



Entwicklungsprojekt

Wintersemester 23/24

Präsentiert von: Robin Kraft, Clemens Brachtendorf und Karam Hayani

Inhaltsverzeichnis

1 Audit 1

Problemstellung, Zielsetzung,
Artefakte, PoC etc.

2 Audit 2

Erweiterung Projektrisiken,
Überarbeitung PoC's, etc.

3 Audit 3

First Rapid Prototype

4 Audit 4

Problemstellung und Herleitung der Zielsetzung

Die Problemstellung unseres Projekts besteht darin, dass trotz des gesellschaftlichen Engagements für barrierefreies Webdesign die Barrierefreiheit im Internet vernachlässigt wird. Viele Webseiten sind nicht zugänglich für Menschen mit Behinderungen. Dies schränkt die Nutzung und den Zugang zu digitalen Ressourcen erheblich ein. Unsere Zielsetzung leitet sich daraus ab: Wir möchten ein System entwickeln, das Barrieren in Webdesigns erkennt und konkrete Verbesserungsvorschläge bietet. Damit wollen wir die digitale Inklusion fördern, indem wir Webdesignern und Entwicklern ein effizientes Werkzeug zur Verfügung stellen, um barrierefreie Websites zu erstellen und die Einhaltung von Barrierefreiheitsstandards sicherzustellen.



Zielsetzung sowie Begründung des Vorgehens zur Erreichung dieser

Unsere Zielsetzungen sind:

Entwicklung

Die Entwicklung eines Systems zur Erkennung von Barrieren in Webdesigns.

Integration

Die Integration dieses Systems in Design- und Entwicklungsumgebungen wie Figma und Visual Studio Code.

Förderung

Die Förderung der Einhaltung von Barrierefreiheitsstandards

Diese Ziele verfolgen wir, um die digitale Inklusion zu verbessern und die Schaffung barrierefreier Websites zu erleichtern. Die Begründung für unser Vorgehen liegt in der Notwendigkeit, bestehende Hürden für Menschen mit Behinderungen im Internet zu überwinden und die Barrierefreiheit auf Websites zu gewährleisten. Dies trägt zur sozialen Integration und zur Schaffung eines inklusiveren digitalen Raums bei.

Alleinstellungsmerkmal

Unser System setzt neue Maßstäbe in der Automatisierung von Barrierefreiheits-Audits, indem es eine deutlich umfassendere Prüfabdeckung bietet. Während herkömmliche automatisierte Lösungen im Durchschnitt etwa 30% der WCAG-Kriterien abdecken, verdoppelt unser innovatives System diese Quote nahezu. Mit der Fähigkeit, mindestens 60% der erforderlichen Konformitätsstandards zu überprüfen, liefert es nicht nur präzisere und relevantere Ergebnisse, sondern setzt auch einen neuen Industriestandard für die automatische Überprüfung der Barrierefreiheit.





Erste Risiken 1

Technologische Risiken:

Komplexität der WCAG-Richtlinien

Das System könnte aufgrund der komplexen und vielschichtigen WCAG-Richtlinien Schwierigkeiten haben, alle Anforderungen abzudecken.

Verwendete und alternative Ressourcen

Schnelle Veränderungen in Webtechnologien können dazu führen, dass das System ständig aktualisiert werden muss, um relevant zu bleiben.

Fehleranfälligkeit

Automatisierte Systeme können bestimmte Arten von Barrierefreiheitsproblemen übersehen, die menschliche Prüfer erkennen würden



Erste Risiken 2

Markt Risiken:

Akzeptanz am Markt

Es besteht das Risiko, dass der Markt automatisierte Lösungen nicht annimmt oder dass die Kunden die Vorteile nicht erkennen

Wettbewerb

Neue oder etablierte Konkurrenten könnten ähnliche oder überlegene Lösungen entwickeln und auf den Markt bringen.

Gesetzliche Änderungen

Neue Vorschriften könnten die Geschäftsgrundlage verändern und zusätzliche Anpassungen erfordern.



Erste Risiken 3

Betriebliche Risiken:

Skalierbarkeit

Das System muss in der Lage sein, mit einem wachsenden Kundenstamm und steigenden Datenvolumen zu skalieren.

