



On-Site-Search-Integration-Framework und Anwendung im Swisscom-Design-System

Bachelorarbeit

Frühjahrssemester 2023

Autoren: Manuel Weber

Rolf Oberhänsli

Betreuer: Prof. Dr. Markus Stolze

Projektpartner: Swisscom

Förrlibuckstrasse 181

8005 Zürich

Experte: Markus Flückiger

Gegenleser: Ivan Bütler

Inhaltsverzeichnis

I	Abstract, Management Summary & Danksagung	1
II	Einleitung	10
1	Einleitung	11
1.1	Dokumentstruktur	11
1.2	Ausgangslage	11
1.3	Problemdefinition	12
1.4	Vision	12
1.5	Herangehensweise	12
III	Produkt Dokumentation	13
2	Domänenanalyse	14
2.1	Swisscom Digital Experience (SDX)	14
2.1.1	Domänenmodell	15
2.2	Hypothesen	16
2.2.1	Benutzergruppen	16
2.2.2	Proto-Personas	18
2.2.3	Proto-Journeys	21
3	Theoretischer Hintergrund	23
3.1	Recherche Suche	23
3.1.1	Gründe für eine Suche	23
3.1.2	Sucharchitekturen	24
3.1.3	Suchmethoden	26
3.1.4	Suchalgorithmen	27
3.1.5	Produkte	28
3.2	Recherche Parser	34
3.2.1	SEO	34
3.2.2	Semantic HTML	34

3.2.3	Synonyme	35
3.3	Technologien	35
3.4	Deployment	35
3.4.1	npm	35
4	Anforderungen	37
4.1	Akteure	38
4.2	Funktionale Anforderungen	39
4.2.1	Use Case Diagramm	39
4.2.2	Use Cases	40
4.3	Nicht-Funktionale Anforderungen	53
4.3.1	Analyse aller ISO25010-Kategorien und -Unterkategorien	61
4.3.2	Analyse der OWASP Top 10	63
4.4	Adressierung nicht-funktionaler Anforderungen	64
4.4.1	Adressierung der NFR per Alpha-Release	64
4.4.2	Adressierung der NFR per Beta-Release	66
5	Lösungsdesign	68
5.1	Konzept	68
5.1.1	Phase 1: Datenextraktion, -anreicherung und -aufbereitung	69
5.1.2	Phase 2: Effektive Suche	71
5.1.3	Deployment	72
5.1.4	Variationen	72
5.2	Architektur	76
5.2.1	Architektur-Constraints	77
5.2.2	Umsetzungsvariante: Clientseitige Architektur	77
5.2.3	Design-Entscheide	78
5.2.4	Technische Entscheide	80
5.2.5	Design Patterns	82
5.3	Evaluation Such-Produkte	83
5.3.1	Produktidentifikation	83
5.3.2	Evaluationskriterien	86
5.3.3	Nutzwertanalyse	87
5.3.4	Entscheid	88
5.4	Wireframes	90
5.4.1	Sucheingabe mit Autocomplete-Vorschlägen	90
5.4.2	Kategorisierte Suchresultate	90
5.4.3	Suchfilter	91
5.4.4	Suchhistorie	91
5.4.5	Suchvorschläge	92
5.4.6	Hover-Informationen	92

5.4.7	Mega-Dropdown	93
5.4.8	Komponenten-Übersicht	94
6	Qualitätsmassnahmen	96
6.1	Coding Guidelines	96
6.1.1	Clean Code	96
6.1.2	Tools	96
6.1.3	Error Handling	97
6.1.4	Logging der Integrationsschritte	97
6.2	Versionskontrolle	98
6.2.1	Struktur Repository	98
6.2.2	Pre-Commit Hook	98
6.2.3	CI/CD	98
6.2.4	Workflow für die Arbeit am Source-Code	100
6.3	Code-Reviews	102
6.4	Definition of Done	102
6.5	Metriken und Code Analysis	102
6.5.1	Code-Coverage	102
6.6	Testkonzept	104
6.6.1	Unit-Tests	104
6.6.2	Performance-Tests	104
6.6.3	End-2-End-Tests	104
6.6.4	Integrations-Tests	104
6.6.5	Last-Tests	105
6.6.6	Usability-Tests	105
6.6.7	System-Tests	112
IV	Projekt Dokumentation	114
7	Projektplan	115
7.1	Entwicklungsprozess	115
7.1.1	Phasen	115
7.2	Meilensteine	116
7.3	Zeitplan	119
7.4	Meetings	120
7.4.1	Team-interne Meetings	120
7.4.2	Reviews mit Betreuer	120
7.4.3	Besprechungen mit Auftraggeber	121
7.5	Rollen	122
7.5.1	Details zu den definierten Rollen	122
7.6	Ansprechpersonen	123

7.7	Risiko-Management	124
7.7.1	Risikomatrix	124
7.7.2	Risiken	124
7.7.3	Risikominimierung und -prävention	127
7.7.4	Risikoentwicklung	128
7.8	Issue-Management	128
7.9	Versionskontrolle	129
7.10	Zeiterfassung	129
V	Ergebnisse	130
8	Resultate	131
8.1	Verifikation des Resultats	131
8.1.1	Grösse der zusätzlich benötigten Ressourcen einer integrierten Suche	131
8.1.2	Lighthouse: Analyse einer mit einer Suche angereicherten Website	132
8.1.3	Performance einer Website mit einer integrierten Suche	134
8.1.4	Performance der integrierten Suche	135
8.2	Verifikation funktionaler und nicht-funktionaler Anforderungen	136
8.2.1	Funktionale Anforderungen	136
8.2.2	Nicht-funktionale Anforderungen	138
8.3	Repository-Statistiken	141
8.3.1	Search-Integrator & Anwendungsbeispiel SDX-Suche	141
8.3.2	BA-Dokumentation	141
8.4	Zeiterfassungsreport	142
9	Fazit	146
9.1	Retrospektive	146
9.2	Ausblick	147
	Literaturverzeichnis	149
	Glossar	153
	Abbildungsverzeichnis	154
	Tabellenverzeichnis	156
VI	Anhang	159
10	Anhang	160
10.1	Test-Protokolle	161
10.1.1	System-Tests	161

10.1.2 Usability-Tests	163
10.2 Weitere Anhänge	170

Teil I

**Abstract, Management Summary &
Danksagung**

Abstract

Ausgangslage

Swisscom betreibt eine Website, genauer ein Design System, mit dem Namen 'Swisscom Digital Experience' oder kurz SDX. Es besteht aus einer Komponentenbibliothek, UX-Prinzipien, Designrichtlinien und Regeln, die eingehalten werden müssen, um den 'Swisscom-Look' zu erhalten. Zur Navigation auf der SDX-Website ist ein Menü auf der linken Seite vorhanden, welches eine Auflistung aller in SDX verfügbaren Webseiten enthält. Benutzer müssen somit gesuchte Inhalte der SDX-Website über das Menü identifizieren, wobei sie anhand der Begriffe im Menü entscheiden müssen, ob eine Webseite für das Erfüllen eines Suchbedürfnisses infrage kommt oder nicht. Dies ist vor allem für Benutzer, die neu mit SDX arbeiten, mit erhöhtem Zeitaufwand verbunden.

In dieser Bachelorarbeit wurde eine Such-Funktionalität für die SDX-Website konzipiert und umgesetzt. Die Suche soll dem Nutzer einen erhöhten Komfort bei der Suche nach Inhalten sowie bei der Navigation innerhalb von SDX bieten. Durch die Suche soll schnell auf alle erwarteten und zusätzlich relevanten Inhalte, die man mit der Navigation übersehen würde, zugegriffen werden können.

Vorgehen

In einem ersten Schritt wurde eine Basisanalyse zum Thema On-Site-Search sowie eine Anforderungsanalyse mit der Swisscom durchgeführt. Darauf aufbauend wurden Anforderungen definiert, Prototypen entwickelt und Wireframes ausgearbeitet. Anschliessend wurde ein Konzept zur Umsetzung der Suche, welches die Bedürfnisse der Swisscom erfüllt, erstellt. Das Konzept basiert auf einer clientseitigen Architektur, was bedeutet, dass die Such-Funktionalität im Browser des Benutzers ausgeführt wird. Während der Implementation wurde die Lösung durch automatisierte Tests sowie Usability-Tests ausgiebig getestet. Das Resultat dieser Bachelorarbeit ist ein generisches Such-Integration-Framework, welches die Bereitstellung einer Suche auf einer Website einfach und schnell ermöglicht. Abschliessend wurde mithilfe des Frameworks eine Suche in die SDX-Website integriert.

Fazit

Usability Tests sowie Gespräche mit der Swisscom haben gezeigt, dass es möglich ist, mit dem entwickelten Such-Integration-Framework eine Suche für die SDX-Website zur Verfügung zu stellen, die den benötigten Zeitaufwand für das Befriedigen von Suchbedürfnissen signifikant reduziert. Es ist möglich mit geringem

Aufwand eine Suche in eine bestehende Website einzubetten. Für ein optimales Ausnutzen der entwickelten Funktionalität wurden Empfehlungen bezüglich Optimierungen an der Struktur der SDX-Website an die Swisscom übergeben, welche die Website nicht nur für die integrierte On-Site-Search, sondern auch für Suchmaschinen besser durchsuchbar gestaltet. Während des Projektverlaufs wurden zusätzliche Features für das Such-Integration-Framework spezifiziert, welche im Anschluss an die Bachelorarbeit umgesetzt werden können. Das Such-Integration-Framework wurde über NPM publiziert und kann somit auch in anderen Websites Anwendung finden. Um die Integration und Individualisierung einer Suche so einfach wie möglich zu gestalten, wurde nebst einer Projektdokumentation auch eine ausführliche Entwicklerdokumentation verfasst.

Management Summary

Ausgangslage

Die Swisscom betreibt das Design-System 'Swisscom Digital Experience', kurz SDX, als Website. Es besteht aus einer Komponentenbibliothek, UX-Prinzipien, Designrichtlinien und Regeln, die eingehalten werden müssen, um den 'Swisscom-Look' zu erhalten.

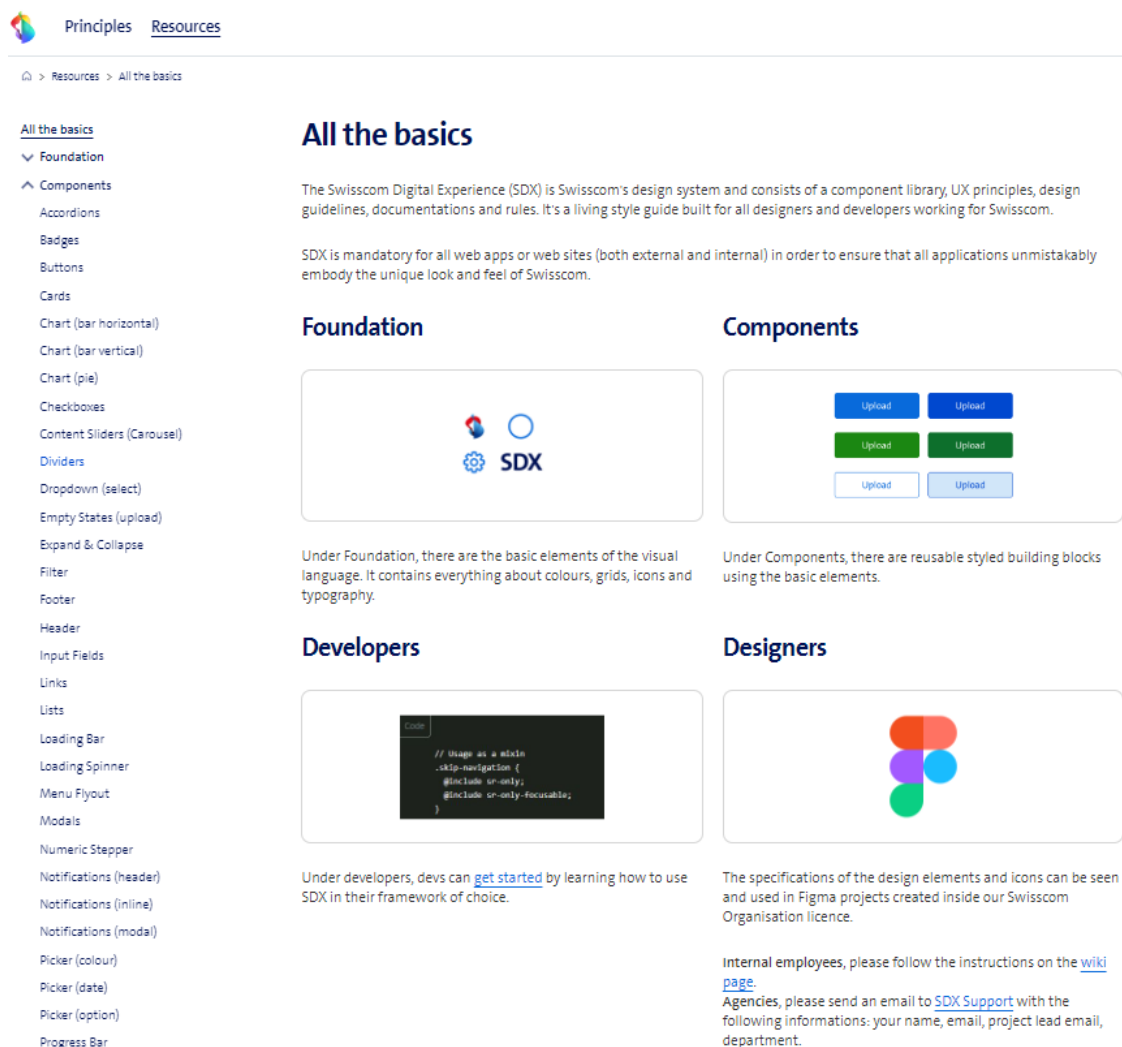


Abbildung 1: Landing-Page der SDX-Website inkl. Menü

Abbildung 1 zeigt die Landing-Page der SDX-Website. Zur Navigation auf der SDX-Website ist ein Menü auf der linken Seite vorhanden, welches eine Auflistung aller in SDX verfügbaren Webseiten enthält. Die Auffindbarkeit der einzelnen Webseiten sowie der darauf enthaltenen Informationen ist erschwert, da die Menüleiste nur beschränkten Aufschluss über den Inhalt der einzelnen Webseiten gibt. Zudem ist anhand des Menüs schwer abzuschätzen, ob man alle relevanten Daten innerhalb der Website gefunden hat. Benutzer sind gezwungen den Inhalt der Website gut zu kennen, um sich über das Menü zielorientiert und umfänglich bewegen zu können und schnell an gesuchte Informationen zu gelangen. Dies stellt insbesondere für neue Benutzer von SDX eine Hürde dar.

Um dieses Problem zu lösen, wurde eine Bachelorarbeit zur Verbesserung der Bedienbarkeit der Website respektive der Auffindbarkeit von Informationen ausgeschrieben. Bezüglich der Umsetzung bestand die Anforderung, dass, wenn möglich, keine zusätzliche Infrastruktur für den Betrieb der Suchfunktion nötig sein soll.

Vorgehen

Zum Start des Projekts wurde die Projektplanung erstellt. Der daraus resultierende Projektplan ist in Abbildung 2 dargestellt. Das Projekt wurde in die vier Phasen des Rational Unified Process (RUP) [RUP] unterteilt. Innerhalb der einzelnen Phasen wurde in Sprints gemäss Scrum [Scr] gearbeitet.

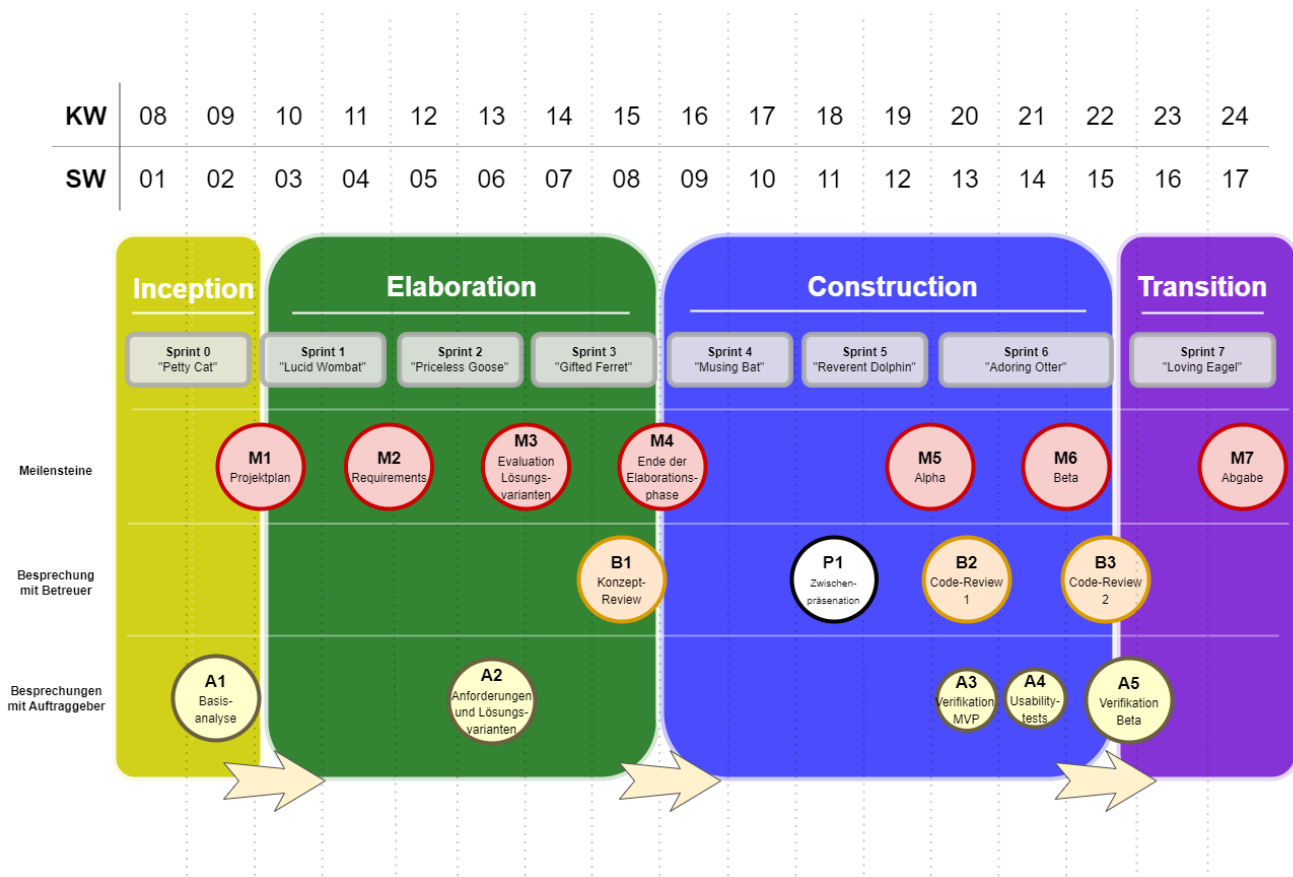


Abbildung 2: Projektplan

Im Anschluss an die Planung wurde eine Basisanalyse, gefolgt von einer Anforderungsanalyse mit dem Auftraggeber (Swisscom) durchgeführt. Die Analyse diente als Basis für die Evaluation der zu verwendeten Frameworks und Libraries im Projekt sowie der allgemeinen Informationsbeschaffung. Um die zur Verfügung stehenden Produkte bewerten zu können, wurden nebst einer Recherche auch Prototypen entwickelt, um die Bedienbarkeit, die Qualität und die Performance einzelner Lösungen sowie die Machbarkeit von Ansätzen zu prüfen. Das Resultat war ein Konzept zur Umsetzung eines Frameworks, welches alle Schritte zur Integration einer clientseitigen On-Site-Search in eine bestehende Website abdeckt. Das erarbeitete Konzept ist in Abbildung 3 visualisiert.

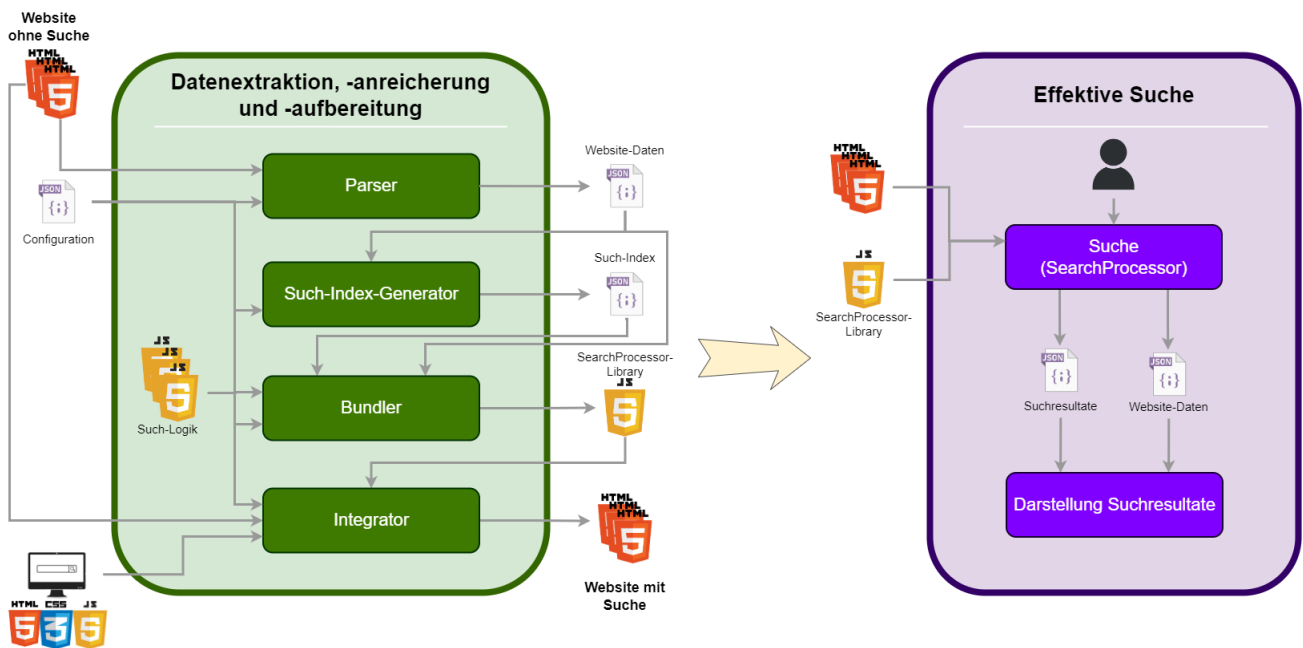


Abbildung 3: Konzept

In der Konstruktionsphase wurde das erarbeitete Konzept umgesetzt. Um die Anforderungen und die Korrektheit des Frameworks zu verifizieren, wurden automatisierte sowie manuelle Tests durchgeführt. Zusätzlich wurden zum Erhalt von Feedback von Benutzern Usability-Tests durchgeführt. Die entwickelte Lösung wurde über NPM verfügbar gemacht. Das Projekt wurde mit der Übergabe einer Such-Integration für die SDX-Website inklusive Empfehlungen bezüglich Optimierungen der Struktur der SDX-Website an die Swisscom abgeschlossen. Die Empfehlungen bezüglich Optimierungen der Webseiten ermöglichen nicht nur das Ausschöpfen des vollen Potenzials des Such-Integration-Frameworks, sondern optimieren die Webseiten auch für Suchmaschinen. Damit die Swisscom und andere zukünftige Nutzer das Framework möglichst einfach einsetzen können, wurde nebst einer Projektdokumentation eine ausführliche Entwicklerdokumentation erstellt, welche auf NPM eingesehen werden kann.

Ergebnis

Das Ergebnis dieser Bachelorarbeit ist ein Search-Integration-Framework, welches folgende Funktionalitäten zur Verfügung stellt:

- Extraktion aller relevanten Informationen einer Website
- Indexierung der extrahierten Informationen
- Anreicherung des Index durch zusätzliche Informationen
- Integration einer Suchkomponente in eine Website
- Suchlogik, für das Bereitstellen einer clientseitigen Suche
- Konfigurationsmöglichkeiten für eine individualisierte Anpassung der Suche auf die Bedürfnisse und das Design einer Website



Abbildung 4: QR-Code zum NPM-Paket 'Search-Integrator'

Das entwickelte Framework kann in beliebigen statischen Websites Anwendung finden. Die einzige Vorgabe zur Verwendung des Frameworks ist der Zugriff auf die einzelnen Webseiten in Form von HTML-Dateien. Die Lösung wurde für Websites mit einer relativ kleiner Datenmenge optimiert, aber grundsätzlich besteht keine Beschränkung bezüglich der Grösse einer Website. Es muss jedoch beachtet werden, dass je mehr Daten eine Website umfasst, desto grösser wird das Bundle, das an den Client ausgeliefert werden muss. Dies hat wiederum Auswirkungen auf die Time-to-Interactive [t_{ti}] der Website.

Das Framework wurde nach dem Grundsatz 'Convention over Configuration' [con] entwickelt. Es wurden für alle benötigten Konfigurationen sinnvolle Default-Werte definiert. Diese werden angewendet, sofern der Benutzer keine spezifischen Konfigurationen vornimmt. Dies erhöht die Benutzerfreundlichkeit und bietet einen einfachen Einstieg in die Anwendung des Frameworks. Die Veränderung von Daten auf einer Website stellt kein Problem dar, da die Anwendung des Frameworks in eine CI/CD-Pipeline integriert werden kann und somit bei jedem Release die aktuellsten Daten für die Suche berücksichtigt werden. Die Lösung wurde mit automatisierten Tests getestet und mit dem Auftraggeber in Form von Usability-Tests verifiziert.

Ausblick

Nach der Übergabe des Projekts an den Auftraggeber wird die Suche in die produktive SDX-Website integriert werden. Die dazu benötigten Informationen wurden anhand eines Auftraggeber-spezifischen Anwendungsbeispiels sowie einer Entwicklerdokumentation festgehalten und an den Auftraggeber übergeben. Auf Wunsch des Auftraggebers wird die Integration des Frameworks nach Abschluss der Bachelorarbeit durch den Auftraggeber selbst durchgeführt. Das Projektteam steht dabei für allfällige Fragen und Problemstellungen zur Verfügung.

Potenzielle Ausbaumöglichkeiten für das Framework wurden in Form von Projekt- und Produktfeatures dokumentiert und können in Zukunft umgesetzt werden. Diese Features wurden während des Projekts erkannt und spezifiziert, konnten jedoch aus zeitlichen Gründen und Architektur-Constraints nicht umgesetzt werden.

Das Projektteam sieht grosses Potenzial in der entwickelten Lösung. Sie ermöglicht eine einfache und schnelle Integration einer Suche in statische Websites. Dadurch kann ein hervorragendes Sucherlebnis bereitgestellt werden und der wachsenden Anforderung, eine Such-Funktionalität auf der eigenen Website anzubieten, gerecht werden. Es ist eine Möglichkeit für Websites die Nutzerfreundlichkeit und Zugänglichkeit zu den darin enthaltenen Informationen erheblich zu erhöhen und somit das Benutzererlebnis der Website zu verbessern. Des Weiteren ist der Aufwand für die Integration einer Suche durch die Anwendung des Frameworks gering, was die Nutzung der Lösung attraktiv gestaltet.

Das Projektteam ist stolz in Zusammenarbeit mit dem Auftraggeber ein generisch anwendbares Framework entwickelt zu haben, das für SDX live gehen wird. Die Projektmitglieder freuen sich darauf das Search-Integration-Framework somit in Zukunft in Aktion zu sehen. Durch die geleistete Arbeit kann die Art und Weise, wie eine Suche in Websites integriert wird, beeinflusst, und Unternehmen sowie private Personen dabei unterstützt werden, ihre Websites zu optimieren und erfolgreich zu sein.

Danksagung

Wir möchten uns bei folgenden Personen für die Unterstützung in unserem Projekt bedanken.

Prof. Dr. Markus Stolze Wir bedanken uns herzlich bei Professor Stolze für die wertvollen Inputs und die Reviews über das ganze Projekt hinweg. Markus hat uns während den Reviews immer wieder neue Perspektiven aufgezeigt und mit seinen kritischen Fragen hat er uns über den Tellerrand hinausblicken lassen. Besten Dank dafür - dies hat das Ergebnis dieser Bachelorarbeit sehr positiv geprägt.

Alex Kühne Wir bedanken uns herzlich bei Alex für die Diskussionen, den kritischen Blick und die unterstützende Art über den gesamten Projektverlauf hinweg.

Markus Flückiger Wir bedanken uns bei Herr Flückiger für spannenden Fragen und Inputs während den Präsentationen der Arbeit.

Michael Gfeller Wir bedanken uns bei Michael für die wertvollen Code-Reviews und sein Expertenwissen.

Ivan Bütler Wir bedanken uns bei Herr Bütler für den neuen Blickwinkel und seine kniffligen Fragen.

Projektteam SDX (Swisscom) Wir bedanken uns beim SDX-Team der Swisscom für ihre Zeit, die spannenden Diskussionen und das hilfreiche Feedback während den Usability-Tests.

Teil II

Einleitung

Kapitel 1

Einleitung

1.1 Dokumentstruktur

Dieses Dokument ist grob in vier Teile gegliedert. Der erste Teil ist eine kurze Einleitung in das Projekt. Danach finden sich die Produkt- und Projektdokumentation. Am Ende wird das Ergebnis des Projektes ausgeführt und ein Fazit gezogen. Die Produktdokumentation zeigt alle relevanten Analysen und Informationen auf. Zuerst wird die Ausgangslage anhand einer Domänenanalyse aufgezeigt. Darauf folgt die Recherche zum theoretischen Hintergrund. Anschliessend werden die funktionalen und nicht-funktionalen Anforderungen spezifiziert, welche die Basis für das Kapitel Lösungsdesign bilden. Das Lösungsdesign zeigt das Konzept, welches für die Entwicklung des Search-Integrator-Frameworks ausgearbeitet wurde, die verwendete Architektur, die durchgeführte Evaluation von Suchprodukten und ausgearbeitete Wireframes. Zum Abschluss der Produktdokumentation werden die festgelegten Qualitätsmassnahmen erläutert. In der Projektdokumentation ist anhand des Projektplans dokumentiert, wie das Projekt aus einer Management-Perspektive geplant und umgesetzt wurde. Schlussendlich wird das Ergebnis der Arbeit genauer analysiert und ein Fazit über den Verlauf des Projektes und das Endprodukt gezogen sowie ein Ausblick gegeben.

1.2 Ausgangslage

Die Swisscom betreibt eine Website mit dem Design System 'Swisscom Digital Experience' oder kurz SDX. Es besteht aus einer Komponentenbibliothek, UX-Prinzipien, Designrichtlinien und Regeln, die eingehalten werden müssen, um den Swisscom-Look zu erhalten. Aktuell ist die Navigation auf der SDX-Website über eine Menüleiste möglich. Die Benutzerbasis der SDX-Website wünscht sich schon lange eine Suche, um die Auffindbarkeit von Informationen auf der Website zu erhöhen und somit gesuchte Webseiten schneller zu finden. Der Projektmanager von SDX hatte den Design-Entscheid getroffen, dass man lieber keine Suche als eine schlechte in die SDX-Website integriert. Aufgrund dieses Entscheides wurde diese Bachelorarbeit ausgeschrieben, mit dem Ziel die Integration einer Suche in die SDX-Website umfassend zu analysieren und eine qualitativ hochstehende Suche zur Verfügung zu stellen.

1.3 Problemdefinition

Aus Entwicklerperspektive ist der Aufwand für die Integration einer qualitativ hochstehenden Suche mit grossem Aufwand verbunden. Aus der Benutzerperspektive ist die Auffindbarkeit der einzelnen Webseiten sowie der darauf enthaltenen Informationen erschwert, da die Menüleiste nur beschränkten Aufschluss über den Inhalt der einzelnen Webseiten gibt. Zudem ist anhand des Menüs schwer abzuschätzen, ob man alle relevanten Daten innerhalb der Website gefunden hat. Benutzer sind gezwungen den Inhalt der Website gut zu kennen, um sich über das Menü zielorientiert und umfänglich bewegen zu können und schnell an gesuchte Informationen zu gelangen. Dies stellt insbesondere für neue Benutzer von SDX eine Hürde dar. Bezüglich der Umsetzung bestand die Anforderung, dass, wenn möglich, keine zusätzliche Infrastruktur für den Betrieb der Suchfunktion benötigt wird.

1.4 Vision

Das Projektteam hat sich zum Ziel gesetzt ein Framework für die Integration einer On-Site-Search, welche clientseitig ausgeführt wird, zu konzipieren und umzusetzen. Die Lösung soll generisch entwickelt werden, um eine Integration in weitere Projekte zu ermöglichen. Die Integration einer Suche in eine statische Website soll für Entwickler in Zukunft mit einem kleinen Aufwand verbunden sein. Und den Benutzern einer Website soll durch die über das Framework integrierte Suche eine bessere Benutzbarkeit und eine erhöhte Zugänglichkeit durch einen schnelleren Zugriff auf gesuchte Informationen geboten werden.

1.5 Herangehensweise

Das Projekt wurde gemäss dem Rational Unified Process (RUP) [RUP] in vier Phasen unterteilt: Inception, Elaboration, Construction und Transition. Innerhalb dieser Phasen wurde nach SCRUM [Scr] in Sprints geplant und gearbeitet. In der Inception-Phase wurden das Projektmanagement sowie die Aufgabenstellung ausgearbeitet und präzisiert. Während der Elaboration-Phase wurde eine intensive Analyse der bestehenden Situation und der Anforderungen zusammen mit dem Auftraggeber durchgeführt und dokumentiert. Daraus resultierten formalisierte funktionale sowie nicht-funktionale Anforderungen, Prototypen der möglichen Umsetzungsvarianten und schlussendlich ein Konzept zu Umsetzung des Frameworks. Die Machbarkeit und der Umfang des Projektes wurden durch das Implementieren von Prototypen verifiziert. In der Construction-Phase wurde die angestrebte Lösung umgesetzt und getestet. Um die Eignung des fertigen Produktes zu verifizieren, wurden Usability Tests mit dem SDX-Projektteam der Swisscom durchgeführt und Anforderungen durch Systemtests untersucht. Zum Abschluss des Projektes wurde eine Übergabe an Swisscom durchgeführt. Die Integration der Lösung in die produktive Umgebung von Swisscom wird ausserhalb der Bachelorarbeit durch den Auftraggeber auf eigenen Wunsch durchgeführt. Das Projektteam steht dem Auftraggeber dabei für allfällige Fragen oder Unklarheiten während der Integration zur Verfügung.

Teil III

Produkt Dokumentation

Kapitel 2

Domänenanalyse

2.1 Swisscom Digital Experience (SDX)

Swisscom Digital Experience (SDX) ist das Design System von Swisscom und besteht aus einer Komponentenbibliothek, UX-Prinzipien, Designrichtlinien, Dokumentationen und Regeln. Es ist ein lebender Styleguide sowohl für alle Designer und Entwickler, die für Swisscom arbeiten, sowie auch für externe Agenturen, die Applikationen oder Webseiten für Swisscom erstellen [\[sdx\]](#).

SDX muss für alle Webseiten und Web-Applikationen der Swisscom eingesetzt werden (extern sowie intern), damit diese den Look von Swisscom verkörpern [\[sdx\]](#).

Die SDX-Website ist statisch. Dies bedeutet, dass selten neue Webseiten hinzugefügt oder bestehende abgeändert werden. Die SDX-Website verfügt zudem über eine überschaubare Datenbasis. Die Datenbasis beinhaltet rund 40 Komponenten sowie Informationen zu grundlegenden Themen in der Webentwicklung und eine Entwicklerdokumentation, welche Informationen zur Integration und weiterem enthält. Der Aufbau der Website kann anhand des Domänenmodells nachvollzogen werden, siehe 2.1.

2.1.1 Domänenmodell

Abbildung 2.1 zeigt das Domänenmodell der SDX-Design-Bibliothek.

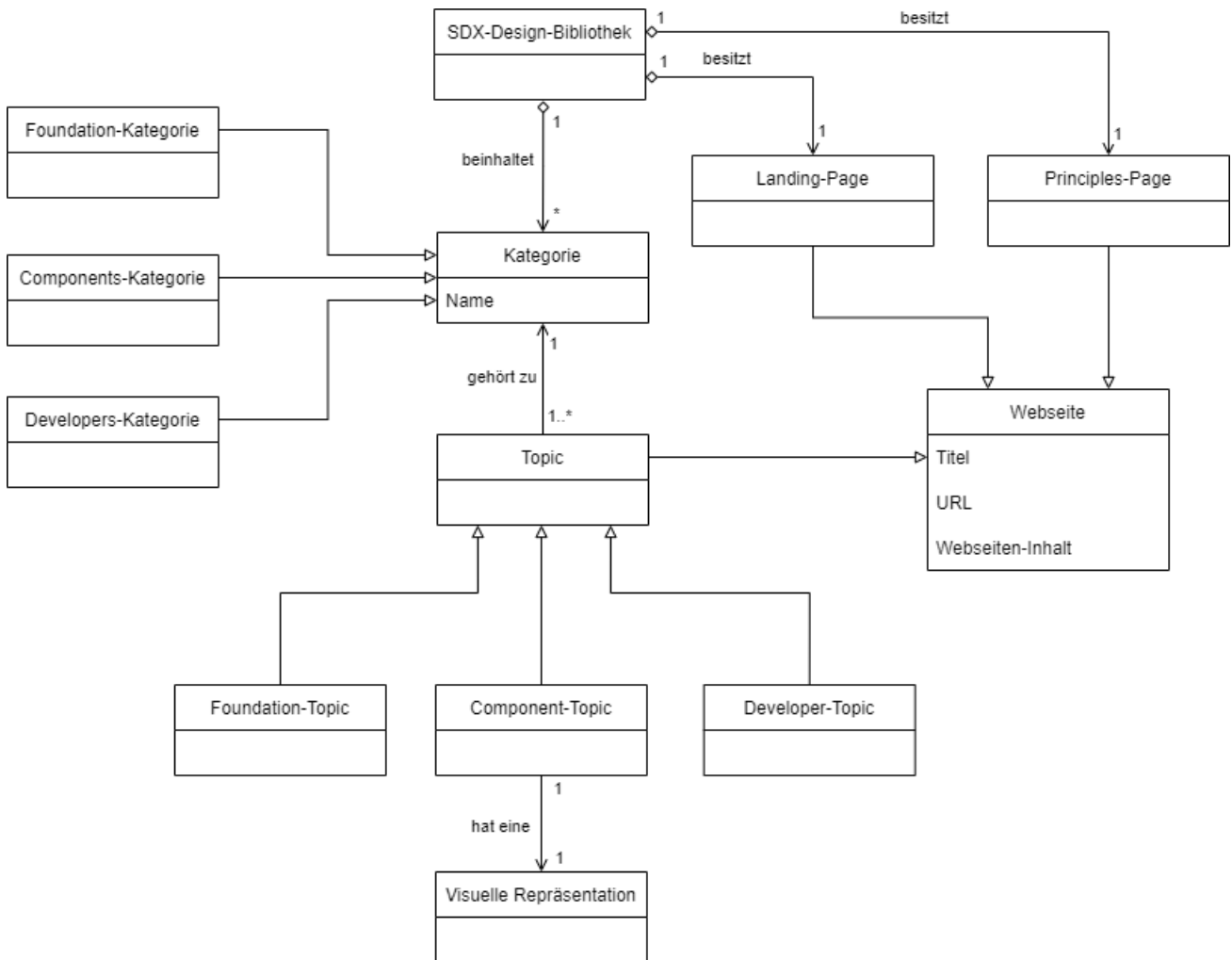


Abbildung 2.1: Domänenmodell SDX

Anmerkung: Damit das Domänenmodell nicht überladen wird, ist nicht explizit modelliert, dass Foundation-Topics zu der Foundation-Kategorie, Component-Topics zu der Components-Kategorie und Developer-Topics zu der Developers-Kategorie gehören.

2.2 Hypothesen

Aufgrund dessen, dass auf Basis von Hypothesen gearbeitet wurde, wurden in diesem Kapitel Benutzergruppen definiert, Proto-Personas [prob] daraus abgeleitet und Proto-Journeys [proa] ausgearbeitet.

Proto-Personas [prob], siehe Abschnitt 2.2.2, stellen in diesem Zusammenhang fiktive Personen dar, von welchen ausgegangen wird, dass sie das Design System von Swisscom nutzen und in einer der Benutzergruppen enthalten sind.

Proto-Journeys [proa], siehe Abschnitt 2.2.3, sind Hypothesen bezüglich Anwendungsfällen, in denen Benutzer die Such-Funktionalität auf der SDX-Website nutzen möchten. Die formulierten Situationen beziehen sich auf Suchbedürfnisse von Benutzern.

2.2.1 Benutzergruppen

Tabelle 2.1 zeigt die erkannten Benutzergruppen sowie deren Bedürfnisse innerhalb der SDX-Website auf.

Benutzergruppe	Bedürfnisse innerhalb SDX
Designer	<ul style="list-style-type: none">• Informationen bezüglich Styleoptionen einer Komponente• Informationen bezüglich Richtlinien• Beispiele der einzelnen Komponenten (Ideenfindung)
Developer	<ul style="list-style-type: none">• Anleitung bezüglich der Implementation von Komponenten• Konfigurationsmöglichkeiten von Komponenten• Anleitung zur Integration von SDX• Informationen bezüglich dem Standard
Laien	<ul style="list-style-type: none">• Interesse am Designsystem• Generelle Informationsbeschaffung

Produktmanager	<ul style="list-style-type: none"> • Beispiele der einzelnen Komponenten (Ideenfindung) • Anleitung bezüglich der Implementation von Komponenten • Konfigurationsmöglichkeiten von Komponenten • Anleitung zur Integration von SDX
----------------	--

Tabelle 2.1: Benutzergruppen und deren Bedürfnisse in SDX

Tabelle 2.2 zeigt auf, welche Benutzergruppen vorwiegend in welchen Proto-Journeys Anwendung finden.

Benutzergruppe	Proto-Journeys SDX
Designer	<ul style="list-style-type: none"> • PS02 - Icon-Größen • PS05 - Schnelle Navigation über die Suche • PS09 - Screen-Reader-Unterstützung
Developer	<ul style="list-style-type: none"> • PS01 - Komponenteninformationen • PS04 - Einarbeitung in SDX • PS05 - Schnelle Navigation über die Suche • PS06 - Einbinden von SDX in einem Projekt • PS07 - Accessibility • PS08 - Fehlerbehandlung • PS09 - Screen-Reader-Unterstützung • PS10 - Mitwirkung bei der Optimierung von SDX
Laien	<ul style="list-style-type: none"> • PS03 - Weiterbildung • PS04 - Einarbeitung in SDX • PS05 - Schnelle Navigation über die Suche • PS09 - Screen-Reader-Unterstützung

Produktmanager	<ul style="list-style-type: none"> • PS04 - Einarbeitung in SDX • PS05 - Schnelle Navigation über die Suche • PS09 - Screen-Reader-Unterstützung
----------------	---

Tabelle 2.2: Benutzergruppen und ihre Proto-Journeys

2.2.2 Proto-Personas

Im Folgenden werden Proto-Personas pro Benutzergruppe dargestellt. Diese wurden anhand folgender Kriterien aufgebaut.

1. Name
2. Benutzerprofil (Bio)
3. Demografische Information
4. Schmerzpunkte, tägliche Herausforderungen
5. Motivation und/oder Goals

Alex, 26 - Junior Developer

Bio Alex hat einen Bachelor-Abschluss in Informatik und ist leidenschaftlich daran interessiert, kreative und elegante Lösungen für komplexe technische Probleme zu entwickeln. Er ist schnell lernfähig, anpassungsfähig und schätzt die Zusammenarbeit im Team. In seiner Freizeit lernt er gerne neue Technologien und Frameworks kennen und spielt Videospiele.

Pain Points

1. Einarbeitung an neue Technologien und Frameworks
2. Effektive Kommunikation im Team
3. Produktiv bleiben während Weiterbildung
4. Herausforderungen im wettbewerbsintensiven Arbeitsmarkt
5. Karriereentwicklung und geeignete Karrieremöglichkeiten finden

Goals und Motivation Alex ist motiviert durch die Möglichkeit, kreative Lösungen zu entwickeln und seine Fähigkeiten zu verbessern. Er schätzt die Zusammenarbeit im Team und die Chance auf Karriereentwicklung in einer wettbewerbsintensiven Branche.

Jessica, 32 - Senior Developer

Bio Jessica hat einen Master-Abschluss in Informatik und über 8 Jahre Erfahrung in der Software-Entwicklung. Sie ist kompetent in verschiedenen Programmiersprachen und Frameworks und hat ein breites Verständnis für die Technologie-Industrie.

Pain Points

- Schwierigkeiten, mit Stress in anspruchsvollen Projekten umzugehen
- Herausforderungen, eine ausgewogene Work-Life-Balance zu finden
- Notwendigkeit, auf dem neuesten Stand der Technologie zu bleiben, um wettbewerbsfähig zu bleiben

Goals und Motivation Jessica ist motiviert durch die Möglichkeit, innovative und effektive Lösungen für komplexe technische Herausforderungen zu entwickeln. Sie schätzt auch die Chance, ihr Wissen zu teilen und anderen Teammitgliedern zu helfen, ihre Fähigkeiten zu verbessern.

Martin, 22 - Junior Designer

Bio Martin hat einen Bachelor-Abschluss in Grafikdesign und Erfahrung in der Gestaltung von digitalen Medien und Druckerzeugnissen. Er ist kreativ, innovativ und verfügt über ein Auge für Ästhetik und Design.

Pain Points

- Schwierigkeiten bei der Umsetzung der kreativen Vision des Kunden
- Zeitmanagement und Deadlines einhalten
- Notwendigkeit, mit verschiedenen Design-Tools und Software auf dem neuesten Stand zu bleiben
- Herausforderungen, sich in einem wettbewerbsintensiven Markt zu behaupten und geeignete Karrieremöglichkeiten zu finden

Goals und Motivation Martin ist motiviert durch die Möglichkeit, kreative und einzigartige Designs zu entwickeln, die seine Fähigkeiten und seine Leidenschaft für Grafikdesign widerspiegeln. Er schätzt auch die Chance, von erfahrenen Designern zu lernen und sich in der Branche zu etablieren.

Peter, 48 - Senior Designer

Bio Peter hat einen Bachelor-Abschluss in Industriedesign und über 20 Jahre Erfahrung in der Gestaltung von Produkten und Marken. Er ist kreativ, detailorientiert und verfügt über ein breites Wissen in verschiedenen Design-Disziplinen.

Pain Points

- Schwierigkeiten, mit den sich verändernden Design-Trends und Technologien Schritt zu halten
- Herausforderungen, Designentscheidungen zu treffen, die sowohl ästhetisch ansprechend als auch funktional sind
- Notwendigkeit, Kundenanforderungen und Budgets im Einklang zu halten
- Schwierigkeiten, ein Gleichgewicht zwischen seiner kreativen Vision und den Anforderungen des Kunden zu finden

Goals and Motivation Peter ist motiviert durch die Möglichkeit, innovative Designs zu entwickeln, die das Markenimage stärken und das Benutzererlebnis verbessern. Er schätzt auch die Chance, jüngere Designer zu Mentoren und sein Wissen und seine Erfahrung weiterzugeben.

Claudia, 16 - Informatiklernende im 1. Lehrjahr (Laie)

Bio Claudia hat keine formale Ausbildung oder Erfahrung in Design oder Entwicklung. Sie ist jedoch neugierig und motiviert, neue Fertigkeiten zu erlernen und ihre Karrierechancen zu verbessern.

Pain Points

- Schwierigkeiten, komplexe Konzepte und Design-Tools zu verstehen und anzuwenden
- Herausforderungen, sich mit erfahrenen Designern und Entwicklern auf Augenhöhe zu verständigen
- Notwendigkeit, zusätzliche Zeit und Ressourcen zu investieren, um ihr Wissen und ihre Fähigkeiten aufzubauen

Goals and Motivation Claudia ist motiviert durch die Chance, neue Fähigkeiten zu erlernen und ihre Karrierechancen zu verbessern. Sie möchte ihre Fähigkeiten in Design und Entwicklung verbessern, um in Zukunft erfolgreich in der Branche zu sein.

Claude, 42 - Produktmanager

Bio Claude hat einen Master-Abschluss in Betriebswirtschaft und über 10 Jahre Erfahrung im Produktmanagement. Sie ist strategisch und analytisch denkend und hat eine Leidenschaft für die Entwicklung und Vermarktung von Produkten.

Pain Points

- Herausforderungen, Kundenbedürfnisse und Wettbewerbsanalyse zu verstehen und umzusetzen
- Schwierigkeiten, Budget- und Ressourcenbeschränkungen einzuhalten und dennoch qualitativ hochwertige Produkte zu entwickeln
- Notwendigkeit, Teams und Stakeholder zu koordinieren und zu führen, um Projekte erfolgreich abzuschliessen

Goals and Motivation Claude ist motiviert durch die Möglichkeit, innovative Produkte zu entwickeln, die den Kundenbedürfnissen entsprechen und die Unternehmensziele unterstützen. Sie schätzt auch die Chance, mit anderen Abteilungen wie Engineering, Design und Marketing zusammenzuarbeiten, um Produkte erfolgreich auf den Markt zu bringen.

2.2.3 Proto-Journeys

Im Folgenden werden Proto-Journeys allgemein dargestellt. Die Proto-Personas aus Kapitel 2.2.2 werden für den Fall, dass das Suchbedürfnis auf alle Personas zutrifft, als 'Benutzer' zusammengefasst.

PS01 - Komponenteninformationen

Ein Benutzer arbeitet an einem neuen Feature für eine Webanwendung und muss dazu eine Komponente entwerfen, die in das bestehende Design der Anwendung passt. Der Akteur öffnet das Design System (SDX) und beginnt mit der Suche nach der passenden Komponente.

PS02 - Icon-Grössen

Ein Akteur arbeitet an einem neuen Projekt und soll herausfinden welche Grössen von Icons zur Verfügung stehen. Er weiss jedoch nicht, ob diese Information im Design System enthalten ist oder nicht.

PS03 - Weiterbildung

Ein Akteur beginnt sich mit der Webentwicklung auseinander zu setzen und fragt sich wie man eine Webseite am einfachsten strukturieren könnte bzw. das Layout durchsetzen kann. Da in seiner Firma das SDX verwendet wird, schaut er, ob diese Informationen enthalten sind oder nicht.

PS04 - Einarbeitung in SDX

Ein Benutzer möchte bei einer Ausschreibung für die Entwicklung einer Web-Applikation für die Swisscom teilnehmen. Teil der Anforderungen an die Web-Applikation ist, dass für die Gestaltung der Web-Applikation die Komponentenbibliothek SDX genutzt wird. Da der Benutzer SDX nicht kennt, möchte er sich mit damit vertraut machen und die Grundkonzepte kennenlernen. Hierzu nutzt er die Such-Funktionalität, um verschiedene Informationen zu der Design-Bibliothek zu recherchieren.

PS05 - Schnelle Navigation über die Suche

Ein Benutzer arbeitet an der Entwicklung einer Website, für welche SDX-Elemente zur Gestaltung eingesetzt werden. Der Benutzer ist bereits vertraut mit den verfügbaren Elementen in SDX, jedoch muss er jedes Mal, wenn er eine Komponente einbinden möchte, die konkrete Syntax nachschlagen. Der Benutzer bevorzugt die Suche gegenüber der Navigation, da er damit die gewünschte Komponenten-Webseite schneller findet.

PS06 - Einbinden von SDX in einem Projekt

Eine Agentur hat eine Ausschreibung für die Entwicklung einer Web-Applikation für die Swisscom gewonnen. Ein Benutzer wird nun beauftragt das Projekt aufzusetzen und Teil davon ist es, die Komponentenbibliothek SDX einzubinden. Da der Benutzer dies noch nie gemacht hat, nutzt er die Such-Funktionalität für die Recherche wie SDX für die von der Agentur gewählten Technologie eingebunden werden kann.

PS07 - Accessibility

Ein Benutzer arbeitet an der Umsetzung einer Web-Applikation, bei welcher der Accessibility eine besondere Aufmerksamkeit gewidmet wird. Für das Design der Website wird SDX eingesetzt. Der Benutzer möchte sich nun informieren was die SDX-Elemente von Haus aus bezüglich Accessibility mitbringen.

PS08 - Fehlerbehandlung

Ein Benutzer stösst bei der Entwicklung einer Web-Applikation mit der SDX-Bibliothek auf ein Problem. Aus diesem Grund nutzt der Benutzer die Such-Funktionalität von der SDX-Website, um ausfindig zu machen, ob es sich um ein bekanntes Problem handelt und somit bereits eine Lösung zur Verfügung steht.

PS09 - Screen-Reader-Unterstützung

Ein Benutzer wurde mit der Umsetzung einer Website beauftragt, für die das Hauptzielpublikum blinde Personen sind. Da es sich um eine Website für die Swisscom handelt, soll die SDX-Komponentenbibliothek eingesetzt werden. Der Benutzer möchte sich informieren, was es für Möglichkeiten gibt die SDX-Komponenten für Screen-Reader zu optimieren und was für Massnahmen diesbezüglich bereits für die SDX-Komponenten umgesetzt sind.

PS10 - Mitwirkung bei der Optimierung von SDX

Ein Benutzer ist bei der Entwicklung einer Web-Applikation mit der SDX-Komponentenbibliothek auf eine Optimierung einer Komponente gestossen. Er fragt sich, ob er bei der Optimierung von SDX mitwirken kann und seine Optimierung als Vorschlag einbringen kann. Um Antworten zu seinen Fragen zu finden, setzt der Benutzer die Such-Funktionalität der SDX-Website ein.

Kapitel 3

Theoretischer Hintergrund

Dieses Kapitel beschreibt die Ausgangslage des Projekts. Des Weiteren ist in diesem Kapitel die Recherchen zu den verschiedenen Themen des Projekts dokumentiert. Der Fokus liegt im Projekt auf einer clientseitigen Suche, weswegen man sich in der Evaluation darauf konzentriert hat. Nichtsdestotrotz wollte eine breite Recherche durchgeführt werden, um eventuell bessere Lösungen zu erkennen.

3.1 Recherche Suche

Die Grundlage für die Recherche war in erster Linie das Thema 'State of the Art der On-Site-Suche'. Ausgehend von dieser wurde eine breite Grundlage geschaffen, indem man sich generell über das Thema Suche informierte, ohne sich dabei auf eine bestimmte Art einzuschränken - ein breiter Rundherum-Schlag. Das Ziel war es, relevante Themen zu identifizieren und gute Quellen zu finden. Die primäre Informationsquelle stellte das Internet sowie Fachliteratur dar. Auf Basis dieser Grundlagen wurden die für dieses Projekt relevanten Informationen erkannt, genauer analysiert und dokumentiert.

3.1.1 Gründe für eine Suche

Benutzern zu ermöglichen gesuchte Informationen einfach zu finden, ist eine der Hauptzutaten für eine benutzerfreundliche Website. Eine verständliche und übersichtliche Navigation ist ein wichtiges Feature einer benutzerfreundlichen Website, denn häufig wünschen sich Benutzer mehr als eine intuitive Navigation, um gesuchte Informationen zu finden. Nebst einer Navigation bietet eine On-Site-Suche Unterstützung für User, damit diese gesuchte Informationen schnell und einfach finden können.

Eine On-Site-Suche ist mehr als nur ein Suchbalken - sie ist ein wichtiger Bestandteil der Navigation und ein Key-Feature um die Benutzerinteraktion zu verbessern. Heutzutage wird erwartet, dass eine Website eine Suche anbietet und die User sind sich gewohnt, dass die Suche sehr gut funktioniert und die Resultate anzeigt, welche Ihr Suchbedürfnis befriedigen.

Welchen Stellenwert die Suche auf Websites haben kann, wird durch YouTube eindrücklich verdeutlicht: Auf YouTube wird pro Minute über 500 Stunden Videomaterial hochgeladen. Ohne eine gute Suche wäre

es praktisch unmöglich die Videos zu finden, die man effektiv sucht.

Quellen: [reaa], [reab], [reac] [yt-]

Features einer guten Suche

Eine gute Suche wird nebst der Eigenschaft, dass sie die Suchergebnisse liefert, welche das Suchbedürfnis der Suchenden befriedigt, durch verschiedene Features ausgezeichnet. Nachfolgend ist eine Auswahl von wichtigen Features aufgeführt.

- Autocomplete / Type-Ahead: Die Suche soll automatisch gestartet werden, wenn der User den Suchbegriff am Eintippen ist. Dies steht im Kontrast zur klassischen Suche, bei der die Suche erst ausgelöst wird, wenn die Eingabe des Suchbegriffs durch Drücken der ENTER-Taste oder Klicken des Such-Buttons bestätigt wird.
- Synonyme: Eine gute Suche findet nicht nur Ergebnisse, die dem exakten Suchbegriff entsprechen, sondern berücksichtigt auch Synonyme des Suchbegriffs.
- Visuelle Darstellung von Suchresultaten: Eine visuelle Darstellung der Suchresultate unterstützt User bei der Entscheidung welches Suchresultat das passendste für ihr Suchbedürfnis ist.
- Kategorisierte Suchresultate: Je nach Anwendungsfall kann die Kategorisierung der Suchresultate den Usern einen Mehrwert bieten.
- Such-Filter: Such-Filter ermöglichen einem User den Suchbereich einzuschränken und somit die Suchergebnisse besser auf sein Suchbedürfnis abzustimmen.
- Search Analytics: Search Analytics beinhalten Statistiken über vergangene Suchen und über das Suchverhalten von Usern und helfen dabei eine Suche kontinuierlich zu optimieren.
- Such-Historien: In Such-Historien werden vergangene (erfolgreiche) Suchen gespeichert. Diese Such-Historien werden einem User bei einer zukünftigen Suche zur Unterstützung angezeigt.

Quellen: [good], [gooc], [gooa]

3.1.2 Sucharchitekturen

Es gibt verschiedene Architekturen, welche für eine Suche infrage kommen. Eine Auflistung davon ist nachfolgend aufgeführt. Die als für das Projekt relevant betrachteten Architekturen werden in den Unterkapiteln dieses Abschnitts beschrieben. Konkret sind dies die clientseitige sowie die serverseitige Suche.

- Clientseitige Suche / On-Site-Search
- Serverseitige Suche
- Distributed Search [dis]

- Federated Search [[fed](#)]

Clientseitige Suche

Clientseitig bezieht sich auf die Verwendung von Javascript-basierten Suchbibliotheken und -werkzeugen, auf der Client-Seite einer Website. Diese Variante bietet sich für uns an, da wir eine statische Website mit geringer Datenveränderungs-Rate sowie einer kleinen Datenbasis haben. Der clientseitige Ansatz bietet zudem folgende Eigenschaften, welche die Wahl als prioritäre Variante unterstützen.

- Schnelle Suche: Da die Suche auf der Client-Seite durchgeführt wird, kann sie schneller ausgeführt werden, da keine Serveranfragen erforderlich sind.
- Offline-Suche: Da eine clientseitige Suche lokal im Browser ausgeführt wird, können Benutzer auch dann suchen, wenn keine Internetverbindung verfügbar ist.
- Natur der Suche in SDX: Für die Suche in SDX werden Webseiten (HTML-Dateien) durchsucht. Dies steht im Gegensatz zu einer klassischen Suche, welche typischerweise eine Datenbank durchforstet.
- Geringere Komplexität: Da bei der clientseitigen Suche keine Client-Server-Kommunikation stattfinden muss, ist der Prozess der Suchabfrage weniger komplex.
- Server- und Netzwerkentlastung: Durch die Verlagerung der Suche auf die Client-Seite wird die Server- sowie die Netzwerkbelastung reduziert, was zu einer besseren Leistung der Website führt.
- Benutzerfreundlichkeit: Die Benutzer können die Suchergebnisse in Echtzeit sehen, während sie ihre Suchanfrage eingeben, was das Benutzererlebnis verbessert.
- Anpassbare Suche: Die Suchfunktion kann an die spezifischen Bedürfnisse der Website angepasst werden, einschliesslich der Filterung und Sortierung von Suchergebnissen.

Quellen: [[cli](#)]

Serverseitige Suche

Serverseitig bezieht sich auf den Prozess der Durchführung einer Suchabfrage auf dem Server anstatt auf dem Client. Die serverseitige Suche wird häufig von Webanwendungen verwendet, um die Effizienz und die Genauigkeit von Suchergebnissen zu verbessern. Sie eignet sich besonders, wenn grosse komplexe Datenbasen durchsucht werden müssen. Bei der Suche wird normalerweise eine Anfrage an einen Server gesendet, der dann eine Datenbasis nach relevanten Einträgen durchsucht und diese als Antwort an den Client schickt. Auf diese Weise kann der Server den Rechenaufwand bei der Suche übernehmen, während der Client nur das Ergebnis präsentieren muss.

- Schnelligkeit und Skalierbarkeit: Eine serverseitige Suche kann auf leistungsstarken Servern ausgeführt werden, wodurch die Suche schnell und skalierbar wird.
- Unterstützung für komplexe Suchanfragen: Eine serverseitige Suche unterstützt häufig komplexe Suchanfragen wie boolesche Operatoren, Phrasensuche und Wildcards, welche die Genauigkeit der

Ergebnisse verbessern können.

- **Mehrsprachige Suche:** Die serverseitige Suche kann eine mehrsprachige Suche unterstützen, die für Unternehmen und Organisationen mit internationaler Präsenz von Vorteil sein kann.
- **Paginierung:** Eine serverseitige Suche kann Suchergebnisse paginieren, d.h. auf mehrere Seiten aufteilen, um eine bessere Benutzererfahrung zu bieten und die Ladezeiten zu reduzieren.
- **Analyse und Reporting:** Eine serverseitige Suche kann Daten über Suchanfragen und Suchergebnisse sammeln und analysieren, um Einblicke in die Nutzung und Performance der Suche zu gewinnen.
- **Filterung und Sortierung:** Die serverseitige Suche kann Suchergebnisse filtern und sortieren, um die relevantesten Ergebnisse zu präsentieren. Dies kann auf der Grundlage von Kriterien wie Relevanz, Datum oder Alphabet erfolgen.

Clientseitiger vs. serverseitiger Ansatz

Wenn es um die Suche auf einer Website mit geringer Datenbasis und wenig Datenänderungen geht, kann eine clientseitige Suche oder eine serverseitige Suche verwendet werden. Beide Ansätze haben ihre eigenen Vor- und Nachteile.

Ein clientseitiger Ansatz ist effektiv, da er leichtgewichtig ist, eine einfachere Integration und eine schnelle Suche ermöglicht. Die Suchanfrage wird direkt vom Browser des Benutzers ausgeführt, wodurch die Serverbelastung minimiert wird. Da die Datenbasis klein und statisch ist, ist es auch wahrscheinlich, dass die Suchergebnisse schnell angezeigt werden.

Ein serverseitiger Ansatz kann jedoch auch Vorteile bieten. Ein Server hat in der Regel mehr Rechenleistung und Speicherplatz als ein Browser und kann daher leistungsstarke Suchalgorithmen ausführen. Serverseitige Suchen können auch auf komplexe Anfragen reagieren und Ergebnisse in Echtzeit liefern. Zudem können Features wie Logging und Fehlerbehandlung sinnvoll eingesetzt werden. Der Wartungs- und Rechenaufwand wird jedoch vergrößert, da der Input auf der Client-Seite geprüft und bereinigt werden muss, die Kommunikation muss gesichert werden und Server sowie Datenbanken müssen gewartet werden.

3.1.3 Suchmethoden

Suchmethoden sind verschiedene Methoden, die verwendet werden um Informationen oder Daten aus einer Datenbank oder einem Textdokument zu extrahieren. Dabei ist zu beachten, dass diese Methoden auch kombiniert werden können.

- **Volltext-Suche:** Eine Suche, die nach einer oder mehreren Zeichenfolgen innerhalb eines Textes oder eines Dokuments sucht. Sie liefert alle Dokumente oder Texte, in denen die gesuchte Zeichenfolge vorkommt, auch wenn sie nicht exakt übereinstimmen.
- **Exakte-Suche:** Eine Suche nach einem exakten Begriff, der genau mit der eingegebenen Zeichenfolge übereinstimmt.

- Fuzzy-Suche: Eine Suche, die Ergebnisse zurückgibt, die dem Suchbegriff ähnlich sind, auch wenn sie nicht genau übereinstimmen.
- Wildcard-Suche: Eine Suche, die Platzhalter wie '*' oder '?' verwendet, um eine beliebige Anzahl von Zeichen in einem Suchbegriff zu ersetzen.
- Phonetische-Suche: Eine Suche, die nach Wörtern sucht, die basierend auf der Aussprache ähnlich wie der Suchbegriff klingen.
- Synonym-Suche: Eine Suche, die auch Ergebnisse zurückgibt, die mit Synonymen des Suchbegriffs übereinstimmen.
- Facettierte-Suche: Eine Suche, die es dem Benutzer ermöglicht, Suchergebnisse auf der Grundlage von vorgegebenen Kategorien, Eigenschaften oder Merkmalen zu filtern.
- Geospatial-Suche: Eine Suche, die auf geografischen Standorten und räumlichen Entfernungen basiert.
- Autocomplete-Suche: Eine Suche, die Suchvorschläge während der Eingabe des Suchbegriffs anzeigt, um die Suche zu beschleunigen.

Die für den im Projekt gewählten Ansatz relevanten Suchmethoden werden folgend aufgelistet. Die Methoden können im Projekt Features darstellen, welche die Suche anbietet. Welche Methoden bzw. welche Kombination der Methoden effektiv eingesetzt wird, ist in Kapitel Konzept 5.1 ersichtlich.

- Volltext-Suche
- Fuzzy-Suche
- Synonym-Suche
- Autocomplete-Search
- Facettierte-Suche

3.1.4 Suchalgorithmen

Nebst den verschiedenen Suchmethoden stehen bezüglich der Implementation verschiedene Algorithmen zur Verfügung. Eine Übersicht von verschiedenen Suchalgorithmen ist nachfolgend aufgelistet. Sofern ein Framework für die Suche eingesetzt wird, sind diese Informationen vor allem interessant um Frameworks zu vergleichen bzw. um diese überhaupt zu verstehen, da sie unterschiedliche Algorithmen einsetzen können.

- Lineare Suche (Linear Search): Dies ist die einfachste Form der Suche, bei der jedes Element in einer Liste nacheinander durchsucht wird, bis das gesuchte Element gefunden wird.
- Binäre Suche (Binary Search): Dieser Algorithmus kann nur verwendet werden, wenn die zu durchsuchende Liste sortiert ist. Die Suche beginnt in der Mitte der Liste und es wird entschieden,

ob das gesuchte Element in der linken oder rechten Hälfte der Liste zu finden ist. Dieser Prozess wird so lange fortgesetzt, bis das gesuchte Element gefunden wird.

- **Interpolation-Suche (Interpolation Search):** Diese Suche ähnelt der binären Suche, verwendet jedoch eine lineare Interpolation zwischen den Schlüsselwerten, um eine Schätzung der Position des gesuchten Elements zu ermitteln. Dadurch kann die Suche schneller sein, wenn die Elemente gleichmässig verteilt sind.
- **Exponentielle Suche (Exponential Search):** Diese Suche beginnt bei der ersten Position und verdoppelt die Suchreichweite exponentiell, bis das gesuchte Element gefunden oder überschritten wurde. Anschliessend wird eine binäre Suche innerhalb dieses Bereichs durchgeführt.
- **Hashing:** Hashing ist ein Algorithmus, der verwendet wird, um schnell auf Datensätze zuzugreifen, indem eine Hash-Funktion auf den Schlüsselwert jedes Datensatzes angewendet wird. Der Hash-Wert wird als Index verwendet, um den Datensatz in einer Hash-Tabelle zu speichern und aufzurufen.
- **Trie-Suche (Trie Search):** Ein Trie ist eine Baumstruktur, die verwendet wird, um Wörter zu speichern und zu suchen. Bei einer Trie-Suche wird der Baum entlang der Zeichen in der Suchanfrage durchsucht, bis das gesuchte Wort gefunden wird.
- **Invertierte Indexierung:** Eine invertierte Indexsuche wird häufig in Suchmaschinen verwendet und ist eine Methode zur Indexierung von Dokumenten. Bei der Indexierung wird jeder Begriff, der in einem Dokument vorkommt, in einen Index aufgenommen. Wenn eine Suche durchgeführt wird, werden alle Dokumente im Index durchsucht, um eine Liste von relevanten Dokumenten zurückzugeben.

Quellen: [[seab](#)], [[seaa](#)]

3.1.5 Produkte

Im folgenden Kapitel werden Bibliotheken und Frameworks, welche eine Suche bereitstellen, kurz vorgestellt und anhand von Schlüsselfeatures vergleichbar gemacht. Diese Informationen dienen als Grundlage für die spätere Evaluation (siehe Abbildung 5.8). Die Informationen stammen aus den Dokumentationen der einzelnen Frameworks und wurden entsprechend zusammengefasst.

Algolia [[alga](#)]

Algolia ist eine Search-as-a-Service-Plattform, die es Entwicklern ermöglicht, schnelle, relevante und intuitive Suchfunktionen in ihre Anwendungen zu integrieren [[alga](#)]. Algolia bietet eine gehostete Such-API, die es Entwicklern erlaubt, die Suche in ihre Anwendungen zu integrieren.

- **Suche als Service:** Algolia ist eine vollständig verwaltete Suchplattform, sodass man sich nicht um die Einrichtung und Wartung der eigenen Suchinfrastruktur kümmern muss.
- **Schnelle Suche:** Algolia ist so konzipiert, dass es selbst bei grossen Datensätzen in weniger als einer Sekunde Suchergebnisse liefert.

- Anpassbare Relevanz: Algolias Suchalgorithmen können an auf spezifischen Anforderungen angepasst werden, damit die relevantesten Ergebnisse für die Benutzer angezeigt werden.
- Autocomplete: Algolia bietet eine Autovervollständigungs-Funktion, die es Benutzern ermöglicht, schnell nach Begriffen zu suchen, während sie tippen.
- Typo-Toleranz: Algolia erkennt und korrigiert Rechtschreibfehler in Suchanfragen automatisch, um sicherzustellen, dass Benutzer trotz Tippfehlern relevante Ergebnisse erhalten.
- Filterung und Facetten: Algolia bietet fortschrittliche Filter- und Facettenfunktionen, mit denen Benutzer ihre Suchergebnisse basierend auf Kriterien wie Kategorie, Preis oder Datum verfeinern können.
- Echtzeit-Updates: Algolia aktualisiert die Suchergebnisse automatisch, sobald neue Daten verfügbar sind, sodass Benutzer immer die aktuellen Ergebnisse sehen.
- Analytics: Algolia bietet umfangreiche Analytics-Funktionen, mit denen das Suchverhalten der Benutzer überwacht und die Suche optimiert werden kann.

Doofinder [\[doo\]](#)

Doofinder ist eine Suchmaschine für E-Commerce-Websites, die es Benutzern ermöglicht, schnell und einfach nach Produkten zu suchen [\[doo\]](#).

- Kostenpflichtig
- Schnelle Suche: Doofinder bietet eine schnelle und präzise Suche, die es Benutzern ermöglicht, schnell nach Produkten zu suchen.
- Automatische Vervollständigung: Die Suchmaschine bietet eine automatische Vervollständigungs-Funktion, welche die Suchanfrage des Benutzers ergänzt und relevante Vorschläge anzeigt.
- Synonym- und Tippfehlererkennung: Doofinder kann Synonyme und Tippfehler erkennen, um sicherzustellen, dass relevante Ergebnisse angezeigt werden.
- Personalisierte Ergebnisse: Die Suchergebnisse können auf den Standort des Benutzers und seine Vorlieben angepasst werden, um ein personalisiertes Einkaufserlebnis zu bieten.
- Filter und Sortierung: Benutzer können die Suchergebnisse nach verschiedenen Kriterien filtern und sortieren, wie z.B. Preis, Marke oder Verfügbarkeit.
- Analytics und Berichterstattung: Doofinder bietet detaillierte Analysen und Berichte über die Nutzung der Suchmaschine, um die Suche und das Einkaufserlebnis zu optimieren.

