

Bachelorarbeit im Studiengang Medieninformatik

# Drawing Code – Auswahl, Vergleich und Bewertung moderner Webdesign-Tools zur Optimierung von Design- und Entwicklungsworkflows

vorgelegt von Laura Seidler

an der Hochschule der Medien Stuttgart

am 2. August 2016

zur Erlangung des akademischen Grades eines Bachelor of Science

Erstprüfer: Prof. Dr. Dirk Heuzeroth  
Zweitprüfer M. Sc. Matthias Zaunseder

## **EHRENWÖRTLICHE ERKLÄRUNG**

Hiermit versichere ich, Laura Seidler, ehrenwörtlich, dass ich die vorliegende Bachelorarbeit mit dem Titel: „Drawing Code – Auswahl, Vergleich und Bewertung moderner Webdesign-Tools zur Optimierung von Design- und Entwicklungsworkflows“ selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ich habe die Bedeutung der ehrenwörtlichen Versicherung und die prüfungsrechtlichen Folgen einer unrichtigen oder unvollständigen ehrenwörtlichen Versicherung zur Kenntnis genommen.

Stuttgart, den 2. August 2016

Laura Seidler

## **KURZFASSUNG**

Die Verwendung von WYSIWYG-Editoren ist für viele Entwickler aufgrund der schlechten Qualität des durch die Editoren generierten Codes keine Option, obwohl die manuelle Erstellung von HTML und CSS im täglichen Arbeitsleben viel Zeit benötigt. In der vorliegenden Arbeit werden die drei Webdesign-Tools Macaw, Pinegrow und Webflow anhand eines zuvor aufgestellten und auf die G + P Glanzer + Partner Werbeagentur GmbH zugeschnittenen Anforderungskataloges auf ihre Eignung zum Einsatz im Design- und Entwicklungsworkflow sowie für die Erstellung von Click-Prototypen geprüft. Nach Analyse und Vergleich der drei Tools wird der Agentur empfohlen, die Web-App Webflow künftig zur Unterstützung der Designer und Entwickler des Digital-Teams einzusetzen.

## **ABSTRACT**

Due to the poor quality of generated code, using a WYSIWYG editor is not an option for most web developers, even though coding HTML and CSS manually takes up a lot of time in the daily work process. Using a list of requirements geared towards the needs of G + P Glanzer + Partner advertising agency, the three web design tools Macaw, Pinegrow and Webflow are reviewed with respect to their ability to support the design and development workflows as well as the creation of clickable website prototypes. After analysing and comparing the three tools, the web app Webflow is recommended to the agency's digital team to assist designers and developers in their work from now on.

## **SPERRVERMERK**

Die vorgelegte Bachelorarbeit basiert auf internen, vertraulichen Daten und Informationen der G + P Glanzer + Partner Werbeagentur GmbH. In diese Arbeit dürfen Dritte, mit Ausnahme der Gutachter und befugten Mitgliedern des Prüfungsausschusses, ohne ausdrückliche Zustimmung des Unternehmens und des Verfassers keine Einsicht nehmen. Eine Vervielfältigung und Veröffentlichung der Bachelorarbeit ohne ausdrückliche Genehmigung – auch auszugsweise – ist nicht erlaubt.

# INHALTSVERZEICHNIS

---

Inhaltsverzeichnis.....	4
1 Einführung.....	6
1.1 Motivation .....	6
1.2 Fragestellungen.....	7
1.3 Ziele der Arbeit.....	7
1.4 Aufbau der Arbeit.....	7
2 Verwandte Arbeiten .....	9
3 Grundlagen .....	10
3.1 Auszeichnungs- und Programmiersprachen.....	10
3.2 Der Designprozess und die eingesetzten Tools .....	11
3.3 Der Entwicklungsprozess und die eingesetzten Tools .....	12
3.4 Projekte.....	13
4 Anforderungen .....	18
4.1 Allgemeines und Projektplanung .....	18
4.2 Werkzeuge und Funktionalität .....	19
4.3 Einsatzmöglichkeiten.....	21
4.4 Anforderungskatalog.....	22
5 Untersuchung von Webdesign-Tools .....	23
5.1 Macaw .....	24
5.2 Pinegrow .....	29
5.3 Webflow.....	34
5.4 Zusammenfassung.....	40
6 Diskussion und Ausblick.....	42
6.1 Diskussion .....	42
6.2 Empfehlung für die Agentur .....	44
6.3 Ausblick.....	45
7 Literaturverzeichnis.....	46
Anhang.....	51

A	Team-Befragung zu den Anforderungen an ein Webdesign-Tool.....	51
B	Voranalyse und -auswahl von Webdesign-Tools.....	54
C	Abkürzungsverzeichnis.....	55
D	Glossar.....	56
E	Abbildungsverzeichnis .....	57
F	Tabellenverzeichnis.....	57

# 1 EINFÜHRUNG

---

Diese Arbeit wurde in Kooperation mit der G + P Glanzer + Partner Werbeagentur GmbH[1] (fortan „Agentur“) erstellt. Innerhalb der Agentur ist das Team „Webunit“ für alle Belange im Digital-Bereich, insbesondere im Web, zuständig. Die Arbeit ist auf die Bedürfnisse der Agentur und speziell des Digital-Teams ausgerichtet.

## 1.1 MOTIVATION

Im täglichen Arbeitsleben von Webdesignern und Entwicklern kommen zahlreiche Tools zum Einsatz. Dabei gibt es auf der einen Seite spezielle Tools für den Designer (z.B. Photoshop[2], Illustrator[3]), und Tools für den Entwickler (Code-Editoren wie Sublime Text[4] oder Atom[5], Git-Clients wie SourceTree[6] oder GitKraken[7]) auf der anderen. Generell ist die Entwicklung von Design und Code in den meisten Fällen getrennt. Herkömmlicherweise wird das Design zuerst erstellt, im Anschluss vom Kunden genehmigt und dann von den Entwicklern in Code umgesetzt.

Die Aufgaben sind dabei nahezu immer gleich: Es wird ein Framework ausgewählt (z.B. Bootstrap[8] oder Foundation[9]), damit grundlegende Layout- und Content-Elemente (z.B. Grid-System, Buttons) zur Verfügung stehen, bevor auf Basis des Frameworks ein grundlegendes Layout erstellt wird. Steht dieses, werden die Elemente der Webseite erst nach und nach in HTML umgesetzt und dann mit CSS gestaltet. Bei komplexeren Elementen wird ggf. noch JavaScript verwendet, bis die komplette Seite statisch umgesetzt ist, und dann in dieser Form veröffentlicht oder als Template in dynamischen Systemen eingesetzt werden kann.

Dieser Prozess wiederholt sich in fast jedem neuen Projekt. Vereinzelt können Teile aus früheren Projekten übernommen werden, um den Prozess zu beschleunigen, doch der grundlegende Prozess bleibt gleich. Dabei ist die Erstellung des HTML- und CSS-Codes in der Regel nur geringfügig anspruchsvoll. Dies gilt insbesondere, wenn es sich um einfache Webseiten handelt, die nur wenige, einfache Content-Elemente enthalten und weitestgehend standardmäßig aufgebaut sind. Dieser Prozess nimmt aber dennoch viel Zeit in Anspruch, die in jedem Projekt erneut aufgebracht werden muss.

Aus den oben aufgeführten Gründen ist die automatische Generierung von qualitativ hochwertigem Code aus Design daher wünschenswert und effizienzfördernd. In den vergangenen Jahren tauchten vermehrt Tools auf, die genau das versprochen – einfache Erstellung des Designs sowie automatische Generierung von qualitativ hochwertigem Code. Da vergleichbare Tools in der Vergangenheit durch wenig semantisches HTML und kaum logisches CSS auffielen, werden diese sogenannten What You See Is What You Get (fortan:

WYSIWYG) Editoren von Entwicklern seit jeher gemieden. Da durch diese Entwickler-Aversion selten und nur vereinzelte Tools in teilweise bereits stark veralteten Artikeln[10]–[12] aufgegriffen werden, soll diese Arbeit einen genauen Blick auf die neueste Generation von WYSIWYG-Tools werfen und untersuchen, ob die Code-Qualität so gut wie versprochen ist und den Ansprüchen der Entwickler genügen kann.

## **1.2 FRAGESTELLUNGEN**

Zur Untersuchung und zum Vergleich von Tools gilt es zuerst, Inventur zu führen – welche Tools gibt es auf dem Markt, und welche Funktionen bieten diese? Was sind die Alleinstellungsmerkmale der Tools, und in welchen Aspekten versprechen sie, Designern und Entwicklern bei der täglichen Arbeit zu helfen?

Darüber hinaus stellt sich die Frage, welche Features ein Tool von Agentur-Seite haben muss, um in Betracht zu kommen. Dazu muss analysiert werden, welche Anforderungen Designer, Entwickler, aber auch Kundenberater an ein Tool stellen und wie hoch die Bereitschaft ist, sich in ein neues Programm einzuarbeiten. Weiterhin gilt es herauszufinden, ob Tools direkt im Designprozess eingesetzt werden können, oder erst später die Entwickler bei der Arbeit unterstützen.

Als weitere wichtige Frage stellt sich die nach der Qualität des generierten Codes – wie semantisch ist das generierte HTML und wie gut kann ein Design tatsächlich in CSS umgesetzt werden? Gleichzeitig soll untersucht werden, wo ggf. auch Abstriche gemacht werden können und wie viel manuelle Nacharbeit mit der jeweiligen Zeitersparnis durch die automatische Generierung des Codes vereinbar ist.

## **1.3 ZIELE DER ARBEIT**

Die Arbeit zielt darauf ab, ein möglichst optimales WYSIWYG-Tool zu finden, welches Designer und Entwickler dabei unterstützen kann, ihren Workflow zu optimieren. Dabei sollen die Anforderungen der Agentur genau analysiert werden, um dann drei möglicherweise geeignete Tools im Detail auf diese Anforderungen hin zu untersuchen.

## **1.4 AUFBAU DER ARBEIT**

Die folgende Auflistung gibt eine kurze Übersicht über die Inhalte, welche in den einzelnen Kapiteln diskutiert werden.

- **Kapitel 2**

Kapitel 2 stellt einen vorhandenen Vergleich von Webdesign-Tools kurz vor und erklärt das Nichtvorhandensein weiterer Arbeiten.



- **Kapitel 3**

In diesem Kapitel werden die in der Entwicklung eingesetzten Programmiersprachen kurz vorgestellt und ein Überblick über den Design- und Entwicklungsprozess und die dabei eingesetzten Tools gegeben. Weiterhin werden drei bisher in der Agentur realisierte Projekte vorgestellt.

- **Kapitel 4**

Dieses Kapitel beinhaltet die Analyse der Anforderungen der Agentur an ein potenzielles Tool. Dabei werden die Bedürfnisse von Entwicklern, Designern und Kundenberatern analysiert, um daraus einen Anforderungskatalog zu erstellen, der dann im weiteren Verlauf zur Analyse und Bewertung der Tools benutzt werden kann.

- **Kapitel 5**

Kapitel 5 beginnt mit einer Übersicht über den Markt an Tools und stellt die Funktionalität verschiedener Tools kurz da. Anschließend werden ausgewählte Tools anhand des erstellten Anforderungskatalogs analysiert, auf positive und negative Aspekte getestet und ihre Eignung für die Agentur für verschiedene Aufgabenbereiche erfragt.

- **Kapitel 6**

In diesem Kapitel werden die Ergebnisse diskutiert und die geeigneten Tools gegenübergestellt. Aus dem Vergleich geht die Auswahl eines Tools für die Agentur hervor. Die Arbeit wird mit einem Ausblick auf die nähere Zukunft im Markt der WYSIWYG-Tools abgeschlossen.

## 2 VERWANDTE ARBEITEN

---

Brian Wood veröffentlichte 2014 einen Artikel[12] auf der Webseite des Smashing Magazine[13], welcher die Webdesign-Tools Macaw[14], Edge Reflow[15] und Webflow[16] vorstellt und ihre grundlegenden Funktionen erläutert. Der Artikel bietet zwar eine gute Übersicht über die Fähigkeiten von Webdesign-Tools, jedoch wurde die Entwicklung von Edge Reflow bereits bevor das Tool die Marktreife erlangte eingestellt und Macaw sowie Webflow haben seit Erstellung des Artikels an Funktionalität gewonnen.

Durch die rapide Entwicklung am Markt für Webdesign-Tools ist demnach ein 2014 geschriebener Artikel bereits zum Zeitpunkt, an dem die vorliegende Arbeit verfasst wurde, nicht mehr aktuell. Weitere Artikel sind mit Publikation in 2010[11] und 2008[10] ebenfalls stark veraltet und empfehlen mitunter Programme, die mittlerweile nicht mehr verfügbar sind oder neueste Standards durch mangelnde Aktualisierungen nicht berücksichtigen.

Generell ist die Verfügbarkeit von Artikeln und Arbeiten zum Thema WYSIWYG-Editoren gering bis mangelhaft. Dies ist darauf zurückzuführen, dass WYSIWYG-Editoren von den meisten erfahrenen Entwicklern grundsätzlich nicht eingesetzt werden, da die Qualität des generierten Codes oftmals schlecht ist und das manuelle Schreiben des Codes bessere und schnellere Ergebnisse erzielt. Die Möglichkeit, dass moderne Webdesign-Tools qualitativ guten Code generieren können, wird von vielen Entwicklern nicht in Betracht gezogen.

