zum Beispiel Hardware, Software und Dokumente (*DIN EN 301549 Barrierefreiheitsanforderungen für IKT-Produkte und -Dienstleistungen*, o. J.).

2.2 Automatische Monitoring-Systeme

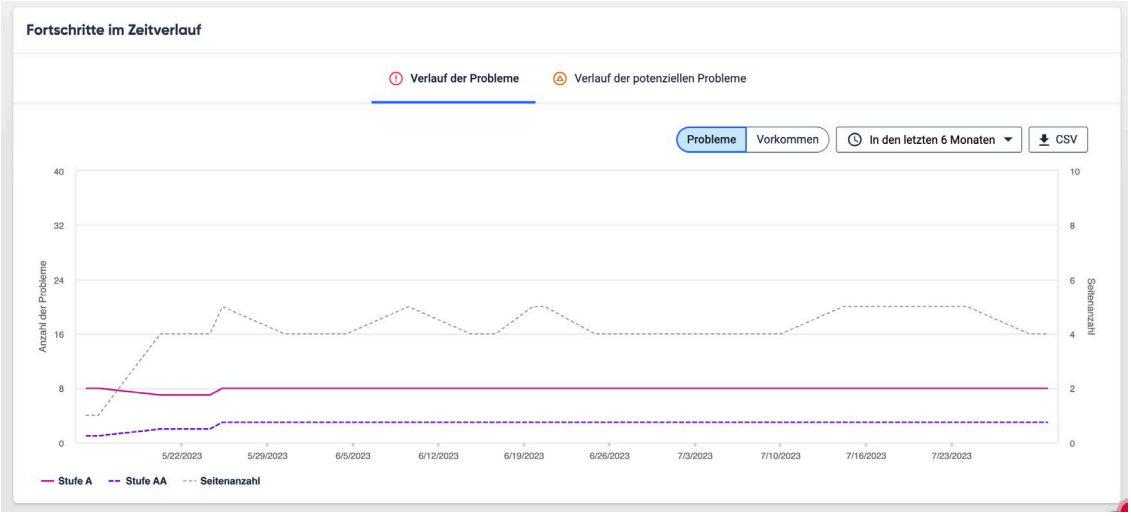
Nach einer Übersicht des W3C gibt es folgende Formen von Prüftools, die zur Evaluation der digitale Barrierefreiheit verwendet werden können: Erweiterungen für Autorentools und Webbrowser, Kommandozeilen-Tools, Desktop-Anwendungen, mobile Anwendungen und Online-Dienste (Initiative (WAI), o. J.).

Hierbei können zwei Arten unterschieden werden: generelle Tools, die umfassend nach Standards oder Regelungen prüfen und Tools, die auf spezielle Themen wie zum Beispiel Farbkontraste zugeschnitten sind (Al-Khalifa, 2012).

Gegenstand dieser Bachelorthesis sind automatische Monitoring Systeme für Barrierefreiheit (AMS). Sie prüfen den HTML-, CSS- und sonstigen Code einer Website mit verschiedenen Algorithmen und Regeln auf Probleme mit der Barrierefreiheit. Ein Monitoring System funktioniert ähnlich zu einem herkömmlichen Prüftool, unterscheidet sich jedoch dadurch, dass die Prüfungen automatisch und regelmäßig vom System vorgenommen werden können. Vorherige Scan-Ergebnisse werden archiviert, sodass der Verlauf der gefundenen Fehler über die Scans hinweg erhalten bleibt. Diese Entwicklung wird von den Systemen häufig durch Graphen visualisiert (siehe Abbildung 1).

Abbildung 1

Zeitlicher Verlauf der gefundenen Probleme im AMS Siteimprove (eigener Screenshot)



2.2.1 Vorteile

Nach Vigo & Brajnik (2011) ermöglicht die Automatisierung der Messung der Barrierefreiheit eine schnelle und unkomplizierte Methode, um Werte für den Stand der Barrierefreiheit einer Website zu ermitteln. Da der Bewertungsprozess ohne menschliches Zutun abläuft, ermöglicht er zudem kostengünstig die Analyse einer großen Anzahl von Webseiten und die Erstellung von Echtzeit-Bewertungen der Barrierefreiheit.

Eine Bewertung der Barrierefreiheit einer Website findet meist summativ am Ende des Entwicklungsprozesses oder für bereits veröffentlichte Websites statt (Vigo et al., 2013). Burkard et al. (2021) suggerieren jedoch, dass es einfacher und kostengünstiger ist, gleich von Beginn auf Barrierefreiheit zu achten, als diese rückwirkend herzustellen. Gerade wenn die Barrierefreiheitsprüfung in den Softwareentwicklungs- und Qualitätssicherungszyklus integriert werden soll, wird von der Kostengünstigkeit und der Schnelligkeit der AMS profitiert.

Nach Burkard et al. (2021) ist ein weiterer Vorteil speziell von AMS, dass sie Scans automatisch immer wieder ablaufen lassen, alte Scanergebnisse sichern und den Verlauf des festgestellten Barrierefreiheitszustands der Website über die letzten Scans hinweg darstellen. So können Nutzende schnell erkennen, ob hinzugekommene Websiteinhalte neue Fehler bei der Barrierefreiheit erzeugen.

Durch den Einsatz von AMS ist der Bewertungsprozess aufgrund seines automatisierten Charakters verlässlich und liefert reproduzierbare Ergebnisse (Vigo & Brajnik, 2011).

2.2.2 Grenzen

Vigo et al. warnen in ihrer Studie davor, alleinig Prüftools zu verwenden, um die Barrierefreiheit einer Website zu überprüfen (Vigo et al., 2013). Um AMS richtig einzusetzen, müssen ihre Vorteile genutzt und ihre Grenzen beachtet werden.

Eine Schwachstelle von AMS sehen Parvin et al. (2021) in ihrem Mangel an Transparenz: Die potenziell variierenden und unklaren Ergebnisse verschiedener Tools können Nutzende verwirren und ihre Akzeptanz verringern.

Die Richtlinien der WCAG 2.0 bzw. 2.1 sind im Gegensatz zu 1.0 testbar und technologieneutral, um für die immer größer werdende Menge an unterschiedlichen Technologien anwendbar zu sein. Die WCAG 2.0 beschreiben keine bestimmte Strategie, um ein Problem zu adressieren, sondern das gewünschte Ergebnis und erlauben Flexibilität bei der Erfüllung. Die gesteigerte Testbarkeit ermöglichte erst die Entwicklung von AMS. Die Flexibilität kann jedoch ein Problem darstellen: AMS mangelt es an der Entscheidungskraft eines menschlichen Bewerters. Sie liefern oft keine kontextualisierte Interpretation oder Aufschluss über den Schweregrad der Auswirkungen des aufgetretenen Fehlers (Guarino Reid & Snow-Weaver, 2009; Vigo et al., 2013).

3 Methodik

3.1 Erstellung der Test-Website

3.1.1 Auswahl der Methode

Für die Prüfung eines AMS bieten sich nach Abduganiev (2017) zwei Möglichkeiten: entweder werden die AMS auf eine selbst erstellte Test-Website angewendet oder auf ein ausgewähltes Sample an bereits bestehenden Websites. Beide Ansätze haben unterschiedliche Vor- bzw. Nachteile und Ziele.

Die Auswahl eines Samples aus bereits existierenden Websites bedarf keines gesonderten Entwicklungsaufwands, was Zeit und Aufwand spart. Ein Nachteil ist, dass bei Websites, die von anderen Webautoren verwaltet werden, jederzeit Änderungen auftreten können, die nicht beeinflussbar sind. Zusätzlich besteht keine Kontrolle über die vorkommenden Fehler: Wahrscheinlich sind in bestehenden Websites vor allem generell sehr häufig vorkommende Fehler enthalten und spezifischere Fehler bleiben aus.

Unser Ziel war es jedoch, alle Success Criteria abzudecken und wenn möglich eine Aussage darüber zu treffen, welchen Bereich der WCAG 2.1 die Tools abdecken können und welchen nicht. Daher entschieden wir uns für das Vorgehen, selbst eine Website mit konstruierten Fehlern zu erstellen.

Nachteil von diesem Ansatz kann neben dem Entwicklungsaufwand sein, dass in einer künstlich geschaffenen Umgebung und nicht unter realen Bedingungen geprüft wird. Um eine repräsentativere Analyse zu erreichen, könnten in einem zweiten Schritt ausgewählte Samples aus bestehenden Websites geprüft und die erzielten Ergebnisse der AMS vergleichen werden. Dies übersteigt jedoch den Umfang dieser Thesis.

Das Ziel dieser Thesis ist nicht, die automatische Überprüfung der Barrierefreiheit auf Websites zu prüfen, die mit Webframeworks erstellt wurden oder Rich Internet Applications (RIAs) sind. Daher greifen wir für die Erstellung der Test-Website auf herkömmlichen Internettechnologien und Sprachen zurück: HTML, CSS und Javascript.

3.1.2 Zusammenstellung des Fehlerkatalogs

Eine Anforderung an den in der Test-Website abgedeckten Fehlerkatalog stellt in Deutschland die EU-Norm, da die Erfüllung dieser mitunter für einige Websites rechtlich verbindlich ist.

Tabelle A.1 der Norm (*DIN EN 301549 Barrierefreiheitsanforderungen für IKT-Produkte und -Dienstleistungen*, o. J.) listet konkret alle Anforderungen an Webcontent auf.

Insgesamt sind es 137 Elemente, die teils unter verschiedenen Bedingungen gelten. Keines der Tools im Vergleich bietet allerdings spezifisch eine Prüfung nach der EU-Norm an, sondern lediglich nach WCAG 2.0 oder 2.1 Kriterien. Es wird also nur ein Teil der EU-Norm abgedeckt, wie bereits erläutert in Abschnitt 2.1. Daher sind auch nur die WCAG 2.1 Success Criteria, speziell Konformitätslevel AA, vergleichbar und für uns bei der Erstellung der Test-Website relevant.

Im BITV-Test, der zur Überprüfung von Websites auf Barrierefreiheit genutzt wird, werden die WCAG Success Criteria 1.1.1 Text alternatives und 1.3.1 Info and Relationships noch einmal in mit Kleinbuchstaben bezeichnete Unterprüfschritte unterteilt. Diese Unterteilung haben wir für unseren Fehlerkatalog übernommen. So ergeben sich 60 abzudeckende Prüfschritte bzw. Success Criteria.

Da wir die Unterteilung der BITV-Prüfschritte verwenden, nutzen wir sowohl im Code die Bezeichnung und Nummerierung der BITV-Prüfschritte anstelle der WCAG. Die Nummern der WCAG Success Criteria lassen sich durch das Weglassen der vorangestellten 9 der BITV-Prüfschrittnummern zuordnen.

3.1.3 Gestaltung der Fehler

Anschließend ermittelten wir, wie die Fehler aussehen sollen, um die gewählten Success Criteria zu verletzen.

Teil der WCAG 2.1 sind, wie in Abschnitt 2.1.2 erwähnt, die Techniques, die eine Hilfestellung bei der Erfüllung der WCAG Success Criteria bieten sollen. Bestandteil hier sind unter anderem häufig gemachte Fehler (Failures), die den einzelnen Success Criteria zugeordnet werden und für diese zu einem Durchfallen führen würden. Die Failures werden auch in den Prüfschritten des BITV-Tests im Bereich Einordnung des Prüfschrittes referenziert.

Um eine Test-Website zu erstellen, die möglichst unkomplizierte Barrierefreiheits-Fehler zu allen Success Criteria bzw. entsprechenden Prüfschritten enthält, entschieden wir uns dafür, eine Kombination aus den W3C-Failures und selbst festgelegten Fehlern zu verwenden. Es existieren nicht für alle Richtlinien zugeordnete Failures und teilweise werden naheliegende Fehler nicht abgebildet. So zum Beispiel für das Success Criterion 1.4.3 Contrast (Minimum). Unser Vorgehen hier beschreiben wir in Abschnitt 3.1.5.

In den Fällen, in denen sich die WCAG Failures nicht eigneten, konstruierten wir die Fehler über das Umkehren der Sektion Erfüllt oder über die Beschreibung in der Sektion Nicht (voll) erfüllt in der Prüfschrittbeschreibung des BITV-Tests (BITV-Test, o. J.).

Um die Auswertung im Nachhinein zu vereinfachen, konstruierten wir pro Success Criterion einen eigenen Fehler. In Fällen, in denen ein Fehler mehrere unterschiedliche Kriterien verletzen würde, versuchten wir, möglichst unterschiedliche Fehler zu verwenden.

Im Prozess stellte sich heraus, dass wir zwei der insgesamt 60 zu verletzenden Success Criteria nicht ohne weiteres abdecken können. In Success Criterion 1.2.4 geht es darum, dass Livestreams Untertitel besitzen müssen, wenn die Tonspur wichtige Informationen enthält. Hierfür müssten wir eigens eine Livestream-Technologie implementieren, die der Anforderung entsprechenden Fehler enthält. Dieser Aufwand ist im Rahmen dieser Thesis nicht umsetzbar. Success Criterion 3.3.4 Fehlervermeidung wird unterstützt ist nur dann anwendbar, wenn die Seite Transaktionen beinhaltet, durch die die Nutzenden rechtlich bindende Verpflichtungen eingehen, wie zum Beispiel eine finanzielle Transaktion. Dies ist ebenfalls im Rahmen dieser Thesis nicht sinnvoll darstellbar.

So reduzierte sich die Anzahl der in der Test-Website einzubauenden Fehler auf 58. Unser Vorgehen bei der Fehlererstellung beschreiben wir im Folgenden anhand von zwei Fehlern exemplarisch.

Der vollständige Code der Test-Website befindet sich im Anhang

A.1 Konzeption der Test-Website

Zuordnung von BITV-Prüfschritten, WCAG Failures und Umsetzung im Code der Website

Link: https://gitlab.mi.hdm-stuttgart.de/ms553/bachelor-thesis/-/blob/main/A-1-Konzeption-Testwebsite.pdf

A.2 Code der Test-Website. Ebenfalls im Anhang befindet sich eine Auflistung aller Fehler und Zuordnung zu den BITV Prüfschritten sowie Angabe von Quellen und verwendeten W3C Failures (siehe Anhang A.1 Konzeption der Test-Website).

3.1.4 Fehlerbeispiel 1: BITV-Prüfschritt 9.1.3.1e Datentabellen richtig aufgebaut

Hierbei handelt es sich um einen Prüfschritt des BITV-Tests, der eine Untergliederung des WCAG Success Criterion „1.3.1 Info and Relationships“ darstellt (*Prüfschritt 9.1.3.1e Datentabellen richtig aufgebaut | BIK BITV-Test Ergebnisse und Methodik | BIK BITV-Test*, o. J.; *Web Content Accessibility Guidelines (WCAG) 2.1*, o. J.).

Ziel ist es, dass die Information und Beziehungen, die visuell und auditiv dargestellt werden, auch dann vorhanden sind, wenn sich die Art der Darstellung ändert (*Understanding Success Criterion 1.3.1: Info and Relationships*, o. J.). Eine Änderung der Darstellung erfolgt zum Beispiel, wenn die Website sehbehinderten Nutzenden durch einen Screenreader vorgelesen wird. In diesem Fall ist die durch CSS-Stylings vorgenommene Gestaltung nicht mehr relevant. Unterscheidet oder kennzeichnet sich die Art eines Elements nur durch das Styling, so ist es für Nutzende eines Screenreaders nicht unterscheidbar.

In der Untergliederung im BITV-Test geht es konkret um den Aufbau von Datentabellen, also Tabellen, die nicht das Attribut role=presentation besitzen.

Es werden fünf Failures referenziert:

- F33: Failure of Success Criterion 1.3.1 and 1.3.2 due to using white space characters to create multiple columns in plain text content (*F33: Failure of Success Criterion 1.3.1 and 1.3.2 due to using white space characters to create multiple columns in plain text content*, o. J.)
- F34: Failure of Success Criterion 1.3.1 and 1.3.2 due to using white space characters to format tables in plain text content (*F34: Failure of Success Criterion 1.3.1 and 1.3.2 due to using white space characters to format tables in plain text content*, o. J.)
- F48: Failure of Success Criterion 1.3.1 due to using the pre element to markup tabular information (*F48: Failure of Success Criterion 1.3.1 due to using the pre element to markup tabular information*, o. J.)
- F91: Failure of Success Criterion 1.3.1 for not correctly marking up table headers (*F91: Failure of Success Criterion 1.3.1 for not correctly marking up table headers*, o. J.)
- F92: Failure of Success Criterion 1.3.1 due to the use of role presentation on content which conveys semantic information (*F92: Failure of Success Criterion 1.3.1 due to the use of role presentation on content which conveys semantic information*, o. J.)

In diesem Fall entschieden wir uns für die Verwendung von F91. F33 und F34 beziehen sich auf mehrere Prüfschritte, sind also weniger spezifisch und werden daher nicht in Betracht gezogen. F92 ordnen wir dem untergliederten Prüfschritt 9.1.3.1g Kein Strukturmarkup für Layouttabellen zu und reservieren den Failure für diesen Prüfschritt.

Die verbleibenden Failures F48 und F91 kommen beide gleichermaßen infrage. F48 beschreibt die Verwendung eines pre-HTML-Elements, um eine Tabelle zu simulieren. Da

wir bereits für den vorherigen Prüfschritt 9.1.3.1d Inhalt gegliedert ein pre-HTML-Element verwendeten, bot sich für uns die Verwendung von F91 an.

In Tabellen, die Daten enthalten und nicht zu Layoutzwecken verwendet werden, sollten für die Kopfzeilen Table Header-Elemente (th), scope Attribute oder die ARIA-Attribute role=columnheader oder role=rowheader verwendet werden. Dies ist wichtig für Personen, die Screenreader benutzen, da diese so zusätzlich zum vorgelesenen Tabellenzelleninhalt auch die zugehörigen Zellen der Kopfzeile referenzieren können, um gerade in komplexen Tabellen den Kontext beizubehalten.

Laut F91 besteht der konstruierte Fehler also aus einer Tabelle, die Daten in Form von Table Data-Elementen (td) enthält, jedoch eben keine Table Header-Elemente.

Der HTML-Code sieht daher wie folgt aus:

```
<table>
    <tr>
        <td>Header</td>
        <td>Header</td>
    </tr>
    <tr>
        <td>Inhalt</td>
        <td>Inhalt</td>
    </tr>
</table>
```

3.1.5 Fehlerbeispiel 2: Success Criterion 1.4.3 Kontraste (Minimum)

In den WCAG Failures wird für dieses Success Criterion als häufiger Fehler angebracht, dass ein Hintergrundbild nicht genug Kontrast zum Text im Vordergrund aufweist (*F83: Failure of Success Criterion 1.4.3 and 1.4.6 due to using background images that do not provide sufficient contrast with foreground text (or images of text)*, o. J.).

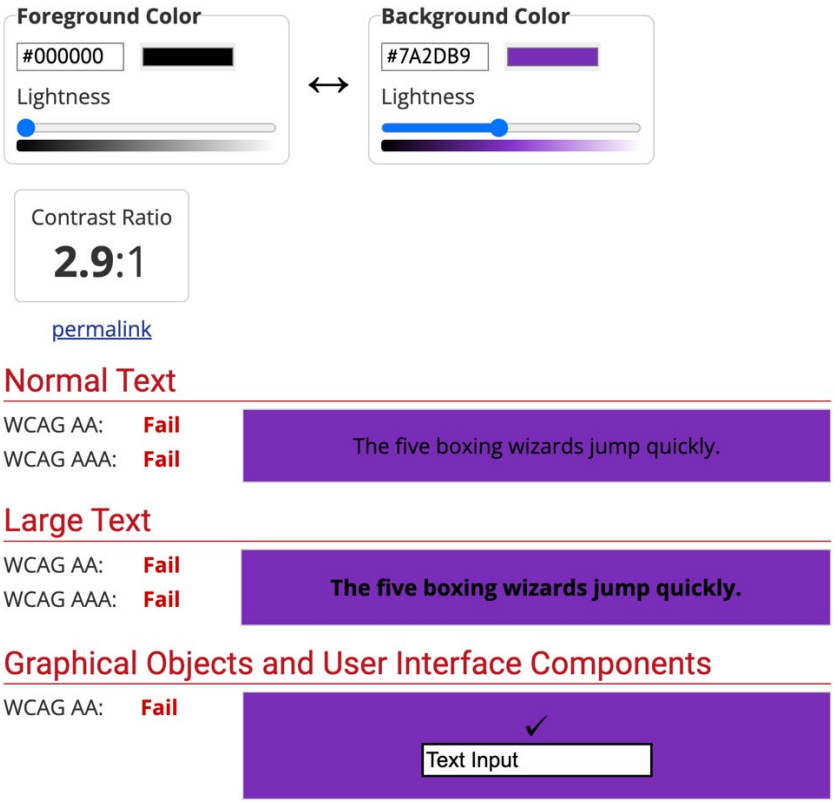
Im anderen zugehörigen Fehler wird beschrieben, dass immer sowohl die Vordergrundfarbe (CSS-Attribut color) als auch die Hintergrundfarbe (CSS-Attribut background-color) angegeben werden sollten (*F24: Failure of Success Criterion 1.4.3, 1.4.6 and 1.4.8 due to specifying foreground colors without specifying background colors or vice versa*, o. J.).