Elasticsearch [\[ela\]](#)

Elasticsearch ist eine Open-Source-Such- und Analyse-Engine, die auf Apache Lucene basiert [\[ela\]](#). Es ist eine skalierbare und verteilte Suchplattform, die schnelle Volltext-Suche, Datenanalyse und Log-Analyse

ermöglicht.

- Volltext-Suche: Elasticsearch bietet eine schnelle und genaue Volltext-Suche in Echtzeit.
- Skalierbarkeit: Elasticsearch ist skalierbar und kann problemlos auf grosse Datenmengen und Anforderungen erweitert werden.
- Verteilte Architektur: Elasticsearch ist auf eine verteilte Architektur ausgelegt, die eine hohe Verfügbarkeit und Ausfallsicherheit bietet.
- Multitenancy: Elasticsearch ermöglicht eine effiziente Verwaltung mehrerer Indizes und Suchanfragen durch seine Multitenancy-Fähigkeiten.
- Aggregationen: Elasticsearch bietet leistungsstarke Aggregationsfunktionen, mit denen Benutzer komplexe Datenanalysen durchführen und Erkenntnisse gewinnen können.
- Echtzeit-Datenanalyse: Elasticsearch ermöglicht die Echtzeit-Analyse grosser Datenmengen und bietet eine schnelle Suche in Log-Dateien und anderen strukturierten oder unstrukturierten Datenquellen.
- Geospatial-Suche: Elasticsearch bietet spezielle Funktionen für die Geospatial-Suche, einschliesslich der Möglichkeit, Entfernungen zwischen Orten zu berechnen und Geofencing-Anfragen auszuführen.
- RESTful-API: Elasticsearch bietet eine RESTful-API, die einfach zu verwenden ist und die Integration mit anderen Anwendungen und Diensten ermöglicht.

Fuse.js [[fus](#)]

Fuse.js ist ein Open-Source-JavaScript-Framework, das eine Fuzzy-Suche in Texten ermöglicht [[fus](#)]. Im Gegensatz zur exakten Volltext-Suche, bei der nur nach genauen Übereinstimmungen gesucht wird, berücksichtigt Fuse.js auch ähnliche Wörter und Schreibweisen. Es verwendet einen Algorithmus namens 'Bitap', um die bestmögliche Übereinstimmung zwischen Suchanfrage und Textinhalt zu finden. Fuse.js kann in Browsern und Node-Anwendungen verwendet werden und bietet eine einfache API und eine Vielzahl von Konfigurationsoptionen, um die Suche anzupassen.

Features von Fuse.js:

- Fuzzy Suche: Fuse.js unterstützt Fuzzy-Searching, was bedeutet, dass es auch Ergebnisse findet, die nur teilweise der Suchanfrage des Benutzers entsprechen. Es verwendet einen angepassten Algorithmus, um unvollständige Suchanfragen zu lösen und Ergebnisse zurückzugeben, die dem Benutzer relevant erscheinen.
- Pattern Suche: Fuse.js unterstützt die Suche nach bestimmten Mustern oder Regex-Pattern, anstatt nur nach exakten Übereinstimmungen zu suchen.
- Mehrsprachigkeit: Fuse.js unterstützt die Suche in mehreren Sprachen und enthält vordefinierte Funktionen für Sprachen wie zum Beispiel Deutsch, Englisch, Französisch, Spanisch, Portugiesisch

oder Russisch.

- Gewichtete Suche: Fuse.js ermöglicht gewichtete Suchen, was bedeutet, dass die Relevanz von Ergebnissen anhand bestimmter Kriterien wie zum Beispiel Datum, Preis oder Anzahl der Treffer gewichtet werden können.
- Search Across Multiple Fields: Fuse.js kann in mehreren Feldern gleichzeitig suchen, was bedeutet, dass eine Suche über verschiedene Datentypen wie Text, Zahlen und Datum ausführt werden kann.
- Search Result Highlighting: Fuse.js kann Suchergebnisse hervorheben und zeigen, wo die Suchanfrage in den Ergebnissen gefunden wurde.
- Easy-to-use API: Fuse.js verfügt über eine einfache API, die einfach zu verwenden und anzupassen ist.
- Small Size: Fuse.js ist eine kleine Bibliothek, die schnell geladen und ausgeführt wird.
- Open Source: Fuse.js ist eine Open-Source-Bibliothek, die unter der MIT-Lizenz veröffentlicht wurde, was bedeutet, dass sie für kommerzielle und nicht-kommerzielle Projekte frei verwendbar und modifizierbar ist.
- High Performance: Fuse.js ist darauf ausgelegt, schnell und effizient zu sein, mit schnellen Suchergebnissen, die in Millisekunden zurückgegeben werden.

Google Programmable Search Engine [\[gps\]](#)

Google Programmable Search Engine ist ein kostenloses Tool, mit dem benutzerdefinierte Suchmaschinen erstellt und in eine Website integriert werden können [\[gps\]](#). Die wichtigsten Features sind hier aufgelistet.

- Anpassbare Suchmaschine: Mit Google Programmable Search Engine können Benutzer benutzerdefinierte Suchmaschinen erstellen, die auf bestimmte Websites oder Inhalte spezialisiert sind.
- Vollständige Kontrolle über Suchergebnisse: Benutzer können die Suchergebnisse steuern und filtern, um sicherzustellen, dass nur relevante Ergebnisse angezeigt werden.
- Einfache Integration: Die Suchmaschine kann einfach in eine Website integriert werden, ohne dass umfangreiche Kenntnisse in der Programmierung erforderlich sind.
- Vielseitige Optionen: Es gibt eine Vielzahl von Optionen für die Anpassung der Suchmaschine, wie z.B. Farbschemata, Suchfeldpositionierung und Autovervollständigung.
- Keine Werbung: Im Gegensatz zu anderen kostenlosen Suchmaschinen-Tools wird bei dieser keine Werbung in den Suchergebnissen angezeigt.
- Analytics und Berichterstattung: Benutzer können die Nutzung der Suchmaschine verfolgen und detaillierte Analysen und Berichte über Suchanfragen und Ergebnisse erhalten.

Inbenta [\[inb\]](#)

Die KI-gestützte semantische Suchmaschine von Inbenta nutzt fortschrittliche Technologien wie Natural Language Processing und Machine Learning, um die Suchanfragen der Benutzer zu verstehen und relevante Ergebnisse bereitzustellen [\[inb\]](#). Sie ist in der Lage, komplexe Abfragen zu verarbeiten und auch Synonyme und ähnliche Ausdrücke zu berücksichtigen, um die Suche zu optimieren. Die Plattform bietet auch die Möglichkeit, die Suchergebnisse durch personalisierte Empfehlungen zu verbessern, die auf dem Verhalten und den Präferenzen des Benutzers basieren.

- **Natural Language Processing (NLP):** Die Suchmaschine nutzt NLP-Technologien, um die Suchanfragen der Benutzer zu verstehen und relevante Ergebnisse zu liefern.
- **Machine Learning:** Die Suchmaschine verwendet Machine-Learning-Algorithmen, um die Qualität der Suchergebnisse im Laufe der Zeit zu verbessern und sich an das Verhalten der Benutzer anzupassen.
- **Synonym- und ähnliche Ausdrücke:** Die semantische Suchmaschine von Inbenta kann Synonyme und ähnliche Ausdrücke berücksichtigen, um die Suche zu optimieren und relevantere Ergebnisse zu liefern.
- **Personalisierte Empfehlungen:** Inbenta bietet die Möglichkeit, die Suchergebnisse durch personalisierte Empfehlungen zu verbessern, die auf dem Verhalten und den Präferenzen des Benutzers basieren.
- **Mehrsprachigkeit:** Die Suchmaschine von Inbenta unterstützt mehrere Sprachen und kann in einer Vielzahl von Anwendungsfällen eingesetzt werden, unabhängig von der Sprache des Benutzers.
- **Automatisierte Workflows:** Inbenta bietet automatisierte Workflows, um den Kundensupport zu optimieren und sicherzustellen, dass die Suchanfragen der Benutzer schnell und effektiv bearbeitet werden.

Lunr.js [\[lun\]](#)

Lunr.js ist ein Open-Source-JavaScript-Framework für die Volltext-Suche in statischen Websites oder Webanwendungen [\[lun\]](#). Es bietet eine einfache API und basiert auf der Inverted-Index-Technologie, um Suchergebnisse schnell und effizient zu liefern. Lunr.js kann in Node.js-Anwendungen und in Browsern verwendet werden. Es ist modular aufgebaut und kann mit verschiedenen Plugins erweitert werden, um Funktionen wie Stemming, Stop-Words-Filterung oder Synonyme zu unterstützen.

Features von Lunr.js:

- **Leichtgewichtig:** Lunr.js ist eine leichtgewichtige Bibliothek, die einfach in eine Website oder Webanwendung integriert werden kann.
- **Clientseitig:** Lunr.js ist eine clientseitige Such-Bibliothek, was bedeutet, dass sie vollständig im Browser des Benutzers ausgeführt wird und keine serverseitigen Ressourcen erfordert.

- **Inverted-Indexierung:** Lunr.js verwendet einen invertierten Index, um einen durchsuchbaren Index des Inhalts auf der Website zu erstellen.
- **Partial Matching:** Lunr.js unterstützt Partial Matching, was bedeutet, dass es auch Wörter finden kann, die nur teilweise der Suchanfrage des Benutzers entsprechen. Zum Beispiel würde eine Suche nach 'Lauf' auch 'Läufer', 'Laufen' und 'Läufe' finden.
- **Benutzerdefinierte Bewertung:** Lunr.js ermöglicht es, den Bewertungsalgorithmus anzupassen, der zur Bestimmung der Relevanz von Suchergebnissen verwendet wird. Dies kann nützlich sein, wenn bestimmte Arten von Inhalten oder Seiten in den Suchergebnissen priorisiert werden sollen.
- **Mehrsprachigkeit:** Lunr.js unterstützt mehrsprachige Suche, was bedeutet, dass es Inhalte in mehreren Sprachen durchsuchen kann.
- **Schnell:** Lunr.js ist darauf ausgelegt, schnell und effizient zu sein, mit Suchergebnissen, die in Millisekunden zurückgegeben werden.
- **Einfach zu verwenden:** Lunr.js ist einfach einzurichten und zu verwenden, mit klarer Dokumentation und Beispielen auf der Lunr.js-Website.
- **Open-Source:** Lunr.js ist eine Open-Source-Bibliothek, die unter der MIT-Lizenz veröffentlicht wurde, was bedeutet, dass sie für kommerzielle und nicht-kommerzielle Projekte frei verwendbar und modifizierbar ist.

Solr [\[sol\]](#)

Solr ist eine skalierbare, Open-Source-Plattform für Such- und Indexierung, die eine Vielzahl von Suchfunktionen und -funktionen unterstützt und leicht in andere Anwendungen integriert werden kann.

1. **Volltextsuche:** Solr unterstützt Volltextsuche, d.h. es kann nach Wörtern in verschiedenen Feldern eines Dokuments gesucht werden und relevante Ergebnisse zurückgeben.
2. **Facettierte Suche:** Solr erlaubt facettierte Suche, wodurch Benutzer Suchergebnisse nach verschiedenen Kriterien (z.B. Preis, Kategorie, Datumsbereich) filtern können.
3. **Geospatial Suche:** Solr unterstützt geospatiale Suche, d.h. es kann nach Dokumenten basierend auf ihrem Standort und ihrer Nähe zu anderen Standorten gesucht werden.
4. **Echtzeit-Indexierung:** Solr erlaubt Echtzeit-Indexierung, d.h. Dokumente können sofort nach ihrer Hinzufügung zum Index indexiert und durchsuchbar gemacht werden.
5. **Trefferhervorhebung:** Solr erlaubt Trefferhervorhebung, d.h. es kann die Suchbegriffe in den Suchergebnissen hervorgehoben werden, um Benutzern zu helfen, relevante Informationen schnell zu identifizieren.
6. **Rechtschreibprüfung:** Solr bietet eingebaute Rechtschreibprüfung, welche Tippfehler korrigieren und die Suchgenauigkeit verbessern kann.

7. Suchvorschläge: Solr kann Suchvorschläge basierend auf den Suchbegriffen des Benutzers bereitstellen, was Benutzern helfen kann, ihre Suche zu verfeinern und relevantere Ergebnisse zu finden.
8. Skalierbarkeit: Solr ist darauf ausgelegt, skalierbar zu sein, d.h. es kann grosse Datenmengen und Suchanfragen verarbeiten.
9. Erweiterbarkeit: Solr ist sehr erweiterbar, d.h. es kann angepasst und erweitert werden, um spezifische Geschäftsanforderungen zu erfüllen.
10. Integration: Solr bietet eine RESTful-API, was die Integration in andere Anwendungen und Systeme erleichtert. Es unterstützt auch eine Vielzahl von Datenformaten wie XML, JSON und CSV.

3.2 Recherche Parser

In diesem Abschnitt sind Recherchen bezüglich der Implementation des Webseiten-Parsers dokumentiert.

3.2.1 SEO

Die Such-Funktionalität soll sich standardmässig an den Best Practices von SEO [\[seo\]](#) orientieren. D.h., dass die Suche für eine Website, die für Suchmaschinen optimiert ist, ohne zusätzliche Konfigurationen gute Ergebnisse liefert.

Wichtige Elemente

Google betrachtet die Inhalte an folgenden Stellen als besonders wichtig [\[goob\]](#):

- Der Titel der Webseite (title)
- Meta-Tag 'Description'
- Inhalte in Überschriftselementen (h1, h2, etc.)
- Alt-Attribute von Bildern/Videos
- Link-Texte (für Verweis auf die verlinkten Seiten)

3.2.2 Semantic HTML

Semantic HTML beschreibt die semantische Verwendung von HTML-Attributen [\[sem\]](#). Die Indexierung der Website geht davon aus, dass sich Websites, welche die Suche integrieren, an die wichtigsten Guidelines von Semantic HTML halten.

Auswahl wichtiger Guidelines

- ``-Elemente beinhalten semantisch wichtigen Content
- Teile einer Webseite werden in `<section>`- und `<article>`-Elementen gegliedert.

3.2.3 Synonyme

Die Such-Funktionalität soll nicht nur Treffer anhand von auf den Web-Pages vorhandenen Wörtern finden, sondern auch Synonyme der als wichtig eingestuften Begriffen auf den Web-Pages berücksichtigen. In diesem Kapitel ist die Recherche bezüglich Lösungsvarianten für eine solche Funktion dokumentiert.

Thesaurus von API Ninjas

Thesaurus von API Ninjas ist eine API für die Abfrage von Synonymen zu englischen Wörtern. Es handelt sich hierbei um einen allgemeinen Thesaurus, der nicht auf UI-Komponenten spezialisiert ist.

Link: <https://api-ninjas.com/api/thesaurus>

Open UI

Open UI bietet eine sehr gute Übersicht über Synonyme zu UI-Komponenten an. Leider gibt es kein API, um Synonyme abzufragen. Somit könnte für den Look-Up eine strukturierte Datei von Hand erstellt werden oder es könnte ein Crawler implementiert werden, der die Informationen von der Website zusammensucht und in einem strukturierten Format speichert.

Link: <https://open-ui.org/>

3.3 Technologien

Grundsätzlich wird das Projekt aus Frontend-Perspektive betrachtet. Es stellt sich demnach die Frage, welche Frontend-Technologien am besten bei der Umsetzung helfen. Da die Plattform grundsätzlich ein Webbrowser auf einem Computer oder einem mobilen Gerät ist, liegt die Entscheidung TypeScript für die Entwicklung zu verwenden nahe. TypeScript bringt gegenüber Javascript einige Vorteile und wird auch so vom Auftraggeber vorgegeben.

3.4 Deployment

Am Ende des Projekts muss die Lösung an Swisscom übergeben werden. Dazu finden sich verschiedene Lösungen. Einerseits wäre es möglich die Codebasis an sich zu übergeben, andererseits könnte man eine komplett generische Lösung oder Teile der Lösung als NPM-Paket anbieten.

3.4.1 npm

npm 'Node Package Manager' ist eine Software-Paketmanager-Anwendung für die Programmiersprache JavaScript. Es wird am häufigsten mit der Laufzeitumgebung Node.js verwendet, um Pakete und Abhängigkeiten zu installieren, zu verwalten und zu aktualisieren, die in Projekten verwendet werden [npma].

Mit npm können Entwicklerinnen und Entwickler Pakete und Module von einem zentralen Repository herunterladen, die von anderen Entwicklern erstellt wurden, und sie in ihren eigenen Projekten

wiederverwenden. npm enthält auch Tools, um es Entwicklerinnen und Entwicklern zu ermöglichen, ihre eigenen Pakete zu erstellen und zu veröffentlichen, damit andere Entwickler sie verwenden können.

Insgesamt hat npm eine wichtige Rolle im Ökosystem von JavaScript und Node.js, da es die Verwendung von Open-Source-Software und die Zusammenarbeit zwischen Entwicklern erleichtert.

Kapitel 4

Anforderungen

In diesem Abschnitt sind die funktionalen und nicht-funktionalen Anforderungen der Software definiert. Für eine einfachere Übersicht sind die funktionalen Anforderungen (Use Cases) mit folgendem Farbcode priorisiert.




	Minimal Viable Product (MVP) Features
	Projekt-Features
	Produkt-Features

Tabelle 4.1: Use Cases - Farbcode

MVP - Definieren die per Meilenstein **M5-Alpha** zu implementierenden Anforderungen.

Projekt-Features - Anforderungen, die zusätzlich zu den MVP-Features im Projekt umgesetzt werden, solange die Zeit reicht.

Produkt-Features - Anforderungen, die nach dem abgeschlossenen Projekt weiterführend implementiert werden können.

4.1 Akteure

Die Akteure sind aus den Benutzergruppen, die in Kapitel 2.2.2 beschrieben sind, abgeleitet und hier nochmals in tabellarischer Form dargestellt.

Akteure	Beschreibung
Junior Designer	Akteur hat wenig Berufserfahrung, keine technische, sondern eine gestalterische Ausbildung und verfügt über ein kleines Vokabular bezüglich UI-Komponenten.
Senior Designer	Akteur hat viel Berufserfahrung, keine technische, sondern eine gestalterische Ausbildung und verfügt über ein ausgeprägtes Vokabular bezüglich UI-Komponenten.
Junior Developer	Akteur hat wenig Berufserfahrung, eine technische Ausbildung im Bereich Software-Entwicklung/Engineering und verfügt über ein grundlegendes technisches Vokabular.
Senior Developer	Akteur hat viel Berufserfahrung, eine technische Ausbildung im Bereich Software Entwicklung/Engineering und verfügt über ein ausgeprägtes technisches Vokabular.
Laie	Akteur ist weder Designer noch Software-Entwickler von Beruf, hat keine technische oder gestalterische Ausbildung und verfügt über ein minimales Vokabular bezüglich UI-Komponenten.
Product Manager oder Product Owner	Akteur hat ein technisches sowie gestalterisches Grundwissen, verfügt über Management-Erfahrung und hat ein breites technisches Vokabular.

Tabelle 4.2: Beschreibung Akteure

4.2.2 Use Cases

Die Reihenfolge (Nummerierung) der Use Cases stellt gleichzeitig die initiale Priorisierung dar. Die Priorität der Use Cases wurden regelmässig mit dem Auftraggeber besprochen und falls nötig neu vergeben. Spätere Änderungen wurden direkt auf dem Issue Board auf GitLab vorgenommen. Die Anforderungen wurden gemäss der Vorlage von Craig Larman [uc] formal ausformuliert.


UC01 Suchbedürfnisse	
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="flex-grow: 1;">Eine bestimmte Information aus der SDX-Bibliothek wird von einem Akteur gesucht.</div> <div style="text-align: right;">  </div> </div>	
Akteur	Alle Akteure
Beschreibung	Ein Akteur möchte Informationen zu einem bestimmten Thema aus der SDX-Bibliothek erhalten. Ein Beispiel: Ein Developer hat sich in einem Projekt für einen Tech-Stack entschieden, möchte zur Unterstützung in der Gestaltung des Produktes SDX-Komponenten verwenden und möchte möglichst schnell Informationen zur Integration von SDX in das Projekt erhalten und sucht entsprechend mit Schlüsselwörtern aus dem Bereich Frontendtechnologien (Angular, React, etc.). Weitere Proto-Journeys von Akteuren sind in Kapitel 2.2.3 aufgeführt.
Wichtigstes Erfolgsszenario	Der Akteur gibt einen Suchbegriff im zur Verfügung stehenden Suchfeld ein und findet die von ihm gesuchten Informationen.
Alternatives Erfolgsszenario	Auch wenn der Akteur den Suchbegriff, der zur Ziel-Webseite führen würde, nicht vollständig korrekt eingibt, wird die Ziel-Webseite gefunden und als Suchresultat angezeigt.
Notwendige UC	-
Detailspezifikation	<ul style="list-style-type: none"> • Es steht eine Suche zur Verfügung, welche jegliche Akteure bedienen können. • Für MVP: <ul style="list-style-type: none"> – Die Suche durchsucht relevante Header-Elemente. – Darstellung der Suchresultate entspricht dem Auto-Complete-Dropdown. – Das Suchfeld soll beim Öffnen einer Webseite den Fokus haben. • Für Projekt-Features: <ul style="list-style-type: none"> – Die Suche durchsucht alle Elemente gemäss Konfiguration.

Tabelle 4.3: Beschreibung Use Case 01: Projektintegration

UC02 Synonyme

Vokabular des Akteurs ist nicht deckungsgleich mit dem in SDX verwendeten.



Akteur	Laie, Junior Developer, Junior Designer
Beschreibung	Ein Akteur kennt die von SDX angebotenen UI-Komponenten unter einem anderen Namen und findet diese deswegen nicht über die Navigation.
Wichtigstes Erfolgsszenario	Der Akteur erhält als Suchresultat über die Eingabe des Synonyms die Webseite der gesuchten Komponente.
Notwendige UC	UC01
Detailspezifikation	<ul style="list-style-type: none">• Während dem Aufbau der Struktur der durchsuchbaren Daten und somit auch während der Suche werden Synonyme von Open-UI [ope] miteinbezogen.• Zudem kann, falls gewünscht, der Thesaurus von API Ninjas [the] miteinbezogen werden.

Tabelle 4.4: Beschreibung Use Case 02: Synonyme

UC03 Autocomplete

Suchergebnisse sollen bereits während dem Eintippen des Suchbegriffs angezeigt werden.



Akteur	Alle Akteure
Beschreibung	Ein Akteur möchte bereits erste Suchresultate während dem Eintippen seines Suchbegriffs erhalten. Dies kann einerseits den Akteur bei der Suche unterstützen und andererseits kann der Suchprozess damit beschleunigt werden, weil unter Umständen nicht der gesamte Suchbegriff eingegeben werden muss.
Wichtigstes Erfolgsszenario	Der Akteur erhält bereits während dem Eintippen des Suchbegriffs passende Suchresultate angezeigt und muss somit eine Sucheingabe nicht zu Ende tippen.
Alternatives Erfolgsszenario	Dem Akteur werden während dem Eintippen des Suchbegriffs Vorschläge für seine Suche angezeigt. Somit wird der Akteur bei der Eingabe unterstützt, in dem ihm das in der SDX-Bibliothek verwendete Vokabular angezeigt wird.
Notwendige UC	UC01, UC02
Detailspezifikation	<ul style="list-style-type: none">• Wenn die Sucheingabe eine Länge von 2 oder grösser hat, wird die Suche im Hintergrund bereits gestartet und erste Suchresultate werden angezeigt.• Optional: In den Autocomplete-Suchergebnissen werden auch Bilder von Komponenten miteinbezogen, um den Akteur bezüglich des Vokabulars weiter zu unterstützen.

Tabelle 4.5: Beschreibung Use Case 03: Autocomplete

UC04 Keine Suchresultate

Ein Akteur möchte informiert werden, wenn keine Suchergebnisse zu einem Suchbegriff gefunden werden können.



Akteur	Alle Akteure
Beschreibung	Ein Akteur möchte eine entsprechende Information angezeigt erhalten, falls keine passenden Suchergebnisse gefunden wurden, damit er seine Suche gegebenenfalls anpassen kann.
Wichtigstes Erfolgsszenario	Ein Akteur wird darauf aufmerksam gemacht, dass mit seinem Suchbegriff keine Treffer auf SDX-Website gefunden werden konnten. Der Akteur kann somit darauf reagieren und einen anderen Suchbegriff oder eine andere Herangehensweise für die Suche (z.B. Navigation oder visuelle Übersicht) verwenden.
Notwendige UC	UC01, UC03
Detailspezifikation	<ul style="list-style-type: none">• Wenn zu einer Sucheingabe keine Resultate gefunden werden können, wird ein entsprechender Hinweis im Dropdown der Autocomplete-Suchresultate angezeigt.

Tabelle 4.6: Beschreibung Use Case 04: Keine Suchresultate

UC05 Konfigurierbare Suche

Ein Akteur möchte die Konfiguration der Suche nach seinen Bedürfnissen anpassen.



Akteur	Developer
Beschreibung	Ein Akteur möchte die Suche nach seinen Bedürfnissen, im Rahmen der vom Such-Library bereitgestellten Konfigurationsmöglichkeiten, konfigurieren.
Wichtigstes Erfolgsszenario	Der Akteur kann die Ergebnisse der Suche beeinflussen. Damit können falsche oder unpassende Resultate minimiert und der Suchprozess beschleunigt werden.
Alternatives Erfolgsszenario	Der Akteur kann entscheiden welche Features angewendet (Synonyme, Thesaurus) und in welcher Form die für die Suche benötigten Daten zur Verfügung gestellt werden sollen (in Webseiten eingebettet, als separate Datei).
Notwendige UC	UC01
Detailspezifikation	<ul style="list-style-type: none">• Es wird eine Standardkonfiguration bereitgestellt.• Die Standardkonfiguration wird angewendet, wenn keine vom User erstellte Konfiguration angegeben wird.• Die Standardkonfiguration kann durch eine vom User erstellte Konfiguration überschrieben werden.• Die Konfigurationsmöglichkeiten umfassen alle, vom Such-Framework/-Library zur Verfügung gestellten Optionen.

Tabelle 4.7: Beschreibung Use Case 05: Konfiguration Suche

UC06 Kategorisierte Suchresultate


Ein Akteur möchte die Suchresultate kategorisiert einsehen.



Akteur	Alle Akteure
Beschreibung	Ein Akteur ist nebst den gefundenen Webseiten für seine Suchanfrage auch daran interessiert zu welcher Kategorie eine Webseite gehört.
Wichtigstes Erfolgsszenario	Die Suchresultate werden kategorisiert angezeigt und der Akteur sieht somit bereits während der Suche zu welcher Kategorie die einzelnen Suchresultate gehören.
Alternatives Erfolgsszenario	Der Akteur kann die Suchresultate von einzelnen Kategorien zuklappen und somit die angezeigten Suchresultate einschränken.
Notwendige UC	UC01
Detailspezifikation	<ul style="list-style-type: none">• Die Suchresultate werden kategorisiert angezeigt.• Die Suchresultate einer Kategorie können auf- und zugeklappt werden.

Tabelle 4.8: Beschreibung Use Case 06: Kategorisierte Suchresultate

UC07 Zusätzliche Informationen zu Suchresultaten

Ein Akteur möchte zu einem Suchresultat weitere Informationen erhalten, ohne dabei die Webseite des Suchresultats öffnen zu müssen. 

Akteur	Alle Akteure
Beschreibung	Ein Akteur hat verschiedene Suchresultate zu seiner Suchanfrage erhalten und ist sich nicht sicher, welches der Suchresultate nun das richtige für seinen Anwendungsfall ist. Der Akteur möchte gerne zusätzliche Informationen zu den Suchresultaten erhalten, ohne dazu die Webseiten der Suchresultate öffnen zu müssen.
Wichtigstes Erfolgsszenario	Dem Akteur werden zusätzliche Informationen zu den Suchresultaten eingeblendet und er kann anhand dieser Informationen entscheiden, welches das korrekte Suchresultat für seinen Anwendungsfall ist.
Notwendige UC	UC01
Detailspezifikation	<ul style="list-style-type: none">• Im Dropdown-Eintrag des Suchresultats sollen zusätzliche Informationen dargestellt werden. Ideen sind:<ul style="list-style-type: none">– Icon, das die Kategorie des Suchresultats repräsentiert.– Zusätzliche Information zum Suchresultat in textueller Form.

Tabelle 4.9: Beschreibung Use Case 07: Zusätzliche Informationen zu Suchresultaten

UC08 Suchfilter

Ein Akteur möchte nicht die gesamte SDX-Website durchsuchen, sondern nur bestimmte Kategorien.



Akteur	Alle Akteure
Beschreibung	Ein Akteur ist nur an der Durchsuchung einer oder mehreren bestimmten Kategorien interessiert. Hierzu möchte der Akteur auswählen können, dass nur in bestimmten Kategorien nach seiner Suchanfrage gesucht wird.
Wichtigstes Erfolgsszenario	Der Akteur kann über die Suchfilter die zu durchsuchenden Kategorien festlegen.
Alternatives Erfolgsszenario	Der Rechen- und somit Zeitaufwand für die Suche kann durch die Reduktion der zu durchsuchenden Webseiten reduziert werden.
Notwendige UC	UC01
Detailspezifikation	<ul style="list-style-type: none">• Es wird pro definierte Kategorie ein Toggle angezeigt.• Standardmässig sind alle Kategorien angewählt.• Durch ein Klick auf ein Toggle kann eine Kategorie abgewählt, respektive wieder angewählt werden.

Tabelle 4.10: Beschreibung Use Case 08: Suchfilter

UC09 Suchhistorie


Einem Akteur sollen bereits verwendete Suchbegriffe angezeigt werden.



Akteur	Alle Akteure
Beschreibung	Ein Akteur möchte nochmals nach einem Suchbegriff suchen, den er bereits verwendet hatte. Dazu sollen dem Akteur die zuletzt eingegebenen Suchbegriffe angezeigt werden.
Wichtigstes Erfolgsszenario	Ein Akteur möchte ein Suchbedürfnis befriedigen, ihm fällt jedoch der passende Suchbegriff nicht ein. Er weiss jedoch, dass er schon einmal das gleiche Suchbedürfnis hatte und damals fündig geworden ist. Somit kann er den bereits verwendeten Suchbegriff aus der Suchhistorie auswählen.
Alternatives Erfolgsszenario	Der Zeitaufwand eines Akteurs beim Suchen wird dadurch verringert, dass eine bereits durchgeführte Suche nicht noch einmal eingetippt werden muss.
Notwendige UC	UC01
Detailspezifikation	<ul style="list-style-type: none">• Beim Klick in das leere Suchfeld wird ein Dropdown mit den vergangenen Suchbegriffen angezeigt.• Die Suchbegriffe, die entweder aus der Autocomplete-Liste ausgewählt wurden oder nach denen effektiv gesucht wurde (ENTER-Taste oder Klick auf das Suchsymbol), werden im Local Storage des Browsers abgelegt.• Idee: Erfolgreiche Einträge aus Suchhistorie als Autocomplete-Vorschläge vorschlagen.

Tabelle 4.11: Beschreibung Use Case 09: Suchhistorie


UC10 Visuelle Unterstützung bei der Komponenten-Suche

Aufgrund eines kleinen Vokabulars bezüglich UI-Komponenten möchte ein Akteur die gewünschte Komponente aus einer visuellen Übersichtsdarstellung aller Komponenten auswählen. 

Akteur	Laie, Junior Developer, Junior Designer
Beschreibung	Der Akteur soll in seinem Suchbedürfnis visuell unterstützt werden, da ihm das Vokabular fehlt.
Wichtigstes Erfolgsszenario	Der Akteur sucht eine bestimmte Komponente, weiss jedoch nicht wie sie heisst. Durch die visuelle Unterstützung kann er die richtige Komponente identifizieren.
Alternatives Erfolgsszenario	Dadurch, dass der Akteur die UI-Komponente über die visuelle Übersicht findet, lernt er auch den in SDX für die UI-Komponente verwendeten Namen.
Notwendige UC	-
Detailspezifikation	<ul style="list-style-type: none"> • Es wird ein Übersichts-Dropdown/Pop-Up eingeblendet, das alle UI-Komponenten der SDX-Bibliothek visuell auflistet.
Bemerkungen	<p>Dieser Use Case wurde im Verlauf des Projektes ausgeschlossen. Die Gründe für den Ausschluss sind:</p> <ul style="list-style-type: none"> • Aufwand für die Erstellung der Grafiken zu hoch wäre. • Wartungsaufwand zur Pflege der Grafiken.

Tabelle 4.12: Beschreibung Use Case 10: Visuelle Unterstützung bei Komponenten-Suche


UC11 Verzeichnisstruktur des Outputs

Der Benutzer des Search-Integrators möchte die Struktur des Input-Verzeichnisses in seinem Output-Verzeichnis wiedergeben. 

Akteur	Junior Developer, Senior Developer
Beschreibung	Die Verzeichnisstruktur des Inputs bei der Generation der aufbereiteten Files wieder aufgebaut (rekursiv).
Wichtigstes Erfolgsszenario	Der Benutzer muss keine Pfadanpassungen vornehmen.
Alternatives Erfolgsszenario	Ist der Pfad des Input-Verzeichnisses gleich wie der des Output-Verzeichnisses, wird die Struktur nicht angepasst und die vorhanden Dateien einfach überschrieben.
Notwendige UC	-
Detailspezifikation	<ul style="list-style-type: none">• Der Pfad der zu bearbeitenden Daten wird bei der Iteration über die Verzeichnisse mitgespeichert.• Bei der Generation der bearbeiteten Files wird der Dateipfad berücksichtigt und das Verzeichnis analog des Input-Verzeichnisses abgebildet.

Tabelle 4.13: Beschreibung Use Case 11: Verzeichnisstruktur des Outputs


UC12 Logging und Auswertung

Die Suchfunktionalität wird durch ein serverseitiges Logging erweitert. Auf Basis der dokumentierten Suchanfragen und Fehlern werden Statistiken erzeugt. Diese können zur kontinuierlichen Verbesserung des Suchalgorithmus oder zum Nachvollziehen von Verhalten genutzt werden. 

Akteur	Junior Developer, Senior Developer
Beschreibung	Setzt der User eine Suchanfrage ab, wird diese weiterhin auf dem Client ausgeführt. Es geschieht jedoch asynchron ein Logging der Suchanfrage an den Server. Serverseitig wird eine Logik genutzt, um diese Informationen auszuwerten. Die gewonnenen Erkenntnisse werden genutzt, um die Suche zu optimieren.
Wichtigstes Erfolgsszenario	Die Suche kann aufgrund der Auswertungen und den daraus resultieren Informationen (Statistiken) optimiert werden.
Alternatives Erfolgsszenario	Es kann verifiziert werden, ob die Suche ordnungsmässig funktioniert.
Notwendige UC	UC01, UC05
Detailspezifikation	<ul style="list-style-type: none">• Auf einem Webserver werden die benötigten Endpoints zur Verfügung gestellt.• Auf dem Webserver wird die Logik zur Behandlung von Request und Auswertung der Daten implementiert.• Die Sicherheitsrisiken werden erneut analysiert und Massnahmen zur Absicherung der Kommunikation sowie Bereinigung des Inputs vorgenommen.• Bei einer Suchanfrage werden die benötigten Informationen an die zur Verfügung gestellten Endpoints übermittelt.

Tabelle 4.14: Beschreibung Use Case 12: Logging und Auswertung

UC13 Anbindung Webcrawler

Es wird ein Webcrawler in das Projekt integriert, welcher dazu verwendet werden kann, Webseiten zu indexieren. 

Akteur	Junior Developer, Senior Developer
Beschreibung	
Wichtigstes Erfolgsszenario	Die Suche kann live-Daten berücksichtigen. Dies im Vergleich zu den aktuell indexierten HTML-Dateien, die unter Umständen zu einem späteren Zeitpunkt (nach der Integration der Suche) noch mit zusätzlichen Daten angereichert werden, welche entsprechend für die Suche nicht zur Verfügung stehen.
Alternatives Erfolgsszenario	-
Notwendige UC	-
Detailspezifikation	<ul style="list-style-type: none">• Der Webcrawler soll eine Webseite nach den vorhandenen Webpages durchsuchen.• Aus den gefundenen Webpages soll das HTML extrahiert und gespeichert werden (Scraping).• Die extrahierten Webpages dienen als Input für den Parser und werden entsprechend für die Suche indexiert.

Tabelle 4.15: Beschreibung Use Case 13: Anbindung Webcrawler

4.3 Nicht-Funktionale Anforderungen

Als Grundlage für die Identifizierung der nicht-funktionalen Anforderungen (NFRs) dient der ISO-Standard ISO25010 [iso] und die OWASP Top 10 [OWA].

Der ISO-Standard ISO25010 besteht aus 8 Kategorien, die je in weitere Unterkategorien unterteilt sind. Im Identifizierungsprozess der NFRs wurden alle Kategorien inklusive Unterkategorien auf die Relevanz in Bezug auf das Projekt geprüft. Dabei wurden die für die Suche in der SDX-Designbibliothek relevanten NFRs identifiziert und in diesem Kapitel aufgeführt. Eine Auflistung der Analyseresultate bezüglich allen Kategorien inklusive Unterkategorien vom ISO-Standard ISO25010 ist in Tabelle 4.29 einsehbar.

Die OWASP Top 10 ist ein Standard-Dokument, welches die kritischsten Web-Applikation-Sicherheitsrisiken auflistet. Diese Liste wird regelmässig aktualisiert. Im Identifizierungsprozess wurden alle 10 Risiken der aktuellen OWASP Top 10 geprüft. Das Resultat dieser Analyse ist in Tabelle 4.30 zu finden.

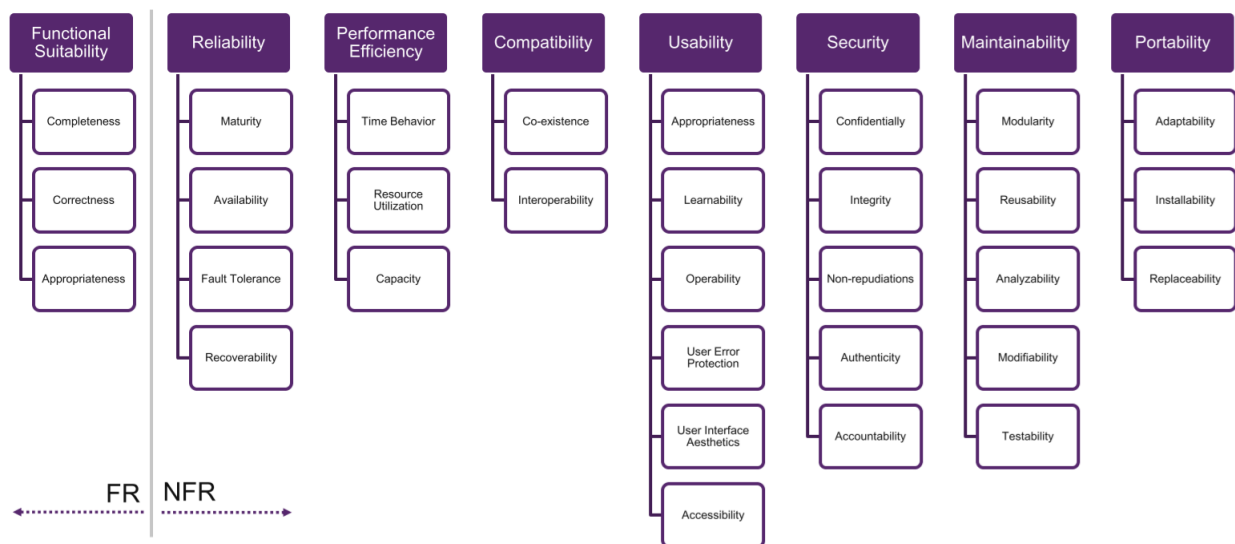


Abbildung 4.2: Kategorien inkl. Unterkategorien gemäss ISO25010-Standard

Die Priorisierung der identifizierten NFRs erfolgt anhand der folgenden Skala:

1. Hohe Priorität (MUSS)
2. Mittlere Priorität (SOLLTE)
3. Tiefe Priorität (KÖNNTE)

ID	NFR01
Kategorie	Usability
Unterkategorie	Operability
Szenario	Die Suche sowie die Anzeige der Suchresultate ist intuitiv verständlich und liefert den Usern die erwünschten Suchergebnisse.
Auslöser	Ein User, der die Such-Funktionalität noch nie genutzt hat, kommt auf die SDX-Website mit einem Suchbedürfnis.
Reaktion	Der User kann die Such-Funktionalität ohne Unterstützung einer Drittperson nutzen und sein Suchbedürfnis befriedigen.
Messkriterium	Es werden Usability-Tests mit Benutzern von verschiedenen Benutzergruppen (siehe Tabelle 2.1) durchgeführt. Den Usern werden verschiedene Aufgaben in Form von Szenarien gestellt. Dabei wird im Durchschnitt in $\geq 80\%$ der Aufgaben das gesuchte Resultat gefunden.
Priorität	1

Tabelle 4.16: Beschreibung NFR01: Operability (Usability)

ID	NFR02
Kategorie	Performance Efficiency
Unterkategorie	Time Behavior
Szenario	Die Such-Funktionalität wird in die bestehende SDX-Design-Bibliothek integriert. Die Time-to-Interactive der Website wird auf modernen Geräten minimal, falls überhaupt, verlängert.
Auslöser	Die SDX-Website wird aufgerufen.
Reaktion	Die Website wird nach der Integration der Such-Funktionalität gleich schnell oder nur minimal langsamer geladen als zuvor.
Messkriterium	Es werden Performance-Messungen bezüglich der Time-to-Interactive der Website und der einzelnen Webseiten vor und nach der Integration der Such-Funktionalität durchgeführt. Die durchschnittliche Ladezeit ist nach der Integration der Such-Funktionalität höchstens 10% länger.
Priorität	1

Tabelle 4.17: Beschreibung NFR02: Time Behavior (Performance Efficiency)

ID	NFR03
Kategorie	Compatibility
Unterkategorie	Co-Existence
Szenario	Die Such-Funktionalität stellt keine Anforderungen an Webseiten von SDX in Bezug auf die erfolgreiche Ausführung des Build-Prozesses der Website. D.h., auch wenn eine Webseite in den Build-Prozess der SDX-Website miteinbezogen wird, die nicht für die Suche optimiert ist, funktioniert der Build-Prozess der SDX-Website weiterhin.
Auslöser	Eine neue Webseite wird in die SDX-Design-Bibliothek eingebunden, ohne dass die Suche für die Implementation der Webseite berücksichtigt wurde.
Reaktion	Der Build-Prozess der SDX-Website funktioniert weiterhin. Zudem wird die neue Webseite von der Suche berücksichtigt.
Messkriterium	<p>Es wird eine neue Webseite in die SDX-Website integriert, die nichts von der Such-Funktionalität weiss. Folgende Kriterien werden dabei erfüllt:</p> <ol style="list-style-type: none"> 1. Der Build-Prozess der Website ist weiterhin erfolgreich. 2. Das Suchfeld ist auf der neuen Webseite vorhanden. 3. Mit geeigneten Suchwörtern wird die neue Webseite über die Suche gefunden. <p>Folgende Voraussetzungen für die Webseite müssen erfüllt sein, damit sie in der Suche gefunden wird:</p> <ul style="list-style-type: none"> • Die Webseite muss valides HTML enthalten. • Die Webseite muss sich an die Best Practices von SEO halten.
Priorität	1

Tabelle 4.18: Beschreibung NFR03: Co-Existence (Compatibility)

ID	NFR04
Kategorie	Performance Efficiency
Unterkategorie	Time Behavior & Resource Utilization
Szenario	Die (clientseitige) Suche auf der SDX-Website soll von Benutzern als performant wahrgenommen werden. Die Such-Funktionalität muss somit effizient implementiert sein, sodass sie trotz den beschränkten Ressourcen im Browser effizient ist.
Auslöser	Ein User löst eine Suche aus.
Reaktion	Die Suchresultate werden dem User praktisch umgehend angezeigt.
Messkriterium	Bei den Usability-Tests werden die Teilnehmenden gebeten die Performanz der Suche zu bewerten. Im Durchschnitt wird die Suche als mindestens 'gut' beurteilt. Des Weiteren werden Performance-Messungen durchgeführt. Dabei wird das Suchresultat in $\geq 80\%$ der Messungen in ≤ 0.5 Sekunden angezeigt.
Priorität	1

Tabelle 4.19: Beschreibung NFR04: Time Behavior & Resource Utilization (Performance Efficiency)

ID	NFR05
Kategorie	Portability
Unterkategorie	Replaceability
Szenario	Für den Fall, dass das eingesetzte Such-Framework nicht mehr weiterentwickelt wird und/oder Sicherheitsrisiken birgt, soll dieses möglichst einfach ausgetauscht werden können. Hierzu wird das verwendete Such-Framework hinter einem Interface gekapselt.
Auslöser	Das eingesetzte Such-Framework muss ausgetauscht werden.
Reaktion	Es ist möglich das verwendete Such-Framework auszutauschen, ohne dass Client-Code verändert werden muss. Zudem sollen nur minimale Anpassungen im Such-Integrator-Code zum Austausch des Such-Frameworks notwendig sein.
Messkriterium	Beim Alpha- sowie beim Beta-Release wird verifiziert, dass die eingesetzte Such-Bibliothek hinter einem Interface gekapselt ist und dass ein direkter Zugriff auf die Library ausschliesslich in der Framework-spezifischen Implementation vom Interface vorhanden ist. Wenn dies der Fall ist, gilt dieses NFR als erfüllt.
Priorität	1

Tabelle 4.20: Beschreibung NFR05: Replaceability (Portability)

ID	NFR06
Kategorie	Usability
Unterkategorie	User Error Protection
Szenario	Bei der Suche wird nicht der exakte Suchbegriff verwendet. Trotzdem soll das Suchbedürfnis eines Users befriedigt werden können.
Auslöser	Der User gibt einen Suchbegriff ein, der nicht exakt so auf der gesuchten Webseite vorhanden ist.
Reaktion	Die gesuchte Webseite wird vom User trotzdem gefunden.
Messkriterium	Es wird nach Webseiten gesucht, wobei jeweils nicht die exakten Suchbegriffe verwendet werden. Konkret wird für die folgenden Szenarien getestet, dass die Suchbedürfnisse befriedigt werden können. <ol style="list-style-type: none"> 1. Der Suchbegriff wird in Einzahl sowie in Mehrzahl formuliert. 2. Es wird nach einem Teilwort (mindestens 70%) des exakten Begriffs gesucht. 3. Es wird nach einem gängigen Synonym eines Begriffes gesucht.
Priorität	2

Tabelle 4.21: Beschreibung NFR06: User Error Protection (Usability)

ID	NFR07
Kategorie	Portability
Unterkategorie	Adaptability
Szenario	Die Such-Funktionalität funktioniert in allen modernen Browsern, die von SDX unterstützt werden. Konkret sind das Google Chrome, Apple Safari, Mozilla Firefox und Microsoft Edge (Chromium). Zudem funktioniert sie sowohl auf Desktop- sowie auch auf mobilen Endgeräten.
Auslöser	Die Such-Funktionalität wird in verschiedenen Browsern und auf unterschiedlichen Gerätetypen ausgeführt.
Reaktion	Die Such-Funktionalität funktioniert wie spezifiziert.
Messkriterium	Es werden Systemtests, sowie alle Testfälle der Usability-Tests, auf verschiedenen Geräten mit verschiedenen Browsern durch das Projektteam durchgeführt. Dabei können die Szenarien der Usability-Tests auf allen Endgeräten und Browsern ohne technische Einschränkungen durchgeführt werden.
Priorität	2

Tabelle 4.22: Beschreibung NFR07: Adaptability (Portability)

ID	NFR08
Kategorie	Usability
Unterkategorie	User Interface Aesthetics
Szenario	Die Darstellung des Suchfeldes sowie der Suchresultate ist sowohl auf mobile Devices sowie auf Desktop-Geräte optimiert.
Auslöser	Die Suche wird auf unterschiedlichen Gerätetypen ausgeführt.
Reaktion	Die Darstellung des Suchfeldes und der Suchresultate ist auf das jeweilige Endgerät optimiert.
Messkriterium	Die Testfälle der Usability-Tests werden auf mobilen Endgeräten sowie auf Desktop-Computern durch das Projektteam durchgeführt. Zudem wird bei der Durchführung der Usability-Tests auf die Darstellung geachtet. Die Bewertung der Darstellung der Suche inklusive Suchresultate wird von den Teilnehmern der Usability-Tests im Durchschnitt mindestens als 'gut' eingestuft.
Priorität	2

Tabelle 4.23: Beschreibung NFR08: User Interface Aesthetics (Usability)

ID	NFR09
Kategorie	Maintainability
Unterkategorie	Testability
Szenario	Die Business Logik des Webseiten-Parsers sowie der Such-Funktionalität weist eine Code-Coverage von $\geq 85\%$ auf.
Auslöser	Die Unit-Tests werden ausgeführt.
Reaktion	Die Metrik bezüglich Code-Coverage ist $\geq 85\%$ für die Business Logik des Webseiten-Parsers sowie der Such-Funktionalität.
Messkriterium	Die Unit-Tests werden ausgeführt. Danach wird verifiziert, dass die Code-Coverage $\geq 85\%$ für die Business-Logik des Webseiten-Parsers sowie der Such-Funktionalität erreicht.
Priorität	2

Tabelle 4.24: Beschreibung NFR09: Testability (Maintainability)

ID	NFR10
Kategorie	Security
Unterkategorie	Confidentiality
Szenario	Die Such-Funktionalität verwendet aktuelle Versionen der eingesetzten Libraries. Es muss nicht zwingend immer die aktuellste Version verwendet werden, jedoch dürfen nur Versionen verwendet werden, für welche keine Sicherheitsrisiken bekannt sind. Das bedeutet, es dürfen keine Versionen verwendet werden, die veraltet sind. Die Prüfung aller eingesetzten Libraries findet periodisch statt. Dabei wird die periodische Prüfung während des Projekts durch das Projektteam durchgeführt. Bei der Übergabe in den Betrieb wird vom Projektteam eine Wartungsempfehlung an die Swisscom abgegeben.
Auslöser	Für eine verwendete Library steht eine neue Version zur Verfügung.
Reaktion	Die Aktualisierung der Library auf die neue Version wird geprüft. Falls man sich dazu entscheidet, dass ein Update der Library gemacht werden soll, wird die Library auf die neue Version aktualisiert und es wird verifiziert, dass die Applikation weiterhin wie spezifiziert funktioniert.
Messkriterium	Beim Beta-Meilenstein wird verifiziert, dass von allen eingesetzten Libraries eine aktuelle Version verwendet wird. Bei der Übergabe des Projekts an den Auftraggeber wird eine Wartungsempfehlung abgegeben.
Priorität	2

Tabelle 4.25: Beschreibung NFR10: Confidentiality (Security)

ID	NFR11
Kategorie	Maintainability
Unterkategorie	Modifiability & Modularity
Szenario	Die Such-Funktionalität ist so aufgebaut, dass zusätzliche Funktionen zur Verfügung gestellt werden können, ohne dass der Code in einer nicht von den neuen Funktionen tangierten Komponenten angepasst werden muss.
Auslöser	Die Such-Funktionalität soll durch eine neue Funktionalität ergänzt werden.
Reaktion	Die neue Funktionalität wird in die bestehende Website eingebaut, ohne dass grosse Änderungen in nicht tangierten Komponenten notwendig sind.
Messkriterium	Nach dem Alpha-Release werden bis zum Beta-Release zusätzliche Funktionen für die Such-Funktionalität eingebaut. Dabei wird verifiziert, dass die Website genügend modular aufgebaut ist, sodass kein oder nur wenig Code in nicht tangierten Komponenten angepasst werden muss.
Priorität	3

Tabelle 4.26: Beschreibung NFR11: Modifiability & Modularity (Maintainability)

ID	NFR12
Kategorie	Usability
Unterkategorie	Accessibility
Szenario	Der Anwender des Frameworks möchte die komplette Kontrolle über die Darstellung des Suchfeldes sowie der Darstellung der Resultate.
Auslöser	Die gewünschte Darstellung und die Anforderungen bezüglich Accessibility des Suchfeldes und der Suchresultate ist von Nutzer zu Nutzer unterschiedlich.
Reaktion	Der Anwender des Frameworks hat die komplette Kontrolle über die Darstellung des Suchfelds sowie der Suchresultate. Somit hat der User die volle Kontrolle über die Accessibility seiner Website.
Messkriterium	Nach dem Alpha-Release werden bis zum Beta-Release Optimierungen an der Darstellung des Suchfelds sowie der Darstellung der Suchresultate durchgeführt. Dabei wird verifiziert, dass es möglich ist die komplette Komponente auszutauschen und ob alle nötigen Informationen zur Darstellung vorhanden sind.
Priorität	3

Tabelle 4.27: Beschreibung NFR12: Accessibility (Usability)

ID	NFR13
Kategorie	Usability
Unterkategorie	Learnability
Szenario	Falls zu einer Suchanfrage keine Resultate gefunden werden konnten, wird dies dem User angezeigt, sodass der User die Möglichkeit hat seine Suchanfrage umzuformulieren. Zudem werden Synonyme eingesetzt, um breitere Suchanfragen zu ermöglichen.
Auslöser	Für eine Suchanfrage wird kein Suchresultat gefunden.
Reaktion	Erst wird geprüft ob der Nutzer ein Synonym verwendet. Ist dies der Fall, wird anhand des Synonyms gesucht. Dem User wird eine informative und handlungsorientierte Meldung angezeigt, sofern Suchresultate gefunden werden.
Messkriterium	Für die Usability-Tests werden Szenarien entwickelt, die keine Suchresultate erzielen. $\geq 60\%$ der Usability-Test-Teilnehmenden versuchen die Suche mit einem anderen Suchbegriff und/oder Synonym weiterzuführen. Es wird erkannt, dass keine relevanten Informationen auf der Website enthalten sind.
Priorität	3

Tabelle 4.28: Beschreibung NFR13: Learnability (Usability)

4.3.1 Analyse aller ISO25010-Kategorien und -Unterkategorien

Es wurde nicht für jede Unterkategorie aus dem ISO25010 [iso] ein NFR definiert. Tabelle 4.29 listet für jede Unterkategorie aus dem ISO25010 auf, ob ein NFR dafür definiert wurde. Falls kein NFR für eine Unterkategorie definiert wurde, wird in Tabelle 4.29 ausgeführt, wieso die Unterkategorie für die Such-Funktionalität im SDX nicht relevant ist.

Kategorie	Subkategorie	Verifikation NFR
Reliability	Maturity	In 4.16 & 4.24 enthalten.
Reliability	Availability	Da die Suche in die bestehende Design-Bibliothek integriert wird, ist die Availability mit dem SDX gekoppelt und somit nicht im Scope dieses Projekts.
Reliability	Fault Tolerance	In 4.21 & 4.28 enthalten.
Reliability	Recoverability	Da die Suche in die bestehende Design-Bibliothek integriert wird, ist die Recoverability mit dem SDX gekoppelt und somit nicht im Scope dieses Projekts.
Performance Efficiency	Time Behaviour	Spezifiziert in 4.17 & 4.19.
Performance Efficiency	Resource Utilization	Spezifiziert in 4.19.
Performance Efficiency	Capacity	In in 4.17 & 4.19 enthalten.
Compatibility	Co-existence	Spezifiziert in 4.18.
Compatibility	Interoperability	Nicht relevant für den Scope des Projekts, da keine Informationen mit anderen Systemen ausgetauscht werden.
Usability	Appropriateness	In 4.16 enthalten.
Usability	Learnability	Spezifiziert in 4.28.
Usability	Operability	Spezifiziert in 4.16.
Usability	User Error Protection	Spezifiziert in 4.21.

Usability	User Interface Aesthetics	Spezifiziert in 4.23.
Usability	Accessibility	Spezifiziert in 4.27.
Security	Confidentiality	Spezifiziert in 4.25.
Security	Integrity	Nicht relevant für das Search-Integrator-Framework, da die gesamte Client-Server-Kommunikation weiterhin unter der vollen Kontrolle des Anwenders liegt.
Security	Non-Repudiation	Nicht relevant für den Scope des Projekts, da über die Suche keine Daten verändert werden.
Security	Authenticity	Nicht relevant für den Scope des Projekts, da für die Suche keine Authentisierung notwendig ist.
Security	Accountability	Nicht relevant für den Scope des Projekts, da über die Suche keine Daten verändert werden.
Maintainability	Modularity	Spezifiziert in 4.26.
Maintainability	Reusability	In 4.26 enthalten.
Maintainability	Analyzability	In 4.17 enthalten.
Maintainability	Modifiability	Spezifiziert in 4.26.
Maintainability	Testability	Spezifiziert in 4.24.
Portability	Adaptability	Spezifiziert in 4.22.
Portability	Installability	In 4.26 enthalten.
Portability	Replaceability	Spezifiziert in 4.20.

Tabelle 4.29: Auslistung aller NFR Kategorien gemäss ISO25010

4.3.2 Analyse der OWASP Top 10

In der folgenden Tabelle wird ausgeführt, ob für ein Risiko der OWASP Top 10 [OWA] ein NFR definiert wurde. Falls für ein Risiko der OWASP Top 10 kein NFR definiert wurde, wird begründet, wieso dies der Fall ist.

Risiko	Verifikation NFR
A01: 2021-Broken Access Control	Nicht relevant, da die Such-Funktionalität uneingeschränkt zugänglich ist.
A02: 2021-Cryptographic Failures	Nicht im Scope des Projekts, da die Such-Funktionalität in die bestehende SDX-Website integriert wird und somit auch als Ganzes an den Client (verschlüsselt) übertragen wird.
A03: 2021-Injection	Nicht relevant, da kein vom User eingegebener Input ausgeführt, ausgegeben oder auf den Server zurückgespielt wird.
A04:2021-Insecure Design	Nicht im Scope des Projekts, da die Such-Funktionalität in ein bestehendes System integriert wird.
A05:2021-Security Misconfiguration	Nicht im Scope des Projekts, da die Such-Funktionalität in ein bestehendes System integriert wird.
A06:2021-Vulnerable and Outdated Components	Spezifiziert in 4.25.
A07:2021-Identification and Authentication Failures	Nicht im Scope des Projekts, da für die Suche keine Identifikation und Authentisierung notwendig ist.
A08:2021-Software and Data Integrity Failures	Spezifiziert in 4.25.
A09:2021-Security Logging and Monitoring Failures	Nicht relevant, da über die Such-Funktionalität keine Daten verändert werden.
A10:2021-Server-Side Request Forgery	Nicht im Scope des Projekts, da die Such-Funktionalität in ein bestehendes System integriert wird.

Tabelle 4.30: Auflistung aller OWASP Top 10 Risiken

4.4 Adressierung nicht-funktionaler Anforderungen

4.4.1 Adressierung der NFR per Alpha-Release

Bis zum Alpha-Release wurden Massnahmen bezüglich den NFR mit Priorität 1 (hohe Priorität) eingeleitet (vergl. Kapitel 7.2). Tabelle 4.31 zeigt den Stand der NFR per Alpha-Release.

NFR	Stand
NFR01	Massnahmen wurden eingeleitet. Die Verifikation ist offen, da noch keine Usability-Tests durchgeführt wurden.
NFR02	Es wurde darauf geachtet, dass die zusätzliche Grösse, welche durch die Integration einer Suche zu einer Webseite hinzugefügt wird, stetig gemonitored wird. Performance-Messungen werden für den Beta-Release durchgeführt.
NFR03	Beim Alpha-Release wurden Systemtests durchgeführt, siehe Tabelle 10.1. Diese haben gezeigt, dass bei der Erweiterung einer Website um eine Webseite die Such-Integration weiterhin sauber funktioniert.
NFR04	Messungen haben gezeigt, dass Suchergebnisse zu einer Suchanfrage zu 100% in unter 0.5 Sekunden angezeigt werden. Die Verifikation durch Usability-Tests ist noch offen, da noch keine Usability-Tests durchgeführt wurden.
NFR05	Die eingesetzte Such-Library wurde mithilfe des Software-Pattern 'Fassade' abgekapselt und wird ausschliesslich im Framework genutzt, indem gegen das Interface programmiert wird. Siehe auch Abschnitt 5.2.5.
NFR06	Manuelle Tests haben gezeigt, dass folgende Punkte erfüllt sind: <ol style="list-style-type: none">1. Suchbegriffe können in Ein- und Mehrzahl formuliert werden.2. 70% eines Suchbegriffs, sofern dieser aus mehr als zwei Buchstaben besteht, reichen für die Suche nach einem bestimmten Begriff aus.3. Synonyme werden gefunden.
NFR07	Die Systemtests, siehe Tabelle 6.5, wurden durchgeführt und haben dieses NFR bestätigt. Die Verifikation durch Usability-Tests ist noch offen, da noch keine Usability-Tests durchgeführt wurden.
NFR08	Verifikation offen, da noch keine Usability-Tests durchgeführt wurden.
NFR09	Aktuell liegt die Test-Coverage unter den angestrebten 85%. NFR09 ist noch nicht erfüllt.

NFR10	Wird beim Beta-Release verifiziert, siehe Tabelle 4.32.
NFR11	Wird beim Beta-Release verifiziert, siehe Tabelle 4.32.
NFR12	Wird beim Beta-Release verifiziert, siehe Tabelle 4.32.
NFR13	Verifikation offen, da noch keine Usability-Tests durchgeführt wurden.

Tabelle 4.31: Stand der NFR per Alpha-Release

4.4.2 Adressierung der NFR per Beta-Release

Tabelle 4.32 zeigt den Stand der NFR per Beta-Release.

NFR01	Die Durchführung der Usability-Tests haben gezeigt, dass die Operability erfüllt ist. Die Testpersonen konnten die Suche intuitiv nutzen und fanden die Suche verständlich. Die Nutzer konnten alle Szenarien lösen und fanden immer die gesuchten Informationen bzw. erkannten das keine Informationen vorhanden waren. Von den Testpersonen gewünschte Optimierungen bezüglich Positionierung, Gestaltung und Nachvollziehbarkeit der Suchergebnisse können dank der Konfigurierbarkeit des Search-Integrator-Frameworks einfach angepasst werden. Diese Detailarbeiten werden jedoch durch den Auftraggeber nach der Übergabe selbst vorgenommen.
NFR02	Die im Kapitel 8.1.2 dargelegten Auswertungen bezüglich Performance bestätigen dieses NFR.
NFR03	Beim Beta-Release wurden Systemtests durchgeführt, siehe Tabelle 10.2. Diese haben gezeigt, dass bei der Erweiterung einer Website um eine Webseite die Such-Integration weiterhin sauber funktioniert.
NFR04	Die Verifikation durch Usability-Test war erfolgreich. Die Testpersonen bestätigten die Performance und lobten diese sogar explizit. Die Suchergebnisse werden in Realtime dargestellt und es ist für den Anwender keine Ladezeit erkennbar.
NFR05	Die eingesetzte Library ist hinter einer Fassade isoliert und ist ausschliesslich im Search-Integrator-Framework vorhanden.
NFR06	Manuelle Tests haben gezeigt, dass folgende Punkte erfüllt sind: <ol style="list-style-type: none"> 1. Suchbegriffe können in Ein- und Mehrzahl formuliert werden. 2. 70% eines Suchbegriffs, sofern dieser aus mehr als zwei Buchstaben besteht, reichen für die Suche nach einem bestimmten Begriff aus. 3. Die manuell eingepflegten Synonyme, sowie die über eine API bezogenen, werden berücksichtigt und können über die Suche gefunden werden.
NFR07	Die Systemtests, siehe Tabelle 10.2, wurden durchgeführt und haben dieses NFR bestätigt. Vor der Durchführung der Usability-Tests wurden alle Szenarien vom Projektteam in den spezifizierten Browsern mit zwei verschiedenen Desktopgeräten sowie zwei mobilen Geräten durchgespielt. Dabei konnten keine technischen Einschränkungen beobachtet werden.

NFR08	Die Darstellung konnte mithilfe von Systemtest, siehe 6.5, verifiziert werden. Auf Desktop- sowie mobilen Geräten werden die Suchresultate optimiert dargestellt. Zudem wurde die Darstellung in den Usability-Tests gelobt und als gut empfunden.
NFR09	Die Test-Coverage liegt aktuell bei >85%. Dieses NFR ist entsprechend erfüllt.
NFR10	Im Projekt sind nur benötigte und aktuelle Dependencies enthalten. Das Projektteam hat dies manuell verifiziert. Eine Wartungsempfehlung wurde erstellt.
NFR11	Die Erweiterung des Frameworks stellte keine Probleme dar. Die einzelnen Komponenten wurden entkoppelt und sind somit unabhängig.
NFR12	Zwischen dem Alpha- und dem Beta-Release wurden verschiedenste Optimierungen an der Darstellung vorgenommen - dies hatte problemlos funktioniert. Zudem wurde eine Demo-Applikation aufgesetzt, in der eine andere Such-Komponente als im SDX-Anwendungsbeispiel verwendet wurde.
NFR13	Die Auswertung der Usability-Tests hat gezeigt, dass die Testenden schnell bemerkten, ob eine Suchanfrage erfolgreich war oder nicht. Die Meldung, welche angezeigt wird sofern keine Suchresultate vorhanden waren, wurde richtig verstanden und interpretiert. Testende haben daraufhin mit alternativen Begriffen oder Synonymen weitergesucht.

Tabelle 4.32: Stand der NFR per Beta-Release

Kapitel 5

Lösungsdesign

In diesem Kapitel wird zuerst das Konzept, welches die Grundlage für die Implementation des Search-Integration-Frameworks bildet, beschrieben. Danach wird die Architektur der Suchfunktionalität aufgezeigt. Im Anschluss werden Design- und technische Entscheide beschrieben sowie die Evaluation und der Entscheid für das eingesetzte Such-Framework (Fuse.js [\[fus\]](#)) aufgezeigt. Der Abschluss dieses Kapitels bilden die ausgearbeiteten Wireframes, die als Grundlage für das Design der Lösung dienen.

5.1 Konzept

Konzeptuell wird der Prozess des Search-Integration-Frameworks in zwei Phasen unterteilt:

- Phase 1: Datenextraktion, -anreicherung und -aufbereitung
- Phase 2: Effektive Suche

Die beiden Phasen des Prozesses sind in Grafik 5.1 ersichtlich. Die Unterscheidung in zwei Phasen basiert auf dem Zeitpunkt zu, sowie dem Ort an, welchem die Aktionen ausgeführt werden. Phase 1 beinhaltet Vorbereitungs- und Integrationsarbeiten, die während dem Build-Prozess und somit auf dem Server der Website ausgeführt werden. Phase 2, die effektive Suche, wird aufgrund einer Suchanfrage eines Benutzers ausgelöst und findet im Webbrowser des Clients statt.

Der grosse Vorteil der Auslagerung der Vorbereitungsarbeiten für die Suchfunktionalität in eine eigene Phase, die während dem Build-Prozess der Website ausgeführt wird, ist, dass für diese Arbeiten keine hohen Anforderungen bezüglich Performance existieren. Der Grund hierfür ist, dass diese Arbeiten zu einem Zeitpunkt durchgeführt werden, zu dem keine Benutzerinteraktion stattfindet.

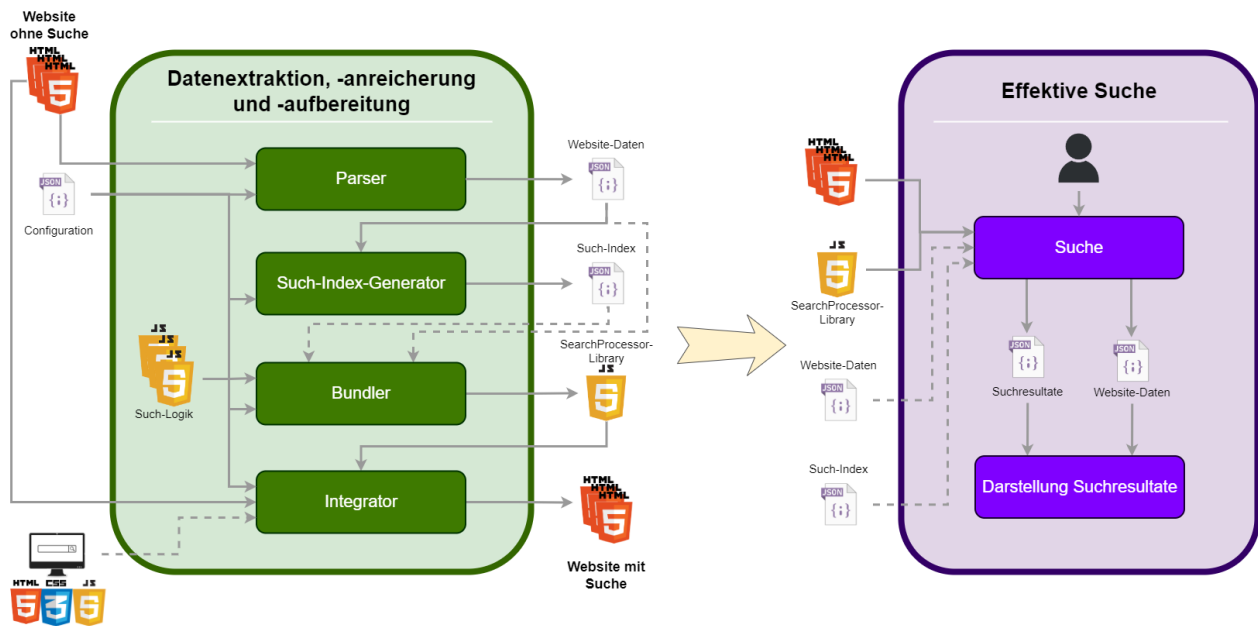


Abbildung 5.1: Konzept zur Integration der Suche

Die gestrichelten Linien repräsentieren optionale Inputs. Das heisst, dass diese für die Integration der Suchfunktionalität auf der Website nicht zwingend benötigt werden. Ausgezogene Linien verbildlichen zwingend notwendige Inputs. Aufgrund dessen, dass die Suchkomponente nicht zwingend während dem Prozess der Such-Integration in die Webseiten eingebunden werden muss, und dass die Website-Daten sowie der Such-Index entweder direkt in die SearchProcessor-Library eingebunden oder vom Client separat abgeholt werden können, entstehen vier verschiedene Variationen des Konzepts. Diese Variationen sind in Abschnitt 5.1.4 aufgezeigt.

5.1.1 Phase 1: Datenextraktion, -anreicherung und -aufbereitung

Der Input der ersten Phase beinhaltet die Website, eine Konfigurationsdatei, sowie optional die Suchkomponente (HTML, CSS und JS für die Darstellung des Suchfeldes und der Suchresultate). Der erste Schritt in Phase 1 ist die Analyse aller Webseiten und das Extrahieren der relevanten Informationen durch den Parser. Aufgrund der extrahierten Informationen wird durch den Such-Index-Generator ein Index erstellt, der für eine performante Suche benötigt wird. Als Nächstes wird durch den Bundler eine JavaScript-Datei basierend auf der Konfigurationsdatei sowie der im Such-Integration-Framework beinhalteten Suchlogik erstellt. Dieses Bundle wird als Library mit dem Namen 'SearchProcessor' zur Verfügung gestellt und beinhaltet als Einstiegspunkt eine Funktion, welche für die effektive Suche in Phase 2 verwendet wird. Dabei können optional die Website-Daten sowie der Such-Index in das Bundle integriert werden. Der letzte Schritt der ersten Phase ist die Integration der SearchProcessor-Library in alle Webseiten. Optional kann durch den Integrator auch die Suchkomponente in alle Webseiten integriert werden.

In den nachfolgenden Abschnitten werden die einzelnen Komponenten der ersten Phase im Detail vorgestellt.