Da die ersten modernen Tools, die qualitativ hochwertigen Code versprechen, erst vor wenigen Jahren veröffentlicht wurden (z.B. Macaw im März 2014 mithilfe einer Kickstarter Kampagne[17]), ist der von Wood veröffentlichte Artikel[12] bis dato die einzige nicht stark veraltete Ressource, die Webdesign-Tools analysiert und vergleicht.

## 3 GRUNDLAGEN

---

Im Folgenden werden die zum Verständnis der in dieser Arbeit behandelten Themen benötigten Grundlagen behandelt. Zu Beginn des Kapitels werden die für die Webentwicklung und insbesondere für den folgenden Vergleich von Webdesign-Tools relevanten Auszeichnungs- und Programmiersprachen dargestellt. Die erwähnten Sprachen sind auf die in der Agentur verwendeten Sprachen beschränkt.

Weiterhin wird ein Überblick über den in der Agentur etablierten Design- und Entwicklungsprozess gegeben sowie für beide Prozesse die im Prozess eingesetzten Tools und ihre für diese Arbeit relevanten Funktionen kurz vorgestellt. Abschließend werden drei in der Vergangenheit umgesetzte Projekte vorgestellt, welche beispielhaft für verschiedene Projekte im Spektrum der Arbeit der Agentur stehen.

### 3.1 AUSZEICHNUNGS- UND PROGRAMMIERSPRACHEN

#### 3.1.1 HTML

HTML[18] wird zur Auszeichnung von Inhalten einer Webseite verwendet. Dabei werden die unterschiedlichen Elemente dazu benutzt, die Inhalte semantisch korrekt auszuzeichnen. So wird z.B. ein Absatz mit dem HTML-Tag `<p>` ausgezeichnet (engl. paragraph).

#### 3.1.2 CSS/LESS/SASS

CSS[19] wird zur Definition des Aussehens der HTML-Elemente benutzt. Dabei werden verschiedene Regeln für das Aussehen von einzelnen Elementen oder Gruppen von Elementen definiert. Die CSS-Erweiterungen LESS[20] und SASS[21] ermöglichen die Definition von Variablen (z.B. für Farben), Mixins und Funktionen, die bei der Erstellung von CSS helfen und am Ende in CSS kompiliert werden.

#### 3.1.3 JavaScript

JavaScript[22] wird hauptsächlich dazu verwendet, um Teile einer Seite dynamisch zu gestalten (z.B. Animationen) oder Daten vom Server zu laden bzw. an den Server zu senden, ohne die Seite neu laden zu müssen. Im Team wird vermehrt nach dem neuen ES6-Standard programmiert, der neue Features bietet, aber wegen mangelnder Browser-Unterstützung noch in reguläres JavaScript kompiliert werden muss.

#### 3.1.4 Serverseitige Programmierung und Template-Sprachen

Zur dynamischen Erstellung von Inhalten (z.B. Einloggen eines Benutzers) und Speicherung von Daten (z.B. Content-Management-Systeme, Gewinnspiele), gibt es zusätzlich zum für den Benutzer sichtbaren Teil der Webseite noch die serverseitige Programmierung. Diese wird meist mit PHP[23] oder node.js[24] (serverseitiges JavaScript) durchgeführt. Mit PHP oder

durch den Einsatz von Template-Sprachen werden in die mit HTML erstellten Webseiten oft dynamische Inhalte eingefügt.

## **3.2 DER DESIGNPROZESS UND DIE EINGESETZTEN TOOLS**

Ein Design wird von einem Designer normalerweise in Photoshop, in Ausnahmefällen in InDesign[25] erstellt. Dabei werden ggf. mehrere Dateien oder Artboards angelegt, um verschiedene Unterseiten darzustellen. Außerdem werden meist eine Desktop- und eine Mobilversion der Seite angelegt, um das Aussehen auf unterschiedlichen Geräten zu veranschaulichen. Weitestgehend wird dabei zusätzlich zum Desktop-Design ein Design für die Smartphone-Breite erstellt. Gestaltungen für Bildschirmgrößen zwischen diesen beiden Designs müssen dann später im Entwicklungsprozess interpoliert werden.

Nach Fertigstellung wird das Design dem Kunden vorgelegt, welcher dann Änderungswünsche äußern kann, die vom Designer umgesetzt werden. Dieser Vorgang wiederholt sich, bis das Design in der Endversion vorliegt.

### **3.2.1 Photoshop**

Photoshop[2] ist das Hauptwerkzeug des Designers zur Erstellung des Screendesigns. Insbesondere die Funktionen zum Zeichnen von Flächen und Linien werden verwendet. Bilder, die in der Webseite eingesetzt werden, können z.B. als Smart-Objekte eingebunden werden, um sie unabhängig vom Screendesign weiter bearbeiten zu können. Durch Ebenen und Ebenen-Gruppen im Programm können die verschiedenen Elemente strukturiert sowie logisch angeordnet werden. Eine logische Benennung der Elemente führt zur Verdeutlichung ihrer Funktion (z.B. Gruppe Slider, Ebene Button), und kann sowohl den Designer als auch den Entwickler bereits unterstützen.

Durch die Anbindung an die Adobe Creative Cloud[26] können häufig verwendete Design-Elemente (z.B. Buttons) in Bibliotheken gespeichert werden, die dann einfach an unterschiedlichen Stellen im Design oder projektübergreifend wiederholt eingesetzt werden können.

### **3.2.2 InDesign**

InDesign[25] ist ein weiteres Tool der Adobe Creative Cloud und dient herkömmlicherweise der Erstellung von Designs und Layout für Print-Produkte sowie für digitale Dokumente wie E-Books. In der Agentur wird es vereinzelt zum Screendesign eingesetzt, wenn das Design von einem Designer aus dem Print-Bereich erstellt wird. Dabei treten in der späteren Umsetzung in Code zum Teil Probleme auf (z.B. bei falscher Einstellung des Programms auf Millimeter statt Pixel). Außerdem ist das Programm für die Entwickler in seiner Verwendung

ungewohnter, was zu einem erhöhten Zeitaufwand zur Extrahierung von Komponenten (z.B. Bildern) und Ablesung von Maßen führt.

### **3.2.3 InVision**

InVision[27] ist ein Tool, mit dem Click-Prototypen erstellt werden können. Auf hochgeladenen Screenshots können klickbare Flächen definiert werden, die dann z.B. auf einen anderen Screenshot verlinken können oder eine Animation auslösen können. InVision wird in der Agentur vor allem in größeren Projekten mit mehreren Unterseiten dazu eingesetzt, dem Kunden das erstellte Design zugänglich zu machen und direkt die angedachten Navigationswege zwischen Unterseiten zu verdeutlichen.

## **3.3 DER ENTWICKLUNGSPROZESS UND DIE EINGESETZTEN TOOLS**

Das Design, welches als Grundlage für die Erstellung der Webseite dient, wird nach oder während der Abstimmung mit dem Kunden an die Entwickler übergeben. Je nach Projekt beginnt im Anschluss daran die Entwicklung der Webseite oder eines Click-Prototypen, falls gefordert und nicht über InVision umsetzbar. Wird ein Click-Prototyp erstellt, so kann die hierfür nötige Struktur nicht automatisch für die Entwicklung der Webseite weiterverwendet werden, sodass hier doppelter Aufwand vonnöten ist.

Zu Beginn der Entwicklung wird oftmals die Basis-Struktur eines früheren Projektes übernommen, um beispielsweise das Build- und Deploy-System schnell aufzusetzen. Dann wird meist ein Framework wie Bootstrap oder Foundation ausgewählt, damit grundlegende Layout-Elemente wie Grids und Komponenten wie Buttons, Formulare und Akkordeons zur Verfügung stehen. Im Anschluss werden HTML und CSS für Seitenlayout und Komponenten erstellt, um die Elemente nach Vorgaben des Designs zu gestalten. Danach werden, bei Bedarf, weitere Funktionalitäten durch JavaScript ergänzt und serverseitige Logik programmiert.

Im Entwicklungsprozess gibt es eine erneute Abstimmung mit dem Kunden, dessen Änderungswünsche teilweise wieder durch den Screendesigner im Design, teilweise durch die Entwickler direkt im Code umgesetzt werden.

### **3.3.1 Code-Editoren**

Zur Entwicklung einer Webseite werden von den Entwicklern verschiedene Code-Editoren eingesetzt. Die am häufigsten eingesetzten Editoren sind dabei Sublime Text und Atom. Beide Code-Editoren bieten ähnliche Funktionen, wozu unter anderem Syntax-Highlighting, Dateiverwaltung, Projektverwaltung und ein Plug-In-System zählen.

### **3.3.2 Webbrowser**

Zur Vorschau der Webseite wird ein Webbrowser wie Google Chrome[28] oder Mozilla Firefox[29] verwendet. Dabei liegt in der Entwicklung das Augenmerk stets darauf, die

Webseite möglichst in allen Browsern gleich aussehen zu lassen, da es oft feine Unterschiede zwischen den verschiedenen Browsern gibt. Alle Desktop-Browser stellen außerdem Entwicklerwerkzeuge zur Verfügung, mit denen Webseiten im Entwicklungsprozess debuggt werden können. Dies ermöglicht es z.B., berechnete Werte bei Abständen oder Schriftgrößen zu überprüfen oder Debug-Ausgaben in der Konsole einzusehen.

Zusätzlich wird die Webseite auch in mobilen Browsern auf Test-Geräten getestet, um das Verhalten der Webseite auf Mobilgeräten zu testen. Ein schmaler Bildschirm und bestimmte Eigenschaften eines Mobilgeräts können ebenfalls über Funktionen der Desktop-Browser emuliert werden, damit das Mobilgerät selbst nicht konstant zur Hand genommen werden muss, was den Prozess beschleunigt.

### **3.3.3 Webserver & Task-Runner**

Ein Webserver wird in der Entwicklung eingesetzt, wenn serverseitige Sprachen wie PHP zum Einsatz kommen oder um den Entwicklungsworkflow zu beschleunigen. Dabei kann der Webserver z.B. mit dem Tool BrowserSync[30] automatisieren, dass die Webseite bei Änderungen im Code im Browser neugeladen wird. Zusammen mit Task-Runnern wie Gulp[31] und/oder Grunt[32] können weitere Funktionen wie die automatisierte Kompilierung von LESS-, SASS- oder ES6-Dateien verwendet werden. Soll die Entwicklungsumgebung für alle Entwickler vereinheitlicht werden, kann z.B. eine Software wie Vagrant[33] eingesetzt werden, die die komplette Entwicklungsumgebung in einer virtuellen Maschine generiert.

## **3.4 PROJEKTE**

### **3.4.1 Lotto Baden-Württemberg Facebook-App**

Die Facebook-App von Lotto Baden-Württemberg (fortan: Lotto BW) wird seit mehreren Jahren von der Agentur gepflegt und regelmäßig aktualisiert. Im Hintergrund steht dabei eine normale Webseite, die in Facebook-Tabs eingebunden wird und so auf der Fan-Page von Lotto BW zur Verfügung steht. Die App besteht aus mehreren Unterseiten, die hauptsächlich Informationen zum Lotto-Spiel oder zum Unternehmen beinhalten. Diese Unterseiten werden von der Agentur in unregelmäßigen Abständen mit Text- oder Bild-Updates ausgestattet, Design-Änderungen werden nur selten und in kleinen Maßstäben vorgenommen.

Zusätzlich zu diesen Unterseiten gibt es einen Gewinnspiel-Tab, der mehrmals im Jahr mit einem neuen Gewinnspiel aktualisiert wird. Dabei gibt es i.d.R. drei bis vier Gewinnspiele pro Jahr, die einzelnen Gewinnspiele werden vor der Umsetzung jeweils vom Designer aufgebaut und nach Freigabe durch den Kunden programmiert.

Der Gewinnspiel-Tab (siehe Abbildung 1) steht in dieser Arbeit beim Test der in Frage kommenden Tools im Fokus, da er ein gutes Beispiel für die Umsetzung eines einfachen One-Pagers bietet (nur eine Seite, wenig Inhalte).

### 3.4.1.1 Technik

Die Facebook-App ist eine in PHP programmierte Web-Anwendung, in der das Slim Framework[34] und die Smarty Template-Engine[35] zum Einsatz kommen. Diese helfen beim Routing von Seite zu Seite und beim Schreiben der Templates. So gibt es ein Basis-Template für die gesamte App, in welches dann die individuellen Inhalte pro Seite eingebunden werden. Das Gewinnspiel-Template selbst besteht final nahezu ausschließlich aus HTML, mit wenigen eingestreuten Smarty-Tags für z.B. Benutzername und Anzahl an Losen bei einem Gewinnspiel mit Verlosung.