Beides sind Fehler, die eine Verletzung des Success Criterions zur Folge haben. Allerdings war es für uns naheliegender, einen Text zu platzieren, der einen zu geringen Kontrast zu einer farbigen Hintergrundfläche aufweist.

Die Farben wählten wir mithilfe des Contrast Checkers von WebAIM aus (siehe Abbildung 2). Für den Texthintergrund wählten wir die Farbe #7A2DB9, für den Text #000000. Für WCAG 2.1 Konformitätslevel AA ist für normalen Text ein Kontrastverhältnis von 4.5:1 notwendig, für großen Text (ab 24px Größe) von 3:1. In unserem Fall liegt ein Kontrastverhältnis von 2.9:1 vor, sodass das Success Criterion verletzt wird (*WebAIM: Contrast Checker*, o. J.).

Abbildung 2

Prüfung des Kontrastverhältnisses im WebAIM Contrast Checker (eigener Screenshot)



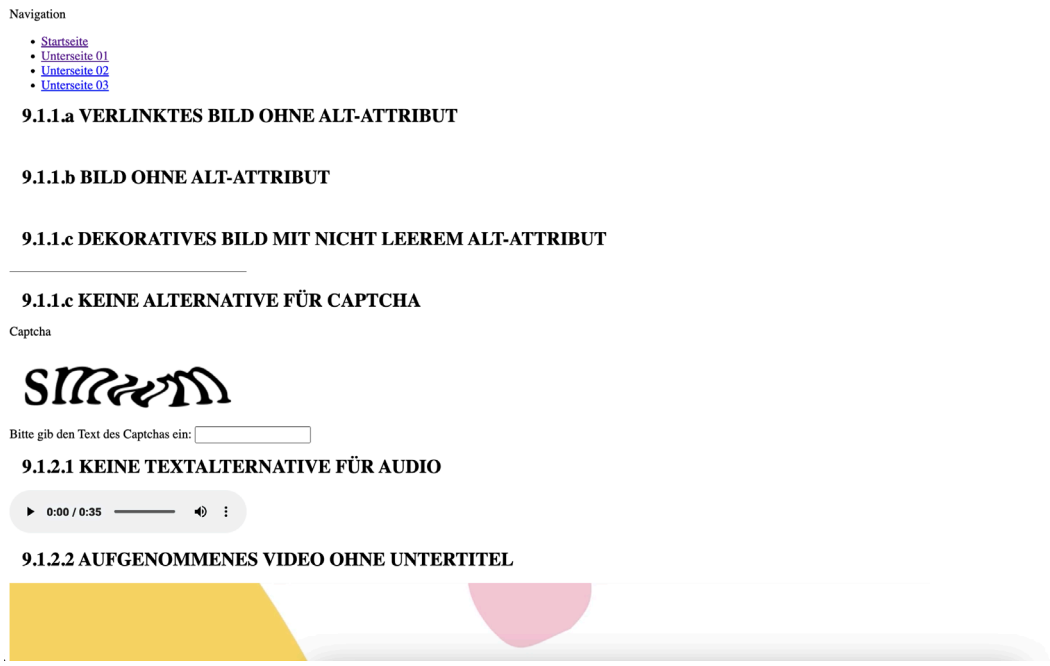
3.1.6 Durchführung der Scans

Um die AMS im Vergleich auf die Website anwenden zu können, ist eine öffentliche URL notwendig, unter der die Website für die Systeme erreichbar ist. Code, der nur lokal vorhanden ist, kann nicht geprüft werden.

Daher veröffentlichten wir die fertiggestellte Website auf einem vom Zentrum für Digitale Barrierefreiheit bereitgestellten Server unter der URL https://ba-mona.bf-lernen.de/ (siehe Abbildung 3). Anschließend konnten wir einen Scan der Website mit allen drei AMS durchführen.

Abbildung 3

Ansicht der erstellten Test-Website (eigener Screenshot)



3.2 Quantitativer Vergleich der AMS

3.2.1 Evaluationskriterien

Als Vorlage für die Wahl der Evaluationskriterien diente uns die komparative Studie von Abduganiev (2017). Von den acht definierten Qualitätskriterien verwenden wir die Kriterien Coverage und Correctness.

Um die AMS nach den Success Criteria zu beurteilen sind drei Metriken relevant:

- Richtig Positive (RP): Tatsächlich vorhandene Fehler, die vom Tool als solche erkannt wurden.
- Falsch Positive (FP): Vom Tool erkannte Fehler, die sich nach einer manuellen Überprüfung nicht als wahre Fehler herausstellen.
- Falsch Negative (FN): Tatsächlich vorhandene Fehler, die nicht vom Tool erkannt wurden.

Es kann nicht ausgeschlossen werden, das Tools auch nicht beabsichtigte Fehler finden, da einige Success Criteria nicht trennscharf unterschieden werden können und sich auf ähnliche Elemente beziehen. Für das Evaluationskriterium Coverage sind für uns nur die Meldungen relevant, die von uns beabsichtigten Fehler beinhalten. Im Evaluationskriterium Correctness betrachten wir alle Meldungen der AMS.

Die drei AMS kategorisieren die gefundenen (potenziellen) Barrierefreiheitsfehler auf unterschiedliche Arten. Die Kategorien bringen auch verschiedene Stufen an Dringlichkeit und Sicherheit mit sich: Wir gehen davon aus, dass es einen Unterschied macht, ob ein erkannter (potenzieller) Verstoß als Problem, potenzielles Problem oder Meldung angezeigt wird. Dies stellen wir dar, indem wir im Evaluationskriterium Coverage die unterschiedlichen Kategorien unterschiedlich bewerten (siehe Tabelle 1).

Tabelle 1

Vergebene Wertung für die unterschiedlichen Meldungen der Tools

Kategorie	Wertung
Problem	1 Punkt
Potenzielles Problem/ Warnung	0,5 Punkte
Hinweis	0,25 Punkte

3.2.2 Coverage der WCAG 2.1 Success Criteria

Bei Abduganiev et al. beschreibt Coverage die Abdeckung der WCAG 2.1 Succes Criteria. Die Coverage der Success Criteria wird wie folgt berechnet:

$$Coverage\ der\ \text{Success Criteria} = \frac{a}{b}$$

a: Durch das Tool gefundene verletzte Success Criteria

b: Insgesamt zu verletzende Success Criteria

Im Fall von Abduganiev et al. musste die Zahl der insgesamt zu verletzenden Success Criteria (b) erst ermittelt werden, da in der Studie bereits bestehende Websites als Samples benutzt wurden. In unserem Fall ist die Zahl jedoch bereits bekannt: Die Test-Website beinhaltet insgesamt 58 Fehler, die je einem Success Criteria entsprechen.

3.2.3 Coverage der WCAG 2.1 Principles

Für dieses Evaluationskriterium teilen wir die RP, die Success Criteria verletzen, nach den Principles der WCAG 2.1 auf.

$$Coverage\ der\ \text{Principles} = \frac{c}{d}$$

c: Durch das Tool gefundene Fehler für das jeweilige Principle

d: Insgesamt mögliche Fehler innerhalb des Principle

3.2.4 Coverage der WCAG 2.1 Guidelines

Für dieses Evaluationskriterium teilen wir die RP, die Succes Criteria verletzen, nach den Guidelines der WCAG 2.1 auf.

$$Abdeckung\ der\ \text{Guidelines} = \frac{e}{f}$$

e: Durch das Tool gefundene Fehler für die jeweilige Guideline

f: Insgesamt mögliche Fehler innerhalb der Guideline

3.2.5 Coverage der betroffenen Gruppen

In Tabelle B.2 der EU-Norm (*DIN EN 301549 Barrierefreiheitsanforderungen für IKT-Produkte und -Dienstleistungen*, o. J.) wird dargestellt, wie die Anforderungen der Norm, also die Success Criteria, mit den von Barrierefreiheitsproblemen betroffenen Benutzendengruppen zusammenhängen. Die Tabelle enthält primäre und sekundäre Zusammenhänge. Wir betrachten nur die Gruppen, die laut Tabelle B.2 primäre Zusammenhänge aufweisen. Die Kategorien Nutzung ohne Sprachvermögen (WVC) und Privatsphäre (P) beinhalten keine primären Zusammenhänge mit den Success Criteria, daher sind sie für uns nicht weiter relevant.

In Tabelle 2 stellen wir die Abkürzung für die einzelnen Gruppen analog zu den in der EU-Norm verwendeten Abkürzungen dar.

Tabelle 2

Verwendete Abkürzungen für die Aussagen zur funktionalen Leistungsfähigkeit

Abkürzung	Aussage zur funktionalen Leistungsfähigkeit
WV	Nutzung ohne Sehvermögen

LV	Nutzung mit eingeschränktem Sehvermögen
WPC	Nutzung ohne Farbwahrnehmung
WH	Nutzung ohne Hörvermögen
LH	Nutzung mit eingeschränktem Hörvermögen
WVC	Nutzung ohne Sprachvermögen
LMS	Nutzung mit eingeschränkter Handhabung oder Kraft
LR	Nutzung mit eingeschränkter Reichweite
PST	Verringerung von Anfallsauslösern bei Photosensibilität
LC	Nutzung mit kognitiven Einschränkungen
P	Privatsphäre

Quelle: DIN EN 301549 Barrierefreiheitsanforderungen für IKT-Produkte und -Dienstleistungen (o. J.)

Die Coverage der betroffenen Benutzendengruppen berechneten wir wie folgt:

$$Coverage\ der\ \text{Benutzendengruppe} = \frac{g}{h}$$

g: Durch das Tool gefundene Fehler für die jeweilige Benutzendengruppe

h: Insgesamt mögliche Fehler innerhalb der Benutzendengruppe

3.2.6 Correctness

Ziel eines Tools ist es, möglichst viele richtige Fehler zu finden. Um dies zu erreichen, könnte ein Tool viele unspezifische Fehlermeldungen ausgeben, um mit höherer Wahrscheinlichkeit hierunter auch einen echten Fehler zu enthalten. Dieses Vorgehen hat jedoch negative Auswirkungen auf die Anwendenden des Tools: Wenn die falschen Meldungen nicht durch Personen mit Expertise herausgefiltert werden, stören sie und werden als RP akzeptiert.

Nur eine manuelle Überprüfung durch Barrierefreiheitsexpert:innen kann entscheiden, ob ein durch ein AMS gefundener Fehler wirklich relevant und zutreffend ist. Das Tool mit der größten Correctness ist das, das nur wahre Barrierefreiheitsfehler findet

(Abduganiev, 2017). Aus diesen Gründen entschieden wir uns dafür, dieses Evaluationskriterium mit aufzunehmen.

Die Correctness wird wie folgt berechnet:

$$Correctness = 1 - \frac{c}{c + d}$$

c: Durch das AMS gefundene RP

d: Durch das AMS gefundene FP

3.3 Heuristische Evaluation

Ein gutes Monitoring Tool für digitale Barrierefreiheit sollte die in der Website beinhalteten Fehler finden und möglichst wenig Barrieren melden, die eigentlich keine sind. Eine schlechte Usability kann dazu führen, dass das Tool zwar alle Fehler gefunden hat, jedoch Nutzende nicht genug dabei unterstützt, die Fehler richtig zu identifizieren und zu beheben. Daher ist auch die Usability eines AMS ein entscheidender Faktor.

Um einen Einblick in die Usability der zu vergleichenden Tools zu erhalten, prüfen wir die von den AMS erstellten Fehlerberichte mithilfe einer Heuristischen Evaluation anhand Nielsens 10 Heuristiken (Nielsen & Mack, 1994, S. 30).

Eine Heuristische Evaluation stellt eine Form des Expert:innenreviews dar. Die teilnehmenden Expert:innen beurteilen den Prototypen oder das Produkt unabhängig voneinander nach einem Set ausgewählter Regeln, sogenannter Heuristiken. Hierfür analysieren die Expert:innen das Produkt und suchen nach Elementen, die gegen die Heuristiken verstoßen. Je nach Umfang des Produkts sollte sich die Evaluation nur auf einen Teilaspekt fokussieren. Nach der Durchführung der Evaluationen werden die Ergebnisse in einem Report festgehalten (Nielsen & Mack, 1994, S. 26).

Die Methode verfolgt den Gedanken, dass so die Usability eines Systems schnell und unkompliziert von Usability-Expert:innen analysiert und Verbesserungspotenzial aufgezeigt werden kann. Ein weiterer Vorteil ist, dass die Methode früh im Entwicklungsprozess eingesetzt werden kann, da das System nicht unbedingt lauffähig sein muss, es genügt ein Prototyp. Um einen angemessen hohen Anteil der Usability Probleme aufzudecken, sind drei bis fünf teilnehmende Expert:innen notwendig (Nielsen & Mack, 1994, S. 33). Nach Nielsen & Mack (1994, S.26) finden unterschiedliche Personen auch unterschiedliche Usability Probleme. Je mehr Personen evaluieren, desto größer ist die

Chance, alle vorhandenen Probleme zu finden. Eine heuristische Evaluation kann mögliche grundlegende Usability Probleme aufdecken, sie ersetzt jedoch nicht die Untersuchung mit echten Endnutzenden (Nielsen & Mack, 1994, S. 45).

Im Report können durch die Vergabe eines Severity Ratings die gefundenen Usability Probleme priorisiert werden. Die Probleme mit dem höchsten Rating werden als am gravierendsten eingeschätzt und sollten am schnellsten beseitigt werden. Nach Nielsen & Mack (1994, S. 49) sind hierfür jedoch mehrere Evaluatoren notwendig, da das von nur einer Person vergebene Severity Rating nicht zuverlässig genug ist. Da die Heuristische Evaluation in dieser Bachelorthesis von nur einer Person vorgenommen wird, verzichten wir auf die Angabe eines Severity Ratings.

4 Vorstellung der Tools

4.1 Pa11y

Pa11y ist ein Open Source Projekt, das 2013 veröffentlicht wurde (*Pa11y on Npm*, 2022). Das Projekt umfasst mehrere Tools: Ein Command Line Interface, das Dashboard, den Webservice und ein Test Runner für Continuous Integration (CI) Umgebungen. In Punkt 5.9 erklären wir das Prinzip und die Vorteile von CI. Für den Vergleich ist vor allem das Pa11y Dashboard relevant, da sich seine Funktionsweise am ehesten mit den anderen AMS überschneidet (*Pa11y*, o. J.).

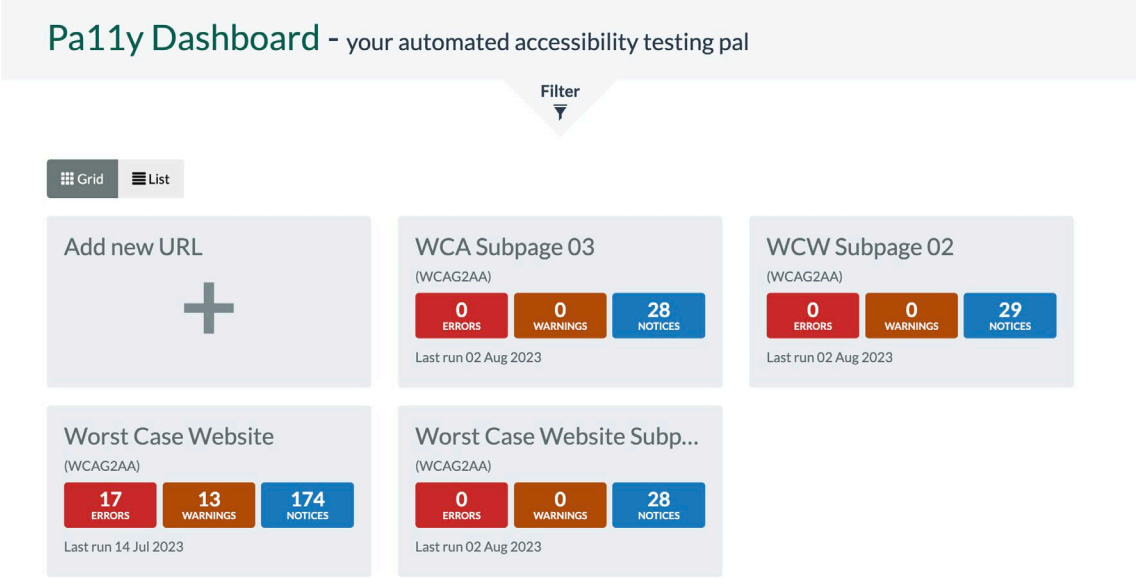
Pa11y baut auf einem anderen Open Source Projekt namens HTML Code Sniffer auf einem client-seitigen Javascript-Skript, das den Code einer Website nach einem festgelegten Standard überprüft (*HTML CodeSniffer*, o. J.).

4.2 Aufbau des Pa11y Dashboards

Nutzende können auf dem Pa11y Dashboard einzelne URLs hinzuzufügen, die nach den WCAG 2.0 Standards A, AA oder AAA analysiert werden (siehe Abbildung 4).

Abbildung 4

Startseite des Pa11y Dashboards (eigener Screenshot)

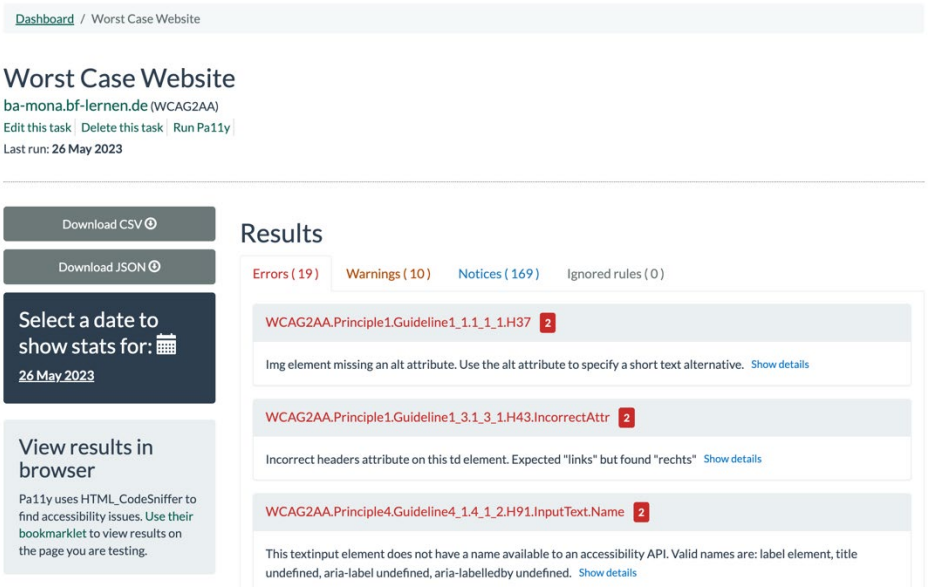


Es ist auch möglich, Interaktionen wie Button-Klicks oder einen Login imitieren. Bei den angelegten Seiten wird automatisch einmal pro Tag ein Scan durchgeführt, dieser kann aber auch manuell auf dem Dashboard angestoßen werden.

Auf der Übersichtsseite für die jeweilige angelegte Website werden die Ergebnisse des letzten Scans angezeigt (siehe Abbildung 5).

Abbildung 5

Übersicht einer angelegten Website in Pa11y (eigener Screenshot)



Das Scanergebnis von Pa11y wird in drei Kategorien aufgeteilt:

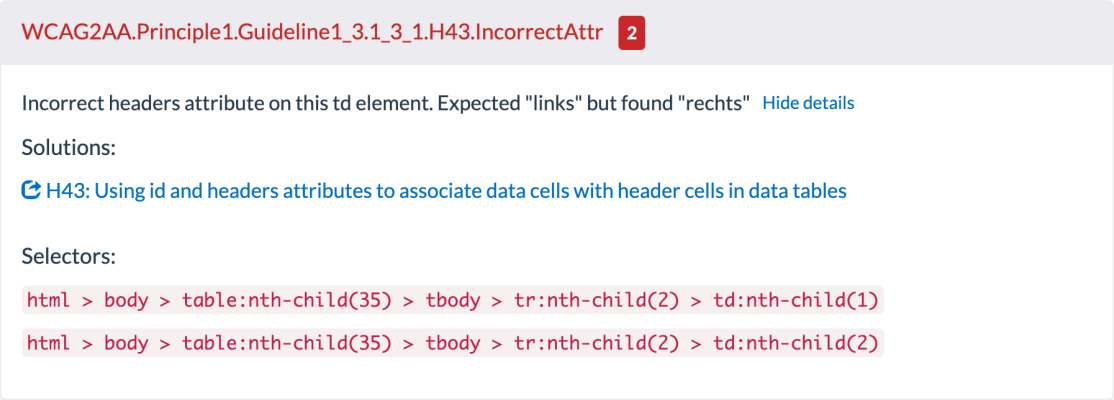
- Errors: müssen nicht manuell überprüft werden
- Warnings: Elemente, die als potenzielle Probleme automatisch gefunden werden, jedoch manuell überprüft werden müssen
- Notices: Elemente, die nicht automatisch gefunden werden können und manuell überprüft werden müssen, um den Standard zu erfüllen

Das Dashboard erläutert die Bedeutung der Kategorien nicht. Die Dokumentation des Tools HTML Codesniffer, auf dem Pa11y basiert, bietet hier mehr Information (*WCAG 2.1 Standard: Summary - HTML CodeSniffer*, o. J.).