Parser

Der Parser ist für die Extraktion der relevanten Informationen (Website-Daten) aus den Webseiten zuständig. Für die Beurteilung, welche Informationen als relevant eingestuft werden, hält sich der Parser an Best Practices aus SEO und Semantic HTML. Über die Konfigurationsdatei kann verfeinert werden, welche Informationen auf den Webseiten als relevant eingestuft oder ignoriert werden. Das Resultat des Parsers ist eine Liste von Records, welche allen möglichen Suchresultaten der Website entspricht. Standardmässig wird pro Webseite ein Record erstellt. Es ist jedoch möglich zu konfigurieren, dass nicht nur pro Webseite, sondern zusätzlich für jedes Section- und Article-HTML-Element ein Record erstellt wird. Somit ist es möglich Suchresultate für einzelne Teile einer Webseite zu erhalten und direkt auf diese Abschnitte zu Linken. Der Parser ist zudem in der Lage die extrahierten Records mit Synonymen anzureichern. Hierzu kann entweder eine Liste mit Synonymen definiert, oder der Parser kann angewiesen werden die Thesaurus-API von API Ninjas [\[the\]](#) zu konsultieren. Eine weitere Funktionalität, die vom Parser zur Verfügung gestellt wird, ist das Parsen einer Breadcrumb-Liste [\[bre\]](#). Falls diese Option aktiviert ist, stehen die Informationen aus den Breadcrumb-Listen der Webseiten für die Darstellung der Suchresultate zur Verfügung. Der Input des Parsers ist ein oder mehrere Verzeichnisse, die HTML-Dateien beinhalten sowie die Konfigurationsdatei. Der Output ist eine JSON-Datei, die alle erstellten Records beinhaltet.

Such-Index-Generator

Der Such-Index-Generator erstellt auf den durch den Parser generierten Records eine Datenstruktur, die für eine effiziente und performante Suche verwendet wird. Zudem beinhaltet der Such-Index Informationen bezüglich der Priorisierung der einzelnen Suchresultate. Die Funktionalität für das Erstellen des Index wird durch das für die Suche eingesetzte Framework, Fuse.js, zur Verfügung gestellt (siehe Kapitel 5.3). Der Input des Such-Index-Generators sind die vom Parser generierten Records sowie die Konfigurationsdatei. Der Output ist eine JSON-Datei, die den erstellten Such-Index beinhaltet.

Bundler

Der Bundler wird für das Erstellen einer einzigen JavaScript-Datei verwendet, welche die gesamte Suchlogik sowie die Konfiguration beinhaltet. Zudem kann konfiguriert werden, dass auch die Website-Daten und der Such-Index in das Bundle gepackt werden. Das Bundle wird als Library mit dem Namen 'SearchProcessor' zur Verfügung gestellt und beinhaltet die gesamte Funktionalität für die Ausführung einer clientseitigen Suche. Falls die Website-Daten und der Such-Index nicht in das Bundle reingepackt werden, wird die Logik in der SearchProcessor-Library so angepasst, dass die beiden Dateien beim Laden der ersten Webseite nachgeladen und im Session-Storage des Browsers abgelegt werden. Der Grund für die Konfigurierbarkeit bezüglich der Integration der beiden Dateien in das Bundle ist, dass dies eine signifikante Auswirkung auf die Grösse des resultierenden Bundles hat. Als Bundling-Software wird Webpack [\[webb\]](#) verwendet. Der Input des Bundlers ist die Konfigurationsdatei, die Suchlogik des Such-Integration-Frameworks sowie optional die Website-Daten und der Such-Index. Der Output ist eine einzige JavaScript-Datei: die SearchProcessor-Library.

Integrator

Der Integrator ist eine generische Komponente, die HTML-Komponenten, JavaScript-Skripte sowie CSS-Stylesheets in Webseiten integrieren kann. Standardmässig wird der Integrator für das Platzieren eines Skripts mit der SearchProcessor-Library verwendet. Zudem kann der Integrator für das Integrieren der Suchkomponente verwendet werden. Hierzu können über die Konfigurationsdatei die für die Suchkomponente relevanten Dateien definiert werden. Da der Integrator generisch implementiert ist, kann er auch für die Integration von HTML-Komponenten, JavaScript-Skripten oder CSS-Stylesheets verwendet werden, die keinen Zusammenhang mit der Suche haben. Der Input des Integrators ist ein oder mehrere Verzeichnisse, die HTML-Dateien beinhalten, die Konfigurationsdatei, das Bundle der SearchProcessor-Library sowie optional die Dateien der Suchkomponente. Der Output besteht aus allen HTML-Dateien, die in den Input-Verzeichnissen gefunden wurden. Dabei wurden die HTML-Dateien gemäss der Konfiguration des Integrators angereichert.

5.1.2 Phase 2: Effektive Suche

Phase 2 wird im Webbrowser auf dem Client aufgrund einer Suchanfrage durch einen Benutzer ausgeführt und ist somit Performance-kritisch. Aus diesem Grund wird nur das Notwendige in dieser Phase platziert. Jegliche Logik, die nicht zwingend zu diesem Zeitpunkt ausgeführt werden muss, wird in Phase 1 ausgelagert. Die effektive Suche besteht aus zwei Schritten: dem effektiven Suchen und der Darstellung der Suchresultate.

Suchen

Der erste Schritt von Phase 2 ist die effektive Suche aufgrund eines von einem User eingegebenen Suchbegriffes im Suchfeld. Hierzu stellt die SearchProcessor-Library, die auf dem Client verfügbar ist, eine Such-Funktion zur Verfügung, die als Input einen Suchbegriff sowie eine Callback-Funktion entgegennimmt. Für die Suche wird, der in Phase 1 generierte Such-Index durchsucht. Die übergebene Callback-Funktion wird durch die SearchProcessor-Library mit einer Liste der gefundenen Suchresultate sowie einer Liste aller Records aufgerufen. Die Suchresultate werden dabei in einer priorisierten Reihenfolge übergeben. Es ist möglich zu konfigurieren wie viele Suchresultate maximal zurückgegeben werden sollen und ob dabei eine Unterscheidung für mobile Devices und Desktop-Geräte gemacht werden soll.

Für die technische Umsetzung der effektiven Suche in Phase 2 sowie für den Aufbau des Such-Index in Phase 1 wird das Open-Source JavaScript-Framework Fuse.js [fus] eingesetzt. Während der Phase 'Elaboration' (vergl. Abschnitt 7.1) der Bachelorarbeit wurde eine ausführliche Analyse und Evaluation von zur Verfügung stehenden Frameworks durchgeführt. Das Resultat der Analyse hat gezeigt, dass für den Anwendungsfall dieser Bachelorarbeit das Framework Fuse.js am besten geeignet ist. Die Details der Analyse sind in Kapitel 5.3 beschrieben.

Darstellung der Suchresultate

Der zweite Schritt in Phase 2 ist die visuelle Darstellung der Suchresultate. Diese Darstellung ist typischerweise von Website zu Website unterschiedlich, denn das Design des Suchresultats soll an das

Design der jeweiligen Website angepasst sein. Aus diesem Grund wird diese Logik in der Suchkomponente definiert. Da die Suchkomponente von einem End-User des Search-Integrator-Frameworks selbst definiert wird, ist das Design der Darstellung der Suchresultate vollständig unter dessen Kontrolle. Die Integration der Suchkomponente und somit der Logik für die Darstellung der Suchresultate findet in Phase 1 statt (siehe Abschnitt 5.1.1). Für die SDX-Website wird in Absprache mit dem Auftraggeber eine SDX-Komponente für die Darstellung des Suchfeldes verwendet. Innerhalb der SDX-Komponente werden die Suchresultate optimiert dargestellt, sodass dem Benutzer die notwendigen Informationen zur Entscheidungsfindung, welches Suchresultat das geeignetste ist, zur Verfügung stehen.

5.1.3 Deployment

Das Search-Integrator-Framework wurde als NPM-Paket publiziert und ist konfigurierbar, sodass es für verschiedene Websites eingesetzt werden kann. Als Standard für die Berücksichtigung und Priorisierung der Elemente der Webseiten dienen Best Practices aus SEO und Semantic HTML (siehe Abschnitte 3.2.1 und 3.2.2). Dies hat zur Folge, dass für eine Webseite, die SEO-optimiert ist und Semantic HTML einsetzt, mit minimaler Konfiguration bereits gute Suchresultate erzielt werden können. Des Weiteren ist es möglich kontextspezifische Synonyme mitzugeben, welche beim Aufbau des Index für die Suche berücksichtigt werden. Zudem ist die Art der zur Verfügungsstellung der Website-Daten und des Such-Index konfigurierbar und es ist auch möglich eine Suchfeld- und Suchresultat-Komponente zu definieren, welche in die bestehenden Webseiten eingebettet wird.

Dieses Projekt löst somit nicht nur die spezifische Anforderung des Auftraggebers, dass die SDX-Website um eine Suchfunktionalität ergänzt werden soll, sondern stellt ein Werkzeug zur Verfügung, mit dem auf einer beliebigen statischen Website eine Suchfunktionalität hinzugefügt werden kann. Die Recherche und Analyse dieser Bachelorarbeit hat ergeben, dass zwar bereits viele Frameworks für eine Suchfunktionalität zur Verfügung stehen, jedoch der Prozess der Datenextraktion, -anreicherung und -aufbereitung meistens nicht oder zu wenig berücksichtigt wird (vergleiche Kapitel 3).

5.1.4 Variationen

Wie in Abbildung 5.1 aufgezeigt, gibt es in Phase 1 optionale Inputs für den Bundler sowie für den Integrator. Somit gibt es vier Variationen des Such-Integration-Prozesses, die in den nachfolgenden Abschnitten ausgeführt sind.

Mit Integration der Suchkomponente & ohne Bundling der JSON-Dateien

In der ersten Variante wird dem Integrator eine Suchkomponente übergeben und diese wird somit auf allen Webseiten platziert. Die beiden JSON-Dateien, die Website-Daten und der Such-Index, werden in dieser Variante nicht in das Bundle gepackt. Somit werden diese beiden Dateien beim Laden der ersten Webseite, welche die Such-Funktionalität enthält, von der SearchProcessor-Library nachgeladen und im Session-Storage des Browsers gespeichert. Abbildung 5.2 zeigt den Prozess dieser Variante.

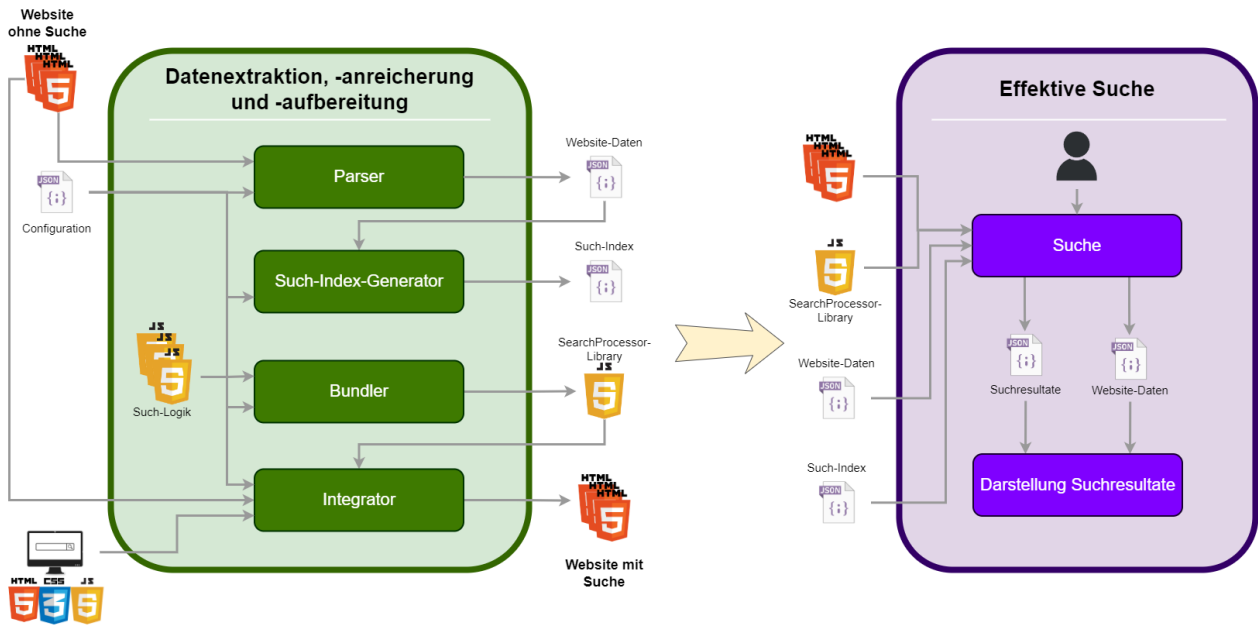


Abbildung 5.2: Konzept mit Suchkomponenten-Integration & ohne Bundling der JSON-Dateien

Mit Integration der Suchkomponente & mit Bundling der JSON-Dateien

In dieser Variante wird ebenfalls eine Suchkomponente mit dem Integrator auf allen Webseiten platziert. Zudem werden die Website-Daten sowie der Such-Index in das Bundle der SearchProcessor-Library integriert. Der resultierende Prozess ist in Abbildung 5.3 veranschaulicht. Die Website-Daten sowie der Such-Index müssen in dieser Variation in Phase 2 nicht explizit zur Verfügung gestellt werden, da diese bereits im Bundle der SearchProcessor-Library beinhaltet sind.

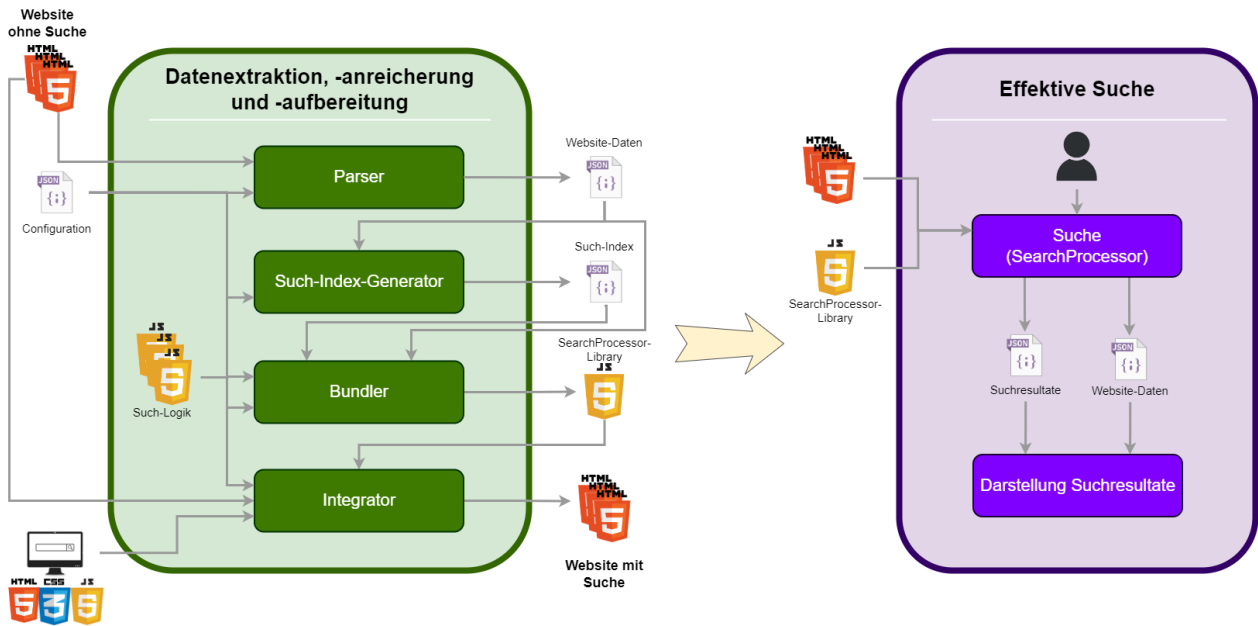


Abbildung 5.3: Konzept mit Suchkomponenten-Integration & mit Bundling der JSON-Dateien

Ohne Integration der Suchkomponente & ohne Bundling der JSON-Dateien

Der Prozess dieser Variante verläuft analog zu dem in Abbildung 5.2, jedoch ohne das Integrieren einer Suchkomponente in Phase 1. Das heisst, dass die Suchkomponente vor oder nach der Integration der Such-Funktionalität auf den Webseiten platziert werden muss. Diese Variation ist in Abbildung 5.4 dargestellt.

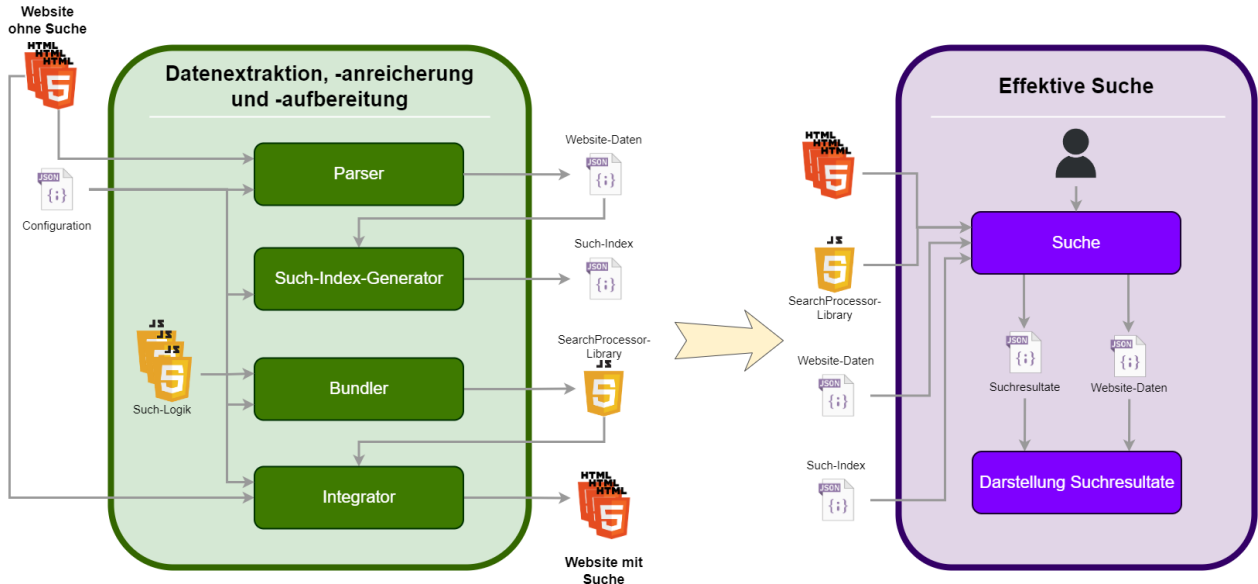


Abbildung 5.4: Konzept ohne Suchkomponenten-Integration & ohne Bundling der JSON-Dateien

Ohne Integration der Suchkomponente & mit Bundling der JSON-Dateien

Auch in dieser Variante wird keine Suchkomponente über den Integrator auf den Webseiten platziert. Jedoch werden im in Abbildung 5.5 dargestellten Prozess die beiden JSON-Dateien analog zu dem Prozess in Abbildung 5.3 in das Bundle der SearchProcessor-Library gepackt.

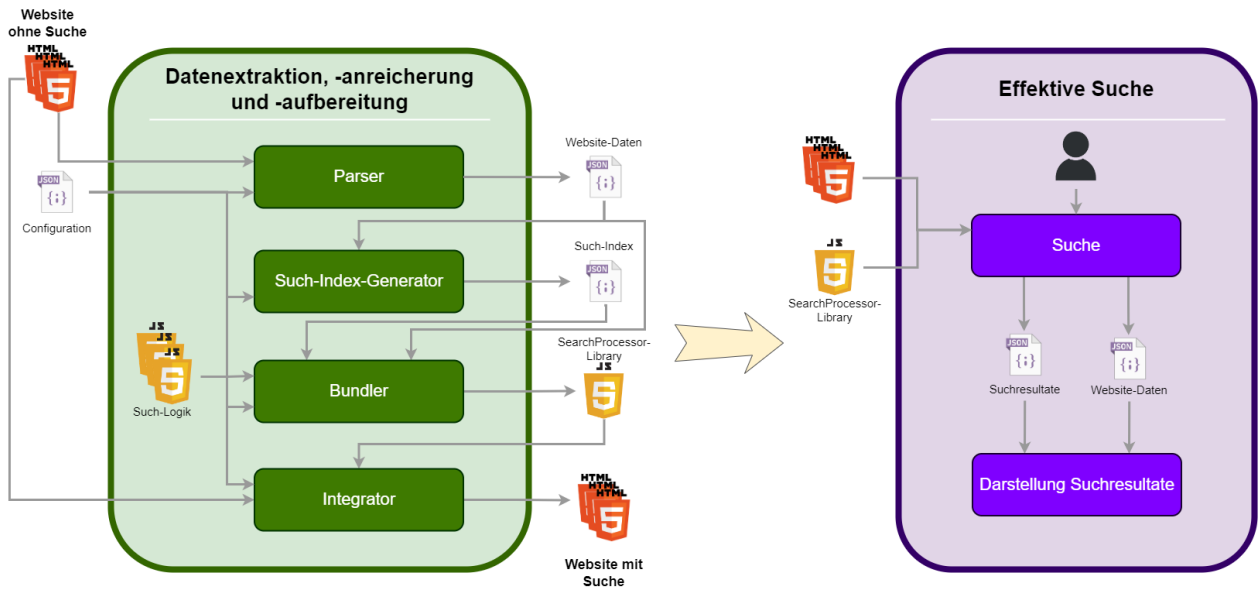


Abbildung 5.5: Konzept ohne Suchkomponenten-Integration & mit Bundling der JSON-Dateien

5.2 Architektur

In der klassischen Client-Server-Architektur sind typischerweise drei Instanzen vorhanden. Das Frontend welches auf dem Client läuft, ein Backend-System und eine Datenbank. Ein User nutzt das Frontend, als Schnittstelle um Informationen einzusehen oder die gebotene Funktionalität zu nutzen.

Folgende Grafik soll dies simplifiziert aufzeigen und die Aufgaben der einzelnen Teile kurz präsentieren.

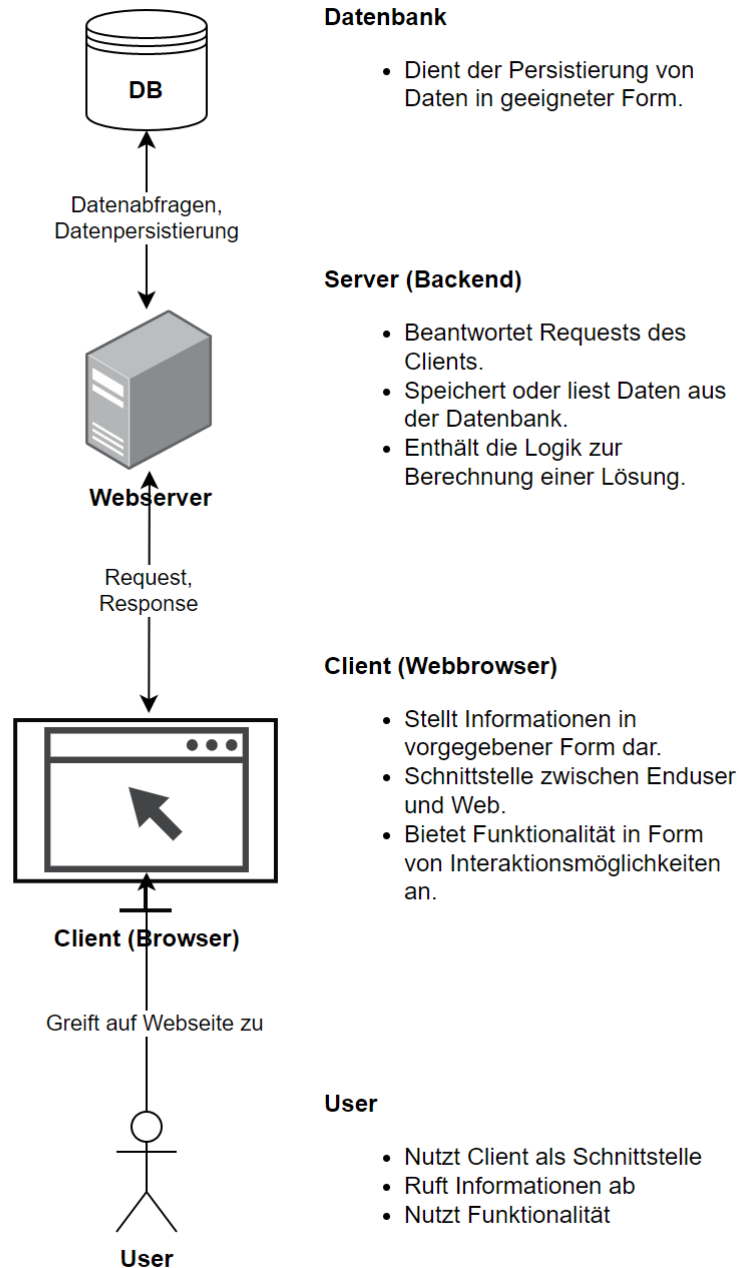


Abbildung 5.6: Klassische Client-Server Architektur

5.2.1 Architektur-Constraints

Vom Auftraggeber wurden folgende Anforderungen an die Architektur gestellt. Diese sind, wenn möglich, einzuhalten.

- Serverinfrastruktur soll nicht verändert werden.
- Integration der Suche soll während dem Build der SDX-Webseite durchgeführt werden können.
- Verwendung von SDX-Komponenten für die Darstellung des Suchfelds und der Suchresultate.

5.2.2 Umsetzungsvariante: Clientseitige Architektur

Die gewählte Architektur für das Such-Integration-Framework ist ein clientseitiger Ansatz. Dies bedeutet, dass die Datenbank sowie der Server als Recheninstanz wegfallen. Die Funktionalität sowie die Daten werden auf den Client verschoben und dort direkt zur Verfügung gestellt. Der Browser des Benutzers ist an dieser Stelle die Recheninstanz bzw. die Runtime Umgebung des Codes. Die Daten und die Funktionalität werden weiterhin vom Server bezogen, so wie auch die einzelnen Webseiten.

Folgende Grafik illustriert den clientseitigen Ansatz im Allgemeinen, sowie spezifisch für das Search-Integration-Framework.

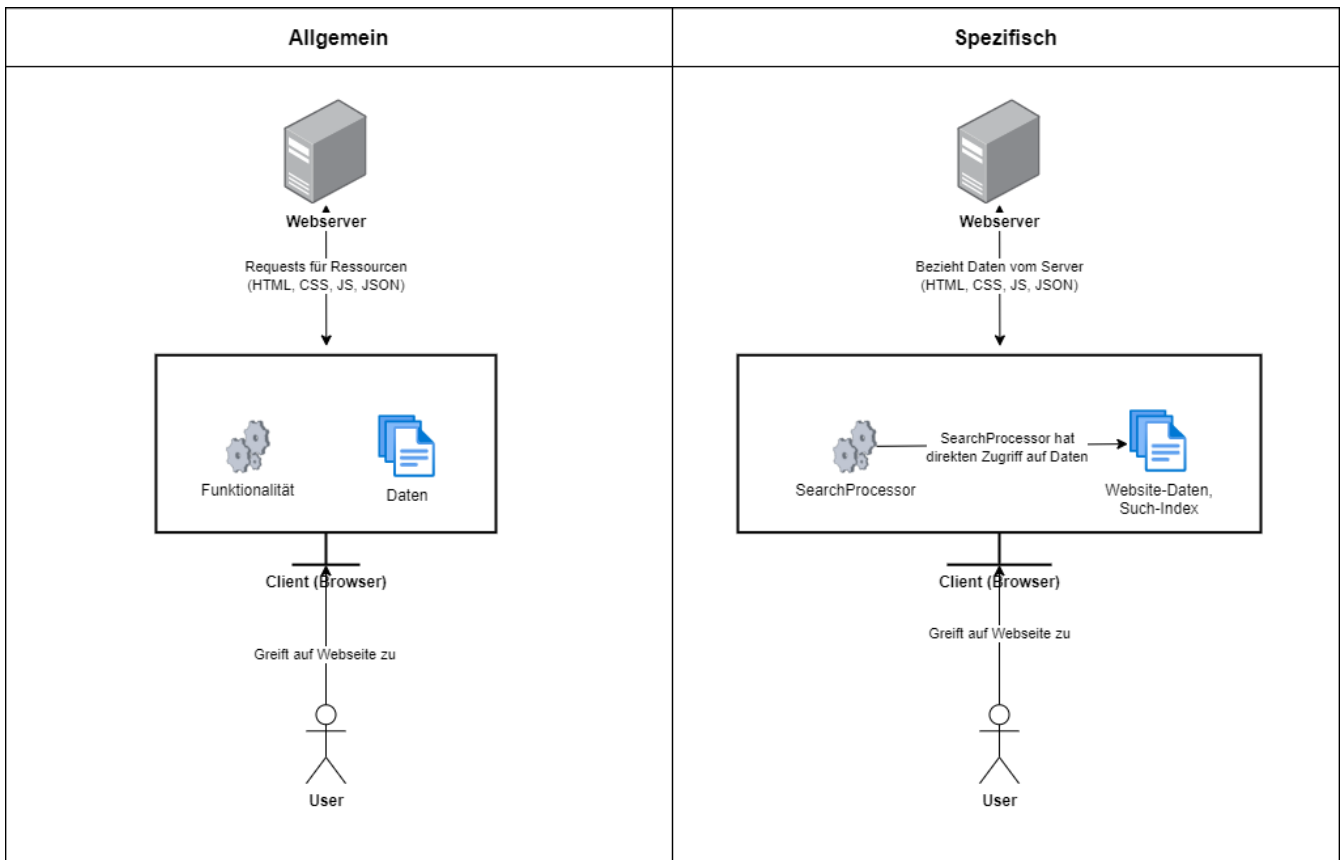


Abbildung 5.7: Clientseitige Architektur im allgemeinen und spezifisch dargestellt

Details bezüglich der Funktionsweise des Search-Integrator-Frameworks können im Abschnitt 5.1 eingesehen werden.

Vorteile eines clientseitigen Ansatzes für das Search-Integrator-Framework Die Vorteile des clientseitigen Ansatzes sind hier nochmals kurz aufgelistet. Genauere Informationen können in Kapitel 3.1.2 eingesehen werden.

1. Schnelle Suche
2. Offline-Suche
3. Geringere Komplexität
4. Server- und Netzwerkentlastung
5. Geringer Integrationsaufwand
6. Natur der SDX-Website

5.2.3 Design-Entscheide

Tabelle 5.1 zeigt die während dem Projekt getroffenen Design-Entscheide und begründet diese.

Entscheid	Begründung
Suche anstatt Filterung und Sortierung	<p>Im Projekt wurden folgende alternative Lösungsansätze einer Suche untersucht:</p> <ul style="list-style-type: none"> • Einfache Filterung • Einfache Sortierung • Filter und Sortierfunktion <p>Diese Lösungsansätze erwiesen sich jedoch bezüglich Benutzerfreundlichkeit als schlechter, da sie schwerer zu verstehen und mühsamer in der Bedienung sind. Die Navigation wird weniger gut unterstützt und der Aufwand, um ein bestimmtes Element oder Eintrag zu finden, ist im Vergleich zu einer Suche erhöht.</p>
Automatische Indexierung	<p>Es wird eine automatische Indexierung vorgenommen damit der Wartungs- und Integrationsaufwand möglichst klein bleibt. Zudem bietet dies die Möglichkeit eine generische Lösung auf Basis eines Standards (SEO, Semantic HTML) umzusetzen.</p>

Verwendung von SDX-Komponenten	<p>Sofern in diesem Projekt ein UI gebaut wird, sei dies in einem Prototypen oder für eine Demo, werden SDX-Komponenten verwendet. Bei der Integration der Suche in die SDX-Website am Ende des Projekts werden ebenfalls SDX-Komponenten verwendet und daher macht es Sinn dies während dem Projektverlauf gleich zu tun. Zudem bieten die SDX-Komponenten bereits den gewünschten Standard bezüglich Accessibility, Design und Funktionalität.</p>
Darstellung des Suchfelds und der Suchresultate	<p>Für die Darstellung des Suchfelds und der Suchresultate wurde das Konzept aus Abschnitt 5.4.1 verwendet. Grund dafür ist, dass für die Realisierung der visuellen Darstellung SDX-Komponenten verwendet werden sollen. Dies wurde vom Auftraggeber so gewünscht. Suchfilter, Suchhistorie, Hover-Informationen, ein Mega-Dropdown, sowie eine Komponenten-Übersicht wurden vom Auftraggeber explizit abgelehnt, da eine klassische Suchfunktion gewünscht wurde. Eine Kategorisierung der Suchresultate sowie die Darstellung von zusätzlichen Optionen in der SDX-Komponente wurde vom Auftraggeber als Option definiert.</p>
Umsetzung einer generischen Lösung	<p>Die erarbeitete Lösung wurde generisch gelöst. Das bedeutet, dass die Lösung nicht nur auf den Auftraggeber angepasst ist, sondern als Framework für alle Projekte, welche den Anforderungen an das Framework entsprechen, genutzt werden kann. Dazu wird der Auftraggeber-spezifische und der Framework-spezifische Code entkoppelt und separiert. Dies hat den Vorteil, dass die Flexibilität erhöht wird, da die Lösung auf den Use Case eines Projektes angepasst werden kann.</p>
Deployment	<p>NPM [npma], siehe auch Abschnitt 3.4.1, ist eine weit verbreitete Plattform für die Veröffentlichung von Software, da sie eine einfache Paketverwaltung, Versionierung und Integration in Projekte ermöglicht. Aus diesen Gründen haben wir uns für die Publizierung auf NPM entschieden.</p>

Tabelle 5.1: Design-Entscheide

5.2.4 Technische Entscheide

Tabelle 5.1 zeigt die während dem Projekt getroffenen technischen Entscheide und begründet diese.

Entscheid	Begründung
Verwendung eines/r Frameworks/Library für die Suche vs. Eigenimplementation	Für die Such-Funktionalität wird eine bestehende Lösung (Framework/Library) verwendet. Von einer eigens entwickelte Lösung wird abgesehen, weil die Implementation zu aufwändig wäre und die gewünschte Qualität der Lösung nicht garantiert werden könnte.
Verwendung von Fuse.js	Für die Such-Funktionalität wird Fuse.js eingesetzt. Fuse.js ist eine leichtgewichtige Fuzzy-Suchbibliothek ohne Abhängigkeiten [fus]. Die in Abschnitt 5.3 beschriebene Evaluation sowie die Konstruktion der Prototypen haben gezeigt, dass Fuse.js die bestmögliche Wahl ist. Fuse.js steht unter einer Apache 2.0 Lizenz [apa].
Clientseitiger Ansatz	<p>Es wurde eine clientseitige Suche umgesetzt. Diese wurde als optimale Lösung für eine statische Website mit kleiner Datenbasis erkannt. Bei dieser Art von Website ist die Verwendung einer serverseitigen Suche ein Overkill, da die Suche effektiv Webseiten und keine Datenbank durchsucht. Dies kann problemlos auf der Client-Seite ausgeführt werden. Eine serverseitige Suche bietet in diesem Kontext keinen wirklichen Mehrwert.</p> <p>Eine clientseitige Suche ist einfacher zu implementieren, ohne Abhängigkeiten und erfordert keine zusätzlichen Systeme, da sie im Browser der Nutzer ausgeführt wird. Die Suchanfragen werden von der Website empfangen und vom Browser interpretiert, wodurch die Auslastung des Servers minimiert wird. Zusätzlich kann eine clientseitige Suche von verschiedenen JavaScript-basierten Bibliotheken unterstützt werden. Die Verwendung einer clientseitigen Suche ermöglicht es auch, eine bessere Benutzererfahrung durch schnellere Suchergebnisse zu bieten, da die Suchanfrage direkt vom Browser des Benutzers ausgeführt wird. Auf diese Weise können Benutzer schnell und effektiv nach Informationen suchen, die sie benötigen, und somit ein gutes Sucherlebnis genießen.</p>

<p>Verwendung von Mocha & Chai</p>	<p>Als Testing-Framework wird Mocha [moca] zusammen mit Chai [cha] als Assertion Library eingesetzt. Grundlage für den Entscheid war die Statistik von State of JavaScript 2022 bezüglich der Verbreitung von JavaScript-Testing-Frameworks [sta]. Die vier verbreitetsten JavaScript-Testing-Frameworks sind Jest [jesa], Storybook [sto], Mocha [moca] und Cypress [Cyp].</p> <p>In der engeren Auswahl standen Mocha und Jest. Grund für den Ausschluss von Storybook war, dass Storybook auf das Testing von UI-Komponenten ausgelegt ist und im Projekt ein Unit-Testing-Framework benötigt wurde. Cypress ist auf End-to-End-Testing und Komponenten-Testing spezialisiert und wurde deshalb ebenfalls ausgeschlossen.</p> <p>Bei der Wahl zwischen Mocha und Jest war die Unterstützung von ECMAScript-Modulen der springende Punkt, da diese im Projekt aufgrund der Vorgaben des Auftraggebers eingesetzt werden. Während Jest nur eine experimentelle Unterstützung für ECMAScript-Module anbietet (vergleiche [jesb]), unterstützt Mocha den Einsatz von ECMAScript-Modulen vollumfänglich (siehe [moch]).</p> <p>Mocha & Chai stehen unter einer MIT-Lizenz [mit].</p>
<p>Logging und kontinuierliche Verbesserung</p>	<p>Es wurde gegen ein Logging und gegen die kontinuierliche Verbesserung entschieden. Das Logging von Suchanfragen und Fehlern zu Auswertung und anschließenden Optimierung von Suchresultaten, hätte eine Client-Serverarchitektur gefordert. Dies wollte man nicht, aufgrund des Aufwands der dadurch entstehen würde. Der Clientseitige Ansatz würde untergraben und die Vorteile minimiert. Es wurde ein Product-Feature aufgenommen, welches in Zukunft bei Bedarf durch den Nutzer implementiert werden kann, siehe 4.14.</p>
<p>node-html-parser</p>	<p>Zur Bearbeitung von HTML wurde der node-html-parser verwendet. Die Entscheidung wurde getroffen, da der node-html-parser viele Vorteile im Vergleich zur Bearbeitung mithilfe von Strings bietet, da direkt die bekannten Funktionen des HTML DOM verwendet werden können. Zudem ist der node-html-parser sehr klein im Vergleich zu alternativen Produkten wie beispielsweise JSDOM. Ebenso ist die Handhabung des node-html-parser sehr einfach, da er sehr kompakt und simple aufgebaut ist.</p> <p>node-html-parser steht unter einer MIT-Lizenz [mit].</p>

UUID	Zur eindeutigen Identifizierung der durch den Parser generierten Records werden Universally Unique Identifiers (UUIDs) verwendet. Der Grund dafür ist, dass die UUIDs als RFC-Standard standardisiert sind [rfc]. Die UUIDs werden mit dem NPM-Paket 'uuid' generiert [npmc]. Das UUID-Paket steht unter einer MIT-Lizenz [mit].
webpack	Für die Erstellung des Bundles mit der Such-Logik, das an den Client ausgeliefert wird, wurde webpack verwendet [webb]. Der Grund dafür ist, dass webpack der am meisten genutzte Web-Bundler ist [npmb] und von bekannten Frameworks, wie Angular [ang] oder Vue.js [vue] eingesetzt wird. webpack steht unter einer MIT-Lizenz [mit].

Tabelle 5.2: Technische Entscheide

5.2.5 Design Patterns

Kapselung des Such-Produktes (Fassade)

Die eingesetzte Such-Library (Fuse.js) wurde hinter einem Interface gekapselt. Dies entspricht der Implementation des Design Patterns Fassade [ref]. Der primäre Grund für den Entscheid die Such-Library hinter eine Fassade zu legen ist, dass somit die Einfachheit der Austauschbarkeit der Such-Library erhöht wird. Ein zusätzlicher Grund ist, dass mit der Fassade ein vereinfachtes Interface zum verwendeten Such-Produktes angeboten werden kann und nicht verwendete Teile des Produktes abstrahiert werden konnten.

5.3 Evaluation Such-Produkte

Die Grundlage für die Evaluation der Such-Produkte bildet die in Abschnitt 3.1.5 in Kapitel 3 durchgeführte Recherche. Die Evaluation existierender Such-Produkte hat zum Ziel geeignete Lösungen zu erkennen und die bestmögliche zu evaluieren. Das Resultat der Evaluation ist die Antwort auf die Frage, ob es bereits eine für das Projekt passende Lösung bzw. Teillösung in Bezug auf die Implementation einer Such-Funktionalität gibt oder nicht.

In den folgenden Unterkapiteln werden die folgenden Fragen beantwortet:

- Welche Such-Produkte aus der Recherche wurden als relevant resp. irrelevant beurteilt?
- Welche Kriterien wurden in der Nutzerwertanalyse für die Bewertung der Such-Produkte verwendet?
- Welches Such-Produkt ist für das Projekt am besten geeignet?

5.3.1 Produktidentifikation

Die Produktidentifikation hat zum Ziel für das Projekt relevante Such-Produkte zu identifizieren. Zu diesem Zweck wurde ein simples Ausschlussverfahren angewendet. Ein Produkt, das mindestens ein Ausschlusskriterium erfüllt, wird in der Evaluation nicht weiter berücksichtigt.

Folgende Kriterien führen zu einem Ausschluss eines Produktes aus der Evaluation:

- Kosten: Die Lösung ist kostenpflichtig und es gibt keine Gratisversion.
- Umfang: Die Lösung ist zu umfangreich und kann im Rahmen des Projektes nicht realisiert werden.
- Overkill: Die Lösung erfordert ein grosses Know-how und bringt einen hohen Wartungsaufwand mit sich.
- Ressourcen: Die Lösung benötigt zu viele Ressourcen.

Irrelevante Produkte Folglich wird aufgezeigt welche Produkte als nicht relevant für dieses Projekt beurteilt wurden und warum. Die Begründung wird in Stichworten geführt. Dabei ist zu beachten, dass die Stichwörter sich auf die Ausschlusskriterien beziehen.

Produkt	Begründung in Stichworten
Doofinder [doo]	<ul style="list-style-type: none">• Kosten• Für E-Commerce optimiert

<p>Google Programmable Search Engine [gps]</p>	<ul style="list-style-type: none"> • Beschränkte Anpassungsmöglichkeiten (Design und Funktionalität) • Kosten: Basisversion kostenlos jedoch gedeckelt mit bestimmter Anzahl Suchanfragen • Einschränkungen bei der Datenquelle • Datenschutzbedenken • Abhängigkeit von Google (Anpassungen API)
<p>Inbenta [inb]</p>	<ul style="list-style-type: none"> • Kosten: Inbenta ist eine kostenpflichtige Lösung • Komplexität
<p>Elasticsearch [ela]</p>	<ul style="list-style-type: none"> • Overkill/Komplexität • Kosten (Aufwand) • Ressourcen
<p>Solr [sol]</p>	<ul style="list-style-type: none"> • Komplexität • Kosten (Aufwand) • Ressourcen

Tabelle 5.3: Irrelevante Produkte

Relevante Produkte Folglich wird aufgezeigt welche Produkte als relevant eingestuft wurden. Die Begründung ist in Stichworten geführt und zeigt die wichtigsten Eckpunkte aus der Entscheidung auf.

Produkt	Begründung in Stichworten
Fuse.js [fus]	<ul style="list-style-type: none"> • Open-Source • Features • Konfigurierbar • Leichtgewichtig
Lunr.js [lun]	<ul style="list-style-type: none"> • Open-Source • Features • Konfigurierbar • Leichtgewichtig
Algolia [alga]	<ul style="list-style-type: none"> • Hat eine Gratisversion, ist jedoch limitiert • Features vollumfänglich enthalten • Konfigurierbar

Tabelle 5.4: Relevante Produkte

5.3.2 Evaluationskriterien

Folgende Tabelle zeigt die Evaluationskriterien, welche in der Nutzwertanalyse verwendet wurden, und zeigt auf, was der Optimalfall pro Kriterium wäre.

Kriterium	Idealfall
Kosten	Die Lösung ist Open-Source und somit gratis.
Community	<ul style="list-style-type: none"> • Community >20'000 Personen • Mitwirkende >100 • Aktive Nutzung des Produktes • Hinter der Lösung steht ein Unternehmen (Gratisversion von kommerziellen Lösungen)
Umfang der Features	<p>Die Features decken die Anforderungen des Projekts ab. Im Optimalfall sind folgende Features beinhaltet:</p> <ul style="list-style-type: none"> • Limitierung der Anzahl Suchresultate • Parametrisierung der Suche insbesondere der Definition der gewünschten Suchqualität • Bestimmung der minimalen Übereinstimmung eines Suchbegriffs um als Resultat infrage zu kommen • Einlesen strukturierter Daten • Suchfunktion, Such-Algorithmus, Fuzzy-/Volltextsuche • Automatische Vervollständigung
Benutzerfreundlichkeit	Es ist eine gute Dokumentation vorhanden und die Lösung ist leichtgewichtig.
Suchqualität	Die Qualität der Suche ist hoch. Zudem muss die Möglichkeit vorhanden sein die Suche zu parametrisieren (siehe auch 'Umfang der Features').

Integration (Aufwand)	<ul style="list-style-type: none"> • Leichtgewichtigkeit: Die Mitgelieferte Lösung soll so klein wie möglich sein (Paketgrösse). • Ressourcen: Es ist kein Server notwendig und die Lösung ist ausschliesslich in JavaScript implementiert. • Aufwand: Der Integrationsaufwand ist minimal.
Performance	Die Suche, einschliesslich der Verarbeitung des Inputs, muss in Echtzeit erfolgen und darf einen Schwellenwert von 120ms nicht überschreiten. Die Suche soll zudem primär für kleine Datenmengen optimiert sein.

Tabelle 5.5: Evaluationskriterien

5.3.3 Nutzwertanalyse

Abbildung 5.8 zeigt die Resultate der durchgeführten Nutzwertanalyse auf. Zudem wird in Tabelle 5.6 erläutert wie die Werte zustande gekommen sind.

Kriterien	Gewichtung	Lunr.js		Fuse.js		Algolia	
		Bewertung	Nutzwert	Bewertung	Nutzwert	Bewertung	Nutzwert
Kosten	10	10	100	10	100	3	30
Community	10	2	20	10	100	7	70
Features	20	8	160	8	160	10	200
Benutzerfreundlichkeit	10	5	50	8	80	9	90
Suchqualität	20	5	100	9	180	10	200
Integration	10	8	80	8	80	7	70
Performance	20	9	180	10	200	8	160
Total	100		690		900		820
Rang			3		1		2

Abbildung 5.8: Nutzwertanalyse Produkte

5.3.4 Entscheid

Aufgrund der durchgeführten Analyse wurde für die Suchlogik im Search-Integration-Framework Fuse.js [fus] eingesetzt.

Kriterium	Produkt	Begründung
Kosten	Lunr.js	Open-Source
	Fuse.js	Open-Source
	Algolia	Algolia for Open Source (AOS), sonst kostenpflichtig
Community	Lunr.js	Gebraucht von 55 Personen, 44 Mitwirkende
	Fuse.js	Gebraucht von 156'000 Personen, 81 Mitwirkende
	Algolia	Firma mit über 17.000 Kunden
Umfang der Features	Lunr.js	Alle Features bis auf die Autovervollständigung vorhanden.
	Fuse.js	Alle Features bis auf die Autovervollständigung vorhanden.
	Algolia	Alle Features vorhanden
Benutzerfreundlichkeit	Lunr.js	Dokumentation ist gut jedoch nicht intuitiv verstanden worden (roter Faden fehlt), leichtgewichtig
	Fuse.js	Gute Dokumentation, leichtgewichtig
	Algolia	Gute Dokumentation, professionelles Auftreten, sehr umfangreich
Suchqualität	Lunr.js	Die Suchqualität wurde innerhalb eines Prototyps getestet. Es wurde festgestellt, dass die Volltextsuche mit Wildcards nicht den gleichen Mehrwert wie die Fuzzy-Search bietet.
	Fuse.js	Die Suchqualität wurde anhand eines Prototyps getestet und als sehr gut eingestuft.
	Algolia	Die Suchqualität ist sehr gut, konnte bei diversen Websites verifiziert (Kunden Algolia) und anhand der angebotenen Demos nachvollzogen werden.

Integration	Lunr.js	Integration ist sehr einfach, dank der Dokumentation und dem kleinen Umfang der Lösung. Paketgrösse 28.6 kB (gem. [bunc])
	Fuse.js	Integration ist sehr einfach, dank der guten Dokumentation und dem kleinen Umfang der Lösung. Paketgrösse 15.4 kB (gem. [bunb])
	Algolia	Integration wurde anhand von Demos nachvollzogen und diskutiert. Anbindung über eine API-Schnittstelle mit Zugang zu einem Kundenkonto. Paketgrösse: 32 kB (gem. [buna])
Performance	Lunr.js	Echtzeit Suche - Suchzeit von 40ms - 100ms
	Fuse.js	Echtzeit Suche - Suchzeit von 25ms - 100ms
	Algolia	Echtzeit Suche - Search Queries von 1ms - 20ms (Angabe vom Hersteller [algb] , Feature-abhängig). Zusätzlich muss die Round-Trip-Time zwischen Server und Client berücksichtigt werden.

Tabelle 5.6: Erläuterung Nutzwertanalyse

5.4 Wireframes

Während der Evaluationsphase (siehe Kapitel 7.1) wurden verschiedene Wireframes bezüglich der Darstellung der Suchresultate ausgearbeitet. Diese dienen als Grundlage für die Präsentation der Lösungsvarianten vor dem Auftraggeber. In diesem Kapitel sind die verschiedenen Lösungsvarianten aufgezeigt.

5.4.1 Sucheingabe mit Autocomplete-Vorschlägen

Die einfachste Variante der ausgearbeiteten Wireframes ist ein Suchfeld, in dem die Suchresultate als Dropdown angezeigt werden. Dies ist in Grafik 5.9 dargestellt. Grafik 5.10 zeigt wie das Dropdown aussieht, falls für eine Suche keine Resultate gefunden werden können. Die Suche kann dabei nach der Eingabe von mindestens drei Zeichen ausgelöst werden, sodass während der Suche bereits Suchresultate angezeigt werden. Eine Bestätigung der Suche über den Search-Button ist somit nicht zwingend notwendig.

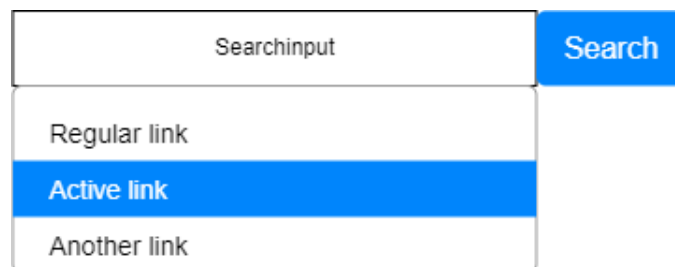


Abbildung 5.9: Suchresultate als Dropdown

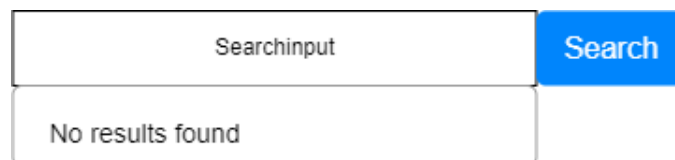


Abbildung 5.10: Suchresultate als Dropdown - keine Suchresultate

5.4.2 Kategorisierte Suchresultate

Grafik 5.11 zeigt eine kategorisierte Variante der Suchresultate nach den drei Kategorien 'Foundation', 'Components' und 'Developers' gemäss der Kategorisierung der Webseiten in der Navigation der SDX-Website. Dabei ist es möglich die einzelnen Kategorien auf- und zuzuklappen.

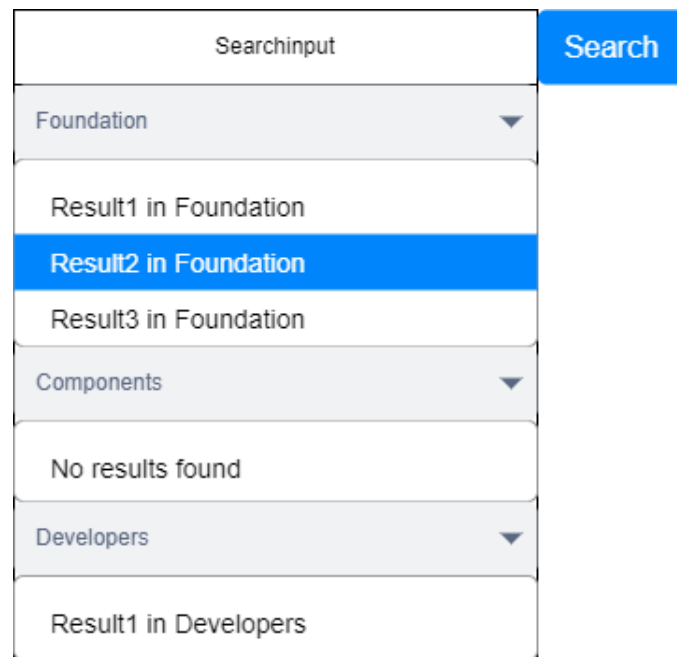


Abbildung 5.11: Kategorisierte Suchresultate

5.4.3 Suchfilter

Eine weitere Variante für das Darstellen der Suchresultate ist das Hinzufügen von Suchfiltern, mit welchen ein User die Suche einschränken kann. Dies ist in Grafik 5.12 veranschaulicht.

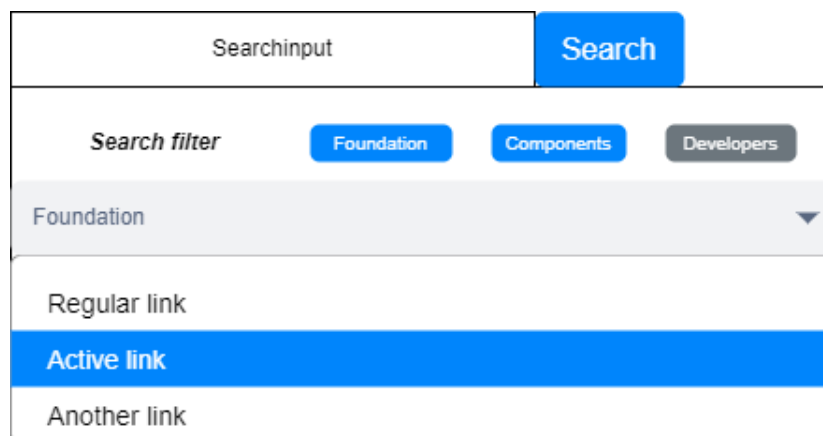


Abbildung 5.12: Suchfilter

5.4.4 Suchhistorie

Eine Erweiterung für die Darstellung der Suche ist das Verwalten und Anzeigen einer Suchhistorie. Dies ist in Grafik 5.13 veranschaulicht. Die Idee ist, dass die Suchhistorie angezeigt wird, wenn in das leere Input-Feld geklickt wird.

Hinweis: Da die Suche clientseitig ausgeführt wird, ist angedacht die Suchhistorie im Local Storage des

Browsers zu verwalten.

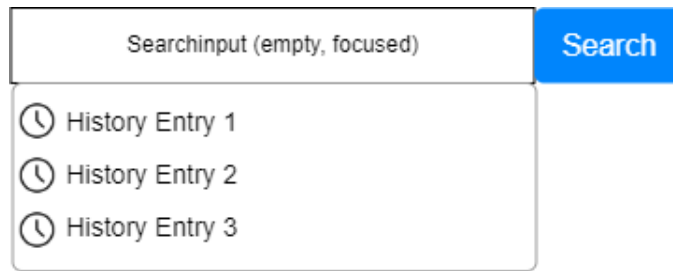


Abbildung 5.13: Suchhistorie

5.4.5 Suchvorschläge

Eine zusätzliche Optimierung der Suche könnte dadurch erzielt werden, dass im Fall, wenn keine Resultate für eine Suchanfrage gefunden werden konnten, alternative Suchvorschläge basierend auf einem Thesaurus oder einer AI angezeigt werden. Wie dies aussehen könnte, ist in Abbildung 5.14 dargestellt.

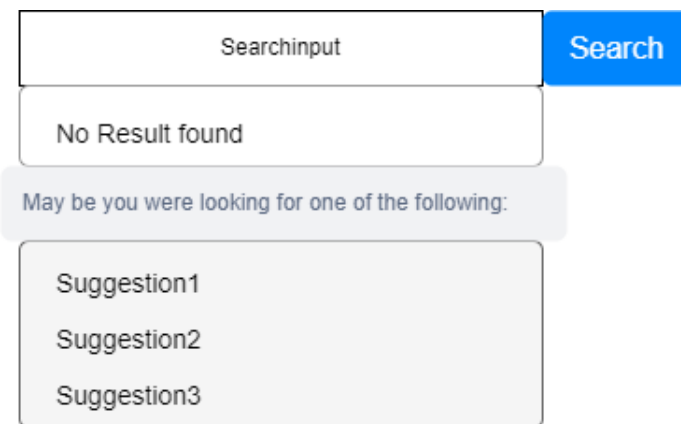


Abbildung 5.14: Suchvorschläge

5.4.6 Hover-Informationen

Eine Idee, um den Benutzern mehr Informationen zu den Suchresultaten direkt in der Resultat-Liste anzeigen zu können, ist, dass ein Dialog eingeblendet wird, wenn mit der Maus auf einem Suchresultat geruht wird (Hovering). Für Komponenten können im Hover-Dialog auch Bilder inkludiert werden. Wie das aussehen könnte, ist in Abbildung 5.15 ersichtlich.

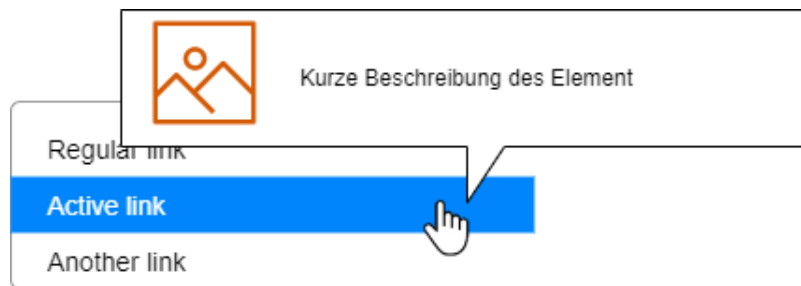


Abbildung 5.15: Hover-Informationen

5.4.7 Mega-Dropdown

Eine andere Idee für die Darstellung von zusätzlichen Informationen zu Suchresultaten ist ein Mega-Dropdown. In einem Mega-Dropdown können die Suchresultate kategorisiert dargestellt werden und es können mehr Informationen als nur der Titel der Webseite angezeigt werden. Wie ein solches Mega-Dropdown aussehen könnte, ist in Abbildung 5.16 veranschaulicht.

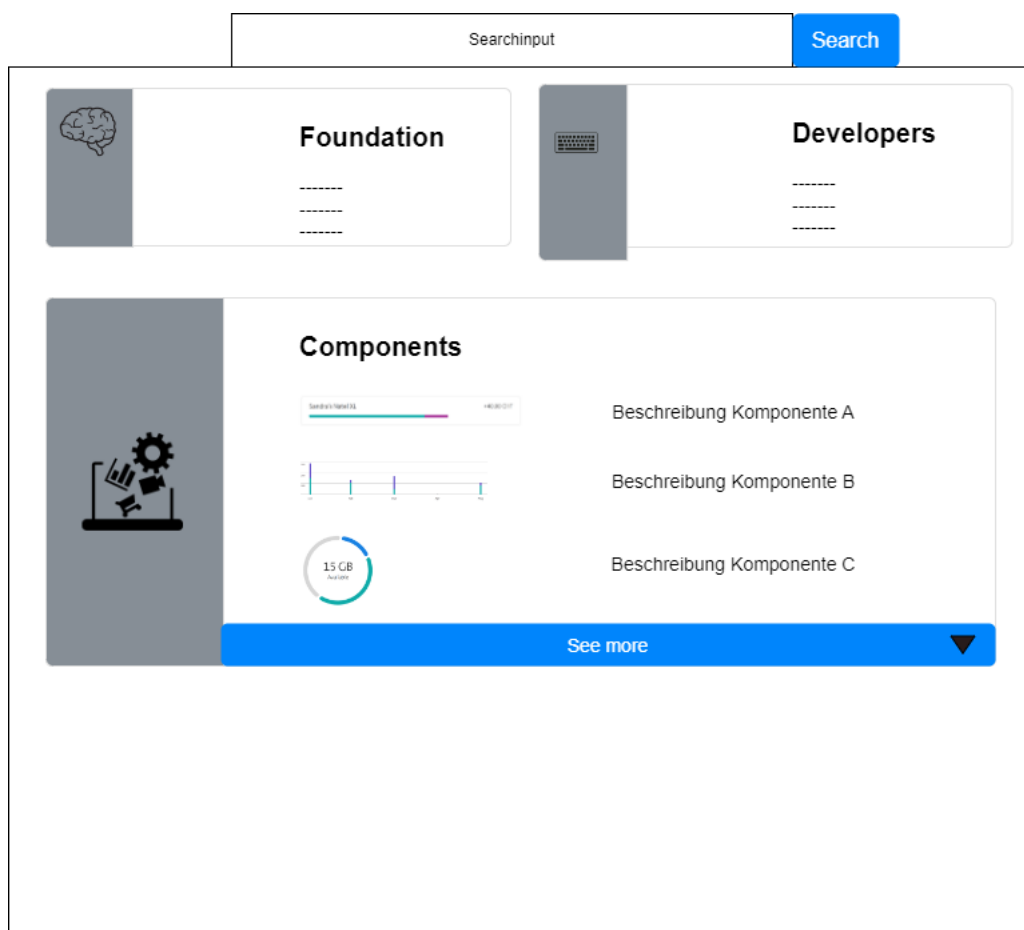


Abbildung 5.16: Mega-Dropdown

5.4.8 Komponenten-Übersicht

Der Hauptbestandteil der SDX-Website sind verschiedene UI-Komponenten. Für die Bezeichnung von solchen UI-Komponenten ist verschiedenes Vokabular in Gebrauch und Personen, die mit dem verwendeten Vokabular nicht vertraut sind, können sich schwer damit tun, die korrekte Komponente zu identifizieren. Um dieses Problem zu adressieren ist die Idee aufgekommen eine visuelle Übersicht aller Komponenten anzubieten. Mit einer solchen Komponenten-Übersicht können Benutzer die gesuchte Komponente visuell identifizieren und müssen nicht wissen, wie eine Komponente bezeichnet wird.

Abbildung 5.17 veranschaulicht, wie eine solche Komponenten-Übersicht aussehen könnte. Die Idee ist es einen zusätzlichen Button anzubieten, mit welchem die Komponenten-Übersicht ein- und ausgeblendet werden kann.

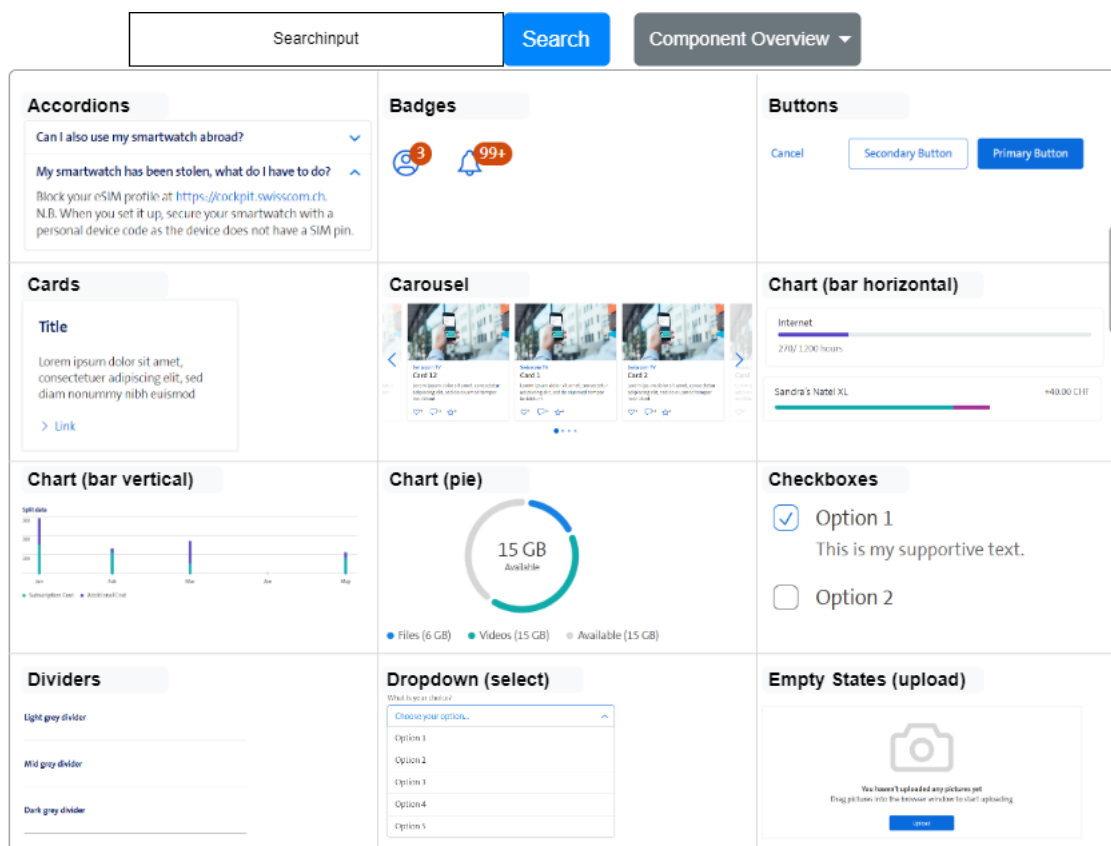


Abbildung 5.17: Komponenten-Übersicht

Eine alternative Darstellung der Komponenten-Übersicht ist eine Ansicht, die alle Komponenten auf einen Blick zeigt, ohne dass dabei gescrollt werden muss. Diese Komponenten-Übersicht enthält pro Komponente einen Titel sowie ein Bild oder Icon welches die Komponente darstellt.

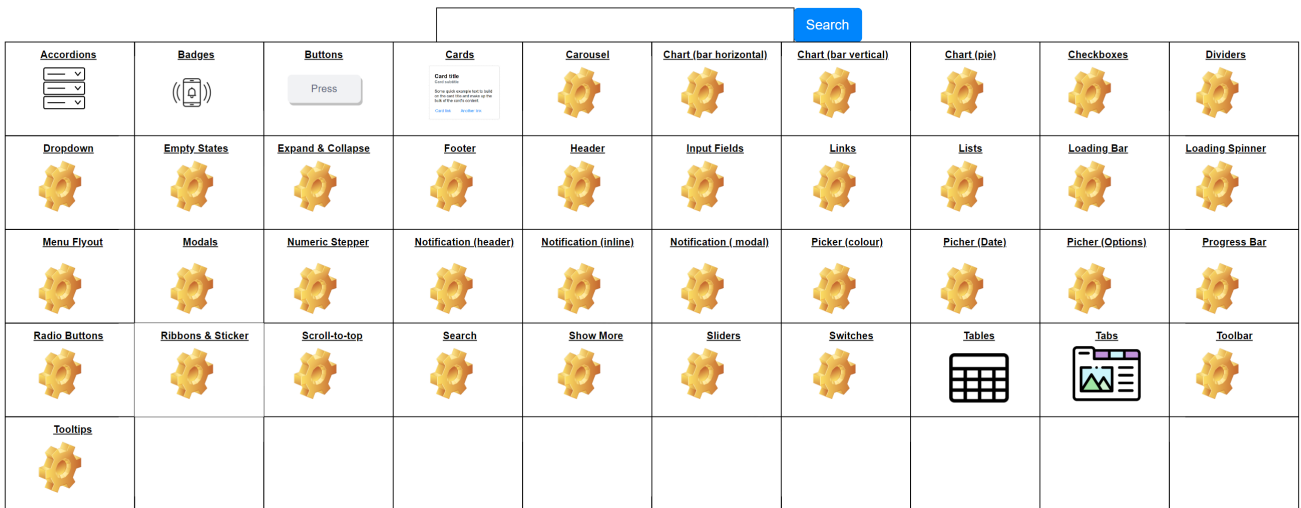


Abbildung 5.18: Komplette Komponenten-Übersicht bei leerem Suchfeld

Die Eingabe von Zeichen könnte auf diesen Lösungsansatz zwei mögliche Auswirkungen haben. Einerseits hätte man die Möglichkeit nicht benötigte Komponenten auszugrauen, wie in der Abbildung 5.19 ersichtlich ist. Andererseits könnte man irrelevante Komponenten ausblenden, wie in Abbildung 5.20 dargestellt.

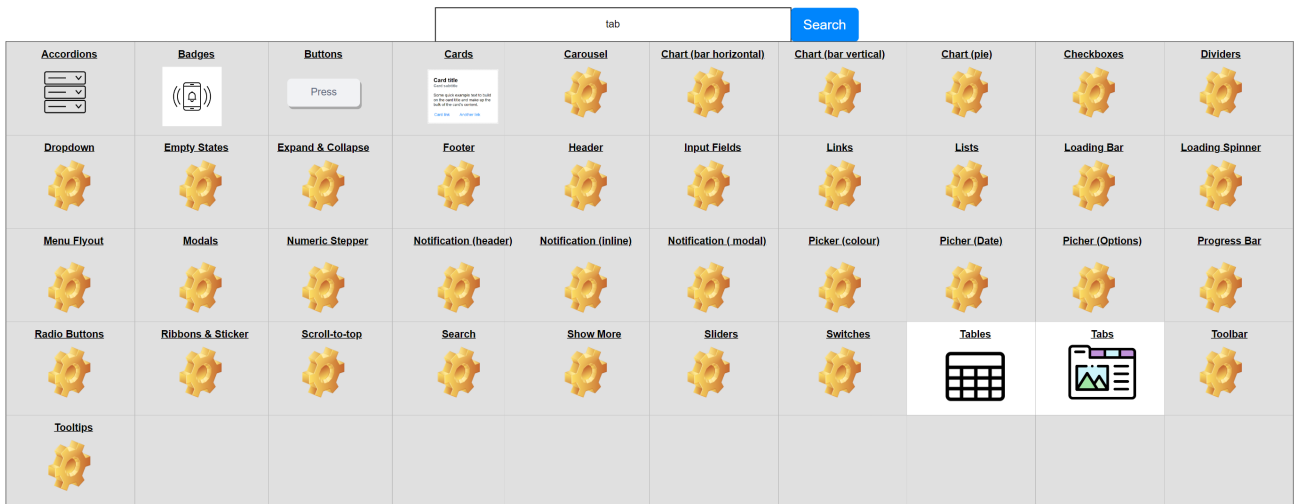


Abbildung 5.19: Komponenten-Übersicht aller Komponenten - Ausgrauen

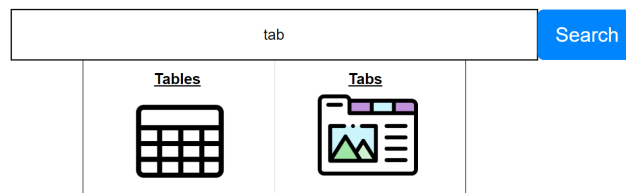


Abbildung 5.20: Komponenten-Übersicht aller Komponenten - Ausblenden

Kapitel 6

Qualitätsmassnahmen

6.1 Coding Guidelines

Coding Guidelines sind von entscheidender Bedeutung für die Code-Qualität und die Einhaltung von Standards in Software-Projekten. Sie minimieren Code-Verstösse, verbessern die Lesbarkeit und Wartbarkeit und erleichtern die Zusammenarbeit im Team. Im folgenden Kapitel werden die im Projekt eingesetzten Tools zur Einhaltung der gesetzten Guidelines sowie die Guidelines selbst vorgestellt.

6.1.1 Clean Code

Für die Erstellung von Code halten sich die Developer an die gängigen Richtlinien von Clean Code [\[Mar09\]](#). Die Zielsetzung ist eine Code-Basis, welche sich als Dokumentation präsentiert.

Folglich eine Auflistung der wichtigsten zu beachtenden Kriterien von Clean Code:

- Die Strukturierung des Codes soll einheitlich sein.
- Funktionen und Klassen sollen so gross wie nötig und klein wie möglich sein.
- Test-, Variablen- und Funktionsnamen müssen aussagekräftig sein.
- Kommentare sollen nur in Ausnahmefällen vermerkt werden.

6.1.2 Tools

In diesem Abschnitt sind Tools aufgeführt, welche für die Erhöhung der Code-Qualität im Projekt eingesetzt werden.

