

# No-Code im Webdesign

Vergleich gängiger Systeme und Webdesignumsetzung  
auf Basis von No-Code-Entwicklungsplattformen

## Masterarbeit

Ausgeführt zum Zweck der Erlangung des akademischen Grades  
**Dipl.-Ing. für technisch-wissenschaftliche Berufe**

am Masterstudiengang Digital Design an der Fachhochschule St. Pölten,  
**Masterklasse Grafik Design**

von:

**Florian Weinhappl, BSc**

dd201529

Betreuer\*in: FH-Prof. Dipl.-Ing. Dr. Kerstin Blumenstein, BSc  
Zweitbetreuer\*in: Teresa Sposato, Bakk. MA

Wien, 09.09.2022

# Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

- ich dieses Thema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der vom Begutachter bzw. der Begutachterin beurteilten Arbeit überein.

Wien, 09.09.2022



.....  
Ort, Datum

.....  
Unterschrift

# Kurzfassung

No-Code- und Low-Code-Plattformen ermöglichen heutzutage die Erstellung von Websites und Webanwendungen über grafische Benutzeroberflächen ohne oder mit nur geringem Einsatz von Code. Am Markt sind unterschiedliche Systeme erhältlich, welche eine einfache und intuitive Bedienung versprechen. Der in diesem Kontext neu etablierte Begriff des „Citizen Developers“ beschreibt die Zielgruppe dieser Plattformen. Personen, welche über ausreichend technisches Verständnis verfügen, jedoch keine Programmierkenntnisse besitzen.

Ziel dieser Arbeit ist es herauszufinden, ob eine moderne, responsive Website vollständig mit einer No-Code-Plattform umgesetzt werden kann, ohne dafür eigenen Code zu schreiben. Um die Forschungsfrage beantworten zu können, wurden im Laufe dieser Masterarbeit vier gängige No-Code-Plattformen einem Vergleich hinsichtlich ihrer Funktionen unterzogen, sowie in einem Experiment die Umsetzbarkeit einer bestehenden Website untersucht.

Die Ergebnisse der empirischen Untersuchung zeigen eine Umsetzbarkeit von 87 bis 98 Prozent, je nach verwendeter No-Code-Plattform. Einschränkungen, welche nicht mit den Bordmitteln der Plattformen behoben werden können und den Einsatz von eigenem Code erfordern, wurden lediglich in Bezug auf Details, welche die Hauptfunktionalität der Website nicht beeinträchtigen, festgestellt.

# Abstract

Nowadays, no-code and low-code platforms enable the creation of websites and web applications via graphical user interfaces with little or no code. Various systems are available on the market that promise simple and intuitive usage. The newly established term "citizen developer" describes the target group of these platforms. People who have sufficient technical understanding, but no programming skills.

The aim of this work is to find out whether a modern, responsive website can be completely implemented with a no-code platform without writing custom code. In order to be able to answer the research question, in the course of this masters thesis, four popular no-code platforms were subjected to a comparison with regard to their functions, furthermore the practicability of implementing an existing website was tested in an experiment.

The results of the empirical study show a feasibility of 87 to 98 percent, depending on the no-code platform used. Restrictions that cannot be remedied with the platforms on-board resources and require the use of custom code were only found with regard to details that do not affect the main functionality of the website.

# Inhaltsverzeichnis

<b>Ehrenwörtliche Erklärung</b>	<b>II</b>
<b>Kurzfassung</b>	<b>III</b>
<b>Abstract</b>	<b>IV</b>
<b>Inhaltsverzeichnis</b>	<b>V</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Ziele und Forschungsfragen	3
1.2 Methodik	3
1.3 Aufbau der Arbeit	5
<b>2 Grundlagen des Webdesigns und der Webentwicklung</b>	<b>7</b>
2.1 Elemente des User Interfaces	7
2.1.1 Layout und Struktur	7
2.1.2 Typografie	11
2.1.3 Farbe	13
2.1.4 Multimedia (Bilder, Grafiken, Videos)	14
2.1.5 Animationen und Interaktionen	16
2.2 Webtechnologien und -standards	18
2.2.1 HTML	18
2.2.2 CSS	19
2.2.3 JavaScript	20
2.2.4 PHP	21
2.3 Best Practices	21
2.3.1 Barrierefreiheit	21
2.3.2 Suchmaschinenoptimierung	24
2.3.3 Performance-Optimierung	25
<b>3 Web-Content-Management</b>	<b>27</b>
3.1 Begriffsdefinition	27
3.1.1 Content	27
3.1.2 Content-Management	28
3.1.3 Content-Management-Systeme	29
3.1.4 Arten von Content-Management-Systemen	29
3.2 Web-Content-Management-Systeme	31
3.2.1 Funktionsweise	31

3.2.2	Vor- und Nachteile	33
3.3	Das Web-CMS WordPress	35
3.3.1	Websitegestaltung mit WordPress	37
3.3.2	Block-Editor und Full-Site-Editing	38
3.4	Page-Builder-Plugins für WordPress	40
<b>4</b>	<b>Low-Code- / No-Code-Programmierstil</b>	<b>42</b>
4.1	Historische Einordnung und Entstehung	43
4.2	Unterschied Low-Code & No-Code	44
4.3	Zielgruppe	45
<b>5</b>	<b>No-Code-Plattformen</b>	<b>47</b>
5.1	Marktüberblick	47
5.2	Aufbau und Charakteristik	49
5.2.1	Bereitstellung in der Cloud	50
5.3	Nutzen und Limitierungen	50
<b>6</b>	<b>Empirische Untersuchung</b>	<b>53</b>
6.1	Recherche	53
6.2	Forschungsmethoden & Evaluation	56
6.3	Forschungsdesign	58
<b>7</b>	<b>Ergebnisse</b>	<b>63</b>
7.1	Ergebnisse des Vergleichs	63
7.1.1	Kriterienkatalog	63
7.1.2	Ergebnisse der Evaluation	69
7.2	Ergebnisse der Webdesignumsetzung	81
7.2.1	Websiteanalyse	81
7.2.2	Kriterienkatalog	83
7.2.3	Ergebnisse der Evaluation	87
7.2.4	Qualitätsmessung	91
<b>8</b>	<b>Fazit</b>	<b>96</b>
8.1	Beantwortung der Forschungsfragen	97
8.2	Methodenkritik, Limitierung und Ausblick	99
	<b>Literaturverzeichnis</b>	<b>100</b>
	<b>Abbildungsverzeichnis</b>	<b>104</b>
	<b>Tabellenverzeichnis</b>	<b>106</b>
	<b>Anhang</b>	<b>107</b>

# 1 Einleitung

Der Fachkräftemangel ist in österreichischen Betrieben quer durch die Branchen sehr stark vertreten. 42,2% der befragten Unternehmen einer im Mai und Juni 2021 durchgeführten Umfrage, erhoben durch das Institut für Bildungsforschung der Wirtschaft, behaupten den Fachkräftemangel in ihrem Betrieb sehr stark zu spüren. Die Branche der EDV- und IT-Dienstleister\*innen ist mit 39,2% ebenfalls sehr stark vom Fachkräftemangel betroffen und befindet sich im Mittelfeld aller befragten Branchen (Wirtschaftskammer Österreich, 2021).

Ein Mittel, um dem Mangel an Fachkräften entgegenzuwirken, könnte der Einsatz von No-Code-Plattformen anstelle der herkömmlichen Betriebsmittel sein, sowie ein generelles Verfolgen des No-Code- / Low-Code-Ansatzes.

Unter Low-Code- / No-Code-Entwicklungsplattformen versteht man visuelle Softwareentwicklungsumgebungen, mit der Entwickler\*innen über eine grafische Benutzeroberfläche Anwendungskomponenten per Drag-and-Drop verschieben, diese verknüpfen und somit mobile- oder Webapplikationen erstellen können (Pratt, 2021).

Diese Low-Code- und No-Code-Ansätze ermöglichen es professionellen Entwickler\*innen Anwendungen in kurzer Zeit zu erstellen, da es nicht mehr notwendig ist, Zeile für Zeile Code zu schreiben. Dieser Aspekt erlaubt zudem Angestellten aus anderen Abteilungen als der IT, Inhaber\*innen kleinerer Unternehmen und allen anderen Personen, die keine Softwareentwickler\*innen sind, Anwendungen zu erstellen und zu testen. Diese Personen können Anwendungen entwickeln, ohne hohe Kenntnisse über herkömmliche Programmiersprachen, Maschinencode oder Wissen über die eigentliche Entwicklungsarbeit, die hinter der eingesetzten Plattform steckt, zu besitzen (Pratt, 2021).

Innerhalb der letzten Jahre ist der Low-Code- / No-Code-Ansatz regelrecht zum Trend geworden und es sind immer mehr Low-Code- / No-Code-Plattformen am Markt erschienen. Das Wirtschaftsmagazin Forbes spricht sogar vom „Most

## 1 Einleitung

---

disruptive Trend of 2021“ (Atkins, 2020). Als „disruptiv“ oder „zerstörerisch“ werden Technologien oder Innovationen bezeichnet, welche etablierte Geschäftsmodelle oder Technologien verdrängen oder zur Gänze ersetzen (Bendel, 2021). Abbildung 1 zeigt das steigende Interesse an No-Code und Low-Code anhand der via Google getätigten Suchen der letzten Jahre.

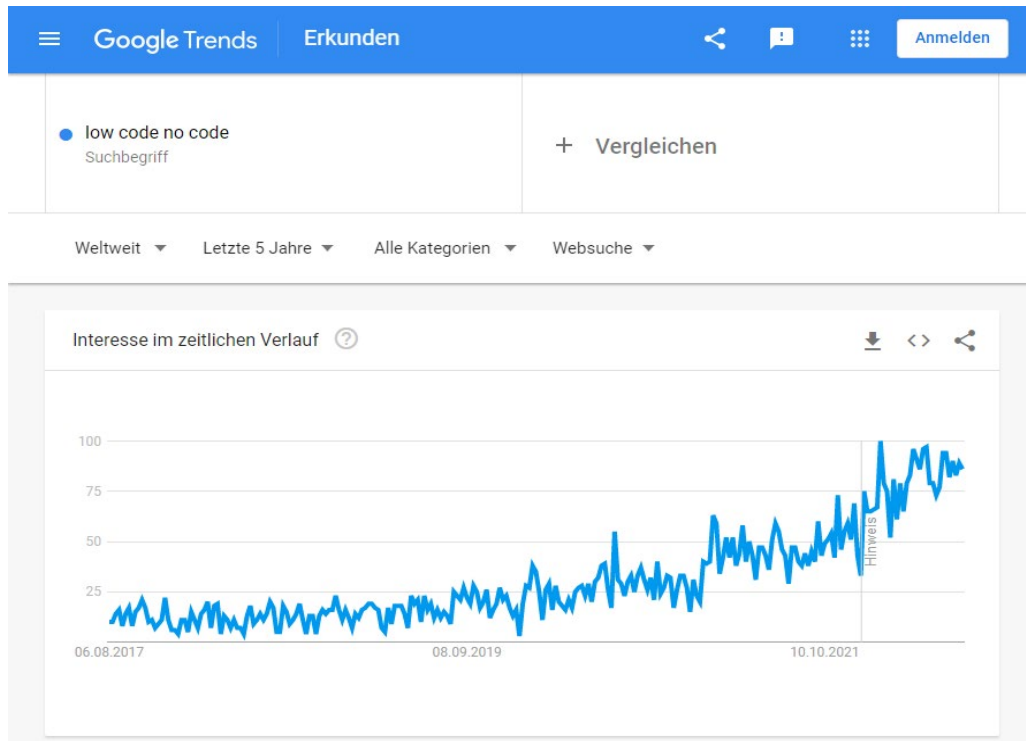


Abbildung 1. Trend der Suchbegriffe Low-Code No-Code (eigener Screenshot)

Analytiker\*innen des IT-Marktforschungsunternehmens Gartner schätzen, dass der Low-Code-Markt bis 2025 auf fast 30 Milliarden USD anwachsen wird. Gartner prognostiziert außerdem dass die Entwicklung von Low-Code-Anwendungen bis 2024 65% aller Softwareentwicklungen ausmachen wird, vor allem bei Projekten kleinerem und mittlerem Ausmaßes (Pratt, 2021).

Zusätzlich werden laut Gartner bis 2024 80% der Technologieprodukte und -dienstleistungen von Personen entwickelt, die keine Expert\*innen sind. „Digitales Business würde von den CEOs als Team sport behandelt werden und sei nicht mehr die alleinige Domäne der IT-Abteilung“, sagt Rajesh Kandaswamy, Distinguished Research Vice President bei Gartner. „Das Wachstum digitaler Daten, von Low-Code-Entwicklungstools und die durch Künstliche Intelligenz (kurz KI) unterstützte Entwicklung zählen zu den vielen Faktoren, die eine

Demokratisierung der Softwareentwicklung über IT-Fachleute hinaus ermöglichen“ (Rimol, 2021).

Dem Low-Code- / No-Code-Markt steht demnach voraussichtlich ein großes Wachstum bevor. Schlussendlich zählt jedoch aber die Qualität des Endprodukts, also der Website oder der Applikation, welche mittels Low-Code- / No-Code-Plattform erstellt wurde. Ob diese Plattformen heutzutage in der Lage sind ein vorgegebenes, modernes Webdesign vollständig, ohne den Einsatz von eigenem Code, umsetzen zu können, wird in dieser Masterarbeit erforscht.

### **1.1 Ziele und Forschungsfragen**

Im Rahmen dieser Masterarbeit werden die folgenden wissenschaftlichen Fragestellungen untersucht:

Inwiefern eignen sich No-Code-Entwicklungsplattformen zur Umsetzung eines modernen, responsiven Webdesigns?

- Welche Einschränkungen treten während der Verwendung einer No-Code-Entwicklungsplattform bei der Umsetzung eines modernen, responsiven Webdesigns auf?

Aus den Forschungsfragen wurde die folgende Hypothese abgeleitet:

Wird eine No-Code-Plattform für die Umsetzung eines modernen, responsiven Webdesigns eingesetzt, so werden Einschränkungen auftreten, welche ausschließlich mit eigens geschriebenem Code behoben werden können.

Ziel dieser Arbeit ist es, mehrere No-Code-Entwicklungsplattformen einem umfassenden Vergleich zu unterziehen. Mit ausgewählten Plattformen soll jeweils eine idente Website auf Basis eines Screendesigns erstellt werden. Die Ergebnisse sowie die Erkenntnisse der Umsetzung werden im Anschluss analysiert.

### **1.2 Methodik**

Im ersten Teil der Arbeit wird anhand einer Literaturrecherche die Theorie des behandelten Themengebiets erarbeitet. Hierbei werden die grundlegende Begriffe Webdesign & Webentwicklung, Web-Content-Management und Low-Code- / No-

Code-Entwicklung erklärt. Für die Beschaffung der Theorie der Kapitel zwei und drei wurde auf Standardwerke und Fachliteratur in Form von Büchern zurückgegriffen, welche in den Bibliotheken der Technischen Universität Wien und der Fachhochschule St. Pölten aufliegen. Ein Großteil der Informationen konnte den Fachbüchern der Verlage Rheinwerk und Springer entnommen werden. Aufgrund der Aktualität der Themengebiete in den Kapiteln vier und fünf liegen kaum gedruckte Fachpublikationen vor, aus diesem Grund wurden die Suchmaschinen Google Suche und Google Scholar genutzt. Ein Suchvorgang mit dem Keyword „low-code development“ lieferte beispielsweise 1 990 000 000 Ergebnisse. Eine weitere Suche mit dem Keyword „no-code plattform“ erzielte 5 730 000 Treffer. Um aus dieser großen Anzahl an Treffern relevante Informationen zu extrahieren, wurden weiters die Online-Bibliotheken ACM Digital Library und IEEE Xplore Digital Library nach wissenschaftlichen Publikationen sowie fachspezifische Online-Blogs nach themenrelevanten Artikeln durchsucht.

Zur Beantwortung der Forschungsfragen wird in weiterer Folge ein Vergleich von vier ausgewählten No-Code-Plattformen hinsichtlich des Funktionsumfangs durchgeführt. Zum Vergleichen der Plattformen wird ein Kategorienkatalog erstellt. Basierend auf diesen Kategorien werden die Plattformen evaluiert und verglichen. Da es zum Zeitpunkt der Ausarbeitung dieser Arbeit keine offizielle, unabhängige Liste über die Anzahl der Nutzer\*innen oder ähnlichen vergleichbaren Metriken gibt, werden jene zu vergleichenden Plattformen durch den Forschenden ausgewählt, welche einen ähnlichen Funktionsumfang aufweisen, sowie basierend auf einer Onlinerecherche als direkte Konkurrenten genannt werden.

Im zweiten Teil der Arbeit wird ein Experiment durchgeführt, welches zeigen soll, inwiefern eine Webdesignumsetzung mittels No-Code-Plattform möglich ist, ohne eigenen Code zu schreiben. Dafür wird eine Website, welche typische Features von modernem, responsivem Webdesign aufweist, ausgewählt und analysiert, sowie designspezifische Kriterien für die Umsetzung in Form eines Anforderungskatalogs abgeleitet. Anschließend wird versucht, das Design der Website mit den zwei No-Code-Plattformen, welche im vorangegangenen Vergleich die meisten Funktionen aufweisen konnten, detailgetreu nachzubilden und dabei die aufgestellten Kriterien vollständig zu erfüllen.

## 1.3 Aufbau der Arbeit

Im zweiten Kapitel der Arbeit wird der Begriff Webdesign sowie dessen Grundlagen erklärt. Im Unterkapitel 2.1 werden die grundlegenden Elemente eines User Interfaces aufgezählt. Nachfolgend werden hier die Merkmale und Einsatzzwecke des Layouts und der Struktur (2.1.1), der Gestaltung mit Typografie (2.1.2), Farbe (2.1.3), Bildern, Grafiken und Videos (2.1.4) sowie von Animationen und Interaktionen (2.1.5) behandelt.

Das Unterkapitel 2.2 widmet sich der Webentwicklung. Dabei werden die aktuellen Webtechnologien und -standards HTML (2.2.1), CSS (2.2.2), JavaScript (2.2.3) und PHP (2.2.4) vorgestellt. Auch Best Practices (2.3) der Bereiche Barrierefreiheit (2.3.1), Suchmaschinenoptimierung (2.3.2) und Performance-Optimierung (2.3.3) werden in diesem Kapitel aufgezeigt.

Das dritte Kapitel befasst sich mit dem Themengebiet Web-Content-Management. Zuerst werden im Unterkapitel 3.1 die Begriffe Content (3.1.1), Content-Management (3.1.2) und Content-Management-System (3.1.3) definiert, sowie eine Auflistung der unterschiedlichen Arten an Systemen (3.1.4) geschaffen. In weiterer Folge (3.2) werden die Funktionsweise (3.2.1) als auch die Vor- und Nachteile (3.2.2) von Content-Management-Systemen im Web erläutert. Am Beispiel des Content-Management-Systems WordPress (3.3) wird gezeigt, wie die Websitegestaltung abläuft (3.3.1), anschließend werden der Block-Editor und das Full-Site-Editing vorgestellt (3.3.2) und zuletzt wird sich mit Page-Builder-Plugins (3.4) für WordPress befasst.

Im vierten Kapitel erfolgt eine Erklärung des Low-Code- / No-Code-Programmierparadigmas. Hierbei wird das Thema zuerst historisch eingeordnet und dessen Entstehung (4.1) erklärt. Danach wird der Unterschied zwischen den Begriffen Low-Code und No-Code (4.2) erklärt. Im letzten Unterkapitel wird die Anwendungszielgruppe (4.3) definiert.

Kapitel fünf beschäftigt sich mit No-Code-Entwicklungsplattformen. Zu Beginn folgt ein Überblick über die am Markt erhältlichen, verschiedenen Arten von Low-Code- / No-Code-Plattformen (5.1). Danach wird der Aufbau und die Charakteristik dieser Systeme vorgestellt (5.2), sowie auf die Art der Bereitstellung (5.2.1) eingegangen. Im Unterkapitel (5.3) werden anschließend der Nutzen und die Limitierungen dieser Plattformen gegenübergestellt.

Die empirische Untersuchung erfolgt im sechsten Kapitel. Das erste Unterkapitel (6.1) erklärt die Vorgehensweise bei der Recherche. Die Forschungsmethoden

## 1 Einleitung

---

und der Ablauf der Evaluierung werden im nächsten Unterkapitel (6.2) erläutert. Im Unterkapitel 6.3 wird zuletzt das Forschungsdesign vorgestellt, sowie die Bewertungskriterien dargelegt.

Kapitel sieben beinhaltet schlussendlich die Ergebnisse der empirischen Untersuchung. In Unterkapitel 7.1 werden zuerst die Ergebnisse des Vergleichs präsentiert. Zu Beginn wird der Kriterienkatalog (7.1.1) gezeigt und anschließend folgen die Ergebnisse der Evaluation (7.1.2). Unterkapitel 7.2 legt die Ergebnisse der Webdesignumsetzung dar. Zuerst wird die Websiteanalyse (7.2.1) präsentiert, gefolgt vom Kriterienkatalog (7.2.2), den Ergebnissen der Evaluation (7.2.3) sowie der Qualitätsmessung (7.2.4).

Das letzte Kapitel der Arbeit, das Fazit, beantwortet die Forschungsfragen (8.1), betrachtet die Limitierung der durchgeführten Forschungsmethode und gibt einen Ausblick (8.2) auf zukünftige, weiterführende Forschungsmöglichkeiten und offene Fragen innerhalb des Themengebiets.

## 2 Grundlagen des Webdesigns und der Webentwicklung

Unter Webdesign versteht man sprichwörtlich: Designen für das Web. Eine genaue Definition der beiden Begriffe „Web“ und „Design“ gibt es jedoch nicht. Das „Web“ steht für World Wide Web und dabei handelt es sich um einen Dienst im Internet, mittels welchem sich (multi-)mediale Inhalte wie Texte, Bilder, Videos, etc. auf Basis der Seitenbeschreibungssprache HTML5 veröffentlichen lassen. Im Gegensatz zum Printbereich lassen sich Webseiten über Hyperlinks miteinander verknüpfen. Auch der Begriff „Design“ lässt sich nicht eindeutig definieren. Im Kontext des Webs bezieht er sich auf die Konzeption und Gestaltung von Websites und Webapplikationen durch Webdesigner\*innen. Heutzutage handelt es sich dabei nicht mehr ausschließlich um Websites, sondern auch um Applikationen für mobile Endgeräte (Böhlinger, Bühler, Schlaich, & Sinner, 2014, S.6).

Webdesign beschreibt grundsätzlich den gesamten Prozess der Website-Erstellung von der Konzeption bis hin zur technischen Umsetzung. Die eng verwandten Begriffe Screendesign und User Interface Design bezeichnen hingegen ausschließlich das visuelle Gestalten für unterschiedliche Endgeräte (Hahn, 2020).

### 2.1 Elemente des User Interfaces

In diesem Abschnitt werden unterschiedliche Elemente des User Interfaces vorgestellt und ihre Einsatzzwecke im Webdesign erläutert.

#### 2.1.1 Layout und Struktur

Das Design spielt in der Webgestaltung nicht nur eine rein ästhetische Rolle, es dient viel mehr als Mittel zur Lösung eines Problems. Der Leitsatz „Form follows function“ ist hier wichtiger denn je, denn das Design soll dafür sorgen, dass eine Website einfach bedienbar ist und die Nutzer\*innen gezielt durch die Website führen, indem es wichtigen Elementen eine optische Bedeutung gibt. Websites sollten übersichtlich gestaltet sein, damit sich Nutzer\*innen schnell zurechtfinden und Informationen ohne langes Suchen finden können. Dafür sind ein

## 2 Grundlagen des Webdesigns und der Webentwicklung

---

übersichtlicher Seitenaufbau sowie eine einfach zu bedienende Navigation essenziell (Hahn, 2020, S.28f.).

Über die letzten Jahre haben sich etliche Konventionen etabliert, welche von der Mehrzahl aller Websites befolgt werden. Die Nutzer\*innen einer Website erwarten sich beispielsweise nach einem Klick auf das Logo wieder auf der Startseite zu landen. Designer\*innen müssen selbst entscheiden, welche Konventionen sie einhalten. Grundsätzlich gilt jedoch: Konventionen nur dann zu brechen, wenn die eigene Idee besser ist und dies sollte unbedingt getestet werden (Rohles, 2017, S.41).

Weitere Gemeinsamkeiten, welche auf fast jeder Website wiedererkennbar sind, betreffen die Positionierung der üblichen Webseitenbereiche. Die in Abbildung 2 dargestellten Elemente können als das Grundgerüst einer typischen Website verstanden werden (Hahn, 2020, S.271f.).

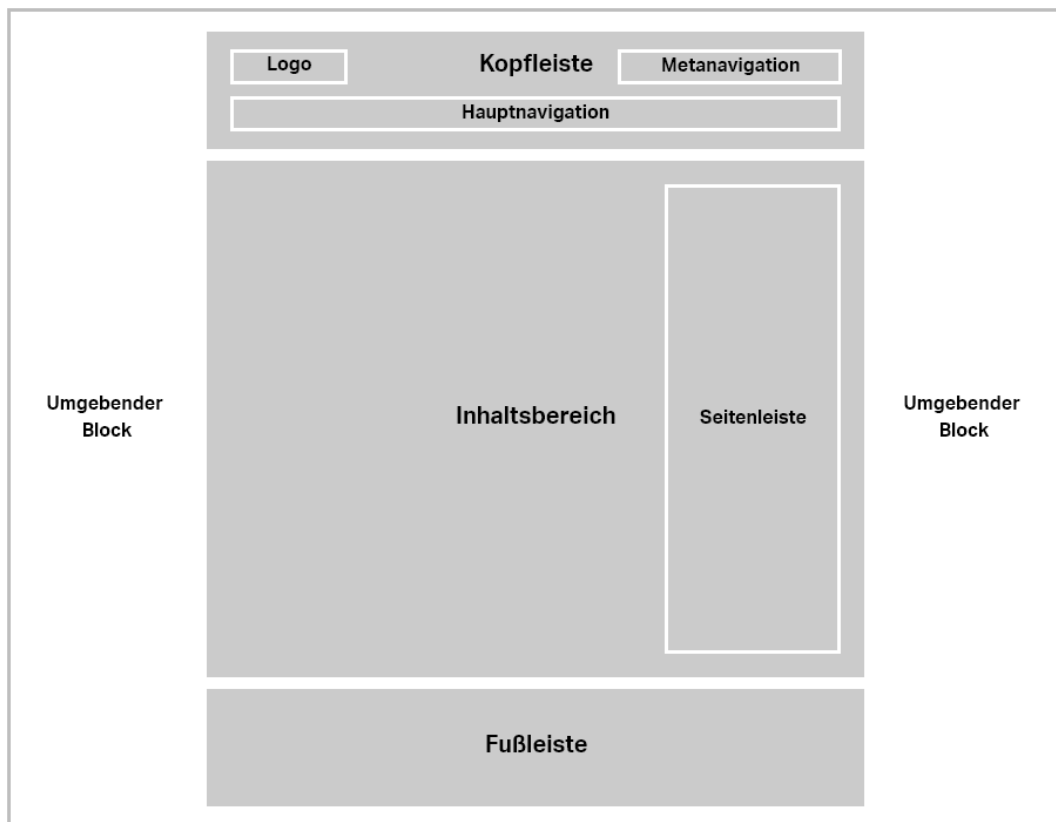


Abbildung 2. Anatomie einer Website (in Anlehnung an Hahn, 2020, S.271)

### **Kopfleiste**

Der oberste Bereich einer Website wird Header oder Kopfleiste genannt und enthält Logo, Haupt- und Metanavigation. Das wichtigste Design-Element einer Website ist das Logo, denn es stellt die Assoziation zum Unternehmen, den Produkten oder Dienstleistungen her und dient zugleich als Differenzierungs- und Wiedererkennungsmerkmal. Die Hauptnavigation ist dafür zuständig, den Besucher\*innen einen strukturierten Überblick über die Inhalte zu verschaffen, deswegen sollte sie immer prominent im oberen Bereich einer Website platziert sein. In der Metanavigation werden sogenannte Serviceinhalte, wie etwa eine Verlinkung auf das Impressum, platziert, welche ebenfalls wichtig sind, jedoch nicht in der Hauptnavigation vorkommen (Hahn, 2020, S.273).

### **Inhaltsbereich**

Der Inhaltsbereich, welcher unter der Kopfleiste platziert wird, enthält an oberster Stelle die wichtigsten Inhalte, oftmals begleitet durch eine markante Überschrift und einem großen Teaser-Bild. Auf modernen Websites streckt sich der Inhaltsbereich über die gesamte Breite der Website, eine Seitenleiste oder Subnavigation wird immer seltener. Den eigentlichen Inhalten wird oft durch mehrspaltige Bereiche, großflächige Bilder oder farbige Hintergrundbalken Struktur verschafft (Hahn, 2020, S.274f.).

### **Fußleiste**

Die Fußleiste, auch Footer genannt, bildet den inhaltlichen und gestalterischen Abschluss einer Website. Ergänzende Inhalte oder weitere Navigationsmöglichkeiten werden hier platziert und visuell wird hier das Design der Website abgerundet. Die Fußleiste bleibt in der Regel wie die Kopfleiste über alle Unterseiten hinweg ident (Hahn, 2020, S.276).

Neben der zuvor beschriebenen Anatomie einer Website unterscheidet sich der Aufbau von Websites auch durch verschiedene Layouttypen. Laut Rohles (2017, S.170f.) unterscheidet man fixe Layouts, fluide Layouts, adaptive und anpassungsfähige (responsive) Layouts.

### **Fixes Layout**

Beim fixen Layout wird die Breite der Spalten und des Gesamtlayouts in Pixel angegeben. Das ermöglicht zwar eine einfache Gestaltung des Designs in Layout- und Gestaltungsprogrammen, da hier pixelgenau gearbeitet werden kann, es kommt dadurch aber auch zu Einschränkungen für die Nutzer\*innen. Bei kleineren

Bildschirmgrößen als der eingestellten Breite des Layouts entstehen horizontale Scrollbalken, welche als störend empfunden werden können. Zudem sind wichtige Inhalte bei Verkleinerung möglicherweise nicht sofort sichtbar (Böhringer et al., 2014, S.52; Rohles, 2017, S.170f.).

### **Fluides Layout**

Fließende oder fluide Layout werden hingegen erreicht, indem für die Breitenangaben relative Einheiten (%) verwendet werden. Diese beziehen sich auf die Breite des Viewports (= sichtbarer Bereich der Website am Bildschirm) und passen sich demnach automatisch an Veränderungen an. Die Vorteile, wie die automatische Anpassung an die Breite des Viewports, das Wegfallen von horizontalem Scrollen und der ständigen Sichtbarkeit der Inhalte, sprechen für sich. Ein Designentwurf mit Grafikprogrammen wie Photoshop oder Illustrator, wie etwa beim fixen Layout, ist aufgrund der relativen Einheiten jedoch nicht möglich. Auch die Darstellung und Lesbarkeit könnte insbesondere bei großer Breite leiden (Böhringer et al., 2014, S.52).

Eine abgewandelte Form des fluiden Layouts, welche auch als „elastisches Layout“ bezeichnet wird, verspricht maximale Flexibilität. Hierbei werden die Breitenangaben relativ (in em) auf die Grundschrift des Browsers, normalerweise 16 Pixel, bezogen. Wird diese von den Nutzer\*innen verändert, so ändert sich auch das Gesamtbild der Website. Diese Möglichkeit der Skalierung der Schrift wird aber ebenfalls durch die Zoomfunktion des Browsers ermöglicht (Böhringer et al., 2014, S.52f.).

### **Adaptives Layout**

Bei adaptiven Layouts wird eine Website für bestimmte Geräte optimiert, deren jeweilige Größe den Webdesigner\*innen bekannt ist. Die Breitenangaben werden häufig in Pixel angegeben, welche mittels verschiedener Breakpoints (= Umbruchpunkte des Layouts) definiert werden. Grundsätzlich handelt es sich bei adaptiven Layouts also um eine Sammlung unterschiedlicher fixer Layouts. Da die verschiedenen Abmessungen bekannt sind, können adaptive Layouts gut in Layoutprogrammen simuliert werden, zudem sind die Layouts und deren Inhalte und Gestaltungselemente technisch relativ unkompliziert und zeitsparend zu entwickeln, da innerhalb der Breakpoints klar definierte Raster genutzt werden können. Der größte Nachteil besteht darin, dass keine Optimierung für Viewports (= sichtbarer Bereich des Bildschirms) außerhalb der definierten Breakpoints stattfindet, zusätzlich steigt der Aufwand mit den zu berücksichtigenden Viewports

und auch bei zukünftigen, technologischen Veränderungen könnten Nachbesserungen beim Layout notwendig sein (Rohles, 2017, S.172f.).

### **Responsives Layout**

Die optimale Lösung ist das responsive Layout, welches auf fluiden Layouts basiert. Responsiv bedeutet in etwa „reaktionsfähig“, denn das Layout reagiert in diesem Falle auf die „äußeren“ Gegebenheiten. Auch beim responsiven Layout kommen Breakpoints zum Einsatz, aber durch den Einsatz von relativen Breitenangaben ist das Layout auch zwischen den Breakpoints im Fluss. Spalten können breiter werden, Bilder skalieren und Abstand größer werden. Responsive Layouts sind absolut flexibel hinsichtlich der Ausgangsbreite, dadurch sind keinerlei Einschränkungen durch Geräte oder Bildschirmauflösungen zu erwarten. Dieser Ansatz lässt sich demnach als zukunftsfähig bezeichnen. In der Umsetzung ist der responsive Ansatz jedoch deutlich komplexer als andere Layouts. Jede mögliche Bildschirmbreite samt prozentueller Anpassung lässt sich nicht in Grafikprogrammen veranschaulichen, häufig ist also die Verwendung von Prototypen und ein Umdenken bei allen Projektbeteiligten notwendig, denn einschätzen lassen sich responsive Designs nur noch im Browser (Hahn, 2020, S.307f.; Rohles, S.173f.)

### **2.1.2 Typografie**

Lesen am Bildschirm ist grundsätzlich anstrengender als auf Papier, dies hat hauptsächlich technisch bedingte Gründe. Einerseits ist die Auflösung von Displays im Vergleich zur Druckauflösung geringer, denn gedruckte Schrift wird stufenlos wahrgenommen, einzelne Monitorpixel lassen sich hingegen erkennen. Auf der anderen Seite sorgt die Kontrast- und Farbwiedergabe von Bildschirmen sowie das Leuchten des Displays generell für Anstrengung der Augen (Bühler et al., 2017, S.60). Claudia Korthaus (2020, S.301) ergänzt diese Liste um die schlechten Lesebedingungen auf Tablets und Smartphones. Diese werden oftmals unter schlechten Lichtverhältnissen oder unterwegs genutzt, um im Web zu surfen. Spiegelnde Displays und die unterschiedliche Menge an Inhalten, bedingt durch die verschiedenen Bildschirmgrößen von Smartphones und Tablets, können sich weiters negativ auf die Konzentrationsfähigkeit der Nutzer\*innen auswirken. Webdesigner\*innen sollten den Anspruch haben, den Nutzer\*innen, ohne deren Eingreifen optimal lesbare Webseiten zu präsentieren. Korthaus zufolge gibt es drei Aspekte zu beachten, damit Text gut lesbar ist: die Schriftgröße und -Art, die Laufweite und der Kontrast.

Über viele Jahre hinweg wurden ausschließlich Systemschriften, auch „websichere Schriften“ genannt, im Web verwendet. Das sind Schriftarten, welche bereits auf den meisten Betriebssystemen vorinstalliert sind. Bekannte Vertreter sind beispielsweise Arial, Courier, Verdana, Times New Roman oder Georgia. Ihr Vorteil besteht aus der einfachen Formsprache und einer optimierten Darstellung am Bildschirm, was insbesondere bei kleinen Schriftgrößen hilfreich ist. Der große Nachteil ist jedoch die sehr beschränkte Auswahl an verschiedenen Schriftarten. Seit der Etablierung von Webfonts, lassen sich auch externe Schriftarten mittels CSS-Befehl in Websites einbinden. Webfonts können entweder selbstgehostet oder von einem Webfont-Dienst kostenlos heruntergeladen oder lizenziert werden (Hahn, 2020, S.434f.).

Auch die Wahl des Schrifttyps ist für den Einsatz im Web essenziell. Serifenschriften haben gegenüber grotesken Schriften im Web den Nachteil, dass sich durch die geringere Auflösung die feinen Serifen oft nicht sauber darstellen lassen, was zur Folge hat, dass die Schrift eher unscharf wirkt. Grundsätzlich gilt: Je einfacher und klarer eine Schrift, desto besser ist sie lesbar. Schriften mit feinen Serifen, Schreibschriften oder handschriftliche Schriften sind demnach in kleineren Größen ungeeignet. Als gute durchschnittliche Standardschriftgröße empfiehlt sich 14px bis 16px. Kleinere Schriftgrößen sollten nur in Ausnahmefällen verwendet werden (Korthaus, 2020, S.284f.).

Die Zeilenlänge trägt ebenfalls zur Lesbarkeit von Schrift im Web bei. Zu lange Zeilen lassen sich schwer verfolgen und können fehlerhafte Zeilenwechsel verursachen. Bei zu kurzen Zeilen hingegen können durch den ständigen Zeilenwechsel Informationen schlechter aufgenommen werden. Laut Korthaus (2020, S.293) sind 40 Zeichen bei mehrspaltigen Layouts ausreichend. Rohles hingegen hält 50 bis 80 Zeichen pro Zeile für optimal. Eine gute Faustformel ist es, von etwa zehn Wörtern pro Zeile auszugehen (Rohles, 2017, S.249).

Auch der Zeilenabstand kann die Lesbarkeit eines Textes deutlich beeinflussen. Im Web sind größere Zeilenabstände als im Printbereich sinnvoll. Als Faustregel kann die Höhe der Versalien für den Abstand zwischen den Zeilen verwendet werden. Schriften mit größeren x-Höhen (= Abstand zwischen Grundlinie und Mittellinie beziehungsweise die Höhe eines Kleinbuchstabens ohne Ober- und Unterlänge) benötigen zudem einen größeren Zeilenabstand, als Schriften mit kleinen x-Höhen, da diese schon von alleine mehr optischen, freien Raum besitzen (Korthaus, 2020, S.293).

### 2.1.3 Farbe

Im Gegensatz zu den Printmedien kommt bei Monitoren die additive Farbmischung zum Einsatz. Dieses additive Farbsystem wird aufgrund der Grundfarben Rot, Grün und Blau als das RGB-System bezeichnet. Jede Farbe, die der Mensch sieht, wird aus diesen Grundfarben zusammengesetzt. Die Mischung aus Rot, Grün und Blau erzeugt Weiß (Bühler et al., 2017, S.65).

Da sich der RGB-Farbraum vom CMYK-Farbraum des Drucks unterscheidet, gibt es Farben, welche zwar druckbar sind, aber nicht auf Monitoren dargestellt werden können, umgekehrt gibt es ebenfalls Farben welche ausschließlich auf Bildschirmen angezeigt werden können. Aus diesem Grund ist es sinnvoll für Designs, bei denen eine durchgängige Farbgestaltung aller Medien unerlässlich ist, Farben aus der Schnittmenge des RGB- und CMYK-Farbraums zu wählen. Sollte dies nicht der Fall sein, so müssen Ersatzfarben gewählt werden, welche den Originalfarben so nahe wie möglich kommen (Böhringer et al., 2014, S.60). Abbildung 3 stellt die Schnittmenge zwischen den RGB- und CMYK-Farbräumen zur besseren Verständlichkeit schematisch dar.