Ressourcenverfügbarkeit

Es könnte an qualifiziertem Personal fehlen, um das System zu warten und weiterzuentwickeln.

Datenschutz und Sicherheit

Sicherheitslücken könnten die Integrität des Systems gefährden und rechtliche sowie imagebezogene Konsequenzen haben.

Spezifikation des ersten technischen/architekturellen Proof-of-Concepts

Die Spezifikation des ersten technischen/architektonischen Proof of Concept (PoC) umfasst die Entwicklung eines Prototyps, der die Kernfunktionalität unseres Systems zur Barriereerkennung in Webdesigns demonstriert. Dieser PoC wird die Fähigkeit des Systems zur Identifizierung von Barriereelementen auf Webseiten zeigen. Wir werden ein Proof-of-Concept-Modell entwickeln, das diese Fähigkeiten veranschaulicht und als Grundlage für die weitere Entwicklung dient. Die Architektur des PoC wird modulare Komponenten umfassen, darunter eine Schnittstelle zur Website-Analyse, Algorithmen zur Barriereerkennung und eine Benutzeroberfläche zur Darstellung der Ergebnisse.



Stakeholder	Anforderung	Erweiterung
Webentwickler	Zugang zu fortschrittlichen Entwicklungswerkzeugen, klare Arbeitsanweisungen, angemessene Schulungen	Karriereentwicklungsmöglichkeiten, regelmäßige Feedback-Sitzungen, Work-Life-Balance-Initiativen
Webagenturen	Integration mit dem System, technische Unterstützung, gemeinsame Marketinginitiativen.	Gemeinsame Produktentwicklung, regelmäßige Strategie-Meetings.
Staatliche Einrichtungen	Einhaltung der WCAG-Richtlinien und Datenschutzbestimmungen.	Proaktive Zusammenarbeit bei der Gestaltung zukünftiger Richtlinien, Bereitstellung von Daten und Berichten.
Unternehmen & Privatkunden (Kunden)	Effiziente WCAG-Audits, detaillierte Berichte, Empfehlungen zur Behebung von Barrierefreiheitsproblemen, Unterstützung bei der Implementierung von Lösungen.	Personalisierte Zugänglichkeitsberatung, kontinuierliche Überwachung und Support.
Endnutzer der Webseiten	Vollständige Zugänglichkeit und Usability der Webseiten, Unterstützung verschiedener Hilfstechnologien.	Feedback-Möglichkeiten zur Website-Zugänglichkeit, benutzerfreundliche Anleitungen zur Nutzung von Zugänglichkeitsfunktionen.



Stakeholder

Begründung der Überarbeitung der Stakeholder Tabelle:

Die Erweiterung der Stakeholder erfolgte aufgrund der Vielfalt von Endnutzern und deren spezifischen Anforderungen an die Barrierefreiheit. Durch die Identifizierung und Spezifizierung verschiedener Endnutzergruppen konnten wir sicherstellen, dass das Barrierefreiheitssystem eine breite Palette von Bedürfnissen abdeckt. Jede spezifizierte Endnutzergruppe repräsentiert einzigartige Herausforderungen und Anforderungen, die es zu berücksichtigen gilt, um eine umfassende und effektive Barrierefreiheitslösung zu entwickeln. Diese Diversifizierung der Stakeholder ermöglicht es uns, ein System zu schaffen, das nicht nur gesetzlichen Anforderungen entspricht, sondern auch eine inklusive und benutzerzentrierte Webumgebung schafft.

Bezeichnung	Einz.	Orga.	Bezug zum System	Objektbereich	Erwartung
Webentwickler	X		Interesse	System/Entwicklungs- und Implementierungstools	Effiziente Integration und Implementierung barrierefreier Technologien.
Webagentur		X	Interesse	System/Projektmanagement und Kundenberatung	Ressourcen und Beratung für erfolgreiche Integration barrierefreier Lösungen.
Staatliche Einrichtungen		X	Anspruch	System/Gesetzeskonformität	Gewährleistung der Einhaltung gesetzlicher Vorschriften zur Barrierefreiheit.
Unternehmen und Privatkunden		X	Interesse	System/Implementierung	Integration barrierefreier Elemente und Schulung für Mitarbeiter.
Blinde Menschen	X		Anspruch	System/Navigation und Inhaltzugänglichkeit	Navigieren und Inhalte mit Screenreadern zugänglich machen.
Gehörlose Menschen	X		Anspruch	System/Multimediale Inhalte	Multimediale Inhalte untertiteln, transkribieren, Gebärdensprachoptionen bieten.
Motorisch-beinträchtigte	x		Anspruch	System/Interaktionsfunktionen	Tastaturzugänglichkeit, leichte Interaktion.