Lint

Ein Linter ist ein Tool, das den Code auf mögliche Fehler, Stil-Verstösse und Sicherheitsprobleme überprüft. Im Falle von TypeScript kann ein Linter dazu beitragen, potenzielle Typ- und Syntaxfehler zu erkennen und zu beheben, bevor der Code ausgeführt wird. Dies kann Zeit und Mühe sparen, indem Entwicklern

geholfen wird, Fehler früher im Entwicklungsprozess zu erkennen und zu beheben. Ein Linter trägt dazu bei, die Code-Qualität zu verbessern und die Einhaltung von Best Practices zu gewährleisten. Im Projekt wurde der ESLint eingesetzt [esl].

Hinweis: Die im Projekt genutzte Linter-Konfiguration wurde vom Auftraggeber vorgegeben.

Prettier

Prettier ist ein Codeformatierungs-Tool, das für viele Programmiersprachen, einschliesslich TypeScript, zur Verfügung steht [pre]. Es hilft dabei, den Code konsistent und lesbar zu halten, indem es automatisch Formatierungen wie Einrückungen, Zeilenumbrüche und Leerzeichen anpasst. Dies ist insbesondere dann wertvoll, wenn die Entwickler in unterschiedlichen IDEs mit verschiedenen Standardeinstellungen bezüglich Einrückungen oder Leerzeichen arbeiten. Der Prettier spart somit viel Zeit und Mühe, da Entwickler nicht manuell den Code formatieren müssen. Ausserdem hilft der Prettier bei der Einhaltung von Coding-Standards und verhindert Formatierungsfehler, die zu Bugs führen können.

Hinweis: Die im Projekt genutzte Prettier-Konfiguration wurde vom Auftraggeber vorgegeben.

TypeScript-Konfiguration

Eine TypeScript-Konfiguration ist notwendig, um die Compiler-Optionen und andere Einstellungen für das Projekt festzulegen und sicherzustellen, dass der Code korrekt kompiliert und ausgeführt werden kann.

Hinweis: Die im Projekt genutzte TypeScript-Konfiguration wurde vom Auftraggeber vorgegeben.

6.1.3 Error Handling

Exceptions werden bei den auf das Dateisystem schreibenden und lesenden Prozessen, sowie aus den einzelnen Komponenten in Phase 1 des Konzepts (siehe Kapitel 5.1) geworfen. Die geworfenen Fehler werden grundsätzlich nicht behandelt, da der User sehen soll, wenn etwas fehlgeschlagen ist. Die Fehlermeldungen sind aussagekräftig geschrieben und enthalten die notwendigen Informationen zur Nachverfolgung des Ursprungs eines Fehlers.

6.1.4 Logging der Integrationsschritte

Standardmässig werden dem User durch eine Logging-Komponente die einzelnen Schritte des Search-Integrators (siehe Kapitel 5.1) bestätigt. Das Logging hat zum Ziel dem User den Stand einer laufenden Integration darzulegen, um im Fehlerfall die Fehlerquelle identifizieren zu können. Das Logging-Level des Search-Integrators kann über die dem Search-Integrator übergebene Konfiguration gesteuert werden. Die möglichen Log-Level sind 'All', 'Debug', 'Info' (Standard) und 'Off'.

6.2 Versionskontrolle

6.2.1 Struktur Repository

Zur Verwaltung der Produkt- und Projekt-Dokumentation wurde GitLab [gita] in der Infrastruktur der OST (siehe <https://gitlab.ost.ch/rolf.oberhaensli/ba-dokumentation> und https://gitlab.ost.ch/rolf.oberhaensli/sdx_search_code) verwendet. Das Search-Integrator-Framework wird über GitHub [gitb] (siehe <https://github.com/ManuWeber93/search-integrator>) verwaltet. Die Dokumentation der Bachelorarbeit, das Auftraggeber-spezifische Anwendungsbeispiel und der Source-Code des Frameworks wurden je in einem separaten Repository gepflegt.

6.2.2 Pre-Commit Hook

Für die beiden Source-Code-Repositories auf GitLab und GitHub wurde ein Pre-Commit-Hook eingerichtet. Dieser wird bei jedem Commit getriggert und führt den Prettier und den Linter für alle gestagten Dateien aus. Der Prettier formatiert dabei die gestagten Files gemäss der Konfiguration und der Linter korrigiert alle konfigurierten Code-Style-Fehler, für die ESLint weiss, wie sie zu korrigieren sind. Des Weiteren bricht der Linter den Commit ab, sollten Code-Style-Fehler vorhanden sein, die er nicht korrigieren kann.

6.2.3 CI/CD

CI/CD-Pipeline der Dokumentation

Die CI/CD-Pipeline der Dokumentation auf GitLab besteht aus zwei Stages: build und diff. In der Build-Stage wird das PDF der Projekt- und Produkt-Dokumentation erstellt. Dabei wird auch das Glossar erstellt und in das PDF integriert. In der Diff-Stage wird ein PDF-Dokument erstellt, das Unterschiede zwischen der aktuellen Version und der Version, die mit dem Tag 'latexdiff-base' in Git markiert wurde, aufzeigt. Der Use Case dieser Stage ist das Review einer überarbeiteten Dokumentation. Wenn die Unterschiede zwischen zwei Versionen hervorgehoben sind, kann sich der Reviewer auf geänderte Abschnitte konzentrieren.

CI/CD-Pipelines des Source-Codes

Pipeline Auftraggeber Die Pipeline für die SDX-Website beim Auftraggeber wird als Blackbox angeschaut. Darin wird die SDX-Website getestet und zusammengebaut. Die daraus für das Projekt relevanten Artefakte sind die generierten HTML-Files der einzelnen Webseiten. Das Projekt dieser Bachelorarbeit wird innerhalb der Pipeline, nach der eigentlichen Generation der Files, ansetzen. Folgendes Ablaufdiagramm soll das Vorgehen bzw. die Erweiterung der Pipeline aufzeigen. Zur Integration in die Pipeline wurde auf Wunsch des Auftraggebers ein Build-Script erstellt, welches in der Pipeline vom Auftraggeber ausgeführt werden kann. Darin werden die einzelnen Schritte aus dem Konzept (siehe Kapitel 5.1), in der Abbildung 6.1 grün dargestellt, mithilfe des NPM-Pakets 'search-integrator' ausgeführt.

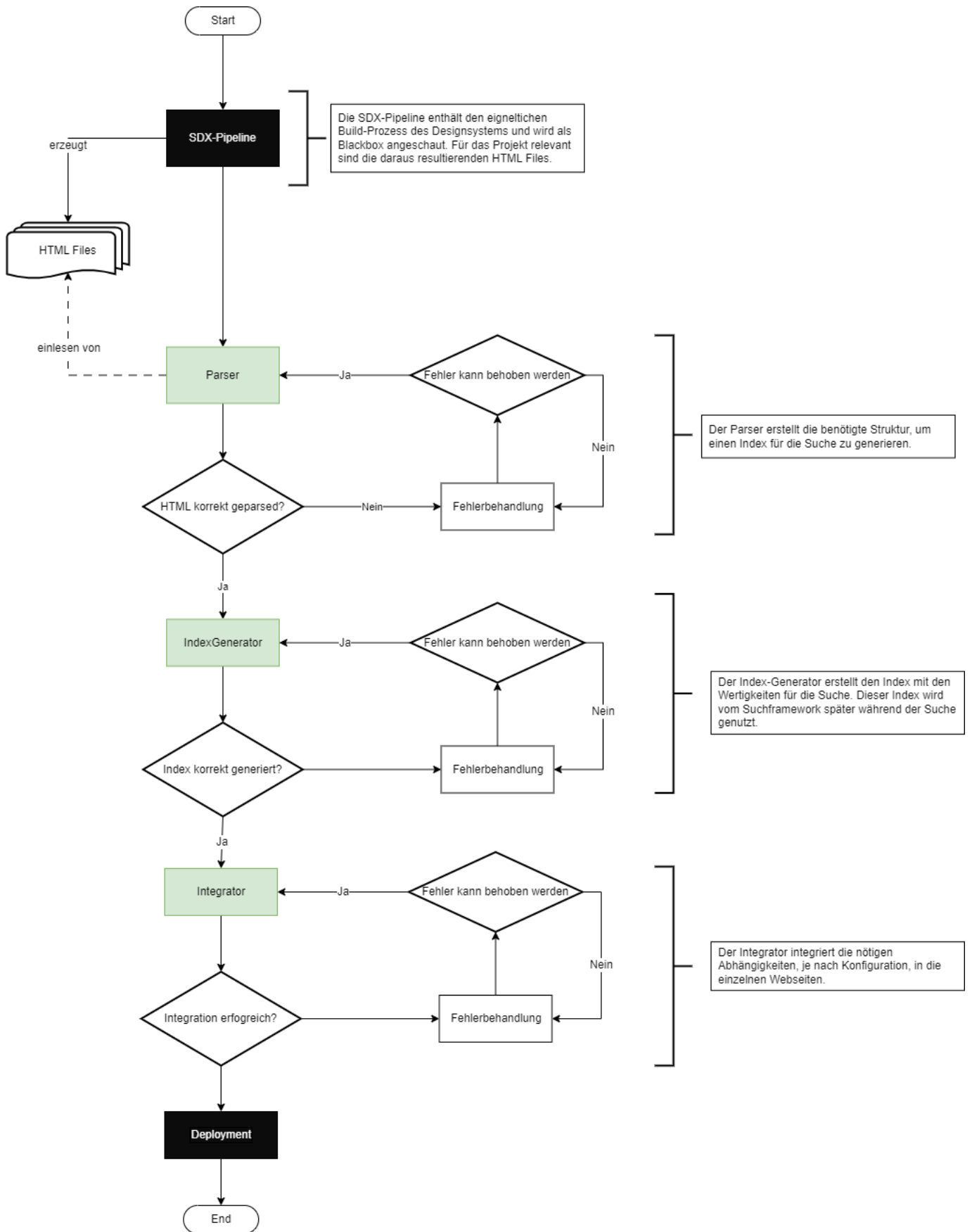


Abbildung 6.1: Pipeline Erweiterung vor Deployment

Hinweis: Zur Simplifizierung des Ablaufs wird das Bundling der Suchlogik in Abbildung 6.1 weggelassen.

Pipeline Source-Code Die Code-Basis im Projekt wurde in zwei Repositorien aufgeteilt. Der Auftraggeber-spezifische Code wurde auf GitLab [gita] geführt. Die öffentliche Code-Basis des Search-Integrator-Frameworks wurde auf GitHub [gitb] ausgelagert. Die Pipeline auf GitLab wurde einfach gehalten - es wird lediglich eine Build-Stage durchlaufen. Der Grund dafür ist, dass der Code in diesem Repository nur die Ausführung des Search-Integrator-Frameworks, sowie die zur Verfügungstellung einer Such-Komponente, beinhaltet. Die Pipeline auf GitHub wird in zwei Stages unterteilt: Build- und Test-Stage. In der Build-Stage wird das TypeScript-Projekt kompiliert und anschliessend werden in der Test-Stage die definierten Tests durchlaufen. Die Test-Stage erzeugt dabei einen Coverage-Report als Artefakt.

6.2.4 Workflow für die Arbeit am Source-Code

Jeder Task, an dem gearbeitet wird, ist als Issue auf GitLab vorhanden. Issues sind auf Branches gemappt, wobei kleine Issues in einem Branch zusammengeführt werden können. Immer wenn ein Issue abgeschlossen ist, wird der relevante Feature-Branch mit dem Main-Branch zusammengeführt. Dazu wird ein Pull Request auf den Main-Branch des Projekts erstellt. Im Zuge dessen wird ein Reviewer hinzugezogen. Der Reviewer muss die Änderungen auf dem Feature-Branch nachvollziehen und verifizieren bzw. falsifizieren und kontrollieren, ob die DoD (siehe Kapitel 6.4) eingehalten wurde. Bevor ein Entwickler ein Review eines Pull Requests anfordert, wird sichergestellt, dass der Feature-Branch die neuesten Änderungen aus dem Main-Branch enthält, um die Wahrscheinlichkeit eines Merge-Konflikts zu reduzieren. Wenn dies der Fall ist, darf ein Review für den Pull-Request angefragt werden. Wenn das Review erfolgreich ist und das DoD eingehalten wurde, bestätigt der Reviewer den Pull-Request und der Feature-Branch wird mit dem Main-Branch des Projekts zusammengeführt. Der Feature-Branch wird gelöscht, sobald der Merge mit dem Main-Branch durchgeführt wurde.

Hinweis: Das Git-Repository wurde so konfiguriert, dass ein direktes Pushen auf den Main-Branch nicht möglich ist. Somit konnte die Arbeit mit Feature-Branche und Merge-Requests forciert werden. Zudem müssen alle Tests erfolgreich durchlaufen, um ein Zusammenführen zweier Branches zu ermöglichen.

Der Workflow für die Arbeit am Source-Code ist in Abbildung 6.2 abgebildet.

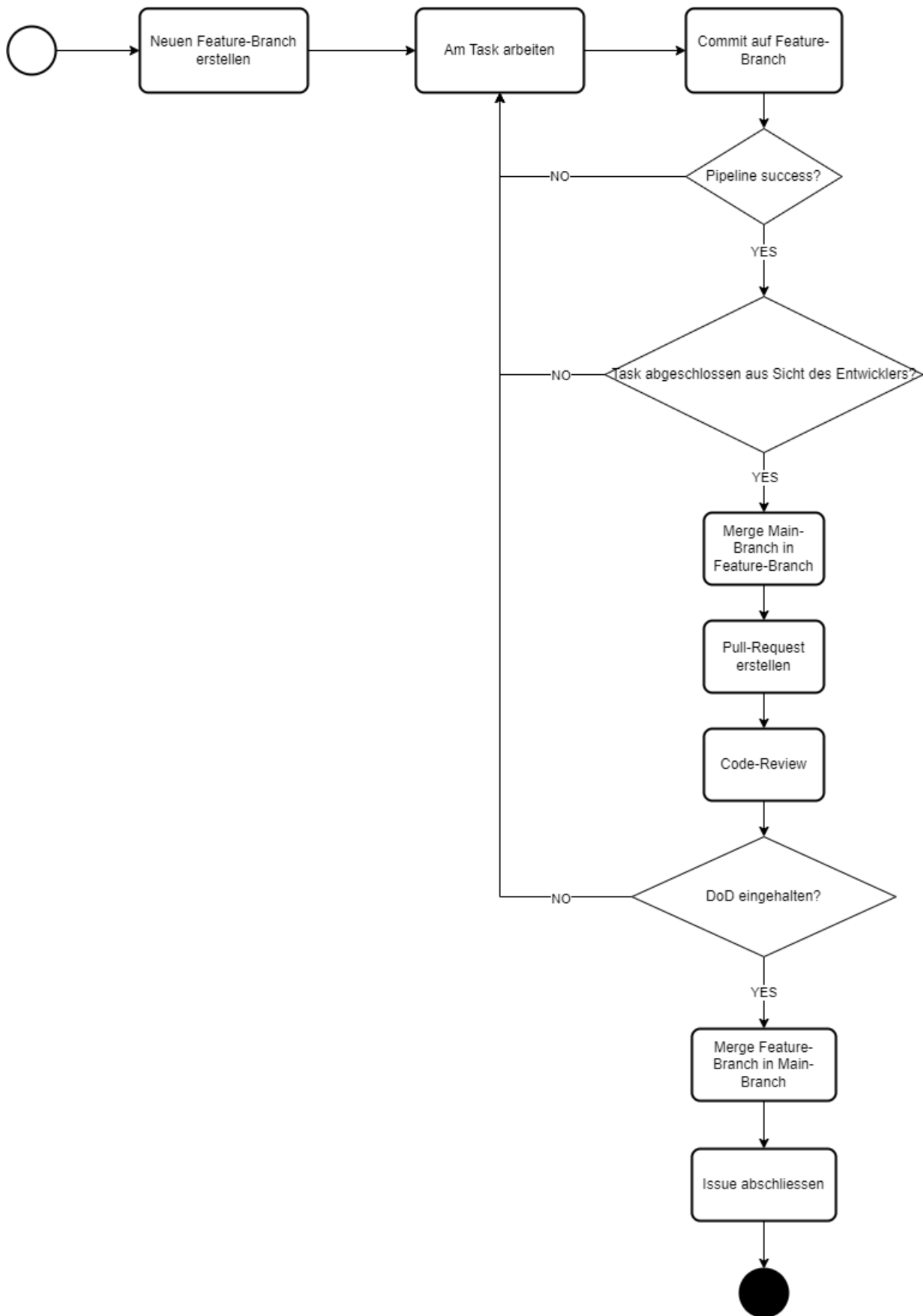


Abbildung 6.2: Workflow für die Arbeit am Source-Code

6.3 Code-Reviews

Für Tasks, in denen Code geschrieben wurde, muss ein Code-Review durchgeführt werden. Erst nach abgeschlossenem Review kann ein Task als abgeschlossen markiert werden (siehe auch Kapitel 6.4). Reviews werden jeweils vom Teampartner durchgeführt. Code-Reviews werden nicht explizit protokolliert, stattdessen wird im Issue festgehalten, dass ein Review stattgefunden hat. Code-Anpassungen, die aus dem Code-Review hervorgehen, sind implizit in den Commits der Versionskontrolle (siehe auch Kapitel 7.9) dokumentiert.

6.4 Definition of Done

Die Definition of Done definiert welche Kriterien erfüllt sein müssen, damit ein Task oder eine User-Story als abgeschlossen gilt. Die Definition of Done muss in jedem Review eines Tasks durchgearbeitet werden. Ein Task muss alle Punkte der Liste erfüllen, damit der Code in den Main-Branch gemergt wird.

- Der Code kompiliert ohne Fehler.
- Falls Warnungen vorhanden sind, müssen diese während dem Review begründet akzeptiert werden. Eine allfällige Begründung wird im Issue dokumentiert.
- Die CI/CD-Pipeline läuft ohne Fehler durch.
- Alle Tests laufen erfolgreich durch.
- Es sind genügend Tests vorhanden, sodass die in Kapitel 6.5 definierte Test-Coverage erreicht wird.
- Die in Kapitel 6.1 definierten Coding Guidelines und Standards sind erfüllt.
- Der Task (inkl. Code) wurde vom Teampartner gereviewed.
- Eine User-Story gilt als Done, wenn alle daraus abgeleiteten Tasks erledigt sind.

6.5 Metriken und Code Analysis

Als Code-Analysis-Metrik wurde die Code-Coverage eingesetzt und analysiert. Zudem wird zur Analyse der Performance und Accessibility der Lighthouse Report eingesetzt [\[lig\]](#).

6.5.1 Code-Coverage

Für die Code-Coverage wird c8 eingesetzt [\[c8\]](#). c8 generiert einen HTML- sowie Text-Report. Darin ist ersichtlich welche Funktionen, Zeilen und Statements durchlaufen werden oder nicht. Für die Unterstützung von TypeScript wird mit Sourcemaps [\[ts-\]](#) gearbeitet. Für die Code-Coverage wird ein Wert von 85% angestrebt.

Abbildung 6.3 zeigt die Auswertung aus dem HTML-Report. Es ist zu erkennen, dass die gesamte Businesslogik auf die Korrektheit der Funktionalität getestet worden ist. Die Code-Coverage liegt leicht

über dem angestrebten Wert von 85%. Der Frontend-Code wurde während des Projekts nicht mehr automatisiert getestet, da dies einen Umbau der Logik gefordert hätte und der Code-Freeze bereits feststand.

All files

86.3% Statements 983/1139 94.4% Branches 152/161 98% Functions 49/50 86.3% Lines 983/1139

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

























File ▲	Statements ⇅	Branches ⇅	Functions ⇅	Lines ⇅
main	 96.89% 187/193	 66.66% 10/15	 100% 7/7	 96.89% 187/193
main/backend/integrator	 99.06% 212/214	 97.05% 33/34	 100% 11/11	 99.06% 212/214
main/backend/parser	 98.99% 394/398	 98.68% 75/76	 100% 15/15	 98.99% 394/398
main/frontend	 0% 0/142	 0% 0/1	 0% 0/1	 0% 0/142
main/search/search-framework	 100% 31/31	 100% 3/3	 100% 3/3	 100% 31/31
main/utlis	 98.75% 159/161	 96.87% 31/32	 100% 13/13	 98.75% 159/161

Abbildung 6.3: Übersicht Code-Coverage

6.6 Testkonzept

Tests sind ein Mittel um sicherzustellen, dass die geforderten Qualitätsziele und Anforderungen eingehalten werden. Das Testkonzept definiert wie die funktionalen und nicht-funktionalen Anforderungen messbar geprüft werden. Dies soll sicherstellen, dass die definierten Anforderungen erfüllt sind, indem diese aus verschiedenen Blickwinkeln getestet werden. Folgend sind die übergreifenden Kriterien, die dabei besonders beachtet werden.

- Funktionale Eignung
- Performance-Effizienz
- Kompatibilität
- Benutzerfreundlichkeit
- Verlässlichkeit
- Sicherheit
- Wartbarkeit

6.6.1 Unit-Tests

Unit-Tests werden eingesetzt, um die funktionale Korrektheit des Codes zu verifizieren. Sie werden manuell während der Entwicklung sowie automatisiert in der Pipeline für Business-Logik durchgeführt und kontrolliert. Als Test-Framework für die Unit-Tests wird Mocha [[moca](#)] zusammen mit Chai als Assertion-Library [[cha](#)] eingesetzt.

6.6.2 Performance-Tests

Performance-Tests werden eingesetzt, um die Performance der Suche zu überprüfen und allfällige Probleme zu identifizieren. Des Weiteren werden Performance-Tests eingesetzt, um verschiedene Lösungsvarianten miteinander zu vergleichen.

6.6.3 End-2-End-Tests

End-2-End-Tests werden eingesetzt, um zu verifizieren, dass der gesamte Prozess vom Extrahieren der für die Suche relevanten Daten aus den HTML-Dateien, über die Erstellung des Indexes bis hin zu der effektiven Suche funktioniert. Dies wird überprüft, indem neue HTML-Seiten zu der Website hinzugefügt werden und schlussendlich verifiziert wird, dass die neue Seite über die Suche gefunden werden kann.

6.6.4 Integrations-Tests

Integrations-Tests werden eingesetzt, um zu überprüfen, dass die Suche reibungslos in das bestehende SDX-System integriert werden kann. Dabei wird verifiziert, dass die SDX-Website auch nach der Integration der Suche weiterhin funktioniert.

Wichtig: Die Integration der Suche in SDX ist nicht im Scope des Projekts, sondern wird im Anschluss an das Projekt durch Swisscom durchgeführt. Somit sind auch die Integrations-Tests nicht im Scope des Projekts. Vollständigkeitshalber werden sie hier trotzdem aufgeführt.

6.6.5 Last-Tests

Last-Tests sind im Kontext des Projekts nicht relevant, da die Suche clientseitig durchgeführt und somit keine Last auf einem Server generiert wird.

6.6.6 Usability-Tests

Usability-Tests werden eingesetzt, um die Intuitivität und Nützlichkeit der Darstellung der Suche sowie der Suchresultate zu verifizieren.

Für die Durchführung der Usability-Tests wird eine Testumgebung aufgesetzt, auf welcher die Testpersonen die definierten Testfälle durchspielen. Die Testumgebung wird auf verschiedenen Geräten und mit verschiedenen Browsern durchgeführt. Dabei soll dokumentiert werden, wie Benutzer die Fälle lösen. Zur Dokumentation und zur vertieften Analyse werden die Tests mithilfe von Ton und Bildaufnahmen unterstützt. Die gesammelten Daten werden im Anschluss ausgewertet, analysiert und Optimierungen abgeleitet.

Nebst den Testfällen werden folgende Punkte beachtet und allgemeine Fragen im Anschluss an die Durchführung an die Testperson gestellt.

1. Wie viele Suchanfragen setzte eine Testperson pro Testfall ab?
2. Wurde die Suche durch die Testperson als Navigation verwendet?
3. Wie gut fanden die Testpersonen die Darstellung der Suchresultate?
4. Wie gut fanden die Testpersonen Suchresultate im Allgemeinen?
5. Hat den Testpersonen etwas gefehlt?
6. Waren die Testpersonen schneller in der Navigation mithilfe der Suche?
7. Wird bemerkt, dass über das Bestätigen der Suche (ENTER oder klick auf Lupe-Icon) mehr Treffer angezeigt werden?

Testfälle

Es ist zu beachten, dass die Testfälle sehr offen und mit wenig Kontext gestellt sind. Dies ist Absicht, da ersichtlich werden soll, wie die Benutzer denken bzw. welche Suchbegriffe sie wählen.

Die Testfälle sind wie folgt gegliedert:

- Allgemeine Testfälle: Testfälle die von allen Testpersonen durchgeführt werden.

- Benutzergruppen spezifische Testfälle: Testfälle die nur von den einzelnen Benutzergruppen durchgeführt werden.

Folgende Testfälle wurden für die Usability-Tests ausgearbeitet. Die Dokumentation der durchgeführten Usability-Tests sowie die daraus abgeleiteten Massnahmen können im Anhang eingesehen werden.

Allgemeine Testfälle

Nr.	Aufgabe	Erwartetes Resultat
A1	Sie möchten sich in das Thema Accessibility einlesen. Suchen Sie nach geeigneten Einträge und navigieren Sie zu diesen.	<ol style="list-style-type: none"> 1. Der Benutzer befindet sich auf der SDX-Website. 2. Der Benutzer nutzt das Suchfeld in SDX als Navigation. 3. Der Benutzer gibt seinen Suchbegriff ein. 4. Der Benutzer kann über die Suchresultate die gesuchte Komponente identifizieren. 5. Wie gut sind die Suchresultate? 6. Ziel: https://sdx.swisscom.ch/foundation_-_accessibility.html
A2	Sie sind auf Ideensuche und möchten eine Übersicht über alle Icons. Navigieren Sie zur gesuchten Webseite.	<ol style="list-style-type: none"> 1. Der Benutzer befindet sich auf der SDX-Website. 2. Der Benutzer nutzt das Suchfeld in SDX als Navigation. 3. Der Benutzer gibt seinen Suchbegriff ein. 4. Der Benutzer kann über die Suchresultate die gesuchte Komponente identifizieren. 5. Wie gut sind die Suchresultate? 6. Ziel: https://sdx.swisscom.ch/foundation_-_icons.html

A3	<p>Sie wollen wissen, ob SDX Informationen über Best Practices mit Bezug zu Projektmanagement enthält. Suchen Sie nach entsprechenden Stichwörtern.</p>	<ol style="list-style-type: none"> 1. Der Benutzer befindet sich auf der SDX-Website. 2. Der Benutzer nutzt das Suchfeld. 3. Der Benutzer gibt seinen Suchbegriff ein. 4. Der Benutzer kann über die Suchresultate keine Inhalte finden. 5. Wurde klar, dass keine Informationen auf SDX vorhanden sind? 6. Ziel: Es wird angezeigt, dass keine Informationen gefunden worden sind.
----	---	---

Tabelle 6.1: Allgemeine Usability-Tests

UX/UI spezifische Testfälle

Nr.	Aufgabe	Erwartetes Resultat
D1	<p>Sie möchten wissen, wie Icons relativ zueinander ausgerichtet werden sollten. Suchen Sie nach einer Empfehlung in SDX.</p>	<ol style="list-style-type: none"> 1. Der Benutzer befindet sich auf der SDX-Website. 2. Der Benutzer nutzt das Suchfeld in SDX als Navigation. 3. Der Benutzer gibt seinen Suchbegriff ein. 4. Der Benutzer kann über die Suchresultate die gesuchte Webseite identifizieren. 5. Ziel: https://sdx.swisscom.ch/foundation_-_icons.html#designers-relative-alignment

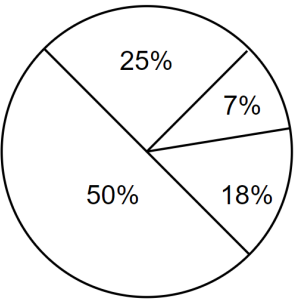
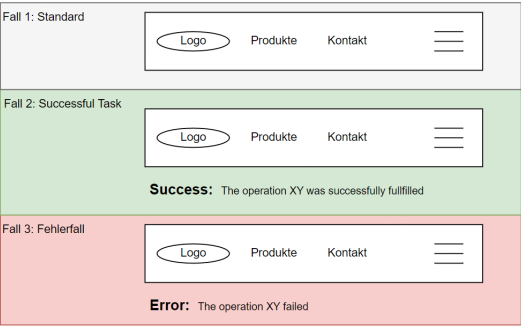

<p>D2</p>	<p>Sie benötigen für die saubere Darstellung ein Diagramm. Die darzustellenden Daten sind Anteile eines Ganzen. Suchen Sie nach geeigneten Komponenten in SDX.</p>  <p>A pie chart with four segments. The largest segment is 50%, followed by 25%, 18%, and 7%.</p>	<ol style="list-style-type: none"> 1. Der Benutzer befindet sich auf der SDX-Website. 2. Der Benutzer erkennt und nutzt das Suchfeld in SDX. 3. Der Benutzer gibt seinen Suchbegriff ein. 4. Der Benutzer kann über die Suchresultate die gesuchte Komponente identifizieren. 5. Ziel: https://sdx.swisscom.ch/components_-_chart_(pie).html
<p>D3</p>	<p>Sie arbeiten in einem Projekt und sind sich nicht mehr sicher, welche Optionen für einen Header gegeben sind. Im Projekt wurde die Anforderung gestellt, dass der Header die Navigation, das Logo sowie ein Dropdown enthält. Zudem sollen Meldungen ebenfalls im Headerbereich dargestellt werden, falls möglich. Da Sie mit SDX arbeiten, möchten sie wissen, welche Optionen bezüglich Header geboten werden.</p>  <p>Three examples of header designs are shown. Each example includes a logo, the text 'Produkte', 'Kontakt', and a hamburger menu icon. The first is 'Fall 1: Standard'. The second is 'Fall 2: Successful Task' and includes a green background and a 'Success: The operation XY was successfully fulfilled' message. The third is 'Fall 3: Fehlerfall' and includes a red background and an 'Error: The operation XY failed' message.</p>	<ol style="list-style-type: none"> 1. Der Benutzer befindet sich auf der SDX-Website. 2. Der Benutzer nutzt das Suchfeld in SDX als Navigation. 3. Der Benutzer gibt seinen Suchbegriff ein. 4. Der Benutzer kann über die Suchresultate die gesuchte Komponente identifizieren. 5. Wie gut sind die Suchresultate? 6. Ziel 1: https://sdx.swisscom.ch/components_-_header.html 7. Ziel 2: https://sdx.swisscom.ch/components_-_notifications_(header).html

Tabelle 6.2: UX/UI spezifische Usability-Tests

Developer spezifische Testfälle

Nr.	Aufgabe	Erwartetes Resultat
E1	<p>Der Arbeitskollege hat Ihnen gesagt, dass die Abstände zwischen zwei Komponenten auf der Webseite ihres Projektes nicht stimmen. Da Sie SDX in Ihrem Projekt verwenden, möchten Sie wissen, welche Abstände es gibt respektive welche Optionen Ihnen zur Verfügung stehen. Suchen Sie nach geeigneten Informationen.</p> 	<ol style="list-style-type: none"> 1. Der Benutzer befindet sich auf der SDX-Website. 2. Der Benutzer nutzt das Suchfeld in SDX als Navigation. 3. Der Benutzer gibt seinen Suchbegriff ein. 4. Der Benutzer kann über die Suchresultate die gesuchte Komponente identifizieren. 5. Ziel: https://sdx.swisscom.ch/developers_-_margin_padding_utils.html
E2	<p>Sie möchten wissen, wie ein Icon aus SDX genutzt werden kann. Suchen Sie nach geeigneten Informationen.</p>	<ol style="list-style-type: none"> 1. Der Benutzer befindet sich auf der SDX-Website. 2. Der Benutzer nutzt das Suchfeld in SDX als Navigation. 3. Der Benutzer gibt seinen Suchbegriff ein (Icon, Icons, etc.). 4. Der Benutzer kann über die Suchresultate die gesuchte Webseite identifizieren. 5. Ziel: https://sdx.swisscom.ch/foundation_-_icons.html#developers

E3	<p>Sie arbeiten an einem Projekt für einen Kunden und benötigen eine Möglichkeit, Auswahlfelder mit verschiedenen Optionen anzubieten. Suchen Sie nach geeigneten Möglichkeiten.</p>	<ol style="list-style-type: none"> 1. Der Benutzer befindet sich auf der SDX-Website. 2. Der Benutzer erkennt und nutzt das Suchfeld in SDX. 3. Der Benutzer gibt seinen Suchbegriff ein (Optionen, Auswahl, Box). 4. Der Benutzer kann über die Suchresultate die gesuchte Komponente identifizieren. 5. Ziel: https://sdx.swisscom.ch/components_-_checkboxes.html
E4	<p>Sie arbeiten in einem neuen Projekt und es wurde festgelegt, dass SDX für die Darstellung genutzt werden soll. Im Projekt verwenden Sie Angular. Nun möchten Sie SDX in Ihr Projekt integrieren und suchen nach den nötigen Informationen.</p>	<ol style="list-style-type: none"> 1. Der Benutzer befindet sich auf der SDX-Website. 2. Der Benutzer erkennt und nutzt das Suchfeld in SDX. 3. Der Benutzer gibt seinen Suchbegriff ein. 4. Der Benutzer kann über die Suchresultate die gesuchte Komponente identifizieren. 5. Ziel: https://sdx.swisscom.ch/developers_-_getting_started.html#angular

Tabelle 6.3: Entwickler spezifische Usability-Tests

SCRUM Master und Product Manager spezifische Testfälle

Nr.	Aufgabe	Erwartetes Resultat
M1	<p>Sie bereiten sich auf ein Meeting mit dem Design-Team vor und möchten wissen, welche Farben im Design System zur Verfügung stehen, um eine Farbauswahl zu diskutieren. Suchen Sie nach geeigneten Informationen.</p>	<ol style="list-style-type: none"> 1. Der Benutzer befindet sich auf der SDX-Website. 2. Der Benutzer erkennt und nutzt das Suchfeld in SDX. 3. Der Benutzer gibt seinen Suchbegriff ein. 4. Der Benutzer kann über die Suchresultate direkt auf die gesuchte Information zugreifen. 5. Ziel: https://sdx.swisscom.ch/foundation_-_colours.html
M2	<p>Sie arbeiten in einem Projekt und möchten den Aufwand für die Integration einer Notifications (modal) abschätzen. Navigieren Sie zu Notifications (modal).</p>	<ol style="list-style-type: none"> 1. Der Benutzer befindet sich auf der SDX-Website. 2. Der Benutzer nutzt das Suchfeld in SDX als Navigation. 3. Der Benutzer gibt seinen Suchbegriff ein. 4. Der Benutzer kann über die Suchresultate die gesuchte Komponente identifizieren. 5. Ziel: https://sdx.swisscom.ch/components_-_notifications_(modal).html#designers

Tabelle 6.4: Management spezifische Usability-Tests

6.6.7 System-Tests

System-Tests werden vor den Usability-Tests, dem Alpha-Release sowie vor dem Beta-Release manuell durchgeführt. Es soll verifiziert werden, dass das gesamte System zu diesem Zeitpunkt korrekt funktioniert. Folgend sind die definierten System-Tests dokumentiert.

Die Tests werden auf verschiedenen Mobile-Devices sowie Laptops durchgeführt. Auf jedem Gerät wird mit den im Kapitel 4.22 aufgeführten Browsern getestet (sofern vorhanden). Die Dokumentation der durchgeführten Tests sowie die daraus abgeleiteten Massnahmen können im Anhang eingesehen werden (siehe Tabelle 10.1 und 10.2).

Nr.	Ref.	Vorgehen	Erwartetes Resultat
1	NFR03, NFR07, NFR08	<ol style="list-style-type: none"> 1. Demo-Website mit integrierter Suche mit einer Teilmenge aller Webseiten wird gebildet. 2. Der Webserver wird gestartet und die Website geöffnet. 3. Eine neue Webseite (.html File) wird dem Projekt hinzugefügt. 4. Website wird neu gebildet. 5. Webserver erhält neuen Index und Records. 6. Ausliefern des neuen Builds. 	<ol style="list-style-type: none"> 1. Der Website-Build erzeugt keine Fehlermeldungen. Es wird ein Index- sowie ein Records-File erzeugt. 2. Die Website ist erreichbar und enthält ein Suchfeld, über welches die Suchfunktionalität genutzt werden kann. 3. Mit geeigneten Suchwörtern wird die neue Webseite über die Suche gefunden.

2	NFR07, NFR08	<ol style="list-style-type: none"> 1. Laden der SDX-Website. 2. Eingabe des Suchbegriffes 'Button'. 3. Navigation zur Button-Seite. 	<ol style="list-style-type: none"> 1. Die Website lädt auf allen Geräten und Browsern problemlos. 2. Auf mobilen Geräten werden die konfigurierte Anzahl Suchergebnisse angezeigt. 3. Auf Desktopgeräten werden die konfigurierte Anzahl Suchergebnisse angezeigt. 4. Die Suchergebnisse werden korrekt dargestellt (gemäss Design). 5. Über die Suchresultate kann navigiert werden (sind klickbar).
---	-----------------	--	--

Tabelle 6.5: System-Tests

Teil IV

Projekt Dokumentation

Kapitel 7

Projektplan

7.1 Entwicklungsprozess

Für die Planung des gesamten Projekts wird der Rational Unified Process [RUP] verwendet. Die effektive Arbeit wird jedoch mit Scrum [Scr] geplant - also in Sprints. RUP unterteilt das Projekt in die folgenden vier Lifecycle-Phasen.

7.1.1 Phasen

1. Inception
2. Elaboration
3. Construction
4. Transition

Hinweis: Da ein grosser Teil der Arbeit aus Recherche und Evaluation von verschiedenen Lösungsansätzen besteht, wurde die Phase 'Elaboration' zulasten der Phase 'Construction' verlängert. Der konkrete Zeitplan ist in Kapitel 7.3 abgebildet.

7.2 Meilensteine

Das Projekt ist nebst den Phasen aus RUP und den Sprints aus Scrum in sieben Meilensteine gegliedert. Pro Meilenstein werden verschiedene Artefakte erarbeitet. Die Meilensteine und die zugehörigen Artefakte sind in 7.1 aufgeführt.

Hinweis: Die Liste der Artefakte kann, wo sinnvoll, im Verlauf des Projektes erweitert werden.

Meilenstein	Artefakte	Datum
M1: Projektplan	<ul style="list-style-type: none">• Initiale Planung• Projektplan (inkl. Risiko-Management)• Dokumentation unter Versionskontrolle (inkl. Pipeline)• Issue Management aufgesetzt• Zeiterfassung aufgesetzt• Aktualisierte Dokumentation	05.03.2023
M2: Anforderungen	<ul style="list-style-type: none">• Recherche State of the Art bez. Suche• Proto-Journeys (Use Cases, User Stories) definiert• Proto-Personas definiert• Lokale Entwicklungsumgebung aufgesetzt• Konzeptentwurf• Qualitätsmanagement dokumentiert• Aktualisierte Dokumentation	19.03.2023

<p>M3: Evaluation Lösungsvarianten</p>	<ul style="list-style-type: none"> • Dokumentation einer Übersicht verschiedener Lösungsvarianten inkl. initialer Bewertung • Prototypen von ausgewählten Lösungsvarianten • Entwurf Domänenmodell • Testkonzept • Funktionale Anforderungen definiert, priorisiert, Aufwand grob geschätzt • Nichtfunktionale Anforderungen nach SMART-Kriterien definiert • Domänen-Analyse • Überarbeiteter Projektplan • Aktualisiertes Risikomanagement (Minderung der grössten Risiken) • Aktualisierte Dokumentation 	<p>02.04.2023</p>
<p>M4: End of Elaboration</p>	<ul style="list-style-type: none"> • Definitiver Entscheid der Umsetzungsvariante inkl. Dokumentation der Entscheidungsfindung und -begründung • Finalisiertes Konzept der Suche • Protokoll Präsentation Lösungsvarianten mit Auftraggeber • Abnahme Use Cases, NFR und Hypothesen durch Auftraggeber • Aktualisierte Dokumentation 	<p>16.04.2023</p>
<p>M5: Alpha</p>	<ul style="list-style-type: none"> • MVP-Implementation • Erste Fassung der Design Decisions (allgemein & abgeleitet aus Usability-Tests) • Massnahmen bezüglich nicht-funktionalen Anforderungen mit hoher Priorität sind eingeleitet • Aktualisierte Dokumentation 	<p>14.05.2023</p>

M6: Beta	<ul style="list-style-type: none"> • Beta-Implementation (Feature Freeze) • Testprotokoll Alpha-Release • Protokoll der Usability-Tests • Bugfixes • Finale Design Decisions • Abstract • Verifikation der Adressierung nicht-funktionaler Anforderungen per Beta-Release • Aktualisierte Dokumentation 	28.05.2023
M7: Abgabe	<ul style="list-style-type: none"> • Test-Protokoll Beta-Release • Definitive Implementation (Code Freeze per 04.06.2023) • Verifikation der Adressierung nicht-funktionaler Anforderungen per finalem Release • Abnahme- und Übergabeprotokoll • Entwicklerdokumentation • Wartungsempfehlung • Finale Dokumentation • Thesis • Abgabe 	16.06.2023

Tabelle 7.1: Liste der Meilensteine

7.3 Zeitplan

In der folgenden Grafik (7.1) ist der Projektplan mit den RUP-Phasen, den geplanten Sprints und Meilensteinen sowie den angedachten Reviews mit dem Betreuer und dem Auftraggeber (siehe Kapitel 7.4) visualisiert.

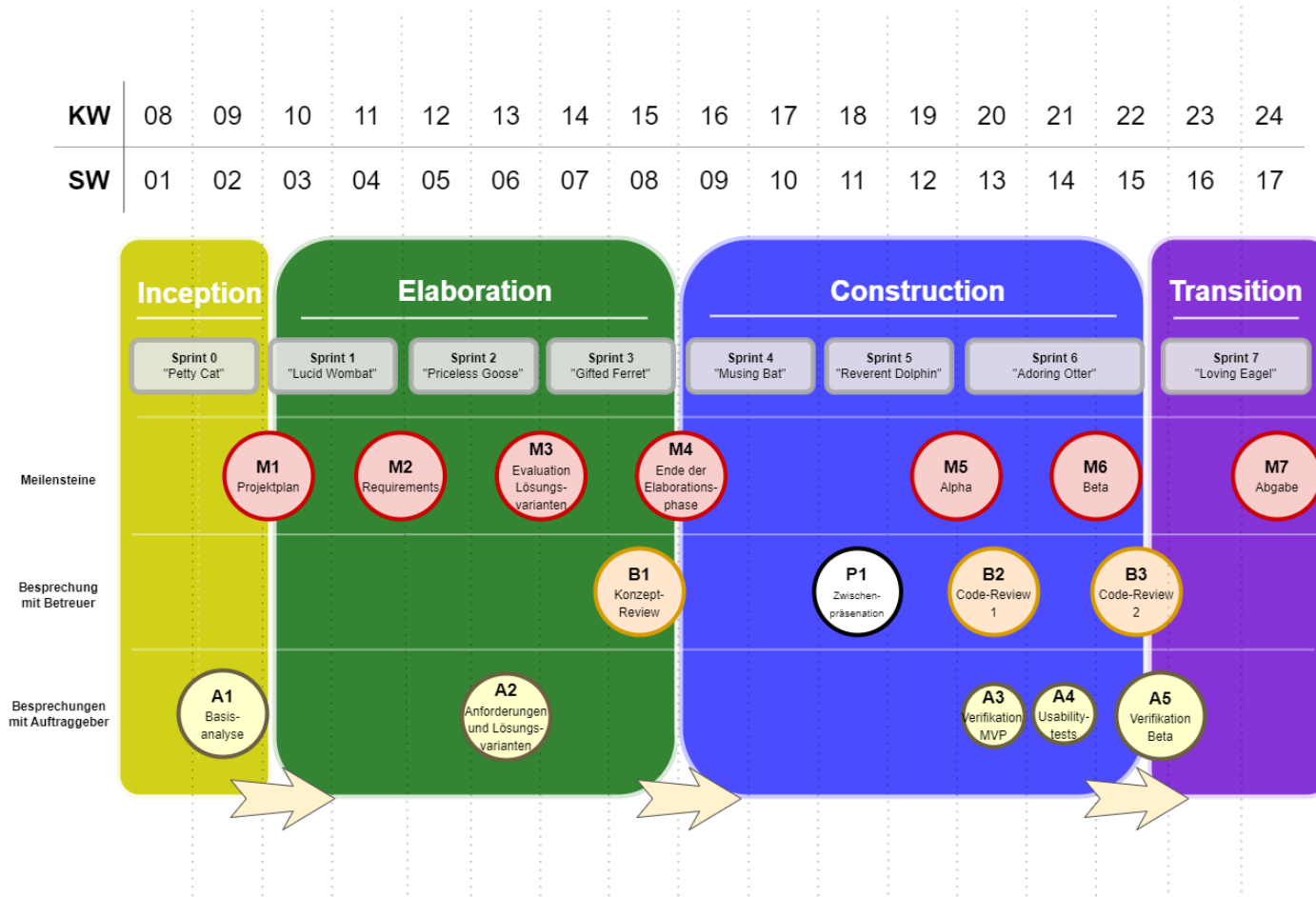


Abbildung 7.1: Projektplan

7.4 Meetings

7.4.1 Team-interne Meetings

Im Projekt-Team werden die folgenden Meetings durchgeführt:

Meeting	Häufigkeit	Wann	Wo
Sprint-Planning	Alle zwei Wochen (vor jedem Sprint)	Montag, 10:00 - 12:00 (Reserve von 14:00 - 16:00)	MS Teams
Sprint-Review	Alle zwei Wochen (nach jedem Sprint)	Montag, 10:00 - 12:00 (Reserve von 14:00 - 16:00)	MS Teams
Sprint-Retrospektive	Alle zwei Wochen (nach jedem Sprint)	Montag, 10:00 - 12:00 (Reserve von 14:00 - 16:00)	MS Teams
Weekly-Scrum	Jede Woche	Montag, 10:00 - 12:00 (Reserve von 14:00 - 16:00)	MS Teams
Backlog-Refinement	Alle zwei Wochen (mit dem Weekly Scrum)	Montag, 10:00 - 12:00 (Reserve von 14:00 - 16:00)	MS Teams

Tabelle 7.2: Team-interne Meetings

Hinweis: Da die Projektmitglieder nur mit begrenztem Pensum (Rolf Oberhänsli 50% & Manu Weber 50%) am Projekt arbeiten, wird anstatt eines 'Daily Scrum' ein 'Weekly Scrum' durchgeführt.

7.4.2 Reviews mit Betreuer

Wöchentlich wird ein Status-Meeting mit dem Betreuer Prof. Dr. M. Stolze durchgeführt. Dieses findet jeweils am Mittwoch von 15:00 - 16:00 Uhr über MS Teams statt. Ausnahmen werden bilateral abgesprochen.

Zudem finden die folgenden drei grösseren Reviews statt:

Review	Datum	Teilnehmer	Wo
B1: Konzept-Review	12.04.2023	M. Stolze, Projekt-Team	MS Teams
B2: Code-Review 1	15.05.2023	M. Gfeller, Projekt-Team	MS Teams
B3: Code-Review 2	01.06.2023	M. Gfeller, Projekt-Team	MS Teams

Tabelle 7.3: Grössere Reviews mit Betreuer oder Assistent

Des Weiteren wurden die folgenden Termine für die Zwischenreview und die Prüfung der BA vereinbart:

Titel	Datum	Teilnehmer	Wo
Zwischenreview	03.05.2023	M. Stolze, M. Flückiger, Ivan Büttler, Projekt-Team	MS Teams
Prüfung	04.07.2023	M. Stolze, M. Flückiger, Ivan Büttler, Projekt-Team	Rapperswil

Tabelle 7.4: Termine BA

7.4.3 Besprechungen mit Auftraggeber

Folgende grösseren Besprechungen werden mit Alex Kühne von der Swisscom (Auftraggeber) geplant. Bei Bedarf können weitere Meetings bilateral vereinbart werden.

Absprache	Datum	Teilnehmer	Wo
A1: Basis Analyse	03.03.2023	A. Kühne, Projekt-Team	MS Teams
A2: Anforderungen und Lösungsvarianten	31.03.2023	A. Kühne, Projekt-Team	MS Teams
A3: Verifikation MVP	17.05.2023	A. Kühne, Projekt-Team	MS Teams
A4: Usability Testing	25.05.2023	A. Kühne, Projekt-Team, Enduser	Förllibuckstrasse 181, 8005 Zürich
A5: Verifikation Beta	02.06.2023	A. Kühne, Projekt-Team	MS Teams

Tabelle 7.5: Reviews Swisscom

7.5 Rollen

Im Projekt-Team sind die folgenden Rollen verteilt:

Rolle	Zugewiesene Person(en)	Stellvertretung
Meeting Organisation	Manu Weber	Rolf Oberhänsli
Protokollführer	Rolf Oberhänsli	Manu Weber
Dokumentations-Master	Manu Weber	Rolf Oberhänsli
Risiko-Manager	Rolf Oberhänsli	Manu Weber
Time-Report-Master	Manu Weber	Rolf Oberhänsli
Scrum Master	Rolf Oberhänsli	Manu Weber
Product Owner	Manu Weber	Rolf Oberhänsli
Entwickler	Rolf Oberhänsli und Manuel Weber	-

Tabelle 7.6: Rollenverteilung

7.5.1 Details zu den definierten Rollen

- Die Scrum-Rollen (Scrum Master, Product Owner, Entwickler) werden wie von Scrum [Scr] definiert eingesetzt. Aufgrund dessen, dass der eigentliche Product Owner extern (Swisscom) ist, wird die Scrum-Rolle zusätzlich im Team vergeben.
- Meeting-Organisation: Organisiert Meetings und Reviews mit dem Auftraggeber und dem Betreuer.
- Protokollführer: Führt das Protokoll während Meetings und Reviews. Hat die Hauptverantwortung für die Protokolle.
- Dokumentations-Master: Hat die Hauptverantwortung dafür, dass die Dokumentation up-to-date gehalten wird. Die Dokumentation wird im Team geschrieben, diese Person sorgt lediglich dafür, dass die Dokumentation stetig aktuell gehalten wird.
- Der Risiko-Manager ist für das Managen der Risiken verantwortlich. Dies beinhaltet die kontinuierliche Neubeurteilung der Risiken sowie das Aktualisieren des Risikoentwicklungsgraph.
- Der Time-Report-Master ist für das Generieren und Aktualisieren der Grafiken bezüglich der Zeiterfassung in der Dokumentation verantwortlich.

7.6 Ansprechpersonen

Folgende Ansprechperson steht bei der Swisscom zur Verfügung.

Name	Rolle
Alex Kühne	Product Owner des Design-Systems (Swisscom Digital Experience)

Tabelle 7.7: Ansprechpersonen Swisscom

7.7 Risiko-Management

7.7.1 Risikomatrix

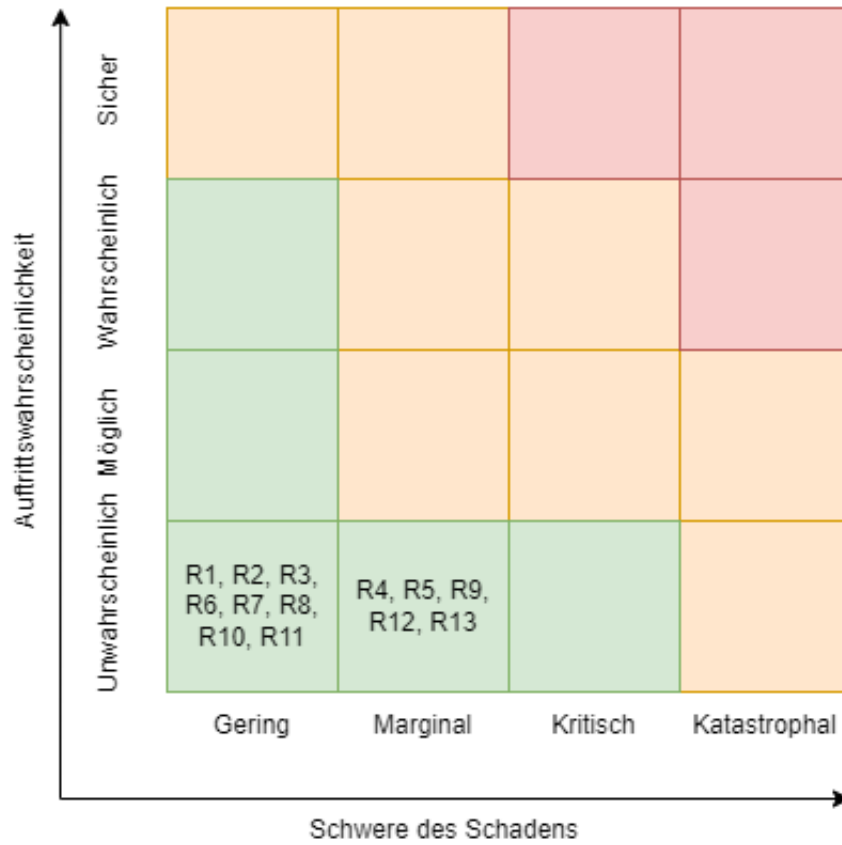


Abbildung 7.2: Risikomatrix

7.7.2 Risiken

In Tabelle 7.8 werden alle erkannten Risiken im Projekt aufgelistet.

Nr.	Titel	Beschreibung	Probabilität	Schwere
1	Technologiewandel	Im Projekt werden Technologien eingesetzt, die sich fortschreitend weiterentwickeln, was ein Mismatch verursacht. Die angestrebte Lösung muss daher ohne grossen Mehraufwand auf solche Änderungen reagieren können.	Unwahrscheinlich	Gering

2	Gesundheit	Ein Teammitglied fällt durch Krankheit oder Unfall für längere Zeit aus. Die Aufgaben dieses Teammitglieds werden nicht ausgeführt.	Unwahrscheinlich	Gering
3	Kommunikation	Ein Teammitglied beteiligt sich nicht an Meetings oder diskutiert Unstimmigkeiten nicht offen. Dies kann zu weiteren Problemen führen, die negativen Einfluss auf das Projekt haben.	Unwahrscheinlich	Gering
4	Projektumfang	Der Umfang des Projekts ist zu gross und die angestrebten Ziele sind nicht realistisch.	Unwahrscheinlich	Marginal
5	Wissen	Mangelndes technisches Wissen und Erfahrung von Teammitgliedern beeinflusst und verzögert die zeitliche Planung.	Unwahrscheinlich	Marginal
6	Externe Gefahren	Die Energiekrise, Unwetter oder andere nicht beeinflussbare Faktoren machen ein Weiterarbeiten am Projekt unmöglich.	Unwahrscheinlich	Gering
7	Anforderungen	Falsche, unvollständige oder fehlende funktionale oder nicht-funktionale Anforderungen führen zu unzureichender Funktionalität in der Anwendung.	Unwahrscheinlich	Gering
8	Konzept	Das ausgearbeitete Konzept für die Suche ist nicht realisierbar oder erfüllt funktionale oder nicht-funktionale Anforderungen nicht.	Unwahrscheinlich	Gering

9	Zeitmanagement	Durch mangelhafte Zeitplanung kann die Applikation nicht im vorgegebenen Zeitrahmen umgesetzt werden.	Unwahrscheinlich	Marginal
10	Softwarequalität	Unzureichende Tests führen zu unentdeckten Fehlern und Bugs im Code.	Unwahrscheinlich	Gering
11	Pandemie und Krieg	Einflüsse aufgrund des Krieges in der Ukraine oder der Pandemie bringen ausserordentliche Situationen hervor und schränken den Fortschritt des Projektes ein.	Unwahrscheinlich	Gering
12	Usability	Die ausgearbeitete Lösung ist nicht benutzerfreundlich und liefert nicht die erwünschten Suchresultate.	Unwahrscheinlich	Marginal
13	Security	Sicherheitsanforderungen werden nicht erkannt und führen zu Sicherheitslücken in der Applikation.	Unwahrscheinlich	Marginal

Tabelle 7.8: Risiken

7.7.3 Risikominimierung und -prävention

In der folgenden Tabelle (Tabelle 7.9) werden alle Massnahmen zur Abschwächung von Risiken aufgelistet.

Betroffene Risiken	Massnahme
R1	Software wird modular und isoliert aufgebaut, sodass das Ersetzen von Technologien ohne grösseren Aufwand umgesetzt werden kann. Zusätzlich werden die Tools ausgiebig evaluiert und analysiert. Es werden nur Bibliotheken verwendet, die langlebig sind und aktiv gepflegt werden.
R2, R3, R4, R6, R11	Es wird Fokus auf Engagement innerhalb des Teams, direkte Kommunikation, wöchentliche Meetings und Statusaktualisierungen gelegt. Falls ein extremer Ausfall eintritt wird eine Verzögerung des Abgabetermins beim Studiengang beantragt.
R4, R7, R12	Es werden intensive Reviews und Usability-Tests mit dem Berater, Industriepartner und Endusern durchgeführt. Zudem wird das Projekt sorgfältig geplant. Dafür werden gängige Praktiken eingesetzt.
R5	Während der Inception-Phase arbeiten sich die Teammitglieder in unbekannte Technologien ein und es werden Pufferzeiten für die Entwicklung eingeplant.
R8, R12	Das Konzept wird im Team geplant und mit dem Berater besprochen. Zudem werden verschiedene Konzepte evaluiert und geprüft. Für eine klare Kommunikation und Planung wird eine Domänen-Analyse durchgeführt und Architekturdiagramme erstellt. Falls nötig werden zusätzliche Experten hinzugezogen, um allfällige Probleme zu lösen.
R9	Die Planung wird sauber aufgeleitet und Pufferzeiten werden miteinberechnet. Im Sprint Review werden geleistete sowie verbleibende Arbeitsstunden analysiert und mit den Arbeitspaketen verglichen. Wo nötig werden Anpassungen an den Arbeitspaketen vorgenommen.
R10, R13	Das Testen wird priorisiert erledigte Tasks müssen die Definition of Done (siehe Kapitel 6.4) erfüllen.

Tabelle 7.9: Massnahmen zur Risikominimierung und -prävention

7.7.4 Risikoentwicklung

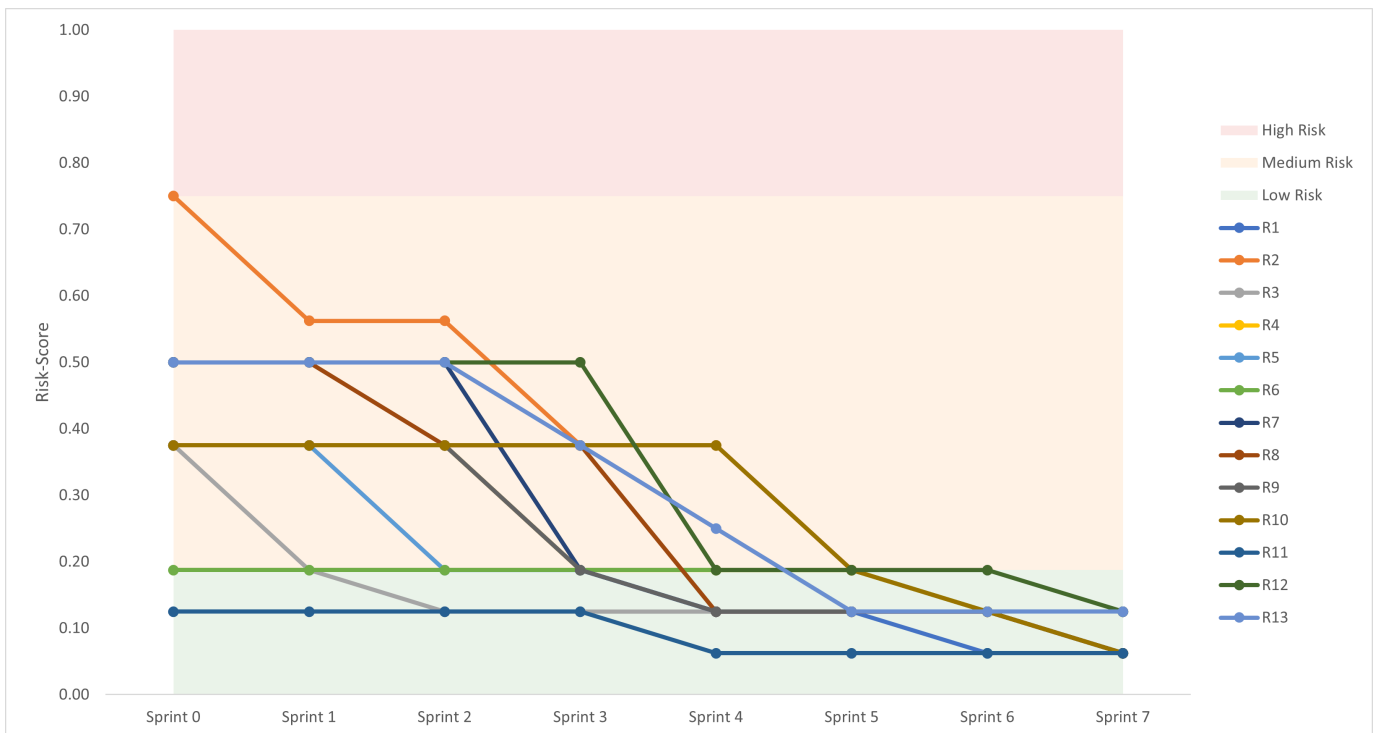


Abbildung 7.3: Risikoentwicklungsgraph

7.8 Issue-Management

Die Issues werden über GitLab in der Infrastruktur der OST verwaltet. Die Issues werden auf einem Board mit folgenden Stages organisiert:

1. Epics

- Anforderungen werden als Epics in dieser Stage grob granular ausgewiesen.
- Der Aufwand von Epics wird aufgrund einer relativen Schätzung-Skala (T-Shirt-Größen) geschätzt.
- Epics verbleiben als Übersicht in dieser Stage.

2. Product Backlog

- Epics werden zu User Stories verfeinert.
- Epics und User Stories werden miteinander verlinkt.
- User Stories werden mit dem Präfix [US] im Product Backlog abgelegt.
- Der Aufwand von User Stories wird mit einer relativen Schätzung-Skala (Story-Punkte) geschätzt.

- Aus den User Stories werden Arbeitspakete abgeleitet.
- Tasks werden mit den User Stories verlinkt und in einem ersten Schritt im Product Backlog abgelegt.
- Der Aufwand von Tasks wird absolut (in Stunden) geschätzt.

3. Sprint Backlog

- Tasks, die im aktuellen Sprint bearbeitet werden sollen, werden während des Sprint Planning im Sprint Backlog abgelegt.
- Wichtig: Nur Tasks, deren Aufwand bereits geschätzt wurde, können in einen Sprint eingeplant werden.

4. In Bearbeitung

- Tasks, an denen die Teammitglieder aktiv arbeiten, befinden sich in dieser Stage.

5. In Review

- Tasks, die bearbeitet und bereit für ein Review sind, befinden sich in dieser Stage.

6. Close

- Tasks und User Stories, die gemäss Kapitel 6.4 abgeschlossen sind, werden hier abgelegt.

7.9 Versionskontrolle

Die Versionskontrolle der Dokumentation sowie des Quellcodes läuft über GitLab in der Infrastruktur der OST. Der Code wird in einem separaten Ordner im Repository der Dokumentation gepflegt. Der Grund hierfür ist einerseits, dass somit alle Issues in einem Repository gepflegt werden können. Andererseits wird der geschriebene Code voraussichtlich eine überschaubare Quantität aufweisen, sodass ein eigenes Repository nicht notwendig ist.

Link auf das GitLab-Repository: <https://gitlab.ost.ch/rolf.oberhaensli/ba-dokumentation>.

7.10 Zeiterfassung

Für die Zeiterfassung werden die von GitLab mitgelieferten Features genutzt. Somit kann die Zeit direkt auf die in GitLab verwalteten Issues gebucht werden. Die Issues an sich enthalten Labels und eine Referenz auf den betreffenden Meilenstein. Für den Export der Time-Reports wird das CMD-Tool `gtt` [gtt] verwendet. Die Time-Reports werden in eine SQLite-Datenbank importiert, woraus anhand vorbereiteter Queries verschiedene Informationen für die Report-Generierung abgefragt werden können.

Details zum Prozess sind hier dokumentiert: <https://gitlab.ost.ch/rolf.oberhaensli/ba-dokumentation/-/wikis/Time-Reports>.

Teil V

Ergebnisse

Kapitel 8

Resultate

Das Resultat dieser Bachelorarbeit ist das Search-Integrator-Framework, das als NPM-Paket publiziert worden ist: <https://www.npmjs.com/package/search-integrator>. Des Weiteren wurde nebst dieser Produkt- und Projektdokumentation ein Anwendungsbeispiel des Frameworks für die Integration einer Suche in die SDX-Website erarbeitet.

8.1 Verifikation des Resultats

Das entstandene Search-Integrator-Framework wurde mit verschiedenen Mechanismen und Metriken verifiziert. Diese Überprüfungen wurden anhand des erstellten Anwendungsbeispiels der Integration einer Suche in die SDX-Website durchgeführt und werden in den folgenden Unterkapiteln ausgeführt.

8.1.1 Grösse der zusätzlich benötigten Ressourcen einer integrierten Suche

Die Grösse der zu ladenden Ressourcen für die SDX-Website ist in Abbildung 8.1 ersichtlich. Für die Darstellung des Suchfelds und der Suchresultate wird aktuell ein Stylesheet mit einer Grösse von rund 996 Bytes sowie die dazugehörige Logik, in Form eines JavaScript-Files, mit einer Grösse von 3.4kB ausgeliefert. Das search-integrator.js File ist mit 175kB die grösste Ressource. Sie beinhaltet die Suchlogik, alle Records der Website und den Suchindex. Insgesamt wird also eine Datenmenge von rund 180kB zusätzlich zu einer SDX-Webseite geladen. Diese Grössen basieren auf dem Anwendungsbeispiel, das für die SDX-Website von Swisscom implementiert wurde und können stark variieren, da der Index je nach Grösse der zu indexierenden Website stark variiert.

<input checked="" type="checkbox"/>	search-component.css	200	stylesheet	996 B
<input checked="" type="checkbox"/>	search-component.js	200	script	3.4 kB
<input checked="" type="checkbox"/>	tabs.js	200	script	2.6 kB
<input checked="" type="checkbox"/>	smooth-anchor-scrolli...	200	script	2.2 kB
<input checked="" type="checkbox"/>	search-integrator.js	200	script	175 kB

Abbildung 8.1: Ausgelieferte Ressourcen

Wichtig zu erwähnen ist, dass alle drei Ressourcen im Browser gecacht werden und somit nur beim Laden der ersten Webseite vom Server geladen werden müssen. Die Grösse der HTML-Datei einer SDX-Webseite wird durch das Hinzufügen der Script- und Link-Tags sowie dem HTML der Such-Komponente nicht (messbar) vergrössert.

8.1.2 Lighthouse: Analyse einer mit einer Suche angereicherten Website

Mithilfe von Lighthouse [lig] wurde das Anwendungsbeispiel der SDX-Website vor und nach der Integration der Suche analysiert. Es wurde mit den verschiedenen Modi von Lighthouse gearbeitet. Die Testumgebung entsprach dabei einer lokalen Entwicklungsumgebung. Mithilfe des Navigation-Modus wurde der Pageload beim Aufruf von SDX gemessen. Der Timespan-Modus wurde verwendet, um die Navigation zwischen den Varianten, mit und ohne Suche, zu vergleichen. Mithilfe des Snapshot-Modus wurde die Darstellung der Suche und der Suchresultate überprüft.

Navigation-Modus Im Navigationsmodus konnten im Report zwischen der Website mit und ohne Suche keine Veränderungen festgestellt werden.

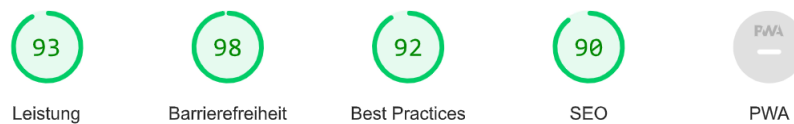


Abbildung 8.2: Lighthouse Navigation Report - Navigation auf der Website ohne integrierter Suche

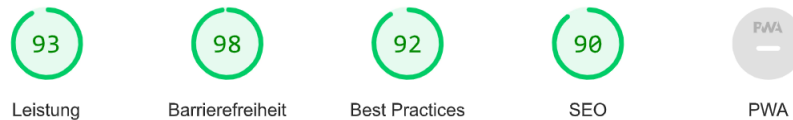


Abbildung 8.3: Lighthouse Navigation Report - Navigation auf der Website mit integrierter Suche

Timespan-Modus Der Time-Span Modus wurde genutzt, um die Navigation auf der Website mit der Suche und über das Menü zu vergleichen. Um die beiden Varianten vergleichbar zu machen wurde der folgende Ablauf definiert.

1. Page Load
2. Navigation zur Webseite 'Accessibility' in der Foundation-Kategorie
3. Navigation zur Webseite 'Accordions' in der Components-Kategorie
4. Navigation zur Webseite 'Buttons' in der Components-Kategorie
5. Navigation zur Webseite 'Getting started' in der Developers-Kategorie
6. Navigation zur Webseite 'Tables' in der Components-Kategorie

7. Navigation zur Webseite 'CSS Variables' in der Developers-Kategorie

8. Navigation zur Webseite 'Icons' in der Foundation-Kategorie

Dieser Ablauf wurde in beiden Varianten durchgeführt. Die Auswertung zeigt auf den ersten Blick keine Veränderungen, da die Messkriterien gleichermassen erfüllt wurden. Bei genauerer Betrachtung der Leistungsmessungen von Lighthouse wurde Folgendes erkannt:

1. Die 'Total Blocking Time' wurde durch den Mehraufwand an Berechnungen durch die Suche erhöht.
2. Die 'Cumulative Layout Shifts' konnten durch die Suche komplett eliminiert werden. Dies ist darauf zurückzuführen, dass das Menü nicht ein und ausgeklappt wird und entsprechend durch das Overlay der Suche keine Änderungen an der zugrunde liegenden Webseite vorgenommen werden.
3. Die 'Interaction to Next Paint' ist etwas kleiner mit der Suche, da man schlicht und einfach weniger Klicks und somit weniger Zeit zum Navigieren benötigt.