Zur lokalen Entwicklung wird eine Vagrant-Box eingesetzt, in der u.a. das für die Facebook-Verbindung benötigte SSL-Zertifikat zur Verfügung steht. Als CSS-Präprozessor kommt LESS zum Einsatz. Der JavaScript-Code ist in Module aufgeteilt, welche über RequireJS[36] geladen werden. Als zugrundeliegendes Frontend-Framework werden Teile des Bootstrap-Frameworks sowohl für CSS als auch für JavaScript eingesetzt.

### 3.4.1.2 Inhalte

Der Aufbau der Facebook-App ist simpel und benötigt wenige Content-Elemente. Für Gewinnspiele mit Benutzereinsendungen werden Formulare verwendet, ggf. hochgeladene Bilder werden später in einer Bildergalerie angezeigt. Falls eine Benutzer-Abstimmung stattfindet, beinhaltet diese auch Abstimmungsfunktionalität. Die Inhalte sind ein- und zweispaltig angeordnet. Eine Ausnahme bildet der Adventskalender, der i.d.R. 24 Türchen in einer Tannenbaum-Form anordnet.

Abbildung 1: Screendesign Lotto BW "Kabinenaktion"



### 3.4.2 Heimatland-Webseite

Die Heimatland-Webseite (siehe Abbildung 2) wurde vor einiger Zeit von der Agentur für die Frankengut GmbH entwickelt. Sie besteht aus momentan sieben Einzelseiten, die jeweils Informationen über die Marke und ihre Produkte enthalten. Gelegentlich werden kleine Gewinnspiele auf der Startseite oder einer Unterseite veranstaltet, welche i.d.R. von den Entwicklern ohne vorheriges Screendesign direkt umgesetzt werden.

Die Heimatland-Webseite dient in dieser Arbeit als Beispiel für eine kleine, mehrseitige Webseite ohne nennenswerte Besonderheiten. Sie befindet sich damit auf der Projektgrößen-Skala zwischen dem One-Pager und dem großen Websitesystem.

#### 3.4.2.1 Technik

Die Heimatland-Webseite ist eine in PHP programmierte Webseite, in der wie auch in der Facebook-App von Lotto BW das Slim Framework und die Smarty Template-Engine zum Einsatz kommen. Dabei werden die auf der Webseite dargestellten Informationen (z.B. Produkte) aus JSON-Dateien ausgelesen und dann durch Smarty-Tags dynamisch in das HTML eingefügt. Als CSS-Präprozessor kommt LESS zum Einsatz, für neu hinzugefügte Komponenten wird die BEM-Methodik verwendet, die vom Team in 2015 adaptiert wurde und seitdem in allen neuen Projekten eingesetzt wird. Teile aus Bootstrap werden als CSS-Framework verwendet. JavaScript wird aus Einzeldateien eingebunden, ein Module-Loader ist nicht im Einsatz.

#### 3.4.2.2 Inhalte

Die Webseite beinhaltet typische Komponenten wie Navbar, Header-Bühne und Footer. Auf Mobilgeräten wird die Navbar zur Off-Canvas-Navigation. Im Inhaltsbereich sind die Inhalte ein- oder mehrspaltig angeordnet und werden teilweise in Boxen dargestellt. Zusätzlich gibt es Slider, Tabs und Formulare.

### 3.4.3 Fichtner Websitesystem

Das Websitesystem für Fichtner (siehe Abbildung 3) ist eine auf Typo3[37] basierende Entwicklung, die für die Fichtner GmbH sowie ihre Tochterfirmen eingesetzt wird. Da Typo3 als Content-Management-System eingesetzt wird, können im System beliebig viele Seiten angelegt und diese mit beliebigen Inhalten befüllt werden. Da die Verantwortung für die



Abbildung 2: Screendesign Heimatland Startseite



Inhalte der Seite nicht auf Seiten der Agentur liegt, ist das System darauf ausgerichtet, möglichst viele Nutzungsmöglichkeiten abzudecken.

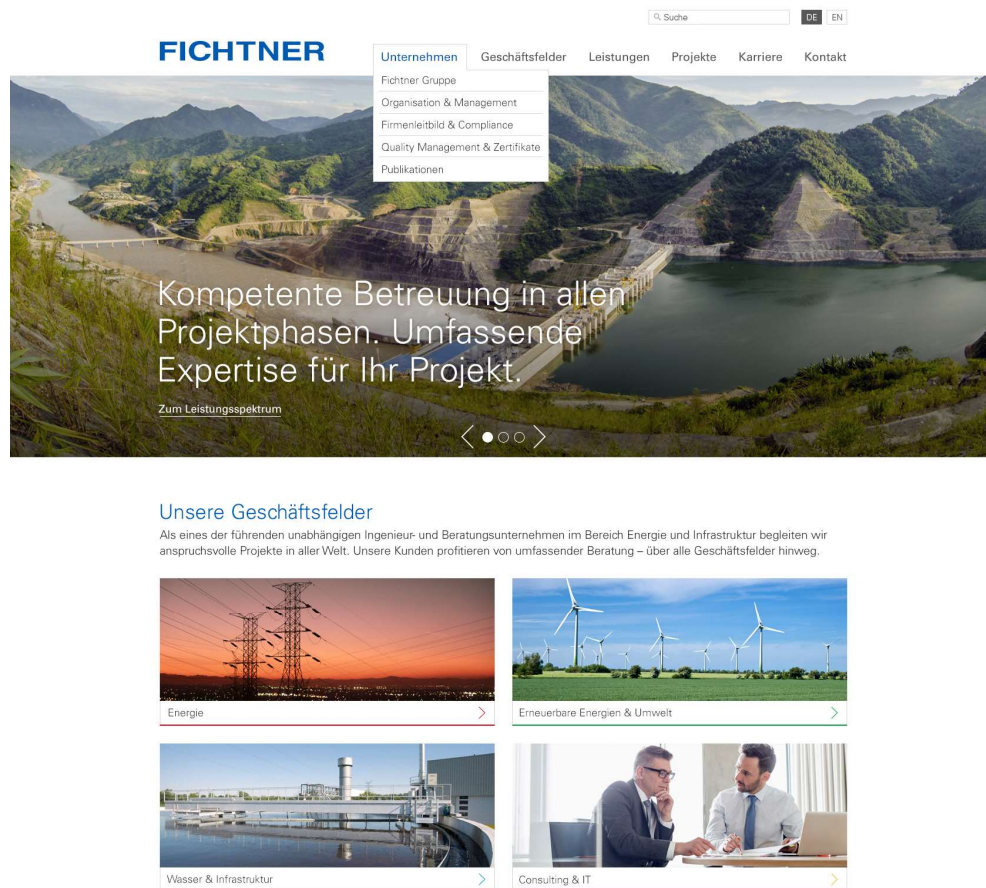


Abbildung 3: Screendesign Fichtner

Zusätzlich zum eigentlichen System wird ein Team-interner Styleguide sowie eine Dokumentation für die mit dem System arbeitenden Mitarbeiter der Fichtner GmbH gepflegt, die eine Einführung in Typo3 bietet und zusätzlich die programmierten Elemente erklärt.

Als komplexes System, welches über mehrere Monate entwickelt wurde, dient es in dieser Arbeit als Beispiel für ein großes Projekt. Da das System Anfang 2016 entwickelt wurde, befindet es sich auf dem neusten Stand der Technik und setzt daher die Maßstäbe in diesem Bereich.

### 3.4.3.1 Technik

Das Fichtner Websitesystem ist eine Typo3-Entwicklung und basiert daher auf PHP. Innerhalb von Typo3 kommt die Erweiterung Fluid[38] zur Template-Entwicklung zum Einsatz. Zur lokalen Entwicklung wird ein Webserver wie XAMPP[39] verwendet. Als CSS-Präprozessor kommt SASS zum Einsatz, beim Schreiben von CSS wird die BEM-Methodik angewandt. JavaScript wird in Module strukturiert, in denen der neuste Standard ES6 verwendet wird. Ein

Gulp-Task kompiliert sowohl CSS als auch JavaScript. Gleichzeitig ist dieser u.a. dafür verantwortlich, den Styleguide und die Dokumentation zu generieren.

#### ***3.4.3.2 Inhalte***

Das Websitesystem enthält zahlreiche Content-Elemente, die über den Typo3-Editor beliebig einsetzbar sind. Die Basis bilden verschiedene Layouts, die Container für Elemente bilden, welche in verschiedenen Kombinationen mehrspaltig anordbar sind. In diesen Grids gibt es verschiedene Content-Elemente, u.a. Keyvisuals, Teaser-Boxen, Icon-Boxen, Akkordeons, Slider, Tabs, Filter, Listen, Karten und Videos. Die Elemente haben fest definierte Felder, welche dann über das Typo3-Backend mit Werten belegt werden können. Alle Elemente sind für sich alleinstehend, passen sich aber dem sie umgebenden Container an.

## 4 ANFORDERUNGEN

---

Zur Analyse der Anforderungen wurde den Entwicklern, Designern und Kundenberatern des Teams ein Fragebogen vorgelegt, der eine Liste von Kriterien enthielt, die in Zusammenhang mit der Auswahl und Verwendung eines Tools stehen. Diese Liste entstand dabei aus Vorgesprächen mit den Mitgliedern des Teams sowie Erfahrung aus bisherigen Projekten in der Agentur.

Den teilnehmenden Personen wurden je nach Tätigkeitsbereich unterschiedliche Fragen gestellt, die erfragten, als wie wichtig die befragte Person das in Frage stehende Feature erachtet. Alle Fragen waren dabei auf einer Skala von 1 (unwichtig) bis 10 (sehr wichtig) zu beantworten.

Zusätzlich gab es die Möglichkeit, in Freitextfeldern Anmerkungen zu möglichen Chancen/Vorteilen und Risiken/Nachteilen beim Einsatz eines Webdesign-Tools sowie sonstige Anmerkungen zu machen. Die Ergebnisse des Fragebogens werden im Folgenden in Kategorien unterteilt dargestellt, um anschließend einen Anforderungskatalog für die darauffolgende Analyse der Tools zu erstellen. Für den Anforderungskatalog werden die ermittelten Wichtigkeiten zu „hoch“ ( $\geq 7,5$ ), „mittel“ ( $\geq 5$  und  $< 7,5$ ) und „gering“ ( $< 5$ ) vereinfacht.

Insgesamt beantworteten sieben Team-Mitglieder den Fragebogen, wovon zwei als Kundenberater sowie in der Projektplanung, zwei als Screendesigner und vier als Entwickler tätig sind (mehrere Tätigkeitsbereiche sind möglich). Zusätzlich zu den tätigkeitspezifischen Fragen wurden zusätzlich noch allgemeine Fragen an alle Team-Mitglieder gerichtet.

In Anhang A ist eine komplette Aufstellung aller Fragen und Antworten des Fragebogens zu finden.

### 4.1 ALLGEMEINES UND PROJEKTPLANUNG

#### 4.1.1 Dokumentation und Tutorials (ID #01)

Um ein neues Programm schnell erlernen zu können und generell die Benutzung so einfach wie möglich zu halten, ist eine adäquate Dokumentation erforderlich, welche die Funktionalität des Programms und seiner Features erläutert. Eine solche Dokumentation wird von den Befragten mit einer Wichtigkeit von 7,7 bewertet. Weiterhin ist es wünschenswert, eine Anzahl an Text- (Wichtigkeit 6,3) und ggf. Video-Tutorials (Wichtigkeit 4,7) zur Verfügung zu haben, um insbesondere komplexe Funktionalität genau zu beschreiben und konkrete Anwendungsfälle genauer zu beleuchten.

Das Kriterium „Dokumentation und Tutorials“ wird daher mit ID #01 und Wichtigkeit mittel in den Anforderungskatalog aufgenommen.

#### **4.1.2 Bestehende Projekte (ID #02)**

Da in den vergangenen Jahren bereits viele Web-Projekte umgesetzt wurden, wäre es wünschenswert, auch bereits bestehende Projekte mit einem entsprechenden Programm bearbeiten zu können. Diese Möglichkeit wird von den Kundenberatern mit einer Wichtigkeit von 6,1 bewertet.

Das Kriterium „Bestehende Projekte“ wird daher mit ID #02 und Wichtigkeit mittel in den Anforderungskatalog aufgenommen.

#### **4.1.3 CSS- (ID #03) und JavaScript-Tools und -Frameworks (ID #04)**

Zur Vereinfachung des Entwicklungsprozesses basieren die meisten im Team entwickelten Webseiten auf einem Framework wie Bootstrap oder Foundation, die vorkonfigurierte Komponenten und Styles/Skripte bieten, die von den Entwicklern bei Bedarf verwendet werden können. Die Auswahl eines solchen Frameworks wird von den Entwicklern mit einer Wichtigkeit von 7,5 bewertet.

Zusätzlich zum verwendeten Framework werden LESS oder SASS als CSS-Präprozessoren sowie der ES6 Standard mit einem Module-Loader als JavaScript-Erweiterungen verwendet. Da sowohl LESS/SASS als auch ES6 in reines CSS bzw. JavaScript kompiliert werden müssen, wird für die Entwicklung ein Build-System aufgesetzt, welches diese und weitere Aufgaben wie z.B. den Deploy übernimmt. Den Einsatz von LESS oder SASS erachten die Entwickler als hoch wichtig (Wichtigkeit 7,8), während der Einsatz von ES6 und weiteren Build-Werkzeugen mit einer geringen Wichtigkeit von 4,8 bewertet wird.