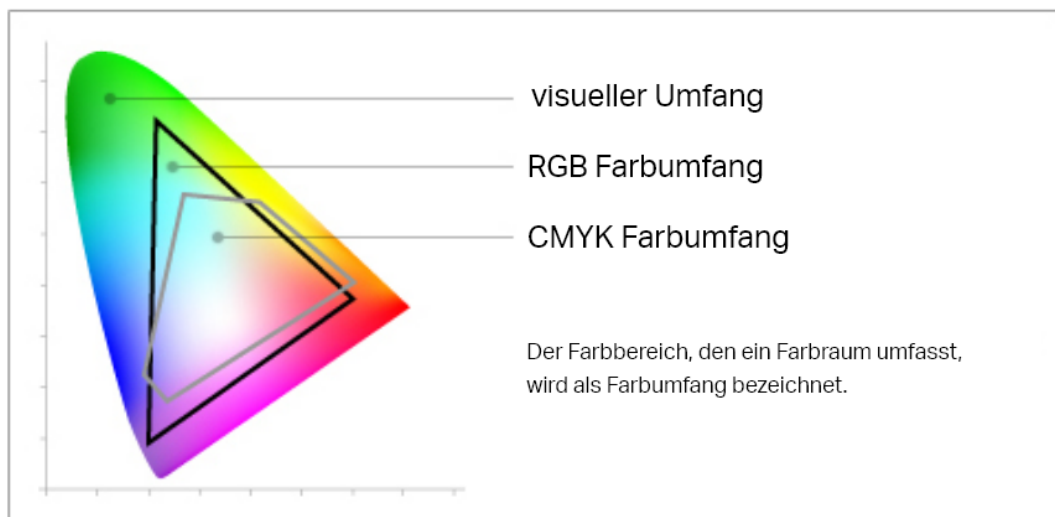


Abbildung 3. RGB und CMYK Farbraum (eigene Darstellung)

Die richtige Farbwahl spielt im Webdesign eine besondere Rolle, da zahlreiche Elemente einer Website farbig gestaltet sind. Somit leisten Farben einen entscheidenden Beitrag für das Gesamtbild der Website. Die gewählten Farben tragen wesentlich zur vermittelten Atmosphäre einer Website bei, doch neben den rein optischen Aspekten, sollten auch die Firmenidentität, die Erwartungen der

Zielgruppe sowie die Benutzerfreundlichkeit nicht außer Acht gelassen werden (Hahn, 2020, S. 338f.).

Laut Bühler, Schlaich und Sinner (2017, S.69) hat Farbe im Webdesign neben der Ästhetik die folgenden Funktionen: sie schafft Assoziationen und Emotionen, leitet und führt durch die Website, gliedert, hebt hervor und setzt Akzente oder wird selbst zur Marke. Demnach ist Farbe das wichtigste Gestaltungsmittel.

Rohles (2017, S.372f.) weist auch auf die wichtigen Aspekte der Barrierefreiheit und Usability (= Gebrauchstauglichkeit) bei der Farbwahl hin. Er nennt folgende Regeln, welche unbedingt beachtet werden sollten:

- Zwischen Textfarbe und Hintergrund sollte immer ein ausreichender Kontrast bestehen. Allgemein ist ein Kontrastverhältnis von 7:1 zu empfehlen, bei ausreichend großer Schrift ist ein Verhältnis von 4,5:1 ausreichend. Außerdem ist bei der Farbwahl von dünnen Schriften zu beachten, dass diese von größerem Kontrast profitieren.
- Farbfehlsichtigkeit, wie zum Beispiel die Rot-Grün-Blindheit, ist ein weiteres Kriterium, welches einige Menschen bei der Nutzung einer Website einschränken kann. Farbe sollte deshalb niemals als das einzige Mittel zur Informationsübermittlung verwendet werden. Aktive Buttons können beispielsweise zusätzlich fett ausgezeichnet oder unterstrichen werden.
- Auch flackernde Lichtquellen oder Farbflächen sollten unbedingt vermieden werden, damit Menschen, welche an fotosensitiver Epilepsie leiden, nicht dadurch beeinträchtigt werden.

### 2.1.4 Multimedia (Bilder, Grafiken, Videos)

Bilder und Grafiken sind laut Bühler, Schlaich und Sinner (2017, S.74) die wichtigsten Komponenten einer Website, da Menschen Informationen am schnellsten und besten über den Sehsinn aufnehmen.

Als bedeutenden Vorteil von Bildern, wenn es um das Erinnern von Informationen geht, nennt Rohles (2017, S.378) den Picture-Superiority-Effect. Beim Vergleich von Informationen in Bild- und Textform bleiben Bilder meist besser in Erinnerung. Wenn Bilder und Worte kombiniert werden, sei dieser Effekt besonders stark.

Hahn (2020, S.500) schreibt Bildern folgende weitere Vorteile und Aufgaben zu:

- hohe Kommunikationsgeschwindigkeit

## 2 Grundlagen des Webdesigns und der Webentwicklung

---

- werden in der Regel zuerst fixiert
- hohe Glaubwürdigkeit
- hohe Anschaulichkeit
- platzsparende Informationen
- allgemeine Verständlichkeit
- räumliche Vorstellung gut vermittelbar
- Erwecken von Interesse

Wenn Bilder als Layoutelemente eingesetzt werden, tragen sie vorwiegend zur Stimmung einer Website bei. Klassische Beispiele sind etwa Hintergrundbilder oder -muster. Da Bilder die Aufmerksamkeit der Websitenutzer\*innen auf sich ziehen, lassen sie sich auch sehr gut als Orientierungselemente nutzen. Bilder, die als Informationsanker dienen oder die Nutzer\*innen beim Scannen der Website unterstützen, können die Übersichtlichkeit einer Website deutlich erhöhen. Ebenso sind Bilder als emotionale Elemente ideal dafür geeignet, einen hohen Wiedererkennungswert zu schaffen, da sie Stimmungen vermitteln, Gefühle hervorrufen und somit eine Website interessant, markant und unverwechselbar machen können (Hahn, 2020, S.501f.).

Bühler, Schlaich und Sinner (2017, S.81f.) beschreiben Icons entweder als minimalisierte grafische Darstellung eines realen Objektes (ikonisch) oder als frei gestaltete Grafik (symbolisch). Symbolische Grafiken haben den Nachteil, dass ihre Bedeutung erst gelernt werden muss. Der Einsatz von Icons ist sinnvoll, wenn sie verständlich, selbsterklärend und intuitiv verwendbar sind. Daraus erübrigen sich folgende Vorteile im Gegensatz zu Text:

- Grafische Elemente sind über Landessprachen und -grenzen hinaus verständlich.
- Sie sind schneller erfassbar als Text, wenn ihre Bedeutung bekannt ist.
- Grafische Elemente können auch von Personen ohne Lesekenntnisse erfasst werden.
- Grafische Elemente wirken ansprechend, wenn sie gut gestaltet sind.

Diesen Vorteilen stehen jedoch folgende Nachteile gegenüber:

- Wenn die Bedeutung der Grafik unklar ist, stehen die Benutzer\*innen vor einem Ratespiel.
- Die symbolische Bedeutung einer Grafik kann sich international unterscheiden.

Videos kommen auf Websites als Inhaltselement oder als Mittel zur emotionalen Gestaltung zum Einsatz. Ein klassisches Beispiel von Videos als Inhaltselement sind Erklärvideos für die Wissensvermittlung. Video-Hintergründe sind ebenso weit verbreitet und können Stimmung vermitteln. Solche Hintergrundvideos werden oft in Endlosschleife und ohne Ton abgespielt, damit sie nicht den Fokus vom Vordergrund ablenken. Für Videos im Hintergrund eignen sich einerseits ruhige Aufnahmen von alltäglichen Situationen oder Landschaftsaufnahmen, aber auch Imagefilme und Storytelling können dafür eingesetzt werden (Rohles, 2017, S.436f.).

### **2.1.5 Animationen und Interaktionen**

Durch Animationen und Transitionen können während der Interaktion mit einer Website Zusammenhänge zwischen den Inhalten verdeutlicht, Ladezeiten überbrückt und veranschaulicht werden, dass das System auf die Eingaben der Nutzer\*innen reagieren wird. All dies kann die User Experience (= Nutzungserlebnis bei der Verwendung Interaktiver Systeme) der Besucher\*innen spürbar steigern. Ein sinnvoller Anwendungsfall von Animation ist beispielsweise das sanfte Animieren des Weges zu den Inhalten nach Klicken eines Ankerlinks. Somit wird den Nutzer\*innen verdeutlicht, dass sie sich noch immer auf derselben Website befinden. Visuelles Feedback sorgt dafür, den Zusammenhang zwischen den Handlungen der Nutzer\*innen und der Reaktion des Systems herzustellen.

Hahn (2020, S.630f.) listet einige interaktive, animierte Elemente auf, welche vermehrt auf Websites vorkommen, diese sind:

#### **Bildergalerien und Thumbnails**

Bilder werden oft in kleinen Vorschaubildern, auch bekannt als Thumbnails, auf Websites platziert. Per Klick auf ein Thumbnail öffnet sich das Bild dann in voller Größe in der sogenannten Lightbox. Ein leicht abgedunkelter Hintergrund sorgt dafür, dass der Fokus auf dem Bild und nicht auf den dahinterliegenden Elementen der Website liegt.

#### **Tabs oder Reiter**

Bei der Tab-Navigation erhält der aktive Menüpunkt die gleiche Hintergrundfarbe wie der Inhaltsbereich. Die restlichen Menüpunkte haben eine dunklere Hintergrundfarbe und erscheinen somit inaktiv. Dieser Effekt hebt den aktiven Tab deutlich hervor und stellt seine Verbindung zum Inhalt her. Tabs können jedoch

auch im Inhaltsbereich eingesetzt werden, wo inaktive Inhalte durch Animationen ein- und ausgeblendet werden.

### **Akkordeons**

Akkordeons ähneln den Tabs, da auch hier viele Informationen anfänglich verdeckt sind und erst durch einen Klick auf das Akkordeon angezeigt werden. Das Akkordeon kann durch einen Mausklick geöffnet und wieder geschlossen werden, wodurch es auseinandergezogen oder wieder zusammengeschoben wird. Üblicherweise sind Akkorden-Listen untereinander angeordnet auf Websites zu finden.

### **Content-Slider**

Slider oder Carousels haben ebenfalls den Zweck Inhalte erst nach und nach anzuzeigen. Dies wird oft durch einen Klick auf die Pfeilnavigation oder durch automatisches Rotieren des Sliders ermöglicht.

### **Header-Slideshows**

Header-Slideshows sind meist selbstrotierende Slider, welche im Kopfbereich der Website platziert werden.

### **Modalboxen**

Modals sind Dialogboxen, welche sich über die Inhalte der Website legen. Anders als bei der Lightbox von Bildergalerien wird mit Modalboxen auch interagiert. Sinnvoll eingesetzte Modalboxen können die Nutzung einer Website vereinfachen. Beispiele dafür sind Warnhinweise, Fehlermeldungen oder auch Anmeldungs- und Buchungsbestätigungen oder andere wichtige Informationen.

### **Tooltips**

Tooltips sind kleine Dialogboxen, welche direkt neben dem Element platziert sind, auf welches sie sich beziehen und weitere Informationen darüber liefern. Sie erscheinen in der Regel durch Hover auf das Element oder erst nach einem Klick.

### **Preloader & Ladebalken**

Preloader und Ladebalken sind sinnvoll, wenn den Nutzer\*innen vermittelt werden soll, wie lange eine Ladezeit andauern wird. Eine schnell ladende Website ist jedoch immer einem Preloader vorzuziehen.

## 2.2 Webtechnologien und -standards

Trennen von Inhalt und Design, sowie das Einhalten von Webstandards, sind die besten Voraussetzungen für guten Quellcode. Unter Webstandards versteht man technische Spezifikationen, welche von Organisationen wie dem W3C (= World Wide Web Consortium) entwickelt werden. Das Ziel dahinter sind browserübergreifende, verbindliche Spezifikationen, damit standardkonforme Websites über alle Browser hinweg fehlerfrei laufen. (Hahn, 2020, S.33f.).

Websites müssen heutzutage vielen unterschiedlichen Ansprüchen gerecht werden und auf einer Vielzahl von Geräten funktionieren. Um diese Aufgaben erfolgreich erfüllen zu können, muss jede Technologie für ihren erschaffenen Zweck eingesetzt werden. Inhalt muss von Design und Verhalten getrennt werden (Rohles, 2017, S.32). Im Folgenden werden die dafür gebräuchlichen Webtechnologien näher erklärt.

### 2.2.1 HTML

HTML steht für Hypertext Markup Language und bezeichnet eine textbasierte Auszeichnungssprache zur Strukturierung von Texten, Grafiken, Videos und Hyperlinks. Da HTML-Dokumente von allen Browsern dargestellt werden, gelten sie als Basis des Webs. Wie in anderen Programmiersprachen üblich, besitzt HTML jedoch keine Anweisungen oder Befehle, sondern Marken (im englischen Tags genannt), welche die einzelnen Bereiche eines HTML-Dokuments strukturieren. HTML wird somit nicht programmiert, sondern viel eher geschrieben. Ein HTML-Dokument kann in jedem beliebigen Texteditor bearbeitet und gespeichert werden. In einem HTML-Dokument werden neben den im Browser sichtbaren Daten auch Metainformationen (= Daten, welche nicht angezeigt, sondern Informationen über das Dokument enthalten), wie der Sprache oder den Autor des Dokuments, abgespeichert. HTML5 ist der derzeitige Standard der Auszeichnungssprache, welcher zusätzliche Elemente wie etwa Video, Audio und 2D-/3D-Grafiken mit sich gebracht hat (Wolf, 2019, S.39f.).

Die Bezeichnung „Hypertext“ steht für das Verlinken von Texten mithilfe von Links. Dadurch wird das Hin- und Herspringen innerhalb von Texten ermöglicht, was die Grundlage für das World Wide Web darstellt (Böhringer et al., 2014, S.86).

In Abbildung 4 wird das Grundgerüst eines HTML-Dokuments präsentiert. Der Doctype teilt dem Browser mit, an welchen Standard sich bei der Erstellung des Dokuments gehalten wurde und welche Befehle er ausführen soll. Im Head werden

die Metainformationen platziert und der Body enthält den eigentlichen Inhalt der Website, in Form von weiteren HTML-Tags.

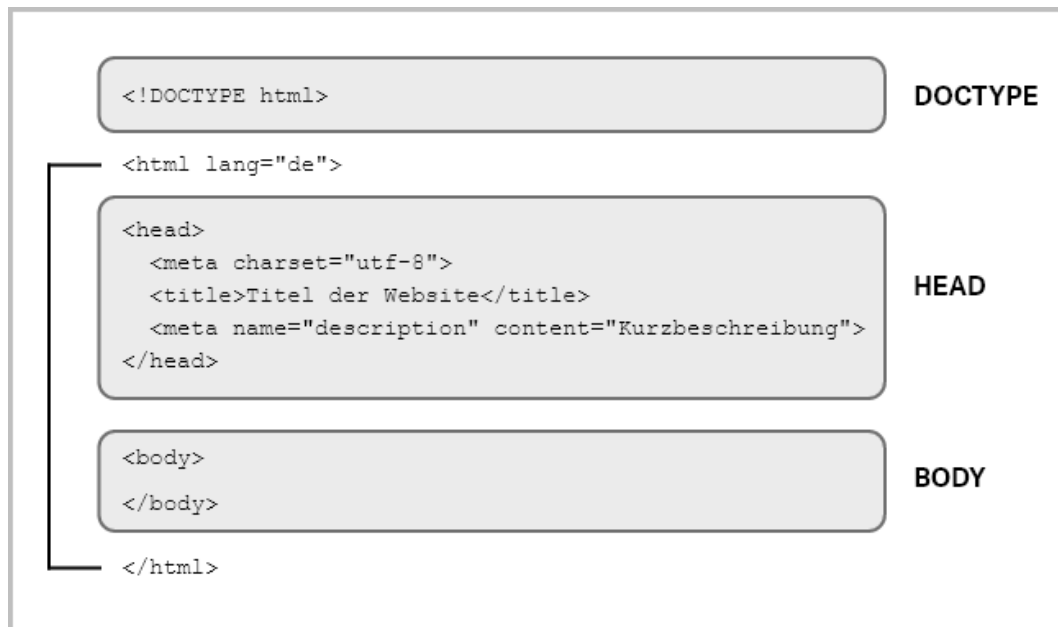


Abbildung 4. HTML-Grundgerüst (eigene Darstellung)

### 2.2.2 CSS

HTML war von Beginn an nicht für die Formatierung oder Gestaltung von Webseiten vorgesehen und verfügt deswegen nur über sehr wenige primitive Formatierungsoptionen. Aus diesem Grund wurde HTML 1996 durch die Formatierungssprache Cascading Style Sheets, kurz CSS, ergänzt, welche pixelgenaues Layouten und typografische Gestaltung von Webseiten ermöglicht (Böhringer et al., 2014, S.126).

Für Böhringer, Bühler, Schlaich und Sinner (2014, S.129) steht der Begriff „Cascading“ im übertragenen Sinne dafür, dass die Stylesheets intern oder extern, zentral oder lokal, also „mehrstufig“ angeordnet werden können. Damit keine Konflikte entstehen, sorgen vordefinierte Regeln für eine „kaskadierte“, nacheinander folgende Ausführung des CSS. Ein wichtiges Merkmal von CSS ist dessen Definition, diese kann folgendermaßen erfolgen:

- Extern in einer eigenen Datei
- Zentral im Dateikopf
- Lokal in einem HTML5-Element

Durch das Trennen von HTML und CSS in separate Dateien, erfolgt auch eine Trennung von Inhalt und Design. Das ermöglicht das Gestalten aller Websites mit nur einer einzigen CSS-Datei (Wolf, 2019, S.41). Böhringer, Bühler, Schlaich und Sinner (2014, S.127) nennen weitere Argumente, die für die konsequente Trennung von Inhalt und Design sprechen:

- Inhalt und Design können unabhängig voneinander entwickelt und bearbeitet werden.
- Inhalte können softwaregestützt ausgewertet werden, z.B. durch Screenreader.
- Inhalte lassen sich dynamisch per Content-Management-System verwalten.
- Für einen Inhalt können mehrere Designs erstellt werden, z.B. für Ausgabe auf PC-Monitoren, Smartphones oder für den Druck.

### 2.2.3 JavaScript

JavaScript ist eine clientseitige Skriptsprache, was bedeutet, dass die webbasierten Skripte lokal auf dem Rechner vom Webbrowser ausgeführt werden. JavaScript erweitert die Möglichkeiten von HTML um Reaktionen auf Interaktionen von Benutzer\*innen, wie das Auswerten, das dynamische Verändern oder das Erzeugen von Inhalten. Heutzutage wird JavaScript auf fast allen modernen Websites eingesetzt (Wolf, 2019, S.42).

Mittels JavaScript lassen sich Funktionalitäten entwickeln, die die Nutzung von Websites vereinfachen und sicherer machen, etwa die Prüfung von Formularen oder die Verschlüsselung von E-Mail-Adressen. Auch für Animationen oder zur Realisierung von multimedialen Anwendungen kann JavaScript verwendet werden. Mit der auf JavaScript basierenden Technologie AJAX (= Asynchronous JavaScript and XML) lassen sich Inhalte einer Website ändern, ohne dass die Seite neugeladen werden muss. Ein typischer Anwendungsfall hierfür ist das automatische Vervollständigen von Suchbegriffen. Ein Kritikpunkt an JavaScript ist die Möglichkeit, die Technologie im Browser deaktivieren zu können. Ist dies der Fall, so würden die JavaScript-Funktionalitäten auf Websites nicht mehr funktionieren. Der Grund dafür ist, dass JavaScript über die letzten Jahre hinweg mehrfach missbraucht wurde, um durch Sicherheitslücken im Browser Schaden anzurichten. Da JavaScript aber eine vielfältige Auswahl an Möglichkeiten bietet und die Vorteile überwiegen, ist die Skriptsprache heute praktisch unverzichtbar. Für Webdesigner\*innen gilt jedoch, die wichtigsten Funktionen einer Website so

zu gestalten, dass sie auch ohne JavaScript funktionieren (Bühler et al., 2017, S.32f.).

### 2.2.4 PHP

Die Abkürzung PHP stand ursprünglich für „Personal Home Page“. Heutzutage ist die Technologie unter „PHP: Hypertext Preprocessor“ bekannt. Bei PHP handelt es sich im Gegensatz zu JavaScript um eine serverseitige Skriptsprache. Dies bedeutet, dass ein Webserver notwendig ist, um PHP-Programme auszuführen. PHP ermöglicht die Entwicklung dynamischer Websites, auf denen die sogenannten Web Applications basieren, wie beispielsweise E-Commerce-Systeme, Chat-Plattformen oder Foren. Im Gegensatz zu statischen Websites, kann sich bei dynamischen Websites der Inhalt entweder durch Interaktion der Nutzer\*innen oder durch neue Informationen aus Datenbanken jederzeit ändern (Theis, 2021, S.18; Rohles, 2017, S.36).

Außerdem lässt sich mit PHP die Auswertung von Formularen, mit welchen Nutzer\*innen Daten an eine Website senden, bewerkstelligen. PHP wird meist im Zusammenspiel mit Datenbanksystemen verwendet. Das bekannteste Datenbanksystem ist MySQL (Theis, 2021, S.18).

## 2.3 Best Practices

In vielen Bereichen des Webdesigns haben sich über die Jahre zahlreiche Best Practices etabliert. Das Befolgen dieser Best Practices bei der Erstellung einer Website erleichtert für Websitebesucher\*innen nicht nur die Nutzung, sondern optimiert auch die technischen Aspekte der Website. Im Folgenden werden einige Best Practices der Bereiche Barrierefreiheit, Suchmaschinenoptimierung und Performance-Optimierung vorgestellt.

### 2.3.1 Barrierefreiheit

Die Web Accessibility Initiative des W3C hat mit den Web Content Accessibility Guidelines (WCAG 2.0) einen Standard entwickelt, der feste Vorschriften hinsichtlich der Barrierefreiheit einer Website festlegt (Hahn, 2020). Diese Vorschriften sind auf der offiziellen Website des W3C unter folgendem Link <https://www.w3.org/Translations/WCAG20-de> abrufbar. Der Begriff Accessibility bedeutet wörtlich übersetzt „Zugänglichkeit“, ist jedoch vermehrt auch unter der Bezeichnung Barrierefreiheit bekannt. Barrierefreiheit im Web zielt darauf ab, dass

den Nutzer\*innen beim Benutzen einer Website keine Hindernisse in den Weg gelegt werden. Vor allem Menschen mit Behinderung, insbesondere blinde oder sehbehinderte Personen, profitieren von Barrierefreien Webdesign, denn für diese Gruppe entstehen oft unüberwindbare Barrieren, welche die Nutzung einer Website stark einschränken oder überhaupt nicht möglich machen. Neben dem Zweck, Menschen mit Behinderungen Zugang zu Websites zu ermöglichen, beschäftigt sich Accessibility auch damit, Nutzer\*innen mit Achtung zu behandeln und ihnen eine gewisse Flexibilität im Umgang mit einer Website zu ermöglichen. Barrierefreie Websites helfen somit allen Nutzer\*innen gleichermaßen (Rohles, 2017, S.51).

Um das Web für Nutzer\*innen mit verschiedenen Behinderungen zugänglicher zu gestalten, muss HTML weitgehend so verwendet werden, wie es ursprünglich beabsichtigt war, nämlich zur Kodierung von Bedeutung und nicht von Aussehen. Man spricht hierbei auch von der Semantik der HTML-Elemente. Solange eine Website semantisch ist, ist es für alternative Browser möglich, die Website auf eine Art und Weise darzustellen, die auf die Bedürfnisse der einzelnen Nutzer\*innen optimiert ist und dadurch die Nutzung des Webs behindertengerecht ermöglicht (Nielsen, 1996).

Die größten Komplikationen hinsichtlich der Zugänglichkeit von Websites treten bei blinden Menschen oder Menschen mit Sehbehinderungen auf, da viele Websites optisch sehr aufwendig gestaltet sind. Besonders auf Websites mit schwachem Kontrast zwischen Vordergrund- und Hintergrundfarben könnten Inhalte für farbenblinde Nutzer\*innen praktisch unlesbar sein. Textbasierte Websites können hingegen für blinde Nutzer\*innen relativ zugänglich sein, da Texte mittels Screenreadern vorgelesen werden können. Problematisch sind jedoch sehr lange Seiten, da es für blinde Nutzer\*innen schwieriger ist, nach interessanten Teilen zu Suchen. Eine geeignete HTML-Struktur kann hierbei Abhilfe schaffen. So sollten `<H1>` Tags ausschließlich für die Überschrift der höchsten Ebene verwendet werden, `<H2>` Tags für die wichtigsten Informationen innerhalb der `<H1>` und `<H3>` Tags für noch feinere Unterteilungen der Informationen. Auf diese Weise kann sich durch Vorlesen der Überschriften mittels des Screenreaders ein Überblick verschafft und uninteressante Abschnitte schnell übersprungen werden (Nielsen, 1996).

Rohles (2017, S.54f.) erwähnt in diesem Zusammenhang eine nützliche Erfindung der Web Accessibility Initiative (kurz WAI). Mit den sogenannten ARIA (= Accessible Rich Internet Applications) Roles lassen sich auf allen Geräten und Programmen, welche ARIA Roles unterstützen, besondere Funktionen

bereitstellen. Beispielsweise kann mit Screenreadern direkt zur Navigation einer Website, beziehungsweise dem `<nav>` Element, gesprungen werden. Das soeben erwähnte `<nav>` Element oder auch das Hauptinhaltelement `<main>` sind Elemente, welche ARIA Roles direkt integriert haben. Es kann aber auch neutralen HTML-Elementen wie dem `<div>` mittels ARIA Role eine bestimmte Rolle zugeteilt werden, wie etwa `<div role="navigation">`. Damit können Browser und Screenreader zwischen dem `<div>` Element für die Navigation und anderen `<div>` Elementen unterscheiden. In der folgenden Tabelle werden häufig eingesetzte ARIA-Roles, samt zugehöriger HTML-Tags, dargestellt.

*Tabelle 1. Wichtige ARIA Roles (in Anlehnung an Rohles, 2017, S.54)*

ARIA Landmark Role	Beschreibung	HTML5-Element
<i>article</i>	Inhaltlich abgeschlossener Abschnitt	<code>&lt;article&gt;</code>
<i>banner</i>	übergreifende Informationen zur Website wie Logo, Titel oder Slogan	<code>&lt;header&gt;</code>
<i>complementary</i>	ergänzende Informationen, die vom Hauptinhalt unabhängig sind	<code>&lt;aside&gt;</code>
<i>contentinfo</i>	Informationen zum Inhalt wie Autor, Copyright oder Fußnoten	<code>&lt;footer&gt;</code>
<i>main</i>	Hauptinhalt	<code>&lt;main&gt;</code>
<i>navigation</i>	Haupt- und Unternavigation	<code>&lt;nav&gt;</code>
<i>note</i>	beiläufig eingeschobene Inhalte	<code>&lt;aside&gt;</code>

Durch Verwendung des ALT-Attributes können Bilder einer Website mit einem alternativen Text ausgestattet werden. Entgegen der Befürwortung einiger Accessibility-Spezialisten von sogenannten beschreibenden Bildern, bei denen Text angegeben wird, welcher verbalisieren soll, was auf den Bildern zu sehen ist, empfiehlt Nielsen (1996) Alternativtext zu schreiben, der die Bedeutung des Bildes im Gesamtkontext erklärt, also was das Bild vermitteln soll und was passiert, wenn es angeklickt wird.

Für taube oder hörgeschädigte Menschen sollten auf Websites mit multimedialen Inhalten Transkripte von gesprochenen Audioclips zur Verfügung gestellt werden.

Videos sollten auch in Versionen mit Untertiteln angeboten werden. Dies kommt auch Menschen zugute, bei welchen die im Video verwendete Sprache nicht ihrer Muttersprache gleicht (Nielsen, 1996).

Benutzer\*innen mit motorischen Behinderungen haben möglicherweise Schwierigkeit präzise Mausbewegungen auszuführen oder mehrere Tasten gleichzeitig zu drücken. Browser ermöglichen es jedoch komplett tastaturgesteuert durch die Links auf einer Website zu navigieren (Nielsen, 1996). Bei Benutzung eines Touchscreen-Displays, beispielsweise am Smartphone, kann es auch zu Problemen bei zu kleinen anklickbaren Bereichen eines Links oder ähnlichen Elementen kommen. Hier sollte auf eine ausreichende Größe, sowie genug Abstand zu anderen Elementen geachtet werden (Hahn, 2020, S.35).

Auch kognitive Beeinträchtigungen können zu Schwierigkeiten mit Websites führen. Nielsen (1996) geht davon aus, dass die meisten Benutzer\*innen wesentlich größere Schwierigkeiten haben werden, sich auf einer Website zurechtzufinden, als die Designer\*innen. Eine vereinfachte Navigation hilft allen Nutzer\*innen, für Menschen mit kognitiven Beeinträchtigungen ist sie unerlässlich. Eine solche Visualisierung in Form einer Sitemap kann eine zusätzliche Hilfe sein.

### 2.3.2 Suchmaschinenoptimierung

Suchmaschinenoptimierung, kurz SEO (= search engine optimization), bezeichnet Maßnahmen zur Optimierung der Sichtbarkeit einer Website und ihrer Inhalte durch Suchmaschinen. Semantischer HTML-Code und Sitemaps sind nicht nur für Nutzer\*innen hilfreich, sondern ermöglichen auch Suchmaschinen ein korrektes Indizieren der Website. Eine Sitemap ist nichts anderes als ein Inhaltsverzeichnis einer Website. Man unterscheidet zwischen HTML-Sitemaps und XML-Sitemaps. Erstere dienen der Orientierung der Nutzer\*innen auf der Website. Für Suchmaschinenoptimierung haben solche Sitemaps den Vorteil, dass sie die interne Verlinkung fördern, denn erst mittels interner Verlinkung werden Unterseiten von Suchmaschinen richtig wahrgenommen. Mittels XML-Sitemaps hingegen kann Suchmaschinen mitgeteilt werden, welche Unterseiten wie häufig und mit welcher Priorität angesteuert werden sollen. Damit jedoch eine Website überhaupt in eine Suchmaschinen-datenbank aufgenommen werden darf, ist eine Indexierung notwendig. Dies kann entweder über das HTML-Tag `<meta name="robots" content="index">` erfolgen oder über eine „robots.txt“ benannte Textdatei mit gleichen Inhalt, welche auf dem Server abgelegt wird. Mit den HTML-Tags Meta Title und Meta Description lässt sich zudem die Überschrift, sowie die Beschreibung des Suchergebnisses anpassen. Für Verlinkungen auf Social-Media

Plattformen sind die Open Graph Tags zuständig, mit denen neben einem Titel und einer Beschreibung auch ein Bild für den geteilten Beitrag definiert werden kann. Schlussendlich ist auch eine schnelle Ladezeit der Website ausschlaggebend für eine gute Platzierung innerhalb der Suchmaschinen (Sens, 2018, S.17f.).

### 2.3.3 Performance-Optimierung

Die Ladezeit einer Website hängt von unterschiedlichen Faktoren ab. Klar ist aber, je kürzer die Ladezeit, desto besser ist die Benutzerfreundlichkeit der Website. Ebenso stellt die Ladezeit einen unwesentlichen Rankingfaktor für Suchmaschinen dar. Vor allem auf mobilen Geräten kann die Ladezeit aufgrund von schwankenden mobilen Verbindungsgeschwindigkeiten sehr stark variieren. In einer 2011 durchgeführten Online-Umfrage hat knapp die Hälfte aller Befragten angegeben, nicht länger als vier Sekunden warten zu wollen, bis eine mobile Website vollständig geladen ist (Equation Research, 2011).

Bestimmte Faktoren wie die Serverleistung oder die Datenverbindung lassen sich nicht durch Webdesigner\*innen beeinflussen, aber es gibt einige Möglichkeiten das Datenvolumen der Website möglichst gering zu halten.

Bilder und JavaScript-Dateien stellen den Großteil des Datenumfangs einer Website dar. HTML und CSS fällt hingegen weniger ins Gewicht. Zu Beginn einer Performance-Optimierung sollte sich demnach zuerst um die Bildoptimierung gekümmert werden. Neben der Wahl eines geeigneten Bildformats, gibt Hahn (2020, S.153) den Tipp in vielen Bereichen auf CSS-Effekte, wie Schatten, Verläufe, usw. und auf SVGs sowie Icon-Fonts, anstelle von Bildern zu setzen. Mit Programmen wie Adobe Photoshop lassen sich Bilder optimiert fürs Web speichern, was eine Verringerung der Dateigröße bewirkt. Auch beim Einsatz von JavaScript sollte dieses frühzeitig auf seine Notwendigkeit überprüft werden. Möglicherweise werden Skripte nicht auf allen Unterseiten einer Website gebraucht. Auch bei der Verwendung mehrerer JavaScript-Frameworks wie etwa jQuery kommt es oft zu einem mehrfachen Einbinden der gleichen Skripte, was in einem mehrfachen Laden der gleichen Daten resultiert.

Beim Entwickeln einer Website ist ein Aufteilen von Skripten und CSS in mehrere Dateien durchaus sinnvoll für eine übersichtlichere und flexiblere Arbeitsweise. Auf diese Weise kommt es jedoch zu einer Vielzahl an Requests (= Serveraufrufe), die eine Website verlangsamen können. Nach Abschluss der Entwicklung einer Website ist es sinnvoller, diese Angaben in einer einzigen Datei unterzubringen.

Auch eine Minimierung der Datei (= Entfernen von Leerzeichen, Absätzen und Kommentaren) kann neben dem Zusammenfassen, eine spürbare Reduzierung der Dateigröße bewirken. Mit dem Komprimierungsverfahren Gzip, bei dem Zeichen eingespart werden, indem auf bereits vorhandene Informationen verwiesen wird, statt sie erneut zu übertragen, lassen sich im Durchschnitt 70% der Daten einsparen. Gzip arbeitet verlustfrei und wird von allen gängigen Browsern unterstützt. Auf vielen Servern ist Gzip bereits vorinstalliert und lässt sich einfach aktivieren (Rohles, 2017, S.458f.).

Beim mehrfachen Aufrufen einer Website werden Dateien immer wieder neu geladen, wenn sie mehrmals benötigt werden. Aber auch wenn diese Dateien über mehrere Unterseiten hinweg gleich sind oder sich über einen längeren Zeitraum hinweg nicht ändern, wie etwa beim Kopfbereich einer Website. In diesem Fall kann der sogenannte Cache, eine im Browser eingebaute Funktion, für Abhilfe schaffen. Dieser bewirkt, dass Dateien nicht erneut vom Server angefragt werden und stattdessen am lokalen System der Nutzer\*innen abgelegt und von dort genutzt werden. Auch Server verfügen über eine Caching-Funktion, welche insbesondere bei dynamischen erzeugten Websites, wie sie etwa beim Einsatz eines Content-Management-Systems vorkommen, hilfreich sind (Rohles, 2017, S.460).

Trotz der zuvor genannten Techniken werden bei einem Website-Aufruf dennoch Daten heruntergeladen. Aber auch während dieser Ladezeit lässt sich die User Experience weiter optimieren. Beispielsweise kann mit dem Verfahren „Lazy Loading“ dem Browser mitgeteilt werden, Ressourcen nicht direkt beim Aufruf der Website zu laden, sondern erst wenn sie tatsächlich benötigt werden. Dies ist besonders sinnvoll bei Bildern, da diese erst im sichtbaren Bereich des Viewports angezeigt werden (Rohles, 2017, S.465).

## 3 Web-Content-Management

Dieses Kapitel beschäftigt sich mit Web-Content-Management und dessen Systemen. Zu Beginn erfolgt eine Definition der wichtigsten Begriffe, sowie eine Abgrenzung von Content-Management-Systemen speziell für das Web zu anderen Systemen. Im Anschluss wird die genaue Funktionsweise von Content-Management-Systemen im Web erklärt. Die Auflistung der Vor- und Nachteile zum Ende des Kapitels soll eine Entscheidung für oder gegen Content-Management-Systeme verdeutlichen.

### 3.1 Begriffsdefinition

Die nachfolgenden Abschnitte erklären die Begriffe Content, Content-Management und Content-Management-Systeme und stellen die unterschiedlichen Arten von Content-Management-Systemen vor.

#### 3.1.1 Content

Der Begriff Content bezeichnet Inhalte, die durch einen redaktionellen Prozess erstellt werden. Unter diesem Prozess wird die Vorbereitung von Informationen zur Veröffentlichung vor einem Publikum verstanden. Dies umfasst das Modellieren, Verfassen, Bearbeiten, Überprüfen, Genehmigen, Versionieren, Vergleichen und Kontrollieren von Inhalten (Barker, 2016, S.18).

Laut Nix (2005, S.25) umfasst Content alle Inhaltsarten in digitaler Form. Demnach beschreibt Content nicht ausschließlich Inhalte, wie etwa Texte, Audio und Video, sondern auch dazugehörige Metadaten wie Autor, Titel oder Inhaltsbeschreibung. Spörrer (2019, S.6) bezeichnet die Möglichkeit, Inhalte mit Metainformationen zu versehen sogar als wesentliches Merkmal von Content. Auch Zugriffsrechte oder Statusinformationen können weitere Metainformationen von Content sein. Mit Metainformationen versehener Content ermöglicht es auch, diesen über eine Suchfunktion einfacher und effektiver aufzuspüren. Content kann entweder bereits vordefinierte Metainformationen enthalten oder später durch die Benutzer\*innen oder das verwendete System mit weiteren Daten versehen werden.

#### 3.1.2 Content-Management

Content-Management bezeichnet den zielgerichteten, systematischen und durchgängigen Umgang mit Informationen. In den meisten Fällen zählt die Erzeugung (Generierung), die Verwaltung (Organisation und Aufbereitung), das Zur-Verfügung-Stellen (Distribution) sowie das Schaffen von Nutzungs- und Verarbeitungsmöglichkeiten (Nutzung) zu den typischen Funktionen von Content-Management. Dadurch werden die notwendigen Voraussetzungen geschaffen, Content einfach, schnell und ohne Probleme auf unterschiedliche Art und Weise zu nutzen (Koop, Jäckel, & Offern, 2001, S.14).

Riggert (2009, S.2) zufolge gleicht das Content-Management einem Lebenszyklus (siehe Abbildung 5). Den Anfang bildet die Idee für den Content, welcher sowohl fremd beschafft oder unternehmensintern erstellt werden kann. Es folgt eine Qualitätssicherung, in welcher Freigabe- als auch Verwaltungsschritte durchgeführt werden, bevor der Content zur weiteren Nutzung und Publikation verfügbar ist. Nach der Veröffentlichung, beziehungsweise wenn der Content ausgedient hat, kann dieser entweder vernichtet beziehungsweise archiviert werden oder nach einer Modifizierungsphase erneut publiziert werden.

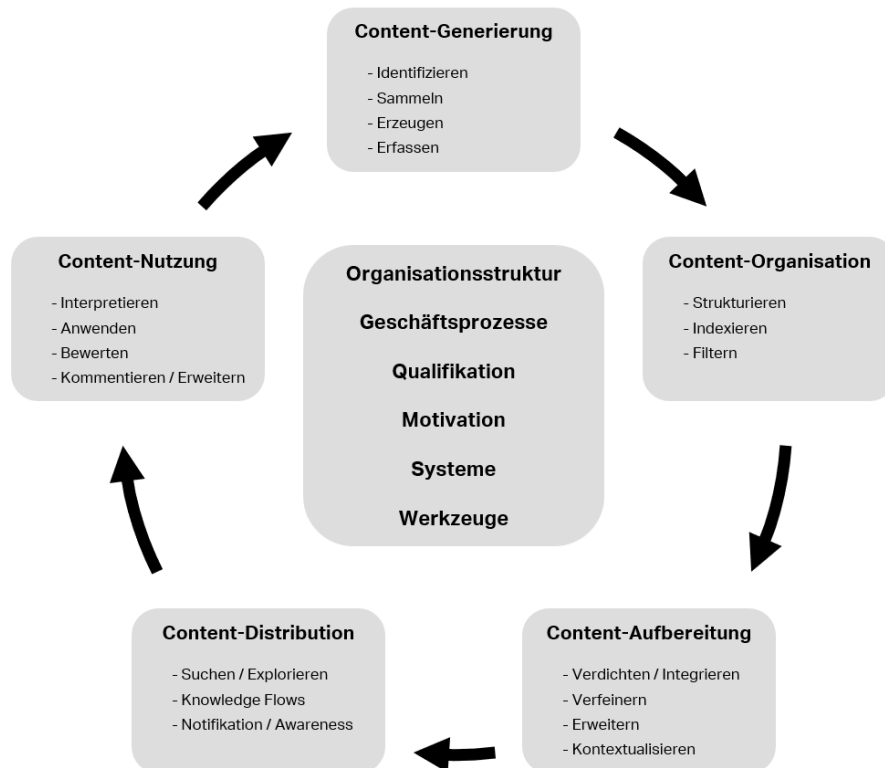


Abbildung 5. Content-Life-Cycle (in Anlehnung an Koop & Jäckel, 2001, S.15)

### 3 Web-Content-Management

---

Neben den drei Hauptfunktionen Erzeugung, Verwaltung und Zur-Verfügung-Stellung nennen Koop, Jäckel & Offern (2001, S.15) folgende weitere Zielsetzungen eines Content-Management:

- Aktualität
- Reliabilität (Zuverlässigkeit)
- Qualitätssicherung
- Gewährleistung inhaltlicher Konsistenz

Darüber hinaus sei Content-Management niemals einem Content-Management-System gleichzusetzen, denn abhängig von Rahmenbedingungen wie dem Umfang oder der Anzahl der Nutzer\*innen, kann Content-Management auch nicht IT-gestützt durchgeführt werden.