Abbildung 6 zeigt die Detailbeschreibung eines einzelnen Fehlers. Neben der Häufigkeit des Fehlers gibt es eine kurze Beschreibung des Fehlers, einen Link zur korrespondierenden Technique aus WCAG 2.0 und eine Darstellung der HTML-Selektoren, die den Fehler verursachen.

Abbildung 6

Ansicht der Fehlerbeschreibung in Pa11y (eigener Screenshot)



4.3 ARC Monitoring

ARC Monitoring ist ein sogenanntes „Software as a Service“ Produkt (SaaS) des Unternehmens TPGi. Auf die Funktionsweise von SaaS-Produkten gehen wir in Punkt 5.2 Hosting ein. Neben Dienstleistungen wie der Durchführung von Barrierefreiheits-Audits oder Trainings bietet TPGi die ARC Plattform an (*About TPGi - The Accessibility Experts*, o. J.). ARC steht für Accessibility Ressource Center.

Die ARC Plattform teilt sich in drei Bestandteile auf: Analytics (die Überwachung der Barrierefreiheit von Websites), Content (E-Learning-Modulen Barrierefreiheits-Wiki) und die Pro Services (zum Beispiel eine Einführung in die Benutzung der Plattform und ein ausgebautes Service-Angebot).

4.4 Aufbau von ARC Monitoring

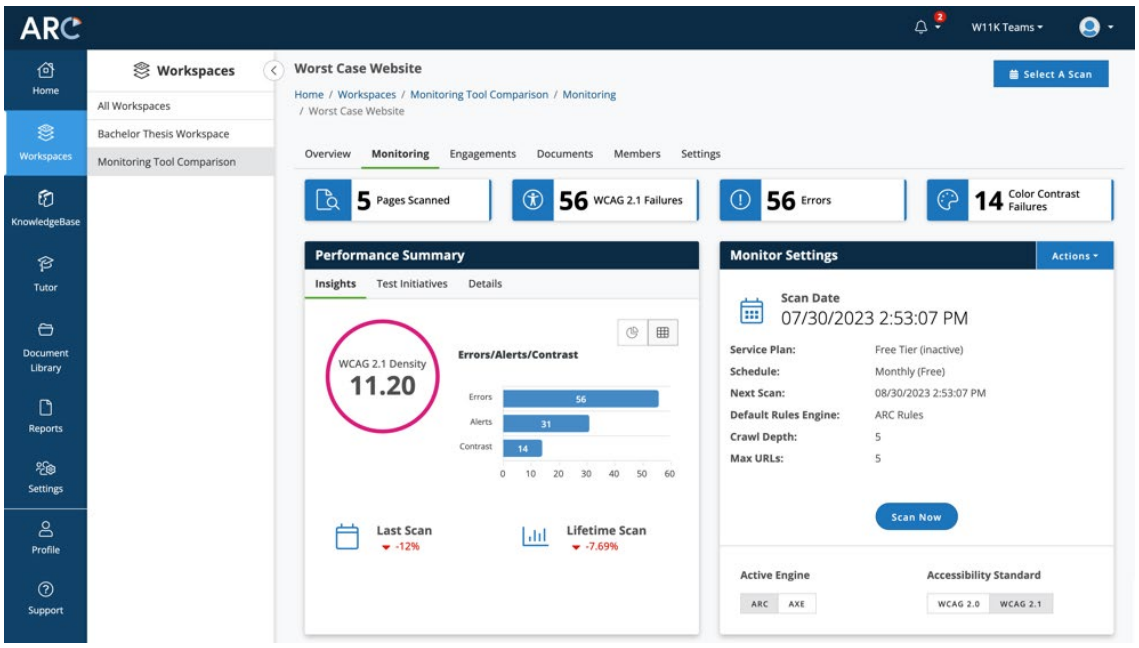
In ARC Monitoring sind die angelegten Scans in Workspaces organisiert. Ein Workspace kann ein oder mehrere Monitoring Dashboards umfassen. Ein Dashboard kann entweder einer gesamten Domain oder einem User Flow zugehörig sein. Mit User Flows können Teile einer Website zusammenhängend abgebildet werden, zum Beispiel ein Registrierungsprozess.

Die Monitoring Dashboards stellen gesammelt die Scanergebnisse aller Components, in unserem Fall alle Unterseiten, innerhalb der Domain dar. Um eine Auflistung aller Fehler einer Unterseite zu erhalten, müssen Nutzende das Monitoring Dashboard der jeweiligen Component aufrufen. Der Aufbau der unterschiedlichen Dashboards ist ähnlich.

Die Performance Summary enthält eine grobe Aufstellung über die letzten Scan-Ergebnisse und den allgemeinen Fortschritt (siehe Abbildung 7).

Abbildung 7

Ansicht des Monitoring Dashboards in ARC Monitoring (eigener Screenshot)



Der Fortschritt über die einzelnen Scans hinweg wird in Form eines Graphen visualisiert. Der Graph stellt die Failure Density dar, also die durchschnittliche Anzahl der gefundenen Fehler pro gescannte Seite (siehe Abbildung 8).

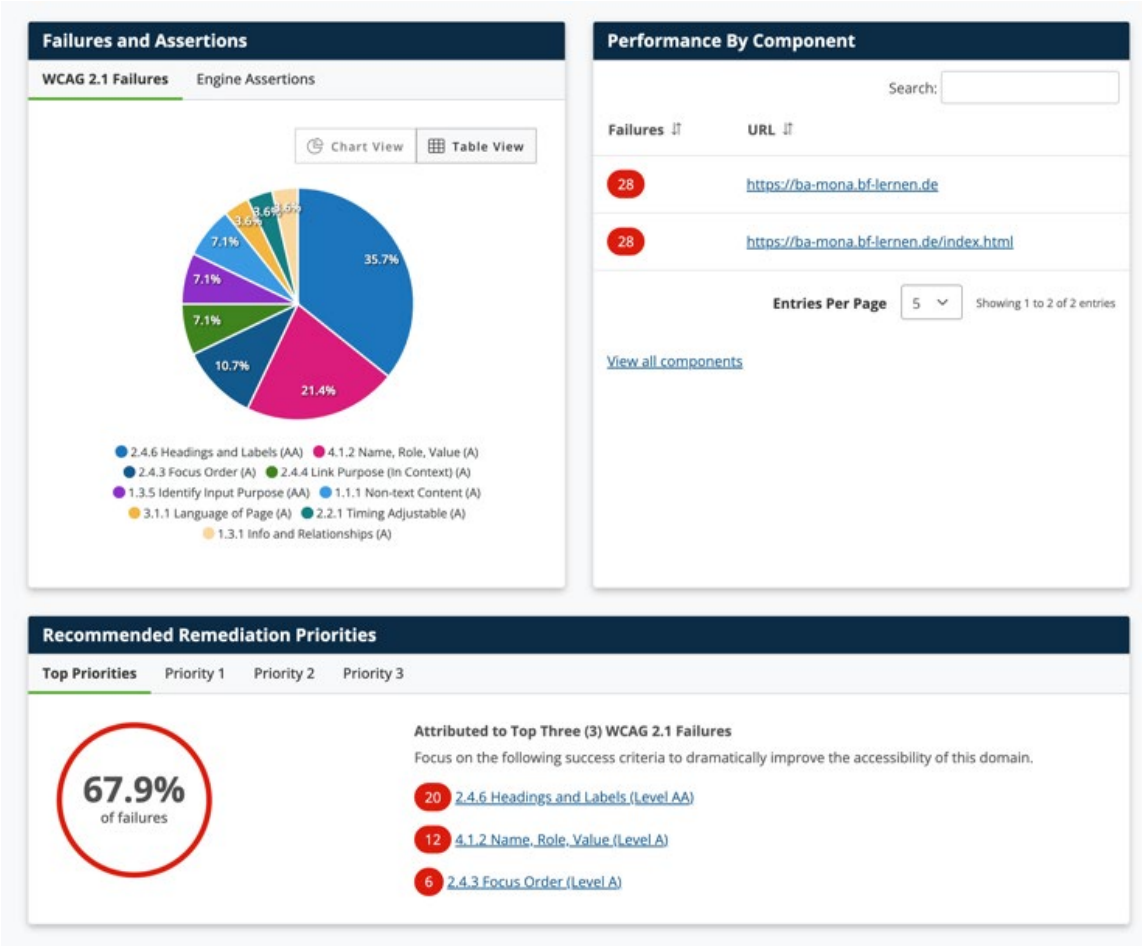
Abbildung 8

Visualisierung des Fortschritts in ARC Monitoring (eigener Screenshot)

Die gefundenen Fehler werden einmal den WCAG Success Criteria und einmal den Regeln der eigenen Engine (Engine Assertions) zugeordnet. ARC Monitoring gibt außerdem eine Empfehlung an die Nutzende ab, welche gefundenen Fehler priorisiert behoben werden sollen (siehe Abbildung 9).

Abbildung 9

Gefundene Fehler und Handlungsempfehlung in ARC Monitoring (eigener Screenshot)

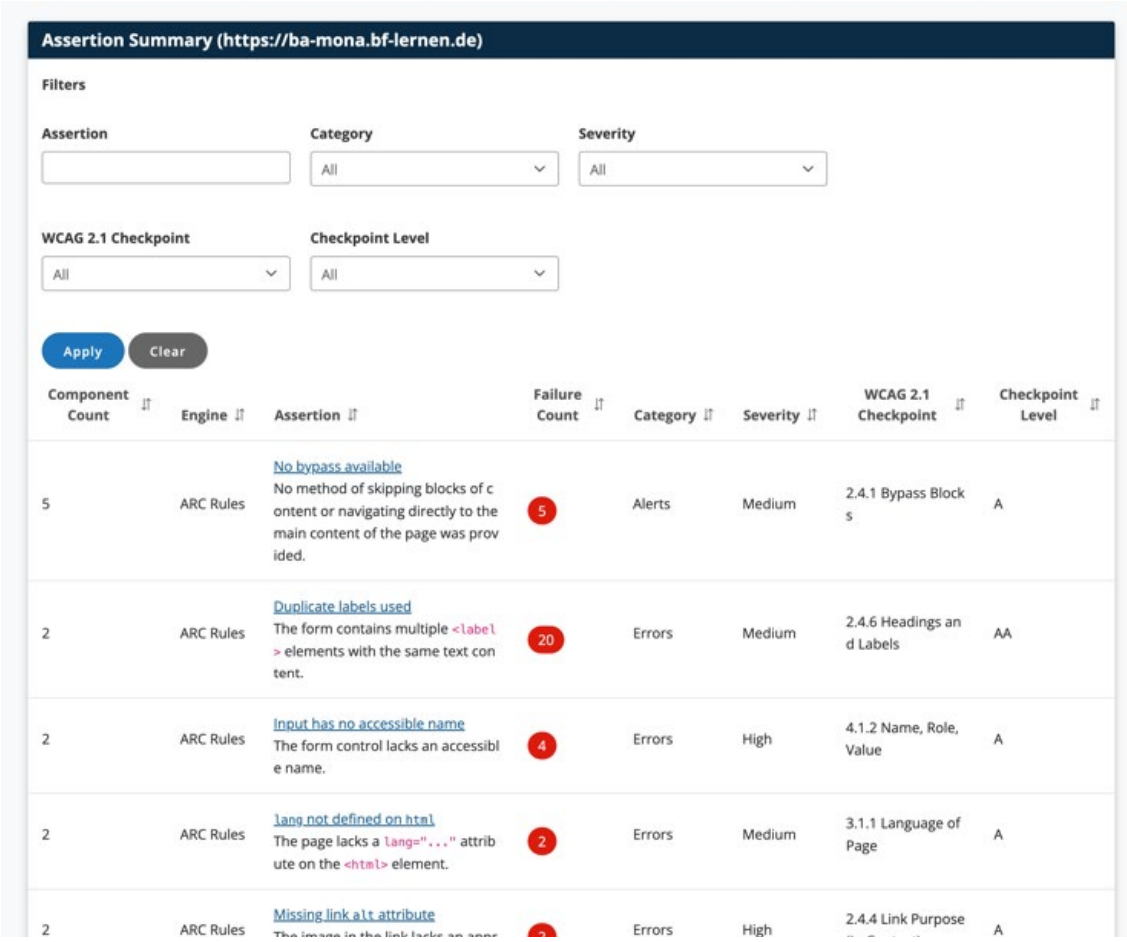


Das Scanergebnis von ARC Monitoring wird in zwei Kategorien aufgeteilt (siehe Abbildung 10):

- Errors: keine manuelle Überprüfung notwendig
- Alerts: potenzielle Probleme, die manuell überprüft werden müssen

Abbildung 10

Auflistung der Ergebnisse eines Scans in ARC Monitoring (eigener Screenshot)

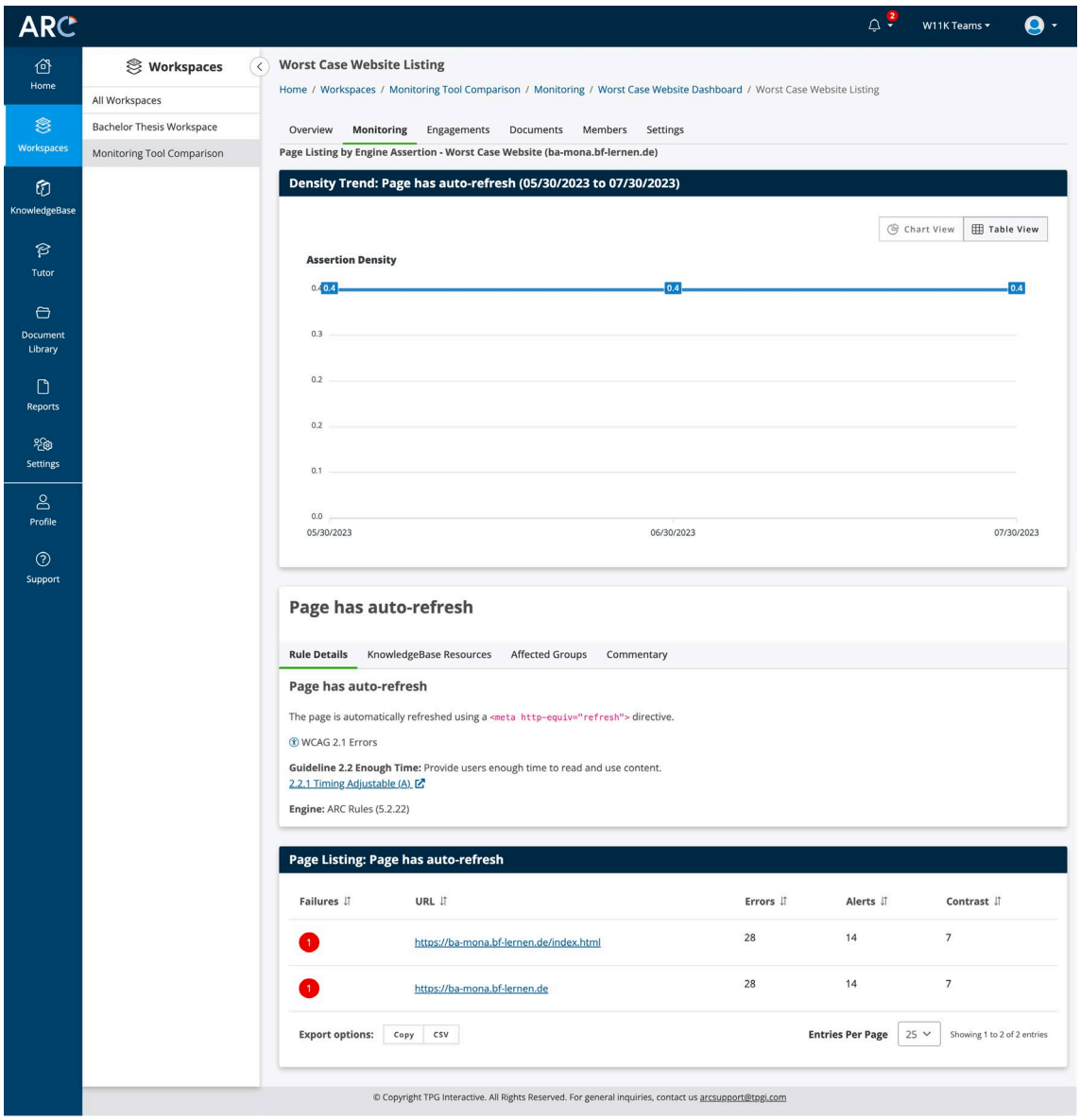


Die Detailansicht (siehe Abbildung 11) der gefundenen Probleme enthält einen Fortschrittsgraphen über die letzten erfolgten Scans sowie eine Beschreibung des Fehlers. Die Beschreibung enthält Links zur Erklärung des Success Criterion durch den W3C sowie zu den eigenen Wissensressourcen. Zusätzlich werden die von der Barriere betroffenen Nutzendengruppen beschrieben. Eine Kollaboration mehrerer Nutzender bei der Arbeit mit ARC Monitoring erfolgt in Form einer Kommentarfunktion.

In der Detailansicht des Problems innerhalb einer Component (Unterseite) sind zusätzlich die konkreten HTML-Elemente dargestellt, die den Fehler verursachen.

Abbildung 11

Ansicht eines einzelnen Fehlers in ARC Monitoring (eigener Screenshot)



4.5 Siteimprove

Siteimprove ist wie die ARC Plattform ein SaaS-Produkt und wird vom gleichnamigen Unternehmen Siteimprove angeboten. Das Produkt verspricht neben einer Überprüfung auf digitale Barrierefreiheit (Bereich Inklusivität) für Websites und darin beinhaltete PDF-Dokumente auch Lösungen für die Bereiche Content Experience (unter anderem Performance der Website) und Marketing-Performance (unter anderem Suchmaschinenoptimierung).

Um Probleme zu finden, führt Siteimprove regelmäßig Crawls der angelegten Websites durch, standardmäßig alle fünf Tage. Hierfür wird mittels HTTP-Requests der Code

gesammelt, der geprüft werden soll. Die gesammelten Daten werden in den Datenbanken von Siteimprove gespeichert (*Introduction to Siteimprove's Crawler*, o. J.).

Das Scanergebnis enthält zwei Kategorien von Meldungen:

- Probleme: keine manuelle Überprüfung notwendig
- Potenzielle Probleme: müssen manuell überprüft werden

4.6 Aufbau von Siteimprove

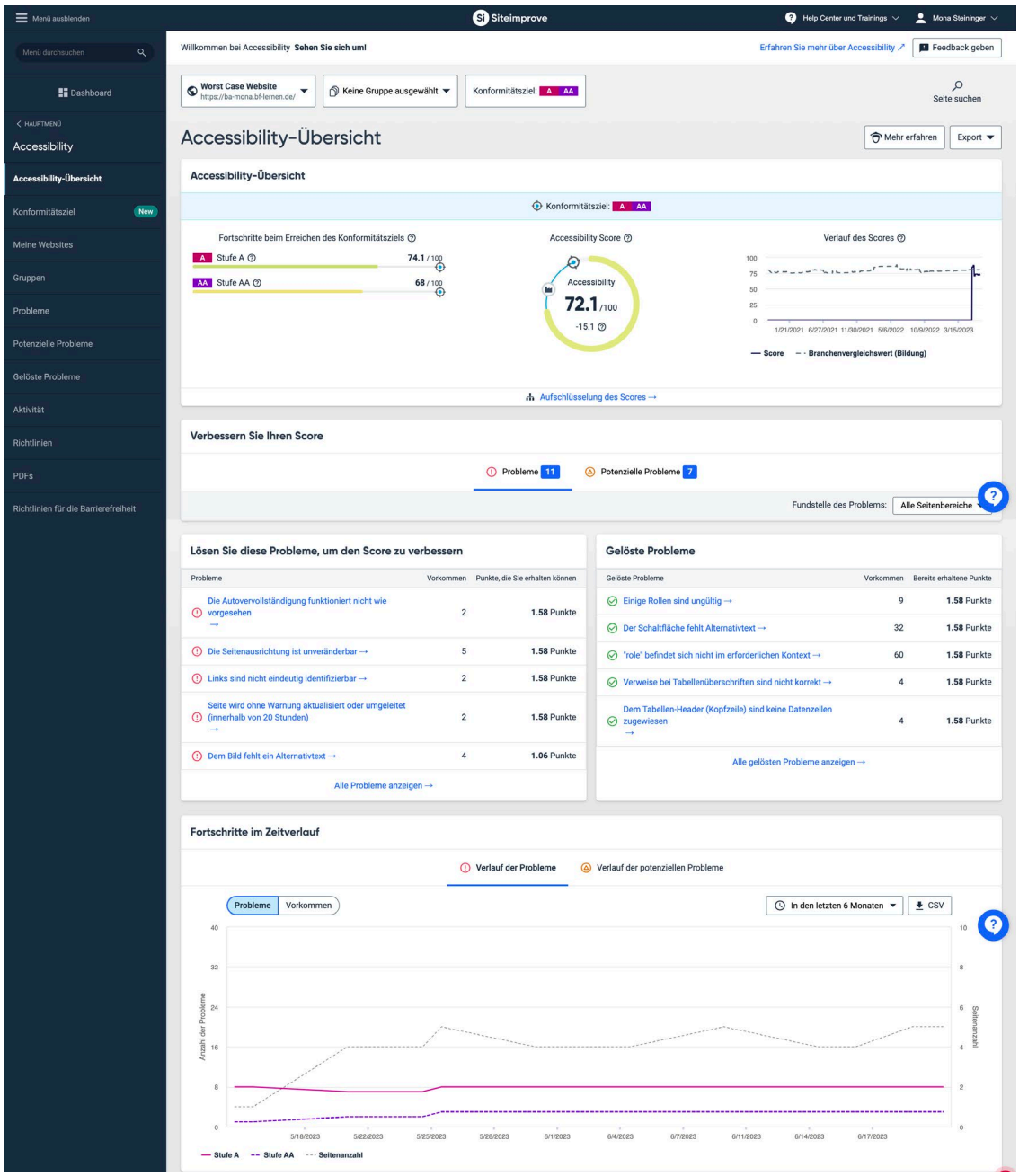
Die gefundenen Probleme werden für Nutzende auf verschiedene Arten dargestellt. Ein Dashboard der Seite gibt Übersicht über den Stand der unterschiedlichen Scores, die Siteimprove für die einzelnen Bereiche anbietet.