Das Kriterium „CSS-Tools und Frameworks“ wird daher mit ID #03 und Wichtigkeit hoch, das Kriterium „JavaScript-Tools und Frameworks“ mit ID #04 und Wichtigkeit gering in den Anforderungskatalog aufgenommen.

### **4.2 WERKZEUGE UND FUNKTIONALITÄT**

#### **4.2.1 Interface (ID #05)**

Screendesigns werden in der Agentur fast ausschließlich in Photoshop erstellt, welches einen breiten Satz an Werkzeugen bietet, jedoch auch viel Funktionalität enthält, die für das Erstellen eines Screendesigns nicht nötig ist. Hier liegen die Chancen eines spezifischen Webdesign-Tools, welches das Interface auf die wesentlichen, Web-relevanten, Werkzeuge beschränken und somit größere Übersicht bieten kann. Jedoch sollte das neue Tool von der für die Designer gewohnten Oberfläche nicht maßgeblich abweichen, um eine schnelle

Eingewöhnung und schnelle Wechsel zwischen den Programmen zu ermöglichen. Dieser Punkt wird von den Designern mit einer Wichtigkeit von 8,5 bewertet. In der Befragung aller Beteiligten wurde ein intuitives Benutzerinterface mit einer Wichtigkeit von 9 bewertet.

Das Kriterium „Interface“ wird mit ID #05 und Wichtigkeit hoch in den Anforderungskatalog aufgenommen.

#### **4.2.2 Responsive Design (ID #06)**

Bisher in Photoshop durch mehrere Dateien oder Artboards realisiert, sollte das Tool Hilfestellung bieten, ein Design für verschiedene Auflösungen zu optimieren. Hierfür ist vor allem der Breitenwert der Auflösung relevant, da die Breite des Bildschirms z.B. bestimmt, ob Elemente nebeneinander oder untereinander gezeigt werden können. Standardmäßig verwendete Breiten sind dabei Desktop (ca. >1200px), Tablet (ca. >700px) und Smartphone (ca. <700px). Dieser Punkt wird von den Designern mit einer Wichtigkeit von 6,5 bewertet. Zu dieser Thematik wurde ebenfalls die Gruppe der Entwickler befragt, die die Ansicht von verschiedenen Auflösungen und das Erstellen von Media Queries jeweils mit der Wichtigkeit 8,5 bewertet. Gemittelt ergibt sich daher eine hohe Wichtigkeit dieses Kriteriums.

Das Kriterium „Responsive Design“ wird daher mit ID #06 und Wichtigkeit hoch in den Anforderungskatalog aufgenommen.

#### **4.2.3 Multi-Site und Bibliotheken (ID #07)**

Da viele kleine und alle Projekte ab mittlerer Größe in der Agentur mehrere Unterseiten enthalten, die bisher in Photoshop durch mehrere Dateien oder Artboards realisiert wurden, sollte ein mögliches Tool die Designer hier bei der Arbeit unterstützen und entsprechende Hilfestellung zum Erstellen von mehreren Unterseiten anbieten (Wichtigkeit 7). Über Unterseiten hinweg gibt es oftmals Elemente (z.B. Akkordeons, Kontakt-Boxen), die auf mehreren Seiten mit gleichem Design eingesetzt werden. Das Tool sollte hierzu Hilfestellung bieten, damit bereits erstellte Elemente nicht wiederholt gestaltet werden müssen (Wichtigkeit 6,5). Weiterhin sollten auf unterschiedlichen Webseiten oft verwendete Elemente (z.B. Navbar) als vordefinierte Elemente vorhanden sein, um den Designern hier die Arbeit zu erleichtern (Wichtigkeit 7,5).

Das Kriterium „Multi-Site und Element-Bibliotheken“ wird mit ID #07 und der Wichtigkeit mittel in den Anforderungskatalog aufgenommen.

#### **4.2.4 Validität und Semantik (ID #08)**

Ein eingesetztes Tool sollte sowohl validen (nach Standards des W3C[40], Wichtigkeit 8) als auch semantisch logischen Code erzeugen. Dies gilt sowohl für die Auszeichnung der Elemente im HTML (Wichtigkeit 7,8) als auch für sinnvolle Klassenbezeichnungen zur Verwendung im

CSS. Dabei sollte, wenn möglich, eine Methodik wie BEM eingesetzt werden, um die Funktion eines Elements eindeutig erkennbar und das Styling des Elements unabhängig vom Rest der Webseite zu machen, um es flexibel einsetzen zu können (Wichtigkeit 7).

Bei der Beantwortung des Fragebogens erteilten die Entwickler der Möglichkeit, volle Kontrolle über den generierten Code zu haben, eine Wichtigkeit von 8,3. Durch weitere Anmerkungen in den Freitext-Feldern und in persönlichen Gesprächen stellte sich heraus, dass dieser Wunsch hauptsächlich zustande kommt, da mangelnde Code-Qualität und weniger Flexibilität befürchtet werden. Weiterhin ist die Möglichkeit, das Tool ohne Programmierkenntnisse nutzen zu können, für einen Designer, welcher nicht als Entwickler arbeitet, von Wichtigkeit 10. Daraus ergibt sich, dass der Codezugriff für die Entwickler zwar wichtig ist, aber zu vernachlässigen ist, wenn der Code ohne manuellen Eingriff von guter Qualität ist. Für die Bewertung der Validität und Semantik wird daher der generierte Code ohne direkten Eingriff in diesen und ohne manuelle Nacharbeit zugrunde gelegt.

Das Kriterium „Validität und Semantik“ wird mit ID #08 und Wichtigkeit hoch in den Anforderungskatalog aufgenommen.

#### **4.3 EINSATZMÖGLICHKEITEN (DESIGNPROZESS ID #09, CLICK-PROTOTYP #ID 10, ENTWICKLUNGSPROZESS ID #11)**

Der Einsatz eines Tools ist sowohl von Beginn des Designprozesses an, als auch später im Verlauf des Projektes im Entwicklungsprozess denkbar. Als Spezialfall wird in vielen Projekten ein Click-Prototyp als Zwischenstufe zwischen Design und fertiger Webseite benötigt.

Die Entlastung von Screendesignern und damit hauptsächlich der Einsatz des Tools im Designprozess wird mit einer Wichtigkeit von 4 als wenig wichtig bewertet und daher als Kriterium „Einsatz im Designprozess“ mit ID #09 und Wichtigkeit gering in den Anforderungskatalog aufgenommen.

Die schnelle und effiziente Erstellung von Click-Prototypen wird mit höchster Wichtigkeit 10 bewertet und daher als Kriterium „Erstellen von Click-Prototypen“ mit ID #10 und Wichtigkeit hoch in den Anforderungskatalog aufgenommen.

Die Entlastung von Entwicklern und damit hauptsächlich der Einsatz eines Tools im Entwicklungsprozess (entweder durch Übernahme des schon im Tool erstellten Designs aus dem Designprozess oder durch Neuaufbau der Webseite im Tool nach einem separat erstellten Design) wird mit einer Wichtigkeit von 7,5 bewertet und daher als Kriterium „Einsatz im Entwicklungsprozess“ mit ID #11 und Wichtigkeit hoch in den Anforderungskatalog aufgenommen.

#### 4.4 ANFORDERUNGSKATALOG

Auf Basis dieses Kapitels wird für die folgende Analyse von Webdesign-Tools folgender Anforderungskatalog zugrunde gelegt:

ID	Name	Wichtigkeit
#01	Dokumentation und Tutorials	mittel
#02	Bestehende Projekte	mittel
#03	CSS-Tools und -Frameworks	hoch
#04	JavaScript-Tools und -Frameworks	gering
#05	Interface	hoch
#06	Responsive Design	mittel
#07	Multi-Site und Elementbibliotheken	mittel
#08	Validität und Semantik	hoch
#09	Einsatz im Designprozess	gering
#10	Erstellung von Click-Prototypen	hoch
#11	Einsatz im Entwicklungsprozess	hoch

*Tabelle 1: Anforderungskatalog*

## 5 UNTERSUCHUNG VON WEBDESIGN-TOOLS

---

Anhand des im vorigen Kapitel aufgestellten Anforderungskataloges sollen nun verfügbare Webdesign-Tools anhand der Kriterien getestet werden. Dazu wurde eine Vorauswahl an Tools getroffen, bestehend aus drei Tools, die nach Angaben der Herausgeber auf das vom Anforderungskatalog aufgestellte Profil passen. Eine vollständige Liste mit voruntersuchten Tools und ihren Aufnahme- bzw. Ablehnungskriterien befindet sich im Anhang B. Die folgenden drei Tools wurden schlussendlich ausgewählt und sollen im Folgenden untersucht werden:

### **Macaw**

Das Webdesign-Tool Macaw wirbt mit dem Slogan „Stop writing code, start drawing it.“[14] und verspricht, bei Flexibilität wie im gewohnten Bildbearbeitungsprogramm, die Generierung von semantischem HTML und prägnantem CSS. Beworbene Hauptfeatures sind dabei vor allem:

- Responsive Design
- Remote-Preview
- Einsatz von Webfonts, Variablen und Skripten
- Multi-Site mit von mehreren Seiten gemeinsam verwendeten Assets, Komponenten und globalen Styles
- Verschiedene Positionierungs-Modi

Weiterhin wirbt das Tool damit, dass alle Elemente so im Dokument positioniert werden, wie „ein erfahrener Entwickler dies tun würde“[14] und dass die zugrundeliegende Engine zur Code-Generierung auf einer „Basis von Best-Practices aus der Community“[14] basiert.

### **Pinegrow**

Mit dem Slogan „Build websites faster with Pinegrow Web Editor“ wirbt der Web-Editor Pinegrow[41]. Dabei verspricht das Programm, responsive Multi-Page Webseiten mit Komponenten für Frameworks wie Bootstrap oder Foundation schneller erstellbar zu machen. Dabei werden u.a. die folgenden Features beworben:

- Öffnen von HTML-Dateien und schnelle HTML-Bearbeitung mit visuellen Tools
- Live-Ansicht der erstellten Webseite
- Frameworks und Blocks
- Codezugriff und gleichzeitiges Bearbeiten im Code-Editor
- Responsive Design
- Projekte, Master Pages, Komponenten, Partialen und Bibliotheken mit *Pinegrow PRO*



Pinegrow gibt außerdem an, sich von anderen Webdesign-Tools wie Dreamweaver[42], Macaw, Jetstrap[43] und Webflow[16] durch das direkte Bearbeiten von HTML-Dateien ohne Abstraktionen abzuheben[41].

## **Webflow**

Die Webdesign-Plattform Webflow vereint Webdesign und -hosting. Im Designer werden Webseiten erstellt und können dann direkt veröffentlicht werden und bei Bedarf über das Webflow CMS mit Inhalt befüllt werden. Webflow verspricht, dynamische und responsive Webseiten zu erstellen und zu veröffentlichen, ohne Code zu schreiben[16]. Hauptfunktionalitäten des Designers sind vor allem:

- Design ohne Code, Export von HTML & CSS
- Responsive Design
- Animationen und Interaktionen
- Vordefinierte Komponenten
- Webfonts, Farbmanagement und wiederverwendbare Elemente
- Erweiterbarkeit

Um den Vergleich mit anderen Tools zu ermöglichen, wird in dieser Arbeit hauptsächlich der Webflow-Designer bewertet. Die CMS-Funktionalität wird jedoch ggf. als zusätzliche Funktionalität berücksichtigt.

## **5.1 MACAW**

### **5.1.1 Dokumentation und Tutorials (Wichtigkeit: mittel)**

Zu Macaw steht auf der Webseite des Tools eine umfangreiche englischsprachige Dokumentation zur Verfügung[44]. Dabei wird auf unterschiedliche Aspekte, wie das Zeichnen von Elementen und deren Positionierung, Komponenten, Texte und Fonts eingegangen. In jedem Unterkapitel wird dabei ein Feature oder ein Prozess durch Texte, Bilder und Kurzvideos in GIF-Form erklärt. Zum Einstieg in die Benutzung des Tools sind die angebotenen Inhalte ausreichend, auch tiefergreifende Mechanismen werden teilweise erklärt.

Zusätzlich zur Dokumentation bietet Macaw selbst einige Video-Tutorials[45] an, in denen insbesondere die verschiedenen Werkzeuge und die Responsive Design Funktionalität erklärt werden. Weiterhin verweist die Webseite auf weitere Tutorials[46], die einen noch tieferen Einblick in Workflows mit Macaw bieten. Zu Macaw sind außerdem mehrere Bücher erschienen[47]; [48], die das Tool erklären und beispielhafte Projekte in Macaw aufbauen.

### 5.1.2 Bestehende Projekte (Wichtigkeit: mittel)

Ein Import von bestehenden Projekten aus Photoshop oder Code-Basis ist in Macaw nicht möglich. Alle Projekte, die nicht im Macaw-eigenen `.mcw`-Format vorliegen, müssten daher erneut im Editor erstellt werden.