#### **3.1.3 Content-Management-Systeme**

Als Content-Management-System wird ein IT-basiertes System zur Organisation, Verwaltung und Durchführung des Content-Managements verstanden (Koop et al., 2001, S.16).

Steyer (2016, S.4) bezeichnet ein Content-Management-System, vermehrt unter der Kurzform CMS bekannt, als System für die Bereitstellung und Verwaltung von Content. Dieser steht im Gegensatz zur Struktur oder der technischen Basis im Mittelpunkt. Bei einem CMS sind der Inhalt, die Struktur und die technische Basis voneinander getrennt.

Barker (2016, S.21) versteht unter einem CMS ein Softwarepaket, das ein gewisses Maß an Automatisierung für die Aufgaben zur effektiven Verwaltung von Inhalten bietet. Ein CMS ermöglicht es Redakteur\*innen neue Inhalte zu erstellen, bestehende Inhalte zu bearbeiten und schlussendlich diese Inhalte anderen Personen zur Verfügung zu stellen. Logischerweise besteht ein CMS daher aus mehreren Teilen. Die Bearbeitungsoberfläche, die Datenbank sowie Mechanismen für die Veröffentlichung von Content oder andere Funktionen können alle separate, autonome Teile des Systems sein. Für technisch nicht versierte Nutzer\*innen werden all diese Teile jedoch als ein großes Ganzes angesehen.

#### **3.1.4 Arten von Content-Management-Systemen**

Content-Management-Systeme werden in unterschiedliche Arten eingeteilt. Barker (2016, S.22) identifiziert die vier Hauptarten folgendermaßen:

#### **Web-Content-Management (WCM)**

Verwaltung von Inhalten für die Massenverbreitung über eine Website. Web-Content-Management-Systeme zeichnen sich durch die Trennung von Inhalt und Aussehen sowie die Veröffentlichung auf mehreren Kanälen aus.

#### **Enterprise-Content-Management (ECM)**

Verwaltung allgemeiner Unternehmensinhalte, welche nicht für die breite Masse bestimmt sind. Diese Variante ist ursprünglich vermehrt unter dem Begriff „Dokumentenmanagement“ bekannt gewesen. Enterprise-Content-Management-Systeme zeichnen sich durch kollaboratives Arbeiten, Zugriffskontrolle und Dateiverwaltung aus.

#### **Digital-Asset-Management (DAM)**

Verwaltung und Bearbeitung sogenannter „Rich Digital Assets“, wie beispielsweise Bilder, Audio und Videos zur Verwendung in anderen Medien. Digital-Asset-Management-Systeme erlauben umfassende Bearbeitung von Metadaten.

#### **Records-Management (RM)**

Verwaltung und Aufbewahrung von Transaktionsinformationen und anderen Aufzeichnungen, die als Nebenprodukt von Geschäftsvorgängen erstellt werden (z.B. Verkaufsunterlagen, Zugangsdaten, Verträge, etc.).

Die Grenzen zwischen diesen Systemen verschwimmen häufig. DAM-Systeme werden oft verwendet, um Inhalte für eine Website durch Integration eines WCM-Systems bereitzustellen. Auch ECM-Systeme verfügen über Funktionen, mit denen sie einen Teil ihrer Informationen im Web veröffentlichen können.

Trotz der unterschiedlichen Arten an Content-Management-Systemen, wird mit der Bezeichnung CMS in den meisten Fällen ein Web-Content-Management-System gemeint. Heutzutage wird bei vielen CMS hoher Wert auf medienneutrale Datenhaltung gelegt. So lassen sich Inhalte auf Wunsch als Druckversion, PDF oder als Website bereitstellen. Ein weiterer Fokus liegt auf der Ausgabe für mobile Geräte. Moderne Open-Source CMS werden von einer Community gepflegt und sind meist auf dem neuesten Stand der Webtechnologie. Somit kann ein CMS benutzerfreundliche Funktionen bereitstellen, zugleich suchmaschinenoptimiert und gegen Attacken abgesichert sein. Die Mehrheit an CMS lässt sich zudem individuell erweitern (Steyer, 2016, S.6).

Im nachfolgenden Teil dieser Arbeit wird unter der Bezeichnung Content-Management-System ein Web-Content-Management-System zur Entwicklung dynamischer Webseiten verstanden.

## 3.2 Web-Content-Management-Systeme

Die meisten Content-Management-Systeme werden serverseitig betrieben und basieren auf modernen Webprogrammiersprachen wie beispielsweise PHP, Ruby oder Python. Auch eine serverseitige Datenbank wie MySQL ist dafür notwendig. CMS generieren dynamische Websites (Wolf, 2019, S.55). Die Funktionsweise von dynamischen Websites sowie der Unterschied zu statischen Websites wird nachfolgend genauer erklärt.

### 3.2.1 Funktionsweise

Grundsätzlich lassen sich zwei verschiedene Möglichkeiten des Zugriffs auf einen Webserver via HTTP (= Hypertext Transfer Protocol) unterscheiden.

#### Statische Websites

Ausgehend von einer Eingabe einer URL in die Adressleiste des Browsers oder dem Klick auf einen Link wird beim statischen Zugriff eine Anfrage (HTTP-Request) an den Zielservers gestellt und nach der angefragten Webseite gesucht. Wenn diese Seite vorhanden ist, wird sie als HTML-Datei an den Client gesendet (HTTP-Response) und vom Webbrowser angezeigt. Jede angeforderte Datei muss also am Webserver abgespeichert und vorhanden sein, was bei besonders großen Webauftritten in einer Vielzahl an Dateien resultiert. Auch bei Änderungen statischer Websites ist es notwendig die betroffene Datei erst lokal am Rechner zu bearbeiten und im Anschluss wieder auf den Webserver hochzuladen. Der Einsatz statischer Websites ist also eher für kleinere Webpräsenzen oder private Homepages empfehlenswert, bei denen Änderungen nur selten notwendig sind (Böhringer et al., 2014, S.244; Wolf, 2019, S.54).

In Abbildung 6 wird der statische Zugriff auf Websites abgebildet. Die roten Buchstaben A bis E stellen dabei die Reihenfolge des Ablaufs dar.

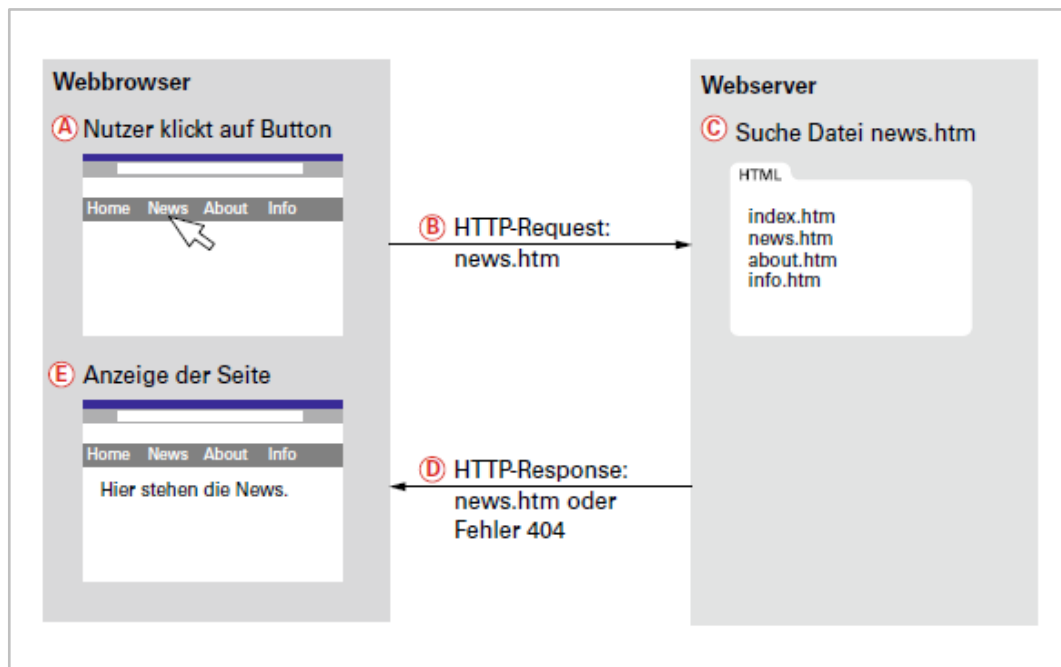


Abbildung 6. Statischer Zugriff (Bühler, Schlaich, Sinner, 2022. S.5)

### Dynamische Websites

Dynamische Websites werden im Normalfall von einem Content-Management-System generiert. Hierbei werden Inhalte wie Texte oder Bilder getrennt von den technischen Elementen aufbewahrt. Beim dynamischen Zugriff wird keine Datei angefragt, sondern ein bestimmter Inhalt. Im Gegensatz zum statischen Zugriff wird am Webserver nicht nach einer Datei gesucht, sondern ein Skript aufgerufen, welches durch einen Interpreter (= die am Server arbeitende Programmiersprache) ausgeführt wird. In dem meisten Fällen handelt es sich hierbei um ein PHP-Skript, das in weiterer Folge die Anfrage nach dem bestimmten Inhalt auswertet, indem es eine Suchanfrage an eine Datenbank (SQL-Query) sendet. Im Anschluss ist es erforderlich, dass die Ergebnisse der Suchanfrage (SQL-Response) durch PHP wieder in HTML-Code umgewandelt werden, da der Browser des Clients PHP-Code nicht interpretieren kann. Zuletzt wird via HTTP-Response wie beim statischen Zugriff eine HTML-Datei an die Nutzer\*innen zurückgesandt, ohne dass diese etwas vom dynamischen Prozess, den ihre Anfrage ausgelöst hat, bemerken. Der große Vorteil dynamischer Websites liegt auf der Hand, Inhalte werden erst auf Anfrage in eine zentrale Datei eingefügt. Alle Inhalte werden darüber hinaus in einer Datenbank verwaltet, welche separat gepflegt wird. Besonders benutzerfreundlich gestaltet sich dieser Prozess durch die Verwendung eines Content-Management-Systems (Böhringer et al., 2014, S.244, Wolf, 2019,

S.57). Abbildung 7 visualisiert den dynamischen Zugriff auf eine Website. Die roten Buchstaben A bis H stellen dabei die Reihenfolge des Ablaufs dar.

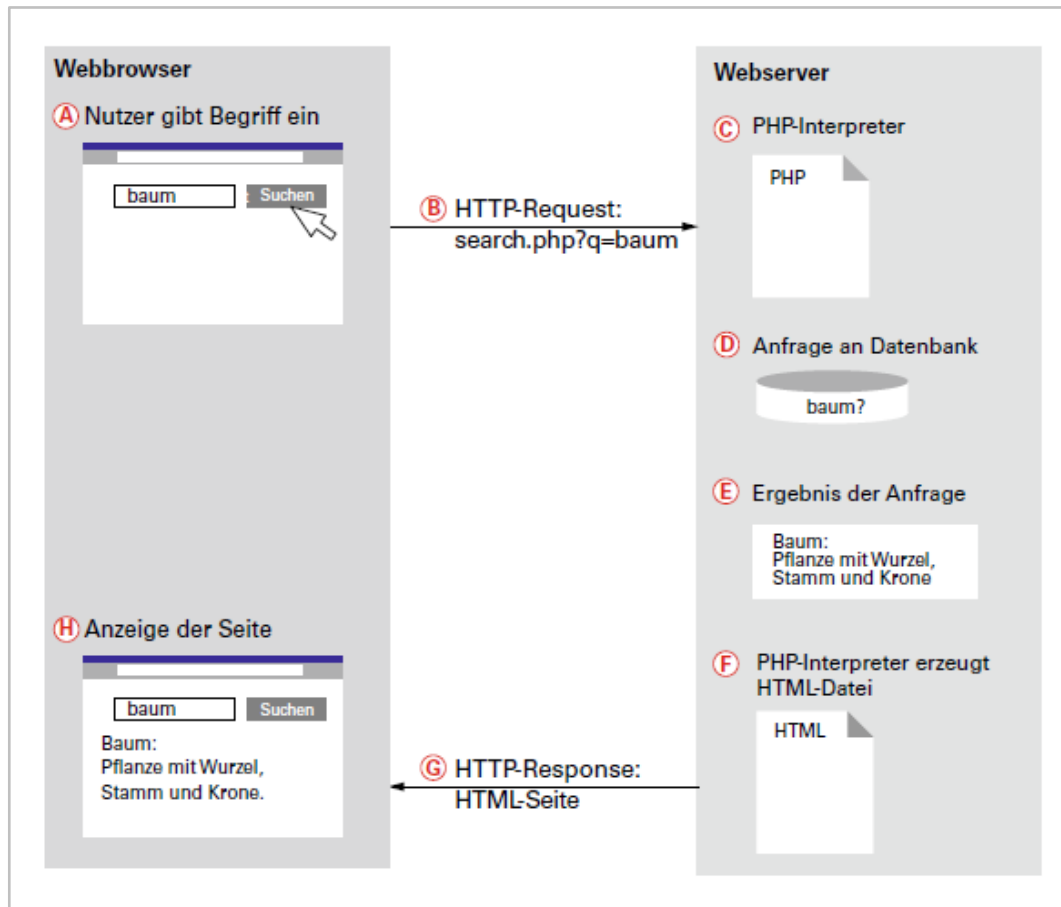


Abbildung 7. Dynamischer Zugriff (Bühler, Schlaich, Sinner, 2022, S.6)

#### 3.2.2 Vor- und Nachteile

Die Verwendung von Content-Management-Systemen bringt sowohl Vorteile als auch Nachteile mit sich. Wenn jedoch eine Website betrieben werden soll, bei der es häufig zu aktualisierende Inhalte gibt, so führt laut Steyer (2016, S.11) kein Weg an einem CMS vorbei.

Die Vorteile eines modernen CMS gegenüber einer konventionell erstellten statischen Website beschreibt Steyer (2016, S.9) folgendermaßen:

- Inhalte lassen sich sehr schnell ergänzen, aktualisieren und löschen
- Zeitliche Verwaltung von Inhalten

### 3 Web-Content-Management

---

- Keine Programmierkenntnisse zur Erstellung und Bearbeitung von Seiteninhalten notwendig
- Optimierung der Inhalte für unterschiedliche Medien möglich
- Unkomplizierte Erweiterung des Systems möglich
- Gemeinsames Arbeiten durch mehrere Benutzer\*innen möglich
- Änderung des Designs über frei verfügbare, professionelle Templates schnell und einfach möglich
- Strikte Trennung von Code, Design und Inhalten
- Aufteilung in Frontend- und Backendbereich (Aufruf über gemeinsame Internetadresse möglich)
- Auswertung diverser Statistiken möglich
- Bereitstellung von Metainformationen
- Erweiterungsmöglichkeiten für unterschiedlichste Anforderungen über bereits vorgefertigte Module (Plugins)

Böhringer, Bühler, Schlaich und Sinner (2014, S.326) ergänzen noch weitere Vorteile:

- Eindeutige Rechtezuweisung durch detaillierte Nutzerverwaltung
- Simple Suchfunktionalität durch Einbindung einer Datenbank
- Freigeben oder Sperren von Publikationen durch autorisierte Personen
- Gewährleistung der referenziellen Integrität der Inhalte

Auch Riggert (2009, S.4) nennt einen zusätzlichen Vorteil von CMS:

- Dynamische Erstellung von Navigationsstrukturen durch das System, Programmierung ist somit nicht erforderlich.

Den Vorteilen von Content-Management-Systemen stehen aber auch Nachteile gegenüber. Steyer (2016, S.10) nennt folgende Gründe, die gegen die Verwendung eines CMS sprechen:

- Absolute Abhängigkeit vom verwendeten CMS. Wechsel auf anderes System ist meist mit sehr hohem Aufwand verbunden.
- Hohe Abhängigkeit vom technischen Aufbau des CMS. Wenn das CMS für ein bestimmtes Problem keine Lösung bereitstellt, so muss eine eigens entwickelte Lösung in das System integriert werden.
- Infrastruktur bestehend aus Webserver, Programmierumgebung und Datenbank für den Betrieb notwendig

### 3 Web-Content-Management

---

- Systemadministrator sowie Einarbeitung in die Arbeitsweise des Systems notwendig
- Mögliche nicht gebrauchte Funktionalitäten und erhöhte Komplexität durch vorgegeben Strukturen

Wolf (2019, S.37) erwähnt an diesem Punkt noch weitere Nachteile von dynamischen Websites:

- Höhere Kosten beim Webhosting wegen spezieller Features wie Skriptsprachen und Datenbanken
- Programmierkenntnisse notwendig, wenn spezielle Plugins entwickelt werden müssen, was in einer längeren und teureren Entwicklung resultiert.

## 3.3 Das Web-CMS WordPress

Dieses Unterkapitel beschäftigt sich mit dem Content-Management-System WordPress. Neben einer kurzen Erklärung des Systems und einer allgemeinen Funktionsbeschreibung wird besonderer Fokus auf den Prozess der Websitegestaltung mit WordPress gelegt. Dabei werden auch die Neuerungen und Änderungen dieses Prozesses erklärt, welche die letzten zwei großen WordPress Versionen mit sich gebracht haben.

Laut eigenen Angaben basieren 43% aller Websites im Internet auf dem Content-Management-System WordPress. (WordPress, 2022a) Das CMS verzeichnet im Juli 2022 einen Marktanteil von 64%. Damit ist WordPress der Konkurrenz weit überlegen und führt somit die Rangliste der meistgenutzten Content-Management-Systeme an (W3Techs, 2022).

WordPress ist ein Open Source Content-Management-System. Das bedeutet, dass der Quelltext des Systems für jeden frei einsehbar ist und ohne jegliche Einschränkung verändert, kopiert und weiterverbreitet werden darf. Die meisten Softwareprogramme heutzutage sind lizenziert, was rechtliche Bestimmungen für die Nutzung oder Verbreitung der Software bedeutet. Es werden verschiedene Arten von Softwarelizenzen verwendet, welche von sehr restriktiv bis hin zu wenig restriktiv reichen. WordPress ist unter der GNU General Public License lizenziert, welche eine der am wenigsten einschränkenden Lizenzen darstellt. Nutzer\*innen von WordPress müssen daher mit keinen bedeutenden Einschränkungen bei der Verwendung rechnen (Steyer, 2016, S.11; Sabin-Wilson, 2019, S. 23).

### 3 Web-Content-Management

---

Die Anfänge von WordPress reichen bis zum Jahr 2003 zurück. WordPress wurde als Fork (= Entwicklungszweig nach der Aufspaltung eines Softwareprojekts) auf Basis des Weblogsystems b2n/cafelog von Matthew Mullenweg und Mike Little entwickelt. Ziel war es ein gut durchdachtes Personal Publishing System zu erschaffen. Somit war WordPress in den ersten Versionen voll und ganz auf die chronologische Darstellung von Beiträgen (Blogs) spezialisiert. Mit Version 1.5 wurde das Verwalten von statischen Seiten hinzugefügt, was seit diesem Zeitpunkt auch das Erstellen von konventionellen Websites ermöglichte. Im Laufe der Zeit hat sich WordPress stets weiterentwickelt und lässt sich heute von Menschen mit begrenzter technischer Erfahrung „out of the box“ verwenden und von technisch versierten Personen auf ihre Bedürfnisse anpassen und erweitern. Heutzutage lassen sich mit WordPress einfache Websites und Blogs aber auch komplexe Portale und Unternehmenswebsites erstellen. Auch die Entwicklung von Applikationen ist mit dem Content-Management-System möglich (Steyer, 2016, S.12f.; WordPress, 2022).

WordPress ist heute in zwei unterschiedlichen Versionen erhältlich. Einerseits gibt es die gehostete Version auf WordPress.com, welche keinerlei Installation erfordert. Andererseits ist eine Version in Form von Selbsthosting auf WordPress.org verfügbar, die etwas mehr Freiheit und Flexibilität erlaubt (Sabin-Wilson, 2019, S.49).

#### **Fremdhosting über WordPress.com**

Da WordPress.com eine Hosting-Lösung ist, bedarf es keinem Erfüllen von Systemanforderungen, keinem Herunterladen und keiner Installation oder Serverkonfiguration. Neben der kostenlosen Standardversion sind zusätzliche, kostenpflichtige Upgrades erhältlich, die einen erweiterten Funktionsumfang bieten. Diese Version von WordPress hat einige Limitierungen im Gegensatz zur selbstgehosteten Version. Es ist nicht möglich, Plugins oder benutzerdefinierte Themes zu installieren. Auch das Editieren des Codes, der Einsatz von Werbung oder Monetarisierung sind ohne das Upgrade auf einen kostenpflichtigen Plan nicht möglich. Ein Umzug auf die selbstgehostete Variante ist jedoch möglich und erlaubt sogar die Mitnahme der Inhalte (Sabin-Wilson, 2019, S.51f.).

Der große Vorteil dieser Version liegt darin, dass sich Nutzer\*innen nicht selbst um Updates oder Sicherheitslücken kümmern müssen (Eisenmenger, 2019, S.42).

### **Selbsthosting von WordPress**

Die selbstgehostete Version von WordPress erfordert eine Installation der Software auf einem eigenen Webserver. Sollte kein eigener Webserver zur Verfügung stehen, so muss dieser von einem Webhosting-Unternehmen gemietet werden, wofür monatliche Kosten anfallen. Für die Installation von WordPress müssen die Systemvoraussetzungen eingehalten werden (Sabin-Wilson, 2019, S.53f.).

Die zum Zeitpunkt der Durchführung dieser Masterarbeit neueste WordPress Version 6.0 erfordert einen Webserver mit PHP in der Version 7.4 oder höher sowie eine MySQL Datenbank mit der Version 5.7 oder alternativ MariaDB mit der Mindestversion 10.3. Außerdem muss HTTPS-Support vom Webhoster gegeben sein (WordPress, 2022b).

Die selbstgehostete Version von WordPress hat den großen Vorteil, dass die Nutzer\*innen die volle Kontrolle über alle Aspekte des Systems haben. Somit können Themes und Plugins hinzugefügt und bearbeitet werden, sowie die Dateien des Systems selbst angepasst werden. Diese umfassende Kontrollmöglichkeit hat jedoch den Nachteil, dass die Software selbst und ebenfalls die installierten Themes und Plugins auf dem neuesten Stand gehalten werden müssen (Krol, 2019, S.25).

#### **3.3.1 Websitegestaltung mit WordPress**

Die Gestaltung einer Website mit WordPress erfolgt durch sogenannte Themes. Ein Theme ist das Zusammenspiel aus Design und Layout des Frontends der Website. Das Theme regelt Gesamtlayout, Formularelemente, Schriftgrößen sowie Abstände und Farben und stimmt diese Gestaltungselemente aufeinander ab, um ein harmonisches Gesamtbild zu schaffen. Eine frische WordPress-Installation bringt bereits hauseigene Themes mit sich. Ebenso können auch Drittanbieter\*innen-Themes direkt im System durchsucht und installiert werden. Auch die Installation eigens entwickelter oder aus dem Internet heruntergeladener Themes ist möglich (Eisenmenger, 2019, S. 367).

Aktiviere Themes können mittels Customizer über eine Live-Vorschau der Website weiter angepasst werden. Welche Elemente verfügbar sind und wie sehr diese verändert werden können, hängt vom ausgewählten Theme ab. Ändern der generellen Website-Informationen wie dem Logo, Titel oder Untertitel sowie die Möglichkeit benutzerdefiniertes CSS hinzuzufügen, sind Standardfunktionen jedes

Themes. Weitere Anpassungsmöglichkeiten müssen explizit von den Theme-Entwickler\*innen hinzugefügt werden. Beispiele für häufig änderbare Theme-Einstellungen sind etwa Layoutbreite, Hintergrundfarbe, Schriftfarben oder Schriftarten. Zudem lassen sich über den Customizer Navigationsmenüs erstellen, sowie die Position der Navigationsmenüs im Theme festlegen. Auch Widgets (= Komponenten für zusätzliche Funktionen) lassen sich über den Customizer an verschiedenen Positionen, wie beispielsweise der Seitenleiste oder dem Footer des Themes platzieren. Die Anzahl der verfügbaren Positionen hängt wieder vom ausgewählten Theme ab (Eisenmenger, 2019, S.375f.).

Die Trennung von Inhalt und Design, die eines der Qualitätsmerkmale von CMS ist, ist auch in WordPress stark vertreten. Auf der einen Seite ermöglicht WordPress die bequeme Pflege und Organisation von Inhalten über die Menüpunkte „Beiträge“ und „Seiten“. Demgegenüber steht auf der anderen Seite das Theme mit Layout- und Designvorlagen. Im Hintergrund wird die fertige Website aus vorgefertigten HTML-Webseitenvorlagen mit Platzhaltern, welche durch den Inhalt ergänzt werden, zusammengebaut (Eisenmenger, 2019, S.372).

Wenn Inhalte in WordPress erstellt oder bearbeitet werden, so wird normale Textverarbeitung betrieben, denn in WordPress steht ein integrierter Texteditor zur Verfügung, der ein Formatieren und Gestalten von Texten ermöglicht, wie man es beispielsweise aus Microsoft Word kennt. Man spricht hier von einem sogenannten WYSIWYG-Editor (= What You See Is What You Get). Dieser Bearbeitungsmodus ist bei normalen Textverarbeitungsprogrammen seit Jahren üblich. Im Falle von WordPress wird jedoch im Browser gearbeitet, was bedeutet, dass der geschriebene Text erst einmal mit HTML-Elementen, CSS und JavaScript abgebildet werden muss, bevor er im Frontend der Website ersichtlich ist (Steyer, 2016, S.99).

Da der Einsatz von WYSIWYG-Editoren Limitierungen mit sich bringt, ist die Bearbeitungsansicht des Editors oft nur eine Annäherung an das finale Ergebnis der Frontend-Darstellung. Seit der Version 5 von WordPress wurde der klassische Editor durch einen neuen Editor mit dem Codenamen „Gutenberg“ ersetzt, welcher ein komplett überarbeitetes Aussehen und Verhalten mit sich bringt (Eisenmenger, 2019, S.218).

#### **3.3.2 Block-Editor und Full-Site-Editing**

WordPress ermöglicht mit der Einführung des Block-Editors „Gutenberg“ eine bequemere Bearbeitung und Pflege der Website-Inhalte. Page-Builder-Themes

### 3 Web-Content-Management

und Plugins verfolgen den Ansatz, alle Bereiche einer Website anpassen zu können, zwar schon etwas länger, doch auch WordPress selbst geht mit dem neuen Editor einige Schritte auf diesem Weg. Auf die erwähnten Page-Builder-Erweiterungen wird im Kapitel 3.4 im Detail eingegangen. Mit dem Block-Editor steht den Nutzer\*innen ein erweiterbarer Baukasten mit Inhalts-Bausteinen, den sogenannten „Blöcken“ zur Verfügung. Im Gegensatz zum alten Editor, bei dem alle Elemente in ein einzelnes Inhaltselement verpackt wurden, sind beim Block-Editor verschiedenste Inhaltselemente wie Absätze, Bilder, Videos etc. eigene Blöcke, welche individuelle Einstellungsmöglichkeiten besitzen und untereinander oder nebeneinander platziert werden können. Dieses flexiblere Bearbeitungsverhalten ermöglicht eine detailgetreue Darstellung im Frontend. Auch die Konkurrenzsysteme von WordPress setzen heutzutage auf ähnliche Baukastensysteme, welche das Hinzufügen verschiedenster Elemente per Mausklick ermöglichen, denn dieser Mechanismus ist benutzerfreundlich, reduziert Fehler, ist erweiterbar und bietet neue Gestaltungsmöglichkeiten. Dieser mit Version 5 von WordPress veröffentlichte Editor ist jedoch erst der Anfang einer weitreichenden Anpassung des Bearbeitungskonzepts von WordPress-Websites (Eisenmenger, 2019, S.219). Die folgende Abbildung zeigt das User-Interface des Block-Editors, mitsamt diverser Inhaltselemente und Einstellungsmöglichkeiten.

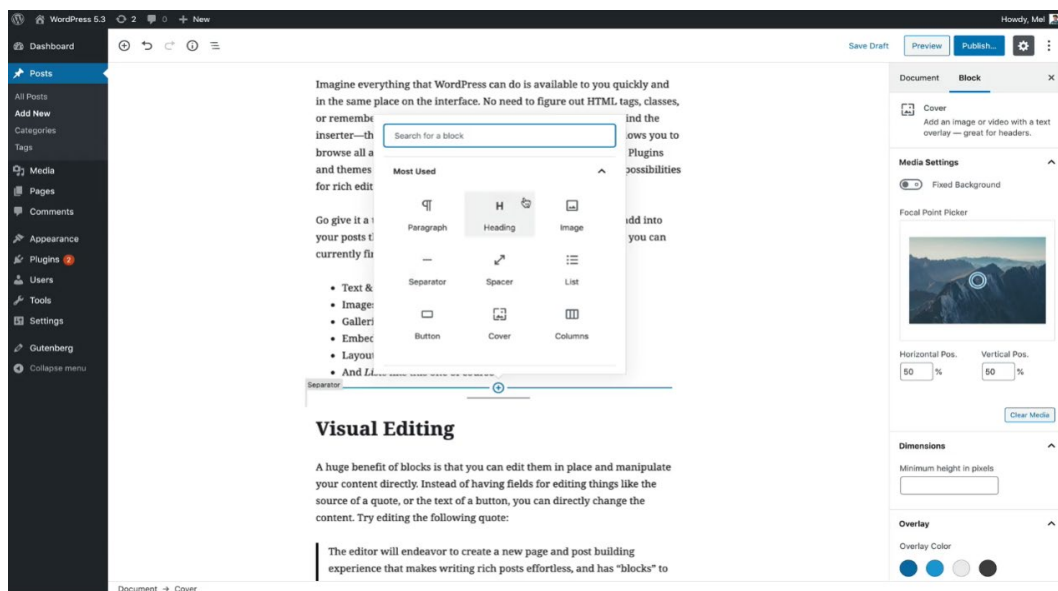


Abbildung 8. User-Interface des Block-Editors (WordPress, 2022)

Der Block-Editor ist der erste Schritt eines vierstufigen Entwicklungsprozesses, welcher das Editieren und Veröffentlichen von WordPress-Websites

revolutionieren soll. Zur Zeit des Schreibens dieser Arbeit befindet sich die Entwicklung in Phase Zwei, welche sich auf die Anpassungsmöglichkeiten zur Gestaltung einer WordPress-Website konzentriert. Ziel des Projekts „Full Site Editing“ ist es, die Möglichkeiten des Block-Editors auch an anderen Stellen in WordPress nutzen zu können, die über das reine Editieren von Seiteninhalten hinausgeht. Mit anderen Worten bedeutet das, die gesamte Website über eine grafische Benutzeroberfläche anpassbar zu machen. Ermöglicht wird dies durch die Einführung von Block-Themes. Ein Block-Theme verwendet dieselben Baukasten-Elemente des Block-Editors für alle Teile einer Website, einschließlich Navigationsmenüs, Kopf- und Fußzeilen, sowie den gesamten Inhaltsbereich (WordPress, 2020).

### 3.4 Page-Builder-Plugins für WordPress

Page-Builder-Plugins stellen einen erweiterten Funktionsumfang bei der Bearbeitung von Seiten und Beiträgen in WordPress bereit. Mittels Page-Builder lassen sich ohne Programmierkenntnisse per Drag & Drop individuelle Layouts erstellen. Dies geschieht im Gegensatz zum herkömmlichen Editor in WordPress in einer Live-Ansicht im Frontend der Website. Die meisten Page-Builder erlauben einen Aufbau der Website mittels Abschnitten, welche wiederum in Spalten aufgeteilt werden können. In diese Spalten werden die einzelnen Elemente wie zum Beispiel Textblöcke, Bilder oder Buttons eingefügt. Darüber hinaus ist jedes Element individuell anpassbar (Rieder, 2022).

Als Basis für einen Page-Builder wird immer noch ein Theme benötigt. Die Kompatibilität zwischen Theme und Page-Builder unterscheidet sich von Theme zu Theme. Einige der Page-Builder-Plugins bieten jedoch auch eigene Themes an, welche besonders gut mit dem eingesetzten Page-Builder harmonieren. Auch gibt es Themes, welche Page-Builder-Funktionalitäten eingebaut haben, die über den Customizer angepasst werden können, diese verfügen aber üblicherweise nicht über die Flexibilität und den Funktionsumfang von Stand-Alone Page-Builder-Plugins (Rieder, 2022).

#### **Vorteile**

Die Verwendung eines Page-Builders bietet laut Rieder (2022) einige Vorteile:

- Page-Builder ermöglichen ein einfaches und intuitives Erstellen und Anpassen von Websites.

### 3 Web-Content-Management

---

- Kleine Veränderungen gehen sehr schnell von der Hand, da kein Code angepasst werden muss, das summiert sich zu einer merkbaren Zeitersparnis.
- Viele Page-Builder bringen Funktionen mit, die den Einsatz anderer Plugins hinfällig machen.
- Page-Builder liefern meistens vorgefertigte Templates mit und bieten die Möglichkeit eigene Designs als Vorlagen abzuspeichern.
- HTML- und CSS-Kenntnisse sind bei der Verwendung von Page-Builder-Plugins nicht erforderlich.

Prohl (2022) ergänzt noch einen weiteren Vorteil:

- Die Kosten für einen Page-Builder fallen im Vergleich zu einer individuell entwickelten Lösung deutlich geringer aus.

#### **Nachteile**

Page-Builder bringen jedoch auch folgende Nachteile mit sich (Rieder, 2022):

- Durch die Verwendung von Page-Builder-Plugins entsteht ein Lock-In-Effekt. Bei der Deaktivierung einiger Page-Builder verschwindet das erstellte Design, angelegte Shortcodes bleiben jedoch bestehen und müssen manuell gelöscht werden.
- Wie alle anderen Plugins beinhalten Page-Builder jede Menge Skript- und Styledateien, die sich negativ auf die Ladezeit einer Website auswirken.

Prohl (2022) und Eisenmenger (2019, S.296) listen weitere Nachteile von Page-Builder-Plugins auf:

- Die einfache Bedienung von Page-Builder-Plugins lockt viele selbsternannte Webdesigner\*innen an, welche nicht über ausreichend Expertise von der Erstellung einer Website verfügen. Das kann unter Umständen die Website selbst aber auch sensible Daten gefährden.
- Jedes Plugin stellt ein potenzielles Sicherheitsrisiko für eine Website dar. Die große Verbreitung von Page-Builder-Plugins macht sich zu einem attraktiven Ziel für Hackangriffe.
- Page-Builder-Plugins halten sich nicht an das CMS-Konzept Inhalte vom Design zu trennen. Bei einer Vielzahl an Inhalten kann somit schnell der Überblick verloren gehen.

## 4 Low-Code- / No-Code-Programmierstil

In diesem Kapitel wird das Low-Code- / No-Code-Programmierparadigma vorgestellt sowie eine Abgrenzung zwischen den beiden Begriffen geschaffen. Aufgrund der Aktualität dieses Themas liegen kaum gedruckte Fachpublikationen vor. Aus diesem Grund wird in diesem Kapitel vorwiegend auf Artikel von Recherche- und Marktforschungsunternehmen sowie auf fachspezifische Online-Blogs zurückgegriffen.

Bei der Low-Code-Entwicklung werden visuelle Hilfsmittel wie grafische Entwicklungsumgebungen zur Erstellung von Software eingesetzt. Mithilfe dieser speziellen Entwicklungsumgebungen werden Anwendungen aus einzelnen Komponenten erstellt. Zum Beispiel lässt sich die Präsentationsschicht einer Anwendung in einem dafür vorgesehenen Editor aus unterschiedlichen UI-Elementen beliebig zusammenfügen. Darüber hinaus stellen Low-Code-Entwicklungsplattformen standardisierte Bausteine zur Verfügung, welche ebenso die Entwicklung der Geschäftslogik und der Datenschicht mit visuellen Mitteln ermöglichen. Im Idealfall wird hierfür keine Zeile Code selbst geschrieben. Bei komplexeren Anwendungen lassen sich individuelle Anpassungen und eigener Code gezielt einpflegen, um den Funktionsumfang an der entsprechenden Stelle zu erweitern (Augsten, 2019).

Der Begriff „Low-Code“ ist noch relativ jung. Geprägt wurde er 2014 von dem Marktforschungsunternehmen Forrester. Um den heutigen Ansprüchen der Kund\*innen und den schnell wechselnden Marktbedingungen gerecht zu werden, bedarf es laut Forrester gravierender Umstrukturierungen im gesamten Softwareentwicklungsprozess. Traditionelle Programmierung sei zu langsam, um Anwendungen zu entwickeln und bereitzustellen, die Unternehmen nutzen um Kund\*innen zu gewinnen und an sich zu binden. Einige Unternehmen wenden sich deshalb den neuen sogenannten „Low-Code-Plattformen“ zu, welche die Bereitstellung von Anwendungen beschleunigen, indem sie den Aufwand der manuellen Programmierung drastisch reduzieren. Neben diesem Hauptvorteil helfen Low-Code-Plattformen den Unternehmen aber auch dabei schneller auf Kundenfeedback nach der Veröffentlichung zu reagieren sowie mobile und Multichannel-Anwendungen bereitzustellen (Richardson & Rymer, 2014).

## 4.1 Historische Einordnung und Entstehung

Der Ansatz hinter dem Low-Code-Programmierstil, also Anwendungen mit möglichst wenig Programmieraufwand zu erstellen, ist jedoch schon deutlich länger verbreitet.

### **Rapid Application Development (RAD)**

Mit Rapid Application Development Programmierumgebungen war es Entwickler\*innen in den 1990er Jahren bereits möglich, Anwendungen visuell zusammenzusetzen, dies galt sowohl für Elemente der Benutzeroberfläche als auch für die dahinterstehende Logik. Solche RAD-Tools hatten jedoch die Einschränkung, dass ein Verlassen der Umgebung des eingesetzten Tools nicht oder nur eingeschränkt möglich war. Auch ein kollaboratives Arbeiten an einer Anwendung, war durch den Mangel an modularem Aufbau kaum möglich. Im 21. Jahrhundert wurden RAD-Umgebungen gänzlich durch neuere Technologien in unterschiedlichsten Bereichen ersetzt (Dewanto & Klein, 2018).

### **Model Driven Software Development (MDSD)**

Das Ziel der modellgetriebenen Softwareentwicklung war es, die Lücke zwischen Fachpersonal und Entwickler\*innen durch den Einsatz von Modellierungssprachen zu verringern. Anwendungen basierten weniger auf einer Programmiersprache, sondern vermehrt auf einer fachspezifischen Modellierungssprache, was den Vorteil hat, dass diese auch von Fachpersonal einfacher zu verstehen und beherrschen ist. Diese Modelle sollten einen ganzheitlichen und gemeinsamen Blick auf die Domäne ermöglichen, was sowohl die technischen als auch insbesondere die fachlichen Anforderungen einbezieht. Im Anschluss wurden Generatoren und Interpreter sowohl während der Entwicklung als auch zur Laufzeit eingesetzt, um den eigentlichen Code so weit wie möglich aus dem Modell zu generieren beziehungsweise zu interpretieren. Ausschließlich der Anteil, welcher nicht als Modell darstellbar ist, wurde manuell und häufig textuell programmiert (Dewanto & Klein, 2018).