Der Accessibility Score wird auf der Seite Accessibility Übersicht näher für die ausgewählte Website aufgeschlüsselt. Zusätzlich ist der zeitliche Verlauf des Scores sowie eine Auswahl von gefundenen Problemen dargestellt (siehe Abbildung 12).

Der Accessibility Score ist eine Metrik von Siteimprove für den Stand der Barrierefreiheit der Website und ergibt sich aus dem Abschneiden in den Prüfungen, die das Tool auf die Website anwendet. Den Prüfungen sind unterschiedliche Punktzahlen zugeordnet, die Nutzenden durch das Lösen des Problems für den Score erhalten können.

Abbildung 12

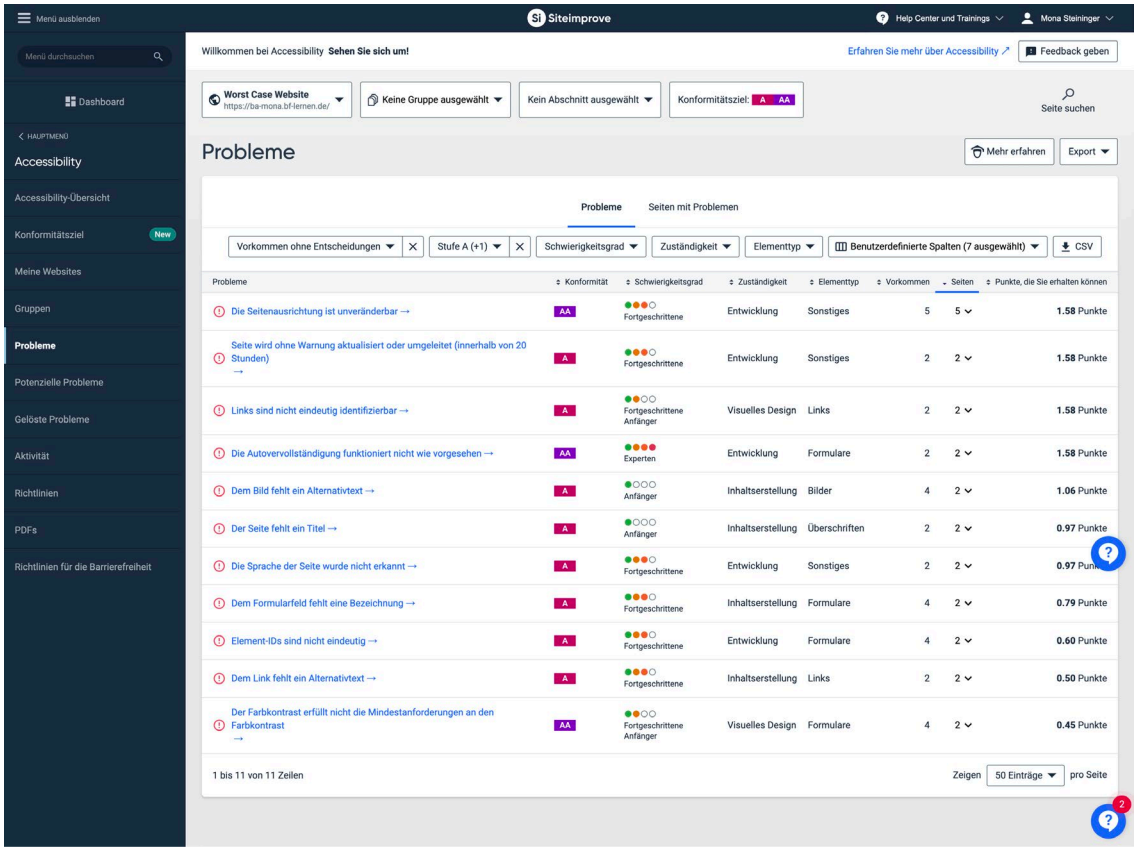
Accessibility Übersicht in Siteimprove (eigener Screenshot)



Siteimprove bietet verschiedene Übersichten für die gefundenen Probleme an: (klar identifizierte) Probleme (siehe Abbildung 13), potenzielle Probleme, bereits gelöste Probleme und eine Zuordnung der Probleme zu WCAG-Richtlinien. Diese Gesamtübersichten können über Filter, zum Beispiel nach Konformitätslevel, individualisiert werden.

Abbildung 13

Auflistung von gefundenen Barrierefreiheits-Problemen in Siteimprove (eigener Screenshot)

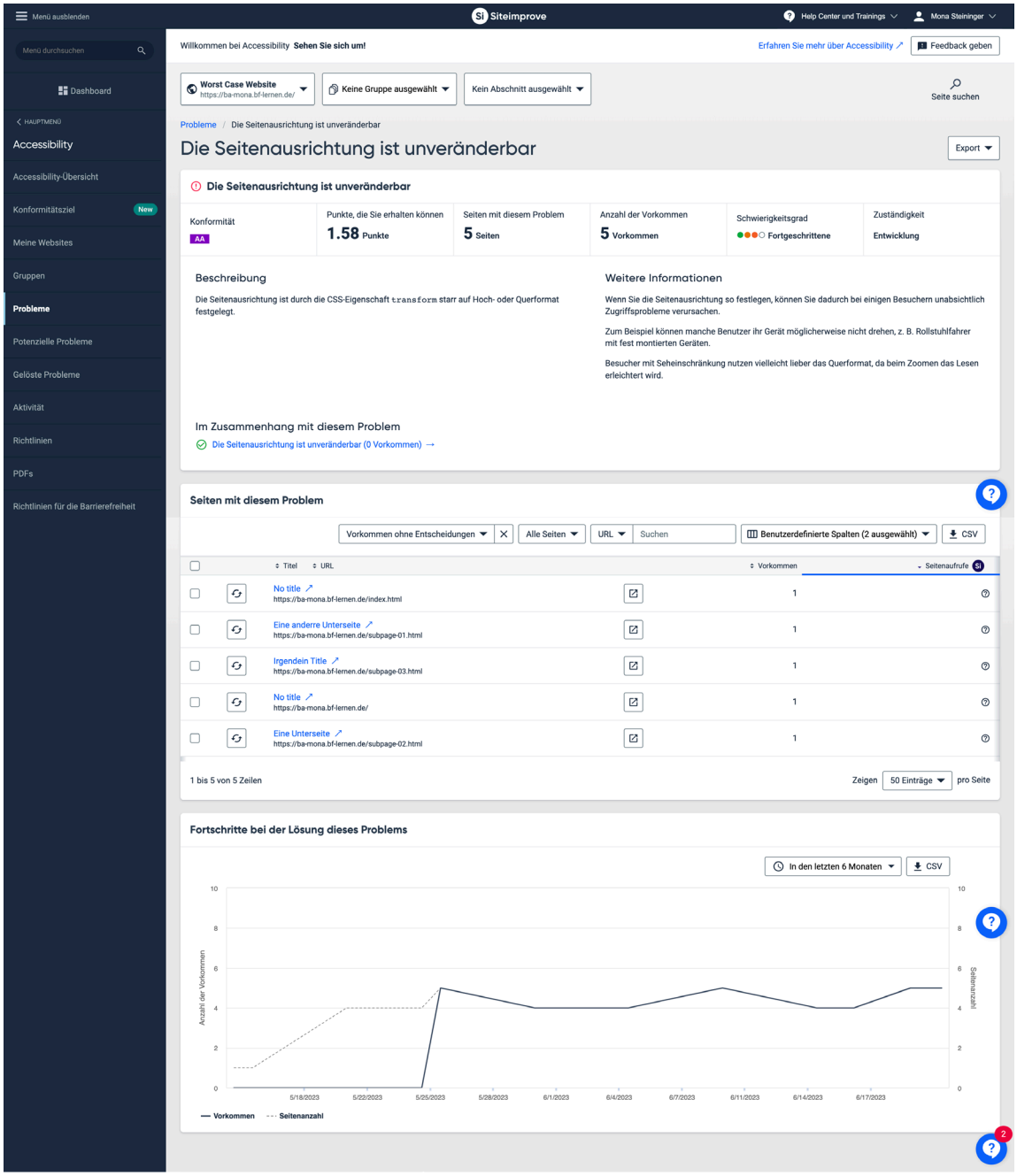


Durch Detailseiten, wie zum Beispiel in Abbildung 14 dargestellt, werden die gefundenen Probleme näher beschrieben. Diese Seiten umfassen Beschreibungen des Problems, das Konformitätslevel, die dem Problem zugeordneten Punkte im Hinblick auf den Accessibility Score, auf wie vielen Seiten das Problem vorkommt und die Häufigkeit des Problems insgesamt.

Zusätzlich wird ein Schwierigkeitsgrad zugeordnet und aufgeführt, welcher Bereich zuständig für die Lösung des Problems ist, zum Beispiel Inhaltserstellung oder Entwicklung. In einem Graphen werden die Fortschritte bei der Lösung des Problems in Form der Anzahl der Vorkommen und der Seitenanzahl dargestellt.

Abbildung 14

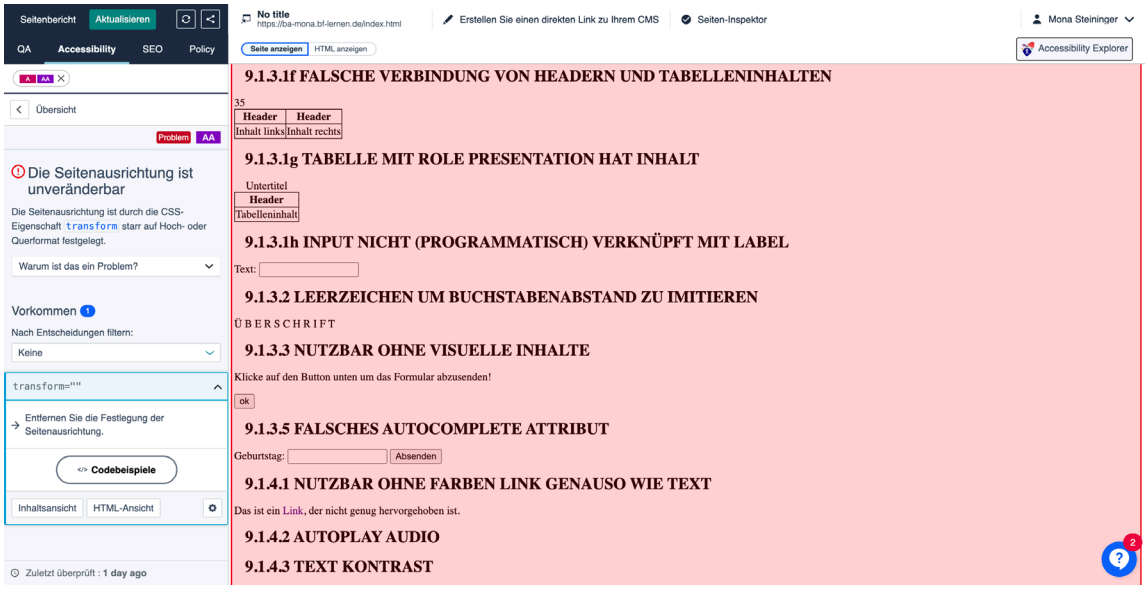
Detailseite des Problems „Die Seitenausrichtung ist unveränderbar“ in Siteimprove (eigener Screenshot)



Auf den Detailansichten ist es möglich, sich das Problem direkt im Seitenbericht der Webseite anzusehen (siehe Abbildung 15). Im Seitenbericht ist das gewählte Vorkommen des Problems farbig markiert, entweder in einer visuellen Ansicht der Webseite oder direkt im Code. Zusätzlich werden Informationen angeboten, die bei der Behebung des Fehlers helfen können, zum Beispiel weitere Erklärungen oder Code-Beispiele.

Abbildung 15

Ansicht des Problems "Die Seitenausrichtung ist unveränderbar" im Seitenbericht von Siteimprove (eigener Screenshot)



5 Qualitativer Vergleich

Tabelle 1 stellt die drei Tools in wichtigen Vergleichspunkten gegenüber. Im Anschluss werden die einzelnen Punkte näher erläutert. Soweit nicht anders angegeben entstammen die Informationen aus dem jeweiligen AMS.

Tabelle 3

Zusammenfassende Vergleichstabelle über die Tools

	Pa11y	**ARC Monitoring**	**Siteimprove**
Open Source	Ja	Nein	Nein
Hosting	Eigener Server	SaaS	SaaS
Preismodell: Kostenlose Möglichkeit	Ja	Ja	Nein
Preismodell: Kostenpflichtiges Abo-Modell	Nein	Ja	Ja
Unterstützung ganzer Websites	Nein	Ja	Ja
Sprachen	Englisch	Englisch	Englisch, Deutsch, Dänisch, Spanisch, Französisch, Italienisch, Niederländisch, Norwegisch, Finnisch, Schwedisch, Japanisch
Unterstützte Richtlinien	WCAG 2.0 A, AA, AAA	WCAG 2.0, WCAG 2.1 ADA, AODA, CVAA, Section 508	WCAG 2.1 A, AA, AAA
Unterstützung zur manuellen Prüfung	Ja	Nein	Ja

	Pa11y	ARC Monitoring	Siteimprove
Einsatz von Gamification	Fortschrittsgraph	Fortschritts-graph, Fehler-dichte, klare Ziele	Fortschritts-graph, Punktesystem, Accessibility Score, klare Ziele
Angebot ergänzender Tools	CLI-Tool, CI-Tool, Web-service-Backend, Browser Book-marklet	API für CI, Brow-ser Erweiterung	Browser Er-weiterung
Eingabeformate	HTML	HTML	HTML, XML, Skripte, PDF
Ausgabeformate	CSV, JSON	CSV	HTML, PDF, CSV

5.1 Open Source

Der gesamte Quelltext von Pa11y ist unter der Creative Commons Lizenz CC BY-NC 4.0 zugänglich und darf geteilt und veröffentlicht werden, solange eine Quellenangabe vorliegt und keine kommerzielle Nutzung stattfindet (*Creative Commons — Attribution-NonCommercial 4.0 International — CC BY-NC 4.0*, o. J.).

Auf dem öffentlichen Github-Repository ist es möglich, selbst am Projekt mitzuarbeiten. Hinter Pa11y steckt kein Unternehmen, sondern ein Team von ehrenamtlichen Entwickler:innen und Designer:innen ohne kommerzielle Motivation.

Das kann ein Problem sein: Wenn niemand bereit ist, aus intrinsischer Motivation heraus die eigene Freizeit für das Projekt zu verwenden, werden keine neuen Features entwickelt, Probleme nicht behoben und auf neu entstehende Standards nicht reagiert. Aktuell ist noch Aktivität in den Github Repositories festzustellen (Stand August 2023) und das Package verzeichnet fast 150.000 Downloads pro Woche über den Node Package Manager (npm) (*Pa11y Dashboard*, 2013/2023).

Bei ARC Monitoring und Siteimprove handelt es sich um geschlossene Systeme, deren Code nicht öffentlich einsehbar ist. Das hat den Nachteil, dass es im Gegensatz zu

Pa11y nicht möglich ist, die Systeme den eigenen Bedürfnissen entsprechend anzupassen und auszubauen. Ein Vorteil ist, dass keine Abhängigkeit von einem System besteht, das auf freiwilliger Arbeit beruht.

5.2 Hosting

Die Tools von Pa11y können auf einem lokalen Server genutzt werden. Um die Scan-Ergebnisse mehreren Nutzenden bereitzustellen ist ein eigener Server notwendig, auf dem die Anwendung sowie eine Datenbank gehosted werden.

Sowohl ARC Monitoring als auch Siteimprove sind sogenannte Software As A Service Produkte (SaaS). Das bedeutet in diesem Fall unter anderem, dass die Herstellenden die Software cloudbasiert als Webservice bereitstellen und die Nutzenden sich nicht um Themen wie Hardware, Hosting, Installation oder Updates der Software kümmern müssen.

5.3 Preismodell

Pa11y ist aufgrund des Open Source Modells kostenfrei nutzbar. Eine Bezahlversion mit ausgebauter Funktionalität ist nicht vorhanden.

Bei ARC Monitoring werden drei unterschiedliche Preisstufen angeboten, in denen die aufgeführten Bestandteile in unterschiedlichem Umfang verfügbar sind (*ARC Tier Comparison - TPGi*, o. J.): eine Free Stufe, eine Essentials Stufe und eine Enterprise Stufe.

Der kostenfreie Account von ARC Monitoring enthält einen geringen Umfang an Wissens-Artikeln und keine Möglichkeit zum kollaborativen Arbeiten mit mehreren Accounts. Pro Monat kann ein kostenloser Scan durchgeführt werden, der auf fünf Seiten limitiert ist. Wenn häufigere oder größere Scans durchgeführt werden sollen, kosten diese je 10 Cent. Die nächsthöhere Stufe „Essentials“ beinhaltet unter anderem ein vergrößertes Angebot an Wissens-Artikeln und E-Learning-Modulen sowie die Möglichkeit, kollaborativ mit mehreren Nutzenden zu arbeiten. Die teuerste Stufe „Enterprise“ kostet 59 Dollar pro Monat pro Lizenz, allerdings beträgt hier die Vertragslaufzeit mindestens ein Jahr und es ist eine Mindestanzahl nötig. Hier ist es zusätzlich zu den Inhalten der anderen beiden Stufen möglich eine Integration des Tools Zapier zu nutzen, sodass zum Beispiel für gefundene Probleme automatisch Tickets in Projektmanagement-Plattformen wie Atlassians Jira erstellt werden können.

Das Preismodell von Siteimprove wird auf der Website nicht transparent dargestellt. Die Kosten richten sich nach den Anforderungen des Unternehmens, das Siteimprove einsetzen möchte, und werden individuell kalkuliert (*Preise*, o. J.).

5.4 Unterstützung ganzer Websites

Bei Pa11y ist es bisher nicht möglich, eine gesamte Website inklusive aller Unterseiten zusammenhängend zu analysieren, alle URLs müssen einzeln hinzugefügt werden. Da der Code von Pa11y zur Verfügung steht könnte diese Funktionalität nachgerüstet werden[1]. Gerade bei umfangreichen Websites kann dies ein Nachteil sein und einen großen Arbeitsaufwand mit sich bringen.

ARC Monitoring und Siteimprove funktionieren als Crawler und können Websites zusammenhängend analysieren. Bei ARC Monitoring fällt hier jedoch negativ auf, dass zumindest in unserem Testfall die Startseite der Website zweimal gewertet wird und so die dargestellte Gesamtzahl aller gefundenen Probleme auf der gesamten Website verfälscht ist.

5.5 Angebotene Sprachen

Siteimprove ist das einzige Produkt innerhalb des Vergleichs, das nicht nur auf Englisch sondern auch auf Deutsch und in mehreren weiteren Sprachen angeboten wird.

5.6 Unterstützte Richtlinien

Pa11y bietet die Standards WCAG 2.0 A bis AAA. Es gibt auch die Möglichkeit, mit eigenen Standards die Konsistenz des Codes einer Website zu überwachen. Bei der Analyse des Fehlerberichts von Pa11y zeigte sich allerdings, dass durchaus Success Criteria von WCAG 2.1 darin enthalten sind. Grund hierfür könnte sein, dass zwar die Regeln für die Überprüfung auf die neue Version der WCAG angepasst wurden, jedoch nicht das Dashboard.