### 5.1.3 CSS-Tools und -Frameworks (Wichtigkeit: hoch)

Über Swatches ähnlich zu Photoshop können Farben pro Projekt gespeichert werden, ansonsten bietet Macaw keine LESS- oder SASS-Funktionalität. Hilfe zur Benutzung von Frameworks wie Bootstrap oder Foundation sind nicht vorhanden, diese könnten durch manuell erstellte Komponenten und externe Stylesheets nachgebildet werden. Der Prozess hierzu wäre jedoch sehr aufwändig. Eldridge empfiehlt, Klassennamen hierarchisch zu benennen, um diese aus dem von Macaw generierten CSS später in eine geschachtelte LESS- oder SASS-Struktur zu konvertieren[47]. Auch dies würde jedoch zusätzlichen Aufwand bedeuten.

### 5.1.4 JavaScript-Tools und -Frameworks (Wichtigkeit: gering)

JavaScript kann zwar manuell hinzugefügt werden, zur Verwendung vorbereitete Komponenten aus Frameworks gibt es jedoch nicht. ES6 kann ohne externen Build-Prozess ebenfalls nicht verwendet werden.

### 5.1.5 Interface (Wichtigkeit: hoch)

Macaw wird als Software auf dem Computer installiert. Die Oberfläche wirkt aufgeräumt und übersichtlich, mit großem Bereich zum Designen in der Mitte und Werkzeugen auf der linken und rechten Seite. Dabei erinnert die Aufteilung stark an Photoshop und andere Programme der Adobe Creative Suite, da die Hauptwerkzeuge mit Icons in einer kleinen Spalte auf der linken Seite verfügbar sind, während in der rechten, etwas größeren Spalte zusätzliche Funktionen zur Verfügung stehen. Da die Hauptwerkzeuge nur als Icons dargestellt sind, die sich stark ähneln, ist jedoch nicht direkt ersichtlich, welches Werkzeug welche Funktion übernimmt. Diese wird jedoch durch vorhandene Tooltips oder durch einen Blick in die Dokumentation ersichtlich. Insbesondere der Unterschied zwischen „Element“ und „Container“, die beide einen `div`-Container auf der Arbeitsfläche erzeugen, ist nicht sofort eindeutig und wird erst während des gründlichen Lesens der Dokumentation ersichtlich.

Beim Erstellen von Elementen zeigen sich in Macaw erste größere Schwächen: Buttons wie der *done*-Button zum Fertigstellen eines gezeichneten Elementes sind wenig intuitiv oben links platziert. Elemente die visuell in andere Elemente gezeichnet werden, werden von Macaw auf die gleiche Ebene wie die Eltern-Elemente gesetzt, wie ein Blick in den in der rechten Seitenleiste befindlichen Element-Baum offenbart. Über diesen Baum lassen sich Elemente dann entsprechend schachteln, ähnlich zu Photoshop-Gruppen. Bedienfelder für

Innen- und Außenabstände, obwohl inhaltlich ähnlich, befinden sich an komplett unterschiedlichen Stellen im Editor.

Weitere Probleme offenbaren sich beim Handling von bereits gezeichneten Elementen. Um Elemente, die verschachtelt sind, bearbeiten zu können, müssen diese entweder über die Baumstruktur oder über mehrere Klicks und Doppelklicks ausgewählt werden. Die entsprechende Ebene im Baum wird dann isoliert, Elemente in höheren Ebenen können dann bis zur Aufhebung der Isolation nicht mehr angewählt werden. Der gesamte Prozess fühlt sich langsam und wenig intuitiv an, da zu Beginn nicht klar ist, warum Elemente außerhalb des momentan isolierten Elements nicht anwählbar sind.

Insgesamt besticht Macaw mit einem nach modernen Standards gefertigten, ansprechenden Design. Funktionell ist das Interface aber stellenweise wenig intuitiv zu bedienen. Um hier effizient zu arbeiten, ist eine hohe Einarbeitungszeit vonnöten.

#### **5.1.6 Responsive Design (Wichtigkeit: mittel)**

Macaw bietet über der Arbeitsfläche eine Leiste, mit der über einen Schieberegler schnell und einfach beliebige Breiten für die Arbeitsfläche einstellbar sind. Im Breakpoint-Menü können dann Breakpoints frei definiert werden. Alle angelegten Breakpoints stehen dann ebenfalls in der bereits erwähnten Leiste zur Anpassung der Arbeitsflächen-Breite zur Verfügung, um hier schnell die exakte Breite eines vordefinierten Breakpoints auswählen zu können.

Styles für einen bestimmten Breakpoint können dann individuell gesetzt werden, ein Breakpoint funktioniert dabei als Standard-Breakpoint, in dem die Seite und deren Elemente inhaltlich bearbeitet werden können, um diese dann für alle anderen Breakpoints bei Bedarf individuell gestalten zu können. Macaw lässt sich dabei nicht nach dem Mobile-First-Ansatz bedienen, da der Standard-Breakpoint eine Größe zwischen 808 und 2000 Pixeln haben muss. Somit muss zuerst die Desktop-Ansicht erstellt und gestaltet werden, um diese dann für Mobilgeräte anzupassen.

Insgesamt eignen sich die Funktionen von Macaw hervorragend dafür, responsive Designs zu erstellen. Abzüge gibt es für den mangelnden Mobile-First-Ansatz.

#### **5.1.7 Multi-Site und Elementbibliotheken (Wichtigkeit: mittel)**

Das Anlegen mehrerer Unterseiten in Macaw ist simpel, es können nach Belieben neue leere Seiten angelegt werden oder vorhandene Seiten dupliziert werden, Master-Templates gibt es hierfür jedoch nicht. Sollen alle Seiten das gleiche Layout haben, so müssen diese von einer vorhandenen Seite dupliziert und nachträgliche strukturelle Änderungen auf allen Seiten einzeln durchgeführt werden. Zum Export des Codes können die Styles aller Seiten

zusammengefasst werden, sodass Code über die Unterseiten hinweg gemeinsam verwendet werden kann.

Aus bestehenden Elementen oder Elementgruppen können Komponenten erstellt werden, die aus einer Liste dann beliebig auf verschiedenen Seiten eingefügt werden können. Die eingefügten Komponenten sind dabei Kopien des ursprünglichen Elements und können daher frei weiterbearbeitet werden. Nachteilig hieran ist, dass Änderungen an der ursprünglichen Komponente nicht auf die Kopie übertragen werden.

Macaw bietet außerdem die Möglichkeit, Global Styles anzulegen, mit denen visuelle Attribute wie Farben und Rahmen auf beliebige Elemente übertragen werden können. Bearbeitet man einen solchen Global Style, werden alle Elemente aktualisiert, die diesen Style zugewiesen haben. Entfernt man den Style, behält das Element die Eigenschaften des Styles.

Demnach eignet sich Macaw insgesamt dazu, Webseiten mit mehreren Unterseiten zu erstellen. Die meisten dazu benutzten Features sind jedoch wenig durchdacht, da z.B. Layouts und Komponenten zwar dupliziert werden können, aber keine Möglichkeit besteht, diese ohne manuelles Nacharbeiten in allen Instanzen zu ändern. So müsste z.B. ein im Fußbereich hinzugefügter Link auf jeder Seite einzeln hinzugefügt werden, wenn der ursprüngliche Fußbereich bereits auf mehrere Seiten dupliziert wurde.

### 5.1.8 Validität und Semantik (Wichtigkeit: hoch)

Der von Macaw generierte Code ist sowohl für HTML als auch für CSS nach Standards des W3C valide, es sind keine Fehler festzustellen.

Im Bereich der Sinnhaftigkeit und Semantik offenbart das Programm dann aber große Schwächen: HTML-Elemente werden grundsätzlich als `div` (für Elemente oder Container) oder `p` (bei Einsatz des Textwerkzeuges) erstellt (siehe Abbildung 4), viele weitere häufig verwendete semantische Elemente bietet das Programm nicht. Im Programm kann zwar z.B. eine Liste erstellt werden, indem der Name des Containers auf `ul` oder `ol` geändert wird und alle Text-Elemente innerhalb auf `li`, jedoch ist dies zeitaufwändig und erfordert zumindest grundlegende HTML-Kenntnisse.

```
<div class="container clearfix">
  <p class="text text-1">Item 1</p>
  <p class="text text-2">Item 1</p>
  <p class="text text-3">Item 1</p>
  <p class="text text-4">Item 1</p>
</div>

```

Abbildung 4: Generische HTML-Elemente und Klassennamen in Macaw

Wie bereits in Abschnitt 5.1.5 erwähnt schachtelt Macaw gezeichnete Elemente nicht automatisch, die hieraus entstehende flache Struktur geht auch in das generierte HTML über. Wenn nicht über den in Macaw vorhandenen Elementbaum genau nachgearbeitet wird, werden Elemente nicht korrekt ineinander verschachtelt, was je nach Struktur zu Problemen in der Darstellung führt.

Auch im Bereich CSS-Generierung zeigt das Programm große Schwächen. Trotz manueller Benennung der Elemente nach BEM, nummeriert das Programm alle Elemente mit gleicher Klasse über alle Seiten hinweg durch (.nav\_\_item-1, .nav\_\_item-2, etc.), um diese dann beispielsweise im Falle der Navigationselemente individuell an der einer bestimmten Position fest zu positionieren, anstatt sie im normalen Textfluss zu belassen (siehe Abbildung 5). Auch Elemente, die mit Macaw eigenen Tools speziell ausgerichtet wurden (z.B. „Horizontal angrenzen“-Button), werden dann zwar visuell entsprechend ausgerichtet, im generierten Code ist diese Einstellung dann jedoch irrelevant, Elemente werden weiterhin einzeln ausgerichtet.

```
.text-1 {  
  margin-right: 246px;  
}  
  
.text-2 {  
  clear: both;  
  margin: -1.375rem 159px 0 0;  
}  
  
.text-3 {  
  clear: both;  
  margin: -1.375rem 72px 0 0;  
}  
  
.text-4 {  
  clear: both;  
  margin: -1.375rem -15px 0 0;  
}
```

Abbildung 5: Nummerierung von Elementen und Positionierungsschwierigkeiten in Macaw

Werden Global Styles eingesetzt, so taucht der Name des Global Styles zwar als Klasse im HTML auf, im CSS werden die entsprechenden Eigenschaften jedoch einzeln und wiederholt den durchnummerierten Elementen zugewiesen, was den Code unnötig aufbläht und redundant macht.

Insgesamt ist der von Macaw generierte Code also zwar valide, aber weder in HTML noch CSS semantisch korrekt und sinnvoll. Durch redundante Code-Teile ist der Code außerdem deutlich länger als nötig, was die Ladezeit der Webseite erhöht. Auch für ein späteres

manuelles Nacharbeiten ist der Code ungeeignet, da große Teile umgeschrieben werden müssten, um den Code effektiv wartbar zu machen.

#### **5.1.9 Einsatz im Designprozess (Wichtigkeit: gering)**

Der Einsatz von Macaw im Designprozess ist bedingt möglich. Soll kein Code generiert werden und ist das Design nicht zu komplex, ist das Programm für schnelle Mock-Ups durchaus geeignet, da das Zeichnen der Elemente ähnlich zu Photoshop erfolgt und ähnliche Werkzeuge vorhanden sind. Sobald die Aufgaben jedoch komplexer werden, mangelt es Macaw an Funktionalität. Des Weiteren ist die Bedienung des Programms an vielen Stellen wenig intuitiv, sodass der Verwendung eine lange Einarbeitungszeit vorausgehen müsste, wodurch gegenüber der Erstellung mit Photoshop nur wenige Vorteile entstehen.

#### **5.1.10 Erstellung von Click-Prototypen (Wichtigkeit: hoch)**

Aufgrund der bereits aufgelisteten Schwächen ist der generierte Code zwar theoretisch für einen Click-Prototypen gut genug, jedoch auf keinen Fall im späteren Verlauf eines Projektes weiterverwertbar.

Für Click-Prototypen gilt daher ähnliches wie für den Designprozess: da im Programm mehrere Seiten erstellt werden können und die Verlinkung einfach möglich ist, ist die Erstellung eines Click-Prototypen grundsätzlich möglich. In der praktischen Anwendung ist jedoch auch hier der Vorteil gegenüber dem Designen in Photoshop und Erstellung des Click-Prototypen über Plattformen wie InVision denkbar gering, da die Verwendung eines neuen Programms erlernt werden muss und sonst nur wenige Vorteile für den Einsatz sprechen.

#### **5.1.11 Einsatz im Entwicklungsprozess (Wichtigkeit: hoch)**

Aufgrund der sehr schlechten Code-Qualität sowohl in HTML als auch in CSS ist die Verwendung von Macaw im Entwicklungsprozess nicht denkbar. Der Zeitaufwand, der für das manuelle Nacharbeiten am Code aufgebracht werden müsste, ist größer als die Zeit, die ein Entwickler der Agentur durchschnittlich benötigen würde, um den gleichen Code von Hand zu erstellen. Die Verwendung des von Macaw generierten Codes ist damit nicht möglich.