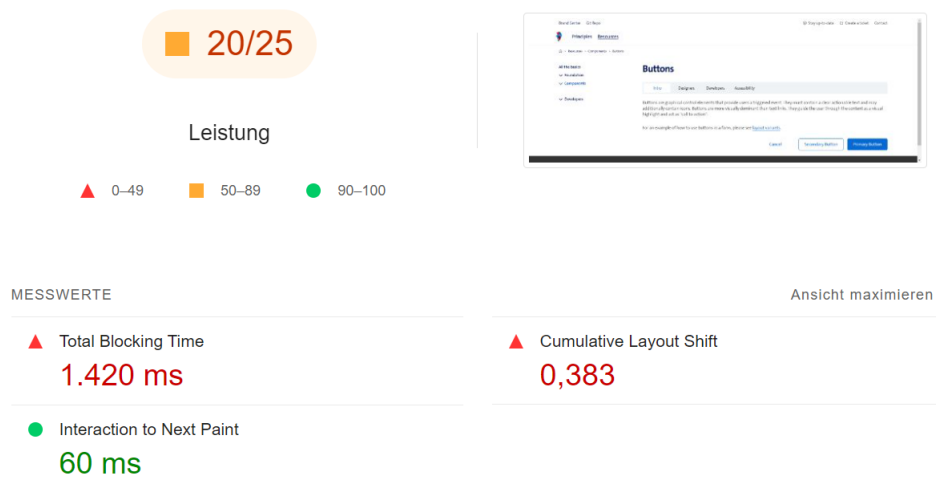


Abbildung 8.4: Lighthouse Timespan Report - Navigation auf der Website ohne integrierter Suche

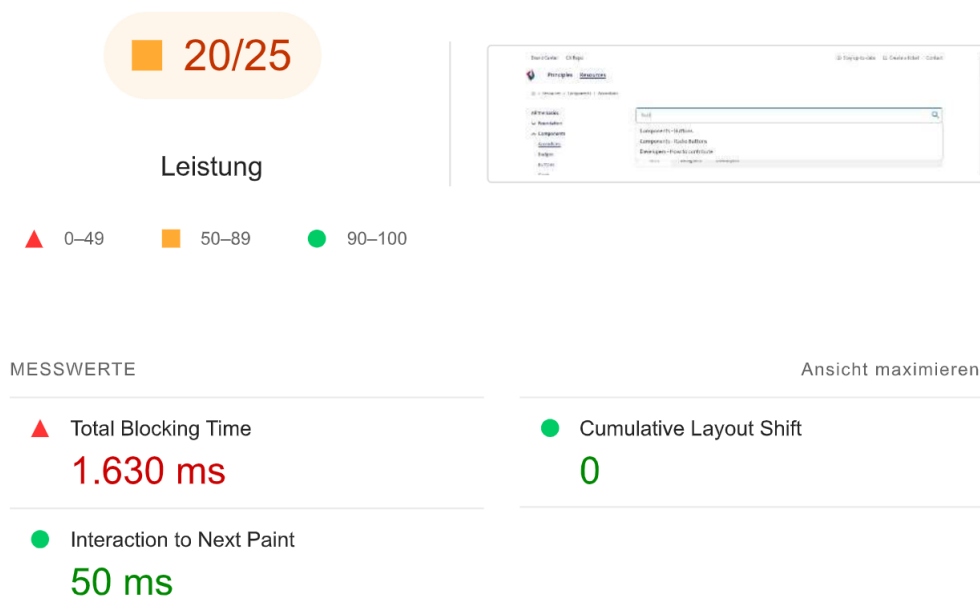


Abbildung 8.5: Lighthouse Timespan Report - Navigation auf der Website mit integrierter Suche

Snapshot Mithilfe des Snapshot-Modus wurde untersucht, ob die Darstellung der Suchresultate und des Suchfelds Fehler enthalten, welche zur Verschlechterung der Webseite führen. Im Snapshot-Report konnten jedoch keine Fehler, welche aus der Integration der Suche entstanden sind, erkannt werden.

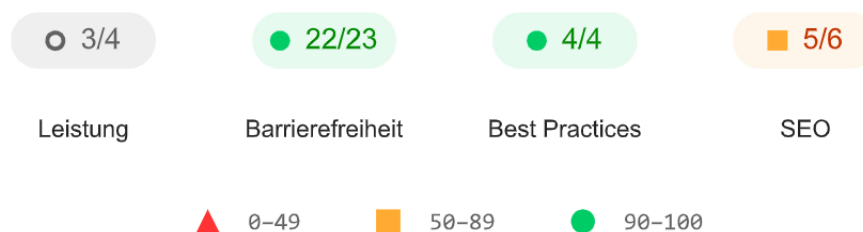


Abbildung 8.6: Lighthouse Snapshot Report - Darstellung Suche und Suchresultate

8.1.3 Performance einer Website mit einer integrierten Suche

Zur Messung des Einflusses der integrierten Suche auf die Performance einer Website wurden die DevTools von Google Chrome verwendet. Bei der Auswertung der Website-Performance wurde vor allem die Time to interactive (TTI) [tim] betrachtet. Die Time to interactive wurde anhand der von web.dev [weba] bereitgestellten Informationen und Anleitung berechnet [tti]. Die Messungen wurden auf einem lokalen Entwicklungssystem durchgeführt. Es wurden folgende Metriken und Zeitpunkte protokolliert. Die protokollierten Werte entsprechen dem Durchschnitt von 5 Messungen pro Variante. Die Daten zeigen, dass die integrierte Suche keinen signifikanten Einfluss auf die Leistung der einer Webseite hat.

- First Contentful Paint (FCP) [fcp]

- DOMContentLoaded (DCL) [[dcl](#)]
- Largest Contentful Paint (LCP) [[lcp](#)]

SDX ohne Suche

- FCP: 249.0 ms
- LCP: 883.0 ms
- DCL: 283.5 ms
- Zeitpunkt Abschluss des letzten Long Tasks (50ms): ca. 650.0 ms
- TTI entspricht 650.0 ms

SDX mit Suche

- FCP: 229.8 ms
- LCP: 863.8 ms
- DCL: 337.8 ms
- Zeitpunkt Abschluss des letzten Long Tasks (50ms): ca. 660.0 ms
- TTI entspricht 660.0 ms

8.1.4 Performance der integrierten Suche

Um die Performance der Suche zu analysieren, wurde mithilfe der Leistungsmessung der Chrome Entwickler Tools eine Analyse von Suchanfragen durchgeführt. Abbildung 8.7 zeigt ein Beispiel der Analyse einer Suchanfrage. Insgesamt wurden rund 50 Suchanfragen analysiert.

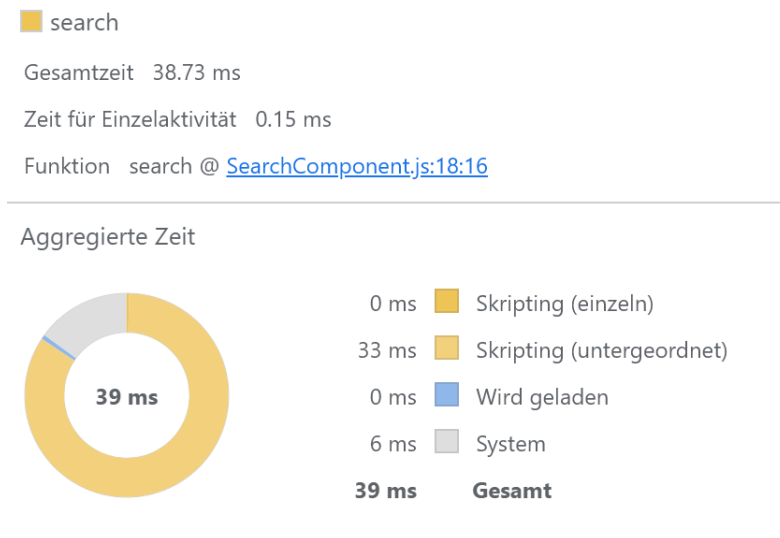


Abbildung 8.7: Beispiel einer Suchanfrage

Die Analyse hat gezeigt, dass sich die Suche im Bereich von 25ms bis 100ms pro Suchanfrage bewegt. Es konnte keine Suchanfrage provoziert werden, welche länger als 100ms dauerte. Die durchschnittliche Dauer einer Suchanfrage über die 50 Messungen war 36ms.

8.2 Verifikation funktionaler und nicht-funktionaler Anforderungen

Zum Abschluss dieses Kapitels wird das Erreichen der funktionalen sowie nicht-funktionalen Anforderungen aufgezeigt und bewertet.

8.2.1 Funktionale Anforderungen

UC01 Suchbedürfnisse

Suchbedürfnisse können nach der Integration der Suche erfolgreich befriedigt werden.

✓ Der Use Case wurde umgesetzt.

UC02 Synonyme

Die Integration von Synonymen in die zu indexierenden Daten ist möglich. Zusätzlich hat man die Möglichkeit eine API zur Abfrage von Synonymen anzubinden, sofern dies gewünscht ist. Suchanfragen mit einem Synonym werden erfolgreich aufgelöst und die richtigen Informationen gefunden. Für Swisscom wurde eine Synonymliste für UI-Komponenten erstellt und berücksichtigt. Die Grundlage der für SDX verwendeten Synonyme bilden die von Open-UI [ope] definierten Komponenten-Namen.

✓ Der Use Case wurde umgesetzt.

UC03 Autocomplete

Bei der Eingabe von Suchanfragen wird ab der Eingabe des zweiten Buchstabens automatisch eine Suchanfrage ausgelöst und die Suchresultate angezeigt. Die Kontrolle über das Verhalten der Autocomplete-Funktionalität obliegt dem Anwender des Search-Integrator-Frameworks.

✓ Der Use Case wurde umgesetzt.

UC04 Keine Suchresultate

Dem Benutzer der Suche wird eine Meldung angezeigt, sofern gar keine oder nur sehr schlechte Suchresultate gefunden wurden. Die Kontrolle über die Darstellung der Meldung, falls keine Suchresultate gefunden werden, obliegt der Kontrolle des Anwenders des Search-Integrator-Frameworks.

✓ Der Use Case wurde umgesetzt.

UC05 Konfigurierbare Suche

Das Framework ist vollumfänglich konfigurierbar und wurde nach dem Grundsatz 'Convention over Configuration' [con] entwickelt. Nicht nur die Suche kann im Rahmen der verwendeten Lösung konfiguriert werden, sondern auch die einzelnen Komponenten des Frameworks können über ein Konfigurationsfile gesteuert werden.

✓ Der Use Case wurde umgesetzt.

UC07 Zusätzliche Informationen zu Suchresultaten

Die Darstellung der Suchresultate kann vollständig durch den Nutzer gesteuert werden. Dabei stehen dem Benutzer jegliche Informationen zu Suchresultaten zur Verfügung.

Für die SDX-Website wird in der Darstellung der Suchresultate ein Breadcrumb dargestellt, damit der Ort der Webseite innerhalb der Website bereits anhand des Suchresultats erkannt werden kann.

✓ Der Use Case wurde umgesetzt.

Ausstehende Use Cases

Folgende Use Cases wurden aus zeitlichen Gründen und/oder Architektur-Constraints nicht umgesetzt. Diese Use Cases wurden an den Auftraggeber in der Transition-Phase (vergleiche Kapitel 7 übergeben.

- UC06 Kategorisierte Suchresultate
- UC08 Suchfilter

- UC09 Suchhistorie
- UC10 Visuelle Unterstützung bei der Komponenten-Suche
- UC11 Verzeichnisstruktur des Outputs
- UC12 Logging und Auswertung
- UC13 Anbindung Webcrawler

8.2.2 Nicht-funktionale Anforderungen

NFR01

Die Suche sowie die Anzeige der Suchresultate ist intuitiv verständlich und liefert den Usern die erwünschten Suchergebnisse.

✓ Das NFR ist erfüllt.

Die Usability-Tests haben gezeigt, dass die Benutzer die Suche intuitiv bedienen können und die erwünschten Resultate erhalten.

NFR02

Die Such-Funktionalität wird in die bestehende SDX-Design-Bibliothek integriert. Die Time-to-Interactive der Website wird auf modernen Geräten minimal, falls überhaupt, verlängert.

✓ Das NFR ist erfüllt.

Die Performance-Messungen haben gezeigt, dass die Metriken keine signifikanten Änderungen erfahren.

NFR03

Die Such-Funktionalität stellt keine Anforderungen an Webseiten von SDX in Bezug auf die erfolgreiche Ausführung des Build-Prozesses der Website. D.h., auch wenn eine Webseite in den Build-Prozess der SDX-Website miteinbezogen wird, die nicht für die Suche optimiert ist, funktioniert der Build-Prozess der SDX-Website weiterhin.

✓ Das NFR ist erfüllt.

Die Systemtests haben gezeigt, dass die Erweiterung einer Website um zusätzliche Webseiten kein Problem für die Integration einer Suche darstellt und dass die neu hinzugefügten Webseiten von der Suche berücksichtigt werden.

NFR04

Die (clientseitige) Suche auf der SDX-Website soll von Benutzern als performant wahrgenommen werden. Die Such-Funktionalität muss somit effizient implementiert sein, sodass sie trotz den beschränkten Ressourcen im Browser effizient ist.

✓ Das NFR ist erfüllt.

Die Suche ist performant wie die Performance-Messungen gezeigt haben. In den Usability-Tests wurde dies ebenso berücksichtigt. Die Testenden waren sehr zufrieden mit der Performance.

NFR05

Für den Fall, dass das eingesetzte Such-Framework nicht mehr weiterentwickelt wird und/oder Sicherheitsrisiken birgt, soll dieses möglichst einfach ausgetauscht werden können. Hierzu wird das verwendete Such-Framework hinter einem Interface gekapselt.

✓ Das NFR ist erfüllt.

Die eingesetzte Such-Bibliothek ist hinter einer Fassade gekapselt.

NFR06

Bei der Suche wird nicht der exakte Suchbegriff verwendet. Trotzdem soll das Suchbedürfnis eines Users befriedigt werden können.

✓ Das NFR ist erfüllt.

System- und Usability-Tests haben dieses NFR bestätigt.

NFR07

Die Such-Funktionalität funktioniert in allen modernen Browsern, die von SDX unterstützt werden. Konkret sind das Google Chrome, Apple Safari, Mozilla Firefox und Microsoft Edge (Chromium). Zudem funktioniert sie sowohl auf Desktop- sowie auch auf mobilen Endgeräten.

✓ Das NFR ist erfüllt.

Systemtests und eine manuelle Durchführung der Szenarien der Usability-Tests haben dieses NFR bestätigt.

NFR08

Die Darstellung des Suchfeldes sowie der Suchresultate ist sowohl auf mobile Devices sowie auf Desktop-Geräte optimiert.

✓ Das NFR ist erfüllt.

Systemtests und eine manuelle Durchführung der Szenarien der Usability-Tests haben dieses NFR bestätigt.

NFR09

Die Business-Logik des Parsers sowie der Such-Funktionalität weist eine Code-Coverage von $\geq 85\%$ auf.

✓ Das NFR ist erfüllt.

Es wurde eine Code-Coverage $> 85\%$ erzielt.

NFR10

Die Such-Funktionalität verwendet aktuelle Versionen von den eingesetzten Libraries.

✓ Das NFR ist erfüllt.

Beim Alpha- sowie Beta-Release und zum Abschluss des Projektes wurden die Abhängigkeiten überprüft und falls nötig aktualisiert. Zudem wurde eine Wartungsempfehlung erstellt und an den Auftraggeber übergeben.

NFR11

Die Such-Funktionalität ist so aufgebaut, dass zusätzliche Funktionen zur Verfügung gestellt werden können, ohne dass der Code in einer nicht von den neuen Funktionen tangierten Komponenten angepasst werden muss.

✓ Das NFR ist erfüllt.

Die Weiterentwicklung hat gezeigt, dass die Applikation wenig Abhängigkeiten enthält und sehr modular aufgebaut ist.

NFR12

Der Anwender des Frameworks möchte die komplette Kontrolle über die Darstellung des Suchfeldes sowie der Darstellung der Resultate.

✓ Das NFR ist erfüllt.

Zwischen Alpha und Beta-Release konnte verifiziert werden, dass der Anwender des Frameworks die volle Kontrolle über die Darstellung des Suchfelds sowie der Suchresultate hat. Der Anwender kann dies durch die Bereitstellung der nötigen Ressourcen (HTML, CSS, JS) steuern.

NFR13

Falls zu einer Suchanfrage keine Resultate gefunden werden konnten, wird dies dem User angezeigt, sodass der User die Möglichkeit hat seine Suchanfrage umzuformulieren. Zudem werden Synonyme eingesetzt, um breitere Suchanfragen zu ermöglichen.

✓ Das NFR ist erfüllt.

Für die Usability-Tests wurden Szenarien definiert, in denen ein Suchbedürfnis nicht befriedigt werden konnte. Die Testenden erkannten erfolgreich, dass keine Resultate respektive Informationen zu einem Thema auf der Website enthalten sind. Zudem konnte in den Usability Tests sowie den Systemtests verifiziert werden, dass Suchresultate auch über Synonyme von Begriffen, die auf einer Webseite enthalten sind, gefunden werden.

8.3 Repository-Statistiken

8.3.1 Search-Integrator & Anwendungsbeispiel SDX-Suche

1. Lines of Code (.ts): ca. 3500
2. Anzahl Klassen: 6
3. Anzahl Commits: 275

8.3.2 BA-Dokumentation

1. Lines of Code (.tex): ca. 6000
2. Anzahl Commits: 356
3. Anzahl Issues: 204

8.4 Zeiterfassungsreport

Die Zeiterfassung erfolgte gemäss Definition von Kapitel Zeiterfassung im Projektplan (Kapitel 7.10). Der Zeitplan konnte erfolgreich eingehalten werden, inklusive allen Meetings, Feature- und Code-Freeze und Meilensteinen. Das Team konnte ein mehrheitlich stetiges Arbeitstempo von Anfang an beibehalten und kam nie in Verzögerung.

Hinweis: Die Sprintdauer war grundsätzlich zwei Wochen. Die Ausnahme bildete Sprint 6 - dieser dauerte drei Wochen. Der Grund dafür war die ungerade Anzahl zur Verfügung stehende Semesterwochen für das Bearbeiten der Bachelorarbeit.

Die gesamten Arbeitszeiten des Teams sind folgende:

- Manu Weber: 395.0
- Rolf Oberhänsli: 382.833
- **Total: 777.833**

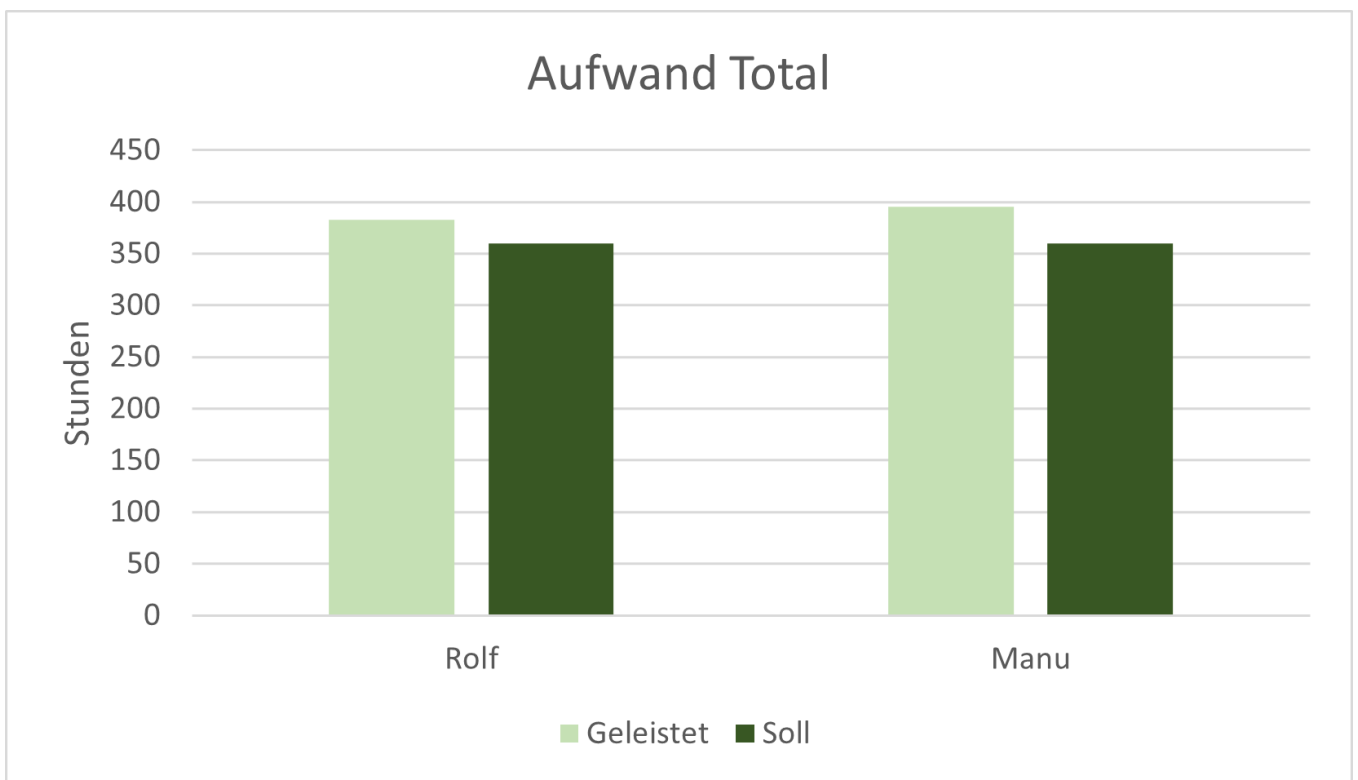


Abbildung 8.8: Aufwand Total

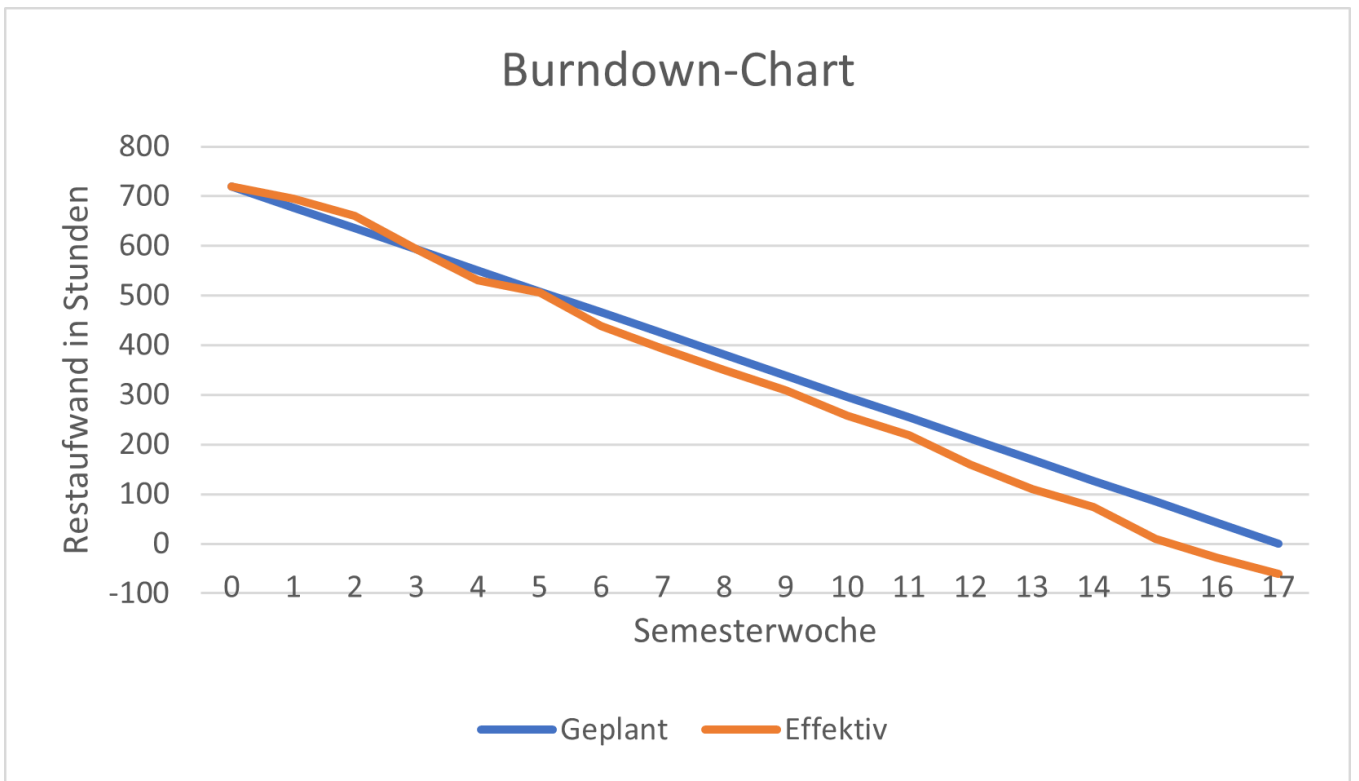


Abbildung 8.9: Burndown-Chart

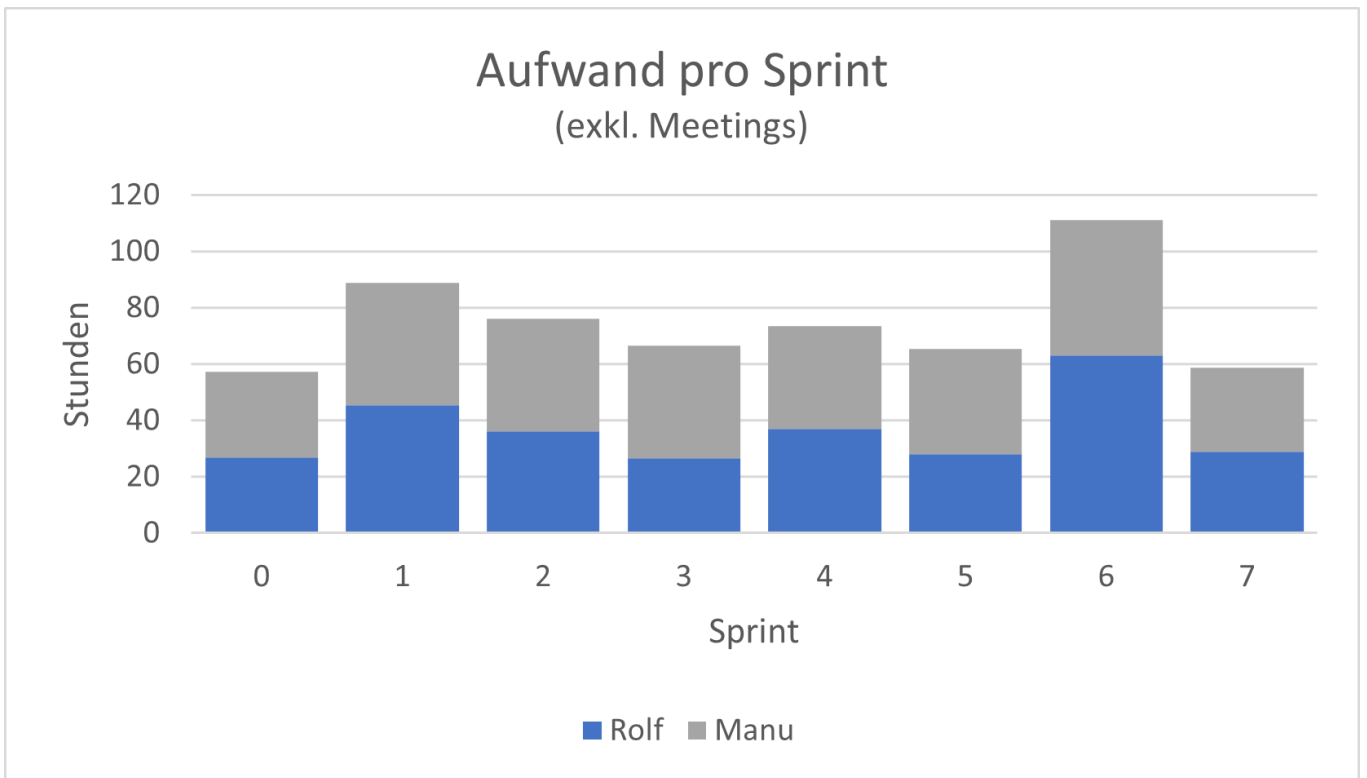


Abbildung 8.10: Aufwand pro Sprint

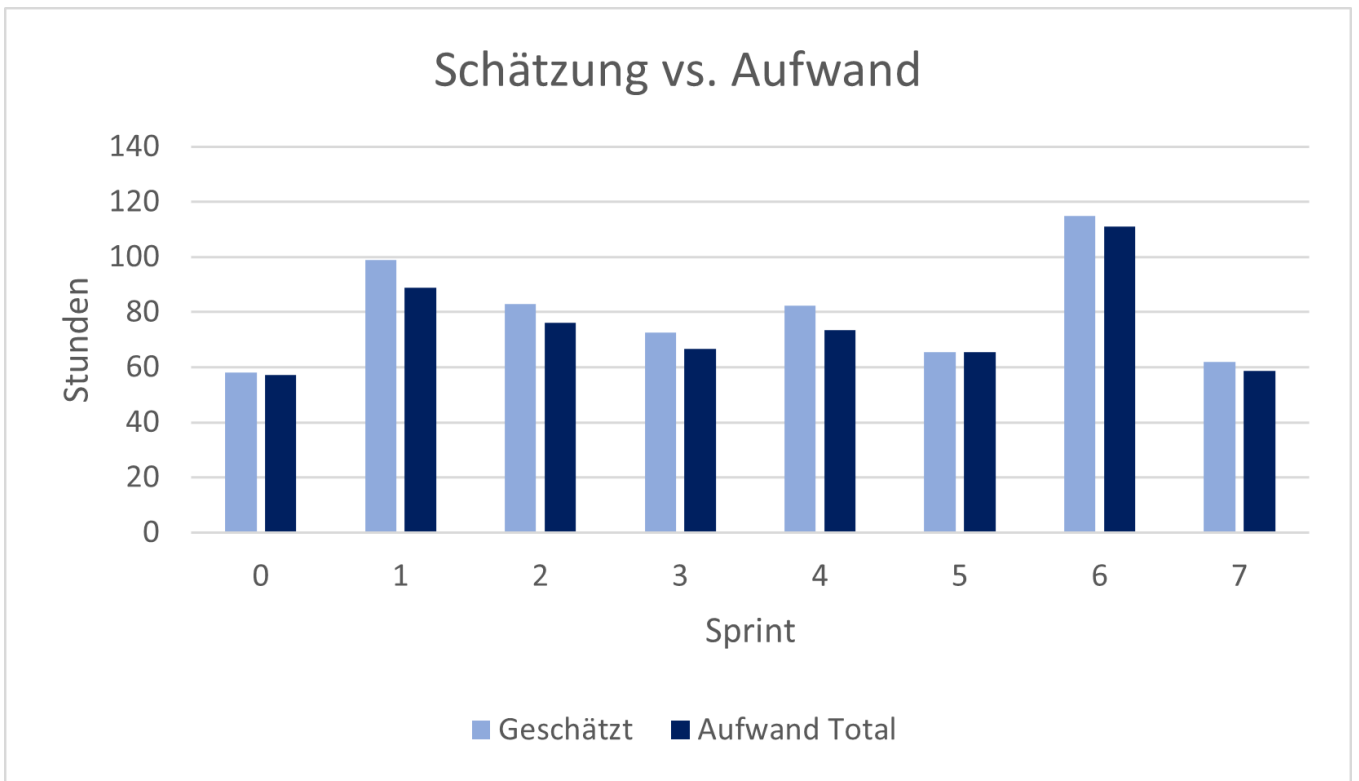


Abbildung 8.11: Geschätzter vs. effektiver Aufwand pro Sprint

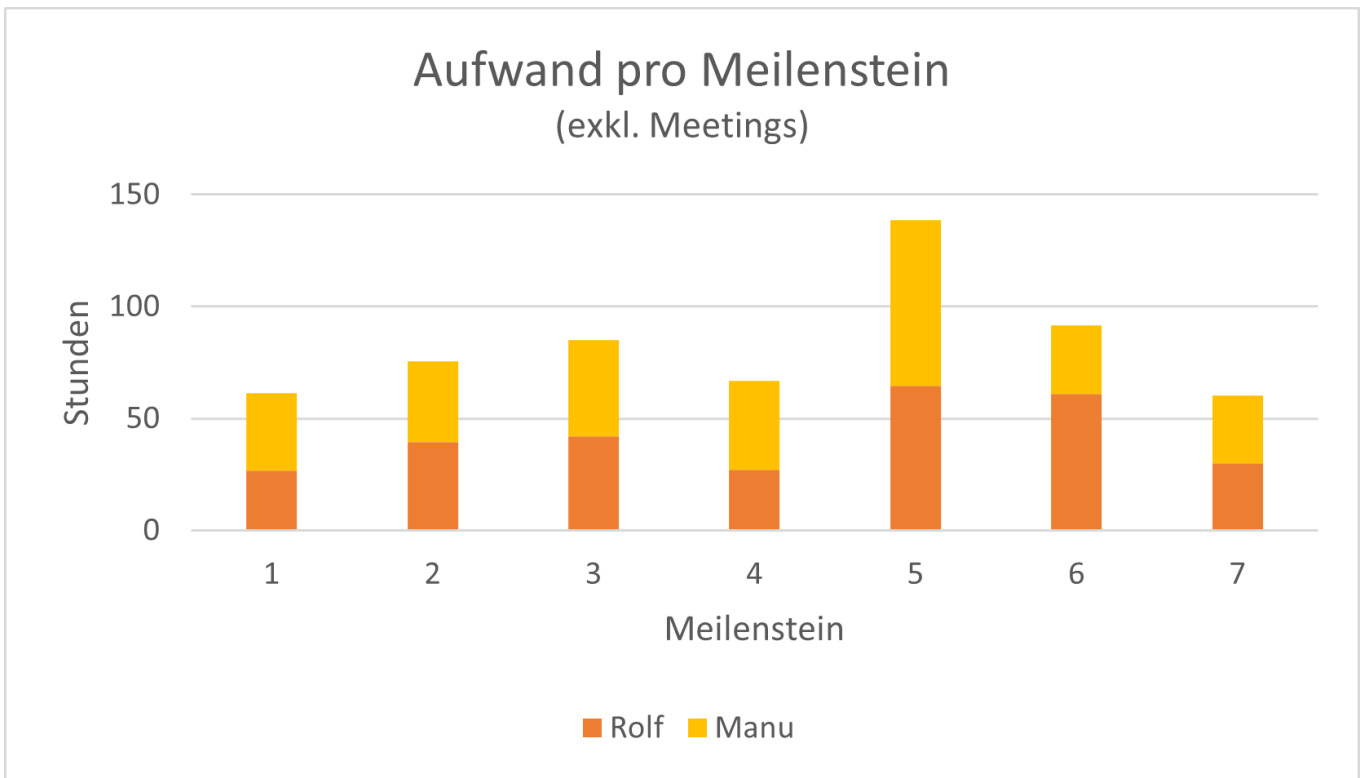


Abbildung 8.12: Aufwand pro Meilenstein

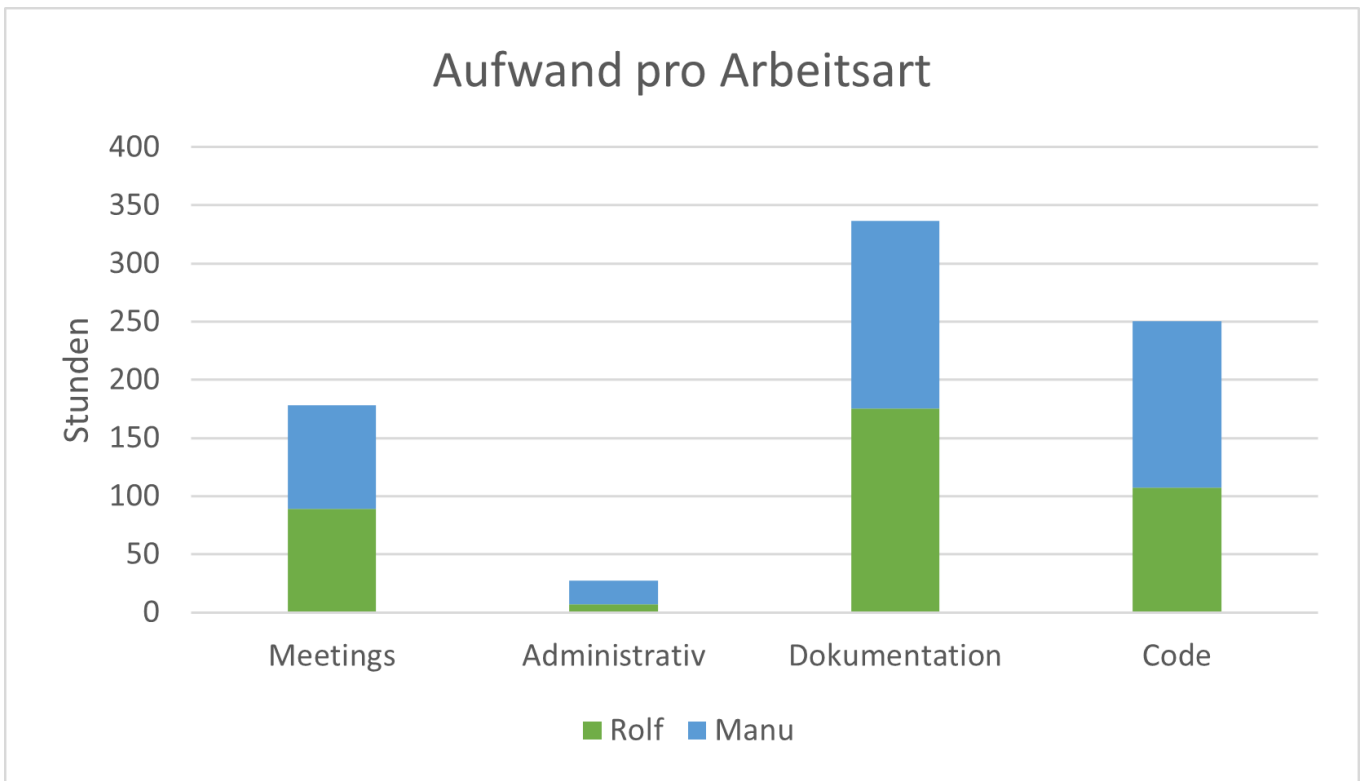


Abbildung 8.13: Aufwand pro Arbeitsart

Kapitel 9

Fazit

9.1 Retrospektive

Produkt Die entwickelte Lösung ist ein Framework zur Integration einer clientseitigen On-Site-Search in eine statische Website. Die Lösung setzt eine semantisch korrekte Website voraus und basiert auf Best Practices von SEO und Semantic HTML. Das Framework kann ohne grossen Aufwand in eine bestehende Website integriert werden und ist stark konfigurierbar. Jeder Baustein innerhalb der Lösung kann auf den Use Case des Entwicklers angepasst und optimiert werden. Die Lösung ist für Websites mit relativ kleinem, statischen Inhalt optimiert.

Anforderungen Im Projekt konnten alle Use Cases, welche das Minimum Viable Product (MVP) beinhalten sollte, umgesetzt werden. Zudem konnten weitere Projekt-Features erfolgreich umgesetzt werden. Spezifizierte jedoch nicht umgesetzte Use Cases wurden dokumentiert und an den Auftraggeber übergeben oder als Ausbaumöglichkeiten festgehalten. Die Verifikation der NFR's haben gezeigt, dass diese alle erfolgreich erfüllt wurden.

Verifikation Zur Bestätigung der funktionalen Korrektheit der Lösung wurden automatisierte Tests durchgeführt. Zudem haben System- und Usability-Tests den Umgang mit dem Framework sowie die Intuitivität und die Qualität der Suche bestätigt. Das ausgearbeitete Integrationsbeispiel für den Auftraggeber zeigt zudem die Einfachheit der Verwendung des entwickelten Search-Integrator-Frameworks.

Usability Tests Die Szenarien der Usability Tests wurden auf Basis der erstellten Proto-Personas und Proto-Journeys erstellt. Die Durchführung der Tests hat gezeigt, dass diese korrekt erkannt wurden und bestätigten somit getätigte Annahmen. Zudem wurden Optimierungen erkannt, welche in die Lösung eingearbeitet, oder als Weiterentwicklungs-Empfehlung spezifiziert werden konnten.

Zusammenarbeit mit Auftraggeber Die Zusammenarbeit mit dem Auftraggeber war von einem offenen Austausch geprägt. Dies führte zu einer Lösung, welche die Bedürfnisse und Anforderungen des Auftraggebers erfüllt. Dank des offenen Austausches konnten Missverständnisse vermieden oder

zeitnah aufgeklärt werden. Entscheidungen konnten getroffen und Optimierungen umgehend erkannt und umgesetzt werden. Dadurch wurde das Risiko einer konzeptionell falschen Lösung minimiert und die Qualität erhöht.

Planung Zu Beginn des Projektes wurde viel Zeit in die Planung und Organisation des Projektes, wie schon in vorgängigen Projekten, investiert. Dies hat sich einmal mehr auszahlt. Alle Termine und Meilensteine konnten eingehalten werden und es gab nie zeitlichen Druck.

Zeitschätzung Die Zeitschätzungen für den Aufwand pro Issue konnte das Projektteam relativ gut einschätzen. Bereits von Beginn an konnten präzise Schätzungen gemacht werden. Im Verlauf des Projektes gab es dennoch einige Aufgaben, bei welchen man sich aufgrund fehlender Erfahrung, verschätzte.

Team Das Projektteam arbeitete sehr gut zusammen. Die Kommunikation innerhalb des Teams war stets offen, direkt und fair. Die Aufgaben wurden nach Interesse und Fähigkeiten der Teammitglieder verteilt. Jeder konnte seine Stärken einbringen und seine Interessen vertreten. Jeglicher Code sowie die gesamte Dokumentation wurde von beiden Projektteammitgliedern gesichtet und abgesegnet.

9.2 Ausblick

Nach der Übergabe des Projekts an den Auftraggeber wird die Suche in die produktive SDX-Website integriert werden. Die dazu benötigten Informationen wurden anhand eines Auftraggeber-spezifischen Anwendungsbeispiels sowie einer Entwicklerdokumentation festgehalten und an den Auftraggeber übergeben. Auf Wunsch des Auftraggebers wird die Integration des Frameworks nach Abschluss der Bachelorarbeit durch den Auftraggeber selbst durchgeführt. Das Projektteam steht dabei für allfällige Fragen und Problemstellungen zur Verfügung.

Potenzielle Ausbaumöglichkeiten für das Framework wurden in Form von Projekt- und Produktfeatures dokumentiert und können in Zukunft umgesetzt werden. Diese Features wurden während des Projekts erkannt und spezifiziert, konnten jedoch aus zeitlichen Gründen und Architektur-Constraints nicht umgesetzt werden. Nachfolgend eine Liste von möglichen Erweiterungen:

- Kategorisierung der Suchresultate als Konfigurationsmöglichkeit anbieten
- Suchfilter als Konfigurationsmöglichkeit anbieten
- Suchhistorie als Feature anbieten
- Rekursive Rekonstruktion des Input-Verzeichnisses beim Output des Frameworks
- Logging (serverseitig) anbinden für das Führen von Statistiken und Bereitstellen von Auswertungen
- Anbindung eines Webcrawler zur Vereinfachung der Datenerhebung

Das Projektteam sieht grosses Potenzial in der entwickelten Lösung. Sie ermöglicht eine einfache und schnelle Integration einer Suche in statische Websites. Dadurch kann ein hervorragendes Sucherlebnis

bereitgestellt werden und der wachsenden Anforderung, eine Such-Funktionalität auf der eigenen Website anzubieten, gerecht werden. Es ist eine Möglichkeit für Websites die Nutzerfreundlichkeit und Zugänglichkeit zu den darin enthaltenen Informationen erheblich zu erhöhen und somit das Benutzererlebnis der Website zu verbessern. Des Weiteren ist der Aufwand für die Integration einer Suche durch die Anwendung des Frameworks gering, was die Nutzung der Lösung attraktiv gestaltet.

Das Projektteam ist stolz in Zusammenarbeit mit dem Auftraggeber ein generisch anwendbares Framework entwickelt zu haben, das für SDX live gehen wird. Die Projektmitglieder freuen sich darauf das Search-Integration-Framework somit in Zukunft in Aktion zu sehen. Durch die geleistete Arbeit kann die Art und Weise, wie eine Suche in Websites integriert wird, beeinflusst, und Unternehmen sowie private Personen dabei unterstützt werden, ihre Websites zu optimieren und erfolgreich zu sein.

Trotz des Potenzials hat sich das Projektteam aus persönlichen und zeitlichen Gründen dazu entschieden, das Framework nicht weiterzuentwickeln. Nichtsdestotrotz steht das Projektteam dem Auftraggeber für die Integration des Frameworks in die produktive Umgebung nach Abschluss des Projektes und ausserhalb des Rahmens der Bachelorarbeit zur Verfügung. Dem Auftraggeber steht es frei, die entwickelte Lösung weiterzuentwickeln und auszubauen.

Literaturverzeichnis

- [alga] *Algolia*. <https://www.algolia.com/>
- [algb] *Algolia Performance*. <https://support.algolia.com/hc/en-us/articles/4406975267089-How-fast-is-Algolia->
- [ang] *Angular*. <https://angular.io/>
- [apa] *Apache-Lizenz - Wikipedia*. <https://de.wikipedia.org/wiki/Apache-Lizenz>
- [bre] *BreadcrumbList - Schema.org Type*. <https://schema.org/BreadcrumbList#:~:text=A%20BreadcrumbList%20is%20an%20ItemList,the%20items%20in%20a%20BreadcrumbList.>
- [buna] *Bundlephobia - Bundle Size Algolia*. <https://bundlephobia.com/package/algoliasearch@4.17.0>
- [bunb] *Bundlephobia - Bundle Size Fuse.js*. <https://bundlephobia.com/package/fuse.js@6.6.2>
- [bunc] *Bundlephobia - Bundle Size Lunr.js*. <https://bundlephobia.com/package/lunr@2.3.9>
- [c8] *c8*. <https://github.com/bcoe/c8>
- [cha] *Chai*. <https://www.chaijs.com/>
- [cli] *Enabling Client-Side Search - Manning*. <https://freecontent.manning.com/enabling-client-side-search/>
- [con] *Wikipedia - Convention over Configuration*. https://en.wikipedia.org/wiki/Convention_over_configuration
- [Cyp] *Cypress*. <https://www.cypress.io/>
- [dcl] *DOMContentLoaded*. https://developer.mozilla.org/en-US/docs/Web/API/Window/DOMContentLoaded_event
- [dis] *Distributed search engine*. https://en.wikipedia.org/wiki/Distributed_search_engine
- [doo] *Doofinder*. <https://www.doofinder.com/de/>
- [ela] *Elasticsearch*. <https://www.elastic.co/>

- [esl] *ESLint*. <https://eslint.org/>
- [fcp] *First Contentful Paint (FCP)*. <https://web.dev/fcp/>
- [fed] *Federated search*. https://en.wikipedia.org/wiki/Federated_search
- [fus] *Fuse.js*. <https://fusejs.io/>
- [gita] *The DevSecOps Platform — GitLab*. <https://about.gitlab.com/>
- [gitb] *GitHub: Let's build from here*. <https://github.com/>
- [gooa] *12 Features Of A Great Site Search Function — SilverDisc Digital Marketing Agency*. <https://www.silverdisc.co.uk/blog/2016/02/19/12-features-great-site-search-function>
- [goob] *Google Search Central*. <https://developers.google.com/search/docs?hl=de>
- [gooc] *On-Site Search Best Practices For SEO and User Experience*. <https://www.searchenginejournal.com/on-site-search-seo/309189/>
- [good] *On Site Search Top Tips and Best Practices - Inbenta*. <https://www.inbenta.com/en/blog/site-search-best-practices/>
- [gps] *Google Programmable Search Engine*. <https://programmablesearchengine.google.com/about/>
- [gtt] *GitLab Time Tracker*. <https://github.com/krisbxb/gitlab-time-tracker/blob/master/documentation.md#iii-reports>
- [inb] *Inbenta*. <https://www.inbenta.com/>
- [iso] *ISO - ISO/IEC 25010:2011 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. <https://www.iso.org/standard/35733.html>
- [jesa] *Jest*. <https://jestjs.io/>
- [jesb] *Jest - ECMAScript Modules*. <https://jestjs.io/docs/ecmascript-modules>
- [lcp] *Largest Contentful Paint (LCP)*. <https://web.dev/lcp/>
- [lig] *Lighthouse*. <https://developer.chrome.com/docs/lighthouse/overview/>
- [lun] *Lunr.js*. <https://lunrjs.com/docs/index.html>
- [Mar09] MARTIN, Robert C.: *Clean Code: A Handbook of Agile Software Craftsmanship*. Pearson, 2009. – ISBN 978-0-13-235088-4
- [mit] *MIT-Lizenz - Wikipedia*. <https://de.wikipedia.org/wiki/MIT-Lizenz>
- [moca] *Mocha*. <https://mochajs.org/>

- [mochb] *Mocha - ESM Support*. <https://mochajs.org/#nodejs-native-esm-support>
- [npma] *npm*. <https://www.npmjs.com/>
- [npmb] *NPM trends: Downloads of web bundlers*. <https://nptrends.com/browserify-vs-esbuild-vs-parcel-vs-rollup-vs-snowpack-vs-swc-vs-vite-vs-webpack>
- [npmc] *uuid*. <https://www.npmjs.com/package/uuid>
- [ope] *Open UI*. <https://open-ui.org/>
- [OWA] *OWASP Top Ten*. <https://owasp.org/www-project-top-ten/>
- [pre] *Prettier*. <https://prettier.io/>
- [proa] *Proto-Journey*. <https://www.collaborative-uxdesign.com/methods/proto-journey>
- [prob] *Proto-Persona*. <https://www.collaborative-uxdesign.com/methods/proto-personas>
- [reaa] *5 Reasons Why Your Website Needs a Search Bar*. <https://www.greenmellenmedia.com/why-your-website-needs-a-search-function/>
- [reab] *Four reasons why site search is vital for online retailers*. <https://econsultancy.com/four-reasons-why-site-search-is-vital-for-online-retailers/>
- [reac] *Why Implement Search Functionality for Your Websites — LoginRadius Blog*. <https://www.loginradius.com/blog/engineering/importance-of-search-functionality-for-websites/>
- [ref] *Refactoring Guru - Facade*. <https://refactoring.guru/design-patterns/facade>
- [rfc] *A Universally Unique Identifier (UUID) URN Namespace*. <https://datatracker.ietf.org/doc/html/rfc4122>
- [RUP] *Rational Unified Process*. https://de.wikipedia.org/wiki/Rational_Unified_Process
- [Scr] *Scrum*. <https://de.wikipedia.org/wiki/Scrum>
- [sdx] *Swisscom Digital Experience*. <https://sdx.swisscom.ch/>
- [seaa] *Introduction to Searching Algorithms*. <https://www.tutorialspoint.com/introduction-to-searching-algorithms>
- [seab] *Searching Algorithms - GeeksforGeeks*. <https://www.geeksforgeeks.org/searching-algorithms/>
- [sem] *W3Schools - HTML Semantic Elements*. https://www.w3schools.com/html/html5_semantic_elements.asp
- [seo] *Search Engine Optimization*. <https://de.wikipedia.org/wiki/Suchmaschinenoptimierung>

- [sol] *Solr*. <https://solr.apache.org/>
- [sta] *State of JavaScript 2022: Testing*. <https://2022.stateofjs.com/en-US/libraries/testing/>
- [sto] *Storybook*. <https://storybook.js.org/>
- [the] *Thesaurus API - API Ninjas*. <https://api-ninjas.com/api/thesaurus>
- [tim] *Time-to-interactive*. https://developer.mozilla.org/en-US/docs/Glossary/Time_to_interactive
- [ts-] *TypeScript - SourceMaps*. <https://www.typescriptlang.org/tsconfig#sourceMap>
- [tti] *Time to Interactive (TTI)*. <https://web.dev/tti/>
- [uc] *Craig Larman. Applying Evolutionary Use Cases*. https://www.craiglarman.com/wiki/downloads/applying_uml/larman-ch6-applying-evolutionary-use-cases.pdf
- [vue] *Vue.js*. <https://vuejs.org/>
- [weba] *web.dev*. <https://web.dev/>
- [webb] *Webpack*. <https://webpack.js.org/>
- [yt-] *Youtube Search*. <https://www.youtube.com/howyoutubeworks/product-features/search/>

Glossar

BA Bachelorarbeit. 121

CD Continuous Deployment. 7, 98, 102

CI Continuous Integration. 7, 98, 102

Design System Sammlung von Komponenten, Prinzipien und Einschränkungen für die Designentwicklung. 14, 16

DoD Definition of Done. 100

gtt GitLab Time Tracker. 129

IDE Integrated Development Environment. 97

JSON Java Script Object Notation.. 70, 72, 74

MVP Minimal Viable Product. 37

NFR Nicht-funktionale Anforderungen. 53, 61, 63, 64, 66, 138–141

NPM Paket-Manager für JavaScript. 3, 6, 72, 79, 82, 131

OST OST - Ostschweizer Fachhochschule. 128, 129

OWASP Open Worldwide Application Security Project. 53, 63

RUP Rational Unified Process. 115, 116

SDX Swisscom Digital Experience. 2–6, 8, 9, 11, 12, 14–18, 21, 22, 25, 40–43, 47, 49, 53–55, 57, 61, 63, 67, 72, 78, 79, 90, 94, 98, 104–111, 113, 131, 132, 136–139, 147, 148, 153, 155, 161, 163, 164, 166

Semantic HTML Semantisches HTML ist die Verwendung von HTML-Markup, um die Semantik oder Bedeutung der Informationen auf Webseiten und Webanwendungen zu verstärken, anstatt nur ihre Präsentation oder ihr Aussehen zu definieren.. 70, 72, 146

SEO Search Engine Optimization. 34, 70, 72, 146

UI User Interface. 38, 79

UX User Experience. 2, 4, 11, 14, 161

Webseite Eine einzige Seite im Web (eine HTML-Datei). 2, 5–7, 11, 12, 14, 21, 22, 25, 34, 40, 41, 44–47, 54–56, 58, 64, 66, 69–74, 77, 80, 90, 93, 98, 106, 107, 109, 112, 131–134, 137, 138, 141, 159, 162–167

Website Die gesamte Sammlung von Webseiten einer Domain. 2–8, 11, 12, 14, 16, 22, 23, 25, 26, 29, 31–35, 43, 47, 54, 55, 59, 63, 64, 66, 68–73, 79, 80, 88, 90, 94, 98, 104, 106–113, 131, 132, 134, 137–139, 146–148, 159–162, 164–167

Abbildungsverzeichnis

1	Landing-Page der SDX-Website inkl. Menü	4
2	Projektplan	5
3	Konzept	6
4	QR-Code zum NPM-Paket 'Search-Integrator'	7
2.1	Domänenmodell SDX	15
4.1	Use Case Diagramm	39
4.2	Kategorien inkl. Unterkategorien gemäss ISO25010-Standard	53
5.1	Konzept zur Integration der Suche	69
5.2	Konzept mit Suchkomponenten-Integration & ohne Bundling der JSON-Dateien	73
5.3	Konzept mit Suchkomponenten-Integration & mit Bundling der JSON-Dateien	73
5.4	Konzept ohne Suchkomponenten-Integration & ohne Bundling der JSON-Dateien	74
5.5	Konzept ohne Suchkomponenten-Integration & mit Bundling der JSON-Dateien	75
5.6	Klassische Client-Server Architektur	76
5.7	Clientseitige Architektur im allgemeinen und spezifisch dargestellt	77
5.8	Nutzwertanalyse Produkte	87
5.9	Suchresultate als Dropdown	90
5.10	Suchresultate als Dropdown - keine Suchresultate	90
5.11	Kategorisierte Suchresultate	91
5.12	Suchfilter	91
5.13	Suchhistorie	92
5.14	Suchvorschläge	92
5.15	Hover-Informationen	93
5.16	Mega-Dropdown	93
5.17	Komponenten-Übersicht	94
5.18	Komplette Komponenten-Übersicht bei leerem Suchfeld	95
5.19	Komponenten-Übersicht aller Komponenten - Ausgrauen	95
5.20	Komponenten-Übersicht aller Komponenten - Ausblenden	95
6.1	Pipeline Erweiterung vor Deployment	99

6.2	Workflow für die Arbeit am Source-Code	101
6.3	Übersicht Code-Coverage	103
7.1	Projektplan	119
7.2	Risikomatrix	124
7.3	Risikoentwicklungsgraph	128
8.1	Ausgelieferte Ressourcen	131
8.2	Lighthouse Navigation Report - Navigation auf der Website ohne integrierter Suche	132
8.3	Lighthouse Navigation Report - Navigation auf der Website mit integrierter Suche	132
8.4	Lighthouse Timespan Report - Navigation auf der Website ohne integrierter Suche	133
8.5	Lighthouse Timespan Report - Navigation auf der Website mit integrierter Suche	134
8.6	Lighthouse Snapshot Report - Darstellung Suche und Suchresultate	134
8.7	Beispiel einer Suchanfrage	136
8.8	Aufwand Total	142
8.9	Burndown-Chart	143
8.10	Aufwand pro Sprint	143
8.11	Geschätzter vs. effektiver Aufwand pro Sprint	144
8.12	Aufwand pro Meilenstein	144
8.13	Aufwand pro Arbeitsart	145

Tabellenverzeichnis

2.1	Benutzergruppen und deren Bedürfnisse in SDX	17
2.2	Benutzergruppen und ihre Proto-Journeys	18
4.1	Use Cases - Farbcode	37
4.2	Beschreibung Akteure	38
4.3	Beschreibung Use Case 01: Projektintegration	40
4.4	Beschreibung Use Case 02: Synonyme	41
4.5	Beschreibung Use Case 03: Autocomplete	42
4.6	Beschreibung Use Case 04: Keine Suchresultate	43
4.7	Beschreibung Use Case 05: Konfiguration Suche	44
4.8	Beschreibung Use Case 06: Kategorisierte Suchresultate	45
4.9	Beschreibung Use Case 07: Zusätzliche Informationen zu Suchresultaten	46
4.10	Beschreibung Use Case 08: Suchfilter	47
4.11	Beschreibung Use Case 09: Suchhistorie	48
4.12	Beschreibung Use Case 10: Visuelle Unterstützung bei Komponenten-Suche	49
4.13	Beschreibung Use Case 11: Verzeichnisstruktur des Outputs	50
4.14	Beschreibung Use Case 12: Logging und Auswertung	51
4.15	Beschreibung Use Case 13: Anbindung Webcrawler	52
4.16	Beschreibung NFR01: Operability (Usability)	54
4.17	Beschreibung NFR02: Time Behavior (Performance Efficiency)	54
4.18	Beschreibung NFR03: Co-Existence (Compatibility)	55
4.19	Beschreibung NFR04: Time Behavior & Resource Utilization (Performance Efficiency)	55
4.20	Beschreibung NFR05: Replaceability (Portability)	56
4.21	Beschreibung NFR06: User Error Protection (Usability)	56
4.22	Beschreibung NFR07: Adaptability (Portability)	57
4.23	Beschreibung NFR08: User Interface Aesthetics (Usability)	57
4.24	Beschreibung NFR09: Testability (Maintainability)	58
4.25	Beschreibung NFR10: Confidentiality (Security)	58
4.26	Beschreibung NFR11: Modifiability & Modularity (Maintainability)	59
4.27	Beschreibung NFR12: Accessibility (Usability)	59
4.28	Beschreibung NFR13: Learnability (Usability)	60

4.29	Auslistung aller NFR Kategorien gemäss ISO25010	62
4.30	Auffistung aller OWASP Top 10 Risiken	63
4.31	Stand der NFR per Alpha-Release	65
4.32	Stand der NFR per Beta-Release	67
5.1	Design-Entscheide	79
5.2	Technische Entscheide	82
5.3	Irrelevante Produkte	84
5.4	Relevante Produkte	85
5.5	Evaluationskriterien	87
5.6	Erläuterung Nutzwertanalyse	89
6.1	Allgemeine Usability-Tests	107
6.2	UX/UI spezifische Usability-Tests	108
6.3	Entwickler spezifische Usability-Tests	110
6.4	Management spezifische Usability-Tests	111
6.5	System-Tests	113
7.1	Liste der Meilensteine	118
7.2	Team-interne Meetings	120
7.3	Grössere Reviews mit Betreuer oder Assistent	120
7.4	Termine BA	121
7.5	Reviews Swisscom	121
7.6	Rollenverteilung	122
7.7	Ansprechpersonen Swisscom	123
7.8	Risiken	126
7.9	Massnahmen zur Risikominimierung und -prävention	127
10.1	Historie System-Tests - Durchführung Alpha	161
10.2	Historie System-Tests - Durchführung Beta	162
10.3	Historie Usability Test Testperson 1	164
10.4	Historie Usability Test Testperson 2	166
10.5	Historie Usability Test Testperson 3	167
10.6	Historie Usability Test Testperson 4	168

Teil VI

Anhang

Kapitel 10

Anhang

10.1 Test-Protokolle

10.1.1 System-Tests

Durchführung Alpha - Mittwoch, 17.05.2023

Wer	Ref	Effektives Resultat	Abgeleitete Massnahmen
ROB	1	<ol style="list-style-type: none"> 1. Der Build der Website sowie die Integration der Suche haben keine Fehlermeldung erzeugt. Es wurden alle benötigten Files erstellt (Records, SearchIndex, angereicherte Webseiten). 2. Der Webserver startet ohne Probleme. 3. Das Suchfeld wird beim Öffnen der Website direkt fokussiert. 4. Der Suchindex enthält die neuen Records. Die Suche nach diesen Einträgen ist erfolgreich. 	<ol style="list-style-type: none"> 1. Die Nachvollziehbarkeit vom Integrator ist aktuell nicht gegeben, da kein Logging der einzelnen Schritte vorhanden ist. Es wird ein Logger implementiert, welcher die Nachvollziehbarkeit erhöhen soll, indem die einzelnen Komponenten des Frameworks loggen, sobald diese erfolgreich durchgelaufen sind. 2. Bei den Usability-Tests soll darauf geachtet werden, ob der Auto-Fokus erwünscht ist oder nicht. Je nach Reaktion würde dieses Feature wieder ausgebaut.
ROB	2	<ol style="list-style-type: none"> 1. Die Website kann auf den Browsern Chrome, Edge und Firefox ohne Problem genutzt werden. 2. Das Suchfeld wird beim Öffnen der Website direkt fokussiert. 3. Die Menge der Suchergebnisse auf verschiedenen Gerätetypen (Desktop, Mobile) ist korrekt. 4. Die Darstellung der Suchergebnisse wird korrekt, nach dem vergebenen Design, dargestellt. 5. Suchergebnisse können zur Navigation genutzt werden. 	<ol style="list-style-type: none"> 1. Keine Massnahmen definiert.

Tabelle 10.1: Historie System-Tests - Durchführung Alpha

Durchführung Beta - Freitag, 02.06.2023

Wer	Ref	Effektives Resultat	Abgeleitete Massnahmen
ROB	1	<ol style="list-style-type: none"> 1. Der Build der Website sowie die Integration der Suche haben keine Fehlermeldung erzeugt. Es wurden alle benötigten Files erstellt (Records, SearchIndex, angereicherte Webseiten). Zudem ist die Nachvollziehbarkeit der einzelnen Schritte gegeben. 2. Der Webserver startet ohne Probleme. 3. Der Suchindex enthält die neuen Records. 4. Die Suche nach diesen Einträgen ist erfolgreich. 	<ol style="list-style-type: none"> 1. Keine Massnahmen definiert.
ROB	2	<ol style="list-style-type: none"> 1. Die Website kann auf den Browsern Chrome, Edge und Firefox ohne Problem genutzt werden. 2. Die Menge der Suchergebnisse auf verschiedenen Gerätetypen (Desktop, Mobile) ist korrekt. 3. Die Darstellung der Suchergebnisse wird korrekt, nach dem vergebenen Design, dargestellt. 4. Suchergebnisse können zur Navigation genutzt werden. 	<ol style="list-style-type: none"> 1. Keine Massnahmen definiert.

Tabelle 10.2: Historie System-Tests - Durchführung Beta

10.1.2 Usability-Tests

Die Usability-Tests wurden am Donnerstag, 25.05.2023, in Zürich durchgeführt.

Zu Beginn der Usability-Tests wurde den Testenden erklärt, wie der Ablauf der Tests ist, worauf geachtet werden soll und was getestet wird. Nach der kurzen Einführung wurden vier Personen getestet. Dabei wurde mit Audio- und Bildschirmaufnahmen gearbeitet. Zum Abschluss wurde eine Diskussionsrunde eröffnet, um offene Fragen zu klären und zusätzliche Inputs abzuholen.

Testperson 1 Testperson 1 ist zwischen 35 und 40 Jahre alt, ist ausgebildeter UX-Designer und hat die Rolle als UX-Lead beim Auftraggeber inne. SDX wird von Testperson 1 täglich genutzt. Die Aufgaben sind die Pflege und das Lesen der Webseite.

Ref.	Effektives Resultat	Abgeleitete Massnahmen
A1	Es wurde zuerst über die Navigation gearbeitet und anschliessend mit der Suche. Die gesuchte Seite konnte ohne Probleme gefunden werden, über beide Varianten.	-
A2	Die gesuchte Seite wurde gefunden. Der Testperson ist klar, wo die Informationen liegen. Die gesuchte Seite konnte ohne Probleme gefunden werden.	-
A3	Es wurde automatisch auf andere Stichworte ausgewichen. Dem Testenden war klar das diese Informationen nicht vorhanden sind.	-
D1	Information ohne Probleme gefunden (Icon Alignment).	-
D2	Die gesuchten Charts wurden sofort gefunden.	-
D3	Header und Header (Notification) gefunden.	-
E1	Vokabular nicht bekannt - eine Suche mit dem Stichwort 'Spacing' war nicht erfolgreich.	Empfehlungen an Swisscom erweitert: Ausbau der Synonymlisten und/oder Indexieren von zusätzlichen Daten.
E2	Information ohne Probleme gefunden. Es wird erwartet, das man direkt im richtigen Tab landet.	Empfehlung an Swisscom bezüglich der Verwendung von Sections und Articles definiert (Umbau Website).
E3	Information ohne Probleme gefunden, es wurde mit 'select' und mit 'dropdown' gesucht.	-
E4	Information ohne Probleme gefunden.	-

AF1	<ul style="list-style-type: none"> • Ein bisschen trocken • Unterkategorien von Webseiten erscheinen nicht • Falls Unterkategorien gefunden werden, muss klar sein, bei welchem Suchresultat es sich um eine Webseite und bei welchem es sich um eine Unterkategorie handelt (Nachvollziehen des Inhaltes) • Visuelle Unterstützung wäre bei der Suche nach Komponenten hilfreich 	Empfehlung an Swisscom bezüglich der Verwendung von Sections und Articles definiert (Umbau Website).
AF2	Es wird erwartet, dass der gesamte Inhalt durchsucht wird.	Information an Swisscom definiert, wie das Framework konfiguriert werden kann, sodass mehr oder sogar der gesamte Inhalt der Website indexiert würde.
AF3	<ul style="list-style-type: none"> • Sektionen und das Linken auf diese • Visuelle Unterstützung 	Empfehlung an Swisscom bezüglich der Verwendung von Sections und Articles definiert (Umbau Website).
AF4	<p>Ja, weil</p> <ul style="list-style-type: none"> • schneller navigiert werden kann. • klarer und zielorientierter navigiert werden kann. • schnell überprüft werden kann, ob etwas im Design System vorhanden ist oder nicht. 	-
AF5	Ja, absolut.	-

Tabelle 10.3: Historie Usability Test Testperson 1

Allgemeine Kommentare zur Durchführung mit Testperson 1

- Suche wurde nie mit Enter oder dem Lupen-Icon bestätigt. Es wird nicht erwartet, dass dadurch noch weitere Suchresultate angezeigt werden.

Testperson 2 Testperson 2 ist zwischen 50 und 60 Jahre alt, ist gelernter Elektroingenieur und hat die Rolle 'Frontend-Entwickler' beim Auftraggeber. SDX wird von Testperson 2 täglich genutzt. Die Aufgaben sind die Ergänzung und Verbesserung von Komponenten und der Darstellung.

Ref.	Effektives Resultat	Abgeleitete Massnahmen
A1	Zuerst wurde die Navigation überflogen, jedoch nichts gefunden. Anschliessend wurde mit der Suche gearbeitet. Die gesuchten Informationen wurden direkt, nach dem Eintippen von vier Buchstaben, gefunden.	-
A2	Direkt in der Navigation gefunden. Zweiter Versuch über die Suche war ebenso erfolgreich.	-
A3	Nach drei Suchanfragen wurde gestoppt, da die entsprechende Meldung angezeigt wurde.	-
E1	Informationen ohne Probleme gefunden.	-
E2	Information ohne Probleme gefunden.	-
E3	Information ohne Probleme gefunden. Es wurde mit 'select' und mit 'dropdown' gesucht.	-
E4	Information ohne Probleme gefunden.	-
AF1		Gut. Die Darstellung ist übersichtlich. -
AF2	Alles wie erwartet gefunden.	-
AF3	<ul style="list-style-type: none"> Nicht wirklich. Es ist irritierend, dass die Suchresultate erst nach dem 3. Buchstaben angezeigt werden. Man weiss nicht warum etwas gefunden wurde (Nachvollziehbarkeit). 	Empfehlung an Swisscom definiert, dass auf allen Webseiten eine BreadcrumbList platziert werden soll, damit diese vom Search-Integrator-Framework ausgelesen werden kann und somit für die Darstellung der Suchresultate zur Verfügung steht.

AF4	Ja, weil <ul style="list-style-type: none"> • die Navigation muss man gut kennen, um schnell zu sein. • Suche ist massiv viel schneller. 	-
AF5	Ja.	-

Tabelle 10.4: Historie Usability Test Testperson 2

Allgemeine Kommentare zur Durchführung mit Testperson 2

1. Suche wurde nie mit Enter oder dem Lupen-Zeichen bestätigt. Es wird nicht erwartet, dass dadurch noch weitere Suchresultate angezeigt werden.
2. Warum wird ein Suchresultat erst nach der Eingabe des 3. Zeichens angezeigt?
3. Grundsätzlich zufrieden, die Performance überraschte positiv.
4. Eine Navigation direkt auf einzelne Teile der Webseiten (Tabs) wäre von Vorteil.

Testperson 3 Testperson 3 ist zwischen 40 und 45 Jahre alt, ist gelernter Informatiker und hat noch eine Ausbildung als Polizist. Aktuell hat sie die Rolle als Frontend-Entwickler beim Auftraggeber. SDX wird von Testperson 3 täglich genutzt. Die Aufgaben sind die Erweiterung und Anpassung der SDX-Website sowie die Erarbeitung von Beispielprojekten.

Ref.	Effektives Resultat	Abgeleitete Massnahmen
A1	Es wurde bewusst die Suche verwendet. Keine Probleme beim Lösen des Szenarios.	-
A2	Keine Probleme beim Lösen des Szenarios. Anmerkung, da der Testperson die SDX-Website gut bekannt ist, hätte Sie die Navigation verwendet.	-
A3	Es wird schnell erkannt, dass keine Informationen vorhanden sind. Die Meldung 'No matches found' ist für die Testperson jedoch nicht 100% eindeutig. Es könnte auch einfach sein, dass er nicht die korrekten Wörter genutzt hat.	-

E1	Testperson setzt die On-Site-Suche für die Durchsuchung des Inhalts ein. Gesuchte Seite ohne Probleme gefunden.	-
E2	Information ohne Probleme gefunden. Ging jedoch über die Navigation.	-
E3	Information ohne Probleme gefunden. Hat Synonyme ausprobiert.	-
E4	Information ohne Probleme gefunden.	-
AF1	Es fehlt eine visuelle oder beschreibende Unterstützung zur Nachvollziehung des Inhalts einer Webseite und der Positionierung derer innerhalb der Website.	Empfehlung an Swisscom zur Anpassung der Website definiert, sodass Breadcrumbs auf allen Webseiten genutzt werden und dass die Webseiten mit Sections und Articles ergänzt werden.
AF2	Suchresultate sind grundsätzlich sinnvoll. Aktuell sind noch nicht viele Daten indexiert. Man findet bzw. landet jedoch schon immer an einem guten Startpunkt.	Empfehlung an Swisscom bezüglich dem Indexieren lassen von zusätzlichen Daten.
AF3	Wie schon gesagt, die Nachvollziehbarkeit wieso und wo etwas gefunden wurde und das Linken auf Tabs (Teile der Webseiten).	Empfehlung an Swisscom zur Anpassung der Website, sodass Breadcrumbs überall genutzt werden und dass die Webseiten mit Sections und Articles ergänzt werden.
AF4	Ja, weil man viel schneller ist und Synonyme verwenden kann.	-
AF5	Ja.	-

Tabelle 10.5: Historie Usability Test Testperson 3

Allgemeine Kommentare zur Durchführung mit Testperson 3

1. Das Autofokus-Feature wird bemerkt - es sei jedoch etwas aufdringlich.
2. Suche wurde nie mit Enter oder dem Lupen-Zeichen bestätigt. Es wird nicht erwartet, dass dadurch noch weitere Suchresultate angezeigt werden.

Testperson 4 Testperson 4 ist zwischen 50 und 60 Jahre alt, ist gelernte Pharmaassistentin und hat eine Leadership-Weiterbildung gemacht. Sie ist aktuell Leader des Frontend- sowie des Designteams beim Auftraggeber. SDX wird von Testperson 4 täglich genutzt. Die Aufgaben sind die Aufbereitung von Informationen und Komponenten für User von SDX.