Bezeichnung	Einz.	Orga.	Bezug zum System	Objektbereich	Erwartung
Kognitiv-beinträchtigte	X		Anspruch	System/Inhaltsverständlichkeit	Klare und leicht verständliche Inhalte.
Senioren	X		Interesse	System/Navigation s- und Lesbarkeitsfunktionen	Einfache Navigation, gut lesbare Schriftarten.
Farbblinde Menschen	X		Interesse	System/Farbgestaltung	Gut differenzierbare Farben, klare Unterscheidung.
Sprach-beeinträchtigte	X		Interesse	System/Kommunikationsfunktionen	Unterstützung für alternative Kommunikationsformen.
Hör-beinträchtigte	X		Interesse	System/Auditivdarstellung	Bereitstellung barrierefreier auditiver Darstellungen.
Neuro-beeinträchtigte	X		Interesse	System/Visuelle Darstellungen und Animationen	Reduzierte visuelle Effekte für Anfallsprävention.

Anforderungen

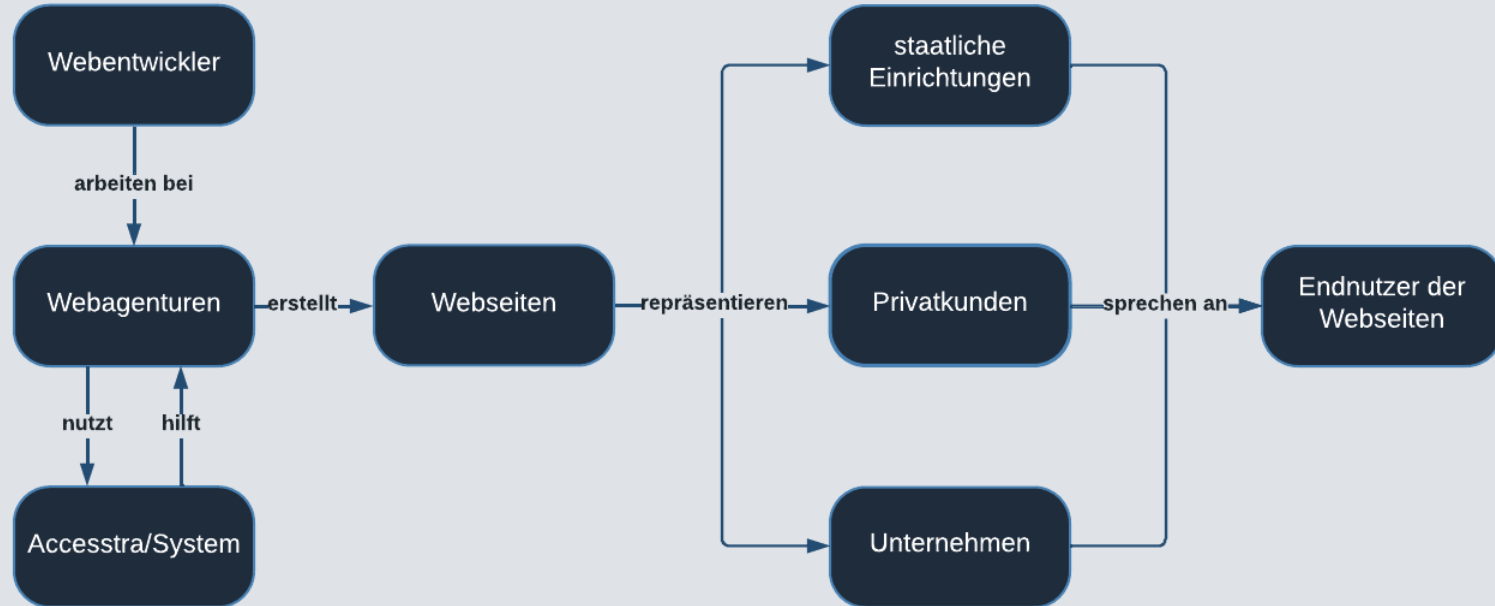
- Das System soll es Webentwicklern erleichtern, Barrieren in ihrem Code aufzudecken und zu beheben.
- Das System sollte bis zum Start der WCAG 3.0 Kriterien 2025 dazu fähig sein, diesen Standards gerecht zu werden. Das System muss dazu im Stande sein, den HTML-Code von Webseiten zu analysieren und nach WCAG Standards Fehler aufzuzeigen, die eine potenzielle Barriere für Nutzer*innen darstellen könnten.
- Das System wird in späteren Iterationen dazu in der Lage sein, alternative Texte für Bilder zu generieren, um diese entweder direkt in den Alt-Tag des Bildes im Code einzufügen falls noch keines vorhanden war oder zu prüfen, ob das aktuelle auch das beschreibt was auf dem Bild zu sehen ist.