## **5.2 PINEGROW**

### **5.2.1 Dokumentation und Tutorials (Wichtigkeit: mittel)**

Pinegrow bietet eine umfangreiche englischsprachige Dokumentation[49], die alle vorhandenen Features dokumentiert. Die Dokumentation liegt in Textform mit unterstützenden Screenshots vor.

In der vorhandenen Dokumentation sind an verschiedenen Stellen Video-Tutorials eingebettet. Weiterhin gibt es zwei umfangreiche Tutorials, die in Text- und Video-Form das Erstellen einer Webseite in Pinegrow anhand von zwei Beispielen erklären.

### **5.2.2 Bestehende Projekte (Wichtigkeit: mittel)**

Pinegrow arbeitet auf Basis von regulären HTML- und CSS-Dateien, daher können bestehende Code-Projekte geöffnet werden. Voraussetzung dafür ist, dass die Seiten als semi-statische Seiten vorhanden sind. PHP- oder Template-Dateien kann das Programm zwar öffnen, die entsprechenden Befehle oder Ausdrücke werden jedoch nicht ausgeführt, sondern im Klartext angezeigt. Das Programm versucht aktiv, im Code verwendete Elemente (z.B. Bootstrap- oder Foundation-Komponenten) auf eigene Funktionalität abzubilden, um diese verwenden zu können.

Bestehende Projekte können so bearbeitet werden, allerdings ist es grundsätzlich am einfachsten, Projekte direkt in Pinegrow zu erstellen, da so alle Funktionen direkt verfügbar sind.

### **5.2.3 CSS-Tools und -Frameworks (Wichtigkeit: hoch)**

Pinegrow bietet die Möglichkeit, LESS-Variablen und Funktionen zu verwenden. Die Zuweisung der Variablen erfolgt dabei übersichtlich über ein separates Menü. Das Programm verhält sich hier intelligent, da der LESS-Parser nur eingeschaltet wird, wenn mindestens eine Variable benutzt wird.

Als Frameworks stehen u.a. Bootstrap und Foundation zur Verfügung, auch andere Bibliotheken wie Materialize[50] werden unterstützt. Dabei sind alle von den Frameworks zur Verfügung gestellten Komponenten im Komponenten-Menü verfügbar und können von dort via Drag und Drop auf die Arbeitsfläche gezogen werden. Anschließend können die Komponenten wie normale Elemente weiterbearbeitet werden.

### **5.2.4 JavaScript-Tools und -Frameworks (Wichtigkeit: gering)**

Benutzt man Komponenten aus zur Verfügung gestellten Frameworks, stellt Pinegrow ebenfalls das dafür benötigte JavaScript zur Verfügung. Eigene JavaScript-Dateien können eingebunden werden, ein Build-System ist nicht integriert. Da Pinegrow jedoch das parallele Bearbeiten in einem Code-Editor ermöglicht, wäre das Bearbeiten von JavaScript im Code-Editor mit separatem Build-System denkbar, um dann die kompilierte Datei von Pinegrow einbinden zu lassen.

### **5.2.5 Interface (Wichtigkeit: hoch)**

Pinegrow ist sowohl als Desktop-Anwendung als auch als Web-App[51] verfügbar, wobei die Web-App jedoch einen eingeschränkten Funktionsumfang hat. Die Bedienoberfläche von

Pinegrow weist im Gegensatz zu Macaw keine große Ähnlichkeit zu Photoshop vor. Das Standardlayout beinhaltet ein Menü mit mehreren Tabs für Komponenten, CSS, etc. auf der linken Seite, während der Seitenbaum sich auf der rechten Seite befindet.

Im Komponenten-Menü befinden sich alle für den Seitenaufbau verfügbaren Komponenten. Dabei stehen alle gängigen HTML-Elemente kategorisiert zur Verfügung und können per Drag und Drop auf die Zeichenfläche gezogen werden. Beim Erstellen der Seite kann ein Framework ausgewählt werden (siehe Abbildung 6), dessen Komponenten dann ebenfalls im Komponenten-Menü zur Verfügung stehen.

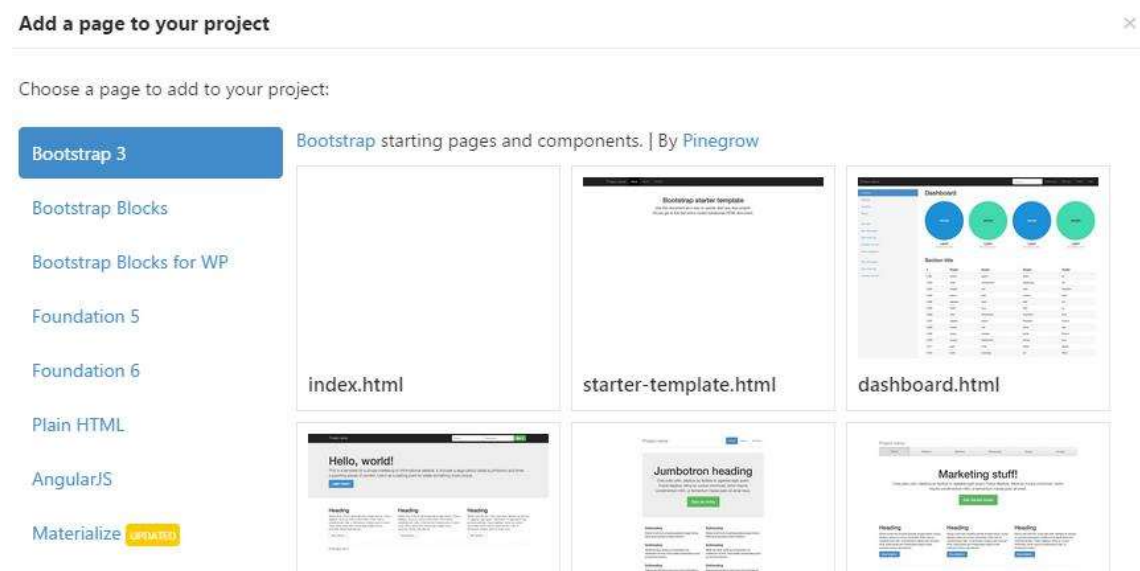


Abbildung 6: Framework-Auswahl in Pinegrow

Wählt man auf der Zeichenfläche ein Element aus, so werden seine Eigenschaften im Eigenschaften-Menü auf der linken Seite angezeigt. In diesem Menü können schnell und einfach Einstellungen wie Link-Ziel, Tooltips, und `data`-Attribute festgelegt werden. Insbesondere bei Verwendung des Bootstrap-Frameworks gibt es hier zahlreiche Einstellungsmöglichkeiten, die von den verschiedenen Bootstrap-Komponenten zur Verfügung gestellt werden.

Im CSS-Menü können vorhandene CSS-Regeln bearbeitet und neue Regeln hinzugefügt werden. Dabei unterscheidet sich das eigentliche Bearbeiten der Regel vom Zuweisen der Regel zu einem Element. Die Regel (quasi eine CSS-Klasse) kann dem Element im Eigenschaften-Menü zugewiesen werden, muss dann aber über das CSS-Menü separat bearbeitet werden. Dieser Vorgang ist holprig und wenig intuitiv. Außerdem bietet Pinegrow keinerlei visuelle Eingabefelder für Attribute wie Abstände und Dimensionen – alle Werte müssen manuell über Textfelder eingegeben oder über Auswahlfeldern ausgewählt werden. Die Werte entsprechen dabei exakt einer Auswahl von möglichen CSS-Attributen, sodass



Benutzer mit CSS-Kenntnissen sich schnell zurechtfinden, Designer ohne Code-Kenntnisse aber eine signifikant höhere Einarbeitungszeit benötigen würden.

Generell gilt, dass die Hauptfunktionen von Pinegrow einfach und intuitiv zu benutzen sind. Das einfache Hinzufügen von Elementen durch Drag und Drop wird durch den Seitenbaum auf der rechten Seite unterstützt, über den Elemente angeordnet werden können. Für tiefergehende Funktionalität ist ein Blick in die Dokumentation unabdingbar, da einige Funktionen wie z.B. das Zuweisen der CSS-Klassen und -Eigenschaften versteckt und wenig intuitiv zu bedienen sind. Nach kurzer Einarbeitungszeit ist das Interface jedoch klar verständlich und einfach zu bedienen. Abzüge gibt es für die mitunter umständliche Zuweisung von Eigenschaften und das Fehlen von visuellen Eingabefeldern.

### 5.2.6 Responsive Design (Wichtigkeit: mittel)

Pinegrow verwendet standardmäßig die vom verwendeten Framework definierten Breakpoints, lässt dem Benutzer aber die Wahl, diese zu verändern oder beliebige Breakpoints hinzuzufügen. Als besonderes Feature bietet Pinegrow die Möglichkeit, im Arbeitsbereich verschiedene Ansichten einer Seite mit verschiedenen Breiten darzustellen. Wird die Hauptansicht geändert, so übernehmen die Ansichten die Änderungen automatisch. So lässt sich eine Webseite schnell und unkompliziert für verschiedene Auflösungen testen und optimieren.

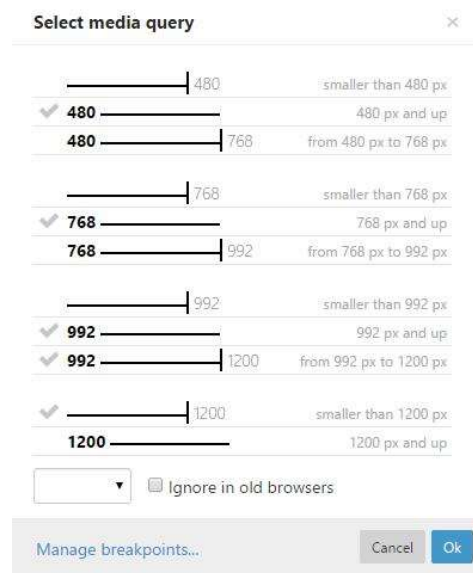


Abbildung 7: Media Query Wizard in Pinegrow

Die eigentliche Verwendung der Breakpoints erfolgt über einen einfachen Wizard für die entsprechende CSS-Regel (siehe Abbildung 7). Dabei ist die Auswahl sehr intuitiv und ermöglicht die Berücksichtigung des Mobile-First-Ansatzes.

### 5.2.7 Multi-Site und Elementbibliotheken (Wichtigkeit: mittel)

Innerhalb eines Projektes können beliebig viele Seiten angelegt werden, die standardmäßig als HTML-Dateien gespeichert werden. Pinegrow bietet außerdem die Möglichkeit, Master Pages anzulegen, die als Seitenlayout-Templates fungieren. In diesen Master Pages können dann bearbeitbare Bereiche und Attribute definiert werden. Einzelnen Seiten können diese Master Pages dann zugewiesen, Attribute gewählt und Bereiche mit Inhalt gefüllt werden.

Weiterhin können Smart Components definiert werden, die ähnlich zu Master Pages bearbeitbare Bereiche, Attribute und Klassen bieten. Die Smart Components können in Bibliotheken organisiert und bei Bedarf über mehrere Projekte hinweg verwendet werden. So ist es z.B. möglich, einen Styleguide als Projekt zu erstellen, und diesen dann als Bibliothek in Webseiten-Projekte zu laden, um die erstellten Komponenten dort zu verwenden.

Insgesamt bieten Master Pages und Smart Components zusammen ein sehr wertvolles Werkzeug, um auch komplexe Webseiten mit Pinegrow zu erstellen. Die Handhabung beider Funktionen ist nach Lesen der Dokumentation einfach und stellt sich nach kurzer Einarbeitungszeit intuitiv und produktiv dar.

### 5.2.8 Validität und Semantik (Wichtigkeit: hoch)

Da die verwendeten Elemente in der Komponenten-Bibliothek bereits Semantik haben, ist der Benutzer auch für die Semantik des generierten HTMLs selbst verantwortlich. Grundwissen über HTML-Elemente ist zwar notwendig, ermöglicht aber den problemarmen Aufbau eines semantisch sinnvollen DOMs. Da Elemente wie Überschriften und Formulare ebenfalls bereits mit entsprechendem Styling vorhanden sind, ergibt es für den Benutzer auch in dieser Hinsicht Sinn, diese anderen, generischen Elementen wie `div` oder `p` vorzuziehen. Lediglich beim Einordnen in den Seitenbaum müssen gelegentlich zusätzliche Arbeitsschritte durchgeführt werden, um die Elemente sinnvoll zu schachteln.

Der von Pinegrow generierte Code ist grundsätzlich valide. Da voller Zugriff auf den Code möglich ist, kann dieser aber durch manuelles Eingreifen invalid werden.

Die Namen der verwendeten CSS-Klassen werden vollständig vom Benutzer festgelegt, daher liegt auch hier die Verantwortung in dessen Hand. Insgesamt unterscheidet sich das von Pinegrow generierte HTML und CSS lediglich geringfügig von dem, was ein Entwickler manuell zum entsprechenden Design erstellt hätte. Allerdings sind auch ohne direkte Bearbeitung des Codes zumindest grundlegende HTML- und CSS-Kenntnisse erforderlich, um semantisch sinnvollen Code zu generieren.