Ref.	Effektives Resultat	Abgeleitete Massnahmen
A1	Es wurde bewusst die Suche verwendet. Keine Probleme beim Lösen des Szenarios.	-
A2	Navigiert über das Menu, da die gesuchte Webseite direkt ersichtlich war.	-
A3	Meldung wurde nicht beachtet. Nimmt nach 5 Suchanfragen an, dass keine Informationen vorhanden sind.	-
M1	Navigiert über das Menu, da die gesuchte Seite direkt ersichtlich war. Im zweiten Versuch wurde es auch noch mit der Suche umgesetzt.	-
M2	Keine Probleme beim Lösen des Szenarios.	-
D1	Information ohne Probleme gefunden.	-
D2	Information ohne Probleme gefunden. Hat zuerst mit dem Synonym 'Diagramm' gesucht, jedoch nichts gefunden.	Empfehlung an Swisscom zur Erweiterung der Synonyme.
D3	Keine Probleme beim Lösen des Szenarios.	-
AF1	Sehr positiv - es ist schlicht, einfach und übersichtlich. Zudem ist die 'Kategorie' der Webseite ersichtlich.	-
AF2	Mehr Synonyme wären angenehm gewesen. Ansonsten waren die Suchresultate immer sehr gut und aufschlussreich.	Empfehlung an Swisscom zur Erweiterung der Synonyme.
AF3	Nein, es hat nichts gefehlt.	-
AF4	Ja definitiv, man ist viel schneller. Die Suchresultate und die Performance der Suche war sehr gut!	-
AF5	Der Nutzen ist sehr gross, egal ob man die Website schon kennt oder nicht. Die Anzahl der Klicks wird immer kleiner sein als bei der Navigation.	-

Tabelle 10.6: Historie Usability Test Testperson 4

Allgemeine Kommentare zur Durchführung mit Testperson 4

1. Gestaltung/Design der Such-Komponente wurde als gut empfunden, schlicht und einfach.
2. Suche wurde nie mit Enter oder dem Lupen-Zeichen bestätigt. Es wird nicht erwartet, dass dadurch noch weitere Suchresultate angezeigt werden.

Positive Punkte

- Suche ist sehr schnell/performant.
- Darstellung zwar minimalistisch, aber gut.
- Nachverfolgung des Orts der Webseite in der Website über Breadcrumbs sehr gut!

Allgemeine Kommentare

- Tracking der Suchanfragen wäre sehr gut gewesen. Da jedoch keine serverseitige Komponente entwickelt werden sollte, wurde dies vernachlässigt. Swisscom ist es freigestellt dieses Feature zu ergänzen.
- Kontext/Nachvollziehbarkeit aktuell noch nicht gegeben. Aufbereitete Demoseiten sind jedoch sehr gut. Eventuell noch den Score anhand eines Ampelsystems in der Darstellung des Suchresultats ergänzen.
- Autofokus ist nicht zielführend.
- Platzierung der Suche sehr prominent gewählt. Dies war Absicht für die Usability-Tests, kann von Swisscom jedoch angepasst werden.
- Darstellung zwar minimalistisch aber gut. Hinweis: Die Darstellung ist unabhängig vom Search-Integrator-Framework und kann somit durch den Auftraggeber weiter optimiert werden.

Massnahmen durch das Projektteam

- Autofokus wird ausgebaut.
- Auf Desktopgeräten sollen max. 10 und nicht nur 5 Such-Resultate angezeigt werden.
- Suche wird nicht erst nach der Eingabe des dritten Buchstabens, sondern sobald die von der Such-Library definierte Mindestinformation vorhanden ist, ausgeführt. Dies stellt kein Problem dar, da die Suche clientseitig ausgeführt wird.

10.2 Weitere Anhänge

Usability Tests SDX-Search

Testdurchführung

Die Tests werden auf von uns zur Verfügung gestellten Geräten durchgeführt.

Bitte beachten Sie, dass während des Usability-Tests der Bildschirm sowie der Ton aufgezeichnet werden, um eine gründliche Analyse durchführen zu können. Die erhobenen Daten werden vertraulich behandelt und ausschliesslich für die Analyse verwendet und nach dem Abschluss der Arbeit vernichtet.

Wir bitten Sie während der Durchführung laut zu denken und Ihr Gedanken zu erläutern. Lesen Sie bitte jede Aufgabe laut vor, bevor Sie diese bearbeiten.

Wichtig: Bei Usability-Tests wird das System und nicht der Benutzer getestet. Sie können somit nichts Falsches machen.

Ablauf

Der Usability-Test gliedert sich in folgende Abschnitte:

1. Intro (5 Minuten):
 - Ziel darlegen
 - Umgebung klären
2. Usability Test I (20 Minuten)
 - Durchführung mit zwei Testpersonen
3. Usability Test II (20 Minuten)
 - Durchführung mit zwei Testpersonen
4. Feedback (15 Minuten)

Einverständniserklärung

Hiermit erkläre ich mich damit einverstanden, dass meine Teilnahme am Usability Test freiwillig erfolgt und dass meine Bildschirm- und Tonaufnahmen zu Analysezwecken aufgezeichnet werden dürfen.

Ort

Datum

Unterschrift

Grundlegende Informationen

Wir bitten Sie folgende Daten auszufüllen.

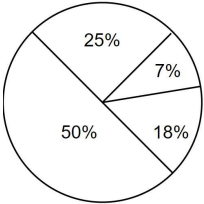
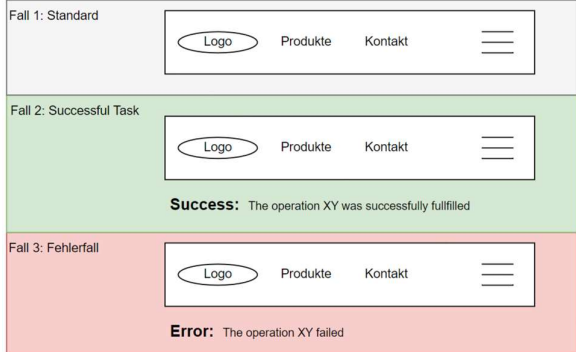
Alter	<input type="checkbox"/> 15 – 20 Jahre <input type="checkbox"/> 20 – 25 Jahre <input type="checkbox"/> 25 – 30 Jahre <input type="checkbox"/> 30 – 35 Jahre <input type="checkbox"/> 35 – 40 Jahre <input type="checkbox"/> 45 – 50 Jahre <input type="checkbox"/> 50 – 60 Jahre <input type="checkbox"/> 60+ Jahre
Gelernter Beruf	
Funktion/Rolle	
Wie oft nutzen Sie die SDX-Webseite?	<input type="checkbox"/> Immer (jeden Tag) <input type="checkbox"/> Oft <input type="checkbox"/> Sporadisch <input type="checkbox"/> Selten <input type="checkbox"/> Nie
Wozu nutzen Sie SDX-Webseite?	

Testfälle


Allgemein

Ref.	Beschreibung
A1	Sie möchten sich in das Thema Accessibility einlesen. Navigieren Sie zur gesuchten Webseite.
A2	Sie sind auf Ideensuche und möchten eine Übersicht über alle Icons. Navigieren Sie zur gesuchten Webseite.
A3	Sie wollen wissen, ob das Design System Informationen über Best Practices mit Bezug zu Projektmanagement enthält. Suchen Sie nach entsprechenden Stichwörtern.

Designer: in

Ref.	Beschreibung
D1	Sie möchten wissen, wie Icons relativ zueinander ausgerichtet werden sollten. Suchen Sie nach einer Empfehlung in SDX.
D2	Sie benötigen für die saubere Darstellung ein Diagramm. Die darzustellenden Daten sind Anteile eines Ganzen. Suchen Sie nach geeigneten Komponenten in SDX. 
D3	Sie arbeiten in einem Projekt und sind sich nicht mehr sicher, welche Optionen für einen Header gegeben sind. Im Projekt wurde die Anforderung gestellt, dass der Header die Navigation, das Logo sowie ein Dropdown enthält. Zudem sollen Meldungen ebenfalls im Headerbereich dargestellt werden, falls möglich. Da Sie mit SDX arbeiten, möchten sie wissen, welche Optionen bezüglich Header geboten werden. 

Developer: in

Ref.	Beschreibung
E1	<p>Der Arbeitskollege hat Ihnen gesagt, dass die Abstände zwischen zwei Komponenten auf der Webseite ihres Projektes nicht stimmen. Da Sie SDX in Ihrem Projekt verwenden, möchten Sie wissen, welche Abstände es gibt respektive welche Optionen Ihnen zur Verfügung stehen. Suchen Sie nach geeigneten Informationen.</p>  <p>The diagram illustrates two components, 'Komponente A' and 'Komponente B', represented as rounded rectangular boxes. A red double-headed arrow connects the two boxes, with the word 'Abstand' (Distance) written below it, indicating the relationship or gap between the components.</p>
E2	<p>Sie möchten wissen, wie ein Icon aus SDX genutzt werden kann. Suchen Sie nach geeigneten Informationen.</p>
E3	<p>Sie arbeiten an einem Projekt für einen Kunden und benötigen eine Möglichkeit, Auswahlfelder mit verschiedenen Optionen anzubieten. Suchen Sie nach geeigneten Möglichkeiten.</p>
E4	<p>Sie arbeiten in einem neuen Projekt und es wurde festgelegt, dass SDX für die Darstellung genutzt werden soll. Im Projekt verwenden Sie Angular. Nun möchten Sie SDX in Ihr Projekt integrieren und suchen nach den nötigen Informationen.</p>

SCRUM Master und Produkt Manager: in

Ref.	Beschreibung
M1	<p>Sie bereiten sich auf ein Meeting mit dem Design-Team vor und möchten wissen, welche Farben im Design System zur Verfügung stehen, um eine Farbauswahl zu diskutieren. Suchen Sie nach geeigneten Informationen.</p>
M2	<p>Sie arbeiten in einem Projekt und möchten den Aufwand für die Integration einer Notifications (modal) abschätzen. Navigieren Sie zu Notifications (modal).</p>

Allgemeine Fragen

Ref.	Beschreibung
AF1	<p>Wie gut fanden Sie die Darstellung der Suchresultate? Bitte begründen Sie ihre Aussage.</p>
AF2	<p>Wie gut fanden Sie die Suchresultate im Allgemeinen? Bitte begründen Sie ihre Aussage.</p>
AF3	<p>Hat Ihnen etwas konkretes gefehlt?</p>
AF4	<p>Denken Sie die Suche bringt ihnen einen Vorteil beim Arbeiten mit der SDX-Webseite? Bitte begründen Sie ihre Aussage.</p>
AF5	<p>Konnten Sie schneller auf der SDX-Webseite navigieren?</p>

Lighthouse Report ohne Suche: Navigation



Leistung



Barrierefreiheit



Best Practices



SEO



PWA



Leistung

Die Werte sind geschätzt und können variieren.
Die [Leistungsbewertung](#) wird direkt aus diesen
Messwerten berechnet. [Siehe Rechner](#).



▲ 0–49 ■ 50–89 ● 90–100

MESSWERTE

[Ansicht maximieren](#)

■ First Contentful Paint
1,0 s

● Total Blocking Time
90 ms

■ Speed Index
1,4 s

■ Largest Contentful Paint
1,4 s

● Cumulative Layout Shift
0

[Strukturkarte anzeigen](#)



Prüfungen anzeigen, die für folgende Messwerte relevant sind: [All](#) [FCP](#) [LCP](#) [TBT](#) [CLS](#)

EMPFEHLUNGEN

Empfehlung

Geschätzte Einsparung

Textkomprimierung aktivieren

0,72 Sek ^

Textbasierte Ressourcen sollten komprimiert (gzip, Deflate oder Brotli) ausgeliefert werden, um die Netzwerkbytes insgesamt zu minimieren. [Weitere Informationen zur Textkomprimierung.](#) FCP

LCP

URL	Übertragungsgröße	Mögliche Einsparungen
localhost 1st Party	1.006,7 KiB	810,8 KiB
...css/sdx.min.css (localhost)	446,9 KiB	389,9 KiB
...js/sdx.min.js (localhost)	193,7 KiB	142,3 KiB
...webcomponents/p-c31cad0b.entry.js (localhost)	97,4 KiB	74,3 KiB
...webcomponents/p-0e7aefe8.entry.js (localhost)	53,9 KiB	43,3 KiB
...webcomponents/p-d831bc11.entry.js (localhost)	54,7 KiB	43,1 KiB
...webcomponents/p-414dd07c.entry.js (localhost)	35,0 KiB	29,2 KiB
...webcomponents/p-53dbf7d7.js (localhost)	28,7 KiB	18,1 KiB
...webcomponents/p-bc913fa4.entry.js (localhost)	19,2 KiB	15,9 KiB
...webcomponents/p-dd3218d3.entry.js (localhost)	18,8 KiB	15,7 KiB
http://localhost:3000	14,5 KiB	10,2 KiB
...webcomponents/p-f772284e.js (localhost)	15,2 KiB	8,9 KiB
...webcomponents/webcomponents.esm.js (localhost)	10,6 KiB	7,6 KiB
...webcomponents/p-7d1868eb.entry.js (localhost)	6,0 KiB	4,4 KiB
...webcomponents/p-6a2de715.entry.js (localhost)	4,8 KiB	3,2 KiB
...webcomponents/p-a5d22636.entry.js (localhost)	4,2 KiB	2,6 KiB
...webcomponents/p-3ed4d64e.entry.js (localhost)	3,2 KiB	2,0 KiB

Reduziere nicht verwendete CSS

0,56 Sek ^

Du kannst ungültige Regeln in Stylesheets reduzieren und CSS-Code zurückstellen, der nicht für ohne Scrollen sichtbare Inhalte („above the fold“) verwendet wird, um den Datenverbrauch durch Netzwerkaktivität zu senken. [Informationen zum Reduzieren von nicht verwendetem CSS-Code.](#)

FCP LCP

URL	Übertragungsgröße	Mögliche Einsparungen
localhost 1st Party	447,2 KiB	398,9 KiB
...css/sdx.min.css (localhost)	447,2 KiB	398,9 KiB
Unattributable	11,2 KiB	11,0 KiB
:root { --sdx-header-height-mobile: 72px; --sdx-header-height-mobile-with-breadcrumbs: 120px; --sdx...	11,2 KiB	11,0 KiB

Ressourcen beseitigen, die das Rendering blockieren

0,56 Sek ^

Ressourcen blockieren den First Paint deiner Seite. Versuche, wichtiges JS und wichtige CSS inline anzugeben und alle nicht kritischen JS und Stile zurückzustellen. [Informationen dazu, wie sich Ressourcen eliminieren lassen, die das Rendering blockieren.](#) FCP LCP

Drittanbieter-Ressourcen anzeigen (1)

URL	Übertragungsgröße	Mögliche Einsparungen
localhost 1st Party	641,2 KiB	1.220 ms
...css/sdx.min.css (localhost)	447,2 KiB	730 ms
...js/sdx.min.js (localhost)	194,0 KiB	490 ms
Tealium Tag-Manager	3,9 KiB	220 ms
...dev/utag.sync.js (tags.tiqcdn.com)	3,9 KiB	220 ms

Mithilfe dieser Empfehlungen lässt sich die Ladezeit deiner Seite möglicherweise verkürzen. Sie haben keinen [direkten Einfluss](#) auf die Leistungsbewertung.

DIAGNOSE

▲ Darauf achten, dass der Text während der Webfont-Ladevorgänge sichtbar bleibt

Nutze das CSS-Feature `font-display`, damit der Text für Nutzer sichtbar ist, während Webfonts geladen werden. [Weitere Informationen zu font-display.](#) FCP LCP

URL	Mögliche Einsparungen
localhost 1st Party	150 ms

URL	Mögliche Einsparungen
...TheSans/TheSansB_400_.woff2 (localhost)	20 ms
...TheSans/TheSansB_700_.woff2 (localhost)	30 ms
...TheSans/TheSansB_600_.woff2 (localhost)	30 ms
...TheSans/TheSansB_500_.woff2 (localhost)	30 ms
...sdx-icons/sdx-icons.woff2 (localhost)	40 ms

▲ **Verwendet keine passiven Listener zur Verbesserung der Scrollleistung** ^

Wenn du deine Event-Listener für Tipp- und Mausevents als passive markierst, kannst du damit die Scrollleistung deiner Seite verbessern. [Weitere Informationen zur Verwendung von passiven Event-Listnern.](#)

Quelle

localhost 1st Party

[./node_modules/@stencil/core/internal/client/index.js?app-data=conditional:3268:47](/node_modules/@stencil/core/internal/client/index.js?app-data=conditional:3268:47)

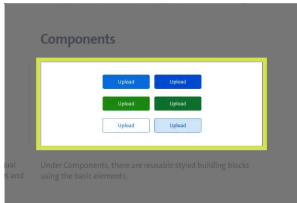
▲ **Bildelemente haben keine explizite width und height** ^

Lege eine explizite Breite und Höhe für Bildelemente fest, um Layoutverschiebungen zu reduzieren und den CLS-Wert zu verbessern. [Informationen zum Festlegen von Bildabmessungen](#)

CLS

URL

localhost 1st Party



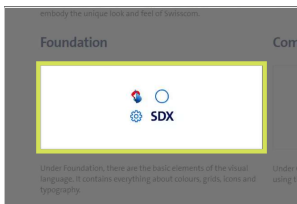
Components

button groups

```

```

...all_the_basics/components.png (localhost)



Foundation

Examples typography, colours and components

```

```

...all_the_basics/foundation.png (localhost)

URL

	<p>Code snippet</p> <pre></pre>	<p>...all_the_basics/utilities.png (localhost)</p>
	<p>Figma logo</p> <pre></pre>	<p>...all_the_basics/designer.png (localhost)</p>

Statische Inhalte mit einer effizienten Cache-Richtlinie bereitstellen — 9 Ressourcen gefunden

Eine lange Verweildauer im Cache kann wiederholte Besuche deiner Seite beschleunigen. [Weitere Informationen zu effizienten Cache-Richtlinien.](#)

URL	Cache-TTL	Übertragungsgröße
Tealium Tag-Manager		59 KiB
...dev/utag.js (tags.tiqcdn.com)	5 Min.	55 KiB
...dev/utag.sync.js (tags.tiqcdn.com)	5 Min.	4 KiB
...tiqapp/utag.v.js?a=swisscom/main/202...&nocookie=1 (tags.tiqcdn.com)	5 Min.	0 KiB
Adobe Tag Manager Tag-Manager		3 KiB
...d199e5db12b9/RC079aeec...-source.min.js (assets.adobedtm.com)	1 Std.	1 KiB
...d199e5db12b9/RC777e9ba...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC991f58f...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC6015071...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
Facebook Social		0 KiB
/tr?id=865...&ev=Adobe-Aud...&cd[segID]=13706038,11471880&noscript=1 (www.facebook.com)	None	0 KiB
/tr?id=865...&ev=Adobe-Aud...&cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB

Übermäßige DOM-Größe vermeiden — 886 Elemente



Ein großes DOM führt zu hoher Arbeitsspeichernutzung, langwierigen [Stilberechnungen](#) und kostspieligen [dynamischen Umbrüchen im Layout](#). [Informationen zum Vermeiden eines zu großen DOMs](#). TBT

Statistik	Element	Wert
DOM-Elemente insgesamt		886
Maximale DOM-Tiefe	Accessibility <code></code>	28
Maximale Anzahl von untergeordneten Elementen	AccordionsBadgesButtonsCardsCarouselChart (bar horizontal)Chart (bar vertical)C... <code><ul class="list vertical"></code>	41

Verkettung kritischer Anfragen vermeiden — 17 Ketten gefunden



In den unten aufgeführten Ketten kritischer Anfragen kannst du sehen, welche Ressourcen mit einer hohen Priorität geladen werden. Versuche, die Ketten zu kürzen, die Downloadgröße von Ressourcen zu reduzieren oder das Herunterladen unnötiger Ressourcen zurückzustellen, um den Seitenaufbau zu beschleunigen. [Informationen dazu, wie sich das Verketteten kritischer Anfragen vermeiden lässt](#). FCP LCP

Maximale Latenz für kritischen Pfad: **1.332,367 ms**

Anfangsnavigation



...webcomponents/p-62030357.js (localhost) - 24,507 ms, 1,15 KiB
...webcomponents/p-c31cad0b.entry.js (localhost)
...webcomponents/p-112455b1.js (localhost) - 20,858 ms, 0,74 KiB
...webcomponents/p-d831bc11.entry.js (localhost) - 9,85 ms, 55,00 KiB
...dev/utag.sync.js (tags.tiqcdn.com) - 28,267 ms, 3,92 KiB
/js/tabs.js (localhost) - 9,038 ms, 2,55 KiB
/js/smooth-anchor-scrolling.js (localhost) - 9,318 ms, 2,15 KiB

○ Halte die Anfrageanzahl niedrig und die Übertragungsgröße gering — 46 Anfragen • 1.618 KiB ^

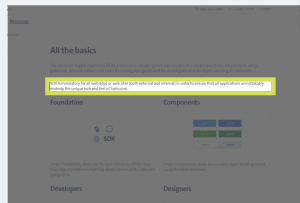
Füge zum Einrichten von Budgets für die Anzahl und Größe von Seitenressourcen eine budget.json-Datei hinzu. [Weitere Informationen zu Leistungsbudgets.](#)

Ressourcentyp	Anfragen	Übertragungsgröße
Gesamt	46,0	1.618,4 KiB
Skript	29,0	702,6 KiB
Stylesheet	1,0	447,2 KiB
Schriftart	5,0	279,8 KiB
Bild	7,0	171,9 KiB
Dokument	1,0	14,8 KiB
Sonstige	3,0	2,1 KiB
Medien	0,0	0,0 KiB
Drittanbieter	13,0	146,6 KiB

○ Largest Contentful Paint-Element — 1 Element gefunden ^

Dies ist das größte Inhaltselement, das im Darstellungsbereich angezeigt wird. [Weitere Informationen zum Largest Contentful Paint-Element](#) LCP

Element



SDX is mandatory for all web apps or web sites (both external and internal) in

...

<p>

○ Lange Hauptthread-Aufgaben vermeiden — 6 lange Aufgaben gefunden ^

Listet die längsten Aufgaben im Hauptthread auf. Das ist nützlich, um die wichtigsten Ursachen für die Eingabeverzögerungen zu ermitteln. [Informationen dazu, wie sich lange Aufgaben im Hauptthread vermeiden lassen](#) TBT

Drittanbieter-Ressourcen anzeigen (3)

URL	Beginn	Dauer
Adobe Tag Manager Tag-Manager		226 ms
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	870 ms	162 ms
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	1.078 ms	64 ms
localhost 1st Party		194 ms
http://localhost:3000	208 ms	84 ms
...webcomponents/p-f772284e.js (localhost)	1.894 ms	56 ms
http://localhost:3000	299 ms	54 ms
Tealium Tag-Manager		59 ms
...dev/utag.js (tags.tiqcdn.com)	467 ms	59 ms

Nicht zusammengesetzte Animationen vermeiden — 2 animierte Elemente gefunden ^

Nicht zusammengesetzte Animationen werden eventuell nicht richtig gerendert und können den CLS-Wert erhöhen. [Informationen dazu, wie sich nicht zusammengefasste Animationen vermeiden lassen](#) CLS

Element	Name
	copy link <pre><button class="clickable button-reset transparent" href="" type="button"></pre>
Nicht unterstützte CSS-Property: visibility	visibility
	ul.list > li.item > a.link > span.img <pre></pre>
Nicht unterstützte CSS-Properties: background-position-x, background-position-y	repeatingAnimation

Element	Name
Nicht unterstützte CSS-Properties: background-position-x, background-position-y	initialAnimation

Weitere Informationen zur Leistung deiner App findest du hier. Diese Angaben haben keinen [direkten Einfluss](#) auf die Leistungsbewertung.

BESTANDENE PRÜFUNGEN (27)

Ausblenden

● Bilder richtig dimensionieren ^

Stelle Bilder bereit, die eine angemessene Größe haben, um mobile Daten zu sparen und die Ladezeit zu verbessern. [Weitere Informationen.](#)

● Nicht sichtbare Bilder aufschieben ^

Wenn du Lazy Loading für nicht sichtbare und versteckte Bilder verwendest und sie zurückstellst, bis alle wichtigen Ressourcen geladen wurden, kannst du die Zeit bis Interaktivität reduzieren. [Weitere Informationen.](#)

● CSS komprimieren — Mögliche Einsparung von 73 KiB ^

Durch das Reduzieren von CSS-Dateien lassen sich Netzwerklasten senken. [Informationen zum Reduzieren von CSS.](#) FCP LCP

URL	Übertragungsgröße	Mögliche Einsparungen
localhost 1st Party	447,2 KiB	73,0 KiB
...css/sdx.min.css (localhost)	447,2 KiB	73,0 KiB

● JavaScript komprimieren ^

Durch die Komprimierung von JavaScript-Dateien können Nutzlastgrößen und die Zeit zum Parsen von Skripts reduziert werden. [Informationen zum Reduzieren von JavaScript.](#) FCP LCP

● Reduziere nicht verwendetes JavaScript — Mögliche Einsparung von 241 KiB ^

Um den Datenverbrauch durch Netzwerkaktivität zu senken, kannst du nicht verwendetes JavaScript reduzieren und das Laden von Skripts zurückstellen, bis sie benötigt werden.

[Informationen zum Reduzieren von nicht verwendetem JavaScript.](#) LCP Drittanbieter-Ressourcen anzeigen (1)

URL	Übertragungsgröße	Mögliche Einsparungen
localhost 1st Party	291,7 KiB	219,1 KiB
...js/sdx.min.js (localhost)	194,0 KiB	164,3 KiB
...webcomponents/p-c31cad0b.entry.js (localhost)	97,7 KiB	54,8 KiB
...node_modules/flatpickr/dist/flatpickr.js	47,4 KiB	44,0 KiB
...webcomponents/components/input/input.tsx	9,0 KiB	9,0 KiB
(unmapped)	2,8 KiB	1,2 KiB
Adobe Tag Manager Tag-Manager	82,2 KiB	22,1 KiB
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	82,2 KiB	22,1 KiB

● Bilder effizient codieren ^

Optimierte Bilder werden schneller geladen und verbrauchen weniger mobile Daten. [Informationen zum effizienten Codieren von Bildern.](#)

● Bilder in modernen Formaten bereitstellen — Mögliche Einsparung von 39 KiB ^

Bildformate wie WebP und AVIF bieten oft eine bessere Komprimierung als PNG oder JPEG, wodurch sie schneller heruntergeladen werden und weniger Daten verbrauchen. [Weitere Informationen zu modernen Bildformaten.](#)

URL	Größe der Ressource	Mögliche Einsparungen
localhost 1st Party	44,3 KiB	38,7 KiB
 ...all_the_basics/utilities.png (localhost)	44,3 KiB	38,7 KiB
Code snippet <pre><img class="image image--frame-light margin-bottom-3" src="/img/design/all_the_</pre>		

URL	Größe der Ressource	Mögliche Einsparungen
<code>basics/utilities.png"</code> <code>alt="Code snippet"></code>		

● Vorverbindung zu erforderlichen Ursprüngen aufbauen ^

Mithilfe von Hinweisen auf Ressourcen wie `preconnect` oder `dns-prefetch` lassen sich frühzeitig Verbindungen zu wichtigen Drittanbieterursprüngen herstellen. [Informationen zum Vorverbinden mit erforderlichen Ursprüngen.](#) FCP LCP

● Kurze Erstreaktionszeit des Servers — Stammdokument brauchte 0 ms ^

Achte auf eine möglichst kurze Serverantwortzeit für das Hauptdokument, weil alle anderen Anfragen davon abhängen. [Weitere Informationen zum Messwert „Time to First Byte“.](#) FCP LCP

URL	Zeitaufwand
localhost 1st Party	0 ms
http://localhost:3000	0 ms

● Mehrere Weiterleitungen auf die Seite vermeiden ^

Weiterleitungen führen zu zusätzlichen Verzögerungen, bevor die Seite geladen werden kann. [Informationen zum Vermeiden von Seitenweiterleitungen.](#) FCP LCP

○ Wichtige Anforderungen vorab laden ^

Mit `<link rel=preload>` kannst du das Abrufen von Ressourcen priorisieren, die derzeit beim Seitenaufbau erst später angefordert werden. [Informationen zum Vorabladen wichtiger Anfragen.](#) FCP LCP

● Verwende HTTP/2 ^

HTTP/2 bietet gegenüber HTTP/1.1 viele Vorteile, wie z. B. binäre Header und Multiplexverfahren. [Weitere Informationen zu HTTP/2.](#)

● Videoformate für animierte Inhalte verwenden ^

Große GIF-Dateien sind nur bedingt für die Auslieferung animierter Inhalte geeignet. Du kannst stattdessen MPEG4- oder WebM-Videos für Animationen und PNG oder WebP für statische Bilder verwenden und so die Netzwerkbytes reduzieren. [Weitere Informationen zu effizienten Videoformaten](#) LCP

● In JavaScript-Bundles doppelt vorhandene Module entfernen ^

Entferne große, doppelt vorhandene JavaScript-Module aus Bundles, um unnötige Datenübertragungen im Netzwerk zu reduzieren. TBT

● Verhindern, dass in modernen Browsern veraltetes JavaScript bereitgestellt wird — Mögliche Einsparung von 7 KiB ^

Dank Polyfills und Transformationen können veraltete Browser die neuen JavaScript-Funktionen nutzen. Bei modernen Browsern hingegen sind viele davon nicht erforderlich. Für dein JavaScript-Bundle solltest du eine moderne Skriptimplementierungsstrategie unter Verwendung der module/nomodule-Funktionserkennung einsetzen. So kannst du einerseits den an moderne Browser übermittelten Code auf das Nötige reduzieren und gleichzeitig veraltete Browser bestmöglich unterstützen. [Informationen zur Verwendung modernen JavaScripts](#) TBT

URL	Mögliche Einsparungen
Tealium Tag-Manager	6,5 KiB
...dev/utag.js (tags.tiqcdn.com)	6,5 KiB
...dev/utag.js:1475:14 (tags.tiqcdn.com)	Object.keys
Adobe Tag Manager Tag-Manager	0,1 KiB
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	0,1 KiB
...4cf4f6043663/launch-0c232d70d3dd-development.min.js:7:1084 (assets.adobedtm.com)	@babel/plugin-transform-classes

○ Largest Contentful Paint-Bild vorab laden ^

Wenn der Seite das LCP-Element dynamisch hinzugefügt wird, solltest du das Bild vorab laden, um den LCP zu verbessern. [Weitere Informationen zum Vorabladen von LCP-Elementen.](#) LCP

● Vermeidet sehr große Netzwerklasten — Die Gesamtgröße war 1.726 KiB ^

Große Netzwerklasten kosten Nutzer bares Geld und hängen eng mit langen Ladezeiten

zusammen. [Informationen zum Verringern der Nutzlastgröße.](#) LCP

Drittanbieter-Ressourcen anzeigen (2)

URL	Übertragungsgröße
localhost 1st Party	1.136,5 KiB
...css/sdx.min.css (localhost)	447,2 KiB
...js/sdx.min.js (localhost)	194,0 KiB
/img/favicon.ico (localhost)	107,9 KiB
...assets/lifeform-spritesheet@2x.png (localhost)	104,4 KiB
...webcomponents/p-c31cad0b.entry.js (localhost)	97,7 KiB
...sdx-icons/sdx-icons.woff2 (localhost)	76,1 KiB
...webcomponents/p-d831bc11.entry.js (localhost)	55,0 KiB
...webcomponents/p-0e7aefe8.entry.js (localhost)	54,2 KiB
Adobe Tag Manager Tag-Manager	82,2 KiB
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	82,2 KiB
Tealium Tag-Manager	54,9 KiB
...dev/utag.js (tags.tiqcdn.com)	54,9 KiB

○ Markierungen und Messungen für das Nutzertiming ^

Du kannst die User Timing API in deine App einbinden. Damit lässt sich die Leistung der App während wichtiger Nutzerinteraktionen in der Praxis messen. [Weitere Informationen zu User Timing-Markierungen.](#)

● JavaScript-Ausführungszeit — 0,7 s ^

Versuche, die Zeit für das Parsen, Kompilieren und Ausführen von JavaScript zu reduzieren. Die Bereitstellung kleinerer JS-Nutzlasten kann dabei helfen. [Informationen zum Reduzieren der JavaScript-Ausführungszeit.](#) TBT

Drittanbieter-Ressourcen anzeigen (2)

URL	CPU-Zeit insgesamt	Skriptauswertung	Parsen von Skripten
localhost 1st Party	932 ms	279 ms	4 ms

URL	CPU-Zeit insgesamt	Skriptauswertung	Parsen von Skripten
http://localhost:3000	533 ms	13 ms	4 ms
...webcomponents/p-f772284e.js (localhost)	266 ms	229 ms	0 ms
...webcomponents/p-53dbf7d7.js (localhost)	133 ms	36 ms	0 ms
Unattributable	326 ms	78 ms	0 ms
Unattributable	326 ms	78 ms	0 ms
Adobe Tag Manager Tag-Manager	253 ms	220 ms	23 ms
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	253 ms	220 ms	23 ms
Tealium Tag-Manager	131 ms	115 ms	15 ms
...dev/utag.js (tags.tiqcdn.com)	131 ms	115 ms	15 ms

● **Minimiert den Aufwand für den Hauptthread — 1,8 s** ^

Versuche, die Zeit für das Parsen, Kompilieren und Ausführen von JavaScript zu reduzieren. Die Bereitstellung kleinerer JS-Nutzlasten kann dabei helfen. [Informationen dazu, wie sich der Aufwand für den Hauptthread minimieren lässt](#) TBT

Kategorie	Zeitaufwand
Script Evaluation	772 ms
Other	534 ms
Style & Layout	309 ms
Rendering	79 ms
Script Parsing & Compilation	62 ms
Parse HTML & CSS	61 ms
Garbage Collection	4 ms

● **Drittanbiaternutzung minimieren — Code von Drittanbietern hat den Hauptthread 80 ms lang blockiert** ^

Code von Drittanbietern kann die Ladegeschwindigkeit erheblich beeinträchtigen. Beschränke die Zahl redundanter Drittanbieter und versuche, solchen Code erst zu laden, nachdem die Seite vollständig geladen wurde. [Informationen dazu, wie sich die Auswirkungen von Drittanbietercode minimieren lassen.](#) TBT

Drittanbieter	Übertragungsgröße	Dauer der Blockierung des Hauptthreads
Adobe Tag Manager Tag-Manager	85 KiB	70 ms
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	82 KiB	70 ms
Tealium Tag-Manager	60 KiB	8 ms
...dev/utag.js (tags.tiqcdn.com)	0 KiB	8 ms
...dev/utag.js (tags.tiqcdn.com)	55 KiB	0 ms
Weitere Ressourcen	4 KiB	0 ms
adobedc.net	2 KiB	0 ms
Facebook Social	0 KiB	0 ms

Lazy Loading von Ressourcen von Drittanbietern mit Fassaden ^

Für einen Teil des eingebetteten Codes von Drittanbietern kann Lazy Loading verwendet werden. Du kannst ihn durch eine Fassade ersetzen, bis er benötigt wird. [Informationen zum Zurückstellen von Drittanbietern mit einer Fassade.](#) TBT

Largest Contentful Paint-Bild wurde nicht mit Lazy Loading geladen ^

Ohne Scrollen sichtbare Bilder („above the fold“) die mit Lazy Loading geladen werden, werden später im Lebenszyklus der Seite gerendert. Dies kann zu Verzögerungen beim Largest Contentful Paint führen. [Weitere Informationen zum optimalen Lazy Loading.](#) LCP

Umfangreiche Layoutverschiebungen vermeiden ^

Diese DOM-Elemente tragen am stärksten zur CLS der Seite bei. [Informationen zur Verbesserung der CLS](#) CLS

Verwendet kein `document.write()` ^

Für Nutzer mit langsamen Verbindungen können externe Skripts, die dynamisch über `document.write()` eingefügt werden, den Seitenaufbau um einige Sekunden verzögern. [Informationen zum Vermeiden von „document.write\(\)“.](#)

Hat ein `<meta name="viewport">`-Tag mit `width` oder `initial-scale` ^

Ein `<meta name="viewport">` optimiert deine App für Bildschirmgrößen von Mobilgeräten und verhindert darüber hinaus [300-Millisekunden-Verzögerungen bei Nutzereingaben](#). [Weitere](#)

[Informationen zur Verwendung des Darstellungsbereich-Meta-Tags.](#) TBT

- Seite hat die Wiederherstellung des Back-Forward-Caches nicht verhindert ^

Viele Bedienvorgänge werden ausgeführt, indem eine vorherige Seite aufgerufen oder zur nächsten gegangen wird. Der Back-Forward-Cache (bfcache) kann diese Vorgänge beschleunigen. [Weitere Informationen zum bfcache](#)



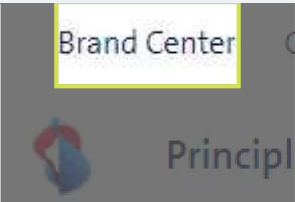

Barrierefreiheit

Mit diesen Prüfungen erfährst du, [wie du die Barrierefreiheit deiner Web-App verbesserst](#). Nur bestimmte Probleme mit der Barrierefreiheit können durch automatisierte Tests erkannt werden. Deshalb ist es empfehlenswert, zusätzlich manuelle Tests durchzuführen.


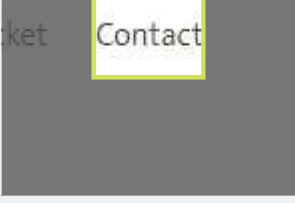


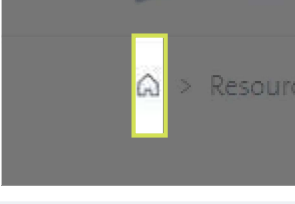
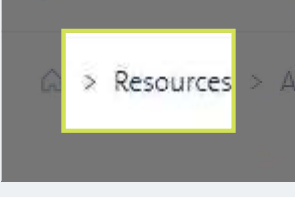
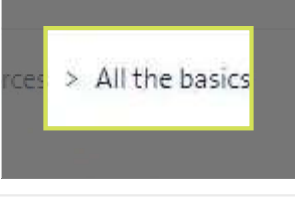
TABELLEN UND LISTEN

- ▲ Listenelemente (``) befinden sich nicht in übergeordneten ``-, ``- oder `<menu>`-Elementen. ^

Listenelemente (``) müssen sich in einem übergeordneten ``-, ``- oder `<menu>`-Element befinden, damit sie von Screenreadern richtig angesagt werden können. [Weitere Informationen zum Strukturieren von Listen.](#)

Fehlerhafte Elemente	
	Brand Center <code><li class="item leaf"></code>
	Git Repo <code><li class="item leaf"></code>

Fehlerhafte Elemente

	Stay up-to-date <code><li class="item leaf"></code>
	Create a ticket <code><li class="item leaf"></code>
	Contact <code><li class="item leaf"></code>
	Principles <code><li class="item leaf"></code>
	Resources <code><li class="item active"></code>
	icon-home <code><li class="item"></code>
	Resources <code><li class="item"></code>
	All the basics <code><li class="item"></code>

Hier findest du Möglichkeiten, um das Lesen von Daten in Tabellen oder Listen mit Hilfstechnologie wie Screenreadern zu verbessern.

The page has a logical tab order ^

Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. [Learn more about logical tab ordering.](#)

 Interactive controls are keyboard focusable ^

Custom interactive controls are keyboard focusable and display a focus indicator. [Learn how to make custom controls focusable.](#)

 Interactive elements indicate their purpose and state ^

Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. [Learn how to decorate interactive elements with affordance hints.](#)

 The user's focus is directed to new content added to the page ^

If new content, such as a dialog, is added to the page, the user's focus is directed to it. [Learn how to direct focus to new content.](#)

 User focus is not accidentally trapped in a region ^

A user can tab into and out of any control or region without accidentally trapping their focus. [Learn how to avoid focus traps.](#)

 Custom controls have associated labels ^

Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. [Learn more about custom controls and labels.](#)

 Custom controls have ARIA roles ^

Custom interactive controls have appropriate ARIA roles. [Learn how to add roles to custom controls.](#)

 Visual order on the page follows DOM order ^

DOM order matches the visual order, improving navigation for assistive technology. [Learn more about DOM and visual ordering.](#)

 Offscreen content is hidden from assistive technology ^

Offscreen content is hidden with display: none or aria-hidden=true. [Learn how to properly hide offscreen content.](#)

○ HTML5 landmark elements are used to improve navigation ^

Landmark elements (<main>, <nav>, etc.) are used to improve the keyboard navigation of the page for assistive technology. [Learn more about landmark elements.](#)

Diese Prüfungen sind für Bereiche vorgesehen, für die automatische Testtools nicht geeignet sind. Weitere Informationen findest du in unserem Leitfaden zur [Durchführung einer Prüfung auf Barrierefreiheit.](#)

BESTANDENE PRÜFUNGEN (18)

Ausblenden

● [aria-*]-Attribute entsprechen ihren Rollen ^

Jede ARIA-role unterstützt eine bestimmte Untergruppe von aria-*-Attributen. Wenn sie jedoch falsch zugeordnet sind, werden die aria-*-Attribute ungültig. [Informationen zum Zuordnen von ARIA-Attributen zu ihren Rollen.](#)

● [aria-hidden="true"] ist in dem Dokument <body> nicht vorhanden ^

Hilfstechnologien wie Screenreader funktionieren nicht richtig, wenn für den <body> des Dokuments aria-hidden="true" festgelegt ist. [Informationen zu den Auswirkungen von aria-hidden auf den Textbereich des Dokuments.](#)

● [role]-Elemente verfügen über alle erforderlichen [aria-*]-Attribute ^

Für einige ARIA-Rollen sind Attribute erforderlich, die Screenreadern den Zustand des Elements beschreiben. [Weitere Informationen zu Rollen und erforderlichen Attributen.](#)

● [role]-Werte sind gültig ^

Für ARIA-Rollen müssen gültige Werte angegeben sein, damit sie die beabsichtigten Hilfsfunktionen erfüllen können. [Weitere Informationen zu gültigen ARIA-Rollen.](#)

● [aria-*]-Attribute weisen gültige Werte auf ^

Hilfstechnologien wie Screenreader können ARIA-Attribute mit ungültigen Werten nicht interpretieren. [Weitere Informationen zu gültigen Werten für ARIA-Attribute.](#)

● [aria-*]-Attribute sind gültig und richtig geschrieben ^

Hilfstechnologien wie Screenreader können ARIA-Attribute mit ungültigen Namen nicht interpretieren. [Weitere Informationen zu gültigen ARIA-Attributen.](#)

● Die Namen der Schaltflächen sind für Screenreader zugänglich ^

Wenn eine Schaltfläche keinen barrierefreien Namen hat, wird sie von Screenreadern nur als „Schaltfläche“ angesagt. Dadurch ist sie für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Informationen zum barrierefreien Gestalten von Schaltflächen.](#)

● ARIA-IDs sind eindeutig ^

Der Wert einer ARIA-ID muss eindeutig sein, damit andere Instanzen nicht von Hilfstechnologien übersehen werden. [Informationen zum Korrigieren doppelter ARIA-IDs.](#)

● Bildelemente verfügen über [alt]-Attribute ^

Für informative Elemente sollte ein kurzer, beschreibenden alternativer Text verwendet werden. Dekorative Elemente können mit einem leeren ALT-Attribut ignoriert werden. [Weitere Informationen zum Attribut alt.](#)

● [user-scalable="no"] wird nicht im <meta name="viewport">-Element verwendet und das [maximum-scale]-Attribut ist nicht kleiner als 5. ^

Wenn du die Zoomfunktion deaktivierst, können Nutzer mit eingeschränktem Sehvermögen, die auf die Bildschirmvergrößerung angewiesen sind, den Inhalt einer Webseite nicht richtig sehen. [Weitere Informationen zum Darstellungsbereich-Meta-Tag.](#)

● [aria-hidden="true"]-Elemente enthalten keine fokussierbaren Unterelemente ^

Fokussierbare Nachfolgerelemente in einem [aria-hidden="true"]-Element führen dazu, dass Nutzer von Hilfstechnologien wie Screenreadern solche interaktiven Elemente nicht verwenden können. [Informationen zu den Auswirkungen von aria-hidden auf fokussierbare Elemente.](#)

● Das Kontrastverhältnis von Hintergrund- und Vordergrundfarben ist ausreichend ^

Text mit geringem Kontrast ist für viele Nutzer schlecht oder gar nicht lesbar. [Informationen zu einem ausreichenden Farbkontrast.](#)

● Dokument enthält ein `<title>`-Element ^

Der Titel gibt Screenreader-Nutzern einen Überblick über die Seite. Nutzer von Suchmaschinen verlassen sich stark auf ihn, um zu entscheiden, ob eine Seite für ihre Suche relevant ist. [Weitere Informationen zu Dokumenttiteln.](#)

● `<html>`-Element hat ein `[lang]`-Attribut ^

Wenn für eine Seite kein `lang`-Attribut angegeben ist, nehmen Screenreader an, dass sie in der Standardsprache vorliegt, die der Nutzer beim Einrichten des Screenreaders ausgewählt hat. Ist das nicht der Fall, gibt der Screenreader den Inhalt der Seite möglicherweise falsch aus. [Weitere Informationen zum `lang`-Attribut.](#)

● Das `<html>`-Element hat einen gültigen Wert für sein `[lang]`-Attribut ^

Wenn ein gültiger [BCP-47-Sprachcode](#) angegeben wird, kann der Text von einem Screenreader korrekt wiedergegeben werden. [Informationen zur Verwendung des Attributs `lang`.](#)

● Links haben einen leicht erkennbaren Namen ^

Linktext, der erkennbar, einzigartig und fokussierbar ist, erleichtert Screenreader-Nutzern die Verwendung. Dies gilt auch für alternativen Text für Bilder, die als Links verwendet werden. [Informationen zu barrierefreien Links.](#)

● Listen enthalten nur ``-Elemente und Elemente zur Skriptunterstützung (`<script>` sowie `<template>`). ^

Screenreader sagen Listen auf bestimmte Art und Weise an. Wenn die Liste richtig strukturiert ist, kann der Screenreader sie besser ausgeben. [Weitere Informationen zum Strukturieren von Listen.](#)

● Überschriftenelemente werden in einer fortlaufenden absteigenden Reihenfolge angezeigt ^

Richtig geordnete Überschriften, die keine Ebenen überspringen, geben der Seite eine semantische Struktur. Nutzer von Hilfstechnologien können sich so leichter auf der Seite

zurechtfinden und die Inhalte besser verstehen. [Weitere Informationen zur Reihenfolge von Überschriften.](#)

NICHT ZUTREFFEND (25)

Ausblenden

[\[accesskey\]-Werte sind eindeutig](#) ^

Mithilfe von Tastenkombinationen können Nutzer schnell den Fokus auf einen Bereich der Seite verschieben. Damit die Navigation richtig funktioniert, darf jede Tastenkombination nur einmal vergeben sein. [Weitere Informationen zu Tastenkombinationen.](#)

[button-, link- und menuitem-Elemente haben zugängliche Namen](#) ^

Wenn ein Element keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Informationen zum barrierefreieren Gestalten von Befehselementen.](#)

[ARIA-Eingabefelder haben zugängliche Namen](#) ^

Wenn ein Eingabefeld keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Weitere Informationen zu Labels für Eingabefelder.](#)

[ARIA meter-Elemente haben zugängliche Namen](#) ^

Wenn ein Messtool-Element keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Weitere Informationen zum Benennen von meter-Elementen](#)

[ARIA progressbar-Elemente haben zugängliche Namen](#) ^

Wenn ein progressbar-Element keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Informationen zum Kennzeichnen von progressbar-Elementen.](#)

[Die Elemente mit einer ARIA-\[role\], deren untergeordnete Elemente eine bestimmte \[role\] enthalten müssen, haben alle erforderlichen untergeordneten Elemente.](#) ^

Einige übergeordnete ARIA-Rollen müssen bestimmte untergeordnete Rollen enthalten, damit sie die beabsichtigten Hilfsfunktionen erfüllen können. [Weitere Informationen zu Rollen und erforderlichen untergeordneten Elementen.](#)

- [\[role\]-Elemente](#) sind ihren jeweils erforderlichen übergeordneten Elementen untergeordnet ^

Einige untergeordnete ARIA-Rollen müssen in bestimmten übergeordneten Rollen enthalten sein, damit sie die beabsichtigten Hilfsfunktionen erfüllen können. [Weitere Informationen zu ARIA-Rollen und erforderlichen übergeordneten Elementen.](#)

- ARIA-Ein-/Aus-Schaltflächen haben zugängliche Namen ^

Wenn eine Ein-/Aus-Schaltfläche keinen barrierefreien Namen hat, wird sie von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist sie für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Weitere Informationen zu Ein-/Aus-Schaltflächen.](#)

- ARIA `tooltip`-Elemente haben zugängliche Namen ^

Wenn ein Kurzinfo-Element keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Weitere Informationen zum Benennen von tooltip-Elementen](#)

- ARIA `treeitem`-Elemente haben zugängliche Namen ^

Wenn ein `treeitem`-Element keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Weitere Informationen zum Kennzeichnen von treeitem-Elementen.](#)

- Die Seite enthält eine Überschrift, einen Link zum Überspringen oder einen Landmark-Bereich ^

Wenn Tastaturnutzer Inhalte umgehen können, die sich wiederholen, sorgt das für eine effizientere Navigation. [Weitere Informationen zum Umgehen von Blockierungen.](#)

- `<dl>`-Elemente enthalten ausschließlich Gruppen aus `<dt>`- und `<dd>`-Elementen sowie `<script>`-, `<template>`- oder `<div>`-Elemente, die richtig angeordnet sind. ^

Wenn Definitionslisten nicht korrekt mit Markup versehen sind, kann es zu verwirrenden oder ungenauen Screenreader-Ausgaben kommen. [Informationen zum Strukturieren von Definitionslisten.](#)

- Definitionlistenelemente sind in `<dl>`-Elemente eingefasst ^

Definitionlistenelemente (`<dt>` und `<dd>`) müssen in ein übergeordnetes `<dl>`-Element eingefasst sein, damit sie von Screenreadern richtig angesagt werden können. [Informationen zum Strukturieren von Definitionslisten.](#)

- `[id]`-Attribute zu aktiven, fokussierbaren Elementen sind eindeutig ^

Für alle fokussierbaren Elemente ist eine eindeutige `id` erforderlich, damit sie von Hilfstechnologien erkannt werden können. [Informationen zum Korrigieren doppelter `ids`.](#)

Kein Formularfeld hat mehrere Labels ^

Formularfelder mit mehreren Labels werden von Hilfstechnologien wie Screenreadern unter Umständen missverständlich angesagt, da sie entweder das erste, das letzte oder alle Labels verwenden. [Informationen zur Verwendung von Formularlabels.](#)

`<frame>`- oder `<iframe>`-Elemente verfügen über einen Titel ^

Screenreader-Nutzer sind auf Frametitel angewiesen, die die Inhalte von Frames beschreiben. [Weitere Informationen zu Frametiteln.](#)

`<input type="image">`-Elemente haben `[alt]`-Text ^

Wenn ein Bild als `<input>`-Schaltfläche verwendet wird, kann alternativer Text Screenreader-Nutzern helfen, den Zweck der Schaltfläche besser zu verstehen. [Informationen zum Alt-Text für Eingabebilder.](#)

Formularelemente sind mit Labels verknüpft ^

Durch Labels wird gewährleistet, dass Steuerelemente für Formulare von Hilfstechnologien wie Screenreadern richtig angesagt werden. [Weitere Informationen zu Labels für Formularelemente.](#)

Dieses Dokument verwendet `<meta http-equiv="refresh">` nicht ^

Nutzer rechnen nicht damit, dass eine Seite automatisch aktualisiert wird. Außerdem wird dadurch der Fokus wieder auf den Seitenanfang verschoben. Das kann für den Nutzer frustrierend oder irritierend sein. [Weitere Informationen zum Meta-Tag „Refresh“.](#)

`<object>`-Elemente haben alternativen Text ^

Screenreader können lediglich Textinhalte interpretieren. Wenn du `<object>`-Elementen alternativen Text hinzufügst, können Screenreader-Nutzer besser verstehen, was diese Elemente darstellen. [Weitere Informationen zum Alt-Text für `object`-Elemente.](#)

Kein Element hat einen `[tabindex]`-Wert größer als 0 ^

Ein Wert größer als 0 impliziert eine explizite Navigationsanordnung. Das ist zwar technisch möglich, aber für Nutzer, die auf Hilfstechnologien angewiesen sind, häufig frustrierend. [Weitere Informationen zum Attribut `tabindex`.](#)

- Zellen in einem "`<table>`"-Element, die das Attribut "`[headers]`" enthalten, verweisen auf Zellen in derselben Tabelle. ^

Screenreader bieten Funktionen, die die Navigation in Tabellen vereinfachen. Wenn du dafür sorgst, dass `<td>`-Zellen, die das Attribut `[headers]` verwenden, nur auf andere Zellen in derselben Tabelle verweisen, kann dies für Screenreader-Nutzer hilfreich sein. [Weitere Informationen zum Attribut headers.](#)

- Für `<th>`-Elemente und Elemente mit `[role="columnheader"/"rowheader"]` sind Datenzellen vorhanden, die sie beschreiben. ^

Screenreader bieten Funktionen, die die Navigation in Tabellen vereinfachen. Wenn du dafür sorgst, dass Tabellenüberschriften immer auf bestimmte Zellen verweisen, kann dies für Screenreader-Nutzer hilfreich sein. [Weitere Informationen zu Tabellenüberschriften.](#)

- `[lang]`-Attribute weisen einen gültigen Wert auf ^

Wenn ein gültiger [BCP-47-Sprachcode](#) für Elemente angegeben wird, kann der Text besser von Screenreadern ausgesprochen werden. [Informationen zur Verwendung des Attributs lang.](#)

- `<video>`-Elemente enthalten ein `<track>`-Element mit `[kind="captions"]` ^

Wenn ein Video Untertitel enthält, können gehörlose und hörgeschädigte Nutzer die Informationen im Video besser verstehen. [Weitere Informationen zu Untertiteln in Videos.](#)



Best Practices

VERTRAUEN UND SICHERHEIT

- ▲ Verwendet nicht HTTPS — 1 unsichere Anfrage gefunden ^

Alle Websites sollten durch HTTPS geschützt werden – selbst wenn sie keine sensiblen Daten enthalten. Auch [gemischte Inhalte](#), bei denen einige Ressourcen über HTTP geladen werden,

obwohl die ursprüngliche Anfrage über HTTPS gestellt wurde, sind zu vermeiden. HTTPS verhindert, dass andere die Website manipulieren oder die Kommunikation zwischen deiner App und deinen Nutzern mitverfolgen können, und ist eine Voraussetzung für HTTP/2 und viele neue Webplattform-APIs. [Weitere Informationen zu HTTPS.](#)

Unsichere URL	Lösung anfordern
Tealium Tag-Manager	
...dev/utag.js (tags.tiqcdn.com)	Zugelassen

○ Sicherstellen, dass CSP effektiv gegen XSS-Angriffe wirkt ^

Eine starke Content Security Policy (CSP) reduziert das Risiko für Cross-Site-Scripting-Angriffe (XSS-Angriffe) erheblich. [Weitere Informationen](#)

Beschreibung	Anweisung	Schweregrad
Kein CSP im erzwungenen Modus gefunden		Hoch

BESTANDENE PRÜFUNGEN (12)

Ausblenden

● Fordert während des Seitenaufbaus keine Berechtigung zur Standortbestimmung an ^

Wenn Websites den Standort ohne Begründung anfordern, sind Nutzer schnell misstrauisch oder irritiert. Versuche stattdessen, die Anforderung mit einer Nutzeraktion zu verbinden. [Weitere Informationen zur Berechtigung zur Standortbestimmung.](#)

● Fordert während des Seitenaufbaus keine Benachrichtigungsberechtigung an ^

Wenn Websites die Berechtigung zum Senden von Benachrichtigungen ohne Begründung anfordern, sind Nutzer schnell misstrauisch oder irritiert. Versuche stattdessen, die Anforderung mit Touch-Gesten zu verbinden. [Weitere Informationen zum verantwortungsvollen Einholen der Berechtigung zum Senden von Benachrichtigungen.](#)

● Allows users to paste into input fields ^

Preventing input pasting is a UX anti-pattern, and undermines good security policy. [Learn more about user-friendly input fields.](#)

● Zeigt Bilder mit einem korrekten Seitenverhältnis an ^

Die Bildgröße sollte dem natürlichen Seitenverhältnis entsprechen. [Weitere Informationen zum Seitenverhältnis von Bildern.](#)

● Stellt Bilder mit angemessener Auflösung bereit ^

Die originalen Abmessungen eines Bildes sollten proportional zu der Displaygröße und dem Pixel-Verhältnis sein, damit das Bild optimal angezeigt wird. [Weitere Informationen zu responsiven Bildern.](#)

● Seite verfügt über den HTML-DOCTYPE ^

Wenn du einen DOCTYPE angibst, verhinderst du, dass der Browser zum Quirks-Modus wechselt. [Weitere Informationen zum Deklarieren eines DOCTYPE.](#)

● Korrekt definierter Zeichensatz ^

Die Zeichencodierung muss deklariert werden. Dazu kann ein <meta>-Tag in den ersten 1024 Byte des HTML-Codes oder im HTTP-Antwortheader „Content-Type“ angegeben werden. [Weitere Informationen zum Deklarieren der Zeichencodierung.](#)

● Keine unload-Ereignis-Listener gefunden ^

Das unload-Ereignis wird nicht zuverlässig ausgelöst. Wenn der Listener darauf wartet, kann das Browseroptimierungen wie den Back-Forward-Cache beeinträchtigen. Bitte verwende stattdessen pagehide- oder visibilitychange-Ereignisse. [Weitere Informationen zum Entfernen von Event-Listnern](#)

● Vermeidet veraltete APIs ^

Verworfen APIs werden aus dem Browser entfernt. [Weitere Informationen zu verworfenen APIs.](#)

● Es wurden keine Browserfehler in der Konsole protokolliert ^

In der Konsole protokollierte Fehler weisen auf ungelöste Probleme hin. Sie können durch fehlgeschlagene Netzwerkanfragen und andere Browserprobleme verursacht werden. [Weitere Informationen](#)

● Keine Probleme im Issues-Bereich der Chrome-Entwicklertools ^

Im Issues-Bereich der Chrome-Entwicklertools wurden ungelöste Probleme protokolliert. Sie können durch fehlgeschlagene Netzwerkanfragen, unzureichende Sicherheitsmaßnahmen und andere Browser-Probleme verursacht sein. Öffne den Bereich mit den Problemen in Chrome-Entwicklertools, um weitere Details zu den einzelnen Problemen aufzurufen.

● Seite hat gültige Quellzuordnungen ^

Quellzuordnungen übersetzen reduzierten Code in den ursprünglichen Quellcode. Dies hilft Entwicklern beim Debugging in der Produktionsphase. Zusätzlich kann Lighthouse weitere Informationen liefern. Wir empfehlen, Quellzuordnungen bereitzustellen, um diese Vorteile zu nutzen. [Weitere Informationen zu Quellzuordnungen.](#)

URL	Zuordnungs-URL
localhost 1st Party	
...	...
webcomponents/webcomponents.esm.js (localhost)	webcomponents/webcomponents.esm.js.map (localhost)
...webcomponents/p-f772284e.js (localhost)	...webcomponents/p-f772284e.js.map (localhost)
...webcomponents/p-dd3218d3.entry.js (localhost)	...webcomponents/p-dd3218d3.entry.js.map (localhost)
...webcomponents/p-d831bc11.entry.js (localhost)	...webcomponents/p-d831bc11.entry.js.map (localhost)
...webcomponents/p-c31cad0b.entry.js (localhost)	...webcomponents/p-c31cad0b.entry.js.map (localhost)
...webcomponents/p-bc913fa4.entry.js (localhost)	...webcomponents/p-bc913fa4.entry.js.map (localhost)
...webcomponents/p-a5d22636.entry.js (localhost)	...webcomponents/p-a5d22636.entry.js.map (localhost)
...webcomponents/p-7d1868eb.entry.js (localhost)	...webcomponents/p-7d1868eb.entry.js.map (localhost)
...webcomponents/p-6a2de715.entry.js (localhost)	...webcomponents/p-6a2de715.entry.js.map (localhost)
...webcomponents/p-62030357.js (localhost)	...webcomponents/p-62030357.js.map (localhost)
...webcomponents/p-53dbf7d7.js (localhost)	...webcomponents/p-53dbf7d7.js.map (localhost)
...webcomponents/p-414dd07c.entry.js (localhost)	...webcomponents/p-414dd07c.entry.js.map (localhost)
...webcomponents/p-3f4649f5.js (localhost)	...webcomponents/p-3f4649f5.js.map (localhost)
...webcomponents/p-3ed4d64e.entry.js (localhost)	...webcomponents/p-3ed4d64e.entry.js.map (localhost)
...webcomponents/p-299c3b04.js (localhost)	...webcomponents/p-299c3b04.js.map (localhost)
...webcomponents/p-112455b1.js (localhost)	...webcomponents/p-112455b1.js.map (localhost)
...webcomponents/p-0e7aefe8.entry.js (localhost)	...webcomponents/p-0e7aefe8.entry.js.map (localhost)

URL	Zuordnungs-URL
...webcomponents/p-057a4d55.js (localhost)	...webcomponents/p-057a4d55.js.map (localhost)

NICHT ZUTREFFEND (2)

Ausblenden

Schriftarten mit `font-display: optional` werden vorab geladen ^

Du solltest `optional`-Schriftarten vorab laden, damit sie von Erstbesuchern verwendet werden können. [Weitere Informationen zum Vorabladen von Schriftarten](#)

JavaScript-Bibliotheken erkannt ^

Alle Front-End-JavaScript-Bibliotheken auf der Seite wurden erkannt. [Weitere Informationen.](#)



SEO

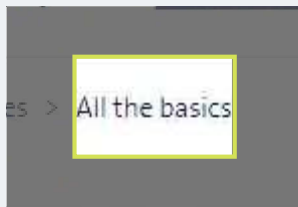
Mit diesen Prüfungen ist gewährleistet, dass bei deiner Seite grundlegende Tipps für die Suchmaschinenoptimierung berücksichtigt werden. Es gibt viele verschiedene Faktoren, die Lighthouse hier nicht bewertet und die sich auf das Such-Ranking deiner Seite auswirken können, einschließlich der [Core Web Vitals](#)-Performance. [Weitere Informationen zu Google Search Essentials](#)

CRAWLING UND INDEXIERUNG

Links können nicht gecrawlt werden ^

Suchmaschinen verwenden möglicherweise `href`-Attribute für Links, um Websites zu crawlen. Das `href`-Attribut von Anchor-Elementen muss auf ein geeignetes Ziel verweisen, damit mehr Seiten auf der Website gefunden werden können. [Informationen dazu, wie Links für Crawler zugänglich gemacht werden](#)

Link kann nicht gecrawlt werden



All the basics

```
<a class="link" aria-current="location" aria-label="All the basics" aria-describedby="87">
```

body.sdx > div.main-wrapper > a#content

```
<a id="content">
```

Damit deine Website in den Suchergebnissen angezeigt werden kann, benötigen Crawler Zugriff auf deine App.

ZUSÄTZLICHE ELEMENTE ZUR MANUELLEN ÜBERPRÜFUNG (1)

Ausblenden

Strukturierte Daten sind gültig ^

Du kannst das [Testtool für strukturierte Daten](#) und den [Lint für strukturierte Daten](#) ausführen, um strukturierte Daten zu validieren. [Weitere Informationen zu strukturierten Daten.](#)

Du kannst diese zusätzlichen Validierungen für deine Website ausführen, um weitere Best Practices für die SEO zu prüfen.

BESTANDENE PRÜFUNGEN (9)

Ausblenden

Hat ein `<meta name="viewport">`-Tag mit `width` oder `initial-scale` ^

Ein `<meta name="viewport">` optimiert deine App für Bildschirmgrößen von Mobilgeräten und verhindert darüber hinaus [300-Millisekunden-Verzögerungen bei Nutzereingaben](#). [Weitere Informationen zur Verwendung des Darstellungsbereich-Meta-Tags.](#) TBT

Dokument enthält ein `<title>`-Element ^

Der Titel gibt Screenreader-Nutzern einen Überblick über die Seite. Nutzer von Suchmaschinen verlassen sich stark auf ihn, um zu entscheiden, ob eine Seite für ihre Suche relevant ist. [Weitere Informationen zu Dokumenttiteln.](#)

Dokument enthält eine Meta-Beschreibung ^

Meta-Beschreibungen können in die Suchergebnisse aufgenommen werden, um die Seiteninhalte kurz zusammenzufassen. [Weitere Informationen zu Meta-Beschreibungen.](#)

Seite hat einen gültigen HTTP-Statuscode ^

Seiten mit ungültigen HTTP-Statuscodes werden möglicherweise nicht richtig indiziert. [Weitere Informationen zu HTTP-Statuscodes.](#)

Links haben beschreibenden Text ^

Wenn du beschreibenden Linktext verwendest, können Suchmaschinen deine Inhalte besser verstehen. [Informationen zu barrierefreien Links.](#)

Seite ist nicht von Indexierung ausgeschlossen ^

Suchmaschinen können deine Seiten nicht in die Suchergebnisse aufnehmen, wenn sie nicht berechtigt sind, sie zu crawlen. [Weitere Informationen zu Crawler-Anweisungen.](#)

Bildelemente verfügen über [alt]-Attribute ^

Für informative Elemente sollte ein kurzer, beschreibenden alternativer Text verwendet werden. Dekorative Elemente können mit einem leeren ALT-Attribut ignoriert werden. [Weitere Informationen zum Attribut alt.](#)

Dokument enthält ein gültiges hreflang-Element ^

Anhand von „hreflang“-Links können Suchmaschinen ermitteln, welche Version einer Seite sie in den Suchergebnissen für eine bestimmte Sprache oder Region anzeigen sollen. [Weitere Informationen zu hreflang.](#)

Dokument verwendet keine Plug-ins ^

Suchmaschinen können keine Plug-in-Inhalte indexieren. Außerdem werden Plug-ins auf vielen Geräten eingeschränkt oder nicht unterstützt. [Weitere Informationen zum Vermeiden von Plug-ins.](#)

NICHT ZUTREFFEND (4)

Ausblenden

robots.txt ist gültig ^

Wenn deine robots.txt-Datei fehlerhaft ist, können Crawler möglicherweise nicht nachvollziehen, wie deine Website gecrawlt oder indexiert werden soll. [Weitere Informationen zu robots.txt.](#)

Dokument enthält ein gültiges rel=canonical-Element ^

Über kanonische Links wird angegeben, welche URL in den Suchergebnissen angezeigt werden soll. [Weitere Informationen zu kanonischen Links.](#)

○ Dokument enthält gut lesbare Schriftgrößen ^

Schriftgrößen unter 12 px sind zu klein. Nutzer von Mobilgeräten müssen den Text mit den Fingern heranzoomen, um ihn lesen zu können. Achte darauf, dass mehr als 60 % des Seitentextes eine Schriftgröße von mindestens 12 px haben. [Weitere Informationen zu lesbaren Schriftgrößen.](#)

○ Größe von Tippzielen ist richtig eingestellt ^

Interaktive Elemente wie Schaltflächen und Links sollten groß genug sein (48 × 48 px) und über genügend Zwischenraum verfügen, damit sie leicht angetippt werden können. Sie dürfen sich nicht mit anderen Elementen überschneiden. [Weitere Informationen zu Tippzielen.](#)



PWA

Diese Prüfungen dienen dazu, die einzelnen Aspekte einer progressiven Web-App zu überprüfen. [Weitere Informationen dazu, was eine gute progressive Web-App ausmacht](#)

⊕ INSTALLIERBAR

▲ Das Manifest der Web-App oder der Service Worker erfüllt die Anforderungen an die Installierbarkeit nicht — 1 Grund ^

Der Service Worker ermöglicht es deiner App, viele Funktionen von progressiven Web-Apps zu nutzen, beispielsweise den Offlinemodus, das Hinzufügen zum Startbildschirm und Push-Benachrichtigungen. Bei ordnungsgemäßen Service Worker- und Manifestimplementierungen können Browser Nutzer direkt auffordern, deine Web-App zum Startbildschirm hinzuzufügen. Das kann zu mehr Interaktionen führen. [Weitere Informationen zu Anforderungen an die Installierbarkeit für Manifeste](#)

Grund für den Fehler

Die Seite hat keine Manifest-URL <link>

★ PWA-OPTIMIERT

- ▲ Es wurde kein Service Worker erkannt, der die Seite und `start_url` kontrolliert ^

Der Service Worker ermöglicht es deiner App, viele Funktionen von progressiven Web-Apps zu nutzen, beispielsweise den Offlinemodus, das Hinzufügen zum Startbildschirm und Push-Benachrichtigungen. [Weitere Informationen.](#)

- ▲ Nicht für einen benutzerdefinierten Startbildschirm konfiguriert ^

Failures: No manifest was fetched.

Wenn du deinen Startbildschirm passend zum Design deiner App gestaltest, vermittelst du den Nutzern schon beim Ladevorgang einen hochwertigen Eindruck. [Weitere Informationen zu Startbildschirmen.](#)

- ▲ Legt keine Designfarbe für die Adressleiste fest. ^

Failures: No manifest was fetched, No ``<meta name="theme-color">`` tag found.

Die Adressleiste des Browsers kann an das Design deiner Website angepasst werden. [Weitere Informationen zum Gestalten der Adressleiste.](#)

- Inhalt hat die richtige Größe für den Darstellungsbereich ^

Wenn die Breite deiner App-Inhalte nicht mit der des Darstellungsbereichs übereinstimmt, ist deine App möglicherweise nicht für Bildschirme von Mobilgeräten optimiert. [Informationen zum Anpassen der Größe von Inhalten an den Darstellungsbereich.](#)

- Hat ein `<meta name="viewport">`-Tag mit `width` oder `initial-scale` ^

Ein `<meta name="viewport">` optimiert deine App für Bildschirmgrößen von Mobilgeräten und verhindert darüber hinaus [300-Millisekunden-Verzögerungen bei Nutzereingaben.](#) [Weitere Informationen zur Verwendung des Darstellungsbereich-Meta-Tags.](#) TBT

- ▲ Das Manifest hat kein maskierbares Symbol **No manifest was fetched** ^

Mit einem maskierbaren Symbol wird die Bildform beim Installieren der App auf einem Gerät vollständig gefüllt, ohne dass es zu einem Letterbox-Effekt kommt. [Informationen zu maskierbaren Manifestsymbolen.](#)

ZUSÄTZLICHE ELEMENTE ZUR MANUELLEN ÜBERPRÜFUNG (3)

Ausblenden

 Website funktioniert auf verschiedenen Browsern ^

Damit du möglichst viele Nutzer erreichen kannst, sollte deine Website mit allen gängigen Browsern kompatibel sein. [Weitere Informationen.](#)


 Seitenübergänge vermitteln nicht das Gefühl von übermäßigen Ladezeiten ^


Übergänge sollten sich auch bei einer langsamen Netzwerkverbindung schnell anfühlen. Dies ist entscheidend dafür, wie der Nutzer die Leistung wahrnimmt. [Weitere Informationen zu Seitenübergängen.](#)


 Jede Seite hat eine URL ^


Für die einzelnen Seiten sollten Deeplinks erstellt werden können. Achte darauf, dass die entsprechenden URLs eindeutig sind, sodass sich die Seiten in sozialen Netzwerken leichter teilen lassen. [Weitere Informationen zum Angeben von Deeplinks.](#)

Diese Prüfungen sind laut der grundlegenden [PWA-Checkliste](#) erforderlich, werden von Lighthouse jedoch nicht automatisch durchgeführt. Sie haben zwar keine Auswirkung auf deine Leistungsbewertung, aber es ist wichtig, sie manuell durchzuführen.