### **Low-Code-Plattformen**

Das Prinzip hinter Low-Code ist es, all diese zuvor genannten Konzepte zu vereinen. Low-Code-Plattformen enthalten die Vorgehensweise von RAD und MDSD und berücksichtigen zudem den gesamten Makro- und Mikro-Entwicklungsprozess (= Entwicklungs-, Test- und Produktionsumgebung). Eine umfassende Low-Code-Plattform sollte alle dargestellten Aspekte der

Softwareentwicklung abdecken. Die Möglichkeit, sämtliche Funktionen als Cloud-Service zu verwenden, erleichtert zudem den Einstieg in eine solche Low-Code-Plattform (Dewanto & Klein, 2018).

### **4.2 Unterschied Low-Code & No-Code**

Neben dem Begriff Low-Code findet sich heutzutage vermehrt in Marketingmaterialien von Herstellern auch immer öfter der Begriff No-Code. Low-Code- und No-Code-Systeme bieten dieselben grundlegenden Vorteile, aber ihre Namen weisen auf den Hauptunterschied zwischen diesen beiden Methoden der Anwendungsentwicklung hin.

Bei der Low-Code-Entwicklung müssen die Benutzer\*innen ein gewisses Maß an eigenem Coding vornehmen, auch wenn dies aber in wesentlich geringerem Umfang als bei der herkömmlichen Softwareentwicklung ausfällt. Professionelle Entwickler\*innen nutzen Low-Code-Systeme in erster Linie, um Anwendungen schnell bereitzustellen aber auch um herkömmliche Programmieraufgaben auf andere Arbeiten zu verlegen, die größere Auswirkungen und höheren Wert für das Unternehmen haben. Auch nicht IT-Fachleute mit nur geringen Programmierkenntnissen nutzen Low-Code-Tools, um einfache Anwendungen oder erweiterte Funktionen innerhalb einer Anwendung zu entwickeln. Die No-Code-Entwicklung hingegen zielt auf nicht technikaffine Nutzer\*innen in verschiedenen Unternehmensfunktionen ab, die zwar die geschäftlichen Anforderungen und Regeln verstehen, aber keine Erfahrung mit der Programmierung haben und keine Programmiersprachen beherrschen (Pratt, 2021).

Es gibt auch Unterschiede in der Art und Weise wie No-Code und Low-Code angewendet werden. Wobei No-Code in der Regel zur Erstellung von Anwendungen für einfache Funktionen eingesetzt wird, wird Low-Code zur Entwicklung von Anwendungen, die geschäftskritische Prozesse abbilden oder Kernfunktionen eines Unternehmens ausführen sollen, verwendet (Pratt, 2021).

Die Grenze zwischen No-Code und Low-Code ist nicht immer eindeutig und das lässt sich auch auf die No-Code- und Low-Code-Plattformen selbst übertragen. No-Code wird vielfach als Teil des Low-Code-Marktes angesehen, da selbst die fortgeschrittensten Plattformen ein gewisses Maß an Code für Teile der Anwendungsentwicklung und des Bereitstellungsprozesses erfordern. Die Unterscheidung zwischen Low-Code und No-Code wird zum großen Teil von den

Anbieter\*innen selbst vorgenommen, da diese ihre Produkte für unterschiedliche Zielgruppen positionieren (Pratt, 2021).

Wong und Vincent (2022) des Marktforschungsunternehmens Gartner bezeichnen No-Code als reinen Marketing-Begriff. Im Grunde genommen gäbe es so etwas wie No-Code gar nicht. Tools, welche sowohl als Low-Code als auch als No-Code bezeichnet werden, sind vom Ansatz her einfach nur Low-Code-Plattformen. Der No-Code-Aspekt bedeutet lediglich, dass das Tool für bestimmte Anwendungsfälle keinen eigenen Code erfordert. Dies gilt jedoch in der Regel auch für die meisten Low-Code-Tools am Markt.

Laut Wong und Vincent (2022) bedeutet No-Code aber nicht unbedingt, dass keine technischen Kenntnisse erforderlich sind. No-Code impliziert zwar, dass keine Programmiersprache verwendet werden muss, jedoch erfordern einige visuelle No-Code-Tools dennoch technisches Fachwissen oder ein generelles Verständnis von Programmiermetaphern.

### 4.3 Zielgruppe

Eine exakte Definition der Zielgruppe von Low-Code- und No-Code-Systemen ist nicht möglich, da mit der ständig steigenden Anzahl an Plattformen für die unterschiedlichsten Anwendungszwecke auch die Zielgruppe mitwächst. Manche Systeme richten sich an Endnutzer\*innen, andere wiederum an IT-Spezialist\*innen. Es wird jedoch deutlich, dass die meisten Plattformen genau auf einen Personenkreis zwischen den soeben Genannten abzielen.

Für diese Gruppe an Personen hat sich der Begriff „Citizen Developer“ etabliert. Citizen Developer bedeutet übersetzt etwa „Fachbereichsentwickler“ und beschreibt technisch versierte Mitarbeiter\*innen, welche Softwareanwendungen für ihren jeweiligen Fachbereich erstellen. Dabei arbeiten diese mit standardisierten Entwicklungsumgebungen, wie etwa Low-Code-Plattformen. Im Gegensatz zu professionellen Softwareentwickler\*innen verfügen Citizen Developer über keine speziellen IT- oder Programmierkenntnisse und sind dadurch auf derartig unterstützende Tools angewiesen. Mit ihrem speziellen Fachwissen kennen sie jedoch die Anforderungen und Prozesse ihrer jeweiligen Abteilung und sind darauf spezialisiert, kontextrelevante Softwarelösungen zu entwickeln (Etemadian, 2022; Gartner, 2022).

Dem Prozess des Citizen Developments können laut Liptak und Horwitz (2022) folgende Vor- und Nachteile zugeschrieben werden:

**Vorteile:**

- Citizen Development mittels Low-Code- oder No-Code-Plattformen kann die IT-Abteilung deutlich entlasten und ermöglicht es den IT-Mitarbeiter\*innen, sich um andere Aufgaben zu kümmern.
- Citizen Development schafft einen größeren Pool von Mitarbeiter\*innen, welche Ressourcen und Entwicklungstools gemeinsam nutzen können. Das reduziert den Bedarf an IT-Fachpersonal und die damit verbundenen Kosten.
- Der Einsatz von Low-Code- oder No-Code-Plattformen steigert die Produktivität und Effizienz, da diese die Erstellung von Anwendungen beschleunigen.

**Nachteile:**

- Aufgrund ständig wechselnder Technologien und Aktualisierungen von Tools und Programmen bedarf es einer konsequenten Schulung und Umschulung der Citizen Developer, um mit diesen Änderungen Schritt halten zu können.
- Wenn die Arbeit von Citizen Developern nicht durch IT-Fachleute überwacht wird, könnten Anwendungen mit geringer Qualität entwickelt werden. IT-Spezialist\*innen bleiben von der Entwicklung bis hin zum Testen des Endproduktes ein wichtiger Faktor für die Gewährleistung der Qualität und Funktionalität der entwickelten Anwendung.

## 5 No-Code-Plattformen

No-Code-Plattformen sind seit einigen Jahren stark im Trend. Viele Unternehmen haben sich in den letzten Jahren der No-Code-Bewegung angeschlossen und ihre eigenen Ableger an No-Code-Tools auf den Markt gebracht. Darunter vertreten sind Global Player wie Google, Amazon und Microsoft, welche No-Code-Tools aufgekauft oder selbst entwickelt haben. Dabei handelt es sich vorwiegend um Tools zur Erstellung interner Geschäftsanwendungen für die Verwaltung von Lagerbeständen oder Geschäftsabläufen. Gleichzeitig haben Anbieter\*innen wie Wix und Webflow im Bereich der No-Code-Webentwicklung stark zugenommen. Auch No-Code-Tools zur Anwendungserstellung im traditionellen Sinne, gibt es heutzutage zuhauf. Nennenswerte Vertreter in diesem Bereich sind beispielsweise Bubble oder Adalo (ElBatanony & Succi, 2021, S.29).

### 5.1 Marktüberblick

Der Markt an No-Code-Tools ist sehr dynamisch und wächst ständig weiter. Viele Plattformen bieten vergleichbare Basisfunktionen und unterscheiden sich vorwiegend über Zusatzfunktionalitäten für spezifische Anwendungsszenarien. Darüber hinaus bieten einige Anbieter\*innen selbst Marktplätze für Erweiterungen und Integrationen an. Die Wahl der richtigen Tools ist nicht einfach, da sich ein und dasselbe Geschäftsmodell mit unterschiedlichen Kombinationen an Tools umsetzen lässt. Eine Kategorisierung verschiedener No-Code-Tools nach ihrem Einsatzzweck sieht laut Sachs (2021, S.34) folgendermaßen aus:

#### **Applikationen**

Unter die Kategorie „Applikationen“ fallen Werkzeuge, welche die Entwicklung von Funktionalitäten der Anwendung und insbesondere die Gestaltung der Benutzeroberfläche mit visuellen Mitteln ermöglichen. Nennenswerte Vertreter zur Erstellung von Websites sind in diesem Fall WordPress, mitsamt zahlreicher Plugins, oder neuere Plattformen wie Webflow. Aber auch Anbieter\*innen wie Adalo oder Glide für mobile Anwendungen zählen zu dieser Kategorie. Viele dieser Anwendungen nutzen Datenbanken zur Speicherung verschiedenster Daten. Durch den No-Code-Ansatz ist kein technisches Know-how zur Verwaltung der Datenbank notwendig, die meisten Anbieter\*innen bieten hierfür eine eigene

Oberfläche in Tabellenform an. Einfache Anwendungen lassen sich oft vollständig mit No-Code-Tools entwickeln. Sollte eine bestimmte Funktionalität nicht vom Basisumfang einer No-Code-Plattform abgedeckt sein, so ist es oft möglich, eigenen Code an der entsprechenden Stelle zu integrieren.

### **Integrationen**

Integrationen ermöglichen die Vernetzung unterschiedlicher Anwendungen durch den Einsatz von externen APIs (= Application Programming Interface) mit nur wenigen Klicks. Zu den größten Anbieter\*innen im No-Code-Bereich zählen Zapier und Integromat. Auf diesen Plattformen finden sich Softwareschnittstellen unterschiedlichster Anbieter\*innen, welche miteinander kombiniert werden können. Nutzer\*innen sind somit in der Lage, selbst auszuwählen welche Tools sie verwenden möchten und welche Workflows dabei entstehen. Mit solchen Integrationsplattformen lassen sich neben einfachen Workflows, wie etwa der Benachrichtigung bei einem bestimmten Ereignis, auch komplexere, mehrstufige und verzweigte Abläufe mit unterschiedlichen Tools erstellen.

### **Datenkonsolidierung**

Der Einsatz von No-Code-Tools dieser Kategorie ist sinnvoll, wenn bereits eine Vielzahl an Daten aus unterschiedlichen Schnittstellen angefallen ist. Diese Daten können beispielsweise aus einer eigens erstellten Anwendung stammen, wie etwa der Öffnungsrate von Marketing-Emails oder der Häufigkeit der Nutzung einer Applikation. Im Normalfall sind diese Daten durch den Einsatz mehrerer Lösungen stark zwischen diesen Plattformen verteilt. Mittels Datenkonsolidierungs-Tools lassen sich diese Daten an einer zentralen Stelle sammeln und weiterverarbeiten.

### **Software as a Service (SaaS)**

Diese Kategorie beinhaltet unterschiedliche einfache, sowie komplexe Anwendungen, welche einen bestimmten Anwendungsfall ermöglichen. Dabei sind diese Tools so konzipiert, dass sie schnell und einfach in eine bestehende Anwendung integriert werden können. Ein Beispiel von SaaS-Anwendungen, die heutzutage weit verbreitet sind, sind Payment-Dienste. Einen Zahlungsprozess selbst zu entwickeln, ist nicht nur sehr komplex, sondern auch sehr zeitaufwändig und fehleranfällig. Anbieter\*innen von Payment-Integrationen wie zum Beispiel Stripe stellen deswegen fertig einsetzbaren Code bereit, welcher problemlos in die eigene Anwendung eingebunden werden kann. Neben Payment-Service-Providern werden heutzutage immer mehr Anwendungen für Bereiche wie

Marketing und Sales, Kundenbetreuung, Ticketing-Systeme für Veranstaltungen, elektronische Formulare und ähnliches als SaaS-Dienstleistung angeboten.

## 5.2 Aufbau und Charakteristik

Sahay et al. (2020, S.3) zufolge bestehen Low-Code- / No-Code-Plattformen aus architektonischer Sicht aus den folgenden vier Hauptschichten:

### **Anwendungsschicht**

Die oberste Schicht (Anwendungsschicht) besteht aus der grafischen Oberfläche, mit der Nutzer\*innen direkt interagieren, um die Anwendung zu erstellen. Werkzeuge und Widgets, die zum Aufbau der Benutzeroberfläche eingesetzt werden, sind Teil dieser Schicht, ebenso wie Authentifizierungs- und Autorisierungsmechanismen. Durch die in dieser Schicht zur Verfügung gestellten Werkzeuge und Einstellungsmöglichkeiten, lässt sich auch das Verhalten der zu entwickelnden Anwendung steuern. Nutzer\*innen können beispielsweise festlegen, wie Daten aus externen Quellen (z.B. Tabellen, Kalender, etc.) oder in Cloud-Diensten gespeicherte Daten abgerufen und in der Anwendung weiterverarbeitet werden.

### **Dienstintegrationsschicht**

Um den soeben genannten Datenverkehr überhaupt zu ermöglichen, wird die Dienstintegrationsschicht genutzt, welche über entsprechende APIs und Authentifizierungsmechanismen eine Verbindung zu verschiedenen Diensten herstellt.

### **Datenintegrationsschicht**

Die Datenintegrationsschicht ermöglicht ein zentrales Verwalten der Daten, auch wenn diese aus unterschiedlichen Datenquellen stammen.

### **Bereitstellungsschicht**

Abhängig von der verwendeten Low-Code- / No-Code-Plattform kann die entwickelte Anwendung auf dedizierten Cloud-Infrastrukturen oder On-Premises (= Lokal auf eigener Server-Infrastruktur) bereitgestellt werden.

### 5.2.1 Bereitstellung in der Cloud

Die Verlagerung von immer mehr Anwendungen ins Web beziehungsweise die Bereitstellung von Anwendungen in der Cloud haben mit Sicherheit einen wesentlichen Beitrag geleistet, dass Low-Code- / No-Code-Plattformen heutzutage im Trend liegen. Denn viele dieser Plattformen werden selbst als Cloud-Lösung, in Form einer Plattform as a Service (kurz PaaS) angeboten. Für die Nutzer\*innen bedeutet das, dass der Betrieb der Plattform sowie der damit entwickelten Anwendung vollständig durch das Anbieterunternehmen erfolgt. Richardson und Rymer (2016, S.2f.) zufolge, hat die Nutzung von Low-Code- / No-Code-Plattformen als PaaS folgende Vorteile:

- Die Nutzung einer PaaS ist einfacher als die eigene Einrichtung, Konfiguration und Pflege aller notwendigen Komponenten für den Betrieb einer Anwendung, denn diese Einstellungen wurden vom Anbieterunternehmen der Cloud-Lösung bereits getroffen. Das beschleunigt zudem die Time-to-Market, da sich nicht mehr um die technischen Voraussetzungen gekümmert werden muss.
- Da bei der Verwendung einer PaaS auch die Infrastruktur des Anbieterunternehmens verwendet wird, ist eine Skalierung der entwickelten Anwendung, je nach gewähltem Paket, in den meisten Fällen problemlos möglich.
- PaaS werden oft mit unterschiedlichen Preismodellen angeboten, somit haben Nutzer\*innen die Möglichkeit, ein auf ihre Ansprüche abgestimmtes Paket zu wählen. Darüber hinaus entfallen die Kosten für eigene Hosting-Server, da diese bereits im Paket miteinberechnet sind.

Entgegen den Vorteilen der Nutzung einer PaaS ist jedoch zu berücksichtigen, dass durch die Nutzung eine starke Abhängigkeit zum Anbieterunternehmen besteht, was einen späteren Anbieter\*innenwechsel schwierig oder gar unmöglich machen könnte. Zudem bedarf es einem ausreichenden Vertrauen in das Anbieterunternehmen, zum Beispiel hinsichtlich der Einhaltung von Datenschutz, da oft keine Möglichkeit zur Einsicht durch die Nutzer\*innen der PaaS besteht.

## 5.3 Nutzen und Limitierungen

Luo et. al (2021) haben den Nutzen als auch die Limitierungen von Low-Code-Plattformen aus Sicht der Nutzer\*innen analysiert. Da am Markt eine Vielzahl an Low-Code-Plattformen verfügbar ist, fallen die erhobenen Erkenntnisse teilweise

sehr unterschiedlich aus. Die Ergebnisse dieser Untersuchung, hinsichtlich der Vorteile und Grenzen von Low-Code-Plattformen, lassen sich wie folgt zusammenfassen.

Low-Code-Plattformen ermöglichen es den Nutzer\*innen Anwendungen schnell und mit minimalen Aufwand zu erstellen. Durch eine Vielzahl von einsatzbereiten Elementen und Werkzeugen innerhalb der Low-Code-Plattform kann die Entwicklungszeit verkürzt werden, gleichzeitig führt dies aber in manchen Fällen auch zu einer Verlangsamung des Ladevorgangs der entwickelten Anwendung. Als weitere Nachteile von Low-Code-Plattformen werden der fehlende Zugang zum Quellcode, sowie die Bindung an ein bestimmtes Anbieterunternehmen genannt. Es konnte auch festgestellt werden, dass einige Vorteile den Einschränkungen von Low-Code-Plattformen widersprechen. Einige Nutzer\*innen behaupten, dass Low-Code-Plattformen im Vergleich zu Programmiersprachen einfacher zu lernen sind. Andere Nutzer\*innen argumentieren allerdings damit, dass auch Low-Code-Plattformen nicht einfach zu bedienen sind und eine hohe Lernkurve aufweisen. Dies lässt sich auf vereinzelt, komplexe Funktionalitäten mancher Low-Code-Plattformen zurückführen, welche mit einem gewissen Lernaufwand verbunden sind (Luo et al., 2021).

Als weiteren Vorteil gegenüber traditioneller Entwicklung werden geringer ausfallende IT-Kosten genannt, da keine teuren Entwickler\*innen eingestellt werden müssen. Manche Nutzer\*innen vertreten jedoch eine gegenteilige Ansicht und sind der Meinung, dass der Einsatz von Low-Code-Plattformen teuer ist. Grund dafür sei der hohe Preis, den einige Anbieter\*innen für den vollständigen Funktionsumfang ihrer Plattform verlangen, sowie die Abrechnung nach der Anzahl der Nutzer\*innen bei manchen Plattformen, was besonders teuer ausfällt, wenn im Team mit Low-Code-Plattformen gearbeitet wird (Luo et al., 2021).

Einerseits vertreten Nutzer\*innen die Ansicht, dass mit Low-Code-Plattformen qualitativ hochwertige, sichere, skalierbare und wartbare Anwendungen erstellt werden können. Auf der anderen Seite behaupten Nutzer\*innen jedoch, dass diese Anwendungen schwer zu ändern, zu warten und zu debuggen sind. Der Grund dafür könnte sein, dass es schwer überprüfbar ist, ob der von Low-Code-Plattformen generierte Code Fehler aufweist. Ebenso werden mit der Nutzung unterschiedlicher Plattformen auch verschiedene Erfahrungen gemacht. Einige Plattformen gelten für Nutzer\*Innen als perfekt anpassungsfähig, während bei anderen Plattformen über mangelnde Anpassungsmöglichkeiten geklagt wird. Eine Low-Code-Plattform mit begrenzter Flexibilität in Bezug auf Funktionalität und Design kann dazu führen, dass mehr Zeit für das Einpflegen von

## 5 No-Code-Plattformen

---

benutzerdefiniertem Code aufgewendet werden muss. Sobald solche Einschränkungen festgestellt werden, kann es zu höherer Komplexität, Kompromissen und Frustration während der Nutzung führen, denn wenn komplexe Funktionen immer noch händisch programmiert werden müssen, bedeutet dies, dass Low-Code-Plattformen weniger leistungsfähig als traditionelle Entwicklung sind (Luo et al., 2021).

# 6 Empirische Untersuchung

Dieses Kapitel beschreibt die Vorbereitung für die Umsetzung der Forschung, stellt die gewählten Forschungsmethoden vor und erläutert die Vorgehensweise der empirischen Untersuchung. Im Rahmen dieser Masterarbeit erfolgt ein Vergleich von vier ausgewählten No-Code-Plattformen sowie ein Experiment einer Webdesignumsetzung mit den zwei funktionsreichsten Plattformen. Ziel dieser Untersuchung ist es, die folgende Forschungsfrage samt Teilfrage zu beantworten:

Inwiefern eignen sich No-Code-Entwicklungsplattformen zur Umsetzung eines modernen, responsiven Webdesigns?

- Welche Einschränkungen treten während der Verwendung einer No-Code-Entwicklungsplattform bei der Umsetzung eines modernen, responsiven Webdesigns auf?

Aus den Forschungsfragen wurde die folgende Hypothese abgeleitet:

Wird eine No-Code-Plattform für die Umsetzung eines modernen, responsiven Webdesigns eingesetzt, so werden Einschränkungen auftreten, welche ausschließlich mit eigens geschriebenem Code behoben werden können.

## 6.1 Recherche

Vorangehend an die weitere Untersuchung wurde eine Online-Recherche zum Thema No-Code-Plattformen zur Erstellung von Websites durchgeführt. Diese Recherche wurde am 10. und 11. August 2022 durchgeführt. Dabei wurde die Suchmaschine Google Search genutzt und die Keywords „no code website builder“ und „no code platform for websites“ eingegeben. Als weitere Ressource wurden Softwarevergleichsplattformen mit den Kategorien „No-Code“ und „Website-Builder“ durchsucht. Ziel dieser Recherche war eine Auflistung der am Markt erhältlichen No-Code-Plattformen. Dabei wurde der Name der No-Code-Software, das Anbieterunternehmen und die offizielle Website protokolliert. Aus den Ergebnissen der Suchanfrage wurde die folgende Tabelle unterschiedlicher Anbieter von No-Code-Plattformen gebildet (siehe Tabelle 2).

Tabelle 2. Recherchierte No-Code-Plattformen

Softwarelösung	Anbieterunternehmen	Offizielle Website
Bubble	Bubble Group, Inc	<a href="https://bubble.io/">https://bubble.io/</a>
Carrd	Carrd, Inc	<a href="https://carrd.co/">https://carrd.co/</a>
Circa	Circa	<a href="https://circa.so/">https://circa.so/</a>
Divi	Elegant Themes	<a href="https://www.elegantthemes.com/">https://www.elegantthemes.com/</a>
Dorik	Dorik	<a href="https://dorik.com/">https://dorik.com/</a>
duda	duda	<a href="https://www.duda.co/">https://www.duda.co/</a>
Editor X	Wix.com, Inc	<a href="https://de.editorx.com/">https://de.editorx.com/</a>
Elementor	Elementor	<a href="https://elementor.com/">https://elementor.com/</a>
Framer	Framer B.V.	<a href="https://www.framer.com/">https://www.framer.com/</a>
Shopify	Shopify, Inc	<a href="https://www.shopify.de/">https://www.shopify.de/</a>
Squarespace	Squarespace, Inc	<a href="https://de.squarespace.com/">https://de.squarespace.com/</a>
Studio	Studio, Inc	<a href="https://studio.design/">https://studio.design/</a>
Typedream	Govest, Inc	<a href="https://typedream.com/">https://typedream.com/</a>
Unbounce	Unbounce	<a href="https://unbounce.com/">https://unbounce.com/</a>
Webflow	Webflow, Inc	<a href="https://webflow.com/">https://webflow.com/</a>
Weebly	Square, Inc	<a href="https://www.weebly.com/at">https://www.weebly.com/at</a>
Wix	Wix.com, Inc	<a href="https://de.wix.com/">https://de.wix.com/</a>
WordPress.com	Automatic	<a href="https://wordpress.com/de/">https://wordpress.com/de/</a>
Ycode	Ycode	<a href="https://www.ycode.com/">https://www.ycode.com/</a>
Zyro	Hostinger	<a href="https://zyro.com/de">https://zyro.com/de</a>

### Eingrenzung der Datenmenge

Diese Liste an No-Code-Tools, welche innerhalb einer Online-Recherche erhoben wurde, zeigt, dass bereits eine Vielzahl unterschiedlicher Anbieter\*innen einen Service oder ein Produkt in Form von No-Code zur Websiteerstellung am Markt anbieten. Dabei lässt sich feststellen, dass sich manche Anbieter\*innen auf einen bestimmten Einsatzzweck, beispielsweise E-Commerce, spezialisiert haben. Andere Anbieter\*innen bieten hingegen eine Vielzahl an Services an, wobei die No-Code-Entwicklung nur einen Teil ihrer Produktpalette ausmacht. Für die weitere Untersuchung in dieser Arbeit erfolgte eine Reduzierung der Datenmenge. Plattformen, welche vorgefertigte Templates als Grundlage voraussetzen, wurden aus der weiteren Auswahl entfernt. Auch wurden Systeme, welche sich in erster Linie auf die Erstellung von Web-Applikationen anstelle von Websites fokussieren von der weiteren Forschung ausgeschlossen. Ebenso fielen No-Code-Tools ausschließlich zur Erstellung von Webshops oder One-Page-Websites weg. Ein weiteres Ausschlusskriterium war der derzeitige Entwicklungsstand mancher Systeme. So wurde beispielsweise der Block-Editor von WordPress oder die No-Code- Plattform Circa aussortiert, da sich diese noch in Entwicklung befinden. Die soeben genannten Ausschlusskriterien resultierten in einer Reduzierung auf vier No-Code-Plattformen für den weiteren Vergleich, abgebildet in folgender Tabelle.

*Tabelle 3. Ausgewählte No-Code-Plattformen für den Vergleich*

<b>Softwarelösung</b>	<b>Anbieterunternehmen</b>	<b>Offizielle Website</b>
duda	duda	<a href="https://www.duda.co/">https://www.duda.co/</a>
Editor X	Wix.com, Inc	<a href="https://de.editorx.com/">https://de.editorx.com/</a>
Elementor	Elementor	<a href="https://elementor.com/">https://elementor.com/</a>
Webflow	Webflow, Inc	<a href="https://webflow.com/">https://webflow.com/</a>

Die ausgewählten No-Code-Plattformen erlauben eine komplette Websiterstellung von Grund auf und ähneln sich dabei auf den ersten Blick im Funktionsumfang. Duda, Editor X und Webflow werden als eigenständige Cloud-Lösung angeboten, Elementor hingegen wird als Website-Builder-Plugin für das Content-Management-System WordPress vertrieben.

## 6.2 Forschungsmethoden & Evaluation

Dieses Kapitel stellt die nachfolgend angewendeten Forschungsmethoden, die Dokumentenanalyse, die qualitative Inhaltsanalyse samt induktiver Kategorienbildung und die Evaluation vor.

### **Dokumentenanalyse**

Mittels Dokumentenanalyse lassen sich empirische Daten gewinnen und auswerten. Die genuine Dokumentenanalyse greift dabei auf bereits vorhandene Dokumente zurück, die unabhängig vom Forschungsprozess produziert wurden. Diese Dokumente können verbal beziehungsweise narrativ, visuell, audiovisuell oder multimedial sein. Der Zugriff auf bereits vorgefundene Dokumente hat den Vorteil, dass durch die Forschungstätigkeit keine Beeinflussung der Dokumente erfolgt. Als Nachteil zählen eventuell unbekannte Kontextbedingungen der Dokumentenproduktion, wie zum Beispiel ein fehlender Autor. Unter einem Dokument wird ein materieller oder virtueller Informationscontainer samt dessen Inhalte verstanden. Ein Dokument muss als Einheit wahrnehmbar, speicherbar oder archivierbar und wieder abrufbar sein. Besonders bei digitalen Dokumenten wird der Einsatz von computergestützten Archivierungs- und Analysetools empfohlen, um die Rohdaten aus dem Internet für die Dokumentenanalyse nutzbar zu machen. Für die nachfolgende Auswertung der Daten können sowohl qualitative als auch quantitative Analysemethoden eingesetzt werden (Döring & Bortz, 2016, S.533).

### **Qualitative Inhaltsanalyse**

Die qualitative Inhaltsanalyse eignet sich zur Bildung von Kategorien durch Techniken der Zusammenfassung. Bei der induktiven Kategorienbildung, welche in dieser Arbeit angewandt wird, werden die Kategorien aus dem Material durch Verallgemeinerung abgeleitet. Dieser Vorgang ermöglicht eine äußerst naturalistische, gegenstandsnahe Abbildung des Materials, ohne dass dabei Verzerrungen durch Vorannahmen der Forschenden oder des Forschenden entstehen. Zu Beginn muss ein Selektionskriterium bestimmt werden, das bestimmt, welches Material Ausgangspunkt für die Kategoriendefinition sein soll. Vom Thema abweichendes Material wird damit ausgeschlossen. Eine zuvor definierte Fragestellung der Analyse ist dabei richtungsweisend. Weiters gilt es, ein Abstraktionsniveau der zu bildenden Kategorien festzulegen. Anschließend wird das Material Zeile für Zeile durchgearbeitet und bei Erfüllung des Selektionskriteriums die erste Kategorie gebildet. Wird das Selektionskriterium

erneut erfüllt, so muss entschieden werden, ob die Textstelle unter die soeben gebildete Kategorie fällt oder eine neue Kategorie erstellt wird. Sobald das Material durchgearbeitet wurde, erfolgt eine Revision des Kategoriensystems. Ergebnis dieser Methode ist ein Kategoriensystem, welches anhand konkreter Textpassagen ein bestimmtes Thema beschreibt. Im Anschluss kann das Kategoriensystem interpretiert werden, entweder durch induktiver oder deduktiver Bildung von Hauptkategorien oder mittels quantitativer Analyse, beispielsweise zu den Häufigkeiten der Kategorien (Mayring, 2015, S.85f.). Im Laufe dieser Arbeit werden Hauptkategorien samt Unterkategorien gebildet, welche im Anschluss durch eine Evaluation der No-Code-Plattformen bewertet werden.

### **Evaluation**

Die wissenschaftliche Evaluation nutzt sozialwissenschaftliche Methoden, um einen Evaluationsgegenstand (z.B. ein Produkt oder eine Maßnahme) zu bewerten. Meist werden die grundlegenden Daten für eine Evaluation neu erhoben, es ist aber auch möglich auf bereits systematisch gesammeltes Datenmaterial zurückzugreifen. Bei der Durchführung einer Evaluation sind relevante Anspruchsgruppen, Evaluationskriterien sowie Bewertungsstandards zu berücksichtigen. Die im Zuge einer Evaluation erarbeitete Bewertung soll unterschiedliche praxisbezogene Evaluationsfunktionen wie beispielsweise die Erkenntnisfunktion oder die Optimierungsfunktion erfüllen. Mit dem Evaluationsergebnis können durch die Evaluierende oder den Evaluierenden Aussagen hinsichtlich Optimierung, Entscheidungen für oder gegen den Einsatz oder Praxisempfehlungen getroffen werden. Evaluationsstudien können in unterschiedliche Typen eingeteilt werden. Im Laufe dieser Arbeit erfolgt eine Selbstevaluation. Bei dieser Art der Evaluation erfolgt der gesamte Prozess durch die Forschende oder den Forschenden selbst. Der Vorteil einer Selbstevaluation liegt im besonders engen Praxisbezug und der dadurch hohen praktischen Verwertbarkeit der Ergebnisse. Dennoch besteht bei der Selbstevaluierung die Gefahr, gewollt oder ungewollt verzerrte oder geschönte Ergebnisse zu produzieren. Mittels einer Metaevaluation, bei der die eigentliche Evaluation bewertet und kritisch betrachtet wird, lässt sich dies jedoch regulieren (Döring & Bortz, 2016, S.977f.).

## 6.3 Forschungsdesign

In diesem Kapitel wird die weitere Herangehensweise der empirischen Untersuchung dokumentiert. Im Laufe dieser Arbeit wurden zwei unterschiedliche Ansätze gewählt, um die Forschungsfragen zu beantworten. Zum einen wurde ein Vergleich von vier ausgewählten No-Code-Plattformen durch Ableitung von Kategorien ausgehend von einer Dokumentenanalyse durchgeführt. Zum anderen erfolgte das Experiment einer Webdesignumsetzung mit den zwei funktionsreichsten Plattformen. Das Experiment dient als Ergänzung des Vergleichs und gibt Auskunft über die Umsetzbarkeit eines vorgegebenen Webdesigns mittels No-Code-Tools sowie über etwaige Einschränkungen dieser Tools. Darüber hinaus wurden auch die Umsetzungsdauer sowie wie die Performance der erstellten Websites gemessen.

### **Vorgehensweise des Vergleichs auf Basis der Kategorienbildung durch die Dokumentenanalyse**

Als Ausgangsbasis des Vergleichs wurde eine Dokumentenanalyse durchgeführt. Die Dokumente wurden in Form von Webseiten, mitsamt deren Inhalte, innerhalb der offiziellen Websites der ausgewählten No-Code-Plattformen, welche öffentlich aufrufbar sind, gesammelt. Im Falle des Page-Builder-Plugins Elementor wurde zusätzlich zur offiziellen Website, die Beschreibung im Plugin-Verzeichnis auf WordPress.org analysiert. Im Zuge der Dokumentenanalyse wurden neben dem Datum des Aufrufs der Websites weiters die Quelle, der Zweck und der Inhalt der Dokumente protokolliert. Die protokollierten Daten der Dokumentenanalyse sind im Anhang dieser Masterarbeit im Abschnitt A „Ergebnisse Dokumentenanalyse“ (Seite 107) ersichtlich.

Nach Durchsicht und Analyse der Dokumente wurden die Kategorien für die spätere Bewertung, nach dem Schema der qualitativen Inhaltsanalyse mitsamt induktiver Kategorienbildung nach Mayring erstellt. Untersuchungsgegenstand der Inhaltsanalyse waren die in der Dokumentenanalyse erhobenen Webseiten. Die Selektionskriterien wurden wie folgt definiert:

- Design-Funktionen
- CMS-Funktionen
- Templates und Komponenten
- SEO
- Integrationen
- Sonstige Funktionen

## 6 Empirische Untersuchung

Funktionen, welche das Marketing einer Website oder das Webhosting betreffen, wurden zur Bildung der Kategorien nicht weiter beachtet. Es wurde ein besonders niedriges Abstraktionsniveau gewählt, was bedeutet, dass ausschließlich explizit angeführte Funktionalitäten der No-Code-Plattformen in den Kategorienkatalog aufgenommen wurden.

Zur Unterstützung bei der Kategorienbildung wurde das Programm MAXQDA 2022 (<https://www.maxqda.de/>) verwendet. MAXQDA ist eine Software zur computergestützten Daten- und Textanalyse im Rahmen von wissenschaftlichen Projekten und Studien, bei denen Interviews, Texte und Medien wie Bild-, Audio- oder Videodateien inhaltlich analysiert werden. Die dazugehörige Browser-Erweiterung erleichtert zudem den Import von Websites als Dokument.

Die Software wurde in Form einer 14-tägigen Testphase heruntergeladen und installiert. Dabei war die Angabe einer E-Mail-Adresse und eines Benutzernamens notwendig. Für jede analysierte No-Code-Plattform wurde ein Ordner erstellt, welcher die jeweiligen Dokumente enthält. Diese Dokumente wurden mittels der Google Chrome Erweiterung von MAXQDA in das Programm importiert. Die Erstellung der Kategorien erfolgte über die Codierungsfunktion des Programms, welche in der folgenden Abbildung dargestellt ist.

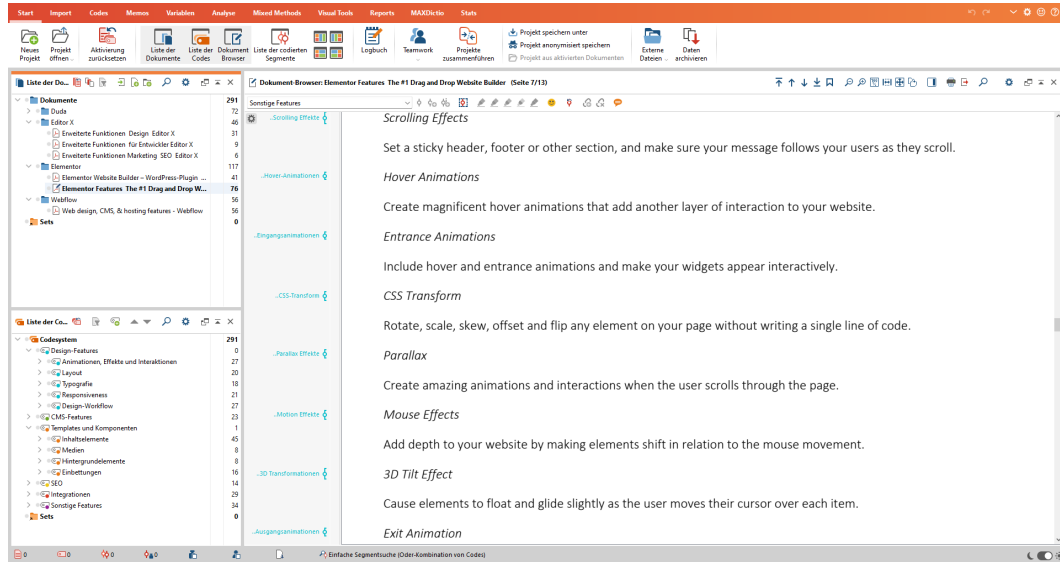


Abbildung 9. Kategorienbildung mit MAXQDA 2022 (eigener Screenshot)

Nach der Erstellung der Kategorien wurden diese über die Exportfunktion als Microsoft Excel-Datei exportiert. In weiterer Folge wurde die Evaluierung der No-

Code-Plattformen durchgeführt. Die exportierten Kategorien dienten hierbei als Bewertungskriterien. Evaluationsgegenstand waren die zu untersuchenden No-Code-Plattformen. Ziel der Evaluation war es herauszufinden, welche der Plattformen, die zuvor definierten Kriterien (Design-Funktionen, CMS-Funktionen, etc.) unterstützen. Zur Evaluation der Plattformen wurde ein Bewertungsschema erstellt, welches in der folgenden Tabelle dargelegt wird.

*Tabelle 4. Bewertungsschema der Evaluation*

<b>Punkte</b>	<b>Beschreibung</b>
<b>2</b>	Das Kriterium kann mit der No-Code-Plattform vollständig, ohne den Einsatz von eigenem Code umgesetzt werden.
<b>1</b>	Das Kriterium kann mit der No-Code-Plattform nur teilweise umgesetzt werden oder es bestehen Einschränkungen, welche ausschließlich mit eigenem Code behoben werden können.
<b>0</b>	Das Kriterium kann mit der No-Code-Plattform nicht umgesetzt werden und erfordert eigenen Code.

Gemäß den Regeln der Selbstevaluation wurde die Evaluierung vom Forschenden selbst durchgeführt. Die zu bewertenden Funktionen wurden vom Forschenden mit der jeweiligen No-Code-Plattform getestet. Damit eine gewisse Expertise vorliegt, wurde sich im Vorhinein mithilfe von Video-Tutorials und offiziellen Dokumentationen in die Plattformen eingearbeitet. Die erarbeiteten Ergebnisse wurden anschließend als Diagramme ausgegeben und interpretiert.