Anforderungen

- Personen mit verschiedensten Beeinträchtigungen werden zukünftig im Prozess der Entwicklung des Systems in Testphasen mit einbezogen, um Barrieren zu erkennen und zu beheben.
- Ein*e Webentwickler*in muss das System in der eigenen Entwicklungsumgebung benutzen können, um direkt im Source Code der Webseiten durch das System Änderungen durchführen lassen kann.
- Ein*e Webentwickler*in wird die Änderungen, die das System durchführen wird, zukünftig in einer Webapplikation im Voraus anschauen können.



Domänenmodell



Audit 2



Weiterentwickelte Projektrisiken 1

Architekturelle Risiken:

Kompatibilitätsprobleme	Schwierigkeiten bei der Integration des Systems in verschiedene Design- und Entwicklungsplattformen.
Modularität und Erweiterbarkeit	Herausforderungen bei der Entwicklung eines modularen Systems, das leicht erweitert und aktualisiert werden kann
Leistung und Effizienz	Risiken bezüglich der Systemleistung, insbesondere bei der Analyse komplexer Webseiten.

Weiterentwickelte Projektrisiken 2

Kommunikation / Interaktion von Anwendungsobjekten Risiken:

API-Abhängigkeiten

Risiken durch Abhängigkeit von externen APIs, die sich ändern oder unzuverlässig sein können.

Datenflussmanagement

Schwierigkeiten bei der Handhabung und Verarbeitung großer Datenmengen.

Synchronisierung von Komponenten

Herausforderungen bei der Gewährleistung einer effektiven Kommunikation zwischen verschiedenen Systemkomponenten.

Weiterentwickelte Projektrisiken 3

Technische Risiken:

Technologische Abhängigkeit

Risiken durch Abhängigkeit von spezifischen Technologien, die veraltet sein könnten.

Sicherheitslücken

Potenzielle Sicherheitsrisiken in der verwendeten Software und bei der Datenübertragung.

Kompatibilität mit verschiedenen Browsern

Herausforderungen bei der Sicherstellung der Funktionsfähigkeit des Systems über verschiedene Browser hinweg.

Weiterentwickelte Projektrisiken 4

Kompetenzorientierte Risiken:

Fachwissen des Teams

Risiken durch fehlendes spezifisches Fachwissen im Team.

Schulungs- und Weiterbildungsbedarf

Notwendigkeit kontinuierlicher Schulung des Teams in neuen Technologien und Methoden.

Fluktuation im Team

Risiko des Wissensverlustes durch Mitarbeiterwechsel.

Proof-of-Concept

Komplexität der WCAG-Richtlinien:

Vorhaben	Entwicklung eines Prototyps, der die Analyse einer Reihe von WCAG-Kriterien demonstriert.
Exit-Kriterien	Erfolgreiche Analyse und Interpretation verschiedener WCAG-Kriterien.
Fail-Kriterien	Unfähigkeit, die Kriterien genau zu analysieren oder zu interpretieren.
Fallbacks	Vereinfachung des Analyseprozesses oder Beschränkung auf Kernkriterien.
Begründung	Der PoC zur WCAG-Analyse wurde gewählt, da dies eine Kernherausforderung des Projekts darstellt. Dieser PoC deckt das Risiko ab, indem er die Fähigkeit des Systems demonstriert, komplexe Richtlinien zu interpretieren.
Bewertung	Sehr gut abgedeckt. Der PoC konzentriert sich direkt auf das Kernrisiko der komplexen WCAG-Anforderungen.