 Captured at 13. Juni 2023, 12:20
MESZ

 Desktopemulation with Lighthouse
10.1.1

 Einzelner Seitenaufbau

 Erster Seitenaufbau

 Benutzerdefinierte Drosselung

 Using Chromium 114.0.0.0 with
devtools

Generated by **Lighthouse** 10.1.1 | [Problem melden](#)

Lighthouse Report ohne Suche: Timespan

20/25

Leistung

7/8

Best Practices

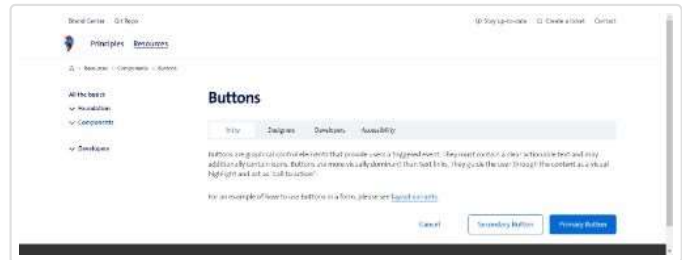
20/25

Leistung

▲ 0–49

■ 50–89

● 90–100



MESSWERTE

[Ansicht maximieren](#)

▲ Total Blocking Time

1.420 ms

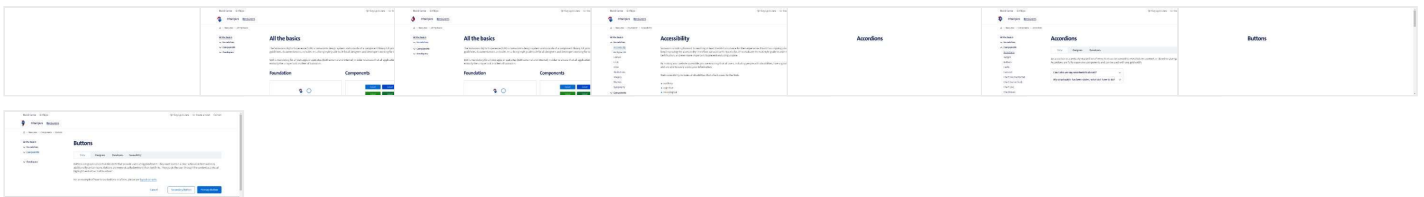
▲ Cumulative Layout Shift

0,383

● Interaction to Next Paint

60 ms

[Strukturkarte anzeigen](#)



Prüfungen anzeigen, die für folgende Messwerte relevant sind: [All](#) [TBT](#) [CLS](#) [INP](#)

DIAGNOSE

▲ Verwendet keine passiven Listener zur Verbesserung der Scrollleistung



- ▲ Aufwand für Hauptthread minimieren — 11,5 s
- Ausführungszeit von JavaScript reduzieren — 3,4 s
- Halte die Anfrageanzahl niedrig und die Übertragungsgröße gering — 373 Anfragen • 2.480 KiB
- Umfangreiche Layoutverschiebungen vermeiden — 1 Element gefunden
- Lange Hauptthread-Aufgaben vermeiden — 20 lange Aufgaben gefunden
- Nicht zusammengesetzte Animationen vermeiden — 8 animierte Elemente gefunden

Weitere Informationen zur Leistung deiner App findest du hier. Diese Angaben haben keinen [direkten Einfluss](#) auf die Leistungsbewertung.

BESTANDENE PRÜFUNGEN (20)

Anzeigen

■ 7/8

Best Practices

VERTRAUEN UND SICHERHEIT


- ▲ Verwendet nicht HTTPS — 2 unsichere Anfragen gefunden


BESTANDENE PRÜFUNGEN (7)


Anzeigen


NICHT ZUTREFFEND (1)


Anzeigen


 Captured at 13. Juni 2023, 12:32
MESZ

 Desktopemulation with Lighthouse
10.1.1

 Einzelner Seitenaufbau

 Erster Seitenaufbau

 Benutzerdefinierte Drosselung

 Using Chromium 114.0.0.0 with devtools

Generated by **Lighthouse** 10.1.1 | [Problem melden](#)

Lighthouse Report mit Suche: Timespan

20/25

Leistung

7/8

Best Practices

20/25

Leistung



▲ 0–49

■ 50–89

● 90–100

MESSWERTE

Ansicht maximieren

▲ Total Blocking Time
1.630 ms

● Cumulative Layout Shift
0

● Interaction to Next Paint
50 ms

 [Strukturkarte anzeigen](#)



Prüfungen anzeigen, die für folgende Messwerte relevant sind: [All](#) [TBT](#) [CLS](#) [INP](#)

DIAGNOSE

▲ Verwendet keine passiven Listener zur Verbesserung der Scrollleistung ^

Wenn du deine Event-Listener für Tipp- und Mausevents als passiv markierst, kannst du damit die Scrollleistung deiner Seite verbessern. [Weitere Informationen zur Verwendung von passiven Event-Listnern.](#)

Quelle
localhost 1st Party
./node_modules/@stencil/core/internal/client/index.js?app-data=conditional:3268:47

▲ Aufwand für Hauptthread minimieren — 13,2 s ^

Versuche, die Zeit für das Parsen, Kompilieren und Ausführen von JavaScript zu reduzieren. Die Bereitstellung kleinerer JS-Nutzlasten kann dabei helfen. [Informationen dazu, wie sich der Aufwand für den Hauptthread minimieren lässt](#) TBT

Kategorie	Zeitaufwand
Other	5.540 ms
Script Evaluation	4.662 ms
Style & Layout	1.764 ms
Rendering	689 ms
Script Parsing & Compilation	304 ms
Parse HTML & CSS	148 ms
Garbage Collection	122 ms

▲ Sehr große Netzwerknutzlasten vermeiden — Die Gesamtgröße war 4.569 KiB ^

Große Netzwerknutzlasten kosten Nutzer bares Geld und hängen eng mit langen Ladezeiten zusammen. [Informationen zum Verringern der Nutzlastgröße.](#)

URL	Übertragungsgröße
localhost 1st Party	2.635,0 KiB
...buttons/fullwidth-desktop@2x.png (localhost)	709,4 KiB
...tables/Group-18@2x-6.jpg (localhost)	262,1 KiB
...icons/03_container_alignment_1_.png (localhost)	244,7 KiB
...buttons/fill-desktop@2x.png (localhost)	243,0 KiB
...webcomponents/p-fb9f0d30.entry.js (localhost)	243,0 KiB
/foundation_-_icons.html (localhost)	227,4 KiB
...buttons/fullwidth-mobile@2x.png (localhost)	219,9 KiB

URL	Übertragungsgröße
...icons/07_available_sizes_.png (localhost)	168,3 KiB
...tables/Group-21@2x-5.jpg (localhost)	163,8 KiB
...buttons/responsive-fill-desktop@2x.png (localhost)	153,5 KiB

▲ Ausführungszeit von JavaScript reduzieren — 4,9 s ^

Versuche, die Zeit für das Parsen, Kompilieren und Ausführen von JavaScript zu reduzieren. Die Bereitstellung kleinerer JS-Nutzlasten kann dabei helfen. [Informationen zum Reduzieren der JavaScript-Ausführungszeit.](#) TBT

Drittanbieter-Ressourcen anzeigen (3)

URL	CPU-Zeit insgesamt	Skriptauswertung	Parsen von Skripten
localhost 1st Party	8.025 ms	2.744 ms	104 ms
/foundation_-_icons.html (localhost)	3.406 ms	16 ms	2 ms
...webcomponents/p-f772284e.js (localhost)	2.204 ms	1.559 ms	3 ms
...webcomponents/p-53dbf7d7.js (localhost)	1.115 ms	278 ms	5 ms
/foundation_-_accessibility.html (localhost)	156 ms	148 ms	1 ms
...js/sdx.min.js (localhost)	150 ms	73 ms	29 ms
/dist/search-integrator.js (localhost)	142 ms	119 ms	23 ms
/components_-_accordions.html (localhost)	125 ms	118 ms	1 ms
...webcomponents/p-fb9f0d30.entry.js (localhost)	122 ms	11 ms	25 ms
/03_developers_-_getting_started-with_sections_and_breadcr...html (localhost)	115 ms	109 ms	1 ms
/components_-_buttons.html (localhost)	98 ms	86 ms	1 ms
http://localhost:3000	74 ms	65 ms	1 ms
/components_-_tables.html (localhost)	70 ms	60 ms	1 ms
...webcomponents/p-c31cad0b.entry.js (localhost)	67 ms	19 ms	8 ms
...webcomponents/p-0e7ae7e8.entry.js (localhost)	67 ms	30 ms	5 ms
/developers_-_css_variables.html (localhost)	61 ms	55 ms	1 ms
...css/sdx.min.css (localhost)	52 ms	0 ms	0 ms

URL	CPU-Zeit insgesamt	Skriptauswertung	Parsen von Skripten
Unattributable	3.274 ms	374 ms	0 ms
Unattributable	3.274 ms	374 ms	0 ms
Adobe Tag Manager Tag-Manager	1.155 ms	1.004 ms	96 ms
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	1.046 ms	902 ms	94 ms
...d199e5db12b9/RC079aeec...-source.min.js (assets.adobedtm.com)	108 ms	101 ms	2 ms
Tealium Tag-Manager	562 ms	481 ms	76 ms
...dev/utag.js (tags.tiqcdn.com)	562 ms	481 ms	76 ms

○ Halte die Anfrageanzahl niedrig und die Übertragungsgröße gering — 420 Anfragen • 4.566 KiB ^

Füge zum Einrichten von Budgets für die Anzahl und Größe von Seitenressourcen eine budget.json-Datei hinzu. [Weitere Informationen zu Leistungsbudgets.](#)

Ressourcentyp	Anfragen	Übertragungsgröße
Gesamt	420,0	4.566,5 KiB
Bild	48,0	3.068,8 KiB
Skript	268,0	1.002,9 KiB
Dokument	8,0	418,0 KiB
Schriftart	52,0	59,6 KiB
Sonstige	30,0	13,7 KiB
Stylesheet	14,0	3,6 KiB
Medien	0,0	0,0 KiB
Drittanbieter	110,0	676,1 KiB

○ Lange Hauptthread-Aufgaben vermeiden — 20 lange Aufgaben gefunden ^

Listet die längsten Aufgaben im Hauptthread auf. Das ist nützlich, um die wichtigsten Ursachen für die Eingabeverzögerungen zu ermitteln. [Informationen dazu, wie sich lange Aufgaben im Hauptthread vermeiden lassen](#) TBT

URL	Beginn	Dauer
localhost 1st Party		1.678 ms
/foundation_-_icons.html (localhost)	49.348 ms	192 ms
...webcomponents/p-f772284e.js (localhost)	49.966 ms	148 ms
...webcomponents/p-53dbf7d7.js (localhost)	50.147 ms	138 ms
...webcomponents/p-f772284e.js (localhost)	42.520 ms	136 ms
...webcomponents/p-53dbf7d7.js (localhost)	26.044 ms	116 ms
/foundation_-_icons.html (localhost)	19.726 ms	109 ms
...webcomponents/p-53dbf7d7.js (localhost)	37.307 ms	109 ms
...webcomponents/p-53dbf7d7.js (localhost)	12.657 ms	108 ms
/foundation_-_icons.html (localhost)	30.797 ms	95 ms
/foundation_-_icons.html (localhost)	12.177 ms	92 ms
/foundation_-_icons.html (localhost)	36.757 ms	90 ms
...webcomponents/p-53dbf7d7.js (localhost)	20.122 ms	89 ms
...webcomponents/p-53dbf7d7.js (localhost)	20.238 ms	88 ms
/foundation_-_icons.html (localhost)	49.247 ms	87 ms
...webcomponents/p-53dbf7d7.js (localhost)	5.243 ms	82 ms
Unattributable		619 ms
Unattributable	48.985 ms	167 ms
Unattributable	41.935 ms	145 ms
Unattributable	36.540 ms	131 ms
Unattributable	30.602 ms	99 ms
Unattributable	25.307 ms	77 ms

○ Nicht zusammengesetzte Animationen vermeiden — 6 animierte Elemente gefunden ^

Nicht zusammengesetzte Animationen werden eventuell nicht richtig gerendert und können den CLS-Wert erhöhen. [Informationen dazu, wie sich nicht zusammengefasste Animationen vermeiden lassen](#) CLS

Element		Name
	<p>Intro</p> <pre></pre>	
	<p>Nicht unterstützte CSS-Property: visibility</p>	<p>visibility</p>
	<p>Developers</p> <pre></pre>	
	<p>Nicht unterstützte CSS-Property: visibility</p>	<p>visibility</p>
	<p>Nicht unterstützte CSS-Property: color</p>	<p>color</p>
	<p>Nicht unterstützte CSS-Property: color</p>	<p>color</p>
	<p>copy link</p> <pre><button class="clickable button-reset transparent" href="" type="button"></pre>	
	<p>Nicht unterstützte CSS-Property: visibility</p>	<p>visibility</p>
	<p>Accessibility</p> <pre></pre>	
	<p>Nicht unterstützte CSS-Property: visibility</p>	<p>visibility</p>
	<p>Nicht unterstützte CSS-Property: color</p>	<p>color</p>
	<p>Nicht unterstützte CSS-Property: color</p>	<p>color</p>
	<p>Designers</p> <pre></pre>	
	<p>Nicht unterstützte CSS-Property: visibility</p>	<p>visibility</p>
	<p>ul.list > li.item > a.link > span.img</p> <pre></pre>	<p>repeatingAnimation</p>
	<p>Nicht unterstützte CSS-Properties: background-position-x, background-position-y</p>	<p>repeatingAnimation</p>

Element	Name
Nicht unterstützte CSS-Properties: background-position-x, background-position-y	initialAnimation

Weitere Informationen zur Leistung deiner App findest du hier. Diese Angaben haben keinen [direkten Einfluss](#) auf die Leistungsbewertung.

BESTANDENE PRÜFUNGEN (20)

Ausblenden

● Bilder richtig dimensionieren ^

Stelle Bilder bereit, die eine angemessene Größe haben, um mobile Daten zu sparen und die Ladezeit zu verbessern. [Weitere Informationen.](#)

● CSS komprimieren ^

Durch das Reduzieren von CSS-Dateien lassen sich Netzwerklasten senken. [Informationen zum Reduzieren von CSS.](#)

● JavaScript komprimieren ^

Durch die Komprimierung von JavaScript-Dateien können Nutzlastgrößen und die Zeit zum Parsen von Skripts reduziert werden. [Informationen zum Reduzieren von JavaScript.](#)

● Reduziere nicht verwendete CSS — Mögliche Einsparung von 11 KiB ^

Du kannst ungültige Regeln in Stylesheets reduzieren und CSS-Code zurückstellen, der nicht für ohne Scrollen sichtbare Inhalte („above the fold“) verwendet wird, um den Datenverbrauch durch Netzwerkaktivität zu senken. [Informationen zum Reduzieren von nicht verwendetem CSS-Code.](#)

URL	Übertragungsgröße	Mögliche Einsparungen
:root { --sdx-header-height-mobile: 72px; --sdx-header-height-mobile-with-breadcrumbs: 120px; --sdx...	11,2 KiB	11,0 KiB

● Reduziere nicht verwendetes JavaScript — Mögliche Einsparung von 169 KiB ^

Um den Datenverbrauch durch Netzwerkaktivität zu senken, kannst du nicht verwendetes JavaScript reduzieren und das Laden von Skripts zurückstellen, bis sie benötigt werden.

[Informationen zum Reduzieren von nicht verwendetem JavaScript.](#)
 Drittanbieter-Ressourcen anzeigen (1)

URL	Übertragungsgröße	Mögliche Einsparungen
localhost 1st Party	243,0 KiB	147,1 KiB
...webcomponents/p-fb9f0d30.entry.js (localhost)	243,0 KiB	147,1 KiB
Adobe Tag Manager Tag-Manager	82,2 KiB	22,1 KiB
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	82,2 KiB	22,1 KiB

● Bilder effizient codieren — Mögliche Einsparung von 13 KiB ^

Warnungen:

- Unable to decode ...buttons/fullwidth-desktop@2x.png
- Unable to decode ...tables/Group-18@2x-6.jpg
- Unable to decode ...buttons/fill-desktop@2x.png
- Unable to decode ...buttons/fullwidth-mobile@2x.png
- Unable to decode ...tables/Group-21@2x-5.jpg
- Unable to decode ...buttons/responsive-fill-desktop@2x.png
- Unable to locate resource ...tables/Group-20@2x-4.jpg

Optimierte Bilder werden schneller geladen und verbrauchen weniger mobile Daten. [Informationen zum effizienten Codieren von Bildern.](#)

URL	Größe der Ressource	Mögliche Einsparungen
localhost 1st Party	66,7 KiB	13,2 KiB
Example main navigation <pre></pre>	... header/SDX%20Header%20Mobile.jpg (local host)	66,7 KiB 13,2 KiB

● Bilder in modernen Formaten bereitstellen — Mögliche Einsparung von 507 KiB ^

Warnungen:

- Unable to decode ...buttons/fullwidth-desktop@2x.png

- Unable to decode ...tables/Group-18@2x-6.jpg
- Unable to decode ...buttons/fill-desktop@2x.png
- Unable to decode ...buttons/fullwidth-mobile@2x.png
- Unable to decode ...tables/Group-21@2x-5.jpg
- Unable to decode ...buttons/responsive-fill-desktop@2x.png
- Unable to locate resource ...buttons/fill-mobile@2x.png
- Unable to locate resource ...buttons/regular-desktop@2x.png
- Unable to locate resource ...buttons/responsive-fill-mobile@2x.png
- Unable to locate resource ...tables/Group-20@2x-4.jpg
- Unable to locate resource ...tables/Group-12@2x-8.png
- Unable to locate resource ...tables/Group-14@2x-6.png
- Unable to locate resource ...buttons/regular-mobile@2x.png
- Unable to locate resource ...samples/angebote-vivo-xl.png
- Unable to locate resource ...samples/angebote-vivo-s.png

Bildformate wie WebP und AVIF bieten oft eine bessere Komprimierung als PNG oder JPEG, wodurch sie schneller heruntergeladen werden und weniger Daten verbrauchen. [Weitere Informationen zu modernen Bildformaten.](#)

URL	Größe der Ressource	Mögliche Einsparungen
localhost 1st Party	702,6 KiB	507,0 KiB
Example of icons alignment <pre></pre>	... icons/03_container_alignment_1.png (localhost)	244,4 KiB 183,3 KiB
Shows icon sizes <pre></pre>	...icons/07_available_sizes.png (localhost)	168,0 KiB 118,1 KiB
Example of a maximum icon size <pre><img class="image image-frame-light bg-sc-white"</pre>	...icons/06_56x56px.png (localhost)	118,7 KiB 96,8 KiB

URL	Größe der Ressource	Mögliche Einsparungen
<pre>src="/img/design/icons/06_56x56px_.png" alt="Example of a maximum icon size"></pre> <p>Example of a minimum icon size</p> <pre></pre> <p>Example main navigation</p> <pre></pre>	<p>...icons/05_16x16px_.png (localhost)</p> <p>header/SDX%20Header%20Mobile.jpg (localhost)</p>	<p>104,8 KiB</p> <p>66,7 KiB</p> <p>77,7 KiB</p> <p>31,1 KiB</p>

● **Textkomprimierung aktivieren** — Mögliche Einsparung von 213 KiB ^

Textbasierte Ressourcen sollten komprimiert (gzip, Deflate oder Brotli) ausgeliefert werden, um die Netzwerkbytes insgesamt zu minimieren. [Weitere Informationen zur Textkomprimierung.](#)

URL	Übertragungsgröße	Mögliche Einsparungen
localhost 1st Party	227,1 KiB	213,5 KiB
/foundation_-_icons.html (localhost)	227,1 KiB	213,5 KiB

● **Verwende HTTP/2** ^

HTTP/2 bietet gegenüber HTTP/1.1 viele Vorteile, wie z. B. binäre Header und Multiplexverfahren. [Weitere Informationen zu HTTP/2.](#)

● **Videoformate für animierte Inhalte verwenden** ^

Große GIF-Dateien sind nur bedingt für die Auslieferung animierter Inhalte geeignet. Du kannst stattdessen MPEG4- oder WebM-Videos für Animationen und PNG oder WebP für statische Bilder

verwenden und so die Netzwerkbytes reduzieren. [Weitere Informationen zu effizienten Videoformaten](#)

● In JavaScript-Bundles doppelt vorhandene Module entfernen ^

Entferne große, doppelt vorhandene JavaScript-Module aus Bundles, um unnötige Datenübertragungen im Netzwerk zu reduzieren. TBT

● Verhindern, dass in modernen Browsern veraltetes JavaScript bereitgestellt wird — Mögliche Einsparung von 0 KiB ^

Dank Polyfills und Transformationen können veraltete Browser die neuen JavaScript-Funktionen nutzen. Bei modernen Browsern hingegen sind viele davon nicht erforderlich. Für dein JavaScript-Bundle solltest du eine moderne Skriptimplementierungsstrategie unter Verwendung der module/nomodule-Funktionserkennung einsetzen. So kannst du einerseits den an moderne Browser übermittelten Code auf das Nötige reduzieren und gleichzeitig veraltete Browser bestmöglich unterstützen. [Informationen zur Verwendung modernen JavaScripts](#) TBT

URL	Mögliche Einsparungen
Adobe Tag Manager Tag-Manager	0,1 KiB
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	0,1 KiB
...4cf4f6043663/launch-0c232d70d3dd-development.min.js:7:1084 (assets.adobedtm.com)	@babel/plugin-transform-classes
Tealium Tag-Manager	0,0 KiB
...dev/utag.js (tags.tiqcdn.com)	0,0 KiB
...dev/utag.js:1475:14 (tags.tiqcdn.com)	Object.keys

● Verwendet eine effiziente Cache-Richtlinie für statische Inhalte — 69 Ressourcen gefunden ^

Eine lange Verweildauer im Cache kann wiederholte Besuche deiner Seite beschleunigen. [Weitere Informationen zu effizienten Cache-Richtlinien.](#)

URL	Cache-TTL	Übertragungsgröße
Tealium Tag-Manager		3 KiB
...tiqapp/utag.v.js?a=swisscom/main/202...&nocookie=1 (tags.tiqcdn.com)	5 Min.	0 KiB

URL	Cache-TTL	Übertragungsgröße
/tr?id=865...&ev=Adobe-Aud... &cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB
/tr?id=865...&ev=Adobe-Aud... &cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB
/tr?id=865...&ev=Adobe-Aud... &cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB
/tr?id=865...&ev=Adobe-Aud... &cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB
/tr?id=865...&ev=Adobe-Aud... &cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB
/tr?id=865...&ev=Adobe-Aud... &cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB
/tr?id=865...&ev=Adobe-Aud... &cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB
/tr?id=865...&ev=Adobe-Aud... &cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB
/tr?id=865...&ev=Adobe-Aud... &cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB
/tr?id=865...&ev=Adobe-Aud... &cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB
/tr?id=865...&ev=Adobe-Aud... &cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB
/tr?id=865...&ev=Adobe-Aud... &cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB
/tr?id=865...&ev=Adobe-Aud... &cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB
/tr?id=865...&ev=Adobe-Aud... &cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB
Adobe Tag Manager Tag-Manager		0 KiB
...d199e5db12b9/RC079aeec...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC079aeec...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC079aeec...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC079aeec...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB

URL	Cache-TTL	Übertragungsgröße
...d199e5db12b9/RC079aeec...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC079aeec...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC079aeec...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC079aeec...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC6015071...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC6015071...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC6015071...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC6015071...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC6015071...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC6015071...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC6015071...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC6015071...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC6015071...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC6015071...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC777e9ba...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC777e9ba...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC777e9ba...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC777e9ba...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC777e9ba...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC777e9ba...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC777e9ba...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC777e9ba...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC777e9ba...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC777e9ba...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC991f58f...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC991f58f...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC991f58f...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC991f58f...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC991f58f...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC991f58f...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC991f58f...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC991f58f...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC991f58f...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC991f58f...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB

URL	Cache-TTL	Übertragungsgröße
...d199e5db12b9/RC991f58f...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB

○ Markierungen und Messungen für das Nutzertiming ^

Du kannst die User Timing API in deine App einbinden. Damit lässt sich die Leistung der App während wichtiger Nutzerinteraktionen in der Praxis messen. [Weitere Informationen zu User Timing-Markierungen.](#)

● Drittanbiernutzung minimieren — Code von Drittanbietern hat den Hauptthread 10 ms lang blockiert ^

Code von Drittanbietern kann die Ladegeschwindigkeit erheblich beeinträchtigen. Beschränke die Zahl redundanter Drittanbieter und versuche, solchen Code erst zu laden, nachdem die Seite vollständig geladen wurde. [Informationen dazu, wie sich die Auswirkungen von Drittanbietercode minimieren lassen.](#) TBT

Drittanbieter	Übertragungsgröße	Dauer der Blockierung des Hauptthreads
Adobe Tag Manager Tag-Manager	658 KiB	6 ms
...4cf6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	658 KiB	6 ms
adobedc.net	14 KiB	0 ms
Tealium Tag-Manager	4 KiB	0 ms
Facebook Social	0 KiB	0 ms

○ Umfangreiche Layoutverschiebungen vermeiden ^

Diese DOM-Elemente tragen am stärksten zur CLS der Seite bei. [Informationen zur Verbesserung der CLS](#) CLS

● Verwendet kein `document.write()` ^

Für Nutzer mit langsamen Verbindungen können externe Skripts, die dynamisch über `document.write()` eingefügt werden, den Seitenaufbau um einige Sekunden verzögern. [Informationen zum Vermeiden von „document.write\(\)“.](#)

● Bildelemente haben eine explizite `width` und `height` ^

Lege eine explizite Breite und Höhe für Bildelemente fest, um Layoutverschiebungen zu reduzieren und den CLS-Wert zu verbessern. [Informationen zum Festlegen von Bildabmessungen](#)

CLS

● Minimiert den Aufwand während der Hauptinteraktion — 50 ms für keydown-Ereignis benötigt ^

Dies sind die Aufgaben zur Blockierung des Threads, die beim Erfassen des Messwerts „Interaction to Next Paint“ ausgeführt werden. [Weitere Informationen zum Messwert „Interaction to Next Paint“](#). INP

Phase	Total time	Script evaluation	Style & Layout	Rendering
Eingabeverzögerung	0 ms			
Verarbeitungszeit	1 ms			
Verzögerung bei der Präsentation	47 ms			
http://localhost:3000	17 ms	17 ms	0 ms	0 ms
/foundation_-_icons.html (localhost)	3 ms	0 ms	2 ms	0 ms
...webcomponents/p-f772284e.js (localhost)	2 ms	1 ms	0 ms	0 ms

● Seite hat die Wiederherstellung des Back-Forward-Caches nicht verhindert ^

Viele Bedienvorgänge werden ausgeführt, indem eine vorherige Seite aufgerufen oder zur nächsten gegangen wird. Der Back-Forward-Cache (bfcache) kann diese Vorgänge beschleunigen. [Weitere Informationen zum bfcache](#)

■ 7/8

Best Practices

VERTRAUEN UND SICHERHEIT

▲ Verwendet nicht HTTPS — 2 unsichere Anfragen gefunden ^

Alle Websites sollten durch HTTPS geschützt werden – selbst wenn sie keine sensiblen Daten enthalten. Auch [gemischte Inhalte](#), bei denen einige Ressourcen über HTTP geladen werden, obwohl die ursprüngliche Anfrage über HTTPS gestellt wurde, sind zu vermeiden. HTTPS verhindert, dass andere die Website manipulieren oder die Kommunikation zwischen deiner App und deinen Nutzern mitverfolgen können, und ist eine Voraussetzung für HTTP/2 und viele neue Webplattform-APIs. [Weitere Informationen zu HTTPS.](#)

Unsichere URL	Lösung anfordern
Tealium <input type="text" value="Tag-Manager"/>	
...dev/utag.js (tags.tiqcdn.com)	Zugelassen
...dev/utag.js (tags.tiqcdn.com)	Zugelassen

BESTANDENE PRÜFUNGEN (7)

Ausblenden

● Zeigt Bilder mit einem korrekten Seitenverhältnis an ^

Die Bildgröße sollte dem natürlichen Seitenverhältnis entsprechen. [Weitere Informationen zum Seitenverhältnis von Bildern.](#)

● Stellt Bilder mit angemessener Auflösung bereit ^

Die originalen Abmessungen eines Bildes sollten proportional zu der Displaygröße und dem Pixel-Verhältnis sein, damit das Bild optimal angezeigt wird. [Weitere Informationen zu responsiven Bildern.](#)

● Keine unload-Ereignis-Listener gefunden ^

Das unload-Ereignis wird nicht zuverlässig ausgelöst. Wenn der Listener darauf wartet, kann das Browseroptimierungen wie den Back-Forward-Cache beeinträchtigen. Bitte verwende stattdessen pagehide- oder visibilitychange-Ereignisse. [Weitere Informationen zum Entfernen von Event-Listnern](#)

● Vermeidet veraltete APIs ^

Verworfen APIs werden aus dem Browser entfernt. [Weitere Informationen zu verworfenen APIs.](#)

● Es wurden keine Browserfehler in der Konsole protokolliert ^

In der Konsole protokollierte Fehler weisen auf ungelöste Probleme hin. Sie können durch fehlgeschlagene Netzwerkanfragen und andere Browserprobleme verursacht werden. [Weitere Informationen](#)

● Keine Probleme im [Issues](#)-Bereich der Chrome-Entwicklertools ^

Im Issues-Bereich der Chrome-Entwicklertools wurden ungelöste Probleme protokolliert. Sie können durch fehlgeschlagene Netzwerkanfragen, unzureichende Sicherheitsmaßnahmen und andere Browser-Probleme verursacht sein. Öffne den Bereich mit den Problemen in Chrome-Entwicklertools, um weitere Details zu den einzelnen Problemen aufzurufen.

● Seite hat gültige Quellzuordnungen ^

Quellzuordnungen übersetzen reduzierten Code in den ursprünglichen Quellcode. Dies hilft Entwicklern beim Debugging in der Produktionsphase. Zusätzlich kann Lighthouse weitere Informationen liefern. Wir empfehlen, Quellzuordnungen bereitzustellen, um diese Vorteile zu nutzen. [Weitere Informationen zu Quellzuordnungen.](#)

URL	Zuordnungs-URL
localhost 1st Party	
<code>/search-component/search-component.js (localhost)</code>	<code>/search-component/search-component.js.map (localhost)</code>
Error: Failed fetching source map (404)	
<code>/search-component/search-component.js (localhost)</code>	<code>/search-component/search-component.js.map (localhost)</code>
Error: Failed fetching source map (404)	
<code>/search-component/search-component.js (localhost)</code>	<code>/search-component/search-component.js.map (localhost)</code>
Error: Failed fetching source map (404)	
<code>/search-component/search-component.js (localhost)</code>	<code>/search-component/search-component.js.map (localhost)</code>
Error: Failed fetching source map (404)	
<code>/search-component/search-component.js (localhost)</code>	<code>/search-component/search-component.js.map (localhost)</code>
Error: Failed fetching source map (404)	

URL	Zuordnungs-URL
<code>/search-component/search-component.js (localhost)</code>	<code>/search-component/search-component.js.map (localhost)</code>
Error: Failed fetching source map (404)	
<code>/search-component/search-component.js (localhost)</code>	<code>/search-component/search-component.js.map (localhost)</code>
Error: Failed fetching source map (404)	
<code>/search-component/search-component.js (localhost)</code>	<code>/search-component/search-component.js.map (localhost)</code>
Error: Failed fetching source map (404)	
<code>...webcomponents/webcomponents.esm.js (localhost)</code>	<code>webcomponents/webcomponents.esm.js.map (localhost)</code>
Error: Waiting for the end of the IO stream exceeded the allotted time.	
<code>...webcomponents/webcomponents.esm.js (localhost)</code>	<code>webcomponents/webcomponents.esm.js.map (localhost)</code>
Error: Waiting for the end of the IO stream exceeded the allotted time.	
<code>...webcomponents/webcomponents.esm.js (localhost)</code>	<code>webcomponents/webcomponents.esm.js.map (localhost)</code>
Error: Waiting for the end of the IO stream exceeded the allotted time.	
<code>...webcomponents/webcomponents.esm.js (localhost)</code>	<code>webcomponents/webcomponents.esm.js.map (localhost)</code>
Error: Waiting for the end of the IO stream exceeded the allotted time.	
<code>...webcomponents/p-fb9f0d30.entry.js (localhost)</code>	<code>...webcomponents/p-fb9f0d30.entry.js.map (localhost)</code>
Error: Waiting for the end of the IO stream exceeded the allotted time.	
<code>...webcomponents/p-fb9f0d30.entry.js (localhost)</code>	<code>...webcomponents/p-fb9f0d30.entry.js.map (localhost)</code>
Error: Waiting for the end of the IO stream exceeded the allotted time.	
<code>...webcomponents/p-fb9f0d30.entry.js (localhost)</code>	<code>...webcomponents/p-</code>

URL	Zuordnungs-URL
<p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>fb9f0d30.entry.js.map (localhost)</p>
<p>...webcomponents/p-fb9f0d30.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-fb9f0d30.entry.js.map (localhost)</p>
<p>...webcomponents/p-fb9f0d30.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-fb9f0d30.entry.js.map (localhost)</p>
<p>...webcomponents/p-f772284e.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-f772284e.js.map (localhost)</p>
<p>...webcomponents/p-f772284e.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-f772284e.js.map (localhost)</p>
<p>...webcomponents/p-f772284e.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-f772284e.js.map (localhost)</p>
<p>...webcomponents/p-f772284e.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-f772284e.js.map (localhost)</p>
<p>...webcomponents/p-f772284e.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-f772284e.js.map (localhost)</p>
<p>...webcomponents/p-e7e1c19a.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-e7e1c19a.entry.js.map (localhost)</p>
<p>...webcomponents/p-dd3218d3.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-dd3218d3.entry.js.map (localhost)</p>
<p>...webcomponents/p-dd3218d3.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-dd3218d3.entry.js.map (localhost)</p>

URL	Zuordnungs-URL
exceeded the allotted time.	
...webcomponents/p-dd3218d3.entry.js (localhost)	...webcomponents/p-dd3218d3.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-dd3218d3.entry.js (localhost)	...webcomponents/p-dd3218d3.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-dd3218d3.entry.js (localhost)	...webcomponents/p-dd3218d3.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-d831bc11.entry.js (localhost)	...webcomponents/p-d831bc11.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-d831bc11.entry.js (localhost)	...webcomponents/p-d831bc11.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-d831bc11.entry.js (localhost)	...webcomponents/p-d831bc11.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-d831bc11.entry.js (localhost)	...webcomponents/p-d831bc11.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-d831bc11.entry.js (localhost)	...webcomponents/p-d831bc11.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-c31cad0b.entry.js (localhost)	...webcomponents/p-c31cad0b.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-c31cad0b.entry.js (localhost)	...webcomponents/p-c31cad0b.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-c31cad0b.entry.js (localhost)	...webcomponents/p-

URL	Zuordnungs-URL
<p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>c31cad0b.entry.js.map (localhost)</p>
<p>...webcomponents/p-c31cad0b.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-c31cad0b.entry.js.map (localhost)</p>
<p>...webcomponents/p-bc913fa4.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-bc913fa4.entry.js.map (localhost)</p>
<p>...webcomponents/p-bc913fa4.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-bc913fa4.entry.js.map (localhost)</p>
<p>...webcomponents/p-bc913fa4.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-bc913fa4.entry.js.map (localhost)</p>
<p>...webcomponents/p-bc913fa4.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-bc913fa4.entry.js.map (localhost)</p>
<p>...webcomponents/p-bc913fa4.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-bc913fa4.entry.js.map (localhost)</p>
<p>...webcomponents/p-bc913fa4.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-bc913fa4.entry.js.map (localhost)</p>
<p>...webcomponents/p-bbcf6480.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-bbcf6480.entry.js.map (localhost)</p>
<p>...webcomponents/p-bbcf6480.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-bbcf6480.entry.js.map (localhost)</p>
<p>...webcomponents/p-bbcf6480.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-bbcf6480.entry.js.map (localhost)</p>
<p>...webcomponents/p-bbcf6480.entry.js (localhost)</p> <p>Error: Waiting for the end of the IO stream exceeded the allotted time.</p>	<p>...webcomponents/p-bbcf6480.entry.js.map (localhost)</p>

URL	Zuordnungs-URL
exceeded the allotted time.	
...webcomponents/p-bbcf6480.entry.js (localhost)	...webcomponents/p-bbcf6480.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-b773df8a.entry.js (localhost)	...webcomponents/p-b773df8a.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-b773df8a.entry.js (localhost)	...webcomponents/p-b773df8a.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-b773df8a.entry.js (localhost)	...webcomponents/p-b773df8a.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-a5d22636.entry.js (localhost)	...webcomponents/p-a5d22636.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-a5d22636.entry.js (localhost)	...webcomponents/p-a5d22636.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-a5d22636.entry.js (localhost)	...webcomponents/p-a5d22636.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-a5d22636.entry.js (localhost)	...webcomponents/p-a5d22636.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-7d1868eb.entry.js (localhost)	...webcomponents/p-7d1868eb.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-7d1868eb.entry.js (localhost)	...webcomponents/p-

URL	Zuordnungs-URL
	7d1868eb.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-7d1868eb.entry.js (localhost)	...webcomponents/p-7d1868eb.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-7d1868eb.entry.js (localhost)	...webcomponents/p-7d1868eb.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-7d1868eb.entry.js (localhost)	...webcomponents/p-7d1868eb.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-6a2de715.entry.js (localhost)	...webcomponents/p-6a2de715.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-6a2de715.entry.js (localhost)	...webcomponents/p-6a2de715.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-6a2de715.entry.js (localhost)	...webcomponents/p-6a2de715.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-6a2de715.entry.js (localhost)	...webcomponents/p-6a2de715.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-6a2de715.entry.js (localhost)	...webcomponents/p-6a2de715.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-62030357.js (localhost)	...webcomponents/p-62030357.js.map (localhost)
Error: Waiting for the end of the IO stream	

URL	Zuordnungs-URL
exceeded the allotted time.	
...webcomponents/p-62030357.js (localhost)	...webcomponents/p-62030357.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-62030357.js (localhost)	...webcomponents/p-62030357.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-62030357.js (localhost)	...webcomponents/p-62030357.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-53dbf7d7.js (localhost)	...webcomponents/p-53dbf7d7.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-53dbf7d7.js (localhost)	...webcomponents/p-53dbf7d7.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-53dbf7d7.js (localhost)	...webcomponents/p-53dbf7d7.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-53dbf7d7.js (localhost)	...webcomponents/p-53dbf7d7.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-53dbf7d7.js (localhost)	...webcomponents/p-53dbf7d7.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-3f4649f5.js (localhost)	...webcomponents/p-3f4649f5.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-3f4649f5.js (localhost)	...webcomponents/p-3f4649f5.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-3f4649f5.js (localhost)	...webcomponents/p-3f4649f5.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	

URL	Zuordnungs-URL
<code>...webcomponents/p-3f4649f5.js (localhost)</code> Error: Waiting for the end of the IO stream exceeded the allotted time.	<code>...webcomponents/p-3f4649f5.js.map (localhost)</code>
<code>...webcomponents/p-3ed4d64e.entry.js (localhost)</code> Error: Waiting for the end of the IO stream exceeded the allotted time.	<code>...webcomponents/p-3ed4d64e.entry.js.map (localhost)</code>
<code>...webcomponents/p-3ed4d64e.entry.js (localhost)</code> Error: Waiting for the end of the IO stream exceeded the allotted time.	<code>...webcomponents/p-3ed4d64e.entry.js.map (localhost)</code>
<code>...webcomponents/p-3ed4d64e.entry.js (localhost)</code> Error: Waiting for the end of the IO stream exceeded the allotted time.	<code>...webcomponents/p-3ed4d64e.entry.js.map (localhost)</code>
<code>...webcomponents/p-3ed4d64e.entry.js (localhost)</code> Error: Waiting for the end of the IO stream exceeded the allotted time.	<code>...webcomponents/p-3ed4d64e.entry.js.map (localhost)</code>
<code>...webcomponents/p-3ed4d64e.entry.js (localhost)</code> Error: Waiting for the end of the IO stream exceeded the allotted time.	<code>...webcomponents/p-3ed4d64e.entry.js.map (localhost)</code>
<code>...webcomponents/p-3ed4d64e.entry.js (localhost)</code> Error: Waiting for the end of the IO stream exceeded the allotted time.	<code>...webcomponents/p-3ed4d64e.entry.js.map (localhost)</code>
<code>...webcomponents/p-2b739f89.entry.js (localhost)</code> Error: Waiting for the end of the IO stream exceeded the allotted time.	<code>...webcomponents/p-2b739f89.entry.js.map (localhost)</code>
<code>...webcomponents/p-299c3b04.js (localhost)</code> Error: Waiting for the end of the IO stream exceeded the allotted time.	<code>...webcomponents/p-299c3b04.js.map (localhost)</code>
<code>...webcomponents/p-299c3b04.js (localhost)</code> Error: Waiting for the end of the IO stream exceeded the allotted time.	<code>...webcomponents/p-299c3b04.js.map (localhost)</code>
<code>...webcomponents/p-299c3b04.js (localhost)</code> Error: Waiting for the end of the IO stream exceeded the allotted time.	<code>...webcomponents/p-299c3b04.js.map (localhost)</code>

URL	Zuordnungs-URL
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-299c3b04.js (localhost)	...webcomponents/p-299c3b04.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-112455b1.js (localhost)	...webcomponents/p-112455b1.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-112455b1.js (localhost)	...webcomponents/p-112455b1.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-112455b1.js (localhost)	...webcomponents/p-112455b1.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-112455b1.js (localhost)	...webcomponents/p-112455b1.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-112455b1.js (localhost)	...webcomponents/p-112455b1.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-0e7aefe8.entry.js (localhost)	...webcomponents/p-0e7aefe8.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-0e7aefe8.entry.js (localhost)	...webcomponents/p-0e7aefe8.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-0e7aefe8.entry.js (localhost)	...webcomponents/p-0e7aefe8.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-0e7aefe8.entry.js (localhost)	...webcomponents/p-0e7aefe8.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-0e7aefe8.entry.js (localhost)	...webcomponents/p-0e7aefe8.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	

URL	Zuordnungs-URL
...webcomponents/p-0e7ae7e8.entry.js (localhost)	...webcomponents/p-0e7ae7e8.entry.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-057a4d55.js (localhost)	...webcomponents/p-057a4d55.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-057a4d55.js (localhost)	...webcomponents/p-057a4d55.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-057a4d55.js (localhost)	...webcomponents/p-057a4d55.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/p-057a4d55.js (localhost)	...webcomponents/p-057a4d55.js.map (localhost)
Error: Waiting for the end of the IO stream exceeded the allotted time.	
...webcomponents/webcomponents.esm.js (localhost)	...webcomponents/webcomponents.esm.js.map (localhost)
...webcomponents/webcomponents.esm.js (localhost)	...webcomponents/webcomponents.esm.js.map (localhost)
...webcomponents/webcomponents.esm.js (localhost)	...webcomponents/webcomponents.esm.js.map (localhost)
...webcomponents/webcomponents.esm.js (localhost)	...webcomponents/webcomponents.esm.js.map (localhost)
...webcomponents/p-fb9f0d30.entry.js (localhost)	...webcomponents/p-fb9f0d30.entry.js.map (localhost)
...webcomponents/p-fb9f0d30.entry.js (localhost)	...webcomponents/p-fb9f0d30.entry.js.map (localhost)
...webcomponents/p-f772284e.js (localhost)	...webcomponents/p-f772284e.js.map (localhost)

URL	Zuordnungs-URL
...webcomponents/p-f772284e.js (localhost)	...webcomponents/p-f772284e.js.map (localhost)
...webcomponents/p-f772284e.js (localhost)	...webcomponents/p-f772284e.js.map (localhost)
...webcomponents/p-f772284e.js (localhost)	...webcomponents/p-f772284e.js.map (localhost)
...webcomponents/p-dd3218d3.entry.js (localhost)	...webcomponents/p-dd3218d3.entry.js.map (localhost)
...webcomponents/p-dd3218d3.entry.js (localhost)	...webcomponents/p-dd3218d3.entry.js.map (localhost)
...webcomponents/p-dd3218d3.entry.js (localhost)	...webcomponents/p-dd3218d3.entry.js.map (localhost)
...webcomponents/p-d831bc11.entry.js (localhost)	...webcomponents/p-d831bc11.entry.js.map (localhost)
...webcomponents/p-d831bc11.entry.js (localhost)	...webcomponents/p-d831bc11.entry.js.map (localhost)
...webcomponents/p-d831bc11.entry.js (localhost)	...webcomponents/p-d831bc11.entry.js.map (localhost)
...webcomponents/p-d831bc11.entry.js (localhost)	...webcomponents/p-d831bc11.entry.js.map (localhost)
...webcomponents/p-c31cad0b.entry.js (localhost)	...webcomponents/p-c31cad0b.entry.js.map (localhost)
...webcomponents/p-c31cad0b.entry.js (localhost)	...webcomponents/p-c31cad0b.entry.js.map (localhost)
...webcomponents/p-c31cad0b.entry.js (localhost)	...webcomponents/p-c31cad0b.entry.js.map (localhost)
...webcomponents/p-c31cad0b.entry.js (localhost)	...webcomponents/p-c31cad0b.entry.js.map (localhost)
...webcomponents/p-bc913fa4.entry.js (localhost)	...webcomponents/p-bc913fa4.entry.js.map (localhost)
...webcomponents/p-bc913fa4.entry.js (localhost)	...webcomponents/p-bc913fa4.entry.js.map (localhost)
...webcomponents/p-bc913fa4.entry.js (localhost)	...webcomponents/p-bc913fa4.entry.js.map (localhost)
...webcomponents/p-bbcf6480.entry.js (localhost)	...webcomponents/p-bbcf6480.entry.js.map (localhost)

URL	Zuordnungs-URL
...webcomponents/p-bbcf6480.entry.js (localhost)	...webcomponents/p-bbcf6480.entry.js.map (localhost)
...webcomponents/p-bbcf6480.entry.js (localhost)	...webcomponents/p-bbcf6480.entry.js.map (localhost)
...webcomponents/p-bbcf6480.entry.js (localhost)	...webcomponents/p-bbcf6480.entry.js.map (localhost)
...webcomponents/p-b773df8a.entry.js (localhost)	...webcomponents/p-b773df8a.entry.js.map (localhost)
...webcomponents/p-b773df8a.entry.js (localhost)	...webcomponents/p-b773df8a.entry.js.map (localhost)
...webcomponents/p-a5d22636.entry.js (localhost)	...webcomponents/p-a5d22636.entry.js.map (localhost)
...webcomponents/p-a5d22636.entry.js (localhost)	...webcomponents/p-a5d22636.entry.js.map (localhost)
...webcomponents/p-a5d22636.entry.js (localhost)	...webcomponents/p-a5d22636.entry.js.map (localhost)
...webcomponents/p-a5d22636.entry.js (localhost)	...webcomponents/p-a5d22636.entry.js.map (localhost)
...webcomponents/p-7d1868eb.entry.js (localhost)	...webcomponents/p-7d1868eb.entry.js.map (localhost)
...webcomponents/p-7d1868eb.entry.js (localhost)	...webcomponents/p-7d1868eb.entry.js.map (localhost)
...webcomponents/p-7d1868eb.entry.js (localhost)	...webcomponents/p-7d1868eb.entry.js.map (localhost)
...webcomponents/p-6a2de715.entry.js (localhost)	...webcomponents/p-6a2de715.entry.js.map (localhost)
...webcomponents/p-6a2de715.entry.js (localhost)	...webcomponents/p-6a2de715.entry.js.map (localhost)
...webcomponents/p-6a2de715.entry.js (localhost)	...webcomponents/p-6a2de715.entry.js.map (localhost)
...webcomponents/p-62030357.js (localhost)	...webcomponents/p-62030357.js.map (localhost)
...webcomponents/p-62030357.js (localhost)	...webcomponents/p-62030357.js.map (localhost)
...webcomponents/p-62030357.js (localhost)	...webcomponents/p-62030357.js.map (localhost)

URL	Zuordnungs-URL
...webcomponents/p-62030357.js (localhost)	...webcomponents/p-62030357.js.map (localhost)
...webcomponents/p-53dbf7d7.js (localhost)	...webcomponents/p-53dbf7d7.js.map (localhost)
...webcomponents/p-53dbf7d7.js (localhost)	...webcomponents/p-53dbf7d7.js.map (localhost)
...webcomponents/p-53dbf7d7.js (localhost)	...webcomponents/p-53dbf7d7.js.map (localhost)
...webcomponents/p-53dbf7d7.js (localhost)	...webcomponents/p-53dbf7d7.js.map (localhost)
...webcomponents/p-4ba49bc1.entry.js (localhost)	...webcomponents/p-4ba49bc1.entry.js.map (localhost)
...webcomponents/p-414dd07c.entry.js (localhost)	...webcomponents/p-414dd07c.entry.js.map (localhost)
...webcomponents/p-414dd07c.entry.js (localhost)	...webcomponents/p-414dd07c.entry.js.map (localhost)
...webcomponents/p-3f4649f5.js (localhost)	...webcomponents/p-3f4649f5.js.map (localhost)
...webcomponents/p-3f4649f5.js (localhost)	...webcomponents/p-3f4649f5.js.map (localhost)
...webcomponents/p-3f4649f5.js (localhost)	...webcomponents/p-3f4649f5.js.map (localhost)
...webcomponents/p-3f4649f5.js (localhost)	...webcomponents/p-3f4649f5.js.map (localhost)
...webcomponents/p-3ed4d64e.entry.js (localhost)	...webcomponents/p-3ed4d64e.entry.js.map (localhost)
...webcomponents/p-3ed4d64e.entry.js (localhost)	...webcomponents/p-3ed4d64e.entry.js.map (localhost)
...webcomponents/p-3ed4d64e.entry.js (localhost)	...webcomponents/p-3ed4d64e.entry.js.map (localhost)
...webcomponents/p-398db1e9.entry.js (localhost)	...webcomponents/p-398db1e9.entry.js.map (localhost)
...webcomponents/p-398db1e9.entry.js (localhost)	...webcomponents/p-398db1e9.entry.js.map (localhost)
...webcomponents/p-2b739f89.entry.js (localhost)	...webcomponents/p-2b739f89.entry.js.map (localhost)
...webcomponents/p-299c3b04.js (localhost)	...webcomponents/p-299c3b04.js.map (localhost)

URL	Zuordnungs-URL
...webcomponents/p-299c3b04.js (localhost)	...webcomponents/p-299c3b04.js.map (localhost)
...webcomponents/p-299c3b04.js (localhost)	...webcomponents/p-299c3b04.js.map (localhost)
...webcomponents/p-299c3b04.js (localhost)	...webcomponents/p-299c3b04.js.map (localhost)
...webcomponents/p-112455b1.js (localhost)	...webcomponents/p-112455b1.js.map (localhost)
...webcomponents/p-112455b1.js (localhost)	...webcomponents/p-112455b1.js.map (localhost)
...webcomponents/p-112455b1.js (localhost)	...webcomponents/p-112455b1.js.map (localhost)
...webcomponents/p-112455b1.js (localhost)	...webcomponents/p-112455b1.js.map (localhost)
...webcomponents/p-0e7aefe8.entry.js (localhost)	...webcomponents/p-0e7aefe8.entry.js.map (localhost)
...webcomponents/p-0e7aefe8.entry.js (localhost)	...webcomponents/p-0e7aefe8.entry.js.map (localhost)
...webcomponents/p-0e7aefe8.entry.js (localhost)	...webcomponents/p-0e7aefe8.entry.js.map (localhost)
...webcomponents/p-057a4d55.js (localhost)	...webcomponents/p-057a4d55.js.map (localhost)
...webcomponents/p-057a4d55.js (localhost)	...webcomponents/p-057a4d55.js.map (localhost)
...webcomponents/p-057a4d55.js (localhost)	...webcomponents/p-057a4d55.js.map (localhost)
...webcomponents/p-057a4d55.js (localhost)	...webcomponents/p-057a4d55.js.map (localhost)

NICHT ZUTREFFEND (1)

Ausblenden

Schriftarten mit `font-display: optional` werden vorab geladen ^

Du solltest `optional`-Schriftarten vorab laden, damit sie von Erstbesuchern verwendet werden können. [Weitere Informationen zum Vorabladen von Schriftarten](#)


 Captured at 13. Juni 2023, 12:11

MESZ


 Erster Seitenaufbau

 Desktopemulation with Lighthouse

10.1.1

 Benutzerdefinierte Drosselung

 Einzelner Seitenaufbau

 Using Chromium 114.0.0.0 with
devtools

Generated by **Lighthouse** 10.1.1 | [Problem melden](#)

Lighthouse Report mit Suche: Navigation



Leistung



Barrierefreiheit



Best Practices



SEO



PWA



Leistung

Die Werte sind geschätzt und können variieren.
Die [Leistungsbewertung](#) wird direkt aus diesen Messwerten berechnet. [Siehe Rechner](#).

▲ 0–49 ■ 50–89 ● 90–100



MESSWERTE

[Ansicht maximieren](#)

■ First Contentful Paint

1,0 s

■ Largest Contentful Paint

1,5 s

● Total Blocking Time

10 ms

● Cumulative Layout Shift

0

● Speed Index

1,3 s

[Strukturkarte anzeigen](#)



Prüfungen anzeigen, die für folgende Messwerte relevant sind: [All](#) [FCP](#) [LCP](#) [TBT](#) [CLS](#)

EMPFEHLUNGEN

Empfehlung

Geschätzte Einsparung

▲ Textkomprimierung aktivieren

0,81 Sek ^

Textbasierte Ressourcen sollten komprimiert (gzip, Deflate oder Brotli) ausgeliefert werden, um die Netzwerkbytes insgesamt zu minimieren. [Weitere Informationen zur Textkomprimierung.](#) FCP

LCP

URL	Übertragungsgröße	Mögliche Einsparungen
localhost 1st Party	1.185,4 KiB	952,1 KiB
...css/sdx.min.css (localhost)	446,9 KiB	389,9 KiB
...js/sdx.min.js (localhost)	193,7 KiB	142,3 KiB
/dist/search-integrator.js (localhost)	170,7 KiB	135,9 KiB
...webcomponents/p-c31cad0b.entry.js (localhost)	97,4 KiB	74,3 KiB
...webcomponents/p-0e7ae8e8.entry.js (localhost)	53,9 KiB	43,3 KiB
...webcomponents/p-d831bc11.entry.js (localhost)	54,7 KiB	43,1 KiB
...webcomponents/p-414dd07c.entry.js (localhost)	35,0 KiB	29,2 KiB
...webcomponents/p-53dbf7d7.js (localhost)	28,7 KiB	18,1 KiB
...webcomponents/p-bc913fa4.entry.js (localhost)	19,2 KiB	15,9 KiB
...webcomponents/p-dd3218d3.entry.js (localhost)	18,8 KiB	15,7 KiB
http://localhost:3000	15,1 KiB	10,6 KiB
...webcomponents/p-f772284e.js (localhost)	15,2 KiB	8,9 KiB
...webcomponents/webcomponents.esm.js (localhost)	10,6 KiB	7,6 KiB
...webcomponents/p-7d1868eb.entry.js (localhost)	6,0 KiB	4,4 KiB
...webcomponents/p-6a2de715.entry.js (localhost)	4,8 KiB	3,2 KiB
...webcomponents/p-bbcf6480.entry.js (localhost)	4,4 KiB	2,9 KiB
...webcomponents/p-a5d22636.entry.js (localhost)	4,2 KiB	2,6 KiB
/search-component/search-component.js (localhost)	3,1 KiB	2,1 KiB
...webcomponents/p-3ed4d64e.entry.js (localhost)	3,2 KiB	2,0 KiB

■ Ressourcen beseitigen, die das Rendering blockieren

0,63 Sek ^

Ressourcen blockieren den First Paint deiner Seite. Versuche, wichtiges JS und wichtige CSS inline anzugeben und alle nicht kritischen JS und Stile zurückzustellen. [Informationen dazu, wie sich Ressourcen eliminieren lassen, die das Rendering blockieren.](#) FCP LCP

Drittanbieter-Ressourcen anzeigen (1)

URL	Übertragungsgröße	Mögliche Einsparungen
localhost 1st Party	642,1 KiB	1.260 ms
...css/sdx.min.css (localhost)	447,2 KiB	690 ms
/search-component/search-component.css (localhost)	0,9 KiB	80 ms
...js/sdx.min.js (localhost)	194,0 KiB	490 ms
Tealium Tag-Manager	3,9 KiB	220 ms
...dev/utag.sync.js (tags.tiqcdn.com)	3,9 KiB	220 ms

■ Reduziere nicht verwendete CSS

▬ 0,37 Sek ^

Du kannst ungültige Regeln in Stylesheets reduzieren und CSS-Code zurückstellen, der nicht für ohne Scrollen sichtbare Inhalte („above the fold“) verwendet wird, um den Datenverbrauch durch Netzwerkaktivität zu senken. [Informationen zum Reduzieren von nicht verwendetem CSS-Code.](#)

FCP LCP

URL	Übertragungsgröße	Mögliche Einsparungen
localhost 1st Party	447,2 KiB	398,9 KiB
...css/sdx.min.css (localhost)	447,2 KiB	398,9 KiB
Unattributable	11,2 KiB	11,0 KiB
:root { --sdx-header-height-mobile: 72px; --sdx-header-height-mobile-with-breadcrumbs: 120px; --sdx...	11,2 KiB	11,0 KiB

■ Reduziere nicht verwendetes JavaScript

▬ 0,28 Sek ^

Um den Datenverbrauch durch Netzwerkaktivität zu senken, kannst du nicht verwendetes JavaScript reduzieren und das Laden von Skripts zurückstellen, bis sie benötigt werden.

[Informationen zum Reduzieren von nicht verwendetem JavaScript.](#) LCP

Drittanbieter-Ressourcen anzeigen (1)

URL	Übertragungsgröße	Mögliche Einsparungen
localhost 1st Party	291,7 KiB	212,6 KiB
...js/sdx.min.js (localhost)	194,0 KiB	164,3 KiB
...webcomponents/p-c31cad0b.entry.js (localhost)	97,7 KiB	48,2 KiB
...node_modules/flatpickr/dist/flatpickr.js	47,4 KiB	44,0 KiB
...webcomponents/components/input/input.tsx	9,0 KiB	3,4 KiB
Adobe Tag Manager Tag-Manager	82,2 KiB	22,1 KiB
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	82,2 KiB	22,1 KiB

Mithilfe dieser Empfehlungen lässt sich die Ladezeit deiner Seite möglicherweise verkürzen. Sie haben keinen [direkten Einfluss](#) auf die Leistungsbewertung.

DIAGNOSE

▲ Darauf achten, dass der Text während der Webfont-Ladevorgänge sichtbar bleibt ^

Nutze das CSS-Feature `font-display`, damit der Text für Nutzer sichtbar ist, während Webfonts geladen werden. [Weitere Informationen zu font-display](#). FCP LCP

URL	Mögliche Einsparungen
localhost 1st Party	90 ms
...TheSans/TheSansB_400_.woff2 (localhost)	20 ms
...TheSans/TheSansB_700_.woff2 (localhost)	20 ms
...TheSans/TheSansB_600_.woff2 (localhost)	20 ms
...TheSans/TheSansB_500_.woff2 (localhost)	20 ms
...sdx-icons/sdx-icons.woff2 (localhost)	20 ms

▲ Verwendet keine passiven Listener zur Verbesserung der Scrollleistung ^

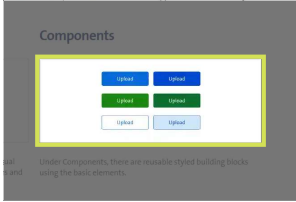


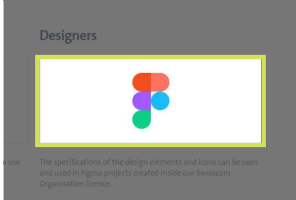
Wenn du deine Event-Listener für Tipp- und Mausradbewegungen als `passive` markierst, kannst du damit die Scrollleistung deiner Seite verbessern. [Weitere Informationen zur Verwendung von passiven Event-Listnern](#).

Quelle
localhost 1st Party
./node_modules/@stencil/core/internal/client/index.js?app-data=conditional:3268:47

▲ Bildelemente haben keine explizite width und height

Lege eine explizite Breite und Höhe für Bildelemente fest, um Layoutverschiebungen zu reduzieren und den CLS-Wert zu verbessern. [Informationen zum Festlegen von Bildabmessungen](#)

CLS

URL	
localhost 1st Party	
 <p>button groups</p> <pre></pre>	... all_the_basics/components.png (localhost)
 <p>Examples typography, colours and components</p> <pre></pre>	...all_the_basics/foundation.png (localhost)
 <p>Code snippet</p> <pre></pre>	...all_the_basics/utilities.png (localhost)
 <p>Figma logo</p> <pre></pre>	...all_the_basics/designer.png (localhost)

■ Statische Inhalte mit einer effizienten Cache-Richtlinie bereitstellen — 9 Ressourcen gefunden

Eine lange Verweildauer im Cache kann wiederholte Besuche deiner Seite beschleunigen. [Weitere](#)

[Informationen zu effizienten Cache-Richtlinien.](#)

URL	Cache-TTL	Übertragungsgröße
Tealium Tag-Manager		59 KiB
...dev/utag.js (tags.tiqcdn.com)	5 Min.	55 KiB
...dev/utag.sync.js (tags.tiqcdn.com)	5 Min.	4 KiB
...tiqapp/utag.v.js?a=swisscom/main/202...&nocookie=1 (tags.tiqcdn.com)	5 Min.	0 KiB
Adobe Tag Manager Tag-Manager		3 KiB
...d199e5db12b9/RC079aeec...-source.min.js (assets.adobedtm.com)	1 Std.	1 KiB
...d199e5db12b9/RC777e9ba...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC991f58f...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
...d199e5db12b9/RC6015071...-source.min.js (assets.adobedtm.com)	1 Std.	0 KiB
Facebook Social		0 KiB
/tr?id=865...&ev=Adobe-Aud...&cd[segID]=13706038,11471880&noscript=1 (www.facebook.com)	None	0 KiB
/tr?id=865...&ev=Adobe-Aud...&cd[segID]=11471880,13706038&noscript=1 (www.facebook.com)	None	0 KiB

■ **Übermäßige DOM-Größe vermeiden — 910 Elemente**

Ein großes DOM führt zu hoher Arbeitsspeichernutzung, langwierigen [Stilberechnungen](#) und kostspieligen [dynamischen Umbrüchen im Layout](#). [Informationen zum Vermeiden eines zu großen DOMs.](#) TBT

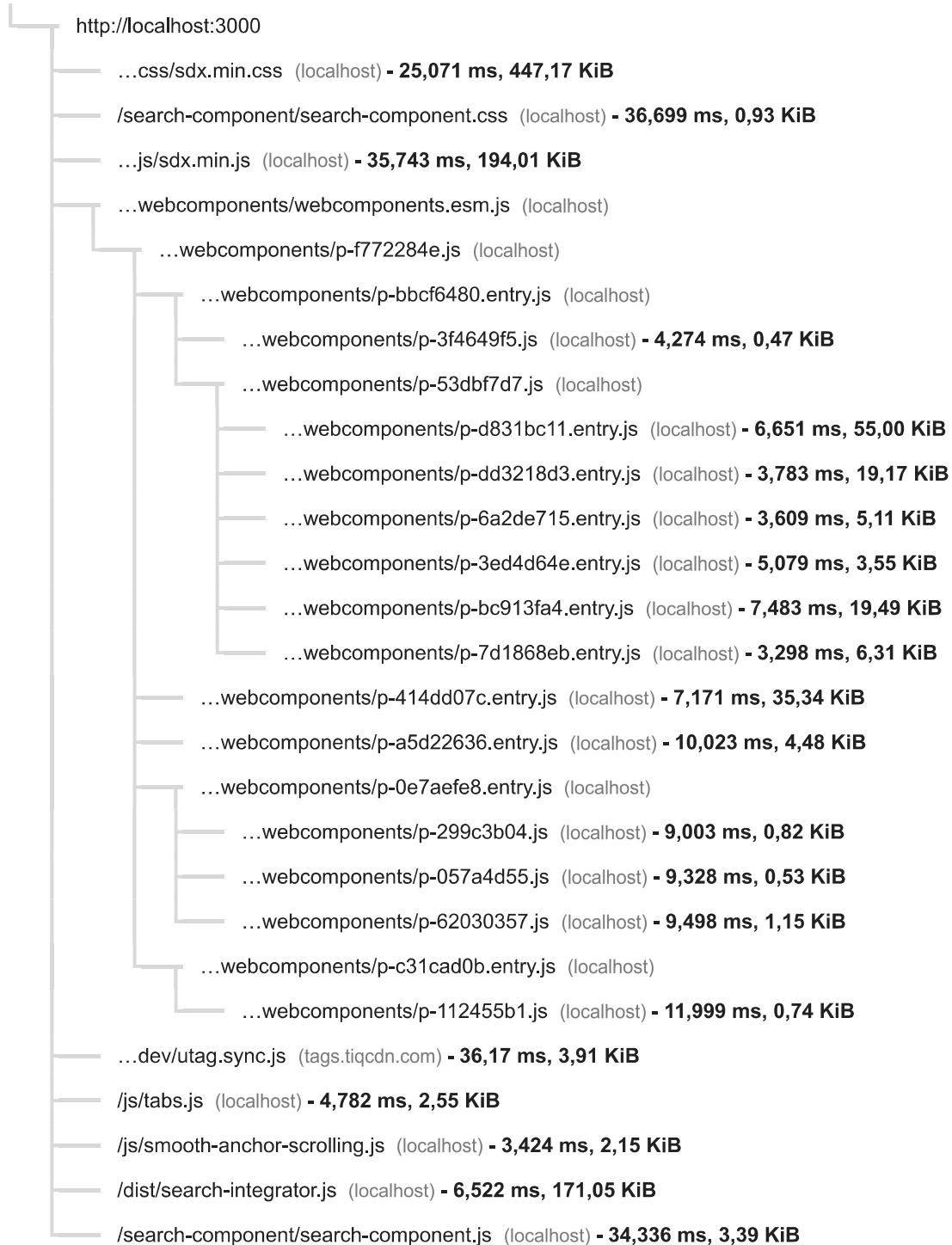
Statistik	Element	Wert
DOM-Elemente insgesamt		910
Maximale DOM-Tiefe	Accessibility 	28
Maximale Anzahl von untergeordneten Elementen	AccordionsBadgesButtonsCardsCarouselChart (bar horizontal)Chart (bar vertical)C... <ul class="list vertical">	41

○ **Verkettung kritischer Anfragen vermeiden — 21 Ketten gefunden**

In den unten aufgeführten Ketten kritischer Anfragen kannst du sehen, welche Ressourcen mit einer hohen Priorität geladen werden. Versuche, die Ketten zu kürzen, die Downloadgröße von Ressourcen zu reduzieren oder das Herunterladen unnötiger Ressourcen zurückzustellen, um den Seitenaufbau zu beschleunigen. [Informationen dazu, wie sich das Verketteten kritischer Anfragen vermeiden lässt.](#) FCP LCP

Maximale Latenz für kritischen Pfad: **693,853 ms**

Anfangsnavigation



Halte die Anfrageanzahl niedrig und die Übertragungsgröße gering — 50 Anfragen • 1.799 KiB

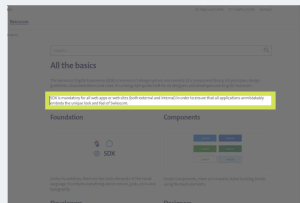
Füge zum Einrichten von Budgets für die Anzahl und Größe von Seitenressourcen eine budget.json-Datei hinzu. [Weitere Informationen zu Leistungsbudgets.](#)

Ressourcentyp	Anfragen	Übertragungsgröße
Gesamt	50,0	1.798,9 KiB
Skript	32,0	881,8 KiB
Stylesheet	2,0	448,1 KiB
Schriftart	5,0	279,8 KiB
Bild	7,0	171,8 KiB
Dokument	1,0	15,4 KiB
Sonstige	3,0	2,1 KiB
Medien	0,0	0,0 KiB
Drittanbieter	13,0	146,6 KiB

○ Largest Contentful Paint-Element — 1 Element gefunden ^

Dies ist das größte Inhaltselement, das im Darstellungsbereich angezeigt wird. [Weitere Informationen zum Largest Contentful Paint-Element](#) LCP

Element



SDX is mandatory for all web apps or web sites (both external and internal) in ...
 ...
 <p>

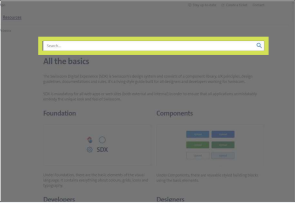
○ Lange Hauptthread-Aufgaben vermeiden — 1 lange Aufgabe gefunden ^

Listet die längsten Aufgaben im Hauptthread auf. Das ist nützlich, um die wichtigsten Ursachen für die Eingabeverzögerungen zu ermitteln. [Informationen dazu, wie sich lange Aufgaben im Hauptthread vermeiden lassen](#) TBT

URL	Beginn	Dauer
Adobe Tag Manager Tag-Manager		75 ms
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	1.201 ms	75 ms

○ Nicht zusammengesetzte Animationen vermeiden — 5 animierte Elemente gefunden ^

Nicht zusammengesetzte Animationen werden eventuell nicht richtig gerendert und können den CLS-Wert erhöhen. [Informationen dazu, wie sich nicht zusammengefasste Animationen vermeiden lassen](#) CLS

Element	Name
 <pre>div.component > div.wrapper > input#1 <input id="1" class="input" placeholder="Search..." autocorrect="off" autocapitalize="off" spellcheck="false" size="1" aria-invalid="false" aria-describedby="additional-information" aria- placeholder="" type="search" style="padding-right: 48px;"></pre>	
Nicht unterstützte CSS-Property: visibility	visibility
 <pre>Search <button class="clickable button-reset transparent" href="" type="button"></pre>	
Nicht unterstützte CSS-Property: visibility	visibility
 <pre>copy link <button class="clickable button-reset transparent" href="" type="button"></pre>	
Nicht unterstützte CSS-Property: visibility	visibility
 <pre>Search... <div pseudo="-webkit-input-placeholder" id="placeholder" style="display: block !important;"></pre>	
Nicht unterstützte CSS-Property: visibility	visibility
 <pre>ul.list > li.item > a.link > span.img </pre>	
Nicht unterstützte CSS-Properties: background-position-x, background-position-y	repeatingA animation
Nicht unterstützte CSS-Properties: background-position-x, background-position-y	initialAnima tion

Weitere Informationen zur Leistung deiner App findest du hier. Diese Angaben haben keinen [direkten Einfluss](#) auf die Leistungsbewertung.