ARC Monitoring unterstützt WCAG 2.0, WCAG 2.1. Darüber hinaus werden die Standards ADA (gültig in den USA), AODA (gültig in Ontario, Kanada), CVAA (gültig in den USA) und Section 508 (gültig in den USA) angeboten. Section 508 ist ein Standard, der in den USA für Bundesbehörden gültig ist (*U.S. Access Board - Revised 508 Standards and 255 Guidelines*, o. J.).

---

[1] Zum Beispiel hier: https://www.lullabot.com/articles/monitoring-web-accessibility-compliance-with-pa11y-dashboard

Mit Siteimprove kann auf WCAG 2.1 Level A bis AAA geprüft werden. Zusätzlich bietet Siteimprove eigene Best Practices und die WAI ARIA Authoring Practices an. Die Beachtung dieser trägt zu einer allgemeinen Verbesserung der Barrierefreiheit bei, ist jedoch für eine Konformität nach WCAG nicht erforderlich.

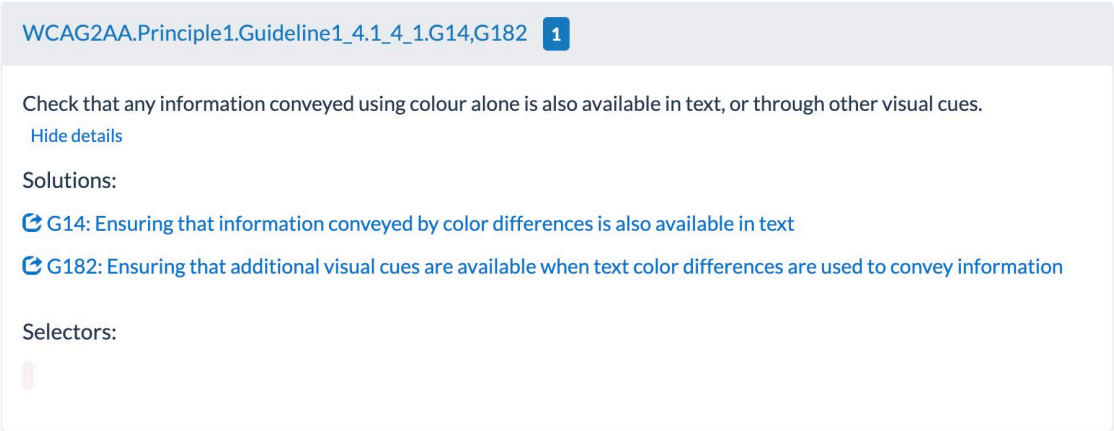
5.7 Unterstützung zur manuellen Prüfung

Ein potenzielles Einsatzgebiet eines AMS könnte auch die Unterstützung zur manuellen Prüfung der Sucess Criteria und Fehler sein, die vom AMS nicht automatisch erkannt werden.

Pa11y bietet hierfür die Meldungskategorie der Notices an, die konkrete Handlungsanweisungen für die Überprüfung bestimmter Success Criteria beinhalten (siehe Abbildung 16).

Abbildung 16

Anleitung zur manuellen Überprüfung durch Pa11y (eigener Screenshot)



ARC Monitoring stellt nur die festgestellten Barrierefreiheitsprobleme dar und gibt keine Anleitung zur manuellen Überprüfung.

Das AMS Siteimprove bietet eine Übersicht, in der die gefundenen Probleme nach der jeweils eingestellten Richtlinie, in unserem Fall nach den einzelnen WCAG Success Criteria, aufgeschlüsselt werden. Hier wird auch eine Aussage darüber getroffen, welche Success Criteria nicht automatisch von Siteimprove abgedeckt werden können und manuell geprüft werden müssen. Nähere Informationen zur Überprüfung werden allerdings nicht bereitgestellt.

5.8 Einsatz von Gamification

Die komparative Studie von Burkard et al. (2021) untersuchte unter anderem den Einsatz von Gamification in vier verschiedenen AMS. Hierfür bewerteten sie die AMS nach folgenden Gamification Pattern, die ursprünglich der Literaturrecherche von Majuri et al. (2018) entstammen:

- Punkte, Punktestand, XP-Fortschritt
- Statusbalken, Fähigkeitenbäume, Wettbewerb
- Herausforderungen, Quests, Missionen, Aufgaben, klare Ziele
- Leistungsstatistiken, Leistungsfeedback

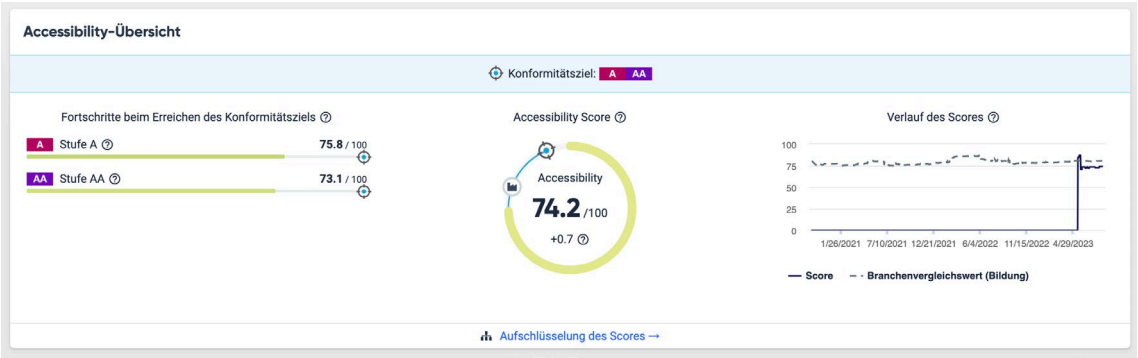
Alle drei AMS nutzen verschiedene Graphen, um den Fortschritt der festgestellten Barrierefreiheit der geprüften Website zu veranschaulichen.

ARC Monitoring spricht den Nutzenden Empfehlungen für die Barrieren aus, die priorisiert gelöst werden sollen und gibt die Fehlerdichte als Metrik für die aktuelle Barrierefreiheit an.

Siteimprove setzt mit seinem Punktesystem für die gefundenen Probleme und dem Accessibility Score (siehe Abbildung 17) Gamification Pattern am stärksten ein. In der Studie von Burkard et al. (2021) wurde Siteimprove von den teilnehmenden Nutzenden als das motivierendste AMS empfunden. Teil der Studie waren neben Siteimprove noch ARC Monitoring und andere AMS.

Abbildung 17

Accessibility Score in Siteimprove (eigener Screenshot)



5.9 Angebot ergänzender Tools

Das Projekt Pa11y umfasst neben dem Dashboard noch weitere Tools. Das Dashboard bringt die Scan-Ergebnisse als User Interface (UI) in den Webbrowser. Die weiteren

Tools haben einen technischen Fokus, ein technisches Grundverständnis ist hierfür Voraussetzung.

Das Command Line Interface (CLI) kann über den npm installiert und in der Kommandozeile oder in Javascript ausgeführt werden.

Pa11y CI ähnelt dem CLI-Tool, ist jedoch für die Verwendung in CI-Umgebungen gedacht. CI ist ein Prinzip aus der Software Entwicklung, bei dem der Code der einzelnen Entwickler:innen eines Teams möglichst oft in eine gemeinsame Codebasis, normalerweise ein Git Repository, integriert wird. Bei der Integration wird das Projekt automatisch gebaut und es laufen Softwaretests ab. So können schnell Fehler und Kompatibilitätsprobleme des neu hinzugefügten Codes gefunden werden (Singh, 2023). In diesem Rahmen kann direkt mit Pa11y auf Barrierefreiheits-Fehler im Code getestet werden, sodass diese früh im Entwicklungsprozess gefunden werden.

Das Backend des Pa11y Dashboards ist der Webservice. Diese Node App kann auch dazu verwendet werden, ein eigenes Dashboard zu bauen, die generierten Daten in bestehende Systeme einzubinden oder die Daten anders weiterzuverwenden.

Auch bei ARC Monitoring ist es möglich, die Barrierefreiheitsprüfung mittels einer API in den CI-Prozess zu integrieren. Zusätzlich wird mit dem ARC Toolkit eine Browser Erweiterung zur Prüfung einzelner Webseiten angeboten (*ARC Toolkit - Page-Level Testing*, o. J.).

Siteimprove bietet zusätzlich die Browser Erweiterung Accessibility Checker an, mit der Webseiten direkt im Browser und auch lokal geprüft werden können. Zusätzlich bietet Siteimprove eine Vielzahl an Integrationen für andere Plattformen an.

5.10 Eingabeformate

Eingabeformate sind die Dateiformate, die von den AMS geprüft werden können.

Pa11y bzw. das Pa11y Dashboard kann den HTML-Code einer Website überprüfen (*HTML CodeSniffer*, o. J.).

Wir konnten leider keine Aussage von ARC Monitoring darüber finden, welche Eingabeformate möglich sind. Angesichts der gefundenen Probleme gehen wir davon aus, dass vor allem HTML-Code und Kontrastverhältnisse geprüft werden.

Von Siteimprove analysiert werden kann HTML-Code, XML-Code, nicht geskriptete Inhalte, geskriptete Inhalte (zum Beispiel in der Sprache Javascript) und dynamisch geladene Inhalte. Zusätzlich wird die Funktion angeboten, PDF-Dateien auf Barrierefreiheit zu prüfen. Nicht analysiert werden können laut eigener Aussage Online Shops (zum

Beispiel Warenkörbe), Bezahlvorgänge und Inhalte, die eine Interaktion benötigen (zum Beispiel Ergebnisse einer Suche) (*Introduction to Siteimprove's Crawler*, o. J.).

5.11 Ausgabeformate

Ausgabeformate sind die Dateiformate, in denen die AMS Prüfberichte oder Ergebnisse exportieren können.

Im Pa11y Dashboard können Nutzende JSON- und CSV-Dateien der Scans exportieren.

Mit ARC Monitoring können CSV-Dateien ausgegeben werden.

In Siteimprove können die Ergebnisse in HTML-, PDF- und CSV-Dateien exportiert werden.

6 Auswertung

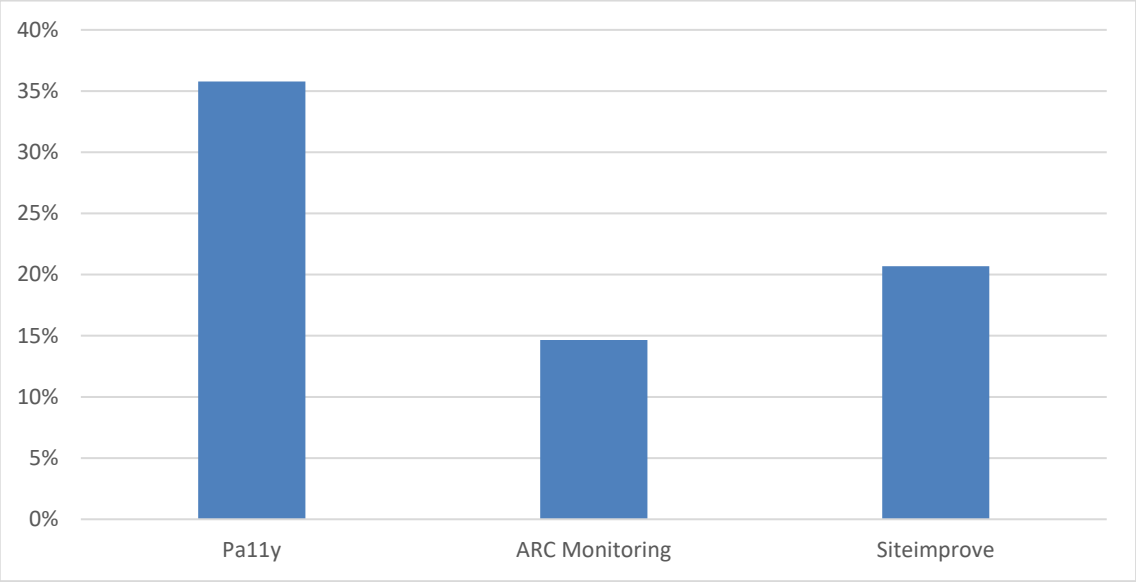
6.1 Quantitativer Vergleich der AMS

Die vollständigen erhobenen Daten, die wir zur Berechnung nutzten, können in der Excel-Tabelle im Anhang nachvollzogen werden (siehe Anhang B.1). Im Fließtext dieser Thesis stellen wir die Ergebnisse summativ dar.

6.1.1 Coverage hinsichtlich der WCAG Success Criteria

Abbildung 18

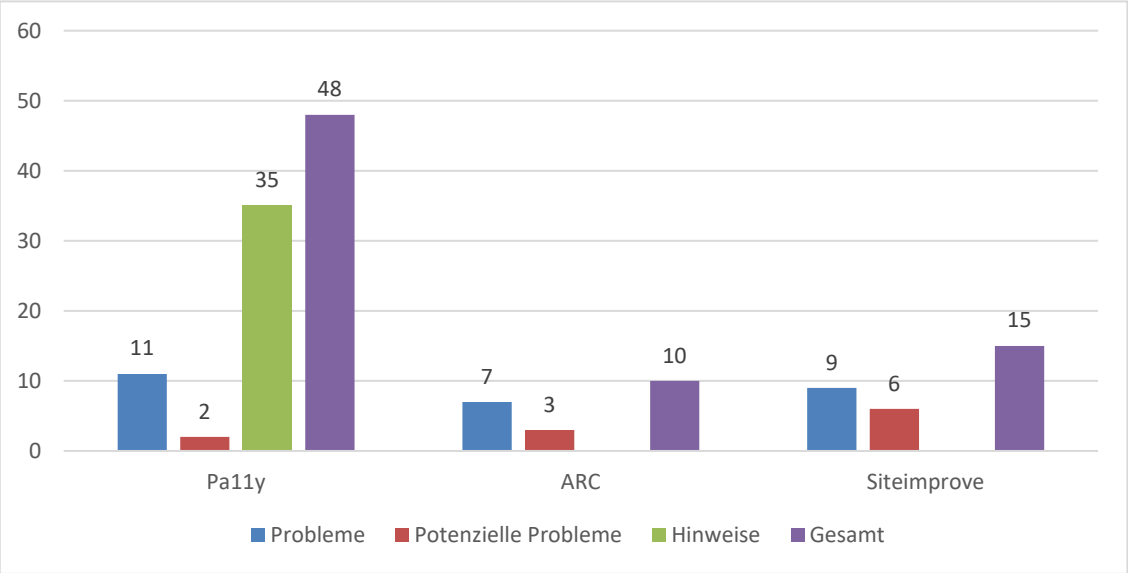
Coverage der WCAG Success Criteria



Pa11y besitzt wie in Abbildung 18 ersichtlich insgesamt die höchste Coverage der Success Criteria. Ein Grund hierfür könnte sein, dass Pa11y neben sicher gefundenen und potenziellen Problemen mit den Notices noch eine dritte Kategorie an Meldungen besitzt, die weniger spezifisch und dadurch öfter zutreffend ist.

Abbildung 19

Meldungen, die Fehler anzeigen, die Success Criteria verletzen

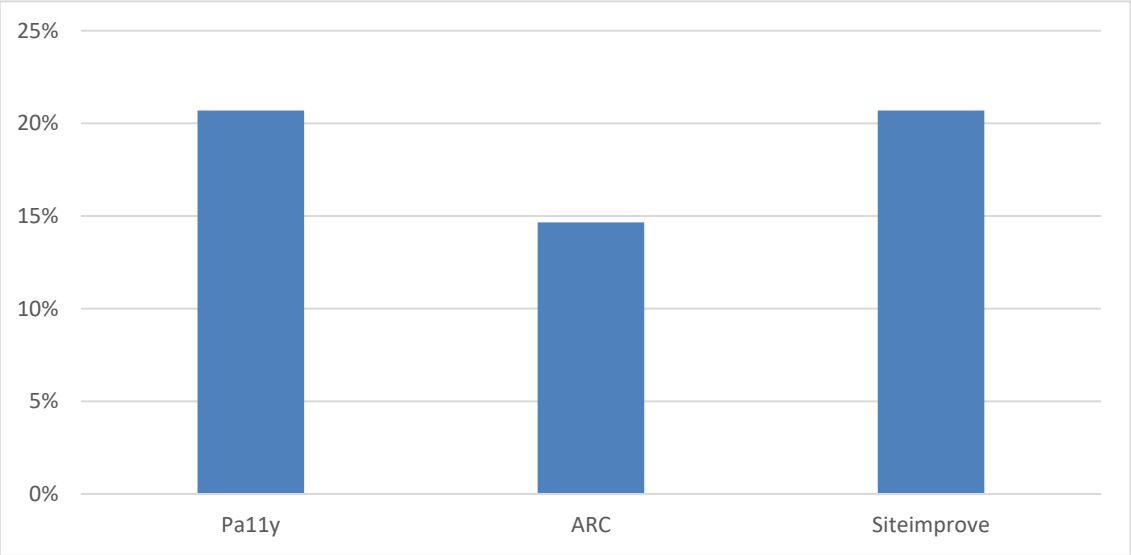


Wenn man nur die Zahl der Success Criteria betrachtet, für die die eingebauten Fehler ohne manuelle Prüfung als solche erkannt werden, so nähern sich die Ergebnisse der drei Tools an. Pa11y deckt mit 11 die meisten Success Criteria sicher ab, gefolgt von Siteimprove mit 9 und ARC Monitoring mit 7 Kriterien (siehe Abbildung 19).

Mit ein Grund für das schlechtere Abschneiden von ARC Monitoring könnte sein, dass das AMS die Gesamtzahl der gefunden Fehler in Zusammenhang mit falschen Kontrastverhältnissen zwar im Dashboard anzeigt, diese jedoch nirgends aufschlüsselt und sie somit auch nicht von Nutzenden behoben werden können.

Abbildung 20

Gesamte Coverage der Succes Criteria ohne Einbezug der Notices von Pa11y



Um den Einfluss der Notices von Pa11y zu beobachten, errechneten wir die insgesamte Coverage aller Success Criteria erneut, nur ohne die Notices mit einzubeziehen.

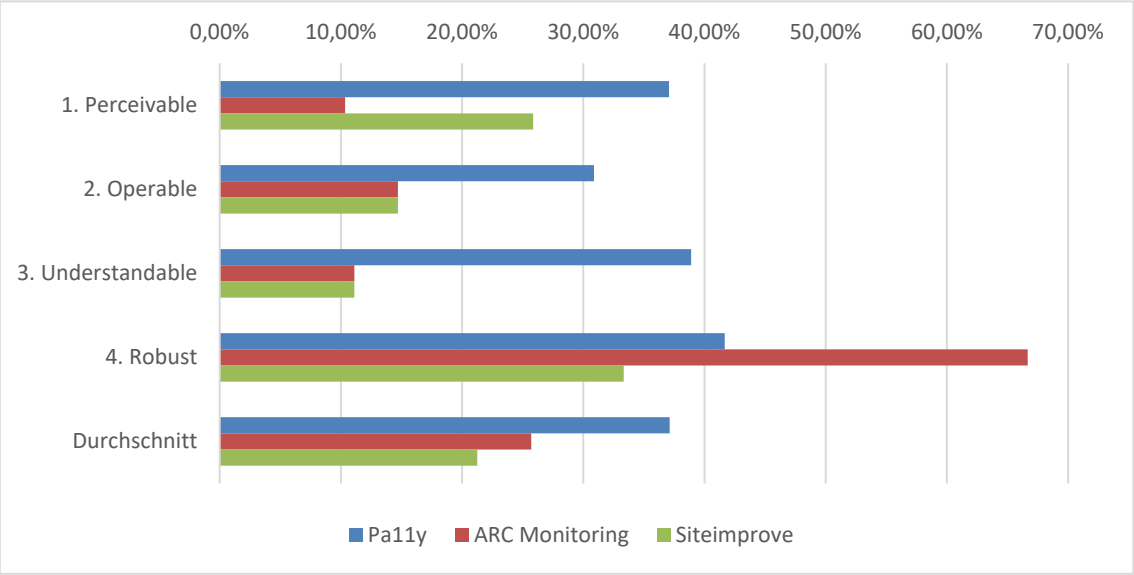
Der Wert von Pa11y verschlechtert sich nun von 36 % (siehe Abbildung 18) auf 21 % (siehe Abbildung 20). Jetzt weisen Pa11y und Siteimprove die gleiche Coverage der Success Criteria auf.

6.1.2 Coverage hinsichtlich der WCAG Principles

Nun ist es uns möglich zu evaluieren, wie die Coverage hinsichtlich der Schichten der WCAG 2.1 ausfällt, die wir in Kapitel 2.1.2 bereits erwähnt haben.

Abbildung 21

Coverage nach WCAG Principles



Pa11y erreicht durchschnittlich die höchste Coverage der WCAG Principles mit 37 %, gefolgt von ARC Monitoring mit 25 % und Siteimprove mit 21 % (siehe Abbildung 21).