### **5.2.9 Einsatz im Designprozess (Wichtigkeit: gering)**

Da das Programm hauptsächlich als visueller HTML-Baukasten und CSS-Editor fungiert und nahezu alle Design-Eigenschaften über CSS-ähnliche Eingabefelder erfolgen, ist der grundsätzliche Prozess der Design-Erstellung aufwändig und setzt HTML- und CSS-Kenntnisse voraus. Da Elemente nicht problemlos und ohne großen Aufwand gezeichnet werden können, ist das Programm auch für schnelle Mock-Ups ungeeignet.

### **5.2.10 Erstellung von Click-Prototypen (Wichtigkeit: hoch)**

Click-Prototypen können zwar mit Pinegrow erstellt werden, allerdings gibt es nur wenig Unterschied zwischen einer fertig entwickelten Webseite und einem Click-Prototypen, da alle Elemente zumindest bis zu einem gewissen Grad funktionell sind. Die Erstellung eines Click-Prototypen auf der Basis von Screenshots ist denkbar, allerdings wäre hier der Vorteil gegenüber Plattformen wie InVision gering. Für Hybrid-Prototypen, die teilweise aus Screenshots und teilweise aus funktionalen Elementen (z.B. Slider) bestehen, ist der Einsatz jedoch denkbar.

### **5.2.11 Einsatz im Entwicklungsprozess (Wichtigkeit: hoch)**

Durch die bereits in Abschnitt 5.2.9 erwähnte Verwendung als visueller HTML-Baukasten und CSS-Editor sowie die Möglichkeit, zwischen Programm und Code-Editor beliebig zu wechseln, kann Pinegrow Entwickler bei der Arbeit unterstützen. Insbesondere bei der Erstellung der HTML-Struktur bietet das Programm große Vorteile gegenüber der manuellen Erstellung, da Elemente einfach und schnell per Drag und Drop hinzugefügt werden können. Durch das Erstellen von Komponenten können wiederkehrende Elemente problemlos als solche markiert und vereinfacht wiederverwertet werden.

Auch beim Responsive Design hilft Pinegrow durch flexible Breakpoints und das gleichzeitige Anzeigen mehrerer Ansichten in verschiedenen Auflösungen. Abstriche gibt es beim Gestalten von Elementen, da die Eingabe manuell über Textfelder erfolgt und die verfügbaren Felder CSS-Eigenschaften 1:1 abbilden. Für erfahrene Entwickler ist hier wenig bis keine Zeitersparnis zu erwarten. Der Einsatz ist daher eher parallel zu einem normalen Code-Editor denkbar, um Strukturen aufzubauen, verschiedene Auflösungen zu testen und Inhalte hinzuzufügen.

## **5.3 WEBFLOW**

### **5.3.1 Dokumentation und Tutorials (Wichtigkeit: mittel)**

Im Gegensatz zu Macaw und Pinegrow gibt es zu Webflow nur sehr wenig reine Text-Dokumentation. Die Grundlagen des Programms werden fast ausschließlich in einer großen Menge an Video-Tutorials erklärt. Für häufig gestellte Fragen gibt es eine ausführliche FAQ-Sammlung, die über das online verfügbare Hilfe-Center[52] abrufbar ist.

Die von Webflow selbst zur Verfügung gestellten Video-Tutorials erklären alle Features des Programms und sind entweder über die Webseite oder direkt aus der App heraus abrufbar. Zusätzlich zu den Erklärungen der Features gibt es eine große Anzahl an Tutorials, die Beispielseiten exemplarisch nachbauen und so die Funktionalität der App an Praxisbeispielen erläutern.

### **5.3.2 Bestehende Projekte (Wichtigkeit: mittel)**

Webflow bietet nach eigener Aussage keinen Import von bestehenden Projekten an, schließt aber ein solches Feature für die Zukunft nicht aus[53]. Bis ein solches Feature verfügbar wäre, müssten bestehende Webseiten also in Webflow erneut erstellt werden.

### **5.3.3 CSS-Tools und -Frameworks (Wichtigkeit: hoch)**

Webflow bietet wie auch Macaw das Photoshop-ähnliche Speichern von Farben als Swatches an. Von Entwicklern oft durch LESS- oder SASS-Mixins generierte Eigenschaften wie Schatten oder Farbverläufe können über spezielle Eingabefelder mit Reglern erstellt werden, sodass entsprechende Mixins durch Webflow-eigene Funktionalität überflüssig werden. LESS- oder SASS-Code werden bisher von Webflow nicht generiert, dieses Feature ist aber ebenfalls für die Zukunft nicht ausgeschlossen[54][55].

Der von Webflow generierte Code basiert lose auf Bootstrap, aus dessen Code-Basis u.a. die Breakpoints übernommen wurden[56]. Auch einzelne Komponenten wie Navbar oder Tabs erinnern an die in Bootstrap enthaltenen Komponenten. Außer den direkt in Webflow enthaltenen ausgewählten Komponenten gibt es jedoch keine Möglichkeit, Komponenten aus Bootstrap oder Foundation direkt zu verwenden.

### **5.3.4 JavaScript-Tools und -Frameworks (Wichtigkeit: gering)**

Webflow generiert automatisch JavaScript, welches die zur Verfügung stehenden Komponenten steuert und mit Funktion versieht. Auch wenn nicht alle Komponenten verwendet werden, ist die Dateigröße dieses Skriptes leider verhältnismäßig groß, was die Ladezeit der Webseite erhöht. Die von Webflow direkt gehosteten und ausgelieferten Seiten arbeiten bereits mit minimierten Inhalten. Erfolgt die Publikation über Webflow, wird dieser Teil des Build-Prozesses also obsolet. Wird der Code zur Weiterverarbeitung exportiert, sind die Inhalte mit Ausnahme des Webflow-eigenen JavaScripts nicht minimiert, sodass ein zusätzlicher Build-Prozess aufgeschaltet werden müsste.

### **5.3.5 Interface (Wichtigkeit: hoch)**

Webflow ist nur als Web-App verfügbar. Die Aufteilung des Screens ähnelt Photoshop, mit Hauptwerkzeugen links und den meisten Einstellungen auf der rechten Seite. Insgesamt wirkt das Interface logisch strukturiert und aufgeräumt, es beinhaltet im Ursprungszustand nur wenige Elemente, die in verschiedene Kategorien unterteilt sind.

In der Werkzeugleiste links befindet sich unter anderem die Element-Bibliothek, die eine kategorisierte Sammlung von Elementen enthält, die der Benutzer der Seite hinzufügen kann. Dabei sind die Elemente eine Mischung aus tatsächlichen HTML-Elementen und zusammengesetzten Komponenten (z.B. Navbar oder Slider). Alle Elemente sind mit Icons versehen, die die Funktionalität gut verdeutlichen. Zu allen Elementen sind hier wie auch in der gesamten App Soforthilfen verfügbar, die die Funktion des Elements kurz erklären und, falls vorhanden, auf entsprechende Tutorials verlinken (siehe Abbildung 8). So erklären sich Elemente des Interfaces selbst, ohne dass der Benutzer die App verlassen muss. Dies erleichtert und verkürzt insbesondere die Einarbeitungsphase.

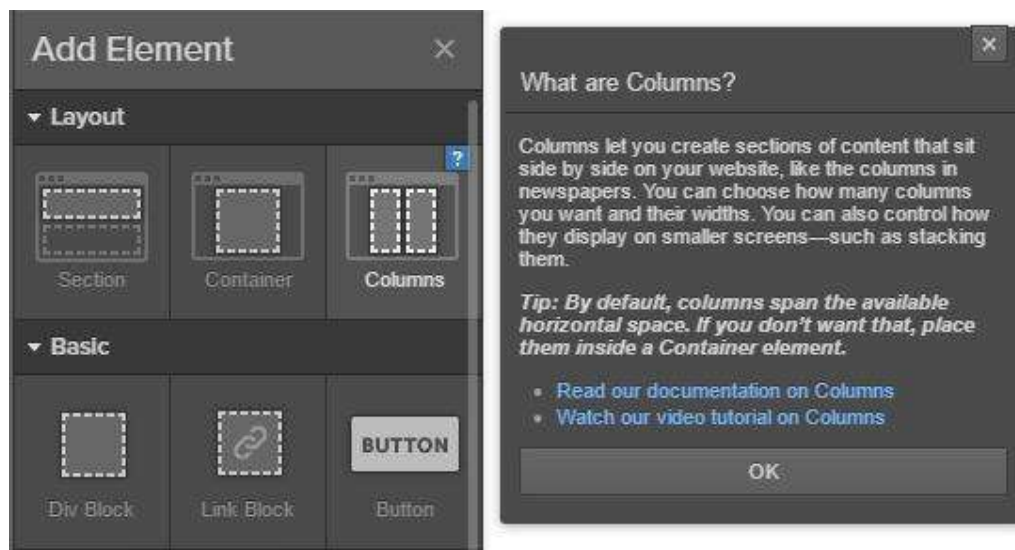


Abbildung 8: Element hinzufügen Dialog mit In-App-Hilfe in Webflow

Beim Einfügen der Elemente per Drag und Drop ist direkt durch verschiedene Umriss ersichtlich, an welche Position im DOM das Element gesetzt wird. Elemente können ebenfalls in den auf der rechten Seite befindlichen Baum gezogen werden. Dieser bietet auch die Möglichkeit, vorhandene Elemente umzusortieren und ist, im Gegensatz zu den anderen beiden Programmen, keine tatsächliche Abbildung des HTML-Baumes. Stattdessen bildet der Baum die Struktur der Webflow-Elemente ab, die dann von Webflow bei der Generierung des Codes in HTML-Elemente transformiert werden. Somit ist die Bedienung für einen Benutzer ohne Code-Kenntnisse deutlich einfacher.

Um einzelne Elemente zu bearbeiten, muss diesen eine Klasse zugewiesen werden. Dabei kann entweder aus bereits vorhandenen Klassen gewählt werden oder eine neue Klasse vergeben werden. Elemente mit der gleichen Webflow-Klasse erhalten immer das gleiche Design, analog zu CSS-Klassen. Die Benennung der Webflow-Klassen ist allerdings nicht an die Regeln von CSS-Klassen gebunden, sodass die Verwendung auch für nicht-Entwickler ohne große Probleme möglich ist.

Das Design von Elementen kann über visuelle Input-Felder eingestellt werden, ähnlich zu Macaw. Besonders hervorzuheben ist dabei der *Advanced*-Modus, der die sonst sehr übersichtliche Auswahl an Eigenschaften (siehe Abbildung 9) um eine Vielzahl an Optionen ergänzt (siehe Abbildung 10). So wirkt das Interface im Grundzustand übersichtlich und aufgeräumt, filigrane Einstellungen sind aber bei Bedarf ebenso möglich.



Abbildung 9: Einstellungs-Dialog, normaler Modus in Webflow

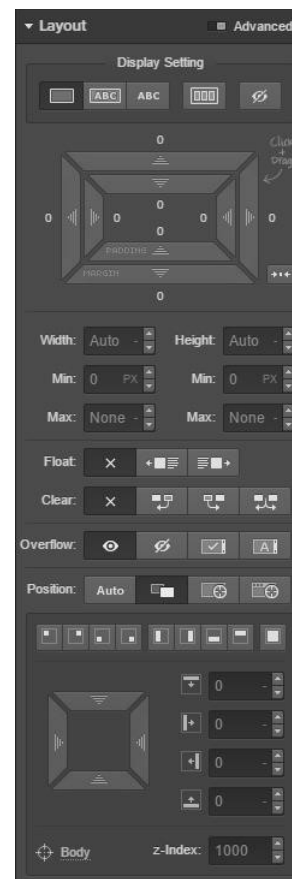


Abbildung 10: Einstellungs-Dialog, Advanced-Modus in Webflow

Insgesamt besteht Webflow durch ein übersichtliches Design, in dem die meisten Funktionen selbsterklärend und intuitiv verwendbar sind. Das Hinzufügen von Elementen zur Webseite geht auch ohne umfangreichere Einarbeitung flüssig von der Hand, und auch komplexere Design-Eigenschaften wie Schatten sind direkt über visuelle Bedienelemente einfach hinzufügbare. Zusätzlich unterstützt wird der Designprozess durch vorgefertigte Komponenten wie Navbar oder Slider, welche schnell bereits funktionstüchtig eingebunden werden können, um dann weiter gestaltet zu werden.

### 5.3.6 Responsive Design (Wichtigkeit: mittel)

Beim Starten der App fallen direkt die über der Arbeitsfläche liegenden Buttons zur Anpassung der Auflösung auf. Diese sind dabei mit Icons (Laptop, Tablet, Smartphone) versehen, die die Breite der Breakpoints und ihre Anwendung verdeutlichen. Nachteilig ist dabei, dass die Breakpoints nicht veränderbar sind. Auch der Mobile-First-Ansatz ist mit Webflow nicht verfolgbar, da alle Styles von größeren auf kleinere Auflösungen vererbt werden und dann für diese angepasst werden können.

Die Bedienung der Responsive Design Werkzeuge ist dafür simpel, insbesondere Spaltengrößen und Sichtbarkeiten können für die verschiedenen Auflösungen einfach eingestellt werden. Die erwähnten Icons werden konsequent als platzsparende und visuell prägnante Methode zur Unterscheidung der Breakpoints verwendet.