### **Vorgehensweise der Webdesignumsetzung auf Basis der Kategorienbildung durch die Websiteanalyse**

Die Erkenntnisse aus dem Vergleich werden durch das Experiment der Webdesignumsetzung mittels No-Code-Plattformen ergänzt. Das Experiment soll zeigen, wie detailgetreu die Umsetzung eines Webdesigns ohne den Einsatz von eigenem Code mit No-Code-Plattformen möglich ist. Zudem soll festgestellt werden, in welchen Bereichen mögliche Einschränkungen auftreten, die durch benutzerdefinierten Code behoben werden können. Da die No-Code-Plattformen Elementor und Webflow im Vergleich mehr Punkte als Duda und Editor X erzielen konnten, und dementsprechend über mehr Funktionen verfügen, wurden diese für die Webdesignumsetzung im Laufe des Experiments ausgewählt. Es ist davon auszugehen, mit diesen Tools ein detailgetreueres Ergebnis erzielen zu können.

## 6 Empirische Untersuchung

Als nachzubildendes Design wurde die Website der Abteilung „Forschung und Wissenstransfer“ der Fachhochschule St. Pölten <https://research.fhstp.ac.at/> gewählt. Diese Website verfügt über ein responsives, mehrspaltiges Layout, statische als auch dynamische Inhalte sowie Hover-Effekte und Interaktionsmöglichkeiten. Im Zuge des Experiments soll die Startseite der Website mit den visuellen Werkzeugen der No-Code-Plattformen nachgebaut werden. Ziel ist es, das Design der originalen Website auf unterschiedlichen Breakpoints (Desktop 1920px, Tablet 768px und Mobile 376px) so ident wie möglich nachzubilden. Die folgende Abbildung zeigt die auf der Website vorhandenen und nachzubildenden Elemente.

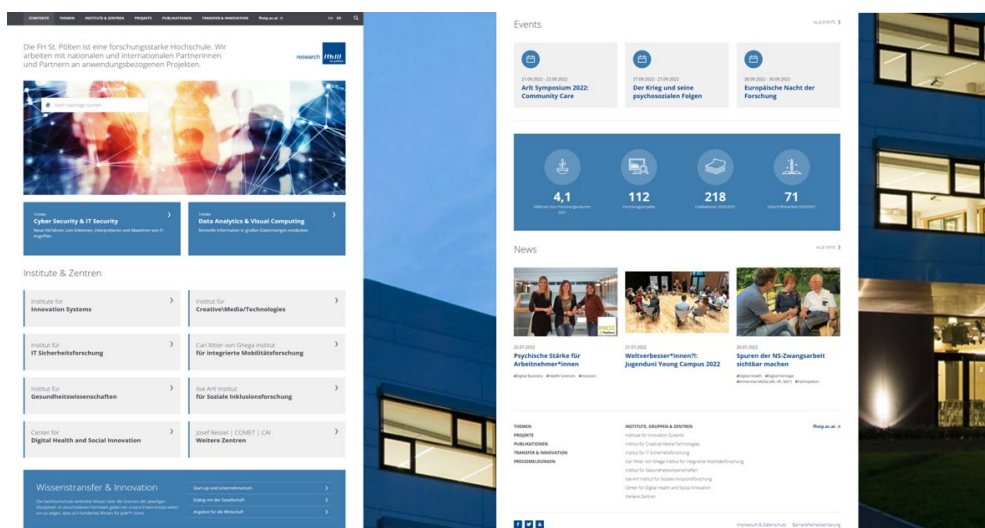


Abbildung 10. Nachzubildende Elemente der Website (eigener Screenshot)

Zu Beginn erfolgte eine Analyse der Website durch Untersuchen der CSS-Datei im Browser. Anschließend wurden drei Versionen der Website (Desktop, Tablet, Mobile) per Screenshot-Funktion des Browsers Google Chrome als png-Datei heruntergeladen und gespeichert, um die unterschiedlichen Darstellungen mit der Grafiksoftware Adobe XD weiter zu analysieren. Die CSS-Eigenschaften der nachzubildenden Elemente wurden in einer Tabelle festgehalten (siehe Tabelle 6, Seite 83). Als Ausgangsbasis für die Kategorienbildung diente die in Adobe XD angelegte Designvorlage mitsamt einer Beschreibung der einzelnen Elemente der Website. Wie zuvor beim Vergleich wurden Hauptkategorien für die weitere Evaluierung gebildet. Für eine umfassendere Bewertung über das Design der erstellten Websites hinaus, wurden die designspezifischen Kriterien um die Dauer der Umsetzung, sowie die Qualität der Websites ergänzt, welche mit dem Open-Source-Tool Google Lighthouse gemessen wurden.

Anschließend folgte die Umsetzung des Designs mit den No-Code-Plattformen. Da die Plattform Elementor als Plugin für das Content-Management-System WordPress vorliegt, wurde die zum Zeitpunkt der Umsetzung aktuellste Version von WordPress (6.0.2) auf der lokalen Entwicklungsumgebung installiert. Hierfür wurde die Software Local (<https://localwp.com/>) genutzt, da diese die Server-Konfiguration, sowie den Installationsvorgang beschleunigt. In weiterer Folge wurde das Plugin Elementor, in der damalig aktuellen Version (3.7.4) installiert und aktiviert. Um den vollen Funktionsumfang von Elementor nutzen zu können, ist ein kostenpflichtiges Upgrade auf die Pro-Variante notwendig. Im Rahmen dieser Arbeit wurde der „Essential Plan“ (<https://elementor.com/pricing-plugin/>) gewählt, wofür die Erstellung eines Accounts notwendig war, die Pro-Version (3.7.4) des Plugins heruntergeladen und ebenfalls in WordPress aktiviert. WordPress-Websites basieren auf Themes und mit der Installation von WordPress werden bereits einige Themes mitgeliefert. Da Komplikationen zwischen diesen Themes und dem Elementor-Plugin jedoch nicht ausgeschlossen werden können, wurde das hauseigene Theme „Hello Elementor“ (<https://wordpress.org/themes/hello-elementor/>) vom gleichnamigen Anbieterunternehmen Elementor installiert und aktiviert. Dieses Theme enthält keine vorgefertigten Styles und bietet eine weiße Leinwand zur Websiteerstellung von Grund auf. Weitere Änderungen an WordPress oder Plugin-Installationen wurden nicht vorgenommen. Die Website wurde gänzlich mit dem Elementor-Editor erstellt und befüllt.

Im Falle von Webflow ist die Einrichtung einer lokalen Entwicklungsumgebung nicht notwendig, lediglich die Erstellung eines Webflow-Accounts. Die Websiteerstellung erfolgt ausschließlich im Webbrowser. Für den vollständigen Funktionsumfang von Webflow ist ein Upgrade auf einen kostenpflichtigen „Site Plan“ erforderlich. Im Zuge dieser Arbeit wurde ein Webflow-Account erstellt, der Site Plan „CMS“ (<https://webflow.com/pricing>) gewählt und anschließend ein neues Projekt im Webflow-Dashboard angelegt. Die vollständige Erstellung der Website erfolgte über die „Designer“ genannte Benutzeroberfläche von Webflow.

Als nächstes erfolgte die Evaluierung anhand der Kriterien und des Bewertungsschemas (siehe Tabelle 4, Seite 60). Hierfür wurde die Evaluationsfrage „Zu wie viel Prozent können die eruierten Kriterien durch die No-Code-Plattform umgesetzt werden?“ definiert. Evaluationsgegenstand waren die ausgewählten No-Code-Plattformen. Aus den Ergebnissen wurden im Anschluss an die Evaluierung die prozentuellen Werte der Hauptkategorien in Diagrammen dargestellt und interpretiert.

# 7 Ergebnisse

In diesem Kapitel werden zuerst die Ergebnisse des Vergleichs und anschließend die Ergebnisse der Webdesignumsetzung präsentiert.

## 7.1 Ergebnisse des Vergleichs

Auf den folgenden Seiten werden der Kriterienkatalog sowie die Ergebnisse resultierend aus dem Vergleich der ausgewählten No-Code-Plattformen (siehe Tabelle 3, Seite 55) dargelegt.

### 7.1.1 Kriterienkatalog

Um den Vergleich durchführen zu können, wurden durch eine Dokumentenanalyse und eine qualitative Inhaltsanalyse 130 Kriterien zur Bewertung der No-Code-Plattformen definiert. Damit die Plattformen vollständig verglichen werden können, wurde der Funktionsumfang umfassend betrachtet. Die analysierten Funktionen widerspiegeln sich in den Kategorien des Kriterienkatalogs. Da einige Hauptkategorien eine große Anzahl an Funktionen enthielten, wurden diese zur besseren Übersicht nochmals in Unterkategorien unterteilt. Der vollständige Kriterienkatalog enthält sechs Hauptkategorien und neun Unterkategorien, welche alle einer Bewertung, ausgeführt durch den Forschenden, unterzogen wurden.

Tabelle 5. Kriterienkatalog des Vergleichs

■ Hauptkategorie    □ Unterkategorie

Kategorie	Beschreibung
<b>Design-Funktionen</b>	
<i>Animationen, Effekte und Interaktionen</i>	
Motion-Effekte	Auf Mausbewegung reagierende Bewegungseffekte
Scrolling-Effekte	Auf Scrolling reagierende Effekte
Parallax-Effekte	Parallax-Effekt beim Scrollen
Masken	Erstellung von Masken auf Basis verschiedener SVG-Formen

## 7 Ergebnisse

Blend-Modes	Hinzufügen von CSS Blend-Modes
Farbüberlagerungen	Farbüberlagerungen für Bilder und Videos
Animierte Überschriften	Verschiedene animierte Effekte für Überschriften
Box-Schatten	Hinzufügen von Box-Schatten auf verschiedene Elementen
CSS-Filter	Verschiedene CSS-Filter (Graustufen, Blur, etc.) für Elemente
CSS-Transform	CSS-Transformationen von Elementen
3D Transformationen	3D CSS-Transformationen
Hover-Animationen	Auf Hover reagierende Effekte
Eingangsanimationen	Eingangsanimationen von Elementen
Ausgangsanimationen	Ausgangsanimationen von Elementen
Seitenübergangsanimationen	Übergangseffekte beim Wechseln von Webseiten
Interaktionen und Trigger	Visueller Interaktionsbuilder
<i>Layout</i>	
CSS-Grid	Layoutgestaltung nach den Regeln von CSS-Grid
CSS-Flexbox	Layoutgestaltung nach den Regeln von CSS-Flexbox
Breitenanpassung	Anpassung der Breite von Elementen
Höhenanpassung	Anpassung der Höhe von Elementen
Unterschiedliche Maßeinheiten	Verschiedene Maßeinheiten (PX, %, VW, VH)
Margin & Padding	Anpassen des Innen- und Außenabstands von Elementen
Inline-Ausrichtung	Ausrichtung von Elementen nebeneinander
Verschachtelung	Verschachtelung von Containern und Elementen
Absolute Positionierung	Absolute Positionierung von Elementen
Fixe Positionierung	Fixe Positionierung von Elementen
Z-Index	Verschieben von Elementen auf der Z-Achse
<i>Typografie</i>	
Schriftart	Auswahl der Schriftfamilie

## 7 Ergebnisse

Schriftgröße	Anpassen der Schriftgröße
Schriftgewicht	Anpassen des Schriftgewichts
Schriftdekoration	Anpassen der Schriftdekoration (Unterstrich, Durchstrich, etc.)
Schrifttransformation	Transformation der Schrift (Kapitälchen, Großbuchstaben, etc.)
Zeilenhöhe	Anpassen der Zeilenhöhe
Zeichenabstand	Anpassen des Zeichenabstands
Initialen	Einstellen von Initialen bei Texten
Text Stroke	Anpassen der Strichstärke von Schriften
Textschatten	Hinzufügen von Textschatten
Webfonts	Einbindungsmöglichkeit von Webfonts
Benutzerdefinierte Schriftarten	Hinzufügen von benutzerdefinierten Schriften
<i>Responsive</i>	
Responsive Bilder	Skalieren und Optimieren von Bildern nach Bildschirmgröße
Responsives Layout	Breakpoints für unterschiedliche Bildschirmgrößen
Responsive Einstellungsmöglichkeiten	Separate Einstellungsmöglichkeiten für mobile Geräte
Benutzerdefinierte Breakpoints	Erstellung von eigenen Umbruchpunkten des Layouts
Ein- und Ausblenden von Elementen	Anzeigen oder Verstecken von Elementen je nach Breakpoint
<i>Design-Workflow</i>	
Drag & Drop Editor	Hinzufügen von Elemente per Drag & Drop
Navigator	Listenansicht der auf der Website platzierten Elemente
Copy-Paste	Kopieren und Einfügen von Elementen & Stilen
Tastaturkürzel	Tastaturkürzel für gängige Funktionen
Wiederverwendbare Elemente	Globale, wiederverwendbare Elemente
Globale Farben	Globale, wiederverwendbare Farben
Globale Schriftarten	Globale, wiederverwendbare Schriftarten
Icon-Bibliothek	Auswahl von vorgefertigten Icons

## 7 Ergebnisse

Vorgefertigte Layouts	Auswahl aus vorgefertigten Layoutcontainern
Vorgefertigte Websites	Website-Designvorlagen
Vorgefertigte Abschnitte	Designvorlagen von Abschnitten und Elemente
<b>CMS-Funktionen</b>	
Medienverwaltung	Upload und Verwaltung diverser Medien
Formulareinträge	Speicherung von Formulareinträgen in der Datenbank
Inhalts-Feldtypen	Unterschiedliche Feldtypen für Inhalte
Dynamische Inhalte	Einbindung von dynamischen Inhalte auf der Website
Dynamische Inhaltstemplates	Automatisch generierte Templates für Inhaltsbereiche
Blog-System	Dynamische Blogbeiträge (Paginierung, verwandte Beiträge, etc.)
CSV-Import	Datei-Import im CSV-Format
CSV-Export	Exportmöglichkeit von Inhalten im CSV-Format
Dynamisches Filtern und Sortieren	Filter und Sortierungsoptionen für dynamische Inhalte
<b>Elemente</b>	
<i>Inhaltselemente</i>	
Überschriften	Erstellung und Anpassung von Überschriften
Text-Editor	Erstellung und Anpassung von Texten
Buttons	Erstellung und Anpassung von Buttons
Formulare	Erstellung und Anpassung von Formularen und Formularfeldern
Akkordeons	Erstellung und Anpassung von Akkordeons
Tabs	Erstellung und Anpassung von Tabs
Slider	Erstellung und Anpassung von Slidern für Bilder oder Inhalte
Listen	Erstellung und Anpassung von Listen
Lightbox & Bildergalerien	Erstellung und Anpassung von Lightboxen und Galerien
Navigationsmenü	Erstellung und Anpassung von responsiven Navigationsmenüs
Ankerlinks	Erstellung und Anpassung von Ankerlinks

## 7 Ergebnisse

---

Abstandshalter	Erstellung und Anpassung von Abstandshaltern
Trenner	Erstellung und Anpassung von Trennern
Shape-Divider	Erstellung und Anpassung von Trennern aus SVG-Formen
Text-Pfad	Erstellung und Anpassung von Text-Pfaden
Bild-Box	Erstellung und Anpassung von Bild-Boxen
Icon-Box	Erstellung und Anpassung von Icon-Boxen
Flip Box	Erstellung und Anpassung einer Flip Box
Zitatblock	Erstellung und Anpassung eines Zitatblocks
Hinweismeldungen	Erstellung und Anpassung von Hinweismeldungen
Counter	Erstellung und Anpassung von Countern
Countdown	Erstellung und Anpassung eines Countdowns
Fortschrittsbalken	Erstellung und Anpassung von Fortschrittsbalken
Testimonials	Erstellung und Anpassung von Testimonials
Sterne-Bewertungen	Erstellung und Anpassung von Sterne-Bewertungen
Preistabelle	Erstellung und Anpassung von Preistabellen
Inhaltsverzeichnis	Erstellung und Anpassung eines Inhaltsverzeichnisses
Lottie-Animation	Erstellung und Anpassung einer Lottie-Animation
<i>Medien</i>	
SVG-Icons	Möglichkeit des Uploads eigener Icons als SVG-Datei
Audio	Einbettung eines Audio-Players
Videos	Erstellung und Anpassung von Video-Elementen
Bilder	Erstellung und Anpassung von Bildern
<i>Hintergrundelemente</i>	
Hintergrundbilder	Erstellung und Anpassung von Hintergrundbildern
Hintergrundvideos	Erstellung und Anpassung von Hintergrundvideos
Hintergrundverläufe	Erstellung und Anpassung von Hintergrundverläufen

## 7 Ergebnisse

Hintergrund-Slideshows	Erstellung und Anpassung von Hintergrund-Slideshows
Hintergrund-Überlagerung	Erstellung und Anpassung von Hintergrund-Überlagerungen
<i>Einbettungen</i>	
HTML	Erstellung von HTML-Einbettungsfeldern
Social-Icons & Inhalte	Erstellung und Anpassung von Social-Icons
Google Maps	Erstellung und Anpassung von interaktiven Google Maps
YouTube-Video	Einbettung eines YouTube-Videos
Vimeo-Video	Einbettung eines Vimeo-Videos
Dailymotion-Video	Einbettung eines Dailymotion-Videos
SoundCloud-Player	Einbettung eines SoundCloud-Players
Mixcloud-Player	Einbettung eines Mixcloud-Players
<b>Integrationen</b>	
Integrationen	Anzahl der auf der offiziellen Website gelisteten Integrationen
<b>SEO &amp; Accessibility</b>	
Anpassbare HTML-Tags	Anpassbare HTML-Tags für unterschiedliche Elemente
Dynamische SEO-Tags und Open Graph Einstellungen	Automatische Erstellung von SEO-Tags und Open Graph Inhalten
Alt-Tags für Bilder	Hinzufügen von alternativen Beschreibungen für Bilder
XML-Sitemap	Autogenerierte XML-Sitemap
Meta-Titel und Meta-Beschreibung	Hinzufügen von Meta-Titel und -Beschreibungen
Open Graph Einstellungen	Hinzufügen von Open Graph Inhalten
<b>sonstige Funktionen</b>	
Popup-Builder	Erstellen und Anpassen von Popups
Widget-Builder	Erstellen von eigenen Elementen
Benutzerrollen	Benutzerrollen mit unterschiedlichen Rechten
Code-Export	Exportmöglichkeit des Website-Quellcodes
Benutzerdefinierter Code	Hinzufügen von benutzerdefiniertem Code

## 7 Ergebnisse

---

Accessibility-Audit	Funktion zum Erkennen von Problemen bei der Barrierefreiheit
API	Programmierschnittstelle
404 Seite	Anpassbare 404-Seiten
Favicon	Hinzufügen eines Website-Icons
Passwortschutz	Passwortgeschützte Seiten
Backups und Versionierung	Website-Backups und Revisionen
Wartungsmodus	Coming Soon Seite während der Websiteerstellung
Suchfunktion	Einbindung einer Suchfunktion in die Website
Spamschutz	Spamschutz via ReCAPTCHA
Website-Statistiken	Einbindung von Google Analytics

### 7.1.2 Ergebnisse der Evaluation

Nachfolgend werden die Ergebnisse, der innerhalb der sechs definierten Hauptkategorien

- Design-Funktionen
- CMS-Funktionen
- Elemente
- Integrationen
- SEO & Accessibility
- sonstige Funktionen

bewerteten Kriterien, präsentiert. Die Ergebnisse werden in Form von Balkendiagrammen dargestellt, welche die zu bewerteten Kriterien sowie deren Umsetzbarkeit ohne eigenen Code zeigen. Aufgrund mangelnder Übersicht wird in den Diagrammen auf die Bezeichnung der verglichenen No-Code-Plattformen verzichtet. Die Diagramme zeigen, ob ein Kriterium ohne eigenen Code möglich, teilweise möglich oder nicht möglich ist. Daraus lässt sich feststellen, welche der Funktionen vermehrt und welche weniger oft in den ausgewählten No-Code-Plattformen verfügbar sind. Eine detaillierte Tabelle mitsamt Bezeichnung der No-Code-Plattform und Bewertung der Umsetzbarkeit der einzelnen Funktionen ist im Anhang im Abschnitt B „Evaluierungsergebnisse Vergleich“ (Seite 115) zu finden.

## Animationen, Effekte und Interaktionen

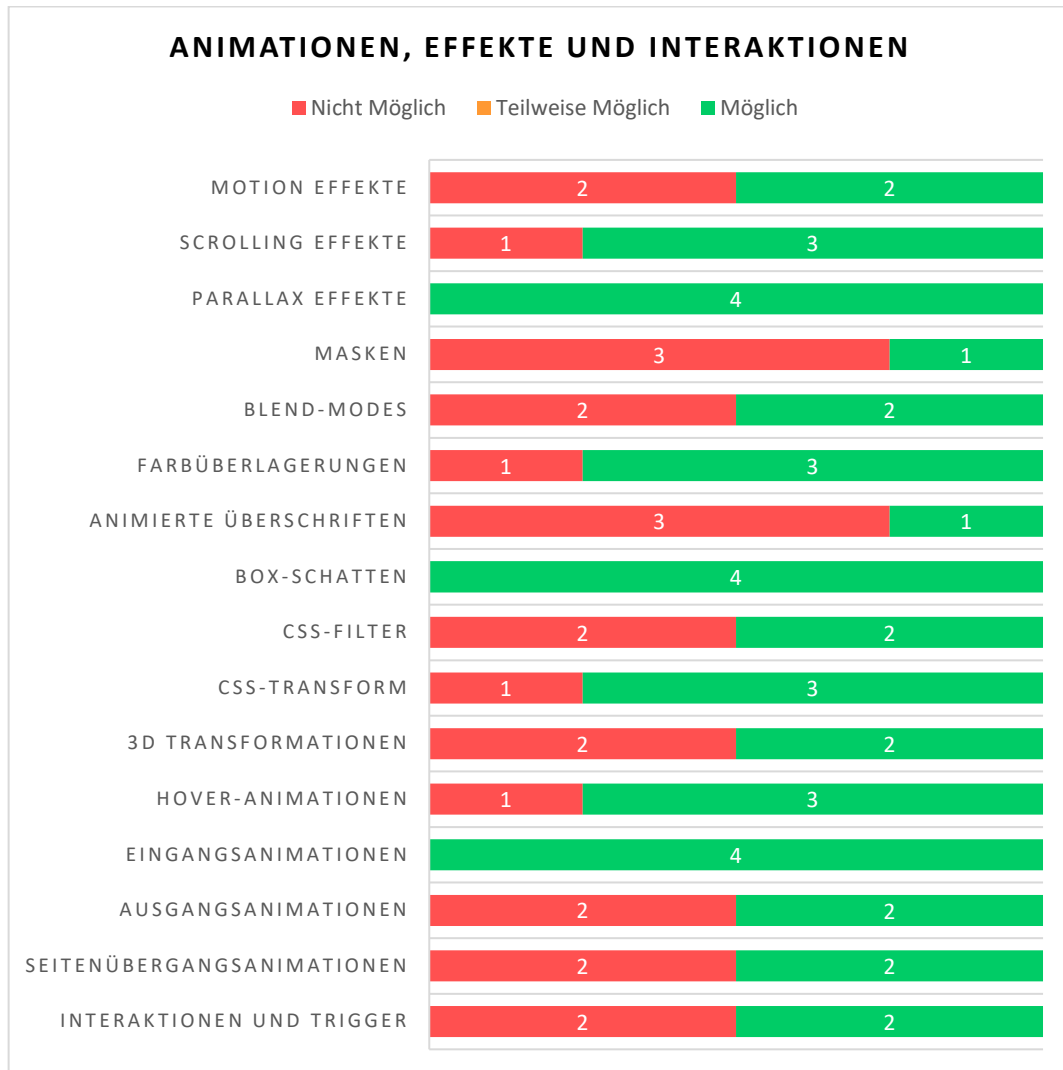


Abbildung 11. Ergebnisse Kategorie Animationen/Effekte (eigene Darstellung)

Die Ergebnisse der Kategorie Animationen, Effekte und Interaktionen in Abbildung 11 zeigen, dass alle untersuchten Plattformen Parallax-Effekte, Box-Schatten sowie Eingangsanimationen von Elementen bereitstellen. Auf Scrolling basierende Effekte, Farbüberlagerungen, CSS-Transformationen und Animationen beim Hover auf Elemente werden von drei Systemen unterstützt. Motion-Effekte, Blend-Modes, CSS-Filter sowie 3D-Transformationen von Elementen lassen sich bei zwei Lösungen konfigurieren. Ebenfalls unterstützen zwei Anbieter\*innen Ausgangsanimationen von Elementen, Übergangsanimationen beim Wechsel von Webseiten sowie die Erstellung von komplexeren Interaktionen auf Basis

## 7 Ergebnisse

unterschiedlicher Trigger wie beispielsweise Page-Load oder Klick auf das Element. Lediglich eine No-Code-Plattform bietet SVG-Masken und animierte Überschriften an.

### Layout

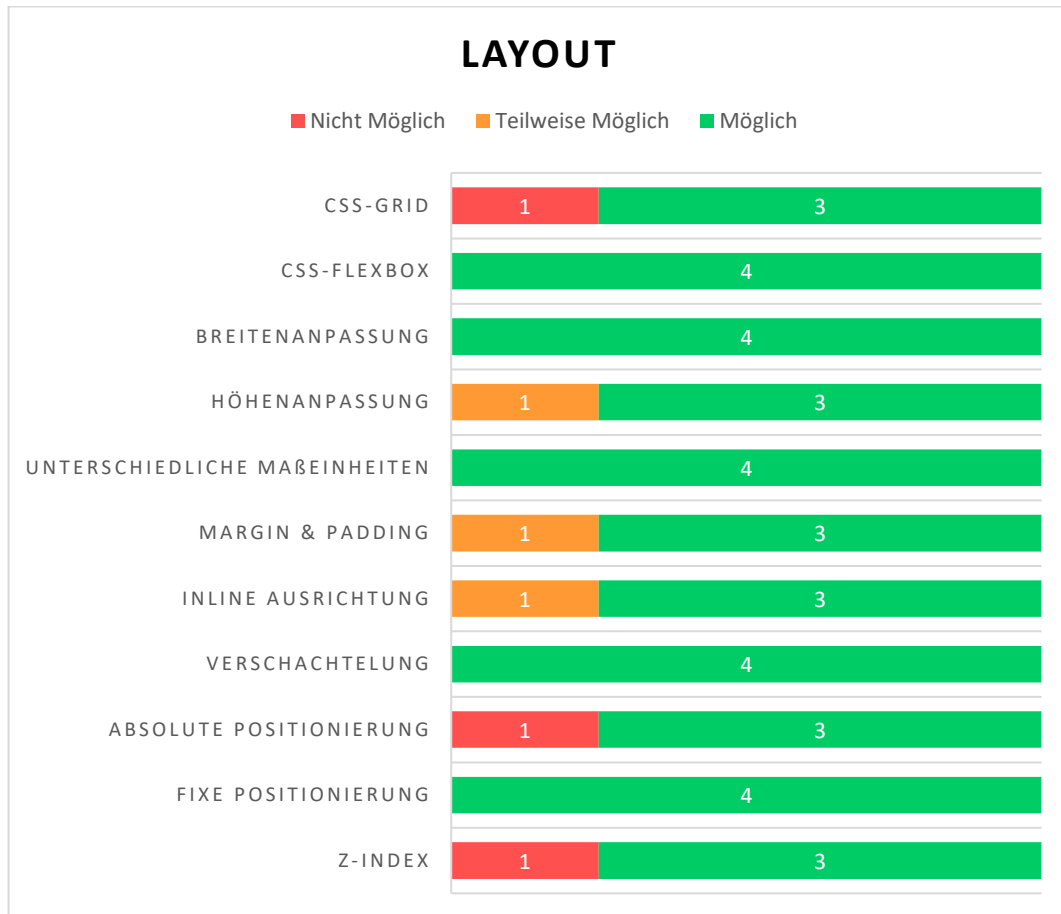


Abbildung 12. Ergebnisse Kategorie Layout (eigene Darstellung)

Die untersuchten Plattformen weisen Abbildung 12 zufolge verschiedene Layoutoptionen auf. Alle Systeme erlauben die Erstellung von Layouts nach den Regeln von CSS-Flexbox. Auch die Anpassung der Breite, die Verwendung verschiedener Einheiten (PX, %, VW, VH) und das Verschachteln von Elementen ineinander ist bei allen Plattformen möglich, ebenso wie die fixe Positionierung von Elementen. Drei Lösungen bieten darüber hinaus auf CSS-Grid basierte Layouts, absolute Positionierung von Elementen und das Einstellen des Z-Indexes an. Die Höhenanpassung von Elementen, das Einstellen des Innen- (Padding) und Außenabstands (Margin) sowie das Ausrichten mehrerer Elemente nebeneinander

(Inline) wird von einer No-Code-Plattform nur teilweise, von allen anderen Plattformen vollständig unterstützt.

### Typografie

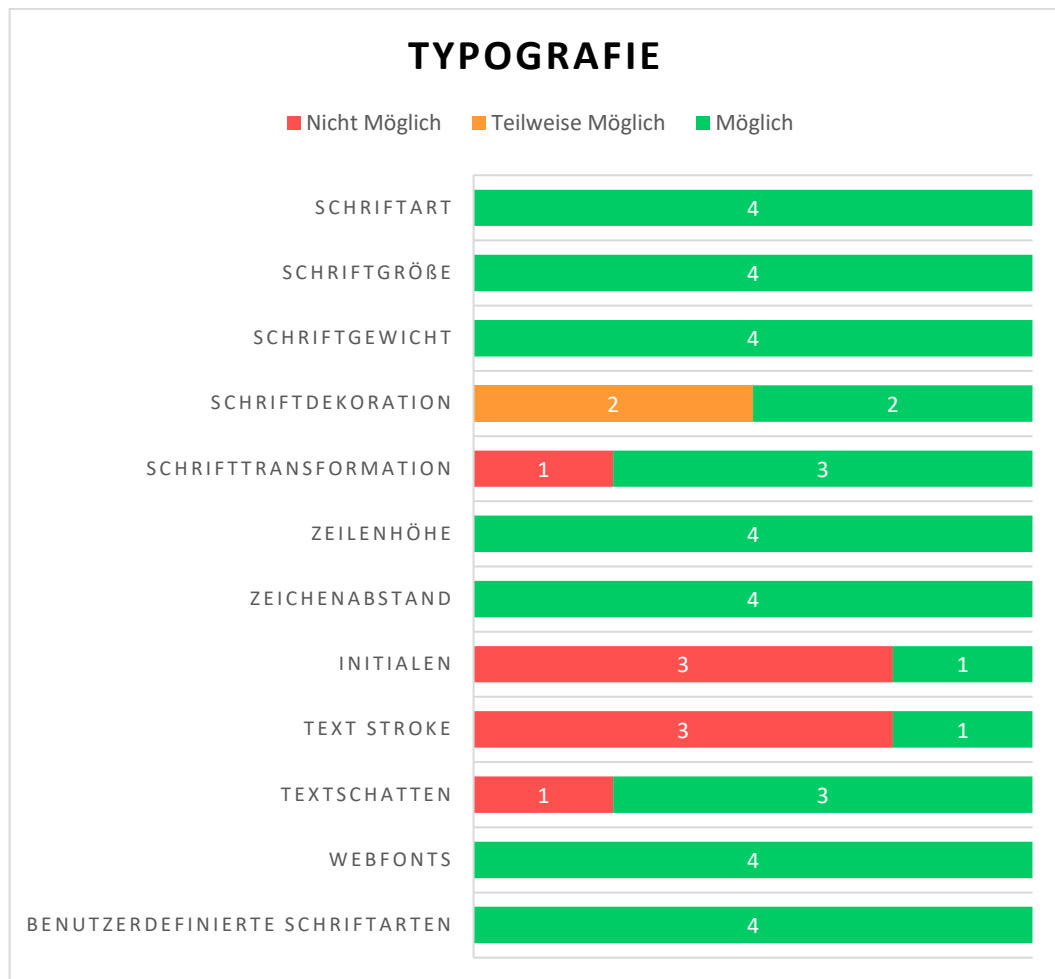
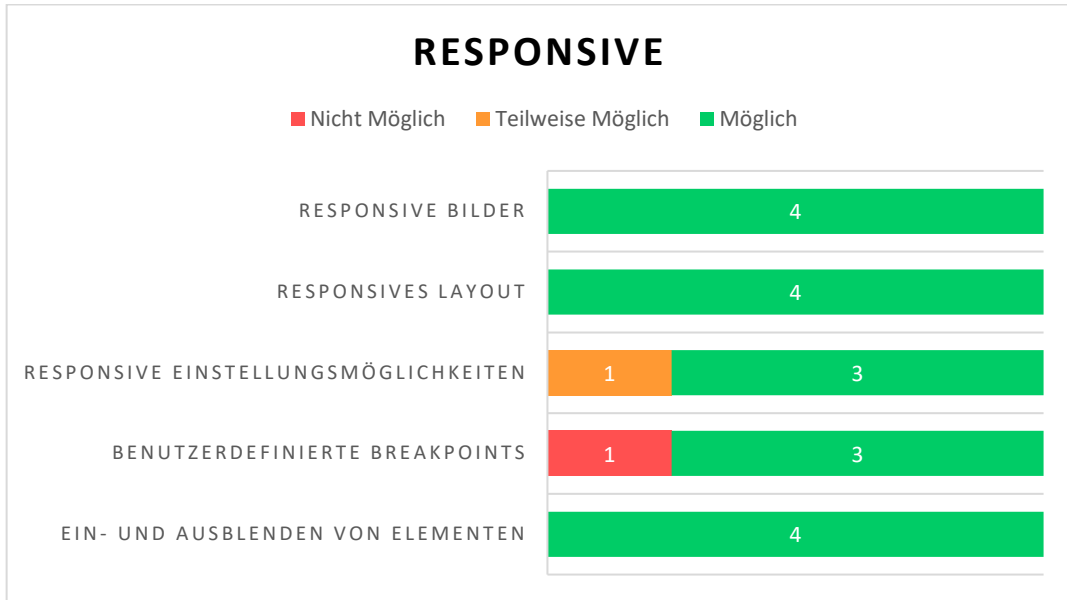


Abbildung 13. Ergebnisse Kategorie Typografie (eigene Darstellung)

Wie in Abbildung 13 ersichtlich ist, ist bei allen getesteten Plattformen Schriftart, -Größe, -Gewicht, Zeilenhöhe und Zeichenabstand einstellbar. Alle Systeme erlauben auch die Einbindung von Webfonts und den Upload eigener Schriftarten. Transformationen (Kapitälchen, Großbuchstaben, etc.) von Schriftarten und Textschatten lassen sich bei drei Plattformen konfigurieren. Zwei Lösungen ermöglichen diverse Schriftdekorationen, bei den verbleibenden Systemen ist dies nur eingeschränkt möglich. Ein System ermöglicht zudem das Erstellen von Initialen bei Textabsätzen, und die Veränderung der Strichstärke bei Schriften.

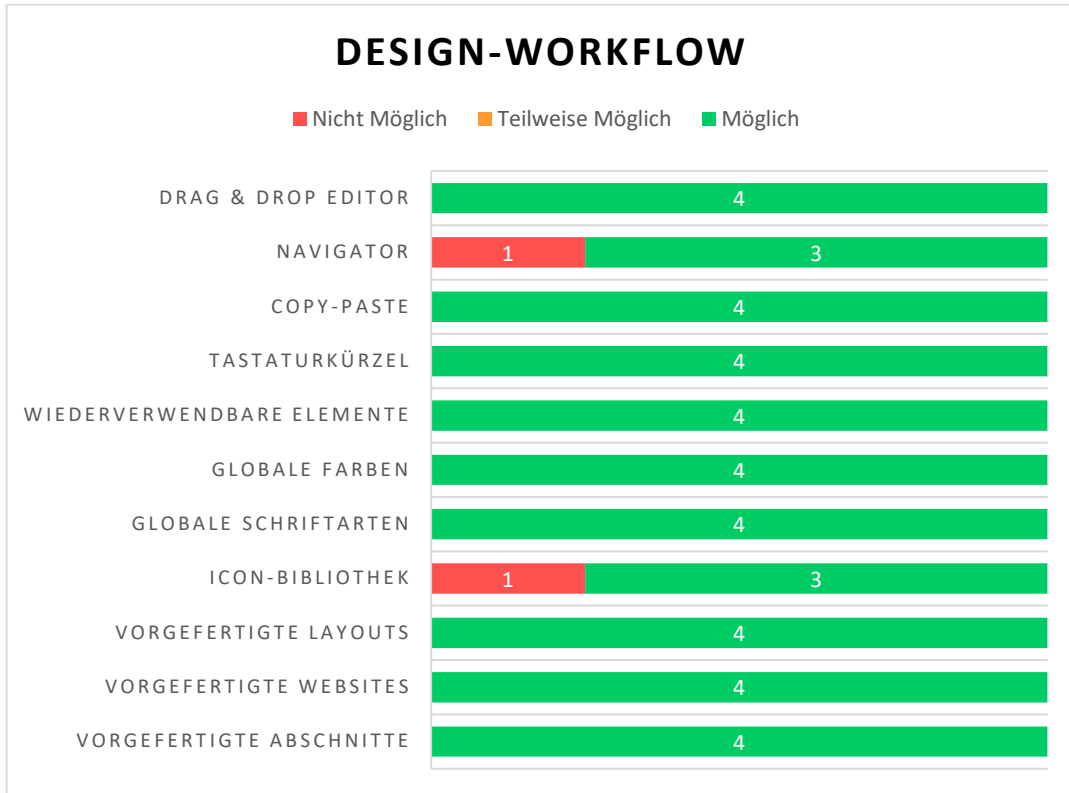
**Responsive**



*Abbildung 14. Ergebnisse Kategorie Responsive (eigene Darstellung)*

Alle untersuchten No-Code-Systeme werben mit der Möglichkeit zur Erstellung responsiver Websites. Die Ergebnisse in Abbildung 14 zeigen, dass dies bei allen vier Anbieter\*innen tatsächlich möglich ist. Bilder sowie das Layout passen sich demnach automatisch der Bildschirmgröße an, auch das Ein- und Ausblenden von Elementen basierend auf Breakpoints wird bei allen Systemen unterstützt. Drei Plattformen ermöglichen separate Einstellungen für mobile Geräte unabhängig von den Einstellungen für Desktop, bei einer Plattform sind hierbei die Einstellungsmöglichkeiten eingeschränkt. Lediglich bei einem System können ohne eigenen Code keine zusätzlichen Breakpoints definiert werden.

**Design-Workflow**



*Abbildung 15. Ergebnisse Kategorie Design-Workflow (eigene Darstellung)*

Die Ergebnisse in Abbildung 15 zeigen, dass alle vier Systeme eine Vielzahl an Funktionen beinhalten, welche den Design-Workflow unterstützen und optimieren. Alle Plattformen bieten einen Drag & Drop Editor an, welcher über eine Copy-Paste Funktion und Tastaturkürzel für unterschiedlichste Funktionen zur Reduzierung der Arbeitszeit verfügt. Ebenso ist das Definieren von globalen Farben und Schriftarten und das Speichern und Wiederverwenden von bereits erstellten Elementen bei allen Systemen möglich. Außerdem bieten alle Lösungen vorgefertigte Layouts, Abschnitte und sogar gesamte Website-Vorlagen an. Drei Systemen erleichtern die Arbeit durch Einsatz eines Navigators (= Listenansicht der auf der Website platzierten Elemente) und bieten eine umfassende Anzahl an frei verfügbaren Icons an.