Tabelle 4

Ergebnisse der Auswertung der Coverage der Principles

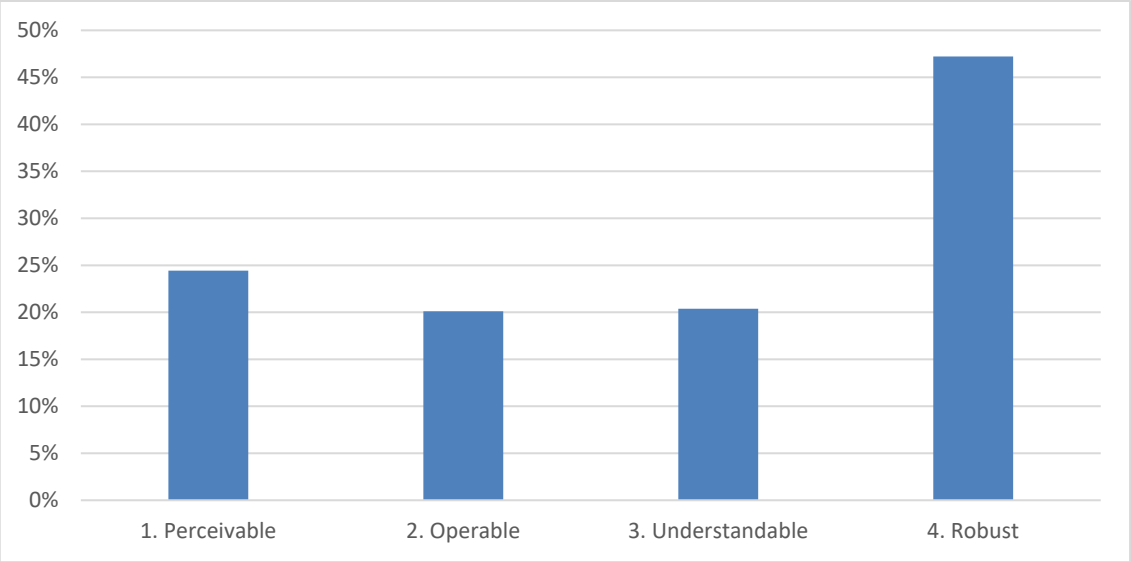
	Pa11y	**ARC Monitoring**	**Siteimprove**	**Durchschnitt**
1. Perceivable	37 %	10 %	26 %	24 %
2. Operable	31 %	15 %	15 %	20 %
3. Understandable	39 %	11 %	11 %	20 %
4. Robust	42 %	67 %	33 %	47 %
Durchschnitt	37 %	26 %	21 %	

Die höchste Coverage eines Principles besitzt ARC Monitoring mit 67 % im Principle 4 Robust. Dieses Principle wird auch durchschnittlich mit 47 % am besten von allen Tools abgedeckt (siehe Tabelle 4).

Mit nur drei zugehörigen Success Criteria handelt es sich bei Robust um das Principle mit dem kleinsten Umfang, das bedeutet einzelne gefundene oder nicht gefundene Fehler haben eine große Auswirkung: ARC Monitoring konnte hier als einziges Tool den Fehler zum Success Criterion 4.1.2 Name, Role, Value als Error identifizieren.

Abbildung 22

Durchschnittliche Coverage der WCAG 2.1 Principles über alle Tools hinweg

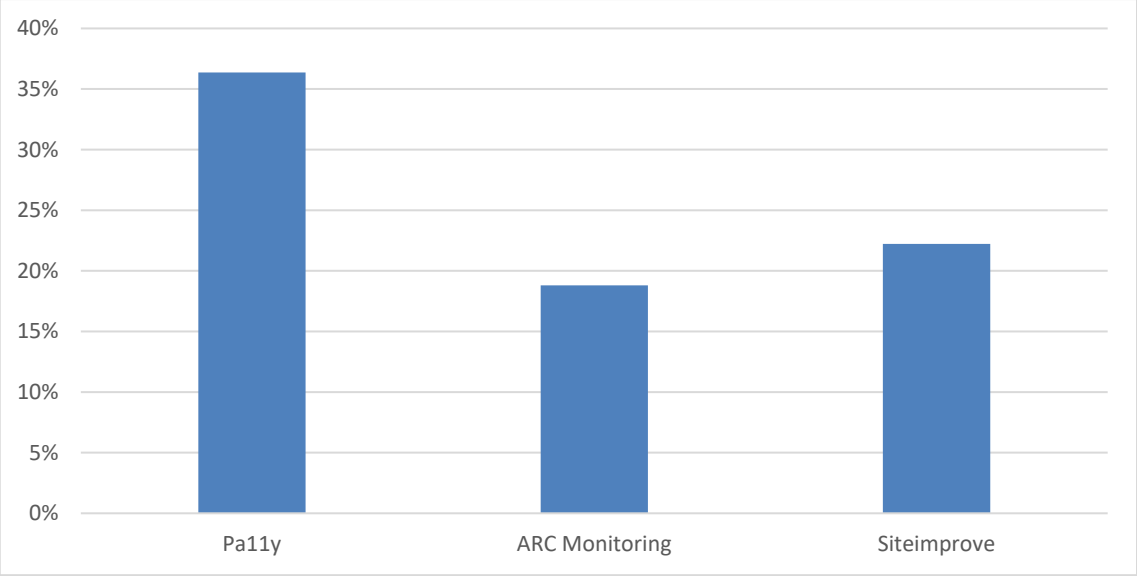


Wenn wir alle Tools zusammenfassen, wird das Principle 4 Robust von den Tools mit 47 % mit Abstand am besten abgedeckt, die Principles 2 Operable und 3 Understandable mit 20 % am schlechtesten (siehe Abbildung 22).

6.1.3 Coverage hinsichtlich der WCAG 2.1 Guidelines

Abbildung 23

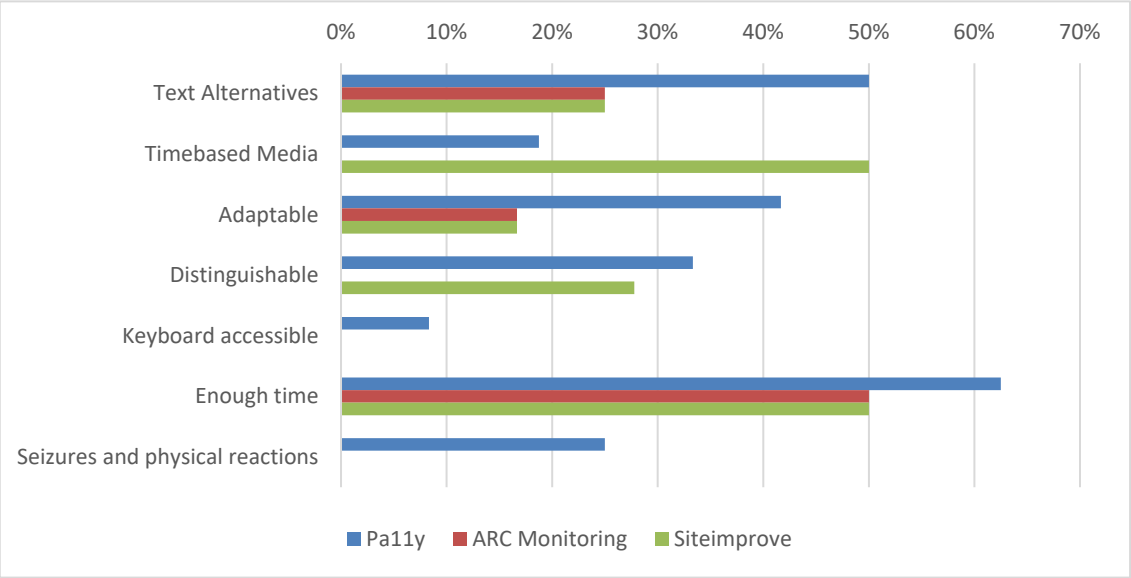
Coverage der WCAG Guidelines der AMS



Pa11y weist wie in Abbildung 23 dargestellt mit 36 % die höchste durchschnittliche Coverage der WCAG Guidelines auf, ARC Monitoring mit 19 % die geringste.

Abbildung 24

Coverage der WCAG Guidelines 1-7

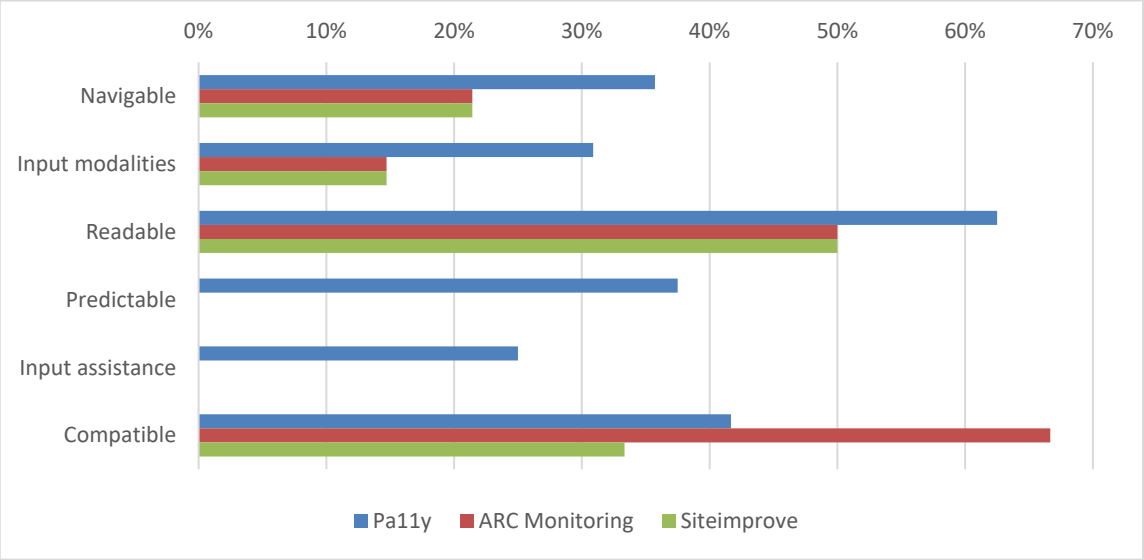


Von den WCAG Guidelines 1-7 wird die Guideline Enough Time mit 64 % vom AMS Pa11y am besten abgedeckt. In dieser Guideline haben die anderen beiden AMS eine Coverage von 50 %.

Bei den Guidelines Keyboard Accessible und Seizures and physical reactions weisen ARC Monitoring und Siteimprove gar keine Coverage auf, nur Pa11y findet hier Fehler.

Abbildung 25

Coverage der WCAG Guidelines 8-13

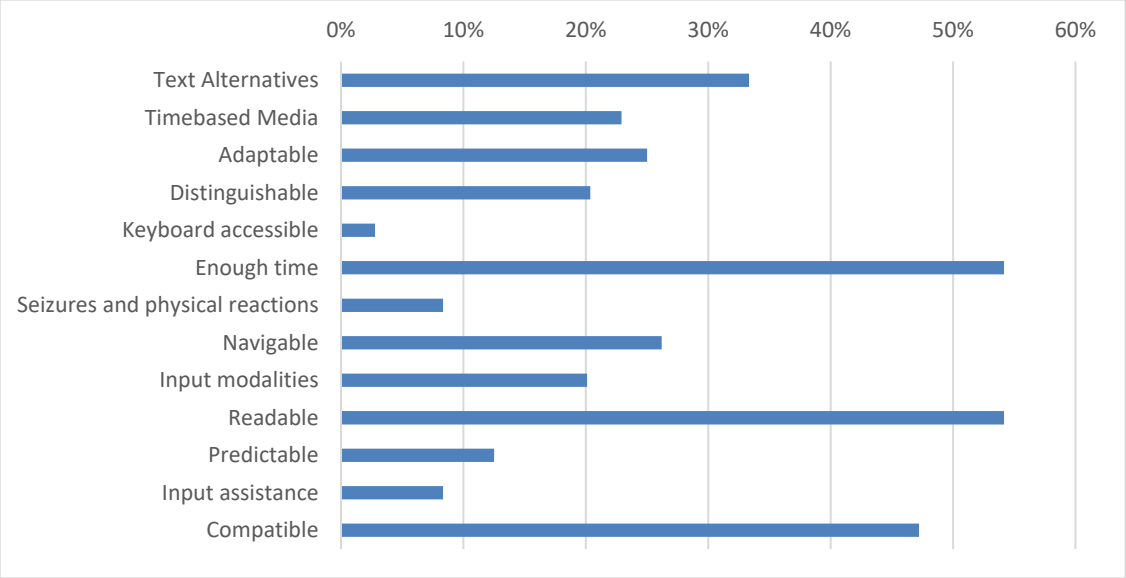


In der letzten Guideline Compatible besitzt ARC Monitoring eine Coverage von nahezu 70 %, die Coverage der beiden anderen Tools fällt viel geringer aus. Die Erklärung hierfür ist die gleiche wie bereits in Punkt 6.1.2 erläutert.

In den Guidelines Predictable und Input Assistance ist Pa11y erneut das einzige AMS, das eine Coverage vorweisen kann.

Abbildung 26

Durchschnittliche Coverage aller Tools nach WCAG Guidelines



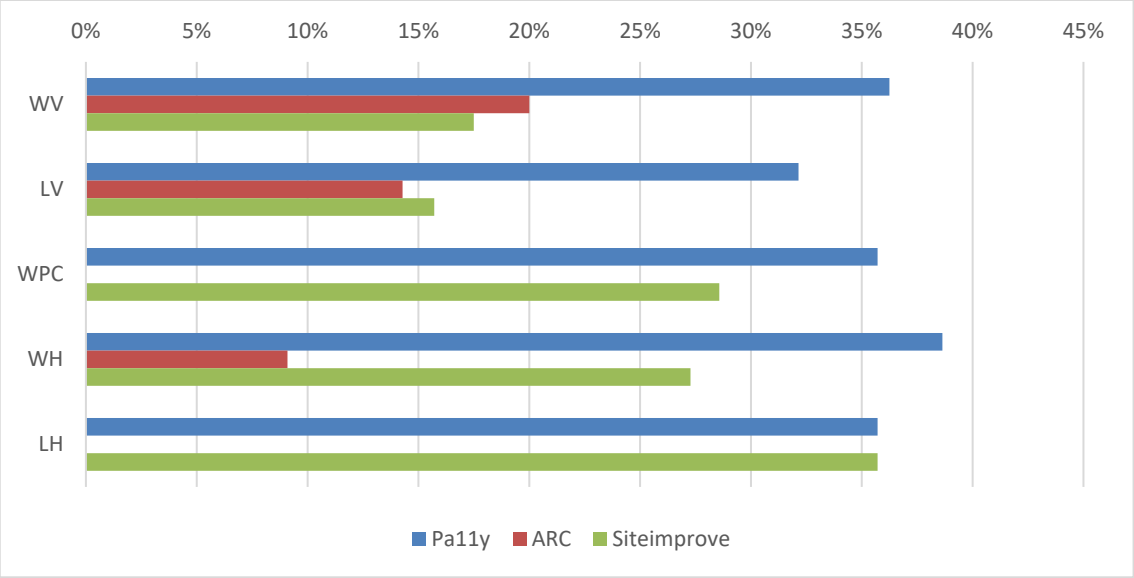
Die Guidelines Readable und Enough Time werden von allen Tools durchschnittlich mit 54 % am besten erfasst, gefolgt von Compatible mit 47 %.

Am schlechtesten abgedeckt wird die Guideline Keyboard Accessible mit 3 % (siehe Abbildung 26).

6.1.4 Coverage hinsichtlich der betroffenen Benutzendengruppen

Abbildung 27

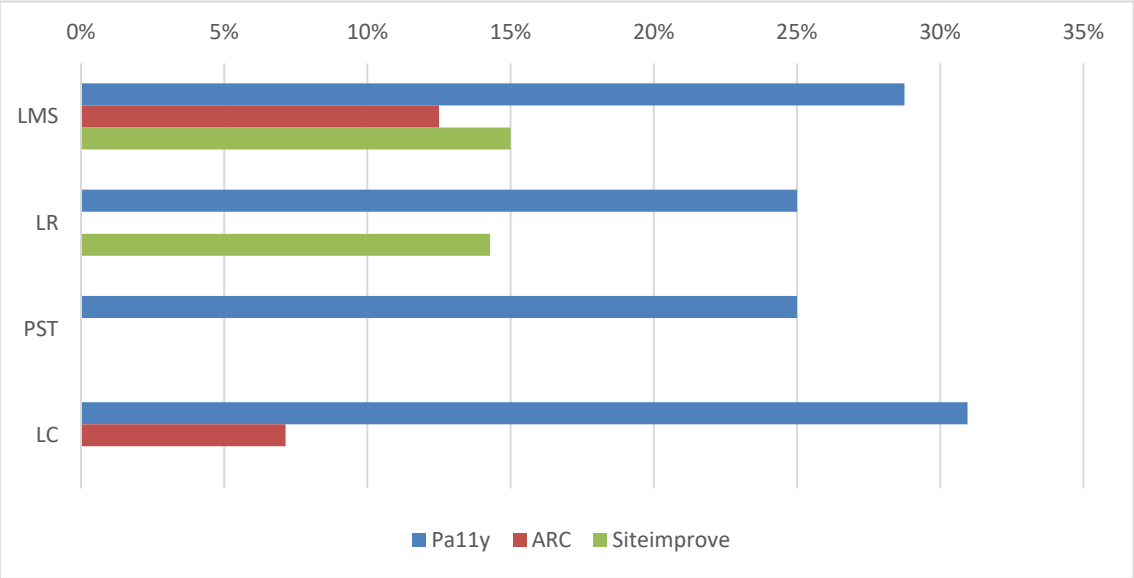
Coverage hinsichtlich der betroffenen Benutzendengruppen 1-5



Die beste Coverage innerhalb der betroffenen Benutzendengruppen 1-5 erzielt Pa11y mit 39 % für die Gruppe WH: Nutzung ohne Hörvermögen. Die schlechteste Coverage besitzt ARC Monitoring in den Kategorien WPC: Nutzung mit eingeschränkter Farbwahrnehmung und LH: Nutzung mit eingeschränktem Hörvermögen, hier besteht gar keine Coverage.

Abbildung 28

Coverage hinsichtlich der betroffenen Benutzendengruppen 6-9



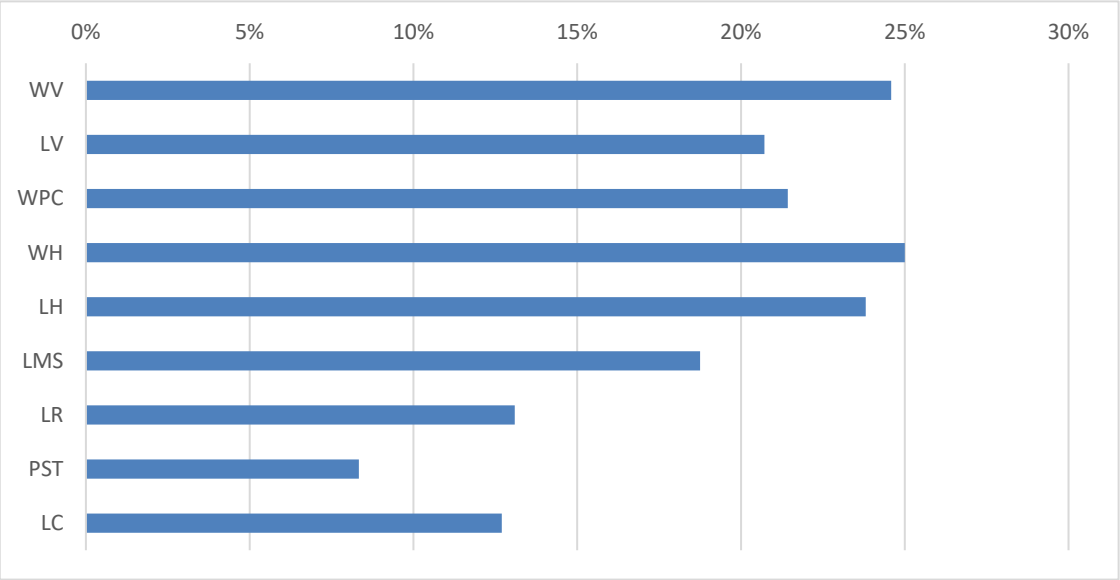
Die größte Coverage innerhalb der Gruppen 6-9 besitzt Pa11y in der Gruppe LC: Nutzung mit körperlichen Einschränkungen mit 31 %.

ARC Monitoring besitzt in den Gruppen LR: Nutzung mit eingeschränkter Reichweite und PST: Verringerung von Anfallsauslösern bei Photosensibilität die geringste Coverage mit 0 %.

Insgesamt erzielt Pa11y Ergebnisse in allen neun betroffenen Benutzendengruppen, Siteimprove in sieben und ARC Monitoring in nur fünf der Gruppen.

Abbildung 29

Durchschnittliche Coverage der betroffenen Benutzendengruppen aller Tools



Die Benutzendengruppe PST weist mit 8 % die geringste durchschnittliche Coverage auf (siehe Abbildung 29).

PST steht nach Tabelle 2 für Verringerung von Anfallsauslösern bei Photosensibilität. Das bedeutet, dass von allen AMS zusammengenommen am wenigsten Success Criteria abgedeckt wurden, die in Zusammenhang mit einer Verringerung von Anfallsauslösern bei Photosensibilität stehen. Konkret handelt es sich um die Success Criteria 2.3.1, das von keinem der AMS, und 4.3.1, das lediglich von Pa11y in Form einer Notice erkannt wurde.