● Bilder richtig dimensionieren ^

Stelle Bilder bereit, die eine angemessene Größe haben, um mobile Daten zu sparen und die Ladezeit zu verbessern. [Weitere Informationen.](#)

● Nicht sichtbare Bilder aufschieben ^

Wenn du Lazy Loading für nicht sichtbare und versteckte Bilder verwendest und sie zurückstellst, bis alle wichtigen Ressourcen geladen wurden, kannst du die Zeit bis Interaktivität reduzieren. [Weitere Informationen.](#)

● CSS komprimieren — Mögliche Einsparung von 73 KiB ^

Durch das Reduzieren von CSS-Dateien lassen sich Netzwerknutzlasten senken. [Informationen zum Reduzieren von CSS.](#) FCP LCP

URL	Übertragungsgröße	Mögliche Einsparungen
localhost 1st Party	447,2 KiB	73,0 KiB
...css/sdx.min.css (localhost)	447,2 KiB	73,0 KiB

● JavaScript komprimieren ^

Durch die Komprimierung von JavaScript-Dateien können Nutzlastgrößen und die Zeit zum Parsen von Skripts reduziert werden. [Informationen zum Reduzieren von JavaScript.](#) FCP LCP

● Bilder effizient codieren ^

Optimierte Bilder werden schneller geladen und verbrauchen weniger mobile Daten. [Informationen zum effizienten Codieren von Bildern.](#)

● Bilder in modernen Formaten bereitstellen — Mögliche Einsparung von 39 KiB ^

Bildformate wie WebP und AVIF bieten oft eine bessere Komprimierung als PNG oder JPEG, wodurch sie schneller heruntergeladen werden und weniger Daten verbrauchen. [Weitere Informationen zu modernen Bildformaten.](#)

URL	Größe der Ressource	Mögliche Einsparungen
localhost 1st Party	44,3 KiB	38,7 KiB
		
Code snippet	...all_the_basics/utilities.png (localhost)	44,3 KiB
<pre></pre>		
	38,7 KiB	

● Vorverbindung zu erforderlichen Ursprüngen aufbauen ^

Mithilfe von Hinweisen auf Ressourcen wie `preconnect` oder `dns-prefetch` lassen sich frühzeitig Verbindungen zu wichtigen Drittanbieterursprüngen herstellen. [Informationen zum Vorverbinden mit erforderlichen Ursprüngen.](#) FCP LCP

● Kurze Erstreaktionszeit des Servers — Stammdokument brauchte 0 ms ^

Achte auf eine möglichst kurze Serverantwortzeit für das Hauptdokument, weil alle anderen Anfragen davon abhängen. [Weitere Informationen zum Messwert „Time to First Byte“.](#) FCP LCP

URL	Zeitaufwand
localhost 1st Party	0 ms
http://localhost:3000	0 ms

● Mehrere Weiterleitungen auf die Seite vermeiden ^

Weiterleitungen führen zu zusätzlichen Verzögerungen, bevor die Seite geladen werden kann. [Informationen zum Vermeiden von Seitenweiterleitungen.](#) FCP LCP

○ Wichtige Anforderungen vorab laden ^

Mit `<link rel=preload>` kannst du das Abrufen von Ressourcen priorisieren, die derzeit beim Seitenaufbau erst später angefordert werden. [Informationen zum Vorabladen wichtiger Anfragen.](#) FCP LCP

● Verwende HTTP/2 ^

HTTP/2 bietet gegenüber HTTP/1.1 viele Vorteile, wie z. B. binäre Header und Multiplexverfahren. [Weitere Informationen zu HTTP/2.](#)

● Videoformate für animierte Inhalte verwenden ^

Große GIF-Dateien sind nur bedingt für die Auslieferung animierter Inhalte geeignet. Du kannst stattdessen MPEG4- oder WebM-Videos für Animationen und PNG oder WebP für statische Bilder verwenden und so die Netzwerkbytes reduzieren. [Weitere Informationen zu effizienten Videoformaten](#) LCP

● In JavaScript-Bundles doppelt vorhandene Module entfernen ^

Entferne große, doppelt vorhandene JavaScript-Module aus Bundles, um unnötige Datenübertragungen im Netzwerk zu reduzieren. TBT

● Verhindern, dass in modernen Browsern veraltetes JavaScript bereitgestellt wird — Mögliche Einsparung von 7 KiB ^

Dank Polyfills und Transformationen können veraltete Browser die neuen JavaScript-Funktionen nutzen. Bei modernen Browsern hingegen sind viele davon nicht erforderlich. Für dein JavaScript-Bundle solltest du eine moderne Skriptimplementierungsstrategie unter Verwendung der module/nomodule-Funktionserkennung einsetzen. So kannst du einerseits den an moderne Browser übermittelten Code auf das Nötige reduzieren und gleichzeitig veraltete Browser bestmöglich unterstützen. [Informationen zur Verwendung modernen JavaScripts](#) TBT

URL	Mögliche Einsparungen
Tealium Tag-Manager	6,5 KiB
...dev/utag.js (tags.tiqcdn.com)	6,5 KiB
...dev/utag.js:1475:14 (tags.tiqcdn.com)	Object.keys
Adobe Tag Manager Tag-Manager	0,1 KiB
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	0,1 KiB

URL	Mögliche Einsparungen
...4cf4f6043663/launch-0c232d70d3dd-development.min.js:7:1084 (assets.adobedtm.com)	@babel/plugin-transform-classes

○ Largest Contentful Paint-Bild vorab laden ^

Wenn der Seite das LCP-Element dynamisch hinzugefügt wird, solltest du das Bild vorab laden, um den LCP zu verbessern. [Weitere Informationen zum Vorabladen von LCP-Elementen.](#) LCP

● Vermeidet sehr große Netzwerknutzlasten — Die Gesamtgröße war 1.907 KiB ^

Große Netzwerknutzlasten kosten Nutzer bares Geld und hängen eng mit langen Ladezeiten zusammen. [Informationen zum Verringern der Nutzlastgröße.](#) LCP

Drittanbieter-Ressourcen anzeigen (2)

URL	Übertragungsgröße
localhost 1st Party	1.253,3 KiB
...css/sdx.min.css (localhost)	447,2 KiB
...js/sdx.min.js (localhost)	194,0 KiB
/dist/search-integrator.js (localhost)	171,0 KiB
/img/favicon.ico (localhost)	107,9 KiB
...assets/lifeform-spritesheet@2x.png (localhost)	104,4 KiB
...webcomponents/p-c31cad0b.entry.js (localhost)	97,7 KiB
...sdx-icons/sdx-icons.woff2 (localhost)	76,1 KiB
...webcomponents/p-d831bc11.entry.js (localhost)	55,0 KiB
Adobe Tag Manager Tag-Manager	82,2 KiB
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	82,2 KiB
Tealium Tag-Manager	55,0 KiB
...dev/utag.js (tags.tiqcdn.com)	55,0 KiB

○ Markierungen und Messungen für das Nutzertiming ^

Du kannst die User Timing API in deine App einbinden. Damit lässt sich die Leistung der App während wichtiger Nutzerinteraktionen in der Praxis messen. [Weitere Informationen zu User Timing-Markierungen.](#)

JavaScript-Ausführungszeit — 0,4 s



Versuche, die Zeit für das Parsen, Kompilieren und Ausführen von JavaScript zu reduzieren. Die Bereitstellung kleinerer JS-Nutzlasten kann dabei helfen. [Informationen zum Reduzieren der JavaScript-Ausführungszeit.](#) TBT

Drittanbieter-Ressourcen anzeigen (2)

URL	CPU-Zeit insgesamt	Skriptauswertung	Parsen von Skripten
localhost 1st Party	461 ms	141 ms	3 ms
http://localhost:3000	252 ms	5 ms	2 ms
...webcomponents/p-f772284e.js (localhost)	129 ms	116 ms	0 ms
...webcomponents/p-53dbf7d7.js (localhost)	79 ms	19 ms	0 ms
Unattributable	174 ms	46 ms	0 ms
Unattributable	174 ms	46 ms	0 ms
Adobe Tag Manager Tag-Manager	120 ms	103 ms	13 ms
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	120 ms	103 ms	13 ms
Tealium Tag-Manager	56 ms	48 ms	7 ms
...dev/utag.js (tags.tiqcdn.com)	56 ms	48 ms	7 ms

Minimiert den Aufwand für den Hauptthread — 0,9 s



Versuche, die Zeit für das Parsen, Kompilieren und Ausführen von JavaScript zu reduzieren. Die Bereitstellung kleinerer JS-Nutzlasten kann dabei helfen. [Informationen dazu, wie sich der Aufwand für den Hauptthread minimieren lässt](#) TBT

Kategorie	Zeitaufwand
Script Evaluation	402 ms
Other	260 ms
Style & Layout	151 ms
Parse HTML & CSS	51 ms
Rendering	44 ms

Kategorie	Zeitaufwand
Script Parsing & Compilation	36 ms
Garbage Collection	5 ms

- **Drittanbiaternutzung minimieren** — Code von Drittanbiestern hat den Hauptthread 10 ms lang blockiert ^

Code von Drittanbiestern kann die Ladegeschwindigkeit erheblich beeinträchtigen. Beschränke die Zahl redundanter Drittanbieter und versuche, solchen Code erst zu laden, nachdem die Seite vollständig geladen wurde. [Informationen dazu, wie sich die Auswirkungen von Drittanbietercode minimieren lassen.](#) TBT

Drittanbieter	Übertragungsgröße	Dauer der Blockierung des Hauptthreads
Adobe Tag Manager Tag-Manager	85 KiB	7 ms
...4cf4f6043663/launch-0c232d70d3dd-development.min.js (assets.adobedtm.com)	82 KiB	7 ms
Tealium Tag-Manager	60 KiB	0 ms
...dev/utag.js (tags.tiqcdn.com)	55 KiB	0 ms
Weitere Ressourcen	5 KiB	0 ms
adobedc.net	2 KiB	0 ms
Facebook Social	0 KiB	0 ms

- **Lazy Loading von Ressourcen von Drittanbietern mit Fassaden** ^

Für einen Teil des eingebetteten Codes von Drittanbietern kann Lazy Loading verwendet werden. Du kannst ihn durch eine Fassade ersetzen, bis er benötigt wird. [Informationen zum Zurückstellen von Drittanbietern mit einer Fassade.](#) TBT

- **Largest Contentful Paint-Bild wurde nicht mit Lazy Loading geladen** ^

Ohne Scrollen sichtbare Bilder („above the fold“) die mit Lazy Loading geladen werden, werden später im Lebenszyklus der Seite gerendert. Dies kann zu Verzögerungen beim Largest Contentful Paint führen. [Weitere Informationen zum optimalen Lazy Loading.](#) LCP

- **Umfangreiche Layoutverschiebungen vermeiden** ^

Diese DOM-Elemente tragen am stärksten zur CLS der Seite bei. [Informationen zur Verbesserung der CLS](#) CLS

● Verwendet kein `document.write()` ^

Für Nutzer mit langsamen Verbindungen können externe Skripts, die dynamisch über `document.write()` eingefügt werden, den Seitenaufbau um einige Sekunden verzögern.

[Informationen zum Vermeiden von „document.write\(\)“](#)

● Hat ein `<meta name="viewport">`-Tag mit `width` oder `initial-scale` ^

Ein `<meta name="viewport">` optimiert deine App für Bildschirmgrößen von Mobilgeräten und verhindert darüber hinaus [300-Millisekunden-Verzögerungen bei Nutzereingaben](#). [Weitere](#)

[Informationen zur Verwendung des Darstellungsbereich-Meta-Tags](#). TBT

● Seite hat die Wiederherstellung des Back-Forward-Caches nicht verhindert ^

Viele Bedienvorgänge werden ausgeführt, indem eine vorherige Seite aufgerufen oder zur nächsten gegangen wird. Der Back-Forward-Cache (bfcache) kann diese Vorgänge beschleunigen. [Weitere Informationen zum bfcache](#)



Barrierefreiheit

Mit diesen Prüfungen erfährst du, [wie du die Barrierefreiheit deiner Web-App verbesserst](#). Nur bestimmte Probleme mit der Barrierefreiheit können durch automatisierte Tests erkannt werden. Deshalb ist es empfehlenswert, zusätzlich manuelle Tests durchzuführen.

TABELLEN UND LISTEN

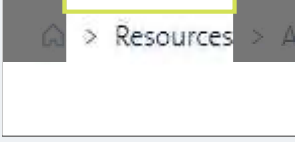
▲ Listenelemente (``) befinden sich nicht in übergeordneten ``-, ``- oder `<menu>`-Elementen. ^

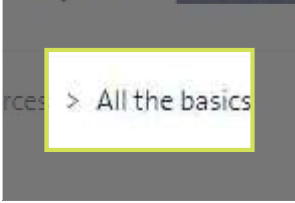
Listenelemente (``) müssen sich in einem übergeordneten ``-, ``- oder `<menu>`-Element befinden, damit sie von Screenreadern richtig angesagt werden können. [Weitere Informationen](#)

[zum Strukturieren von Listen.](#)

Fehlerhafte Elemente	
	Brand Center <code><li class="item leaf"></code>
	Git Repo <code><li class="item leaf"></code>
	Stay up-to-date <code><li class="item leaf"></code>
	Create a ticket <code><li class="item leaf"></code>
	Contact <code><li class="item leaf"></code>
	Principles <code><li class="item leaf"></code>
	Resources <code><li class="item active"></code>
	icon-home <code><li class="item"></code>
	Resources

Fehlerhafte Elemente


<li class="item">


<li class="item">

Hier findest du Möglichkeiten, um das Lesen von Daten in Tabellen oder Listen mit Hilfstechnologie wie Screenreadern zu verbessern.

ZUSÄTZLICHE ELEMENTE ZUR MANUELLEN ÜBERPRÜFUNG (10)

Ausblenden

The page has a logical tab order ^

Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. [Learn more about logical tab ordering.](#)

Interactive controls are keyboard focusable ^

Custom interactive controls are keyboard focusable and display a focus indicator. [Learn how to make custom controls focusable.](#)

Interactive elements indicate their purpose and state ^

Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. [Learn how to decorate interactive elements with affordance hints.](#)

The user's focus is directed to new content added to the page ^

If new content, such as a dialog, is added to the page, the user's focus is directed to it. [Learn how to direct focus to new content.](#)

User focus is not accidentally trapped in a region ^

A user can tab into and out of any control or region without accidentally trapping their focus. [Learn how to avoid focus traps.](#)

Custom controls have associated labels ^

Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. [Learn more about custom controls and labels.](#)

Custom controls have ARIA roles ^

Custom interactive controls have appropriate ARIA roles. [Learn how to add roles to custom controls.](#)

Visual order on the page follows DOM order ^

DOM order matches the visual order, improving navigation for assistive technology. [Learn more about DOM and visual ordering.](#)

Offscreen content is hidden from assistive technology ^

Offscreen content is hidden with display: none or aria-hidden=true. [Learn how to properly hide offscreen content.](#)

HTML5 landmark elements are used to improve navigation ^

Landmark elements (<main>, <nav>, etc.) are used to improve the keyboard navigation of the page for assistive technology. [Learn more about landmark elements.](#)

Diese Prüfungen sind für Bereiche vorgesehen, für die automatische Testtools nicht geeignet sind. Weitere Informationen findest du in unserem Leitfaden zur [Durchführung einer Prüfung auf Barrierefreiheit.](#)

BESTANDENE PRÜFUNGEN (20)

Ausblenden

[aria-*]-Attribute entsprechen ihren Rollen ^

Jede ARIA-ro1e unterstützt eine bestimmte Untergruppe von aria-*-Attributen. Wenn sie jedoch falsch zugeordnet sind, werden die aria-*-Attribute ungültig. [Informationen zum Zuordnen von ARIA-Attributen zu ihren Rollen.](#)

[aria-hidden="true"] ist in dem Dokument <body> nicht vorhanden ^

Hilfstechnologien wie Screenreader funktionieren nicht richtig, wenn für den <body> des Dokuments aria-hidden="true" festgelegt ist. [Informationen zu den Auswirkungen von aria-hidden auf den Textbereich des Dokuments.](#)

[role]-Elemente verfügen über alle erforderlichen [aria-*]-Attribute ^

Für einige ARIA-Rollen sind Attribute erforderlich, die Screenreadern den Zustand des Elements beschreiben. [Weitere Informationen zu Rollen und erforderlichen Attributen.](#)

● [role]-Werte sind gültig ^

Für ARIA-Rollen müssen gültige Werte angegeben sein, damit sie die beabsichtigten Hilfsfunktionen erfüllen können. [Weitere Informationen zu gültigen ARIA-Rollen.](#)

● [aria-*]-Attribute weisen gültige Werte auf ^

Hilfstechnologien wie Screenreader können ARIA-Attribute mit ungültigen Werten nicht interpretieren. [Weitere Informationen zu gültigen Werten für ARIA-Attribute.](#)

● [aria-*]-Attribute sind gültig und richtig geschrieben ^

Hilfstechnologien wie Screenreader können ARIA-Attribute mit ungültigen Namen nicht interpretieren. [Weitere Informationen zu gültigen ARIA-Attributen.](#)

● Die Namen der Schaltflächen sind für Screenreader zugänglich ^

Wenn eine Schaltfläche keinen barrierefreien Namen hat, wird sie von Screenreadern nur als „Schaltfläche“ angesagt. Dadurch ist sie für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Informationen zum barrierefreien Gestalten von Schaltflächen.](#)

● ARIA-IDs sind eindeutig ^

Der Wert einer ARIA-ID muss eindeutig sein, damit andere Instanzen nicht von Hilfstechnologien übersehen werden. [Informationen zum Korrigieren doppelter ARIA-IDs.](#)

● Bildelemente verfügen über [alt]-Attribute ^

Für informative Elemente sollte ein kurzer, beschreibenden alternativer Text verwendet werden. Dekorative Elemente können mit einem leeren ALT-Attribut ignoriert werden. [Weitere Informationen zum Attribut alt.](#)

● Formularelemente sind mit Labels verknüpft ^

Durch Labels wird gewährleistet, dass Steuerelemente für Formulare von Hilfstechnologien wie Screenreadern richtig angesagt werden. [Weitere Informationen zu Labels für Formularelemente.](#)

- `[user-scalable="no"]` wird nicht im `<meta name="viewport">`-Element verwendet und das `[maximum-scale]`-Attribut ist nicht kleiner als 5. ^

Wenn du die Zoomfunktion deaktivierst, können Nutzer mit eingeschränktem Sehvermögen, die auf die Bildschirmvergrößerung angewiesen sind, den Inhalt einer Webseite nicht richtig sehen.

[Weitere Informationen zum Darstellungsbereich-Meta-Tag.](#)

- `[aria-hidden="true"]`-Elemente enthalten keine fokussierbaren Unterelemente ^

Fokussierbare Nachfolgerelemente in einem `[aria-hidden="true"]`-Element führen dazu, dass Nutzer von Hilfstechnologien wie Screenreadern solche interaktiven Elemente nicht verwenden können. [Informationen zu den Auswirkungen von aria-hidden auf fokussierbare Elemente.](#)

- Das Kontrastverhältnis von Hintergrund- und Vordergrundfarben ist ausreichend ^

Text mit geringem Kontrast ist für viele Nutzer schlecht oder gar nicht lesbar. [Informationen zu einem ausreichenden Farbkontrast.](#)

- Dokument enthält ein `<title>`-Element ^

Der Titel gibt Screenreader-Nutzern einen Überblick über die Seite. Nutzer von Suchmaschinen verlassen sich stark auf ihn, um zu entscheiden, ob eine Seite für ihre Suche relevant ist. [Weitere Informationen zu Dokumenttiteln.](#)

- `[id]`-Attribute zu aktiven, fokussierbaren Elementen sind eindeutig ^

Für alle fokussierbaren Elemente ist eine eindeutige `id` erforderlich, damit sie von Hilfstechnologien erkannt werden können. [Informationen zum Korrigieren doppelter ids.](#)

- `<html>`-Element hat ein `[lang]`-Attribut ^

Wenn für eine Seite kein `lang`-Attribut angegeben ist, nehmen Screenreader an, dass sie in der Standardsprache vorliegt, die der Nutzer beim Einrichten des Screenreaders ausgewählt hat. Ist das nicht der Fall, gibt der Screenreader den Inhalt der Seite möglicherweise falsch aus. [Weitere Informationen zum lang-Attribut.](#)

Das `<html>`-Element hat einen gültigen Wert für sein `[lang]`-Attribut ^

Wenn ein gültiger [BCP-47-Sprachcode](#) angegeben wird, kann der Text von einem Screenreader korrekt wiedergegeben werden. [Informationen zur Verwendung des Attributs lang.](#)

Links haben einen leicht erkennbaren Namen ^

Linktext, der erkennbar, einzigartig und fokussierbar ist, erleichtert Screenreader-Nutzern die Verwendung. Dies gilt auch für alternativen Text für Bilder, die als Links verwendet werden. [Informationen zu barrierefreien Links.](#)

Listen enthalten nur ``-Elemente und Elemente zur Skriptunterstützung (`<script>` sowie `<template>`). ^

Screenreader sagen Listen auf bestimmte Art und Weise an. Wenn die Liste richtig strukturiert ist, kann der Screenreader sie besser ausgeben. [Weitere Informationen zum Strukturieren von Listen.](#)

Überschriftenelemente werden in einer fortlaufenden absteigenden Reihenfolge angezeigt ^

Richtig geordnete Überschriften, die keine Ebenen überspringen, geben der Seite eine semantische Struktur. Nutzer von Hilfstechnologien können sich so leichter auf der Seite zurechtfinden und die Inhalte besser verstehen. [Weitere Informationen zur Reihenfolge von Überschriften.](#)

NICHT ZUTREFFEND (23)

Ausblenden

`[accesskey]`-Werte sind eindeutig ^

Mithilfe von Tastenkombinationen können Nutzer schnell den Fokus auf einen Bereich der Seite verschieben. Damit die Navigation richtig funktioniert, darf jede Tastenkombination nur einmal vergeben sein. [Weitere Informationen zu Tastenkombinationen.](#)

`button-`, `link-` und `menuitem-`Elemente haben zugängliche Namen ^

Wenn ein Element keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Informationen zum barrierefreieren Gestalten von Befehlselementen.](#)

○ ARIA-Eingabefelder haben zugängliche Namen ^

Wenn ein Eingabefeld keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Weitere Informationen zu Labels für Eingabefelder.](#)

○ ARIA `meter`-Elemente haben zugängliche Namen ^

Wenn ein Messtool-Element keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Weitere Informationen zum Benennen von meter-Elementen](#)

○ ARIA `progressbar`-Elemente haben zugängliche Namen ^

Wenn ein `progressbar`-Element keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Informationen zum Kennzeichnen von progressbar-Elementen.](#)

○ Die Elemente mit einer ARIA-`[role]`, deren untergeordnete Elemente eine bestimmte `[role]` enthalten müssen, haben alle erforderlichen untergeordneten Elemente. ^

Einige übergeordnete ARIA-Rollen müssen bestimmte untergeordnete Rollen enthalten, damit sie die beabsichtigten Hilfsfunktionen erfüllen können. [Weitere Informationen zu Rollen und erforderlichen untergeordneten Elementen.](#)

○ `[role]`-Elemente sind ihren jeweils erforderlichen übergeordneten Elementen untergeordnet ^

Einige untergeordnete ARIA-Rollen müssen in bestimmten übergeordneten Rollen enthalten sein, damit sie die beabsichtigten Hilfsfunktionen erfüllen können. [Weitere Informationen zu ARIA-Rollen und erforderlichen übergeordneten Elementen.](#)

○ ARIA-Ein-/Aus-Schaltflächen haben zugängliche Namen ^

Wenn eine Ein-/Aus-Schaltfläche keinen barrierefreien Namen hat, wird sie von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist sie für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Weitere Informationen zu Ein-/Aus-Schaltflächen.](#)

○ ARIA `tooltip`-Elemente haben zugängliche Namen ^

Wenn ein Kurzinfo-Element keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Weitere Informationen zum Benennen von tooltip-Elementen](#)

○ ARIA `treeitem`-Elemente haben zugängliche Namen ^

Wenn ein `treeitem`-Element keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Weitere Informationen zum Kennzeichnen von `treeitem`-Elementen.](#)

- Die Seite enthält eine Überschrift, einen Link zum Überspringen oder einen Landmark-Bereich ^

Wenn Tastaturnutzer Inhalte umgehen können, die sich wiederholen, sorgt das für eine effizientere Navigation. [Weitere Informationen zum Umgehen von Blockierungen.](#)

- `<d1>`-Elemente enthalten ausschließlich Gruppen aus `<dt>`- und `<dd>`-Elementen sowie `<script>`-, `<template>`- oder `<div>`-Elemente, die richtig angeordnet sind. ^

Wenn Definitionslisten nicht korrekt mit Markup versehen sind, kann es zu verwirrenden oder ungenauen Screenreader-Ausgaben kommen. [Informationen zum Strukturieren von Definitionslisten.](#)

- Definitionlistenelemente sind in `<d1>`-Elemente eingefasst ^

Definitionlistenelemente (`<dt>` und `<dd>`) müssen in ein übergeordnetes `<d1>`-Element eingefasst sein, damit sie von Screenreadern richtig angesagt werden können. [Informationen zum Strukturieren von Definitionslisten.](#)

- Kein Formularfeld hat mehrere Labels ^

Formularfelder mit mehreren Labels werden von Hilfstechnologien wie Screenreadern unter Umständen missverständlich angesagt, da sie entweder das erste, das letzte oder alle Labels verwenden. [Informationen zur Verwendung von Formularlabels.](#)

- `<frame>`- oder `<iframe>`-Elemente verfügen über einen Titel ^

Screenreader-Nutzer sind auf Frametitel angewiesen, die die Inhalte von Frames beschreiben. [Weitere Informationen zu Frametiteln.](#)

- `<input type="image">`-Elemente haben `[alt]`-Text ^

Wenn ein Bild als `<input>`-Schaltfläche verwendet wird, kann alternativer Text Screenreader-Nutzern helfen, den Zweck der Schaltfläche besser zu verstehen. [Informationen zum Alt-Text für Eingabebilder.](#)

- Dieses Dokument verwendet `<meta http-equiv="refresh">` nicht ^

Nutzer rechnen nicht damit, dass eine Seite automatisch aktualisiert wird. Außerdem wird dadurch der Fokus wieder auf den Seitenanfang verschoben. Das kann für den Nutzer frustrierend oder

irritierend sein. [Weitere Informationen zum Meta-Tag „Refresh“](#).

`<object>`-Elemente haben alternativen Text ^

Screenreader können lediglich Textinhalte interpretieren. Wenn du `<object>`-Elementen alternativen Text hinzufügst, können Screenreader-Nutzer besser verstehen, was diese Elemente darstellen. [Weitere Informationen zum Alt-Text für object-Elemente](#).

Kein Element hat einen `[tabindex]`-Wert größer als 0 ^

Ein Wert größer als 0 impliziert eine explizite Navigationsanordnung. Das ist zwar technisch möglich, aber für Nutzer, die auf Hilfstechnologien angewiesen sind, häufig frustrierend. [Weitere Informationen zum Attribut tabindex](#).

Zellen in einem "`<table>`"-Element, die das Attribut "`[headers]`" enthalten, verweisen auf Zellen in derselben Tabelle. ^

Screenreader bieten Funktionen, die die Navigation in Tabellen vereinfachen. Wenn du dafür sorgst, dass `<td>`-Zellen, die das Attribut `[headers]` verwenden, nur auf andere Zellen in derselben Tabelle verweisen, kann dies für Screenreader-Nutzer hilfreich sein. [Weitere Informationen zum Attribut headers](#).

Für `<th>`-Elemente und Elemente mit `[role="columnheader"/"rowheader"]` sind Datenzellen vorhanden, die sie beschreiben. ^

Screenreader bieten Funktionen, die die Navigation in Tabellen vereinfachen. Wenn du dafür sorgst, dass Tabellenüberschriften immer auf bestimmte Zellen verweisen, kann dies für Screenreader-Nutzer hilfreich sein. [Weitere Informationen zu Tabellenüberschriften](#).

`[lang]`-Attribute weisen einen gültigen Wert auf ^

Wenn ein gültiger [BCP-47-Sprachcode](#) für Elemente angegeben wird, kann der Text besser von Screenreadern ausgesprochen werden. [Informationen zur Verwendung des Attributs lang](#).

`<video>`-Elemente enthalten ein `<track>`-Element mit `[kind="captions"]` ^

Wenn ein Video Untertitel enthält, können gehörlose und hörgeschädigte Nutzer die Informationen im Video besser verstehen. [Weitere Informationen zu Untertiteln in Videos](#).



Best Practices

VERTRAUEN UND SICHERHEIT

▲ Verwendet nicht HTTPS — 1 unsichere Anfrage gefunden ^

Alle Websites sollten durch HTTPS geschützt werden – selbst wenn sie keine sensiblen Daten enthalten. Auch [gemischte Inhalte](#), bei denen einige Ressourcen über HTTP geladen werden, obwohl die ursprüngliche Anfrage über HTTPS gestellt wurde, sind zu vermeiden. HTTPS verhindert, dass andere die Website manipulieren oder die Kommunikation zwischen deiner App und deinen Nutzern mitverfolgen können, und ist eine Voraussetzung für HTTP/2 und viele neue Webplattform-APIs. [Weitere Informationen zu HTTPS](#).

Unsichere URL	Lösung anfordern
Tealium Tag-Manager	
...dev/utag.js (tags.tiqcdn.com)	Zugelassen

○ Sicherstellen, dass CSP effektiv gegen XSS-Angriffe wirkt ^

Eine starke Content Security Policy (CSP) reduziert das Risiko für Cross-Site-Scripting-Angriffe (XSS-Angriffe) erheblich. [Weitere Informationen](#)

Beschreibung	Anweisung	Schweregrad
Kein CSP im erzwungenen Modus gefunden		Hoch

BESTANDENE PRÜFUNGEN (12)

Ausblenden

● Fordert während des Seitenaufbaus keine Berechtigung zur Standortbestimmung an ^

Wenn Websites den Standort ohne Begründung anfordern, sind Nutzer schnell misstrauisch oder irritiert. Versuche stattdessen, die Anforderung mit einer Nutzeraktion zu verbinden. [Weitere Informationen zur Berechtigung zur Standortbestimmung](#).

● Fordert während des Seitenaufbaus keine Benachrichtigungsberechtigung an ^

Wenn Websites die Berechtigung zum Senden von Benachrichtigungen ohne Begründung anfordern, sind Nutzer schnell misstrauisch oder irritiert. Versuche stattdessen, die Anforderung mit Touch-Gesten zu verbinden. [Weitere Informationen zum verantwortungsvollen Einholen der Berechtigung zum Senden von Benachrichtigungen.](#)

● Allows users to paste into input fields ^

Preventing input pasting is a UX anti-pattern, and undermines good security policy. [Learn more about user-friendly input fields.](#)

● Zeigt Bilder mit einem korrekten Seitenverhältnis an ^

Die Bildgröße sollte dem natürlichen Seitenverhältnis entsprechen. [Weitere Informationen zum Seitenverhältnis von Bildern.](#)

● Stellt Bilder mit angemessener Auflösung bereit ^

Die originalen Abmessungen eines Bildes sollten proportional zu der Displaygröße und dem Pixel-Verhältnis sein, damit das Bild optimal angezeigt wird. [Weitere Informationen zu responsiven Bildern.](#)

● Seite verfügt über den HTML-DOCTYPE ^

Wenn du einen DOCTYPE angibst, verhinderst du, dass der Browser zum Quirks-Modus wechselt. [Weitere Informationen zum Deklarieren eines DOCTYPE.](#)

● Korrekt definierter Zeichensatz ^

Die Zeichencodierung muss deklariert werden. Dazu kann ein `<meta>`-Tag in den ersten 1024 Byte des HTML-Codes oder im HTTP-Antwortheader „Content-Type“ angegeben werden. [Weitere Informationen zum Deklarieren der Zeichencodierung.](#)

● Keine `unload`-Ereignis-Listener gefunden ^

Das `unload`-Ereignis wird nicht zuverlässig ausgelöst. Wenn der Listener darauf wartet, kann das Browseroptimierungen wie den Back-Forward-Cache beeinträchtigen. Bitte verwende stattdessen `pagehide`- oder `visibilitychange`-Ereignisse. [Weitere Informationen zum Entfernen von Event-Listnern](#)

● Vermeidet veraltete APIs ^

Verworfen APIs werden aus dem Browser entfernt. [Weitere Informationen zu verworfenen APIs.](#)

● Es wurden keine Browserfehler in der Konsole protokolliert ^

In der Konsole protokollierte Fehler weisen auf ungelöste Probleme hin. Sie können durch fehlgeschlagene Netzwerkanfragen und andere Browserprobleme verursacht werden. [Weitere Informationen](#)

● Keine Probleme im `Issues`-Bereich der Chrome-Entwicklertools ^

Im `Issues`-Bereich der Chrome-Entwicklertools wurden ungelöste Probleme protokolliert. Sie können durch fehlgeschlagene Netzwerkanfragen, unzureichende Sicherheitsmaßnahmen und andere Browser-Probleme verursacht sein. Öffne den Bereich mit den Problemen in Chrome-Entwicklertools, um weitere Details zu den einzelnen Problemen aufzurufen.

● Seite hat gültige Quellzuordnungen ^

Quellzuordnungen übersetzen reduzierten Code in den ursprünglichen Quellcode. Dies hilft Entwicklern beim Debugging in der Produktionsphase. Zusätzlich kann Lighthouse weitere Informationen liefern. Wir empfehlen, Quellzuordnungen bereitzustellen, um diese Vorteile zu nutzen. [Weitere Informationen zu Quellzuordnungen.](#)

URL	Zuordnungs-URL
localhost 1st Party	
<code>/search-component/search-component.js (localhost)</code>	<code>/search-component/search-component.js.map (localhost)</code>
Error: Failed fetching source map (404)	
...	...
<code>webcomponents/webcomponents.esm.js (localhost)</code>	<code>webcomponents/webcomponents.esm.js.map (localhost)</code>
<code>...webcomponents/p-f772284e.js (localhost)</code>	<code>...webcomponents/p-f772284e.js.map (localhost)</code>
<code>...webcomponents/p-dd3218d3.entry.js (localhost)</code>	<code>...webcomponents/p-dd3218d3.entry.js.map (localhost)</code>
<code>...webcomponents/p-d831bc11.entry.js (localhost)</code>	<code>...webcomponents/p-d831bc11.entry.js.map (localhost)</code>
<code>...webcomponents/p-c31cad0b.entry.js (localhost)</code>	<code>...webcomponents/p-c31cad0b.entry.js.map (localhost)</code>

URL	Zuordnungs-URL
...webcomponents/p-bc913fa4.entry.js (localhost)	...webcomponents/p-bc913fa4.entry.js.map (localhost)
...webcomponents/p-bbcf6480.entry.js (localhost)	...webcomponents/p-bbcf6480.entry.js.map (localhost)
...webcomponents/p-a5d22636.entry.js (localhost)	...webcomponents/p-a5d22636.entry.js.map (localhost)
...webcomponents/p-7d1868eb.entry.js (localhost)	...webcomponents/p-7d1868eb.entry.js.map (localhost)
...webcomponents/p-6a2de715.entry.js (localhost)	...webcomponents/p-6a2de715.entry.js.map (localhost)
...webcomponents/p-62030357.js (localhost)	...webcomponents/p-62030357.js.map (localhost)
...webcomponents/p-53dbf7d7.js (localhost)	...webcomponents/p-53dbf7d7.js.map (localhost)
...webcomponents/p-414dd07c.entry.js (localhost)	...webcomponents/p-414dd07c.entry.js.map (localhost)
...webcomponents/p-3f4649f5.js (localhost)	...webcomponents/p-3f4649f5.js.map (localhost)
...webcomponents/p-3ed4d64e.entry.js (localhost)	...webcomponents/p-3ed4d64e.entry.js.map (localhost)
...webcomponents/p-299c3b04.js (localhost)	...webcomponents/p-299c3b04.js.map (localhost)
...webcomponents/p-112455b1.js (localhost)	...webcomponents/p-112455b1.js.map (localhost)
...webcomponents/p-0e7ae7e8.entry.js (localhost)	...webcomponents/p-0e7ae7e8.entry.js.map (localhost)
...webcomponents/p-057a4d55.js (localhost)	...webcomponents/p-057a4d55.js.map (localhost)

NICHT ZUTREFFEND (2)

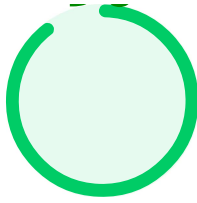
Ausblenden

- Schriftarten mit `font-display: optional` werden vorab geladen ^

Du solltest `optional`-Schriftarten vorab laden, damit sie von Erstbesuchern verwendet werden können. [Weitere Informationen zum Vorabladen von Schriftarten](#)

- JavaScript-Bibliotheken erkannt ^

Alle Front-End-JavaScript-Bibliotheken auf der Seite wurden erkannt. [Weitere Informationen.](#)



SEO

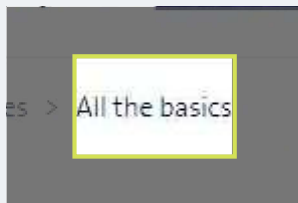
Mit diesen Prüfungen ist gewährleistet, dass bei deiner Seite grundlegende Tipps für die Suchmaschinenoptimierung berücksichtigt werden. Es gibt viele verschiedene Faktoren, die Lighthouse hier nicht bewertet und die sich auf das Such-Ranking deiner Seite auswirken können, einschließlich der [Core Web Vitals](#)-Performance. [Weitere Informationen zu Google Search Essentials](#)

CRAWLING UND INDEXIERUNG

▲ Links können nicht gecrawlt werden ^

Suchmaschinen verwenden möglicherweise href-Attribute für Links, um Websites zu crawlen. Das href-Attribut von Anchor-Elementen muss auf ein geeignetes Ziel verweisen, damit mehr Seiten auf der Website gefunden werden können. [Informationen dazu, wie Links für Crawler zugänglich gemacht werden](#)

Link kann nicht gecrawlt werden



All the basics

```
<a class="link" aria-current="location" aria-label="All the basics" aria-describedby="88">
```

```
body.sdx > div.main-wrapper > a#content  
<a id="content">
```

Damit deine Website in den Suchergebnissen angezeigt werden kann, benötigen Crawler Zugriff auf deine App.

ZUSÄTZLICHE ELEMENTE ZUR MANUELLEN ÜBERPRÜFUNG (1)

Ausblenden

○ Strukturierte Daten sind gültig ^

Du kannst das [Testtool für strukturierte Daten](#) und den [Lint für strukturierte Daten](#) ausführen, um strukturierte Daten zu validieren. [Weitere Informationen zu strukturierten Daten.](#)

Du kannst diese zusätzlichen Validierungen für deine Website ausführen, um weitere Best Practices für die SEO zu prüfen.

● Hat ein `<meta name="viewport">`-Tag mit `width` oder `initial-scale` ^

Ein `<meta name="viewport">` optimiert deine App für Bildschirmgrößen von Mobilgeräten und verhindert darüber hinaus [300-Millisekunden-Verzögerungen bei Nutzereingaben](#). [Weitere Informationen zur Verwendung des Darstellungsbereich-Meta-Tags](#). TBT

● Dokument enthält ein `<title>`-Element ^

Der Titel gibt Screenreader-Nutzern einen Überblick über die Seite. Nutzer von Suchmaschinen verlassen sich stark auf ihn, um zu entscheiden, ob eine Seite für ihre Suche relevant ist. [Weitere Informationen zu Dokumenttiteln](#).

● Dokument enthält eine Meta-Beschreibung ^

Meta-Beschreibungen können in die Suchergebnisse aufgenommen werden, um die Seiteninhalte kurz zusammenzufassen. [Weitere Informationen zu Meta-Beschreibungen](#).

● Seite hat einen gültigen HTTP-Statuscode ^

Seiten mit ungültigen HTTP-Statuscodes werden möglicherweise nicht richtig indiziert. [Weitere Informationen zu HTTP-Statuscodes](#).

● Links haben beschreibenden Text ^

Wenn du beschreibenden Linktext verwendest, können Suchmaschinen deine Inhalte besser verstehen. [Informationen zu barrierefreien Links](#).

● Seite ist nicht von Indexierung ausgeschlossen ^

Suchmaschinen können deine Seiten nicht in die Suchergebnisse aufnehmen, wenn sie nicht berechtigt sind, sie zu crawlen. [Weitere Informationen zu Crawler-Anweisungen](#).

● Bildelemente verfügen über `[alt]`-Attribute ^

Für informative Elemente sollte ein kurzer, beschreibenden alternativer Text verwendet werden. Dekorative Elemente können mit einem leeren ALT-Attribut ignoriert werden. [Weitere Informationen zum Attribut `alt`](#).

Dokument enthält ein gültiges `hreflang`-Element ^

Anhand von „hreflang“-Links können Suchmaschinen ermitteln, welche Version einer Seite sie in den Suchergebnissen für eine bestimmte Sprache oder Region anzeigen sollen. [Weitere Informationen zu hreflang.](#)

Dokument verwendet keine Plug-ins ^

Suchmaschinen können keine Plug-in-Inhalte indexieren. Außerdem werden Plug-ins auf vielen Geräten eingeschränkt oder nicht unterstützt. [Weitere Informationen zum Vermeiden von Plug-ins.](#)

NICHT ZUTREFFEND (4)

Ausblenden

robots.txt ist gültig ^

Wenn deine robots.txt-Datei fehlerhaft ist, können Crawler möglicherweise nicht nachvollziehen, wie deine Website gecrawlt oder indexiert werden soll. [Weitere Informationen zu robots.txt.](#)

Dokument enthält ein gültiges `rel=canonical`-Element ^

Über kanonische Links wird angegeben, welche URL in den Suchergebnissen angezeigt werden soll. [Weitere Informationen zu kanonischen Links.](#)

Dokument enthält gut lesbare Schriftgrößen ^

Schriftgrößen unter 12 px sind zu klein. Nutzer von Mobilgeräten müssen den Text mit den Fingern heranzoomen, um ihn lesen zu können. Achte darauf, dass mehr als 60 % des Seitentextes eine Schriftgröße von mindestens 12 px haben. [Weitere Informationen zu lesbaren Schriftgrößen.](#)

Größe von Tippzielen ist richtig eingestellt ^


Interaktive Elemente wie Schaltflächen und Links sollten groß genug sein (48 × 48 px) und über genügend Zwischenraum verfügen, damit sie leicht angetippt werden können. Sie dürfen sich nicht mit anderen Elementen überschneiden. [Weitere Informationen zu Tippzielen.](#)



PWA

Diese Prüfungen dienen dazu, die einzelnen Aspekte einer progressiven Web-App zu überprüfen. [Weitere Informationen dazu, was eine gute progressive Web-App ausmacht](#)

INSTALLIERBAR

-  Das Manifest der Web-App oder der Service Worker erfüllt die Anforderungen an die Installierbarkeit nicht — 1 Grund ^

Der Service Worker ermöglicht es deiner App, viele Funktionen von progressiven Web-Apps zu nutzen, beispielsweise den Offlinemodus, das Hinzufügen zum Startbildschirm und Push-Benachrichtigungen. Bei ordnungsgemäßen Service Worker- und Manifestimplementierungen können Browser Nutzer direkt auffordern, deine Web-App zum Startbildschirm hinzuzufügen. Das kann zu mehr Interaktionen führen. [Weitere Informationen zu Anforderungen an die Installierbarkeit für Manifeste](#)

Grund für den Fehler

Die Seite hat keine Manifest-URL <link>

PWA-OPTIMIERT

-  Es wurde kein Service Worker erkannt, der die Seite und `start_url` kontrolliert ^

Der Service Worker ermöglicht es deiner App, viele Funktionen von progressiven Web-Apps zu nutzen, beispielsweise den Offlinemodus, das Hinzufügen zum Startbildschirm und Push-Benachrichtigungen. [Weitere Informationen.](#)

-  Nicht für einen benutzerdefinierten Startbildschirm konfiguriert ^
Failures: No manifest was fetched.

Wenn du deinen Startbildschirm passend zum Design deiner App gestaltest, vermittelst du den Nutzern schon beim Ladevorgang einen hochwertigen Eindruck. [Weitere Informationen zu Startbildschirmen.](#)

▲ Legt keine Designfarbe für die Adressleiste fest.

Failures: No manifest was fetched, No ``<meta name="theme-color">`` tag found.

Die Adressleiste des Browsers kann an das Design deiner Website angepasst werden. [Weitere Informationen zum Gestalten der Adressleiste.](#)

○ Inhalt hat die richtige Größe für den Darstellungsbereich

Wenn die Breite deiner App-Inhalte nicht mit der des Darstellungsbereichs übereinstimmt, ist deine App möglicherweise nicht für Bildschirme von Mobilgeräten optimiert. [Informationen zum Anpassen der Größe von Inhalten an den Darstellungsbereich.](#)

● Hat ein `<meta name="viewport">`-Tag mit `width` oder `initial-scale`

Ein `<meta name="viewport">` optimiert deine App für Bildschirmgrößen von Mobilgeräten und verhindert darüber hinaus [300-Millisekunden-Verzögerungen bei Nutzereingaben](#). [Weitere Informationen zur Verwendung des Darstellungsbereich-Meta-Tags.](#) TBT

▲ Das Manifest hat kein maskierbares Symbol **No manifest was fetched**

Mit einem maskierbaren Symbol wird die Bildform beim Installieren der App auf einem Gerät vollständig gefüllt, ohne dass es zu einem Letterbox-Effekt kommt. [Informationen zu maskierbaren Manifestsymbolen.](#)

ZUSÄTZLICHE ELEMENTE ZUR MANUELLEN ÜBERPRÜFUNG (3)

Ausblenden

○ Website funktioniert auf verschiedenen Browsern

Damit du möglichst viele Nutzer erreichen kannst, sollte deine Website mit allen gängigen Browsern kompatibel sein. [Weitere Informationen.](#)

○ Seitenübergänge vermitteln nicht das Gefühl von übermäßigen Ladezeiten

Übergänge sollten sich auch bei einer langsamen Netzwerkverbindung schnell anfühlen. Dies ist entscheidend dafür, wie der Nutzer die Leistung wahrnimmt. [Weitere Informationen zu Seitenübergängen.](#)

○ Jede Seite hat eine URL

Für die einzelnen Seiten sollten Deeplinks erstellt werden können. Achte darauf, dass die entsprechenden URLs eindeutig sind, sodass sich die Seiten in sozialen Netzwerken leichter teilen lassen. [Weitere Informationen zum Angeben von Deeplinks.](#)

Diese Prüfungen sind laut der grundlegenden [PWA-Checkliste](#) erforderlich, werden von Lighthouse jedoch nicht automatisch durchgeführt. Sie haben zwar keine Auswirkung auf deine Leistungsbewertung, aber es ist wichtig, sie

manuell durchzuführen.

 Captured at 13. Juni 2023, 11:59 MESZ	 Desktopemulation with Lighthouse 10.1.1	 Einzelner Seitenaufbau
 Erster Seitenaufbau	 Benutzerdefinierte Drosselung	 Using Chromium 114.0.0.0 with devtools

Generated by **Lighthouse** 10.1.1 | [Problem melden](#)

Lighthouse Report mit Suche: Snapshot Suchresultate

○ 3/4

Leistung

● 22/23

Barrierefreiheit

● 4/4

Best Practices

■ 5/6

SEO



0-49



50-89



90-100

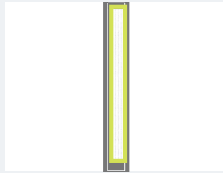
○ 3/4

Leistung

DIAGNOSE

▲ Übermäßige DOM-Größe vermeiden — 3.321 Elemente

Ein großes DOM führt zu hoher Arbeitsspeichernutzung, langwierigen [Stilberechnungen](#) und kostspieligen [dynamischen Umbrüchen im Layout](#). [Informationen zum Vermeiden eines zu großen DOMs.](#)

Statistik	Element	Wert
DOM-Elemente insgesamt		3.321
Maximale DOM-Tiefe	Accessibility 	28
Maximale Anzahl von untergeordneten Elementen	 .icon-account .icon-arrow-down .icon-arrow-left .icon-arrow-right .icon-arr... <div class="row margin-bottom-2--clear">	631

Weitere Informationen zur Leistung deiner App findest du hier. Diese Angaben haben keinen [direkten Einfluss](#) auf die Leistungsbewertung.

BESTANDENE PRÜFUNGEN (3)

Ausblenden

● Bildelemente haben eine explizite [width](#) und [height](#)

Lege eine explizite Breite und Höhe für Bildelemente fest, um Layoutverschiebungen zu reduzieren und den CLS-Wert zu verbessern. [Informationen zum Festlegen von Bildabmessungen](#)

● Hat ein `<meta name="viewport">`-Tag mit `width` oder `initial-scale` ^

Ein `<meta name="viewport">` optimiert deine App für Bildschirmgrößen von Mobilgeräten und verhindert darüber hinaus [300-Millisekunden-Verzögerungen bei Nutzereingaben](#). [Weitere Informationen zur Verwendung des Darstellungsbereich-Meta-Tags](#).

● Bilder entsprechen der angezeigten Größe ^

Stelle Bilder bereit, die eine angemessene Größe haben, um mobile Daten zu sparen und die Ladezeit zu verbessern. [Weitere Informationen](#).

● 22/23

Barrierefreiheit

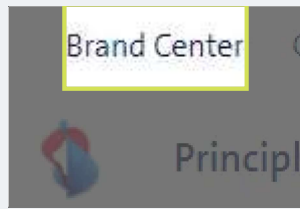
Mit diesen Prüfungen erfährst du, [wie du die Barrierefreiheit deiner Web-App verbesserst](#). Nur bestimmte Probleme mit der Barrierefreiheit können durch automatisierte Tests erkannt werden. Deshalb ist es empfehlenswert, zusätzlich manuelle Tests durchzuführen.

TABELLEN UND LISTEN

▲ Listenelemente (``) befinden sich nicht in übergeordneten ``-, ``- oder `<menu>`-Elementen. ^

Listenelemente (``) müssen sich in einem übergeordneten ``-, ``- oder `<menu>`-Element befinden, damit sie von Screenreadern richtig angesagt werden können. [Weitere Informationen zum Strukturieren von Listen](#).

Fehlerhafte Elemente



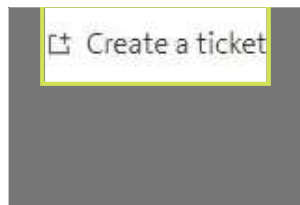
Brand Center
<li class="item leaf">



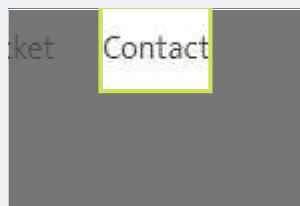
Git Repo
<li class="item leaf">



Stay up-to-date
<li class="item leaf">



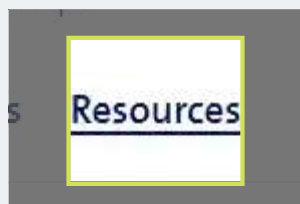
Create a ticket
<li class="item leaf">



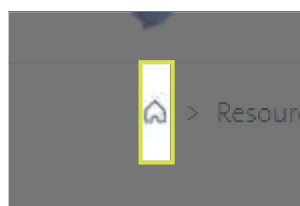
Contact
<li class="item leaf">



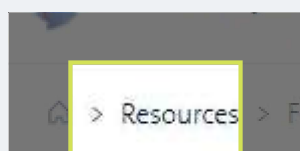
Principles
<li class="item leaf">



Resources
<li class="item active">

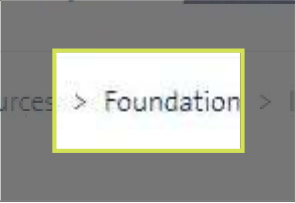
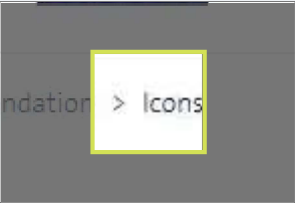


icon-home
<li class="item">



Resources
<li class="item">

Fehlerhafte Elemente

	Foundation <code><li class="item"></code>
	Icons <code><li class="item"></code>

Hier findest du Möglichkeiten, um das Lesen von Daten in Tabellen oder Listen mit Hilfstechnologie wie Screenreadern zu verbessern.

ZUSÄTZLICHE ELEMENTE ZUR MANUELLEN ÜBERPRÜFUNG (10)

Ausblenden

The page has a logical tab order ^

Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. [Learn more about logical tab ordering.](#)

Interactive controls are keyboard focusable ^

Custom interactive controls are keyboard focusable and display a focus indicator. [Learn how to make custom controls focusable.](#)

Interactive elements indicate their purpose and state ^

Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. [Learn how to decorate interactive elements with affordance hints.](#)

The user's focus is directed to new content added to the page ^

If new content, such as a dialog, is added to the page, the user's focus is directed to it. [Learn how to direct focus to new content.](#)

User focus is not accidentally trapped in a region ^

A user can tab into and out of any control or region without accidentally trapping their focus. [Learn how to avoid focus traps.](#)

Custom controls have associated labels ^

Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. [Learn more about custom controls and labels.](#)

Custom controls have ARIA roles ^

Custom interactive controls have appropriate ARIA roles. [Learn how to add roles to custom controls.](#)

Visual order on the page follows DOM order ^

DOM order matches the visual order, improving navigation for assistive technology. [Learn more about DOM and visual ordering.](#)

Offscreen content is hidden from assistive technology ^

Offscreen content is hidden with display: none or aria-hidden=true. [Learn how to properly hide offscreen content.](#)

HTML5 landmark elements are used to improve navigation ^

Landmark elements (<main>, <nav>, etc.) are used to improve the keyboard navigation of the page for assistive technology. [Learn more about landmark elements.](#)

Diese Prüfungen sind für Bereiche vorgesehen, für die automatische Testtools nicht geeignet sind. Weitere Informationen findest du in unserem Leitfaden zur [Durchführung einer Prüfung auf Barrierefreiheit.](#)

BESTANDENE PRÜFUNGEN (22)

Ausblenden

[aria-*]-Attribute entsprechen ihren Rollen ^

Jede ARIA-role unterstützt eine bestimmte Untergruppe von aria-*-Attributen. Wenn sie jedoch falsch zugeordnet sind, werden die aria-*-Attribute ungültig. [Informationen zum Zuordnen von ARIA-Attributen zu ihren Rollen.](#)

[aria-hidden="true"] ist in dem Dokument <body> nicht vorhanden ^

Hilfstechnologien wie Screenreader funktionieren nicht richtig, wenn für den <body> des Dokuments aria-hidden="true" festgelegt ist. [Informationen zu den Auswirkungen von aria-hidden auf den Textbereich des Dokuments.](#)

[role]-Elemente verfügen über alle erforderlichen [aria-*]-Attribute ^

Für einige ARIA-Rollen sind Attribute erforderlich, die Screenreadern den Zustand des Elements beschreiben. [Weitere Informationen zu Rollen und erforderlichen Attributen.](#)

- Die Elemente mit einer ARIA-`[role]`, deren untergeordnete Elemente eine bestimmte `[role]` enthalten müssen, haben alle erforderlichen untergeordneten Elemente. ^

Einige übergeordnete ARIA-Rollen müssen bestimmte untergeordnete Rollen enthalten, damit sie die beabsichtigten Hilfsfunktionen erfüllen können. [Weitere Informationen zu Rollen und erforderlichen untergeordneten Elementen.](#)

- `[role]`-Elemente sind ihren jeweils erforderlichen übergeordneten Elementen untergeordnet ^

Einige untergeordnete ARIA-Rollen müssen in bestimmten übergeordneten Rollen enthalten sein, damit sie die beabsichtigten Hilfsfunktionen erfüllen können. [Weitere Informationen zu ARIA-Rollen und erforderlichen übergeordneten Elementen.](#)

- `[role]`-Werte sind gültig ^

Für ARIA-Rollen müssen gültige Werte angegeben sein, damit sie die beabsichtigten Hilfsfunktionen erfüllen können. [Weitere Informationen zu gültigen ARIA-Rollen.](#)

- `[aria-*]`-Attribute weisen gültige Werte auf ^

Hilfstechnologien wie Screenreader können ARIA-Attribute mit ungültigen Werten nicht interpretieren. [Weitere Informationen zu gültigen Werten für ARIA-Attribute.](#)

- `[aria-*]`-Attribute sind gültig und richtig geschrieben ^

Hilfstechnologien wie Screenreader können ARIA-Attribute mit ungültigen Namen nicht interpretieren. [Weitere Informationen zu gültigen ARIA-Attributen.](#)

- Die Namen der Schaltflächen sind für Screenreader zugänglich ^

Wenn eine Schaltfläche keinen barrierefreien Namen hat, wird sie von Screenreadern nur als „Schaltfläche“ angesagt. Dadurch ist sie für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Informationen zum barrierefreien Gestalten von Schaltflächen.](#)

● ARIA-IDs sind eindeutig ^

Der Wert einer ARIA-ID muss eindeutig sein, damit andere Instanzen nicht von Hilfstechnologien übersehen werden. [Informationen zum Korrigieren doppelter ARIA-IDs.](#)

● Formularelemente sind mit Labels verknüpft ^

Durch Labels wird gewährleistet, dass Steuerelemente für Formulare von Hilfstechnologien wie Screenreadern richtig angesagt werden. [Weitere Informationen zu Labels für Formularelemente.](#)

● `[user-scalable="no"]` wird nicht im `<meta name="viewport">`-Element verwendet und das `[maximum-scale]`-Attribut ist nicht kleiner als 5. ^

Wenn du die Zoomfunktion deaktivierst, können Nutzer mit eingeschränktem Sehvermögen, die auf die Bildschirmvergrößerung angewiesen sind, den Inhalt einer Webseite nicht richtig sehen. [Weitere Informationen zum Darstellungsbereich-Meta-Tag.](#)

● `[aria-hidden="true"]`-Elemente enthalten keine fokussierbaren Unterelemente ^

Fokussierbare Nachfolgerelemente in einem `[aria-hidden="true"]`-Element führen dazu, dass Nutzer von Hilfstechnologien wie Screenreadern solche interaktiven Elemente nicht verwenden können. [Informationen zu den Auswirkungen von `aria-hidden` auf fokussierbare Elemente.](#)

● Das Kontrastverhältnis von Hintergrund- und Vordergrundfarben ist ausreichend ^

Text mit geringem Kontrast ist für viele Nutzer schlecht oder gar nicht lesbar. [Informationen zu einem ausreichenden Farbkontrast.](#)

● Dokument enthält ein `<title>`-Element ^

Der Titel gibt Screenreader-Nutzern einen Überblick über die Seite. Nutzer von Suchmaschinen verlassen sich stark auf ihn, um zu entscheiden, ob eine Seite für ihre Suche relevant ist. [Weitere Informationen zu Dokumenttiteln.](#)

● `[id]`-Attribute zu aktiven, fokussierbaren Elementen sind eindeutig ^

Für alle fokussierbaren Elemente ist eine eindeutige `id` erforderlich, damit sie von Hilfstechnologien erkannt werden können. [Informationen zum Korrigieren doppelter `ids`.](#)

● `<html>`-Element hat ein `[lang]`-Attribut ^

Wenn für eine Seite kein `lang`-Attribut angegeben ist, nehmen Screenreader an, dass sie in der Standardsprache vorliegt, die der Nutzer beim Einrichten des Screenreaders ausgewählt hat. Ist das nicht der Fall, gibt der Screenreader den Inhalt der Seite möglicherweise falsch aus. [Weitere Informationen zum `lang`-Attribut.](#)

● Das `<html>`-Element hat einen gültigen Wert für sein `[lang]`-Attribut ^

Wenn ein gültiger [BCP-47-Sprachcode](#) angegeben wird, kann der Text von einem Screenreader korrekt wiedergegeben werden. [Informationen zur Verwendung des Attributs `lang`.](#)

● Links haben einen leicht erkennbaren Namen ^

Linktext, der erkennbar, einzigartig und fokussierbar ist, erleichtert Screenreader-Nutzern die Verwendung. Dies gilt auch für alternativen Text für Bilder, die als Links verwendet werden. [Informationen zu barrierefreien Links.](#)

● Listen enthalten nur ``-Elemente und Elemente zur Skriptunterstützung (`<script>` sowie `<template>`). ^

Screenreader sagen Listen auf bestimmte Art und Weise an. Wenn die Liste richtig strukturiert ist, kann der Screenreader sie besser ausgeben. [Weitere Informationen zum Strukturieren von Listen.](#)

● Kein Element hat einen `[tabindex]`-Wert größer als 0 ^

Ein Wert größer als 0 impliziert eine explizite Navigationsanordnung. Das ist zwar technisch möglich, aber für Nutzer, die auf Hilfstechnologien angewiesen sind, häufig frustrierend. [Weitere Informationen zum Attribut `tabindex`.](#)

● Überschriftenelemente werden in einer fortlaufenden absteigenden Reihenfolge angezeigt ^

Richtig geordnete Überschriften, die keine Ebenen überspringen, geben der Seite eine semantische Struktur. Nutzer von Hilfstechnologien können sich so leichter auf der Seite zurechtfinden und die Inhalte besser verstehen. [Weitere Informationen zur Reihenfolge von Überschriften.](#)

NICHT ZUTREFFEND (21)

Ausblenden

 [accesskey]-Werte sind eindeutig ^

Mithilfe von Tastenkombinationen können Nutzer schnell den Fokus auf einen Bereich der Seite verschieben. Damit die Navigation richtig funktioniert, darf jede Tastenkombination nur einmal vergeben sein. [Weitere Informationen zu Tastenkombinationen.](#)

 button-, link- und menuitem-Elemente haben zugängliche Namen ^

Wenn ein Element keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Informationen zum barrierefreien Gestalten von Befehlselementen.](#)

 ARIA-Eingabefelder haben zugängliche Namen ^

Wenn ein Eingabefeld keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Weitere Informationen zu Labels für Eingabefelder.](#)

 ARIA meter-Elemente haben zugängliche Namen ^

Wenn ein Messtool-Element keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Weitere Informationen zum Benennen von meter-Elementen](#)

 ARIA progressbar-Elemente haben zugängliche Namen ^

Wenn ein progressbar-Element keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Informationen zum Kennzeichnen von progressbar-Elementen.](#)

 ARIA-Ein-/Aus-Schaltflächen haben zugängliche Namen ^

Wenn eine Ein-/Aus-Schaltfläche keinen barrierefreien Namen hat, wird sie von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist sie für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Weitere Informationen zu Ein-/Aus-Schaltflächen.](#)

 ARIA tooltip-Elemente haben zugängliche Namen ^

Wenn ein Kurzinfo-Element keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Weitere Informationen zum Benennen von tooltip-Elementen](#)

○ ARIA `treeitem`-Elemente haben zugängliche Namen ^

Wenn ein `treeitem`-Element keinen barrierefreien Namen hat, wird es von Screenreadern mit einer allgemeinen Bezeichnung angesagt. Dadurch ist es für Nutzer, die auf Screenreader angewiesen sind, unbrauchbar. [Weitere Informationen zum Kennzeichnen von `treeitem`-Elementen.](#)

○ Die Seite enthält eine Überschrift, einen Link zum Überspringen oder einen Landmark-Bereich ^

Wenn Tastaturnutzer Inhalte umgehen können, die sich wiederholen, sorgt das für eine effizientere Navigation. [Weitere Informationen zum Umgehen von Blockierungen.](#)

○ `<d1>`-Elemente enthalten ausschließlich Gruppen aus `<dt>`- und `<dd>`-Elementen sowie `<script>`-, `<template>`- oder `<div>`-Elemente, die richtig angeordnet sind. ^

Wenn Definitionslisten nicht korrekt mit Markup versehen sind, kann es zu verwirrenden oder ungenauen Screenreader-Ausgaben kommen. [Informationen zum Strukturieren von Definitionslisten.](#)

○ Definitionlistenelemente sind in `<d1>`-Elemente eingefasst ^

Definitionlistenelemente (`<dt>` und `<dd>`) müssen in ein übergeordnetes `<d1>`-Element eingefasst sein, damit sie von Screenreadern richtig angesagt werden können. [Informationen zum Strukturieren von Definitionslisten.](#)

○ Kein Formularfeld hat mehrere Labels ^

Formularfelder mit mehreren Labels werden von Hilfstechnologien wie Screenreadern unter Umständen missverständlich angesagt, da sie entweder das erste, das letzte oder alle Labels verwenden. [Informationen zur Verwendung von Formularlabels.](#)

○ `<frame>`- oder `<iframe>`-Elemente verfügen über einen Titel ^

Screenreader-Nutzer sind auf Frametitel angewiesen, die die Inhalte von Frames beschreiben. [Weitere Informationen zu Frametiteln.](#)

○ Bildelemente verfügen über `[alt]`-Attribute ^

Für informative Elemente sollte ein kurzer, beschreibenden alternativer Text verwendet werden. Dekorative Elemente können mit einem leeren ALT-Attribut ignoriert werden. [Weitere Informationen zum Attribut `alt`.](#)

○ `<input type="image">`-Elemente haben `[alt]`-Text ^

Wenn ein Bild als `<input>`-Schaltfläche verwendet wird, kann alternativer Text Screenreader-Nutzern helfen, den Zweck der Schaltfläche besser zu verstehen. [Informationen zum Alt-Text für Eingabebilder.](#)

- Dieses Dokument verwendet `<meta http-equiv="refresh">` nicht ^

Nutzer rechnen nicht damit, dass eine Seite automatisch aktualisiert wird. Außerdem wird dadurch der Fokus wieder auf den Seitenanfang verschoben. Das kann für den Nutzer frustrierend oder irritierend sein. [Weitere Informationen zum Meta-Tag „Refresh“.](#)

- `<object>`-Elemente haben alternativen Text ^

Screenreader können lediglich Textinhalte interpretieren. Wenn du `<object>`-Elementen alternativen Text hinzufügst, können Screenreader-Nutzer besser verstehen, was diese Elemente darstellen. [Weitere Informationen zum Alt-Text für object-Elemente.](#)

- Zellen in einem `<table>`-Element, die das Attribut `[headers]` enthalten, verweisen auf Zellen in derselben Tabelle. ^

Screenreader bieten Funktionen, die die Navigation in Tabellen vereinfachen. Wenn du dafür sorgst, dass `<td>`-Zellen, die das Attribut `[headers]` verwenden, nur auf andere Zellen in derselben Tabelle verweisen, kann dies für Screenreader-Nutzer hilfreich sein. [Weitere Informationen zum Attribut headers.](#)

- Für `<th>`-Elemente und Elemente mit `[role="columnheader"/"rowheader"]` sind Datenzellen vorhanden, die sie beschreiben. ^

Screenreader bieten Funktionen, die die Navigation in Tabellen vereinfachen. Wenn du dafür sorgst, dass Tabellenüberschriften immer auf bestimmte Zellen verweisen, kann dies für Screenreader-Nutzer hilfreich sein. [Weitere Informationen zu Tabellenüberschriften.](#)

- `[lang]`-Attribute weisen einen gültigen Wert auf ^

Wenn ein gültiger [BCP-47-Sprachcode](#) für Elemente angegeben wird, kann der Text besser von Screenreadern ausgesprochen werden. [Informationen zur Verwendung des Attributs lang.](#)

- `<video>`-Elemente enthalten ein `<track>`-Element mit `[kind="captions"]` ^

Wenn ein Video Untertitel enthält, können gehörlose und hörgeschädigte Nutzer die Informationen im Video besser verstehen. [Weitere Informationen zu Untertiteln in Videos.](#)

● 4/4

Best Practices

BESTANDENE PRÜFUNGEN (4)

Ausblenden

● Allows users to paste into input fields ^

Preventing input pasting is a UX anti-pattern, and undermines good security policy. [Learn more about user-friendly input fields.](#)

● Zeigt Bilder mit einem korrekten Seitenverhältnis an ^

Die Bildgröße sollte dem natürlichen Seitenverhältnis entsprechen. [Weitere Informationen zum Seitenverhältnis von Bildern.](#)

● Stellt Bilder mit angemessener Auflösung bereit ^

Die originalen Abmessungen eines Bildes sollten proportional zu der Displaygröße und dem Pixel-Verhältnis sein, damit das Bild optimal angezeigt wird. [Weitere Informationen zu responsiven Bildern.](#)

● Seite verfügt über den HTML-DOCTYPE ^

Wenn du einen DOCTYPE angibst, verhinderst du, dass der Browser zum Quirks-Modus wechselt. [Weitere Informationen zum Deklarieren eines DOCTYPE.](#)

NICHT ZUTREFFEND (1)

Ausblenden

○ JavaScript-Bibliotheken erkannt ^

Alle Front-End-JavaScript-Bibliotheken auf der Seite wurden erkannt. [Weitere Informationen.](#)

5/6

SEO

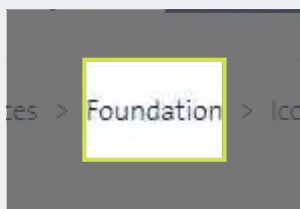
Mit diesen Prüfungen ist gewährleistet, dass bei deiner Seite grundlegende Tipps für die Suchmaschinenoptimierung berücksichtigt werden. Es gibt viele verschiedene Faktoren, die Lighthouse hier nicht bewertet und die sich auf das Such-Ranking deiner Seite auswirken können, einschließlich der [Core Web Vitals](#)-Performance. [Weitere Informationen zu Google Search Essentials](#)

CRAWLING UND INDEXIERUNG

▲ Links können nicht gecrawlt werden

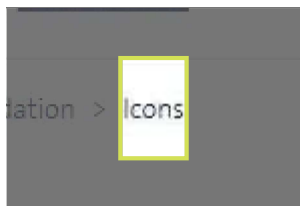
Suchmaschinen verwenden möglicherweise href-Attribute für Links, um Websites zu crawlen. Das href-Attribut von Anchor-Elementen muss auf ein geeignetes Ziel verweisen, damit mehr Seiten auf der Website gefunden werden können. [Informationen dazu, wie Links für Crawler zugänglich gemacht werden](#)

Link kann nicht gecrawlt werden



Foundation

```
<a class="link" aria-label="Foundation" aria-describedby="88">
```



Icons

```
<a class="link" aria-current="location" aria-label="Icons" aria-describedby="89">
```

body.sdx > div.main-wrapper > a#content

```
<a id="content">
```

Link kann nicht gecrawlt werden

Download

```
<a class="link icon-download-cloud">
```

Damit deine Website in den Suchergebnissen angezeigt werden kann, benötigen Crawler Zugriff auf deine App.

ZUSÄTZLICHE ELEMENTE ZUR MANUELLEN ÜBERPRÜFUNG (1)

Ausblenden

Strukturierte Daten sind gültig ^

Du kannst das [Testtool für strukturierte Daten](#) und den [Lint für strukturierte Daten](#) ausführen, um strukturierte Daten zu validieren. [Weitere Informationen zu strukturierten Daten.](#)

Du kannst diese zusätzlichen Validierungen für deine Website ausführen, um weitere Best Practices für die SEO zu prüfen.

BESTANDENE PRÜFUNGEN (5)

Ausblenden

Hat ein `<meta name="viewport">`-Tag mit `width` oder `initial-scale` ^

Ein `<meta name="viewport">` optimiert deine App für Bildschirmgrößen von Mobilgeräten und verhindert darüber hinaus [300-Millisekunden-Verzögerungen bei Nutzereingaben](#). [Weitere Informationen zur Verwendung des Darstellungsbereich-Meta-Tags.](#)

Dokument enthält ein `<title>`-Element ^

Der Titel gibt Screenreader-Nutzern einen Überblick über die Seite. Nutzer von Suchmaschinen verlassen sich stark auf ihn, um zu entscheiden, ob eine Seite für ihre Suche relevant ist. [Weitere Informationen zu Dokumenttiteln.](#)

Dokument enthält eine Meta-Beschreibung ^

Meta-Beschreibungen können in die Suchergebnisse aufgenommen werden, um die Seiteninhalte kurz zusammenzufassen. [Weitere Informationen zu Meta-Beschreibungen.](#)

Links haben beschreibenden Text ^

Wenn du beschreibenden Linktext verwendest, können Suchmaschinen deine Inhalte besser verstehen. [Informationen zu barrierefreien Links.](#)

Dokument verwendet keine Plug-ins ^

Suchmaschinen können keine Plug-in-Inhalte indexieren. Außerdem werden Plug-ins auf vielen Geräten eingeschränkt oder nicht unterstützt. [Weitere Informationen zum Vermeiden von Plug-ins.](#)

NICHT ZUTREFFEND (4)

Ausblenden

 robots.txt ist gültig ^

Wenn deine robots.txt-Datei fehlerhaft ist, können Crawler möglicherweise nicht nachvollziehen, wie deine Website gecrawlt oder indexiert werden soll. [Weitere Informationen zu robots.txt.](#)

 Bildelemente verfügen über [alt]-Attribute ^


Für informative Elemente sollte ein kurzer, beschreibenden alternativer Text verwendet werden. Dekorative Elemente können mit einem leeren ALT-Attribut ignoriert werden. [Weitere Informationen zum Attribut alt.](#)

 Dokument enthält gut lesbare Schriftgrößen ^

Schriftgrößen unter 12 px sind zu klein. Nutzer von Mobilgeräten müssen den Text mit den Fingern heranzoomen, um ihn lesen zu können. Achte darauf, dass mehr als 60 % des Seitentextes eine Schriftgröße von mindestens 12 px haben. [Weitere Informationen zu lesbaren Schriftgrößen.](#)


 Größe von Tippzielen ist richtig eingestellt ^

Interaktive Elemente wie Schaltflächen und Links sollten groß genug sein (48 × 48 px) und über genügend Zwischenraum verfügen, damit sie leicht angetippt werden können. Sie dürfen sich nicht mit anderen Elementen überschneiden. [Weitere Informationen zu Tippzielen.](#)


 Captured at 13. Juni 2023, 12:13
MESZ

 Desktopemulation with Lighthouse
10.1.1

 Einzelner Seitenaufbau

 Erster Seitenaufbau

 Benutzerdefinierte Drosselung

 Using Chromium 114.0.0.0 with
devtools

Generated by **Lighthouse** 10.1.1 | [Problem melden](#)