## CMS-Funktionen

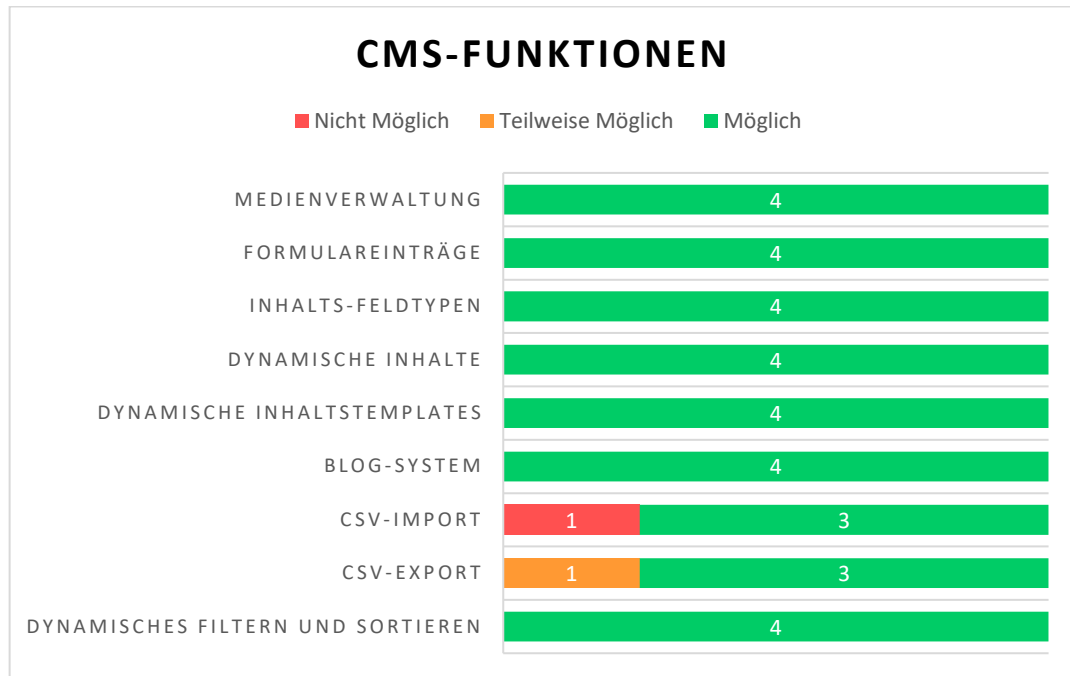


Abbildung 16. Ergebnisse Kategorie CMS-Funktionen (eigene Darstellung)

Wie Abbildung 16 zeigt, bieten die getesteten No-Code-Plattformen auch Funktionalitäten von klassischen Content-Management-Systemen an. Alle Lösungen verfügen über ein System zur Verwaltung unterschiedlicher Medien und Dateien. Sie erlauben es auch dynamische Inhalte in Websites zu platzieren und dafür eigene Templates zu erstellen. Für die dynamischen Inhalte werden bei allen Plattformen verschiedene Feldtypen bereitgestellt. Die Plattformen ermöglichen zudem die Erstellung eines Blogs und bieten unterschiedliche Filter- und Sortierungsoptionen für dynamische Inhalte an. Bei drei von vier Lösungen können Inhalte sowohl per CSV-Datei importiert als auch exportiert werden. Ein System erlaubt den Import nicht und verfügt nur über eine eingeschränkte Exportfunktion.

## Inhaltselemente

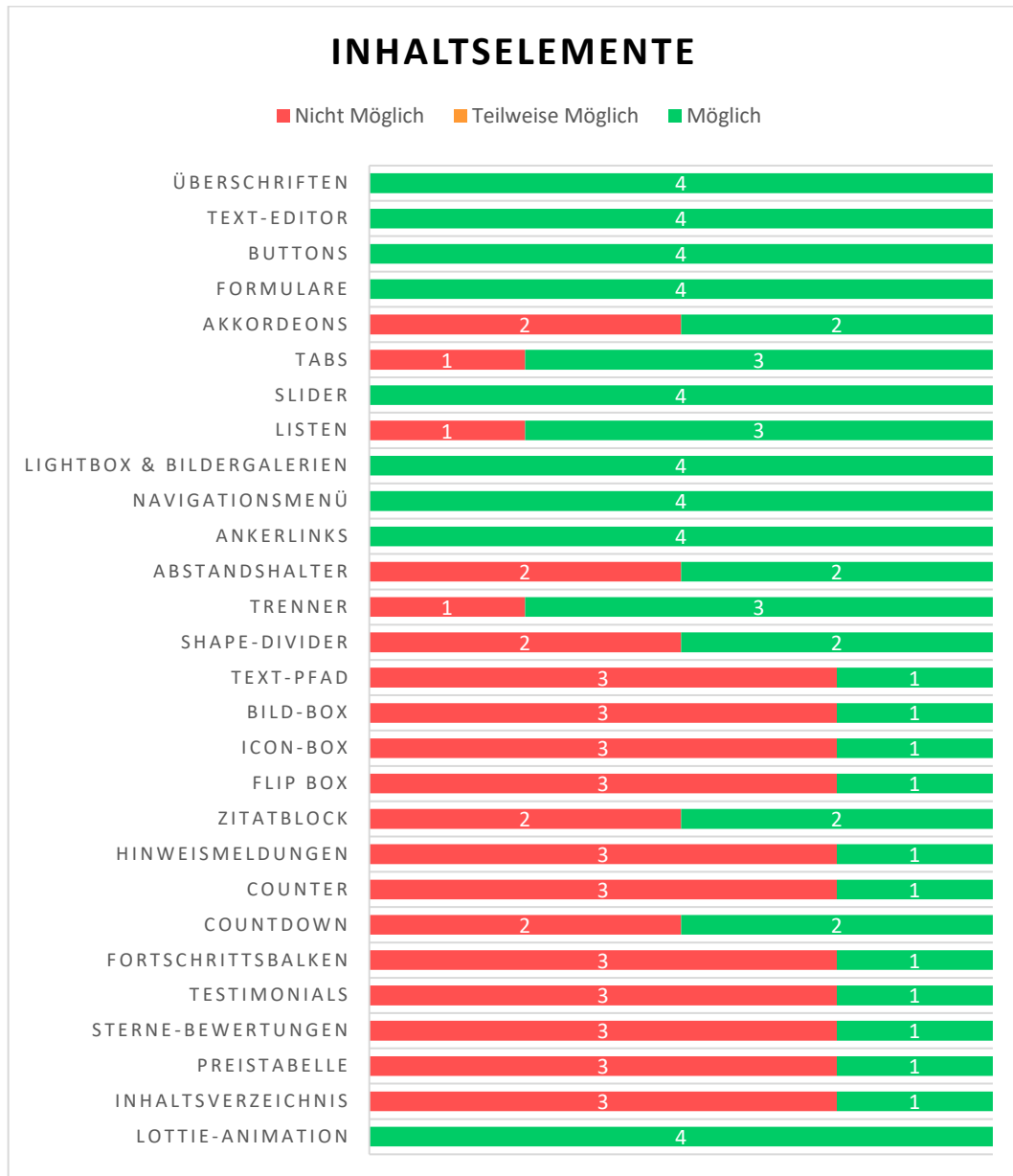


Abbildung 17. Ergebnisse Kategorie Inhaltselemente (eigene Darstellung)

Alle untersuchten Plattformen verfügen Abbildung 17 zufolge über grundlegende Inhaltselemente wie Überschriften, Absätze, Buttons, Formulare, aber auch über Slider, Bildergalerien mit Lightboxen, Navigationsmenüs und Ankerlinks. Ebenso bieten alle Systeme die Einbettung von Lottie (= Vektorgrafik für Animationen, auf Basis von SVG und JSON) an. Drei Plattformen verfügen über Tabs, Listen und

## 7 Ergebnisse

---

horizontale Trennlinien. Bei zwei Systemen können Akkordeons, Abstandshalter, Shape-Divider, Zitatblöcke und Countdowns ausgewählt werden. Eine Plattform verfügt darüber hinaus über zahlreiche weitere Inhaltselemente, welche in Abbildung 17 ersichtlich sind.

### Medien

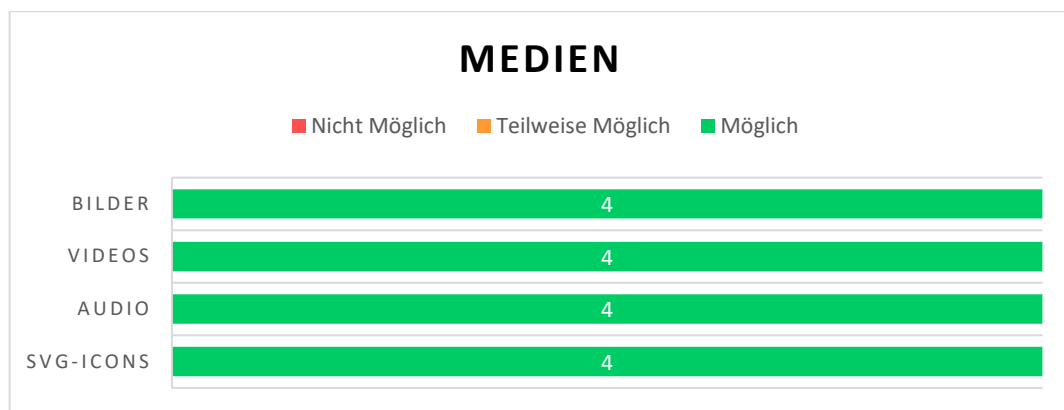


Abbildung 18. Ergebnisse Kategorie Medien (eigene Darstellung)

Alle untersuchten No-Code-Plattformen können Medien wie Bilder, Videos, Audiosequenzen und SVG-Grafiken verarbeiten (siehe Abbildung 18).

### Hintergrundmedien

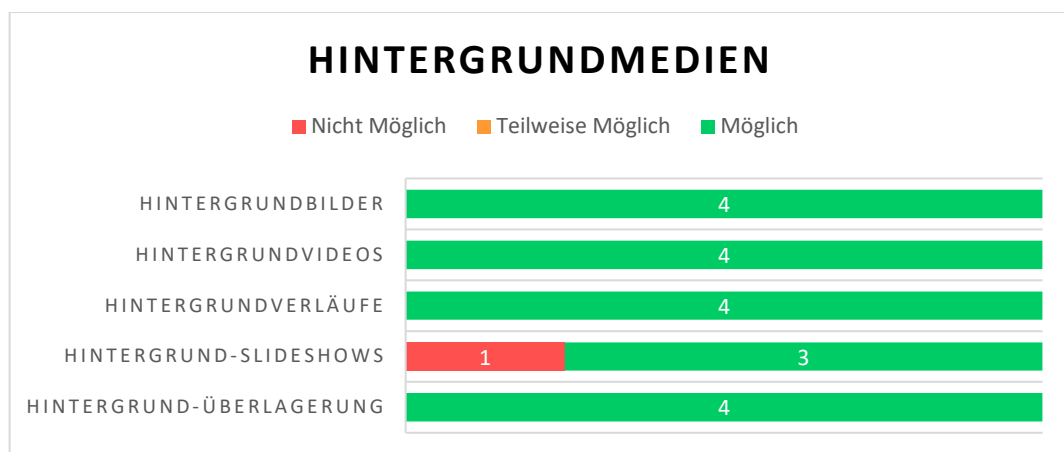


Abbildung 19. Ergebnisse Kategorie Hintergrundmedien (eigene Darstellung)

Abbildung 19 zeigt, dass Bilder, Videos, Farbverläufe und -Überlagerungen bei allen Systemen im Hintergrund platziert werden können. Ein System unterstützt keine Slideshows der zuvor genannten Medien im Hintergrund.

**Einbettungen**

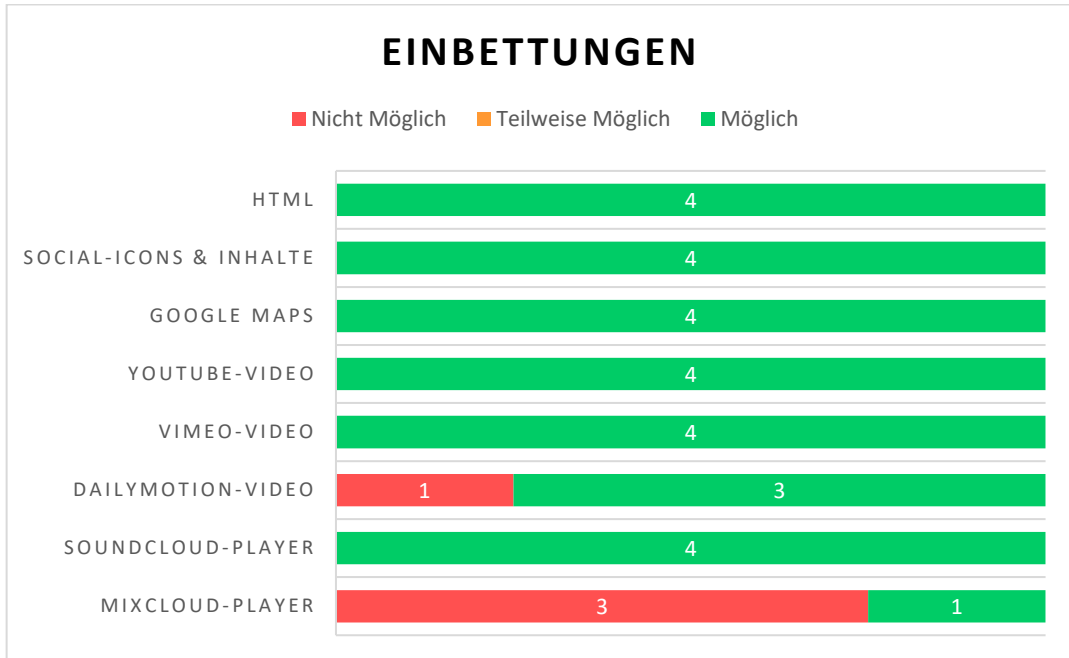


Abbildung 20. Ergebnisse Kategorie Einbettungen (eigene Darstellung)

Abbildung 20 zeigt, dass alle vier untersuchten No-Code-Plattformen die Einbettung von eigenem HTML-Code, unterschiedlicher Sozial-Media Inhalte, Google Maps, Videos der Dienste YouTube und Vimeo und Audioplayer des Dienstes SoundCloud ermöglichen. Drei Systeme ermöglichen auch Dailymotion-Videos und ein Audioplayer von Mixcloud.

**Integrationen**

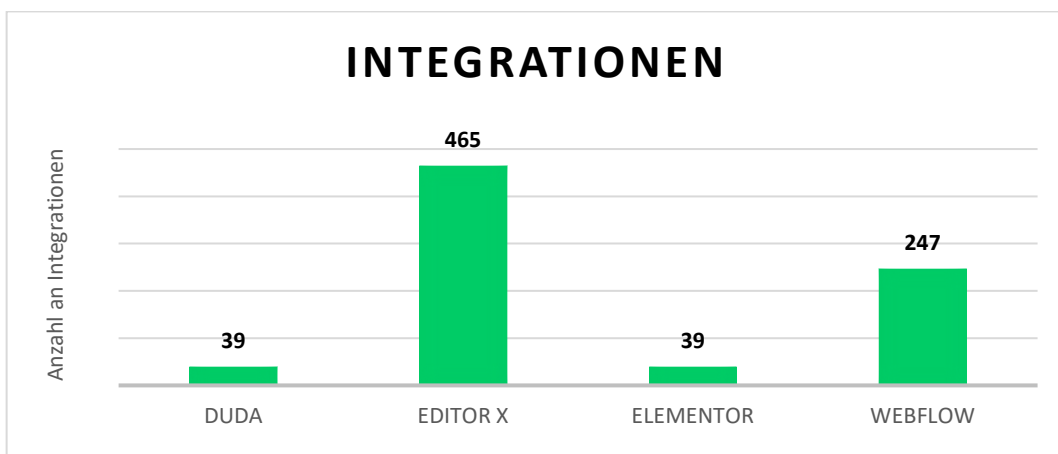


Abbildung 21. Ergebnisse Kategorie Integrationen (eigene Darstellung)

## 7 Ergebnisse

Die untersuchten Systeme werben auf ihren offiziellen Websites mit einer Vielzahl an integrierbaren Diensten. Editor X und Webflow geben dabei deutlich mehr Integrationsmöglichkeiten an als ihre Konkurrenten (siehe Abbildung 21). Es ist jedoch anzumerken, dass die tatsächliche Anzahl der unterstützten Integrationen von den auf den offiziellen Websites angeführten Zahlen abweichen kann.

### SEO & Accessibility

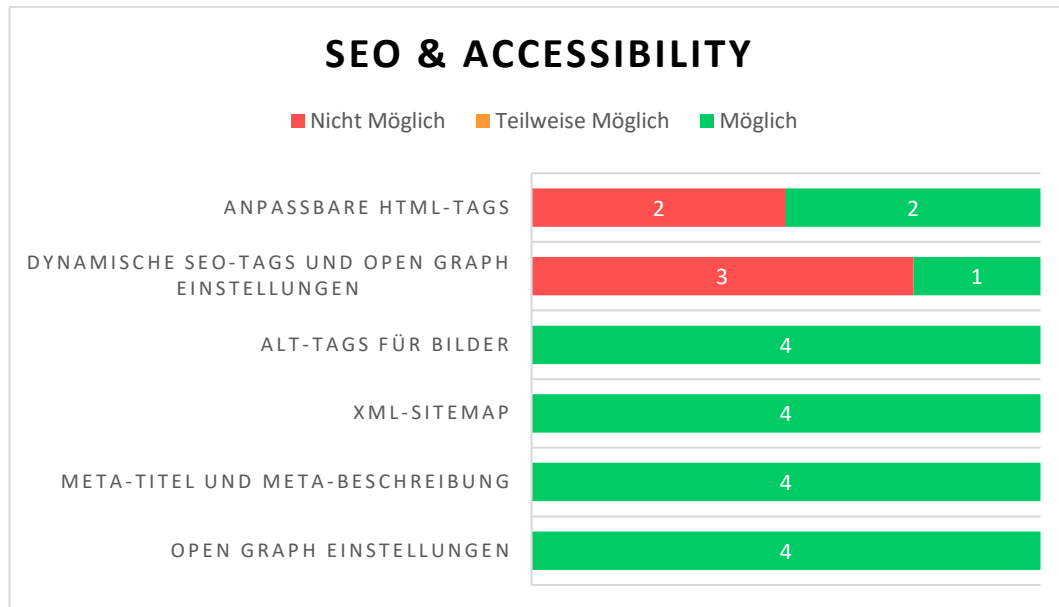


Abbildung 22. Ergebnisse Kategorie SEO & Accessibility (eigene Darstellung)

In Abbildung 22 ist ersichtlich, dass die getesteten Plattformen Funktionen zur Optimierung für Suchmaschinen und der Barrierefreiheit bereitstellen. In allen vier Systemen ist es möglich alternative Beschreibungen für Bilder einzugeben und SEO-Tags und Open Graph Inhalte festzulegen. Ebenfalls wird von allen Plattformen eine XML-Sitemap bei Veröffentlichung der Website erstellt. Zwei Plattformen ermöglichen es, die HTML-Tags der auf der Website platzierten Elemente nach Wunsch anzupassen. In einem System ist es zudem möglich, SEO-Tags und Open Graph Inhalte dynamisch aus den Inhalten der Website generieren zu lassen.

## Sonstige Funktionen

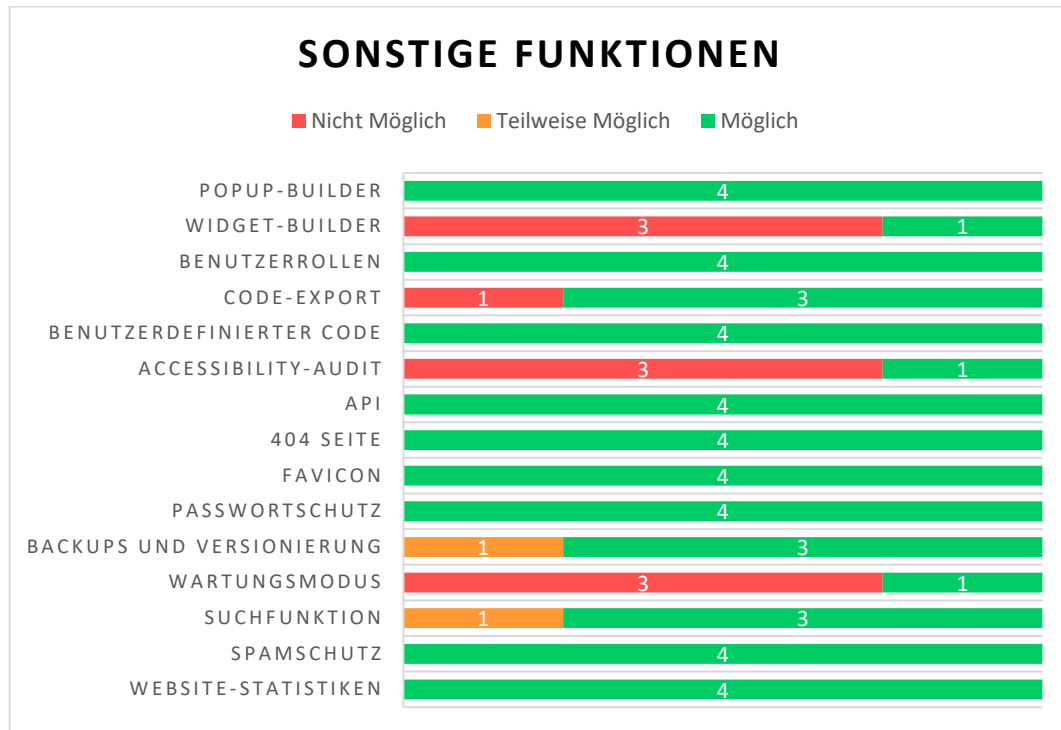


Abbildung 23. Ergebnisse Kategorie sonstige Funktionen (eigene Darstellung)

Über die zuvor genannten Features hinaus, verfügen die untersuchten Systeme über zusätzliche Funktionen (siehe Abbildung 23). Alle vier Plattformen ermöglichen es Popups und Modalboxen zu erstellen. Mit einer Plattform können auch eigene Widgets erstellt werden. Mithilfe verschiedener Rollen für die Nutzer\*innen lassen sich die Einstellungsmöglichkeiten der Systeme den Bedürfnissen zuschneiden. Alle Systeme erlauben den Einsatz von eigenem Code und bieten dafür auch eine API (= Programmierschnittstelle) an. Das Erstellen einer 404-Seite und das Hinzufügen eines Favicons ist in allen Lösungen problemlos möglich. Auch werden Funktionen zur Prävention von Spam und zur Aufzeichnung von Nutzungsstatistiken in allen Systemen bereitgestellt. Nach außen hin können die erstellten Websites bei allen Anbieter\*innen mit einem Passwort geschützt werden. In drei Systemen ist der Export des Codes der Website möglich, auch werden regelmäßig Website-Backups generiert und die Websites können mit einer Suchfunktion ausgestattet werden, in einer Plattform jedoch nur durch einen externen Dienst. Ein System verfügt darüber hinaus über weitere Funktionen wie einen Assistenten für die Optimierung der Barrierefreiheit oder einen Wartungsmodus während der Websiteerstellung.

## 7.2 Ergebnisse der Webdesignumsetzung

In diesem Kapitel werden die Ergebnisse der Websiteanalyse präsentiert. Die auf der Website vorhandenen Elemente wurden erfasst und im Rahmen einer qualitativen Inhaltsanalyse analysiert. Daraus wurden Kriterien abgeleitet und in einem Kriterienkatalog gesammelt, welcher die Basis für die Evaluierung bildet.

### 7.2.1 Websiteanalyse

Die nachfolgenden Abbildungen 24 bis 28 zeigen den Aufbau und die nachzubildenden Elemente der Website. Diese wurde zur einfacheren Unterscheidung farblich hinterlegt und benannt. Die Bezeichnungen der Elemente widerspiegeln die im Kriterienkatalog erstellten Kategorien.



Abbildung 24. Elemente Header & Aside (eigener Screenshot)



Abbildung 25. Elemente Mobile Navigation (eigener Screenshot)

## 7 Ergebnisse

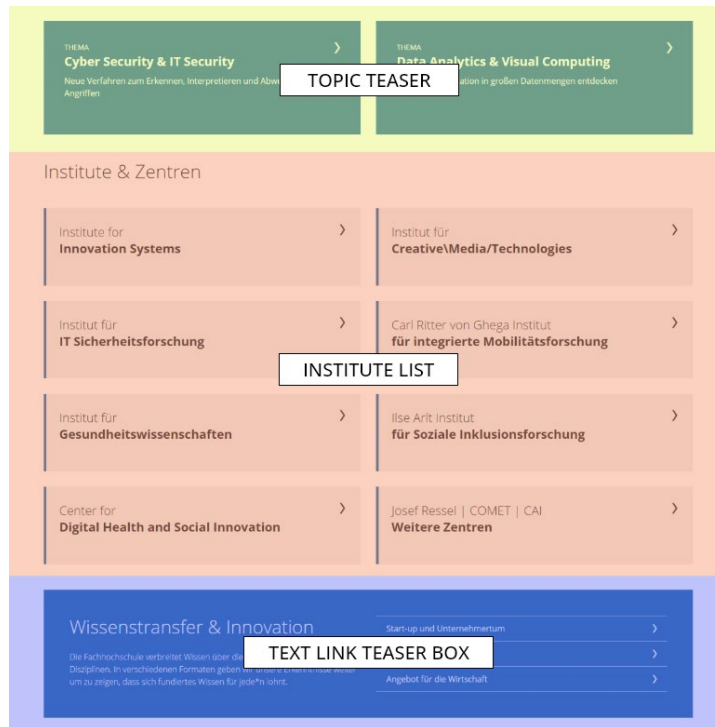


Abbildung 26. Elemente Inhaltsbereich 1 (eigener Screenshot)

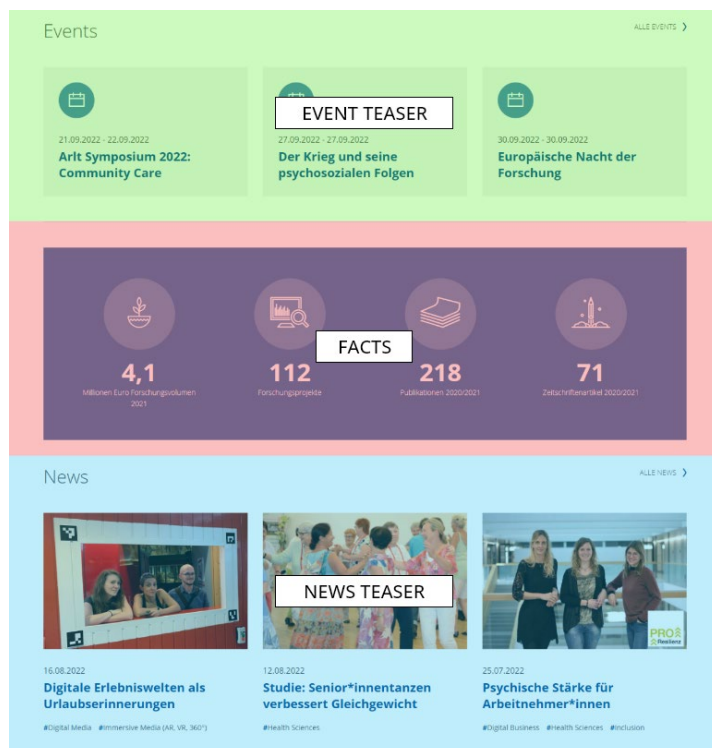


Abbildung 27. Elemente Inhaltsbereich 2 (eigener Screenshot)

## 7 Ergebnisse



Abbildung 28. Elemente Footer (eigener Screenshot)

### 7.2.2 Kriterienkatalog

Die folgende Tabelle zeigt die, im Laufe der qualitativen Inhaltsanalyse eruierten, Kriterien für die weitere Evaluierung. Es wurden vier Hauptkategorien, zwölf Unterkategorien und 64 Kriterien für die Bewertung der No-Code-Tools gebildet.

Tabelle 6. Kriterienkatalog der Webdesignumsetzung

Hauptkategorie     Unterkategorie

Kategorie	Beschreibung
<b>Layout</b>	
<i>Main</i>	
HTML-Tag	main
Breite	1400px
Höhe	100vh
Hintergrund	#FFFFFF
Rahmen-Oben (nur Mobil)	5px solid #004F9A
<i>Aside</i>	
HTML-Tag	aside
Hintergrund	Hintergrundbild (50% 50%, cover, no-repeat)
Position	fixed
Abstand Links	1400px

## 7 Ergebnisse

<i>Article (Inhaltsbereich)</i>	
HTML-Tag	article
Breite	1290px
Padding	Links Rechts 15px
2-spaltig	50% 50%
3-spaltig	33% 33% 33%
<b>Header</b>	
<i>Navigation</i>	
HTML-Tag	nav
Breite	1290px
Höhe	60px
Rahmen	1px solid #5C5D64
Hintergrund	#4A4B53
Anordnung	Horizontal
Hover-Effekt	Anpassung der Hintergrundfarbe auf #5C5D64
Suche	Benutzerdefiniertes Icon Beim Anklicken: Suchformular-Modal öffnen & schließen
<i>Mobile Navigation</i>	
Menü-Icon	Beim Anklicken: Animation zu „X“, Beim Anklicken: Ein- und Ausblenden der Navigation
Breite	100%
Hintergrund	#4A4B53
Anordnung	Vertikal
Suchformular	300px breit, 42px hoch
<i>Hero-Bereich</i>	
Lead-Text	Open Sans / Größe 30px / Gewicht: 300 / Zeilenhöhe: 36px
Logo	190px breit, 90px hoch
Suchformular-Container	Hintergrund: Hintergrundbild Höhe: 360px Padding: 70px

## 7 Ergebnisse

Suchformular	Breite: 420px Höhe: 60px Textfarbe: #999999
Suchformular-Interaktion	Beim Anklicken: Dropdown-Auswahlsfeld öffnen
<b>Inhaltsbereich</b>	
<i>Topic Teaser</i>	
Padding	40px
Hintergrund	#4A4B53
Text	Open Sans / Größe: 14px, 24px, 16px / Gewicht: 400, 700 / Zeilenhöhe: 22px, 24px
Hover-Effekt	Hintergrundfarbe auf #004F9A eingehend von links, ausgehend von rechts Pfeil: Keyframe-Animation
<i>Institute List</i>	
Padding	30px
Hintergrund	#F2F2F2
Text	Open Sans / Größe: 36px, 24px / Gewicht: 300, 700 / Zeilenhöhe: 43px, 32px
Rahmen-Links	5px solid #4A4B53
Hover-Effekt	Rahmen: Farbe auf #004F9A Text: Farbe auf #004F9A Pfeil: Keyframe-Animation
<i>Text Link Teaser Box</i>	
Padding	50px
Hintergrund	#4C84B8
Text	Open Sans / Größe: 36px, 16px / Gewicht: 300, Zeilenhöhe: 43px, 24px
Rahmen Oben & Unten	1px solid #FFFFFF / Deckkraft 50%
Hover-Effekt	Rahmen: alle Seiten / Deckkraft 100% Pfeil: Keyframe-Animation
<i>Event Teaser</i>	
Padding	30px
Hintergrund	#F2F2F2
Text	Open Sans / Größe: 36px, 24px, 16px, 14px / Gewicht: 700, 400, 300 / Zeilenhöhe: 43px, 24px
Icon	benutzerdefiniertes Icon / Hintergrund: #4C84B8 / Border-Radius: 999px

## 7 Ergebnisse

Horizontale Linie	1px solid #DBDBDD
Hover-Effekt	Links: Unterstreichen Pfeil: Keyframe-Animation
<i>Facts</i>	
Padding	60px
Hintergrund	#4C84B8
Text	Open Sans / Größe: 48px, 14px / Gewicht: 700 / Zeilenhöhe: 48px, 22px
Icon	benutzerdefiniertes Icon / Hintergrund: #709DC6 / Border-Radius: 999px
<i>News Teaser</i>	
Padding	Oben & Unten 25px, Rechts & Links 15px
Text	Open Sans / Größe: 36px, 24px, 16px, 14px / Gewicht: 700, 400, 300 / Zeilenhöhe: 43px, 24px
Hover-Effekt	Links: Unterstreichen Pfeil: Keyframe-Animation
<b>Footer</b>	
<i>Footer-Top</i>	
Padding	Oben & Unten 50px
Horizontale Linie	1px solid #DBDBDD
Text	Open Sans / Größe: 16px / Gewicht: 700, 300
Externer Link	Position: absolute
Hover-Effekt	Links: Unterstreichen
<i>Footer-Bottom</i>	
Icons	benutzerdefiniertes Icon / Hintergrund: #004F9A
Hover-Effekt	Links: Unterstreichen Icons: Deckkraft ändern auf 0.7

### 7.2.3 Ergebnisse der Evaluation

Dieses Unterkapitel enthält die Ergebnisse, der im Rahmen der Webdesignumsetzung durchgeführten Evaluation beider No-Code-Plattformen.

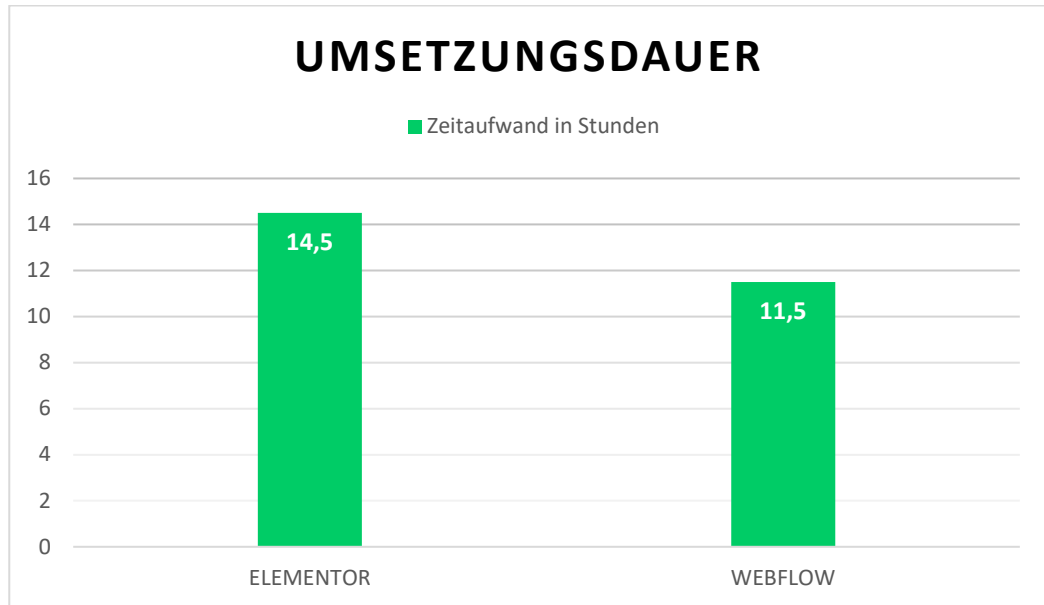


Abbildung 29. Dauer der Webdesignumsetzung (eigene Darstellung)

Abbildung 29 zeigt den Zeitaufwand der Webdesignumsetzung in Stunden. Die vollständige Umsetzung der Website mit der No-Code-Plattform Elementor dauerte von der Einrichtung und Installation der lokalen Entwicklungsumgebung bis zur Vervollständigung der Website 14 Stunden und 30 Minuten.

Mit Webflow konnte die Website drei Stunden früher fertiggestellt werden, nämlich innerhalb von 11 Stunden und 30 Minuten. Hierbei sollte erwähnt werden, dass durch die Bereitstellung von Webflow als Plattform as a Service, die Serverkonfiguration beziehungsweise Software-Installation zu Beginn wegfällt.

### Elementor

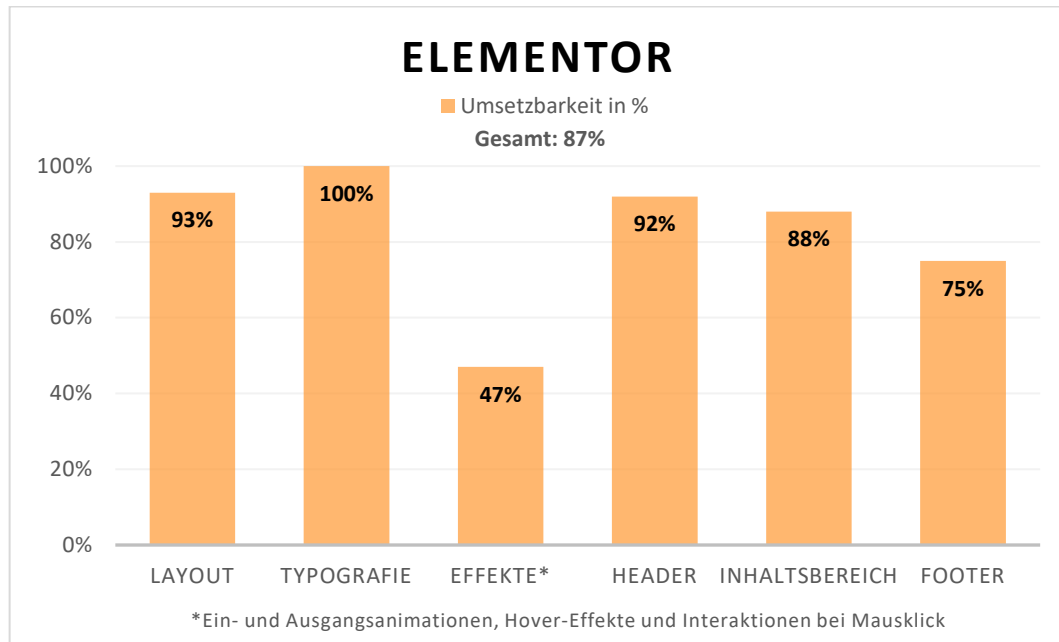


Abbildung 30. Ergebnisse der Evaluation von Elementor (eigene Darstellung)

Wie in Abbildung 30 ersichtlich ist, ist eine detailgetreue Nachbildung des Webdesigns mit der No-Code-Plattform Elementor zu 87% möglich. Das Layout der Website lässt sich beinahe vollständig nachbilden. Das Ändern des HTML-Tags ist bei Containern problemlos, bei weiteren Elementen aber nur bedingt möglich, jedoch ist beispielsweise bei Navigationselementen der HTML-Tag `<nav>` bereits vorausgewählt. Ebenfalls lassen sich Breite und Höhe bei Container-Elementen mit verschiedenen Einheiten festlegen. Innen- und Außenabstände sowie Hintergrundfarben oder Rahmen können bei allen Elementen angepasst werden. Einschränkungen treten in dieser Kategorie ausschließlich bei der Positionierung des Hintergrundbildes im `Aside`-Tag der Website auf. Beim Verändern der Breite des Browserfensters, skaliert das Hintergrundbild trotz gleicher CSS-Anweisungen nicht wie auf der originalen Website in vollständiger Höhe mit. Hier wäre der Einsatz von zusätzlichem CSS-Code notwendig, um ein identisches Verhalten wie in der Designvorlage zu erreichen.

Die Texte auf der Website lassen sich durch die umfassenden Typografie-Einstellungen in Elementor zu 100% designgleich abbilden. Jedes Element, das Text enthält, verfügt über dieselben Typografie-Optionen, welche auf kleineren Bildschirmgrößen nochmals mit unterschiedlichen Werten überschrieben werden können. Die Einbindungsmöglichkeit von Webschriften via Google Fonts oder

## 7 Ergebnisse

---

Adobe Fonts sowie das Hochladen eigener Schriften erweitern zudem die Auswahl der verfügbaren Schriften.

Die Kategorie Effekte, zu der Eingangs- und Ausgangsanimationen von Elementen, Effekte beim Hover oder Anklicken von Elementen und weitere Interaktionen auf der Website gezählt werden, schneidet mit nur 47% Umsetzbarkeit am schlechtesten ab. Dies lässt sich dadurch begründen, dass Elementor Hover-Effekte nur bei bestimmten Elementen ermöglicht. Mit CSS-Code lassen sich jedoch mittels der Pseudoklasse `:hover` jedem beliebigen Element unterschiedliche Stile im Hover-Zustand verpassen. Unterstrichene Links beim Hover, wie auf der originalen Website, lassen sich demnach mit Elementor nicht nachbilden. Ein weiterer Schwachpunkt besteht darin, dass beim Hover auf ein Element, nur der Stil dieses Elements selbst verändert werden kann. Kind-Elemente innerhalb des Elements reagieren auf den Hover-Zustand des Eltern-Elements nicht. Ebenso kann die Ausgangsanimation des Suchformular-Modals nicht repliziert werden, da in Elementor nur Eingangsanimationen von Elementen konfiguriert werden können. Aufwändigere Keyframe-Animationen, wie etwa beim Hover auf die nach rechts zeigenden Pfeile oder beim Klick auf das Menü-Icon der mobilen Navigation, können ohne eigenen Code in Elementor nicht umgesetzt werden. Elementor bietet für Icons stattdessen eine begrenzte Auswahl vorgefertigter Animationen an, von denen jedoch keine mit der Animation auf der originalen Website übereinstimmt.