Die höchste Coverage besteht bei den Benutzendengruppen WV: Nutzung ohne Sehvermögen und WH: Nutzung ohne Hörvermögen mit jeweils 25 % (siehe Abbildung 17). Die AMS finden also am meisten Fehler, die Barrieren für Personen darstellen, die die geprüfte Website ohne Seh- oder ohne Hörvermögen nutzen.

6.1.5 Correctness

Tabelle 5

Ergebnisse der Auswertung der Correctness

	Pa11y	ARC Monitoring	Siteimprove
Richtig Positive	66	45	20
Falsch Positive	233	58	7

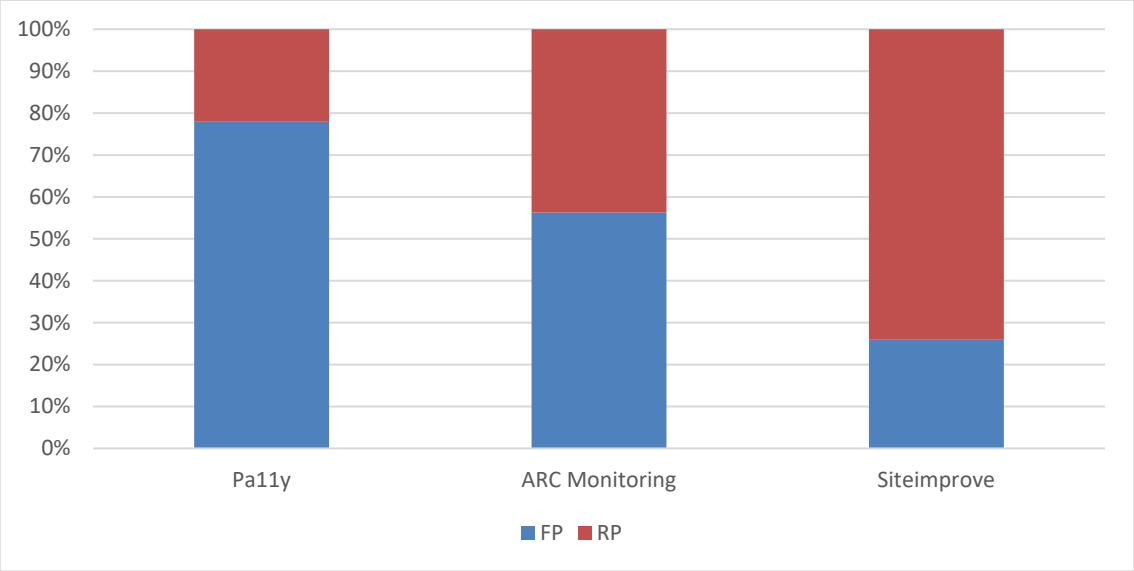
Correctness mit Pa11y Notices	22 %	45 %	74 %
Correctness ohne Pa11y Notices	73 %	45 %	74 %

Pa11y weist mit 22 % bei weitem den niedrigsten Correctness Wert auf, während Siteimprove mit 74 % den Höchsten Wert hat (siehe Tabelle 5 und Abbildung 30).

Pa11y meldet mit 233 bei weitem die meisten FP, verglichen mit 7 FP bei Siteimprove, das die geringste Anzahl meldet (siehe Tabelle 5).

Abbildung 30

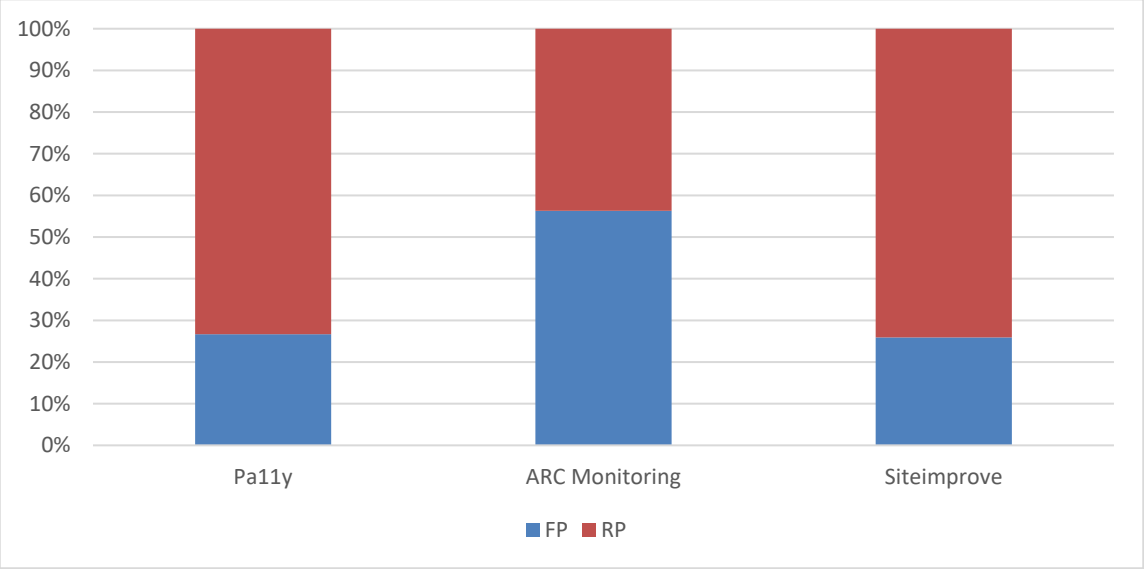
Correctness: Verhältnis von RP und FP der AMS



Das schlechte Abschließen von Pa11y hängt vermutlich damit zusammen, dass das Tool viele unspezifische Meldungen zur manuellen Überprüfung ausgibt. Dadurch, dass sich diese oft auf alle Elemente eines bestimmten den Fehler betreffenden Typs, zum Beispiel Formulare, beziehen entstehen auch viele FP.

Abbildung 31

Correctness: Verhältnis von RP und FP der AMS (ohne Einbezug der Pa11y Notices)



Wie bereits bei der Coverage der Succes Criteria berechneten wir auch hier die Correctness ein zweites Mal, nur ohne die Notices von Pa11y einzubeziehen. Dies hat eine signifikante Verbesserung des Werts für Pa11y zur Folge: von 22 % (siehe Tabelle 5) auf 73 % (Siehe Abbildung). Die Correctness von Pa11y und Siteimprove ist nun gleich.

6.2 Heuristische Evaluation

Die heuristische Evaluation soll einen Überblick über die Usability der Fehlerberichte der AMS geben und Verbesserungsmöglichkeiten aufzeigen. Sie wurde im Rahmen dieser Bachelorthesis von einer Person durchgeführt.

6.2.1 Report zur Heuristischen Evaluation von Pa11y

1. Unverständliche Überschriften

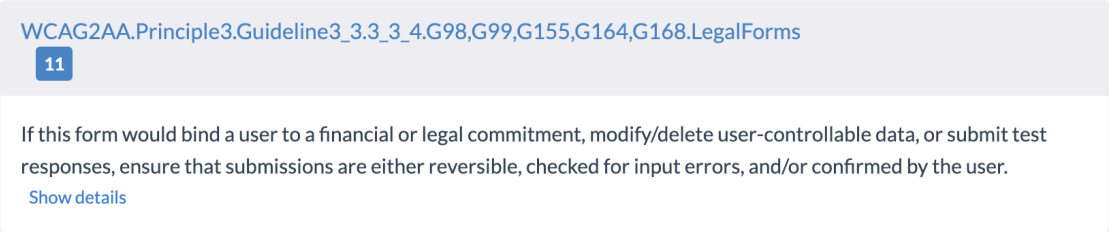
Verletzte Regel: Nummer 3 – Match between system and the real world

Beschreibung: Die Überschriften für die einzelnen Verstöße sind lang und eine Kombination verschiedener aneinandergereihter Informationen ohne Leerzeichen (siehe Abbildung 32). Es ist möglich, Informationen wie das übergeordnete WCAG Principle, das Konformitätslevel oder damit verbundene WCAG Techniques aus der Überschrift herauszufiltern, jedoch nur umständlich und mit Vorwissen.

Lösungsvorschlag: Aufteilung der Informationen in einzelne, klar unterscheidbare Textfelder. So wird die Überschrift so reduziert, dass der Fehler leicht verständlich und schnell erkennbar ist.

Abbildung 32

Fehlermeldung im Pa11y Dashboard (eigener Screenshot)



2. Fehlende Dokumentation

Verletzte Regel: Nummer 10: Help and documentation

Beschreibung: Für das Pa11y Dashboard gibt nur eine Dokumentation in Form einer readme-Datei im zugehörigen Github Repository. Hier wird mehr die Installation und Konfiguration beschrieben als die tatsächliche Nutzung des Dashboards.

Lösungsvorschlag: Bereitstellung einer Dokumentation zur Benutzung des Pa11y Dashboards, die auch innerhalb der Anwendung im passenden Kontext referenziert wird.

6.2.2 Report zur Heuristischen Evaluation von ARC Monitoring

1. Keine Anzeige über den Ladestand der Seite

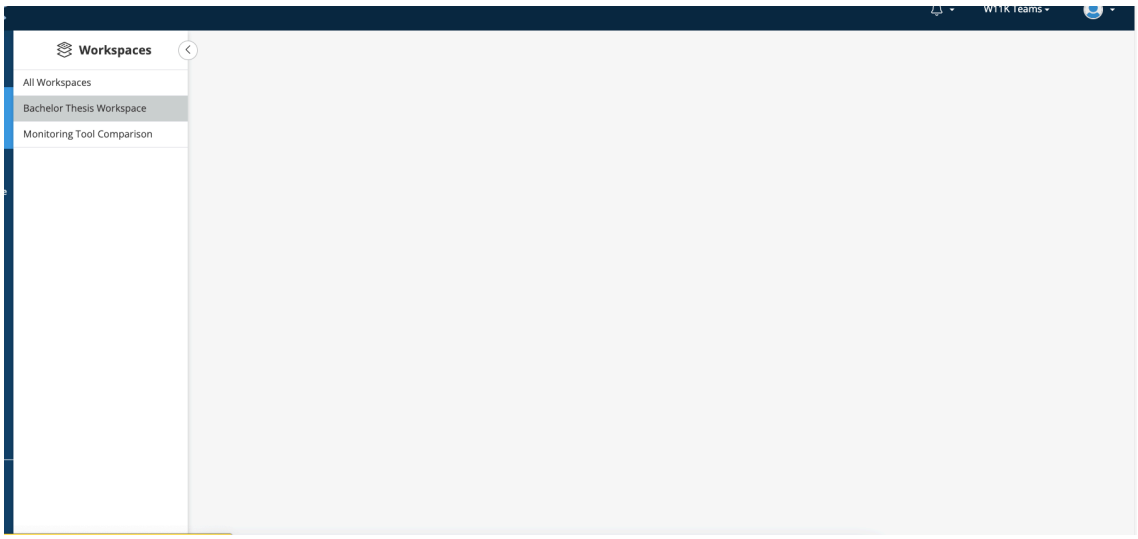
Verletzte Regel: Nummer 1 – Visibility of system status

Beschreibung: Beim Laden der Inhalte kommt es zu merklichen Ladezeiten. An manchen Stellen im System ist hier ein Ladebalken vorhanden, beim Laden der Fehlermeldungen zum Beispiel jedoch nicht. Nutzende wissen so nicht, was das System gerade macht und ob möglicherweise ein Fehler vorliegt. Dies stört das Vertrauen in das System.

Lösungsvorschlag: Einbindung eines Ladebalkens oder anderen Ladeindikators für eine klare Kommunikation zwischen System und Nutzenden

Abbildung 33

Ansicht von ARC Monitoring während Inhalte geladen werden (eigener Screenshot)



2. Aktuelle Position nicht deutlich

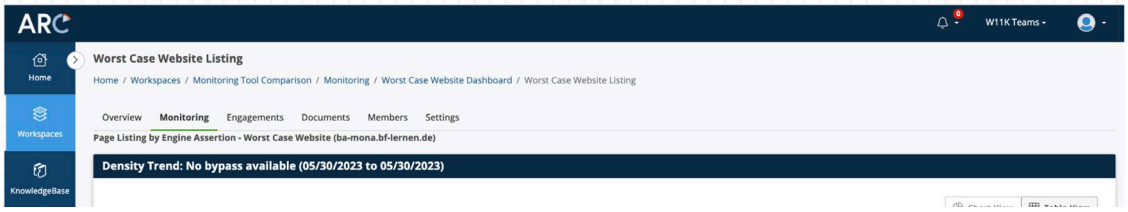
Verletzte Regel: Nummer 1 – Visibility of system status

Beschreibung: In der Breadcrumb-Navigation wird nicht dargestellt, dass gerade die Detailseite einer konkreten Fehlermeldung geöffnet ist, sondern diese allgemein als „Listing“ bezeichnet. Die genaue Bezeichnung des Fehlers ist erst weiter unten auf der Seite auffindbar. So kann es zu Verwirrung kommen, wo sich Benutzende genau befinden und ob die richtige Fehlermeldung geöffnet wurde.

Lösungsvorschlag: Erweiterung der Breadcrumb-Navigation um den Namen der spezifischen Fehlermeldung und Hinzufügen einer Überschrifft zu Beginn der Seite mit dem genauen Namen der Fehlermeldung

Abbildung 34

Breadcrumb-Navigation in ARC Monitoring (eigener Screenshot)



3. Verwirrende Hierarchie-Ebenen in der Navigation

Verletzte Regel: Nummer 3 – User control and freedom

Beschreibung: Die vielen verschiedenen Inhalte in der ARC-Plattform machen den Aufbau des Systems komplex. Diese Komplexität wird auch in den verschiedenen Menü-Formen deutlich: Eine Seitenleiste stellt die erste Ebene dar, innerhalb eines sogenannten Workspaces eine Ebene darunter wird mithilfe einer Tab-Navigation navigiert. Die Breadcrumb-Navigation mischt beides, indem sie Benutzenden anzeigt, wo sie sich ausgehend von der Startseite des Systems befinden.

Beim Versuch, über die vorhandenen Navigationselemente von einer Fehlermeldung aus eine Ebene nach oben zur Übersicht über alle Fehlermeldungen zu navigieren kann es zu Fehlern kommen, wenn zum Beispiel aus Versehen die Tab-Navigation benutzt wird. Die Tab-Navigation ist da, um zwischen in der Hierarchie des Systems viel höher gelegenen Bereichen zu navigieren, ist jedoch näher am Seiteninhalt platziert als die Breadcrumb-Navigation.

Lösungsvorschlag: Die Tab-Navigation sollte über der Breadcrumb-Navigation platziert werden, um die höhere Hierarchie-Ebene zu verdeutlichen. Die Breadcrumb-Navigation sollte die genaue aktuelle Position vom jeweils gewählten Tab aus anzeigen und nicht von der Startseite aus.

Abbildung 35

Hierarchieebenen der Navigation in ARC Monitoring (eigener Screenshot)

4. Darstellung von Links

Verletzte Regel: Nummer 4 – Consistency and standards

Beschreibung: Abbildung 36 zeigt eine Tabelle, die für einen bestimmten Fehler (hier „Page has auto-refresh“) die gescannten Seiten anzeigt, auf denen der Fehler vorliegt. Die Links in der Spalte URL verlinken zu einer Übersicht, auf der unter anderem alle der jeweiligen Seite zugeordneten Fehler aufgelistet sind. Die Darstellung des Links mit farbiger Hervorhebung und Unterstreichung in Kombination mit dem Linktext, der nur aus

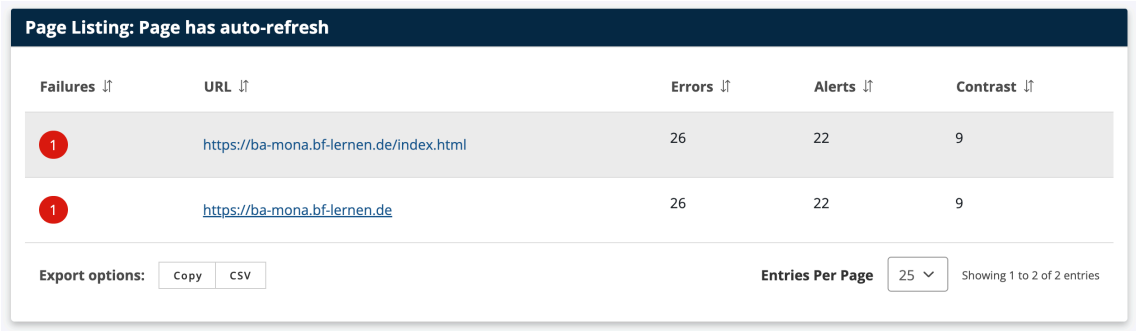
der URL besteht, passt in die übliche Darstellung von Links in Dokumenten und auf Websites.

So entsteht der Eindruck, dass der Link auf die Website mit der angegebenen URL verweist und nicht wie tatsächlich auf eine Übersichtsseite im AMS.

Lösungsvorschlag: Erweiterung des Linktexts um Stichworte wie Dashboard oder Übersicht oder Verlagerung der Verlinkung auf einen Button, der das Ziel der Aktion durch einen Buttontext oder ein Icon angibt

Abbildung 36

Darstellung der Vorkommnisse eines Fehlers in den Components von ARC Monitoring



5. Darstellung von zugehörigen Code-Elementen

Verletzte Regel: Nummer 4 – Consistency and standards

Beschreibung: Abbildung 37 zeigt, wie in einer Fehlermeldung auf das konkrete HTML-Element hingewiesen wird, das mit dem Fehler zusammenhängt. Beim Hovern über das Element mit der Maus erhält es wie dargestellt eine graue Kontur. Durch die Verwendung eines Hover-Effekts entsteht hier die Erwartung, dass das Element als Button fungiert und anklickbar ist. Dies ist jedoch nicht der Fall, der Hover-Effekt ist unnötig.

Lösungsvorschlag: Entfernung des Hover-Effekts

Abbildung 37

Hinweis auf Elemente im Source Code in einer Fehlermeldung in ARC Monitoring

6.2.3 Report zur Heuristischen Evaluation von Siteimprove

1. Gleiches Icon für gelöste Probleme und nicht relevante Probleme

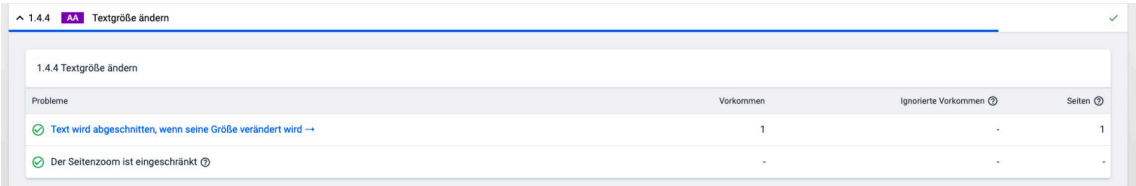
Verletzte Regel: Nummer 4 – Consistency and standards

Beschreibung: In der Übersicht Richtlinien stellt Siteimprove die gefundenen tatsächlichen potenziellen und gelösten Probleme zugeordnet zu dem als Seitenziel ausgewählten Standard dar. Hierbei wird auch eine Aussage darüber getroffen, wenn ein Problem als nicht relevant für die Website eingeschätzt wird. Die nicht relevanten Probleme (erkennbar durch die fehlende Verlinkung und ein Fragezeichen-Icon, welches mit einem Tooltip weitere Erklärung anbietet) werden ebenso wie die gelösten Probleme mit einem grünen Haken-Icon dargestellt (siehe Abbildung 38). Dies verringert die Unterscheidbarkeit der beiden Problemtypen.

Lösungsvorschlag: Zuordnung eines anderen Icons zu einem der beiden Problemtypen

Abbildung 38

Darstellung von gelösten Problemen und nicht relevanten Problemen in Siteimprove (eigener Screenshot)



6.2.4 Fazit zur Heuristischen Evaluation

Das Pa11y Dashboard bietet von allen Tools den geringsten Funktionsumfang. Das und der somit simplere Aufbau des AMS begünstigen die reduzierte Gestaltung des UI. Aus den gefundenen Usability Problemen wird deutlich, dass das System die Nutzenden besser anleiten und mehr ihre Sprache sprechen sollte, um eine möglichst intuitive Benutzung zu ermöglichen.

Beim AMS ARC Monitoring fanden wir die meisten Usability Probleme. Die vielen unterschiedlichen Navigationsebenen und -mechanismen führen zu Verwirrung und Unklarheit über die aktuelle Position im System. Gerade wenn oft zwischen einzelnen Fehlerberichten und der Gesamtübersicht aller Fehler hin- und hergewechselt werden muss passieren unbeabsichtigte Navigationsfehler, die ein effizientes Arbeiten behindern. Direkt im Kontext der Anwendung angebrachte Erklärungen der Funktionen und

Begriffsbedeutungen würden das Erlernen des AMS erleichtern und womöglich auf lange Sicht die unbeabsichtigten Fehler in der Benutzung verringern.