Proof-of-Concept

Technologiewandel:

Vorhaben	Implementierung eines Update-Mechanismus für das System.
Exit-Kriterien	Erfolgreiche Aktualisierung des Systems für neue Webtechnologien.
Fail-Kriterien	Unfähigkeit, das System effizient zu aktualisieren.
Fallbacks	Entwicklung eines flexibleren Systemdesigns.
Begründung	Der Update-Mechanismus-PoC ist entscheidend, um sicherzustellen, dass das System mit der sich schnell entwickelnden Webtechnologie Schritt hält. Dieser PoC adressiert direkt das Risiko veralteter Technologie.
Bewertung	Gut abgedeckt. Der PoC adressiert das Risiko, erfordert aber kontinuierliche Überprüfung und Anpassung.

Proof-of-Concept

Fehleranfälligkeit:

Vorhaben	Entwicklung eines Feedback-Mechanismus für die Fehlererkennung.
Exit-Kriterien	Verbesserung der Fehlererkennungsrate durch Nutzerfeedback.
Fail-Kriterien	Keine signifikante Verbesserung der Fehlererkennung.
Fallbacks	Implementierung zusätzlicher manueller Überprüfungsprozesse.
Begründung	Der Feedback-Mechanismus-PoC ist wichtig, um die Genauigkeit des Systems zu verbessern und Nutzerfeedback zu integrieren. Er deckt das Risiko des Übersehens von Barrierefreiheitsproblemen ab.
Bewertung	Moderat abgedeckt. Während der PoC hilft, die Genauigkeit zu verbessern, ist eine vollständige Beseitigung von Fehlern schwierig.

Audit

Ξ



First rapid Prototype

Unseren ersten Prototypen haben wir als Node.js-Anwendung, die die HTML-Seite einer Webseite abrufen, den HTML-Code mit Cheerio analysiert und die Alt-Texte der Bilder extrahiert implementiert.



JS parseWebsite.js X

JS parseWebsite.js > ...

```
1  const axios = require('axios');
2  const cheerio = require('cheerio');
3
4  // Funktion zum Abrufen des HTML-Codes einer Webseite
5  async function getHtml(url) {
6      try {
7          const response = await axios.get(url);
8          return response.data;
9      } catch (error) {
10         console.error('Fehler beim Abrufen der Webseite:', error.message);
11         return null;
12     }
13 }
```

```
15 // Funktion zum Extrahieren der Alt-Texte aus dem HTML-Code mit Cheerio
16 function extractAltTexts(htmlCode) {
17     const webpage = cheerio.load(htmlCode);
18     const imgTags = webpage('img');
19
20     if (imgTags.length > 0) {
21         imgTags.each((index, element) => {
22             const altText = webpage(element).attr('alt');
23             if (altText) {
24                 console.log(`Bild ${index + 1} hat Alt-Tag: ${altText}`);
25             } else {
26                 console.log(`Bild ${index + 1} hat keinen Alt-Tag.`);
27             }
28         });
29     } else {
30         console.log('Keine Bilder gefunden.');
```

```
34 // URL der Webseite, deren HTML analysiert werden soll
35 const websiteUrl = 'https://www.kinopolis.de/ko';
36
37 // HTML-Code der Webseite abrufen
38 getHtml(websiteUrl).then(htmlCode => {
39     if (htmlCode) {
40         // Alt-Tags aus dem HTML-Code extrahieren und ausgeben
41         extractAltTexts(htmlCode);
42     }
43 });
```

Danke!

Gibt es noch Fragen?



Bitte lösche diese Folie nicht, es sei denn du bist ein Premium Nutzer

CREDITS: Diese Präsentationsvorlage wurde von Slidesgo erstellt, inklusive Icons von Flaticon und Infografiken & Bilder von Freepik