Die Elemente im Header der Website können zu 92% mit Elementor umgesetzt werden. Punkteabzug gibt es für das Suchformular-Modal, denn dieses lässt sich nur über den dedizierten „X“-Button schließen und nicht per Mausklick auf den abgedunkelten Hintergrund, wie es bei der originalen Website der Fall ist. Auch das Formularfeld im Hero-Bereich der Website lässt sich mit Elementor nicht exakt abbilden, da das Standard-Auswahlfeld des Elementor-Formulars nur über eingeschränkte Optionen verfügt. Auf der originalen Website kommt hier ein Select2-Auswahlfeld zum Einsatz. Select2 ist eine JavaScript-Bibliothek (<https://select2.org/>), welche die Funktionen des normalen HTML-Auswahlfeldes erweitert. Für beide soeben genannten Funktionalitäten bedarf es an eigenem CSS- und JavaScript-Code.

Der Inhaltsbereich der Website kann zu 88% nachgebildet werden. Elementor unterstützt die Regeln von CSS-Flexbox, was die mehrspaltigen Layouts im Inhaltsbereich problemlos umsetzbar macht und für eine responsive Anpassung der Elemente auf mobilen Geräten sorgt. Einzig die oben genannten

## 7 Ergebnisse

---

Einschränkungen bei Hover-Effekten sorgen dafür, dass die Elemente im Inhaltsbereich nicht designgleich umgesetzt werden können.

Der Footer lässt sich zu 75% detailgetreu abbilden. Ohne den Einsatz von eigenem CSS-Code lassen sich die Navigationslinks mit den Bordmitteln von Elementor beim Hover nicht unterstreichen.

### Webflow

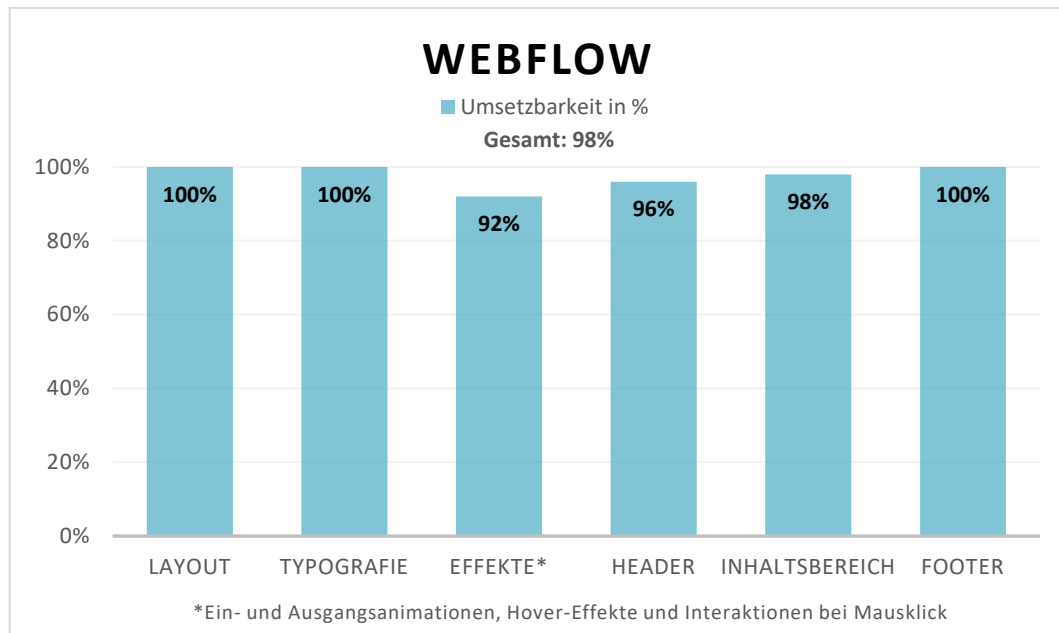


Abbildung 31. Ergebnisse der Evaluation von Webflow (eigene Darstellung)

Abbildung 31 zeigt die Ergebnisse der Umsetzung mit der No-Code-Plattform Webflow. Es lässt sich erkennen, dass eine idente Nachbildung der Website ohne den Einsatz von eigenem Code beinahe möglich ist. Die Gesamtwertung beläuft sich auf 98% Umsetzbarkeit.

Das Layout der Website lässt sich zu 100% detailgetreu nachbilden. Grund dafür sind die umfangreichen Einstellungsmöglichkeiten, die die Plattform bietet. Im Gegensatz zu Elementor, lassen sich in Webflow sämtliche Einstellungen auf alle verfügbaren Elemente anwenden. Somit konnten die meisten CSS-Eigenschaften der originalen Website in Webflow nachgeahmt werden, was in einem identischen Layout der Website resultiert.

Auch die Typografie der Designvorlage lässt sich zu 100% kopieren. Webflow bietet alle notwendigen Typografie-Einstellungen, welche auf der originalen Website eingesetzt werden.

In der Kategorie Effekte erzielt Webflow eine Umsetzbarkeit von 92%. Die Plattform ermöglicht das Anpassen der unterschiedlichen Zustände von Elementen, wie etwa `:hover`, `:focus` oder `:active`. Dadurch ist es möglich ohne eigenen Code Hover-Effekte auf alle eingesetzten Elemente anzuwenden. Weiters beinhaltet Webflow eine Funktion zur Erstellung von Animationen und Interaktionen. Damit lassen sich eigene Animationen erstellen, welche auf unterschiedliche Trigger wie zum Beispiel Hover, Mausklick oder Scrollen reagieren. Die Animationen und Interaktionen der originalen Website lassen sich damit vollständig replizieren. Abzug gibt es in dieser Kategorie lediglich dafür, dass sich das Suchformular-Modal nicht ohne weiteres per Mausklick auf den abgedunkelten Hintergrund schließen lässt.

Die detailgetreue Umsetzung des Header-Bereichs ist zu 96% möglich. Wie Elementor bietet auch Webflow nicht die Möglichkeit das Auswahlfeld des Suchformulars im Hero-Bereich gleich darzustellen wie auf der originalen Website. Zusätzlicher CSS- und JavaScript-Code sind hierfür notwendig.

Die restlichen Bereiche der Website, der Inhaltsbereich und der Footer, lassen sich mit Webflow vollständig designgleich nachbilden. In der Kategorie Inhaltsbereich wurde jedoch ein Punkt abgezogen, da die Hintergrundanimation der Topic Teaser Elemente nicht durch eine Verschiebung des Hintergrundverlaufes umgesetzt werden konnte, sondern durch die Anpassung der Breite eines zusätzlichen `<div>` Elementes, welches hinter den Text-Elementen platziert wurde. Der Effekt sieht jedoch mit beiden Herangehensweisen identisch aus.

Die vollständige Bewertungstabelle beider untersuchten No-Code-Plattformen ist im Anhang dieser Arbeit im Abschnitt C Evaluierungsergebnisse Experiment (Seite 121) ersichtlich.

### 7.2.4 Qualitätsmessung

In diesem Kapitel werden die Ergebnisse der Qualitätsmessung präsentiert, welche nach vollständiger Umsetzung der Websites mit beiden No-Code-Plattformen durchgeführt wurde.

Da die Software Webflow in der Cloud bereitgestellt wird und die mit Webflow erstellten Websites auf der Serverinfrastruktur des Anbieterunternehmens laufen,

## 7 Ergebnisse

Elementor hingegen als WordPress-Plugin auf eigenen Servern installiert wird, musste vor Durchführung der Messung erst eine faire Testumgebung geschaffen werden. Um dies zu bewerkstelligen, wurde die komplette WordPress-Installation samt Datenbank aus der lokalen Entwicklungsumgebung exportiert und auf den, von der Fachhochschule Sankt Pölten zur Verfügung gestellten, Webserver des Forschenden hochgeladen. In Webflow wurde die Funktion „Code Export“ genutzt, welche die erstellte Website in ihren Einzelteilen als HTML-, CSS- und JavaScript-Dateien zum Download verfügbar macht. Auch die genutzten Assets (Bilder und Grafiken) wurden exportiert. All diese Dateien wurden anschließend ebenfalls auf den Webserver des Forschenden hochgeladen. Somit kann sichergestellt werden, dass beide Messungen auf gleicher Serverinfrastruktur durchgeführt werden und mögliche Unterschiede allein durch die Daten der Websites verursacht werden.

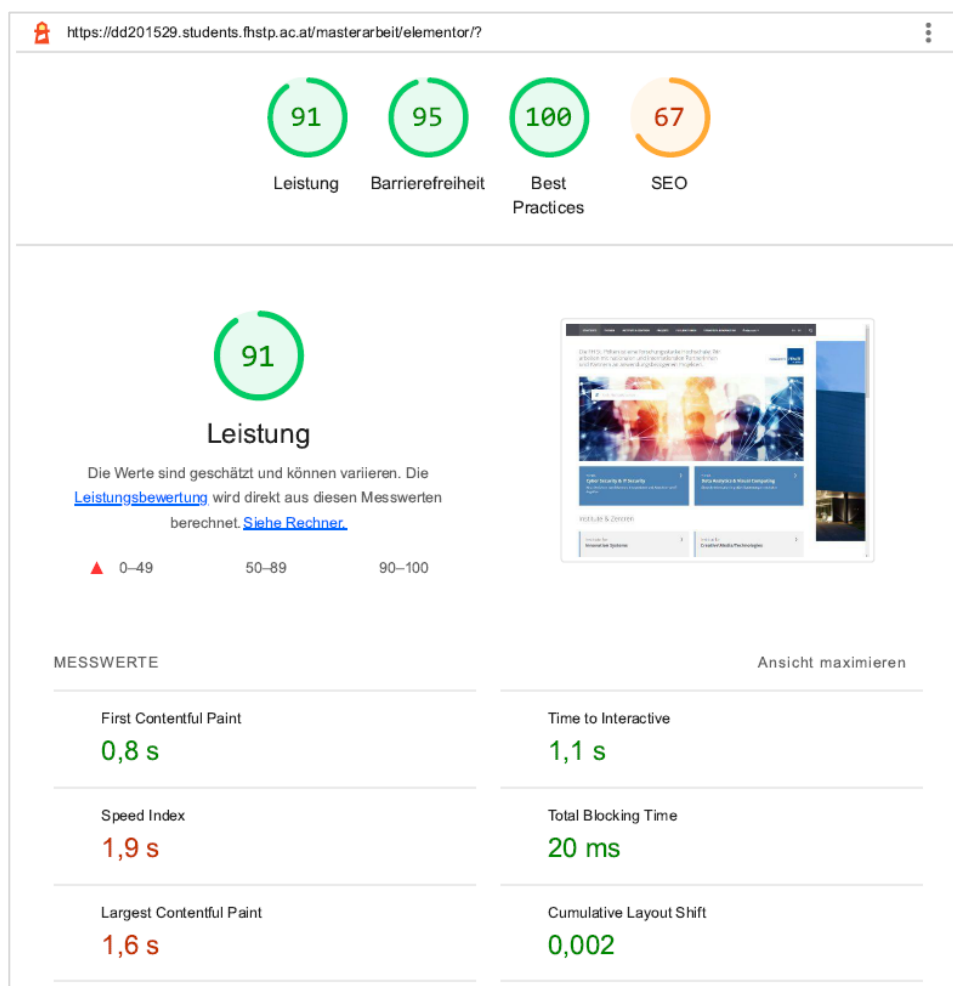


Abbildung 32. Ergebnisse Qualitätsmessung Elementor (eigener Screenshot)

Abbildung 32 zeigt die Ergebnisse der Qualitätsmessung, der mittels Elementor erstellten Website. Grundsätzlich ist zu erwähnen, dass das Messergebnis schwanken kann. Diese Schwankungen lassen sich laut Google (Google, 2022a) auf folgende sich ändernde Umstände und Rahmenbedingungen zurückführen:

- A/B Tests oder Änderungen von Werbeanzeigen auf Websites
- Änderungen bei der Weiterleitung des Internetverkehrs
- Tests auf unterschiedlichen Geräten, z.B. auf einem leistungsstarken Desktop und einem leistungsschwachen Laptop
- Browser-Erweiterungen, welche JavaScript einfügen und Netzwerk-anfragen verändern
- Antivirus-Software

Dennoch lässt sich durch die ermittelten Messwerte ein klares Bild über die Qualität der erfassten Website zeichnen. Die Messwerte reichen von 0 bis 100 und sind in Bereiche aufgeteilt, welche die folgende Farbcodierung aufweisen:

- 0 bis 49 (Rot): Schlecht
- 50 bis 89 (Orange): Verbesserungsbedürftig
- 90 bis 100 (Grün): Gut

Um ein gutes Nutzungserlebnis zu bieten, sollten Websites eine gute Bewertung (90-100) anstreben. Ein „perfekter“ Wert von 100 ist hingegen extrem schwierig zu erreichen und sollte nicht erwartet werden (Google, 2022a).

Die Wertung 91 in der Kategorie Leistung setzt sich aus den darunter stehenden Messwerten zusammen. Google (Google, 2022b) erklärt diese folgendermaßen:

- **First Contentful Paint** gibt an, wann der erste Text oder das erste Bild gezeichnet wird.
- **Largest Contentful Paint** gibt an, wie lange das Zeichnen des längsten Textes oder des größten Bildes dauert.
- **Speed Index** Der Geschwindigkeitsindex gibt an, wie schnell die Inhalte einer Seite sichtbar dargestellt werden.
- **Time to Interactive** Die „Zeit bis Interaktivität“ entspricht der Zeit, die vergeht, bis die Seite vollständig interaktiv ist.
- **Total Blocking Time** Summe aller Zeiträume (in Millisekunden) zwischen FCP und Time to Interactive, wenn die Aufgabendauer 50ms überschreitet.
- **Cumulative Layout Shift** misst die Bewegung sichtbarer Elemente innerhalb des Darstellungsbereichs.

## 7 Ergebnisse

---

Aus diesen Werten lässt sich interpretieren, dass die mit Elementor erstellte Website während dieses Messvorgangs in unter zwei Sekunden vollständig geladen wurde und bereits in einer Sekunde interaktiv war. Somit konnten in Bezug auf die Ladezeit durchaus gute Werte erzielt werden. Empfehlungen, um die Ladezeit weiter zu verringern, welche die Website und nicht den Server betreffen, wären beispielsweise „Bilder in modernen Formaten bereitzustellen“ oder „Ressourcen zu beseitigen, die das Rendering blockieren“. Ein weiterer Vorschlag ist es „übermäßige DOM-Größe zu vermeiden“, denn der Report hat auf der Website 899 Elemente gezählt. Die originale Website enthält lediglich 392 Elemente. Elementor benötigt demnach mehr als doppelt so viele DOM-Elemente, um das gleiche Webdesign darzustellen.

Die Barrierefreiheit der Website wurde mit dem Wert 95 bewertet. Elementor verwendet semantische HTML-Tags sowie ARIA-Roles bei den vorgefertigten Elementen, dadurch dürfte die gute Bewertung zustande kommen. Eine Warnung bezüglich nicht ausreichendem Kontrastverhältnis von Hintergrund- und Vordergrundfarben sorgt für Punkteabzug. Dieser Punkt betrifft jedoch auch die originale Website von welcher die Farben übernommen wurden.

In der Kategorie Best Practices erzielte die Website die volle Punktzahl. Dieser Wert setzt sich unter anderem daraus zusammen, dass die Website HTTPS verwendet, Bilder mit korrekten Seitenverhältnissen und angemessener Auflösung anzeigt, den HTML-Doctype und einen korrekt definierten Zeichensatz nutzt, veraltete APIs vermeidet, keine Browserfehler in der Konsole protokolliert und ähnliche Metriken.

67 Punkte konnten in der Kategorie SEO erreicht werden. Begründet wird diese Wertung damit, dass die Website keine Meta-Beschreibung enthält. Da die originale Website ebenfalls über keine Meta-Beschreibung verfügt, wurde darauf im Rahmen des Experiments verzichtet. Ebenfalls ist die Website von der Indexierung durch Suchmaschinen ausgeschlossen. Auch dies war eine bewusste Entscheidung, da diese Website ausschließlich als Experiment im Rahmen dieser Masterarbeit entwickelt wurde und nicht zur Veröffentlichung gedacht ist.

Abbildung 33 zeigt die Ergebnisse der Qualitätsmessung, der mit Webflow erstellten Website. Die ermittelten Werte ähneln denen, der Elementor-Website, was wiederum ebenfalls für eine Website mit guter Qualität spricht. Ein paar Unterschiede lassen sich jedoch erkennen. Die Kategorie Leistung ergibt eine Wertung von 95. Dafür sorgt vermutlich der Speed Index, welcher mit 0,7 Sekunden wesentlich kürzer ausfällt als bei Elementor. Die restlichen Messwerte

## 7 Ergebnisse

unterscheiden sich nur gering. Somit lässt sich sagen, dass beide Websites in unter 2 Sekunden vollständig laden. Die Webflow-Website lädt dabei eine Spur schneller als die Elementor-Website. Grund dafür könnte die Anzahl der DOM-Elemente sein, welche in diesem Fall 416 beträgt und damit deutlich näher an der originalen Website liegt, als es bei der Elementor-Website der Fall ist.

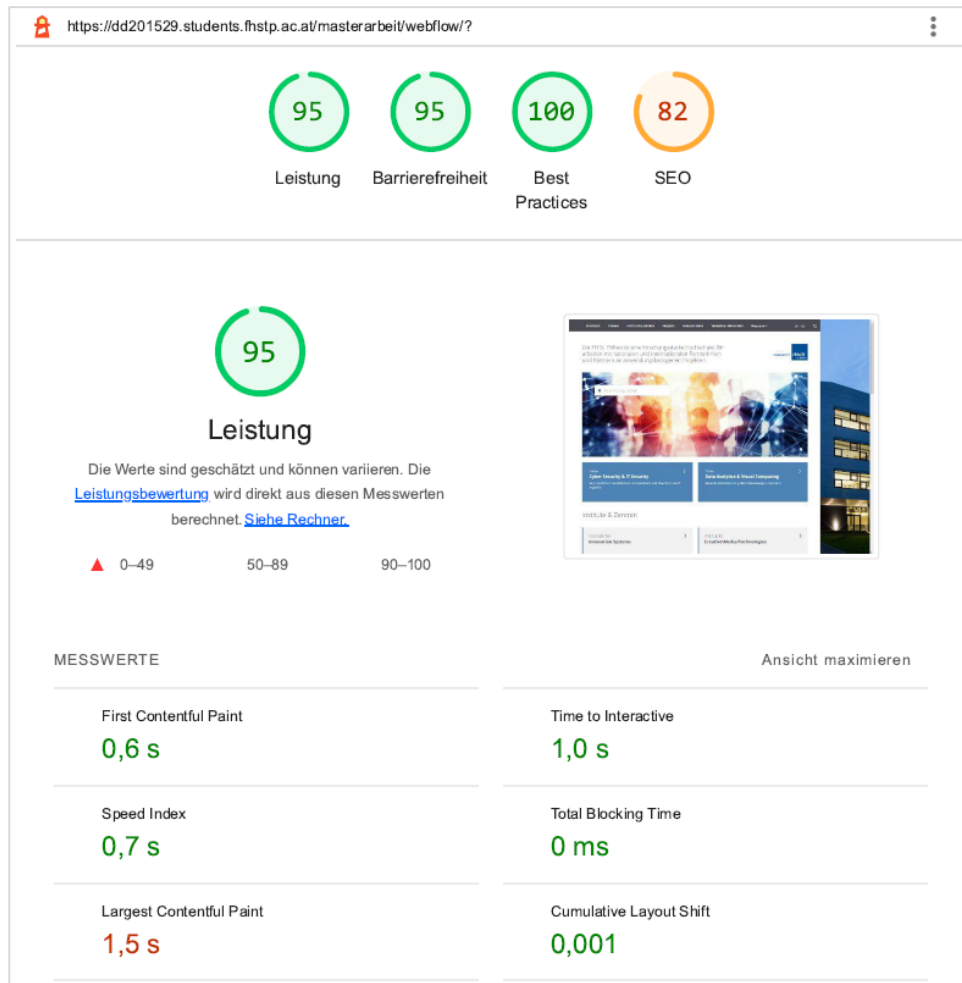


Abbildung 33. Ergebnisse Qualitätsmessung Webflow (eigener Screenshot)

In den Kategorien Barrierefreiheit und Best Practices schneidet die Webflow-Website gleich gut wie die Elementor-Website ab. Es ist also davon auszugehen, dass mit beiden Plattformen bereits ohne zusätzliches Einwirken ein gewisses Maß an Barrierefreiheit mitgeliefert wird.

In der Kategorie SEO erreicht die Webflow-Website 82 Punkte. Die fehlende Meta-Beschreibung wird hier ebenfalls negativ angemerkt. Die entsprechende Einstellung lässt sich in Webflow jedoch problemlos vornehmen.

## 8 Fazit

Im ersten Teil dieser Masterarbeit wurden mittels einer Literaturrecherche die Themen Grundlagen des Webdesigns und der Webentwicklung, Web-Content-Management, der Low-Code / No-Code Programmierstil und No-Code-Plattformen erläutert. Im zweiten Teil wurde die empirische Untersuchung mitsamt eines Vergleichs von vier ausgewählten No-Code-Plattformen sowie einem Experiment der Webdesignumsetzung mit den zwei funktionsreichsten No-Code-Plattformen durchgeführt und die dadurch erarbeiteten Ergebnisse präsentiert. Diese Vorarbeit wurde zur Beantwortung der folgenden Forschungsfragen:

Inwiefern eignen sich No-Code-Entwicklungsplattformen zur Umsetzung eines modernen, responsiven Webdesigns?

- Welche Einschränkungen treten während der Verwendung einer No-Code-Entwicklungsplattform bei der Umsetzung eines modernen, responsiven Webdesigns auf?

durchgeführt, welche im Fazit beantwortet werden und dadurch die zu Beginn der Arbeit aufgestellte Hypothese:

Wird eine No-Code-Plattform für die Umsetzung eines modernen, responsiven Webdesigns eingesetzt, so werden Einschränkungen auftreten, welche ausschließlich mit eigens geschriebenem Code behoben werden können.

entweder verifiziert oder falsifiziert.

Die Forschungsfragen werden im nachfolgenden Unterkapitel 8.1. beantwortet. Im Unterkapitel 8.2 folgt anschließend eine kritische Betrachtung der eingesetzten Forschungsmethode sowie eine Erläuterung von angefallenen Limitierungen im Laufe der Arbeit und ein Ausblick auf weitere Forschungsmöglichkeiten innerhalb dieses Themengebiets.

## 8.1 Beantwortung der Forschungsfragen

Dieses Unterkapitel widmet sich der Beantwortung der Forschungsfragen. Die Frage „Inwiefern eignen sich No-Code-Entwicklungsplattformen zur Umsetzung eines modernen, responsiven Webdesigns?“ lässt sich anhand der Ergebnisse des durchgeführten Vergleichs und des Experiments beantworten. Dementsprechend eignen sich No-Code-Entwicklungsplattformen zur 87% bis 98% detailgetreuen Umsetzung des in dieser Arbeit ausgewählten modernen, responsiven Webdesigns, je nach gewähltem No-Code-System.

Die individuell ausgewählte No-Code-Plattform trägt maßgeblich dazu bei, inwieweit ein modernes, responsives Webdesign umgesetzt werden kann. Das Experiment hat gezeigt, dass das Layout der Website zu 93% bis 100% umgesetzt werden kann. Diese gute Bewertung begründet sich darin, dass alle untersuchten Plattformen Layoutwerkzeuge basierend auf dem CSS Flexible Box Layout, kurz Flexbox, bereitstellen, mit dem die automatische Anordnung von responsiven Elementen in Abhängigkeit von der Größe des Viewports ermöglicht wird. Drei der vier verglichenen Plattformen unterstützen darüber hinaus CSS Grid zur Erstellung von komplexeren Layouts. Die Möglichkeit der Anpassung der Breite und Höhe von Elementen, sowie der Innen- und Außenabstände in allen verglichenen Systemen trägt ebenso zur pixelgenauen Umsetzung eines Webdesigns bei.

Auch die Typografie einer Website lässt sich mit den untersuchten No-Code-Plattformen zu 100% nachbilden. Die umfassenden typografische Einstellungsmöglichkeiten der Systeme machen den Einsatz von eigenem HTML- und CSS-Code zur Anpassung von Überschriften und Texten hinfällig.

Ebenfalls eignen sich No-Code-Plattformen für die Einbindung diverser Medien, wie Bilder, Videos, Audiosequenzen oder SVG-Grafiken in Websites. Bilder und Videos lassen sich dabei in allen untersuchten Systemen zudem als Hintergrundmedien einsetzen. Ein großer Vorteil dieser Systeme besteht darin, dass zahlreiche vorgefertigte Inhaltselemente einsatzbereit zur Verfügung gestellt werden und nicht erst von Grund auf gecodet und gestylt werden müssen. Alle in dieser Untersuchung behandelten Systeme stellen außerdem vorgefertigte Layouts, vorgefertigte Abschnitte mit Elementen und sogar vorgefertigte, komplette Websites bereit, was die Websiteerstellung nochmals beschleunigt.

Obwohl ein Vergleich der Umsetzungsdauer einer manuell entwickelten Website und einer mittels No-Code-Plattform erstellten Website nicht Bestandteil dieser Masterarbeit war, so ist durch die Webdesignumsetzung im Rahmen der

empirischen Untersuchung durchaus der Eindruck entstanden, dass die erstellten Websites in vergleichsweise kurzer Zeit fertiggestellt werden konnten und dabei dem Original hinsichtlich der Qualität in nichts nachstehen.

Die hohe Qualität der erstellten Websites konnte mittels Google Lighthouse Messung bestätigt werden. Beide Websites erreichten in sämtlichen Bereichen eine Wertung über 90 und Ladezeiten unter zwei Sekunden, was wiederum für die Erzeugung von qualitativem Code der No-Code-Systeme und die Einhaltung von Best Practices von Seiten der No-Code-Anbieter\*innen spricht.

Die Forschungsfrage „Welche Einschränkungen treten während der Verwendung einer No-Code-Entwicklungsplattform bei der Umsetzung eines modernen, responsiven Webdesigns auf?“ kann durch die Erkenntnisse des Experiments ebenfalls beantwortet werden. Denn während der Webdesignumsetzung sind Einschränkungen vor allem in der Kategorie „Animationen, Effekte und Interaktionen“ aufgetreten, welche zur Behebung eigenen Code erfordern. Diverse Hover-Effekte und Animationen von Elementen konnten ausschließlich mit einer der getesteten No-Code-Plattformen zufriedenstellend umgesetzt werden. Hierbei sollte auch erwähnt werden, dass nicht alle der verglichenen Anbieter\*innen den Fokus auf diese Funktionalitäten legen.

Eine weitere Einschränkung der No-Code-Plattformen konnte bei der Nachbildung eines Suchformularfeldes, welches erweiterte Funktionalitäten durch den Einsatz einer JavaScript-Bibliothek aufweist, festgestellt werden. Beide Systeme stellen zwar Formularfelder zur Verfügung, diese sind jedoch in ihren Funktionen begrenzt. Es kann also davon ausgegangen werden, dass individuelle Funktionen und Lösungen immer ein gewisses Maß an eigens geschriebenen Code erfordern, da die Anbieter\*innen der No-Code-Systeme nicht jede Funktionalität abdecken.

Die zu Beginn aufgestellte These kann somit verifiziert werden. Denn die Umsetzung einer Website mittels No-Code-Plattform konnte vollständig durchgeführt werden, es sind jedoch Einschränkungen aufgetreten, welche mit den Bordmitteln der jeweiligen Plattform nicht gelöst werden konnten und zusätzlichen, eigenen Code erfordern.

Dennoch kann der Einsatz von No-Code-Plattformen zur Websiteerstellung vom Forschenden grundsätzlich empfohlen werden, da sich die Einschränkungen vorwiegend auf Details beziehen, welche die Hauptfunktionen der Website nicht negativ beeinflussen. Zusätzlich bieten alle untersuchten Systeme die Möglichkeit

den Funktionsumfang mit eigenem Code zu erweitern und die genannten Einschränkungen zu beheben.

## 8.2 Methodenkritik, Limitierung und Ausblick

Die Bewertung der in dieser Arbeit verglichenen No-Code-Plattformen wurde in Form der Selbstevaluierung durch den Forschenden durchgeführt. Aus diesem Grund ist eine eventuelle Verzerrung der Ergebnisse nicht vollständig auszuschließen. Der Forschende hat sich zwar durch Einarbeiten in die verwendeten Systeme eine gewisse Expertise angeeignet, fortgeschrittenere Nutzer\*innen dieser Systeme könnten allerdings andere Ergebnisse hinsichtlich der Umsetzbarkeit erzielen. Des Weiteren wurde in dieser Arbeit ressourcenbedingt nur eine limitierte Anzahl an No-Code-Plattformen behandelt. Es wurden vier ausgewählte Plattformen verglichen, von denen die zwei funktionsreichsten Systeme für die weitere Webdesignumsetzung zum Einsatz kamen. Wie es in Kapitel 5.1 (siehe Seite 47) bereits erwähnt wurde, ist der No-Code-Markt sehr groß und dynamisch. Es kann daher nicht ausgeschlossen werden, dass eventuelle No-Code-Tools für die Umsetzung von Websites im Laufe der Online-Recherche, aufgrund der verwendeten Suchbegriffe, übersehen wurden. Ebenfalls ist zu erwähnen, dass die Erkenntnisse dieser Untersuchung zum Zeitpunkt der Durchführung der Arbeit gelten. Da die untersuchten No-Code-Plattformen laufend weiterentwickelt werden, ist es möglich, dass durch zukünftige Aktualisierungen, der Funktionsumfang erweitert und negativ angemerkte Aspekte der Systeme verbessert oder gänzlich behoben werden.

Im Zuge dieser Arbeit wurden vorwiegend die technischen Aspekte von No-Code-Plattformen erarbeitet und erforscht. Die No-Code-Entwicklung steht aufgrund angeblicher Qualifikationslücken der Endnutzer\*innen hin und wieder in Kritik. Es wäre sinnvoll, in zukünftigen Forschungen auch die Usability und User Experience von, mit No-Code-Plattformen entwickelten, Websites zu berücksichtigen und diese mit herkömmlich programmierten Websites zu vergleichen. Abschließend stellt sich die Frage über die Zukunft von No-Code. Es ist davon auszugehen, dass No-Code keine Trenderscheinung bleibt und immer weitere No-Code-Systeme am Markt erscheinen werden. Auch zeichnen sich mittlerweile erste Anwendungen ab, welche mithilfe von Künstlicher Intelligenz den Prozess der Websiteerstellung deutlich vereinfachen. Somit bleibt die Frage offen, ob zukünftig überhaupt noch Programmierkenntnisse zur Erstellung einer Website notwendig sein werden.

# Literaturverzeichnis

- Atkins, B. (24. November 2020). The Most Disruptive Trend Of 2021: No Code / Low Code. Abgerufen am 20. Oktober 2021 von <https://www.forbes.com/sites/betsyatkins/2020/11/24/the-most-disruptive-trend-of-2021-no-code--low-code/>
- Augsten, S. (8. März 2019). Was ist Low-Code Development? Abgerufen am 23. Juli 2022 von <https://www.dev-insider.de/was-ist-low-code-development-a-794952/>
- Barker, D. (2016). *Web content management: Systems, features, and best practices* (First edition). Beijing ; Boston: O'Reilly.
- Bendel, O. (13. Juli 2021). Definition: Disruptive Technologien. Abgerufen am 25. Oktober 2021 von <https://wirtschaftslexikon.gabler.de/definition/disruptive-technologien-54194/version-384599>
- Böhringer, J., Bühler, P., Schlaich, P., & Sinner, D. (2014). *Kompendium der Mediengestaltung. 4: Medienproduktion digital* (6., vollst. überarb. u. erw. Aufl.). Berlin Heidelberg: Springer Vieweg.
- Bühler, P., Schlaich, P., & Sinner, D. (2017). *Webdesign: Interfacedesign - Screendesign - Mobiles Webdesign*. Berlin: Springer Vieweg.
- Dewanto, L., & Klein, M. (14. August 2018). Low-Code == Low Quality? Abgerufen am 9. Juli 2022 von <https://www.heise.de/ratgeber/Low-Code-Low-Quality-4134288.html>
- Döring, N., & Bortz, J. (2016). *Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften* (5. vollständig überarbeitete, aktualisierte und erweiterte Auflage). Berlin Heidelberg: Springer. <https://doi.org/10.1007/978-3-642-41089-5>
- Eisenmenger, R. (2019). *WordPress 5 Das umfassende Handbuch*. Bonn: Rheinwerk Verlag.
- ElBatanony, A., & Succi, G. (2021). Towards the no-code era: A vision and plan for the future of software development. *Proceedings of the 1st ACM SIGPLAN International Workshop on Beyond Code: No Code*, 29–35. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3486949.3486965>
- Equation Research. (1. Juli 2011). Wie lange sind Sie bereit zu warten, bis eine mobile Website auf Ihrem Smartphone vollständig geladen hat? Abgerufen am 15. Juli 2022 von

<https://de.statista.com/statistik/daten/studie/202650/umfrage/wartezeit-bis-zum-verlassen-einer-mobilen-website/>

- Etemadian, R. (29. Juli 2022). Was ist ein Citizen Developer? Abgerufen am 27. Juli 2022 von <https://www.computerwoche.de/a/was-ist-ein-citizen-developer,3550357>
- Gartner. (2022). Definition of Citizen Developer—Gartner Information Technology Glossary. Abgerufen am 25. Oktober 2021 von <https://www.gartner.com/en/information-technology/glossary/citizen-developer>
- Google. (2022a). Lighthouse performance scoring. Abgerufen am 3. September 2022 von <https://web.dev/performance-scoring/>
- Google. (2022b). Performance audits. Abgerufen am 4. September 2022 von <https://web.dev/lighthouse-performance/>
- Hahn, M. (2020). *Webdesign: Das Handbuch zur Webgestaltung* (3., aktualisierte und überarbeitete Auflage). Bonn: Rheinwerk Verlag.
- Koop, H. J., Jäckel, K. K., & Offern, A. L. van. (2001). *Erfolgsfaktor Content Management: Vom Web Content bis zum Knowledge Management*. Braunschweig Wiesbaden: Vieweg.
- Korthaus, C. (2020). *Grundkurs Typografie und Layout Für Ausbildung und Praxis*. (6. Auflage). Bonn: Rheinwerk Verlag.
- Krol, K. (2019). *WordPress 5 Complete — Seventh Edition*. Packt Publishing.
- Liptak, J., & Horwitz, L. (2022). What is citizen development, and what is a citizen developer? Abgerufen am 27. Juli 2022 von <https://www.techtarget.com/searchsoftwarequality/definition/citizen-development>
- Luo, Y., Liang, P., Wang, C., Shahin, M., & Zhan, J. (2021). Characteristics and Challenges of Low-Code Development: The Practitioners' Perspective. *Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 1–11. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3475716.3475782>
- Mayring, P. (2015). *Qualitative Inhaltsanalyse: Grundlagen und Techniken* (12., überarb. Aufl). Weinheim Basel: Beltz.
- Nielsen, J. (30. September 1996). Accessible Design for Users With Disabilities. Abgerufen am 14. Juli 2022 von <https://www.nngroup.com/articles/accessible-design-for-users-with-disabilities/>
- Nix, M. (2005). *Web Content Management: CMS verstehen und auswählen* (1. Auflage). Frankfurt am Main: Software-und-Support-Verlag.