Siteimprove schneidet in der heuristischen Evaluation am besten ab. Bis auf einen Fall von Inkonsistenz bei verwendeten Icons sind uns keine weiteren Usability Probleme aufgefallen. Das UI von Siteimprove ist modern und klar gestaltet. Verwendete Begriffe und System Features werden mithilfe von Tooltips direkt im Kontext erklärt und fördern so das Lernen des Systems. Die Navigationshierarchie ist klar, da nur eine Navigation verwendet wird.

7 Diskussion

7.1 Diskussion der Ergebnisse

Ziel dieser Thesis war es, die AMS Pa11y, ARC Monitoring und Siteimprove nach qualitativen und quantitativen Aspekten zu vergleichen. Die vielen unterschiedlichen qualitativen Aspekte sind auf der einen Seite wichtig, um die unterschiedlichen Modelle und Funktionalitäten der AMS widerzuspiegeln. Auf der anderen Seite erschweren sie ein eindeutiges Ergebnis. Daher stellen sie vor allem eine Entscheidungshilfe dar, die in konkreten Sachverhalten angewendet werden kann.

Das Pa11y Dashboard ist von allen AMS am stärksten auf das Finden und Anzeigen von Barrierefreiheitsfehlern fokussiert und bietet als komplett kostenloses Tool die niedrigste Einstiegshürde. Für die Installation und das Hosting des Pa11y Dashboards ist ein technischer Hintergrund notwendig. Das Tool erweckt den Eindruck, besonders Webentwickler:innen als Zielgruppe zu haben, die die starke Anpassbarkeit des Tools aufgrund seines Open Source Charakters und die ergänzenden Tools für CI und CLI zu schätzen wissen. Ein Problem von Pa11y ist, dass nur einzelne URLs und keine zusammenhängenden Websites analysiert werden können, was bei umfangreichen Websites einen großen Mehraufwand erzeugt.

Die Analyse der Usability ergab, dass das Pa11y Dashboard die Nutzenden noch besser mit einer umfassenden Dokumentation und Erklärungen innerhalb des Tools unterstützen könnte. Gerade wenn sich gefundene potenzielle Fehler auf viele Elemente auf der Seite beziehen, kann es ohne HTML-Kenntnisse schwierig sein, durch das Tool das richtige Element zu finden.

In der quantitativen Auswertung erzielt Pa11y insgesamt die höchste Coverage der WCAG Success Criteria mit 36 %. Im Gegensatz zu den anderen AMS hat Pa11y mit den Notices eine dritte, unspezifische Meldungskategorie, die oft zutreffend ist. Bei umfangreichen Websites erschwert die dadurch entstehende große Anzahl an Meldungen das Finden der wirklichen Fehler, was sich in der niedrigen Correctness und der hohen Anzahl an FP ausdrückt. Eine erneute Berechnung ohne Einbezug der Notices von Pa11y verschlechtert das Coverage Ergebnis bei Pa11y zwar, verbessert allerdings die Correctness signifikant. Pa11y weist die insgesamt beste Coverage einer betroffenen Benutzendengruppe mit 39 % für die Gruppe Nutzung ohne Hörvermögen auf.

ARC Monitoring ist aufgrund seiner flexiblen Preisgestaltung attraktiv. Die Dokumentation des Systems und die Wissensinhalte zur Barrierefreiheit sind umfangreich. Im Gegensatz zu Pa11y ist es hier möglich, ganze Websites auf einmal zu crawlen. Allerdings

werden die Ergebnisse der Startseite doppelt dargestellt, was angezeigte Gesamtergebnisse teilweise verfälscht.

Bei ARC Monitoring konnten wir durch die Heuristische Evaluation viele Usability Probleme aufdecken, die die Benutzung erschweren: der Aufbau des Tools ist komplex, die Navigation zwischen den Bereichen ist verwirrend und fehleranfällig. Zusätzlich fehlen Erklärungen zur Nutzung des Tools direkt im Kontext, um das Lernen des Tools zu fördern. So ist das AMS für uns das Tool mit der schlechtesten Usability.

ARC Monitoring weist in der quantitativen Auswertung in der Kategorie Coverage insgesamt die schlechtesten Werte auf. Innerhalb der Coverage der Guidelines gibt es jedoch ein paar Abweichungen gegenüber den anderen beiden AMS, wenn ARC Monitoring Success Criteria abdeckt, die die anderen beiden nicht finden konnten. Das Tool hat einen mittleren Correctness-Wert mit 45 %.

Siteimprove ist ein umfangreiches Tool, das genau wie ARC Monitoring gesamte zusammenhängende Websites auf einmal prüft. Es bietet professionellen Teams viele Möglichkeiten an, Websites nach weiteren Kriterien neben der Barrierefreiheit zu analysieren. Zusätzlich gibt es viele mögliche Schnittstellen zu anderen Tools. Gefundene Fehler werden erklärt und direkt im Code und einer visuellen Ansicht der Webseite markiert, was besonders unerfahrenen Benutzenden das Finden und Beheben der Barrieren erleichtert.

Das AMS Siteimprove ist für uns das Tool mit der besten Usability, wir konnten insgesamt in den Fehlerberichten die wenigsten Usability Probleme finden.

Siteimprove erzielt eine schlechtere Coverage der Success Criteria als Pa11y, jedoch eine bessere Correctness. Nach der erneuten Berechnung der Werte für Pa11y sind die Ergebnisse von Pa11y und Siteimprove annähernd gleich.

Die Analyse der durchschnittlichen Coverage der AMS soll zeigen, welche Teile der WCAG automatisch prüfbar sind. Das Principle Robust wird am besten von den AMS abgedeckt, die Coverage ist fast doppelt so groß wie die der anderen Principles. Die AMS erfassen die Guidelines Readable und Enough Time durchschnittlich am besten, die Guideline Keyboard Accessible am schlechtesten.

Die maximale durchschnittliche Coverage der betroffenen Benutzendengruppen über alle AMS beträgt nur 25 % in den Gruppen Nutzung ohne Hörvermögen und Nutzung ohne Sehvermögen. Vor allem Photosensibilität wird nicht gut von den AMS abgedeckt.

Insgesamt zeigen die durchschnittlichen quantitativen Resultate, dass ein Großteil der WCAG nicht automatisch geprüft werden konnte. Auch wir kommen so wie bereits

Burkard et al. (2021) und Vigo & Brajnik (2011) zu dem Ergebnis, dass AMS nicht alleinig zur Überprüfung der Barrierefreiheit einer Website eingesetzt werden sollten.

Vor allem die Notices von Pa11y deuten für uns auf einen möglichen Einsatzzweck von automatischen Prüftools hin: als Checkliste für manuelle Prüfungen. Automatisch prüfbare Success Criteria werden vom AMS analysiert und bei allen anderen leistet das Tool eine Anleitung zur manuellen Überprüfung. Dies würde den Nutzen von AMS vergrößern, vor allem solange noch so große Bereiche der WCAG nicht automatisch geprüft werden können.

7.2 Limitationen

Der Umfang der Test-Website beinhaltet einen Fehler passend zu jedem WCAG Success Criterion. Wir können jedoch nicht ausschließen, dass die AMS im Fall eines FN einen anders konstruierten Fehler zum selben Success Criterion doch hätten finden können und so ein RP erzielt hätten. Daher stellt sich uns die Frage, inwieweit es überhaupt möglich ist, eine absolut repräsentative Test-Website zu erstellen.

Die Usability der vorliegenden Tools wurde im Rahmen dieser Bachelorthesis anhand einer Heuristischen Evaluation bewertet. Um repräsentative Ergebnisse über die Usability der AMS zu erzielen, wäre ein umfangreicher User Test mit einer repräsentativen Anzahl an Teilnehmenden sinnvoll. Die Heuristische Evaluation wurde zudem nur von einer Person durchgeführt, daher sind die Ergebnisse als subjektiv zu bewerten und gewähren lediglich einen Einblick in die Usability der AMS.

Gegenstand unseres Vergleichs sind insgesamt drei AMS. In unseren qualitativen Ergebnissen versuchen wir Aussagen zur WCAG Coverage über alle drei AMS hinweg zu treffen. Aufgrund der geringen Anzahl an AMS lassen sich aus den Ergebnissen jedoch keine repräsentativen Aussagen über alle AMS allgemein treffen.

8 Fazit und Ausblick

Gegenstand dieser Bachelorthesis war der Vergleich der drei AMS Pa11y, ARC Monitoring und Siteimprove. Nach einer Vorstellung der einzelnen Systeme stellten wir die AMS zuerst in qualitativen Aspekten gegenüber, um die potenziellen Vor- und Nachteile aufzuzeigen. Hierbei ist es wichtig, dass es in vielen Punkten auf die individuellen Bedürfnisse und Anforderungen ankommt und die vorgestellten Aspekte eine Entscheidungshilfe darstellen sollen.

Anschließend wendeten wir die AMS auf eine selbst konstruierte Test-Website an und evaluierten die quantitativen Ergebnisse nach den Evaluationskriterien Coverage und Correctness.

Mithilfe einer Heuristischen Evaluation anhand Nielsens 10 Heuristiken konnten wir einen Einblick in die Usability der Tools erhalten.

Wir kommen zu dem Ergebnis, dass wir die Verwendung des Tools ARC Monitoring am wenigsten empfehlen können, da die quantitativen Resultate mit am schlechtesten ausfielen und am meisten Usability Probleme gefunden wurden. Überraschend für uns ist, dass das kostenlose AMS Pa11y quantitativ ähnlich gut abschneidet wie das Enterprise-Tool Siteimprove. Wenn der große Umfang an Funktionalität nicht notwendig und technische Expertise vorhanden ist, stellt Pa11y eine gute Alternative mit niedriger Einstiegshürde dar. Siteimprove hat die wenigsten Usability Probleme und empfehlen wir vor allem für unerfahrene Nutzende.

Die quantitative Analyse der Coverage der verschiedenen Schichten der WCAG zeigt, dass AMS in unserem Testaufbau manuelle Tests nicht ersetzen können. Die erzielten Werte, egal ob durchschnittlich oder von einem AMS, übersteigen selten 50 %.

In der Analyse der Coverage nutzen wir eine Abstufungsskala, sodass unspezifische Meldungen weniger ins Gewicht fallen als sicher gefundene Fehler. Ziel weiterer Forschung könnte es sein anhand von User Tests zu überprüfen, welche Auswirkung die unterschiedlichen Abstufungen tatsächlich auf Endnutzende haben.

Anhand der drei AMS die Gegenstand dieser Bachelorthesis sind trafen wir übergreifende Aussagen über die Coverage der WCAG. Diese repräsentieren jedoch nur die drei AMS in diesem Vergleich. Um festzustellen, wie der Entwicklungsstand von AMS aktuell ist und wie weit eine automatische eine manuelle Prüfung ersetzen kann, sollte eine Studie mit einer repräsentativen Anzahl an AMS durchgeführt werden. So könnte auch festgestellt werden, wie prüfbar die WCAG 2.1 allgemein sind. Im Rahmen einer solchen Untersuchung könnte auch genauer analysiert werden, mit welchen Faktoren es zusammenhängt, wenn ein Fehler von einem AMS nicht gefunden werden konnte. Diese

Faktoren könnten zum Beispiel sein, welche Technologie verwendet wurde oder ob der tatsächliche Inhalt der Webseite dafür relevant ist. Aufgrund der großen Anzahl der konstruierten Fehler der Test-Website war dies im Rahmen dieser Bachelorthesis nicht möglich.

Um automatische Prüftools mit speziell manueller Prüfung zu vergleichen könnten die erzielten Ergebnisse der AMS noch mit einer repräsentativen Anzahl an manuellen Prüfungen verglichen werden.

Anhang A: Test-Website

A.1 Konzeption der Test-Website

Zuordnung von BITV-Prüfschritten, WCAG Failures und Umsetzung im Code der Website

Link: https://gitlab.mi.hdm-stuttgart.de/ms553/bachelor-thesis/-/blob/main/A-1-Konzeption-Testwebsite.pdf

A.2 Code der Test-Website

Link: https://gitlab.mi.hdm-stuttgart.de/ms553/bachelor-thesis/-/tree/main/A-2-Code-Testwebsite

Anhang B: Auswertung

B.1 Auswertung

Tabelle zur Auswertung der Evaluationskriterien des quantitativen Vergleichs

Link: https://gitlab.mi.hdm-stuttgart.de/ms553/bachelor-thesis/-/blob/main/B-1-Auswertung.xlsx

Literaturverzeichnis

Abduganiev, S. G. (2017). Towards Automated Web Accessibility Evaluation: A Comparative Study. *International Journal of Information Technology and Computer Science*, *9*(9), 18–44. https://doi.org/10.5815/ijitcs.2017.09.03

About TPGi—The Accessibility Experts. (o. J.). TPGi. Abgerufen 2. August 2023, von https://www.tpgi.com/about/

Al-Khalifa, H. S. (2012). WCAG 2.0 Semi-automatic Accessibility Evaluation System: Design and Implementation. *Computer and Information Science*, *5*(6), p73. https://doi.org/10.5539/cis.v5n6p73

ARC Tier Comparison—TPGi. (o. J.). Abgerufen 2. August 2023, von https://www.tpgi.com/arc-platform/arc-tier-feature-and-pricing-comparison/

ARC Toolkit—Page-Level Testing. (o. J.). TPGi. Abgerufen 22. August 2023, von https://www.tpgi.com/arc-platform/arc-toolkit/

Behinderungen in der EU: Fakten und Zahlen. (2022, Juli 4). https://www.consilium.europa.eu/de/infographics/disability-eu-facts-figures/

BGG - Gesetz zur Gleichstellung von Menschen mit Behinderungen, (2002). https://www.gesetze-im-internet.de/bgg/BJNR146800002.html

BITV 2.0—Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz. (o. J.). Abgerufen 10. August 2023, von https://www.gesetze-im-internet.de/bitv_2_0/BJNR184300011.html

BITV-Test, B. I. K. (o. J.). *BIK BITV-Test | BITV-Test*. Abgerufen 10. August 2023, von https://www.bitvtest.de/bitv_test.html

Borgstedt, Silke & Heide Möller-Slawinski. (2019). *Digitale Teilhabe von Menschen mit Behinderung*.

Burkard, A., Zimmermann, G., & Schwarzer, B. (2021). Monitoring Systems for Checking Websites on Accessibility. *Frontiers in Computer Science*, *3*, 628770. https://doi.org/10.3389/fcomp.2021.628770

Creative Commons—Attribution-NonCommercial 4.0 International—CC BY-NC 4.0. (o. J.). Abgerufen 25. Juli 2023, von https://creativecommons.org/licenses/by-nc/4.0/

Der „European Accessibility Act“. (o. J.). Bundesfachstelle Barrierefreiheit. Abgerufen 10. August 2023, von https://www.bundesfachstelle-barrierefreiheit.de/DE/Fachwissen/Produkte-und-Dienstleistungen/European-Accessibility-Act/european-accessibility-act_node.html

DIN EN 301549 Barrierefreiheitsanforderungen für IKT-Produkte und -Dienstleistungen. (o. J.).

European accessibility act. (o. J.). Abgerufen 2. August 2023, von https://ec.europa.eu/social/main.jsp?catId=1202&intPageId=5581&langId=en

F24: Failure of Success Criterion 1.4.3, 1.4.6 and 1.4.8 due to specifying foreground colors without specifying background colors or vice versa. (o. J.). Abgerufen 24. Juli 2023, von https://www.w3.org/WAI/WCAG21/Techniques/failures/F24.html

F33: Failure of Success Criterion 1.3.1 and 1.3.2 due to using white space characters to create multiple columns in plain text content. (o. J.). Abgerufen 14. August 2023, von https://www.w3.org/WAI/WCAG21/Techniques/failures/F33.html

F34: Failure of Success Criterion 1.3.1 and 1.3.2 due to using white space characters to format tables in plain text content. (o. J.). Abgerufen 14. August 2023, von https://www.w3.org/WAI/WCAG21/Techniques/failures/F34.html

F48: Failure of Success Criterion 1.3.1 due to using the pre element to markup tabular information. (o. J.). Abgerufen 14. August 2023, von https://www.w3.org/WAI/WCAG21/Techniques/failures/F48.html

F83: Failure of Success Criterion 1.4.3 and 1.4.6 due to using background images that do not provide sufficient contrast with foreground text (or images of text). (o. J.). Abgerufen 24. Juli 2023, von https://www.w3.org/WAI/WCAG21/Techniques/failures/F83.html

F91: Failure of Success Criterion 1.3.1 for not correctly marking up table headers. (o. J.). Abgerufen 14. August 2023, von https://www.w3.org/WAI/WCAG21/Techniques/failures/F91.html

F92: Failure of Success Criterion 1.3.1 due to the use of role presentation on content which conveys semantic information. (o. J.). Abgerufen 14. August 2023, von https://www.w3.org/WAI/WCAG21/Techniques/failures/F92.html

HTML CodeSniffer. (o. J.). Abgerufen 2. August 2023, von http://squizlabs.github.io/HTML_CodeSniffer/

Initiative (WAI), W. W. A. (o. J.). *Selecting Web Accessibility Evaluation Tools*. Web Accessibility Initiative (WAI). Abgerufen 22. Juni 2023, von https://www.w3.org/WAI/test-evaluate/tools/selecting/

Introduction to Siteimprove's crawler: An FAQ. (o. J.). Siteimprove. Abgerufen 8. August 2023, von https://help.siteimprove.com/support/solutions/articles/80000448489-siteimprove-s-crawler-frequently-asked-questions

Majuri, J., Koivisto, J., & Hamari, J. (2018). *Gamification of education and learning: A*

review of empirical literature.

Nielsen, J., & Mack, R. L. (Hrsg.). (1994). *Usability inspection methods*. Wiley.

Nielsen, Jakob. (o. J.). *Severity Ratings for Usability Problems: Article by Jakob Nielsen*. Nielsen Norman Group. Abgerufen 26. Juni 2023, von https://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/

Pa11y. (o. J.). Abgerufen 1. August 2023, von https://pa11y.org/

Pa11y Dashboard. (2023). [JavaScript]. Pa11y. https://github.com/pa11y/pa11y-dashboard (Original work published 2013)

Pa11y on npm. (2022, April 11). Npm. https://www.npmjs.com/package/pa11y

Parvin, P., Palumbo, V., Manca, M., & Paternò, F. (2021). The transparency of automatic accessibility evaluation tools. *Proceedings of the 18th International Web for All Conference*, 1–5. https://doi.org/10.1145/3430263.3452436

Preise: Flexibel genug für Ihre speziellen Anforderungen. (o. J.). Siteimprove. Abgerufen 1. August 2023, von https://www.siteimprove.com/de/pricing/

Prüfschritt 9.1.3.1e Datentabellen richtig aufgebaut | BIK BITV-Test. (o. J.). Abgerufen 14. August 2023, von https://ergebnis.bitvtest.de/pruefschritt/bitv-20-web/bitv-20-web-9-1-3-1e-datentabellen-richtig-aufgebaut

Richtlinie (EU) 2016/ 2102 des Europäischen Parlaments und des Rates über den barrierefreien Zugang zu den Websites und mobilen Anwendungen öffentlicher Stellen, (2016).

Richtlinien für barrierefreie Webinhalte WCAG 2.1. (o. J.). Abgerufen 27. Juli 2023, von https://outline-rocks.github.io/wcag/translations/WCAG21-de/

Singh, N. (2023). CI/CD Pipeline for Web Applications. *International Journal for Research in Applied Science and Engineering Technology*, *11*(5), 5218–5226. https://doi.org/10.22214/ijraset.2023.52867

Types of documents W3C publishes. (o. J.). W3C. Abgerufen 1. August 2023, von https://www.w3.org/standards/types/

Understanding Success Criterion 1.3.1: Info and Relationships. (o. J.). Abgerufen 25. Juli 2023, von https://www.w3.org/WAI/WCAG21/Understanding/info-and-relationships.html

U.S. Access Board—Revised 508 Standards and 255 Guidelines. (o. J.). Abgerufen 1. August 2023, von https://www.access-board.gov/ict/

Vigo, M., & Brajnik, G. (2011). Automatic web accessibility metrics: Where we are and

where we can go. *Interacting with Computers*, *23*(2), 137–155. https://doi.org/10.1016/j.intcom.2011.01.001

Vigo, M., Brown, J., & Conway, V. (2013). Benchmarking web accessibility evaluation tools: Measuring the harm of sole reliance on automated tests. *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, 1–10. https://doi.org/10.1145/2461121.2461124

WCAG 2.1 Standard: Summary—HTML CodeSniffer. (o. J.). Abgerufen 2. August 2023, von https://squizlabs.github.io/HTML_CodeSniffer/Standards/WCAG2/

Web Content Accessibility Guidelines (WCAG) 2.1. (o. J.). Abgerufen 25. Juli 2023, von https://www.w3.org/TR/WCAG21/

Web Content Accessibility Guidelines (WCAG) 2.2. (o. J.). Abgerufen 1. August 2023, von https://www.w3.org/TR/WCAG22/

WebAIM: Contrast Checker. (o. J.). Abgerufen 21. August 2023, von https://webaim.org/resources/contrastchecker/?fcolor=000000&bcolor=7A2DB9