- Pratt, M. K. (2021, März 1). What are Low-Code and No-Code Development Platforms? Abgerufen am 20. Oktober 2021 von <https://searchsoftwarequality.techtarget.com/definition/low-code-no-code-development-platform>
- Prohl. (25. April 2022). WordPress Page Builder: Fluch oder Segen? Abgerufen am 22. Juli 2022 von <https://raidboxes.io/blog/webdesign-development/wordpress-pagebuilder-nachteile/>
- Richardson, C., & Rymer, J. (9. Juni 2014). New Development Platforms Emerge For Customer-Facing Applications. Abgerufen am 23. Juli 2022 von <https://www.forrester.com/report/New-Development-Platforms-Emerge-For-CustomerFacing-Applications/RES113411>
- Richardson, C., & Rymer, J. (15. Jänner 2016). Vendor Landscape: The Fractured, Fertile Terrain Of Low-Code Application Platforms. Abgerufen am 7. August 2022 von <https://www.forrester.com/report/vendor-landscape-the-fractured-fertile-terrain-of-lowcode-application-platforms/RES122549>
- Rieder, S. (6. Juli 2022). Page Builder für WordPress. Abgerufen am 21. Juli 2022 von <https://www.webtimiser.de/page-builder-wordpress/>
- Riggert, W. (2009). *ECM - Enterprise Content Management: Konzepte und Techniken rund um Dokumente* (1. Auflage). Wiesbaden: Vieweg + Teubner.
- Rimol, M. (14. Juni 2021). Gartner Says the Majority of Technology Products and Services Will Be Built by Professionals Outside of IT by 2024. Abgerufen am 29. Juni 2022 von <https://www.gartner.com/en/newsroom/press-releases/2021-06-10-gartner-says-the-majority-of-technology-products-and-services-will-be-built-by-professionals-outside-of-it-by-2024>
- Rohles, B. (2017). *Grundkurs gutes Webdesign: Alles, was Sie über Gestaltung im Web wissen müssen* (2., aktualisierte und erweiterte Auflage). Bonn: Rheinwerk Verlag.
- Sabin-Wilson, L. (2019). *Wordpress all-in-one for dummies, 4th edition* (4th edition). Indianapolis, IN: John Wiley and Sons.
- Sachs, A. (2021). Wie No-Code die Entwicklung von digitalen Geschäftsmodellen deutlich zugänglicher macht. In S. Detscher (Hrsg.), *Digitales Management und Marketing: So nutzen Unternehmen die Marktchancen der Digitalisierung* (S. 29–38). Wiesbaden: Springer Fachmedien. [https://doi.org/10.1007/978-3-658-33731-5\\_3](https://doi.org/10.1007/978-3-658-33731-5_3)
- Sahay, A., Indamutsa, A., Di Ruscio, D., & Pierantonio, A. (2020). Supporting the understanding and comparison of low-code development platforms. *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 171–178. <https://doi.org/10.1109/SEAA51224.2020.00036>

- Sens, B. (2018). *Suchmaschinenoptimierung: Erste Schritte und Checklisten für bessere Google-Positionen*. Wiesbaden: Springer Gabler. <https://doi.org/10.1007/978-3-658-21524-8>
- Spörrer, S. (2019). *Content Management Systeme: Begriffsstruktur und Praxisbeispiel*. Wiesbaden: Springer Gabler. <https://doi.org/10.1007/978-3-658-24351-7>
- Steyer, R. (2016). *WordPress: Einführung in das Content Management System*. Wiesbaden: Springer Vieweg.
- Theis, T. (2021). *Einstieg in PHP 8 und MySQL: Ideal für Programmieranfänger*.
- W3Techs. (2022). Usage Statistics and Market Share of Content Management Systems, July 2022. Abgerufen am 20. Juli 2022 von [https://w3techs.com/technologies/overview/content\\_management](https://w3techs.com/technologies/overview/content_management)
- Wirtschaftskammer Österreich. (September 2021). Österreich—Fachkräftemangel nach Branchen 2021. Abgerufen am 2. Juli 2022 von <https://de.statista.com/statistik/daten/studie/914110/umfrage/staerke-des-fachkraeftemangels-in-oesterreich-nach-branchen/>
- Wolf, J. (2019). *HTML5 und CSS3 Das umfassende Handbuch* (3. Auflage). Bonn: Rheinwerk Verlag.
- Wong, J., & Vincent, P. (8. Februar 2022). Quick Answer: What Is the Difference Between No-Code and Low-Code Development Tools? Abgerufen am 23. Juli 2022 von <https://www.gartner.com/doc/reprints?id=1-27L9CRZ0&ct=211004&st=sb>
- WordPress. (13. Mai 2020). Full site editing. Abgerufen am 21. Juli 2022 von <https://make.wordpress.org/design/handbook/focuses/full-site-editing/>
- WordPress. (2022a). Democratize Publishing. Abgerufen am 20. Juli 2022 von <https://wordpress.org/about/>
- WordPress. (2022b). Hosting Requirements for WordPress. Abgerufen am 20. Juli 2022 von <https://wordpress.org/about/requirements/>

# Abbildungsverzeichnis

Abbildung 1. Trend der Suchbegriffe Low-Code No-Code (eigener Screenshot) . 2	
Abbildung 2. Anatomie einer Website (in Anlehnung an Hahn, 2020, S.271) .....	8
Abbildung 3. RGB und CMYK Farbraum (eigene Darstellung).....	13
Abbildung 4. HTML-Grundgerüst (eigene Darstellung).....	19
Abbildung 5. Content-Life-Cycle (in Anlehnung an Koop & Jäckel, 2001, S.15) .	28
Abbildung 6. Statischer Zugriff (Bühler, Schlaich, Sinner, 2022. S.5).....	32
Abbildung 7. Dynamischer Zugriff (Bühler, Schlaich, Sinner, 2022, S.6).....	33
Abbildung 8. User-Interface des Block-Editors (WordPress, 2022) .....	39
Abbildung 9. Kategorienbildung mit MAXQDA 2022 (eigener Screenshot) .....	59
Abbildung 10. Nachzubildende Elemente der Website (eigener Screenshot).....	61
Abbildung 11. Ergebnisse Kategorie Animationen/Effekte (eigene Darstellung) .	70
Abbildung 12. Ergebnisse Kategorie Layout (eigene Darstellung).....	71
Abbildung 13. Ergebnisse Kategorie Typografie (eigene Darstellung).....	72
Abbildung 14. Ergebnisse Kategorie Responsive (eigene Darstellung).....	73
Abbildung 15. Ergebnisse Kategorie Design-Workflow (eigene Darstellung).....	74
Abbildung 16. Ergebnisse Kategorie CMS-Funktionen (eigene Darstellung).....	75
Abbildung 17. Ergebnisse Kategorie Inhaltselemente (eigene Darstellung) .....	76
Abbildung 18. Ergebnisse Kategorie Medien (eigene Darstellung).....	77
Abbildung 19. Ergebnisse Kategorie Hintergrundmedien (eigene Darstellung) ..	77
Abbildung 20. Ergebnisse Kategorie Einbettungen (eigene Darstellung) .....	78
Abbildung 21. Ergebnisse Kategorie Integrationen (eigene Darstellung).....	78
Abbildung 22. Ergebnisse Kategorie SEO & Accessibility (eigene Darstellung) .	79
Abbildung 23. Ergebnisse Kategorie sonstige Funktionen (eigene Darstellung) .	80
Abbildung 24. Elemente Header & Aside (eigener Screenshot) .....	81

Abbildung 25. Elemente Mobile Navigation (eigener Screenshot).....	81
Abbildung 26. Elemente Inhaltsbereich 1 (eigener Screenshot).....	82
Abbildung 27. Elemente Inhaltsbereich 2 (eigener Screenshot).....	82
Abbildung 28. Elemente Footer (eigener Screenshot).....	83
Abbildung 29. Dauer der Webdesignumsetzung (eigene Darstellung) .....	87
Abbildung 30. Ergebnisse der Evaluation von Elementor (eigene Darstellung) ..	88
Abbildung 31. Ergebnisse der Evaluation von Webflow (eigene Darstellung).....	90
Abbildung 32. Ergebnisse Qualitätsmessung Elementor (eigener Screenshot) ..	92
Abbildung 33. Ergebnisse Qualitätsmessung Webflow (eigener Screenshot) ....	95

# Tabellenverzeichnis

Tabelle 1. Wichtige ARIA Roles (in Anlehnung an Rohles, 2017, S.54) .....	23
Tabelle 2. Recherchierte No-Code-Plattformen .....	54
Tabelle 3. Ausgewählte No-Code-Plattformen für den Vergleich.....	55
Tabelle 4. Bewertungsschema der Evaluation .....	60
Tabelle 5. Kriterienkatalog des Vergleichs .....	63
Tabelle 6. Kriterienkatalog der Webdesignumsetzung .....	83

# Anhang

## A. Ergebnisse Dokumentenanalyse

Duda	
Quelle der Dokumente	<a href="https://www.duda.co/features">https://www.duda.co/features</a>
Zweck der Dokumente	Information, Marketing
Datum des Aufrufs	15.08.2022
Inhalt der Dokumente	<p><b>Design-Features</b></p> <p><i>Animationen, Effekte und Interaktionen</i></p> <ul style="list-style-type: none"><li>• Interaktionen und Trigger</li><li>• Farbüberlagerungen</li></ul> <p><i>Layout</i></p> <ul style="list-style-type: none"><li>• CSS-Grid</li><li>• CSS-Flexbox</li></ul> <p><i>Typografie</i></p> <ul style="list-style-type: none"><li>• Webfonts</li></ul> <p><i>Responsiveness</i></p> <ul style="list-style-type: none"><li>• Responsives Layout</li><li>• Ein- und Ausblenden von Elementen</li><li>• Responsive Einstellungsmöglichkeiten</li><li>• Responsive Bilder</li></ul> <p><i>Design-Workflow</i></p> <ul style="list-style-type: none"><li>• Wiederverwendbare Elemente &amp; Abschnitte</li><li>• Drag &amp; Drop Editor</li><li>• Copy-Paste</li><li>• Icon-Bibliothek</li><li>• Globale Schriftarten</li><li>• Globale Farben</li><li>• Vorgefertigte Abschnitte</li><li>• Vorgefertigte Websites</li></ul> <p><i>CMS-Features</i></p> <ul style="list-style-type: none"><li>• <i>Dynamische Inhaltstemplates</i></li><li>• <i>Formulareinträge</i></li><li>• <i>Medienverwaltung</i></li><li>• <i>Dynamische Inhalte</i></li><li>• <i>Inhalts-Feldtypen</i></li></ul> <p><b>Templates und Komponenten</b></p> <p><i>Inhaltselemente</i></p> <ul style="list-style-type: none"><li>• Lightbox &amp; Bildergalerien</li><li>• Text-Editor</li></ul>

	<ul style="list-style-type: none"> <li>• Slider</li> <li>• Formulare</li> <li>• Trenner</li> <li>• Listen</li> <li>• Navigationsmenü</li> </ul> <p><i>Medien</i></p> <ul style="list-style-type: none"> <li>• Videos</li> </ul> <p><i>Hintergrundelemente</i></p> <ul style="list-style-type: none"> <li>• Hintergrundvideos</li> <li>• Hintergrundbilder</li> <li>• Hintergrund-Slideshows</li> </ul> <p><i>Einbettungen</i></p> <ul style="list-style-type: none"> <li>• Google Maps</li> <li>• HTML-Embed</li> <li>• YouTube-Video</li> <li>• Mixcloud-Player</li> <li>• Dailymotion-Video</li> <li>• Vimeo-Video</li> <li>• Social-Icons &amp; Inhalte</li> <li>• SoundCloud-Player</li> </ul> <p><b>SEO</b></p> <ul style="list-style-type: none"> <li>• Open Graph Einstellungen</li> <li>• Meta-Titel und Meta-Beschreibung</li> <li>• XML-Sitemap</li> <li>• Alt-Tags für Bilder</li> </ul> <p><b>Integrationen</b></p> <ul style="list-style-type: none"> <li>• Disqus</li> <li>• PayPal</li> <li>• Yelp</li> <li>• Open Table</li> <li>• AirTable</li> <li>• MailChimp</li> <li>• TripAdvisor</li> <li>• Usercentrics</li> <li>• Google Analytics</li> </ul> <p><b>Sonstige Features</b></p> <ul style="list-style-type: none"> <li>• Popup-Builder</li> <li>• API</li> <li>• Benutzerrollen</li> <li>• Benutzerdefinierter Code</li> <li>• Widget-Builder</li> <li>• Backups und Versionierung</li> <li>• Favicon</li> <li>• 404 Seite</li> </ul>
--	--

<b>Editor X</b>	
<b>Quelle der Dokumente</b>	<a href="https://de.editorx.com/features/">https://de.editorx.com/features/</a>
<b>Zweck der Dokumente</b>	Information, Marketing
<b>Datum des Aufrufs</b>	15.08.2022
<b>Inhalt der Dokumente</b>	<p><b>Design-Features</b></p> <p><i>Animationen, Effekte und Interaktionen</i></p> <ul style="list-style-type: none"> <li>• Scrolling Effekte</li> <li>• Eingangsanimationen</li> <li>• Ausgangsanimationen</li> <li>• Interaktionen und Trigger</li> </ul> <p><i>Layout</i></p> <ul style="list-style-type: none"> <li>• CSS-Grid</li> <li>• CSS-Flexbox</li> <li>• Margin &amp; Padding</li> <li>• Unterschiedliche Maßeinheiten</li> <li>• Fixe Positionierung</li> <li>• Höhenanpassung</li> </ul> <p><i>Typografie</i></p> <ul style="list-style-type: none"> <li>• Benutzerdefinierte Schriftarten</li> </ul> <p><i>Responsiveness</i></p> <ul style="list-style-type: none"> <li>• Responsives Bilder</li> <li>• Responsive Layout</li> <li>• Responsive Einstellungsmöglichkeiten</li> </ul> <p><i>Design-Workflow</i></p> <ul style="list-style-type: none"> <li>• Wiederverwendbare Elemente &amp; Abschnitte</li> <li>• Copy-Paste</li> <li>• Icon-Bibliothek</li> <li>• Vorgefertigte Layouts</li> <li>• Vorgefertigte Abschnitte</li> </ul> <p><i>CMS-Features</i></p> <ul style="list-style-type: none"> <li>• <i>Dynamische Inhaltstemplates</i></li> <li>• <i>Dynamische Inhalte</i></li> <li>• <i>CSV-Import</i></li> <li>• <i>CSV-Export</i></li> </ul> <p><b>Templates und Komponenten</b></p> <p><i>Inhaltselemente</i></p> <ul style="list-style-type: none"> <li>• Lightbox &amp; Bildergalerien</li> <li>• Ankerlinks</li> <li>• Formulare</li> <li>• Navigationsmenü</li> </ul> <p><i>Medien</i></p> <ul style="list-style-type: none"> <li>• Videos</li> </ul>

	<ul style="list-style-type: none"> <li>• SVG-Icons</li> </ul> <p><b>SEO</b></p> <ul style="list-style-type: none"> <li>• Open Graph Einstellungen</li> <li>• Meta-Titel und Meta-Beschreibung</li> <li>• XML-Sitemap</li> </ul> <p><b>Integrationen</b></p> <ul style="list-style-type: none"> <li>• HotJar</li> <li>• Facebook SDK</li> <li>• Google Analytics</li> </ul> <p><b>Sonstige Features</b></p> <ul style="list-style-type: none"> <li>• API</li> <li>• Benutzerdefinierter Code</li> </ul>
--	--

Elementor	
<b>Quelle der Dokumente</b>	<a href="https://elementor.com/features/">https://elementor.com/features/</a> <a href="https://de.wordpress.org/plugins/elementor/">https://de.wordpress.org/plugins/elementor/</a>
<b>Zweck der Dokumente</b>	Information, Marketing
<b>Datum des Aufrufs</b>	15.08.2022
<b>Inhalt der Dokumente</b>	<p><b>Design-Features</b></p> <p><i>Animationen, Effekte und Interaktionen</i></p> <ul style="list-style-type: none"> <li>• Motion Effekte</li> <li>• Box-Schatten</li> <li>• Masken</li> <li>• Blend-Modes</li> <li>• Ausgangsanimationen</li> <li>• Parallax Effekte</li> <li>• Seitenübergangsanimationen</li> <li>• Hover-Animationen</li> <li>• Scrolling Effekte</li> <li>• CSS-Transform</li> <li>• Eingangsanimationen</li> <li>• 3D Transformationen</li> <li>• CSS-Filter</li> <li>• Animierte Überschriften</li> </ul> <p><i>Layout</i></p> <ul style="list-style-type: none"> <li>• Höhenanpassung</li> <li>• Breitenanpassung</li> <li>• Verschachtelung</li> <li>• Inline-Ausrichtung</li> <li>• Margin &amp; Padding</li> <li>• Z-Index</li> <li>• Absolute Positionierung</li> <li>• CSS-Flexbox</li> </ul> <p><i>Typografie</i></p> <ul style="list-style-type: none"> <li>• Textschatten</li> <li>• Initialen</li> <li>• Benutzerdefinierte Schriftarten</li> </ul>

- Text Stroke
- Schriftdekoration
- Schrifttransformation
- Webfonts
- Zeichenabstand
- Zeilenhöhe
- Schriftart
- Schriftgewicht
- Schriftgröße

#### *Responsiveness*

- Responsive Einstellungsmöglichkeiten
- Responsives Layout
- Ein- und Ausblenden von Elementen
- Benutzerdefinierte Breakpoints

#### *Design-Workflow*

- Icon-Bibliothek
- Globale Schriftarten
- Drag & Drop Editor
- Vorgefertigte Websites
- Wiederverwendbare Elemente & Abschnitte
- Globale Farben
- Tastaturkürzel
- Navigator

#### *CMS-Features*

- *Dynamische Inhaltstemplates*
- *Dynamische Inhalte*

#### **Templates und Komponenten**

##### *Inhaltselemente*

- Tabs
- Slider
- Formulare
- Zitatblock
- Countdown
- Lottie-Animation
- Preistabelle
- Inhaltsverzeichnis
- Flip Box
- Shape-Divider
- Überschriften
- Trenner
- Abstandshalter
- Sterne-Bewertungen
- Akkordeons
- Ankerlinks
- Hinweismeldungen
- Listen
- Icon-Box
- Fortschrittsbalken
- Counter
- Buttons
- Testimonials
- Bild-Box
- Lightbox & Bildergalerien
- Text-Pfad
- Text-Editor
- Lightbox & Bildergalerien

	<ul style="list-style-type: none"> <li>• Ankerlinks</li> <li>• Formulare</li> <li>• Navigationsmenü</li> </ul> <p><i>Medien</i></p> <ul style="list-style-type: none"> <li>• Bilder</li> <li>• Videos</li> <li>• SVG-Icons</li> </ul> <p><i>Hintergrundelemente</i></p> <ul style="list-style-type: none"> <li>• Hintergrundvideos</li> <li>• Hintergrundverläufe</li> <li>• Hintergrund-Slideshows</li> <li>• Hintergrund-Überlagerung</li> </ul> <p><i>Einbettungen</i></p> <ul style="list-style-type: none"> <li>• HTML-Embed</li> <li>• Social-Icons &amp; Inhalte</li> <li>• SoundCloud-Player</li> </ul> <p><b>SEO</b></p> <ul style="list-style-type: none"> <li>• Anpassbare HTML-Tags</li> </ul> <p><b>Integrationen</b></p> <ul style="list-style-type: none"> <li>• Zapier</li> <li>• MailChimp</li> <li>• GetResponse</li> <li>• Discord</li> <li>• ReCaptcha</li> <li>• Drip</li> <li>• ActiveCampaign</li> <li>• HubSpot</li> <li>• ConvertKit</li> <li>• MailerLite</li> <li>• Facebook SDK</li> <li>• Slack</li> </ul> <p><b>Sonstige Features</b></p> <ul style="list-style-type: none"> <li>• Benutzerrollen</li> <li>• Benutzerdefinierter Code</li> <li>• Backups und Versionierung</li> <li>• Popup-Builder</li> <li>• Wartungsmodus</li> </ul>
--	---

<b>Webflow</b>	
<b>Quelle der Dokumente</b>	<a href="https://webflow.com/features">https://webflow.com/features</a>
<b>Zweck der Dokumente</b>	Information, Marketing
<b>Datum des Aufrufs</b>	15.08.2022
<b>Inhalt der Dokumente</b>	<p><b>Design-Features</b></p> <p><i>Animationen, Effekte und Interaktionen</i></p> <ul style="list-style-type: none"> <li>• Interaktionen und Trigger</li> <li>• 3D Transformationen</li> <li>• CSS-Filter</li> <li>• CSS-Transform</li> </ul> <p><i>Layout</i></p> <ul style="list-style-type: none"> <li>• Z-Index</li> <li>• CSS-Grid</li> <li>• CSS-Flexbox</li> </ul> <p><i>Typografie</i></p> <ul style="list-style-type: none"> <li>• Webfonts</li> </ul> <p><i>Responsiveness</i></p> <ul style="list-style-type: none"> <li>• Responsives Layout</li> <li>• Responsive Einstellungsmöglichkeiten</li> <li>• Responsive Bilder</li> </ul> <p><i>Design-Workflow</i></p> <ul style="list-style-type: none"> <li>• Wiederverwendbare Elemente &amp; Abschnitte</li> <li>• Vorgefertigte Abschnitte</li> <li>• Globale Farben</li> <li>• Vorgefertigte Websites</li> </ul> <p><i>CMS-Features</i></p> <ul style="list-style-type: none"> <li>• Blog-System</li> <li>• CSV-Export</li> <li>• Dynamisches Filtern und Sortieren</li> <li>• Dynamische Inhaltstemplates</li> <li>• Formulareinträge</li> <li>• Dynamische Inhalte</li> <li>• Inhalts-Feldtypen</li> </ul> <p><b>Templates und Komponenten</b></p> <p><i>Inhaltselemente</i></p> <ul style="list-style-type: none"> <li>• Lightbox &amp; Bildergalerien</li> <li>• Text-Editor</li> <li>• Navigationsmenü</li> <li>• Slider</li> <li>• Formulare</li> </ul> <p><i>Hintergrundelemente</i></p> <ul style="list-style-type: none"> <li>• Hintergrundvideos</li> </ul>

	<p><b>Einbettungen</b></p> <ul style="list-style-type: none"><li>• Google Maps</li><li>• HTML-Embed</li><li>• Social-Icons &amp; Inhalte</li></ul> <p><b>SEO</b></p> <ul style="list-style-type: none"><li>• Dynamische SEO-Tags und Open Graph Einstellungen</li><li>• Open Graph Einstellungen</li><li>• Meta-Titel und Meta-Beschreibung</li><li>• XML-Sitemap</li><li>• Alt-Tags für Bilder</li></ul> <p><b>Integrationen</b></p> <ul style="list-style-type: none"><li>• Zapier</li><li>• MailChimp</li><li>• Google Optimize</li><li>• Google Analytics</li></ul> <p><b>Sonstige Features</b></p> <ul style="list-style-type: none"><li>• API</li><li>• Benutzerrollen</li><li>• Suchfunktion</li><li>• Accessibility-Audit</li><li>• Benutzerdefinierter Code</li><li>• Code-Export</li><li>• Backups und Versionierung</li><li>• Passwortschutz</li><li>• Favicon</li><li>• 404 Seite</li></ul>
--	---

## B. Evaluierungsergebnisse Vergleich

Liste der Kriterien	Beschreibung	Duda	Editor X	Elementor	Webflow
<b>Designspezifische Funktionen</b>					
<i>Animationen, Effekte und Interaktionen</i>					
Motion-Effekte	Auf Mausbewegung reagierende Bewegungseffekte	0	0	2	2
Scrolling-Effekte	Auf Scrolling reagierende Effekte	0	2	2	2
Parallax-Effekte	Parallax-Effekt beim Scrollen	2	2	2	2
Masken	Erstellung von Masken auf Basis verschiedener SVG-Formen	0	0	2	0
Blend-Modes	Hinzufügen von CSS Blend-Modes	0	0	2	2
Farbüberlagerungen	Farbüberlagerungen für Bilder und Videos	2	0	2	2
Animierte Überschriften	Verschiedene animierte Effekte für Überschriften	0	0	2	0
Box-Schatten	Hinzufügen von Box-Schatten auf verschiedene Elementen	2	2	2	2
CSS-Filter	Verschiedene CSS-Filter (Graustufen, Blur, etc.) für Elemente	0	0	2	2
CSS-Transform	CSS Transformationen von Elementen	0	2	2	2
3D Transformationen	3D CSS-Transformationen	0	0	2	2
Hover-Animationen	Auf Hover reagierende Effekte	0	2	2	2
Eingangsanimationen	Eingangsanimationen von Elementen	2	2	2	2
Ausgangsanimationen	Ausgangsanimationen von Elementen	0	0	1	2
Seitenübergangsanimationen	Übergangseffekte beim Wechseln von Webseiten	0	2	2	0
Interaktionen und Trigger	Visueller Interaktionsbuilder	2	0	0	2
<i>Layout</i>					
CSS-Grid	Layoutgestaltung nach den Regeln von CSS-Grid	2	2	0	2
CSS-Flexbox	Layoutgestaltung nach den Regeln von CSS-Flexbox	2	2	2	2
Breitenanpassung	Anpassung der Breite von Elementen	2	2	2	2

Höhenanpassung	Anpassung der Höhe von Elementen	1	2	2	2
Unterschiedliche Maßeinheiten	Verschiedene Maßeinheiten (PX, %, VW, VH)	2	2	2	2
Margin & Padding	Anpassen des Innen- und Außenabstands von Elementen	2	1	2	2
Inline-Ausrichtung	Ausrichtung von Elementen nebeneinander	1	2	2	2
Verschachtelung	Verschachtelung von Containern und Elementen	2	2	2	2
Absolute Positionierung	Absolute Positionierung von Elementen	0	2	2	2
Fixe Positionierung	Fixe Positionierung von Elementen	2	2	2	2
Z-Index	Verschieben von Elementen auf der Z-Achse	0	2	2	2
<i>Typografie</i>					
Schriftart	Auswahl der Schriftfamilie	2	2	2	2
Schriftgröße	Anpassen der Schriftgröße	2	2	2	2
Schriftgewicht	Anpassen des Schriftgewichts	2	2	2	2
Schriftdekoration	Anpassen der Schriftdekoration (Unterstreichen, Überstreichen, Durchstreichen, etc.)	1	1	2	2
Schrifttransformation	Transformation der Schrift (Kapitälchen, Großbuchstaben, etc.)	2	0	2	2
Zeilenhöhe	Anpassen der Zeilenhöhe	2	2	2	2
Zeichenabstand	Anpassen des Zeichenabstands	2	2	2	2
Initialen	Einstellen von Initialen bei Texten	0	0	2	0
Text Stroke	Anpassen der Strichstärke von Schriften	0	0	2	0
Textschatten	Hinzufügen von Textschatten	2	0	2	2
Webfonts	Einbindungsmöglichkeit von Webfonts	2	2	2	2
Benutzerdefinierte Schriftarten	Hinzufügen von benutzerdefinierten Schriften	2	2	2	2
<i>Responsive</i>					
Responsive Bilder	Skalieren und Optimieren von Bildern nach Bildschirmgröße	2	2	2	2
Responsives Layout	Breakpoints für unterschiedliche Bildschirmgrößen	2	2	2	2
Responsive Einstellungsmöglichkeiten	Separate Einstellungsmöglichkeiten für mobile Geräte	1	2	2	2
Benutzerdefinierte Breakpoints	Erstellung von eigenen Umbruchspunkten des Layouts	0	2	2	2

Ein- und Ausblenden von Elementen	Anzeigen oder Verstecken von Elementen je nach Breakpoint	2	2	2	2
<i>Design-Workflow</i>					
Drag & Drop Editor	Hinzufügen von Elemente per Drag & Drop	2	2	2	2
Navigator	Listenansicht der auf der Website platzierten Elemente	0	2	2	2
Copy-Paste	Kopieren und Einfügen von Elementen & Stilen	2	2	2	2
Tastaturkürzel	Tastaturkürzel für gängige Funktionen	2	2	2	2
Wiederverwendbare Elemente	Globale, wiederverwendbare Elemente	2	2	2	2
Globale Farben	Globale, wiederverwendbare Farben	2	2	2	2
Globale Schriftarten	Globale, wiederverwendbare Schriftarten	2	2	2	2
Icon-Bibliothek	Auswahl von vorgefertigten Icons	2	2	2	0
Vorgefertigte Layouts	Auswahl aus vorgefertigten Layoutcontainern	2	2	2	2
Vorgefertigte Websites	Website-Designvorlagen	2	2	2	2
Vorgefertigte Abschnitte	Designvorlagen von Abschnitten und Elemente	2	2	2	2
<b>CMS-Funktionen</b>					
Medienverwaltung	Upload und Verwaltung diverser Medien	2	2	2	2
Formulareinträge	Speicherung von Formulareinträgen in der Datenbank	2	2	2	2
Inhalts-Feldtypen	Unterschiedliche Feldtypen für Inhalte	2	2	2	2
Dynamische Inhalte	Einbindung von dynamischen Inhalte auf die Website	2	2	2	2
Dynamische Inhaltstemplates	Automatisch generierte Templates für Inhaltsbereiche	2	2	2	2
Blog-System	Dynamische Blogeinträge (Paginierung, verwandte Beiträge, etc.)	2	2	2	2
CSV-Import	Datei-Import im CSV-Format	2	2	0	2
CSV-Export	Exportmöglichkeit von Inhalten im CSV-Format	2	2	1	2
Dynamisches Filtern und Sortieren	Filter und Sortierungsoptionen für dynamische Inhalte	2	2	2	2
<b>Komponenten</b>					
<i>Inhaltselemente</i>					
Überschriften	Erstellung und Anpassung von Überschriften	2	2	2	2
Text-Editor	Erstellung und Anpassung von Texten	2	2	2	2

Buttons	Erstellung und Anpassung von Buttons	2	2	2	2
Formulare	Erstellung und Anpassung von Formularen und Formularfeldern	2	2	2	2
Akkordeons	Erstellung und Anpassung von Akkordeons	2	0	2	0
Tabs	Erstellung und Anpassung von Tabs	2	0	2	2
Slider	Erstellung und Anpassung von Slidern für Bilder oder Inhalte	2	2	2	2
Listen	Erstellung und Anpassung von Listen	2	0	2	2
Lightbox & Bildergalerien	Erstellung und Anpassung von vollflächigen Lightboxen und Galerien	2	2	2	2
Navigationsmenü	Erstellung und Anpassung von responsiven Navigationsmenüs	2	2	2	2
Ankerlinks	Erstellung und Anpassung von Ankerlinks	2	2	2	2
Abstandshalter	Erstellung und Anpassung von Abstandshaltern	2	0	2	0
Trenner	Erstellung und Anpassung von Trennern	2	2	2	0
Shape-Divider	Erstellung und Anpassung von Trenn-Elementen auf Basis unterschiedlicher Formen	0	2	2	0
Text-Pfad	Erstellung und Anpassung von Text-Pfaden	0	0	2	0
Bild-Box	Erstellung und Anpassung von Bild-Boxen	0	0	2	0
Icon-Box	Erstellung und Anpassung von Icon-Boxen	0	0	2	0
Flip Box	Erstellung und Anpassung einer Flip Box	0	0	2	0
Zitatblock	Erstellung und Anpassung eines Zitatblocks	0	0	2	2
Hinweismeldungen	Erstellung und Anpassung von Hinweismeldungen	0	0	2	0
Counter	Erstellung und Anpassung von Countern	0	0	2	0
Countdown	Erstellung und Anpassung eines Countdowns	2	0	2	0
Fortschrittsbalken	Erstellung und Anpassung von Fortschrittsbalken	0	0	2	0
Testimonials	Erstellung und Anpassung von Testimonials	0	0	2	0
Sterne-Bewertungen	Erstellung und Anpassung von Sterne-Bewertungen	0	0	2	0
Preistabelle	Erstellung und Anpassung von Preistabellen	0	0	2	0
Inhaltsverzeichnis	Erstellung und Anpassung eines Inhaltsverzeichnisses	0	0	2	0
Lottie-Animation	Erstellung und Anpassung einer Lottie-Animation	2	2	2	2

<i>Medien</i>					
SVG-Icons	Möglichkeit des Uploads eigener Icons als SVG-Datei	2	2	2	2
Audio	Einbettung eines Audio-Players	2	2	2	2
Videos	Erstellung und Anpassung von Video-Elementen	2	2	2	2
Bilder	Erstellung und Anpassung von Bildern	2	2	2	2
<i>Hintergrundelemente</i>					
Hintergrundbilder	Erstellung und Anpassung von Hintergrundbildern	2	2	2	2
Hintergrundvideos	Erstellung und Anpassung von Hintergrundvideos	2	2	2	2
Hintergrundverläufe	Erstellung und Anpassung von Hintergrundverläufen	2	2	2	2
Hintergrund-Slideshows	Erstellung und Anpassung von Hintergrund-Slideshows	2	0	2	2
Hintergrund-Überlagerung	Erstellung und Anpassung von Hintergrund-Überlagerungen	2	2	2	2
<i>Einbettungen</i>					
HTML	Erstellung von HTML-Einbettungsfeldern	2	2	2	2
Social-Icons & Inhalte	Erstellung und Anpassung von Social-Icons	2	2	2	2
Google Maps	Erstellung und Anpassung von interaktiven Google Maps	2	2	2	2
YouTube-Video	Einbettung eines YouTube-Videos	2	2	2	2
Vimeo-Video	Einbettung eines Vimeo-Videos	2	2	2	2
Dailymotion-Video	Einbettung eines Dailymotion-Videos	2	2	0	2
SoundCloud-Player	Einbettung eines SoundCloud-Players	2	2	2	2
Mixcloud-Player	Einbettung eines Mixcloud-Players	2	0	0	0
<b>Accessibility &amp; SEO</b>					
Anpassbare HTML-Tags	Anpassbare HTML-Tags für unterschiedliche Elemente	0	0	2	2
Dynamische SEO-Tags und Open Graph Einstellungen	Automatische Erstellung von SEO-Tags und Open Graph Inhalten	0	0	0	2
Alt-Tags für Bilder	Hinzufügen von alternativen Beschreibungen für Bilder	2	2	2	2
XML-Sitemap	Autogenerated XML-Sitemap	2	2	2	2
Meta-Titel und Meta-Beschreibung	Hinzufügen von Meta-Titel und -Beschreibungen	2	2	2	2

Open Graph Einstellungen	Hinzufügen von Open Graph Inhalten	2	2	2	2
<b>sonstige Funktionen</b>					
Popup-Builder	Erstellen und Anpassen von Popups	2	2	2	2
Widget-Builder	Erstellen von eigenen Elementen	2	0	0	0
Benutzerrollen	Benutzerrollen mit unterschiedlichen Rechten	2	2	2	2
Code-Export	Exportmöglichkeit des Website-Quellcodes	2	0	2	2
Benutzerdefinierter Code	Hinzufügen von benutzerdefiniertem Code	2	2	2	2
Accessibility-Audit	Funktion zum Erkennen von Problemen bei der Barrierefreiheit	0	0	0	2
API	Programmierschnittstelle	2	2	2	2
404 Seite	Anpassbare 404-Seiten	2	2	2	2
Favicon	Hinzufügen eines Website-Icons	2	2	2	2
Passwortschutz	Passwortgeschützte Seiten	2	2	2	2
Backups und Versionierung	Website-Backups und Revisionen	2	1	2	2
Wartungsmodus	Coming Soon Seite während der Websiteerstellung	0	0	2	0
Suchfunktion	Einbindung einer Suchfunktion in die Website	1	2	2	2
Spamschutz	Spamschutz via ReCAPTCHA	2	2	2	2
Website-Statistiken	Einbindung von Google Analytics	2	2	2	2
	<b>Gesamtpunkte</b>	<b>185</b>	<b>177</b>	<b>234</b>	<b>204</b>
<b>Integrationen</b>		<b>Duda</b>	<b>Editor X</b>	<b>Elementor</b>	<b>Webflow</b>
Integrationen	Anzahl der auf der offiziellen Website gelisteten Integrationen	39	465	39	247

## C. Evaluierungsergebnisse Experiment

Kategorie	Beschreibung	Mögliche Punkte	Elementor	Webflow	Anmerkung
<b>Layout</b>		<b>10</b>	<b>10</b>	<b>10</b>	
<i>Main</i>					
HTML-Tag	main	2	2	2	
Breite	1400px	2	2	2	
Höhe	100vh	2	2	2	
Hintergrund	#FFFFFF	2	2	2	
Rahmen-Oben (nur Mobil)	5px solid #004F9A	2	2	2	
<i>Aside</i>		<b>8</b>	<b>6</b>	<b>8</b>	
HTML-Tag	aside	2	2	2	
Hintergrund	Hintergrundbild (50% 50%, cover, no-repeat)	2	1	2	Elementor: Aside-Hintergrundbild skaliert falsch
Position	fixed	2	2	2	
Abstand Links	1400px	2	1	2	Elementor: Aside-Hintergrundbild skaliert falsch
<i>Article (Inhaltsbereich)</i>		<b>10</b>	<b>10</b>	<b>10</b>	
HTML-Tag	article	2	2	2	
Breite	1290px	2	2	2	
Padding	Links Rechts 15px	2	2	2	
2-spaltig	50% 50%	2	2	2	
3-spaltig	33% 33% 33%	2	2	2	
<b>Header</b>		<b>20</b>	<b>18</b>	<b>19</b>	
<i>Navigation</i>					

HTML-Tag	nav	2	2	2	
Text	Open Sans / Größe 14px / Gewicht: 700 / Zeilenhöhe 20px	2	2	2	
Breite	1290px	2	2	2	
Höhe	60px	2	2	2	
Rahmen	1px solid #5C5D64	2	2	2	
Hintergrund	#4A4B53	2	2	2	
Anordnung	Horizontal	2	2	2	
Hover-Effekt	Anpassung der Hintergrundfarbe auf #5C5D64, Deckkraft von 0,7 auf 1	2	1	2	Elementor: Ändern der Deckkraft ohne Code nicht möglich
Suche	Benutzerdefiniertes Icon	2	2	2	
	Beim Anklicken: Suchformular-Modal öffnen & schließen	2	1	1	Elementor & Webflow: Modal lässt sich nur bei Klick auf "X" schließen
<i>Mobile Navigation</i>		<b>12</b>	<b>11</b>	<b>12</b>	
Menü-Icon	Beim Anklicken: Animation zu „X“, Ein-Ausblenden der Navigation	2	1	2	Elementor: Animation des Menu-Icons ohne Code nicht möglich
Text	Open Sans / Größe 14px / Gewicht: 300 / Zeilenhöhe: 22px	2	2	2	
Breite	100%	2	2	2	
Hintergrund	#4A4B53	2	2	2	
Anordnung	Vertikal	2	2	2	
Suchformular	300px breit, 42px hoch	2	2	2	
<i>Hero-Bereich</i>		<b>18</b>	<b>17</b>	<b>17</b>	
Lead-Text	Open Sans / Größe 30px / Gewicht: 300 / Zeilenhöhe: 36px	2	2	2	
Logo	190px breit, 90px hoch	2	2	2	
Suchformular-Container	Hintergrund: Hintergrundbild	2	2	2	
	Höhe: 360px	2	2	2	
	Padding: 70px	2	2	2	
Suchformular	Breite: 420px	2	2	2	

	Höhe: 60px	2	2	2	
	Textfarbe: #999999	2	2	2	
	Beim Anklicken: Dropdown-Auswahlsfeld öffnen	2	1	1	Elementor & Webflow: Select2 Auswahlsfeld ohne Code nicht möglich
<b>Inhaltsbereich</b>		<b>12</b>	<b>9</b>	<b>11</b>	
<i>Topic Teaser</i>					
Padding	40px	2	2	2	
Hintergrund	#4A4B53	2	2	2	
Text	Open Sans / Größe: 14px, 24px, 16px / Gewicht: 400, 700 / Zeilenhöhe: 22px, 24px	2	2	2	
Hover-Effekt	Hintergrundfarbe auf #004F9A	2	2	2	
	eingehend von links, ausgehend von rechts	2	0	1	Elementor: Hintergrundanimation ohne Code nicht möglich; Webflow: Animation der Breite möglich, aber nicht durch Transformation von Hintergrundverläufen
	Pfeil: Keyframe-Animation	2	1	2	Elementor: nur bereits vorgefertigte Animationen auswählbar
<i>Institute List</i>		<b>14</b>	<b>13</b>	<b>14</b>	
Padding	30px	2	2	2	
Hintergrund	#F2F2F2	2	2	2	
Text	Open Sans / Größe: 36px, 24px / Gewicht: 300, 700 / Zeilenhöhe: 43px, 32px	2	2	2	
Rahmen-Links	5px solid #4A4B53	2	2	2	
Hover-Effekt	Rahmen: Farbe auf #004F9A	2	2	2	
	Text: Farbe auf #004F9A	2	2	2	
	Pfeil: Keyframe-Animation	2	1	2	
<i>Text Link Teaser Box</i>		<b>12</b>	<b>11</b>	<b>12</b>	
Padding	50px	2	2	2	
Hintergrund	#4C84B8	2	2	2	

Text	Open Sans / Größe: 36px, 16px / Gewicht: 300, Zeilenhöhe: 43px, 24px	2	2	2	
Rahmen Oben & Unten	1px solid #FFFFFF / Deckkraft 50%	2	2	2	
Hover-Effekt	Rahmen: alle Seiten / Deckkraft 100%	2	2	2	
	Pfeil: Keyframe-Animation; Rahmen: Ein- und Ausblenden des Rahmens	2	1	2	Elementor: nur bereits vorgefertigte Animationen auswählbar
<i>Event Teaser</i>		<b>16</b>	<b>13</b>	<b>16</b>	
Padding	30px	2	2	2	
Hintergrund	#F2F2F2	2	2	2	
Text	Open Sans / Größe: 36px, 24px, 16px, 14px / Gewicht: 700, 400, 300 / Zeilenhöhe: 43px, 24px	2	2	2	
Icon	benutzerdefiniertes Icon / Hintergrund: #4C84B8	2	2	2	
	Border-Radius: 999px	2	2	2	
Horizontale Linie	1px solid #DBDBDD	2	2	2	
Hover-Effekt	Links: Unterstreichen	2	0	2	Elementor: Unterstreichen von Links bei Hover ohne Code nicht möglich
	Pfeil: Keyframe-Animation	2	1	2	Elementor: nur bereits vorgefertigte Animationen auswählbar
<i>Facts</i>		<b>10</b>	<b>10</b>	<b>10</b>	
Padding	60px	2	2	2	
Hintergrund	#4C84B8	2	2	2	
Text	Open Sans / Größe: 48px, 14px / Gewicht: 700 / Zeilenhöhe: 48px, 22px	2	2	2	
Icon	benutzerdefiniertes Icon / Hintergrund: #709DC6	2	2	2	
	Border-Radius: 999px	2	2	2	
<i>News Teaser</i>		<b>8</b>	<b>5</b>	<b>8</b>	
Padding	Oben & Unten 25px, Rechts & Links 15px	2	2	2	
Text	Open Sans / Größe: 36px, 24px, 16px, 14px / Gewicht: 700, 400, 300 / Zeilenhöhe: 43px, 24px	2	2	2	

Hover-Effekt	Links: Unterstreichen	2	0	2	Elementor: Unterstreichen von Links bei Hover ohne Code nicht möglich
	Pfeil: Keyframe-Animation	2	1	2	Elementor: nur bereits vorgefertigte Animationen auswählbar
<b>Footer</b>		<b>10</b>	<b>8</b>	<b>10</b>	
<i>Footer-Top</i>					
Padding	Oben & Unten 50px	2	2	2	
Horizontale Linie	1px solid #DBDBDD	2	2	2	
Text	Open Sans / Größe: 16px / Gewicht: 700, 300	2	2	2	
Externer Link	Position: absolute	2	2	2	
Hover-Effekt	Links: Unterstreichen	2	0	2	
<i>Footer-Bottom</i>					
Icons	benutzerdefiniertes Icon / Hintergrund: #004F9A	2	2	2	
Hover-Effekt	Links: Unterstreichen	2	0	2	Elementor: Unterstreichen von Links bei Hover ohne Code nicht möglich
	Icons: Deckkraft ändern auf 0.7	2	2	2	
<b>Gesamtpunkte</b>		<b>166</b>	<b>145</b>	<b>163</b>	
<b>Prozentueller Wert</b>		<b>100%</b>	<b>87%</b>	<b>98%</b>	