Praktisch ist außerdem, dass die Webflow-eigenen Komponenten wie die Navbar schon mit definiertem Verhalten auf kleineren Auflösungen versehen sind, welches leicht anpassbar ist. So kann z.B. die Richtung der Off-Canvas-Navigation leicht konfiguriert werden, was die Arbeit des Designers erleichtert.

Insgesamt sind die Tools des Responsive Designs in Webflow hervorragend aufeinander abgestimmt und intuitiv verwendbar, lediglich die unveränderlichen Breakpoints und das Fehlen des Mobile-First-Ansatzes sind als negative Punkte zu nennen.

### 5.3.7 Multi-Site und Elementbibliotheken (Wichtigkeit: mittel)

Webflow erlaubt das Erstellen von beliebig vielen Unterseiten, die bei Bedarf in Ordner sortiert werden können. Auffällig beim Erstellen der Seiten ist, dass direkt zusätzliche Informationen wie eine Beschreibung eingegeben werden können, welche dann z.B. im entsprechenden `meta`-Tag im generierten HTML zum Einsatz kommt. Die verschiedenen Seiten können über Link-Elemente einfach verlinkt werden.

Elemente oder Elementgruppen können beliebig in sogenannte Symbole konvertiert werden, die dann aus der Symbol-Bibliothek in verschiedene Seiten eingefügt werden können. Wird ein Symbol an einer Stelle im Projekt geändert, so werden alle anderen Instanzen ebenfalls automatisch aktualisiert. So können z.B. Header und Footer leicht auf verschiedenen Seiten eingesetzt werden.

Durch die Vergabe von Klassen an Elemente können seitenübergreifende Styles generell einfach erstellt werden, ein separates Abspeichern von Global Styles wie z.B. bei Macaw ist nicht notwendig.

### 5.3.8 Validität und Semantik (Wichtigkeit: hoch)

Der generierte HTML- und CSS-Code ist nach neusten Standards valide. Vorausgesetzt, der Benutzer verwendet die von Webflow angebotenen Elemente korrekt, ist der HTML-Code auch semantisch sinnvoll aufgebaut. Insbesondere fällt auf, dass nur sehr wenige zusätzliche `div`-Elemente verwendet werden. Kleinere Abzüge gibt es für das Nichtverwenden der HTML5-Elemente wie `section` für Abschnitte. Ein Abschnitt wird stattdessen mit einem `div` simuliert, der die Klasse `section` erhält.

Die sinnvolle Bezeichnung der Klassennamen für die Elemente hängt vom jeweiligen Nutzer ab, da dieser die Bezeichnungen selbst vergibt. Die Verwendung von BEM ist möglich, da Unterstriche in Klassennamen verwendet werden können.

Insgesamt ist der von Webflow generierte Code von sehr hoher Qualität. HTML-Elemente werden korrekt verwendet und das generierte CSS ist sinnvoll strukturiert.

### 5.3.9 Einsatz im Designprozess (Wichtigkeit: gering)

Insbesondere für wenig komplexe Layouts können durch die sinnvoll vordefinierten Komponenten sehr schnell Designs erstellt werden, daher ist Webflow für den Einsatz im Designprozess anstelle von Photoshop durchaus zu empfehlen. Das Interface ist klar aufgebaut, und bietet ähnlich wie in Photoshop auf Webdesign zugeschnittene Bedienelemente, mit denen die wichtigsten Design-Einstellungen schnell und einfach auch ohne große Einarbeitungszeit von der Hand gehen. Auch für Projekte, für die eine automatische Code-Generierung nicht in Frage kommt, ist der Einsatz als Photoshop-Ersatz durchaus denkbar. Da das Interface im Gegensatz zu Photoshop auf Webdesign ausgerichtet ist, kann die Produktivität des Designprozesses gesteigert werden, insbesondere im Hinblick auf Responsive Design und Design von mehreren Unterseiten.

### 5.3.10 Erstellung von Click-Prototypen (Wichtigkeit: hoch)

Ähnliches wie für den Designprozess gilt ebenfalls für die Erstellung von Click-Prototypen. Durch das schnelle Zusammenstellen der Seite aus vordefinierten Elementen können Click-Prototypen zeiteffizient und einfach ohne Verwendung von Screenshots erstellt werden. Da Webflow Funktionalität zum Erstellen von mehreren Seiten sowie Hilfestellung zum Erstellen von Animationen innerhalb einer Seite bietet, kann die App sowohl für ein- als auch für mehrseitige Click-Prototypen verwendet werden. Wird die Hosting-Möglichkeit der Webflow-Plattform genutzt, kann für zusätzliche Zeitersparnis sogar das Hochladen des Click-Prototypen auf einen Server der Agentur übersprungen werden.

### 5.3.11 Einsatz im Entwicklungsprozess (Wichtigkeit: hoch)

Generell ist die App darauf ausgelegt, Benutzern, die keine HTML- und CSS-Kenntnisse besitzen, die Benutzung der App zu ermöglichen, weshalb zur Benutzung selbst nur



grundlegende Code-Kenntnisse benötigt werden. Da Webflow außerdem weder Codezugriff vor dem Code-Export noch einen Re-Import zulässt, ist die App im Entwicklungsprozess nur vor eventueller manueller Code-Erstellung einsetzbar.

Die Qualität des erstellten Codes ist jedoch hochwertig genug, um diesen ohne größeren Bearbeitungsaufwand weiterverwenden zu können. So könnte z.B. die Grundstruktur einer Seite in Webflow schnell erstellt werden, um anschließend den Code zu exportieren und manuell weiter zu entwickeln. Dies würde insbesondere bei Seiten mit vielen Inhalten eine deutliche Zeitersparnis gegenüber dem manuellen Erstellen des HTMLs bedeuten. Auch die direkte Veröffentlichung von kleineren Webseiten über das Webflow-eigene Hosting ist denkbar, da somit wertvolle Zeit gespart werden kann.

Als Sonderfall bietet Webflow für die Entwickler Vorteile, wenn der generierte Code zur Verwendung in einem kleineren CMS wie WordPress[57] vorgesehen ist. WordPress wird im Team hauptsächlich für kleinere Seiten eingesetzt, die lediglich einen Blog und/oder wenige durch den Kunden selbst zu pflegende Inhalte umfassen. Die Einrichtung eines Blogs mit Kategorien und Autoren ist beim Verfolgen der verfügbaren Video-Tutorials innerhalb einer halben Stunde abgeschlossen, was einen großen zeitlichen Vorteil gegenüber der Einrichtung von WordPress bietet.

## 5.4 ZUSAMMENFASSUNG

In der folgenden Übersicht werden die in diesem Kapitel vorgestellten Ergebnisse tabellarisch zusammengefasst. Dazu werden die Begriffe „ja“ (erfüllt Kriterium komplett), „teilweise“ (erfüllt Kriterium, jedoch nur zu Teilen oder mit größeren Schwächen) und „nein“ (erfüllt Kriterium nicht) zur Kurzbeurteilung der Ergebnisse verwendet.

Anforderung	Macaw	Pinegrow	Webflow
#01 Dokumentation und Tutorials	ja	ja	ja
#02 Bestehende Projekte	nein	ja	nein
#03 CSS-Tools und Frameworks	nein	ja	teilweise
#04 JavaScript-Tools und Frameworks	nein	ja	teilweise
#05 Interface	teilweise	teilweise	ja
#06 Responsive Design	ja	ja	ja
#07 Multi-Site und Elementbibliotheken	teilweise	ja	ja

<b>#08</b> Validität und Semantik	nein	ja	ja
<b>#09</b> Einsatz im Designprozess	teilweise	nein	ja
<b>#10</b> Erstellung von Click-Prototypen	nein	teilweise	ja
<b>#11</b> Einsatz im Entwicklungsprozess	nein	teilweise	teilweise

*Tabelle 2: Zusammenfassung der Tool-Analyse*

## 6 DISKUSSION UND AUSBLICK

---

### 6.1 DISKUSSION

Nachdem die drei ausgewählten Tools im vorherigen Kapitel auf die Anforderungen des Anforderungskataloges hin untersucht wurden, werden diese Ergebnisse im Folgenden gegenübergestellt und bewertet.

#### 6.1.1 Allgemeines und Projektplanung

Für die drei getesteten Programme sind adäquate Dokumentationen und Tutorials vorhanden. Lediglich für Webflow sind Erklärungen zu Funktionalitäten fast ausschließlich als Video-Tutorials verfügbar. Dies erschwert das schnelle Nachschlagen von Informationen, da Videos nicht wie Text-Dokumentation durchsucht werden können. Diesen Nachteil gleicht die App jedoch durch die hervorragend umgesetzte In-App-Hilfe wieder aus, die dem Benutzer schnell und effektiv eine Kurzhilfe zur Verfügung stellt und zu relevanten Hilfe-Videos führt. Da diese Funktion insbesondere die Einarbeitungszeit deutlich verkürzt und die zur Verfügung stehenden Videos die entsprechenden Inhalte kurz und prägnant erklären, ist die Webflow In-App-Hilfe mit Video-Tutorials den Dokumentationen von Macaw und Pinegrow vorzuziehen.

Pinegrow ist das einzige Tool im Test, welches auch bestehende Projekte öffnen kann. Durch das direkte Bearbeiten von HTML- und CSS-Dateien mit Pinegrow ist das Programm für bestehende Projekte deutlich besser geeignet als Macaw und Webflow. In beiden Tools müssten Projekte aus der Vergangenheit erneut neu angelegt werden, um eine Weiterbearbeitung zu ermöglichen.

Als einziges der getesteten Tools bietet Pinegrow ebenfalls native Unterstützung für Frameworks wie Bootstrap und Foundation. Der Einsatz von LESS inklusive Variablen ist ebenfalls möglich, in Macaw und Webflow können lediglich Farben zur Wiederverwendung in Swatches abgespeichert werden. Im Gegensatz zu Macaw, das keinerlei native Unterstützung für Frameworks bietet, basiert der Webflow-Designer auf einer abgewandelten Form von Bootstrap, aus dem u.a. Breakpoints und einige Komponenten adaptiert wurden. Somit sind sowohl in Pinegrow als auch in Webflow vorgefertigte Komponenten vorhanden, während Macaw keinerlei Unterstützung in diesem Bereich bietet.

#### 6.1.2 Werkzeuge und Funktionalität

Webflow bietet von den drei getesteten Programmen das am meisten intuitiv zu verwendende Interface. Macaw und Pinegrow offenbaren hier Schwächen, da bei beiden essenzielle Funktionalitäten teilweise versteckt und/oder nur wenig intuitiv zu bedienen sind. Damit geht beim Aufbau eines Designs wertvolle Zeit verloren, da auch nach einer gewissen Einarbeitungszeit noch häufig die Dokumentation zu Rate gezogen werden muss.

Die drei getesteten Tools eignen sich hervorragend dafür, die erstellten Designs auch für Mobilgeräte verfügbar zu machen. Abzüge gibt es für Macaw und Webflow, da mit diesen im Gegensatz zu Pinegrow der Mobile-First-Ansatz nicht verfolgt werden kann. In Pinegrow fällt vor allem der Media Query Wizard positiv ins Auge, da mit diesem bestehende Regeln schnell und einfach unter Berücksichtigung des Mobile-First-Ansatzes für verschiedene Auflösungen angepasst werden können. Webflow punktet in diesem Bereich vor allem mit einem einfach gestalteten Interface, in dem Funktionalität durch Icons effizient verdeutlicht wird.

Insbesondere im Bereich der Seitenverwaltung zeigen alle Tools Vorteile gegenüber Photoshop, da mehrere Unterseiten einer Webseite einfach erstellt und untereinander verlinkt werden können. Pinegrow bietet mit Master Pages und Smart Components die beste Lösung. Hiermit können Template-artige Strukturen für Seitenlayout bzw. Komponenten zur beliebigen Verwendung in mehreren Seiten erstellt werden. In Webflow wird ein ähnliches Verhalten durch die Verwendung von Symbol-Elementen erreicht. Diese werden automatisch in allen Instanzen auf allen Unterseiten aktualisiert, wenn eine Änderung in einer beliebigen Instanz erfolgt. Lediglich Macaw bietet hier nur sehr unzureichende Funktionalität, da Elemente zwar abgespeichert und dupliziert werden können, ein automatisches Aktualisieren aller Instanzen aber nicht möglich ist. Dadurch geht bei nachträglichen Änderungen von auf mehreren Seiten vorhandenen Elementen viel Zeit verloren.

Auch beim Thema Validität und Semantik des generierten Codes können sich Pinegrow und Webflow gegenüber Macaw durchsetzen. Zwar produzieren alle drei Tools validen Code, allerdings ist der von Macaw generierte Code durchweg semantisch nicht logisch und redundant. Sowohl Pinegrow als auch Webflow produzieren hier deutlich hochwertigeren Code. Abstriche gibt es bei Pinegrow, da insbesondere CSS-Kenntnisse zur Benutzung des Programms unbedingt erforderlich sind, was die Benutzung des Programms auf Entwickler oder Designer mit Entwickler-Kenntnissen beschränkt. Bei der Benutzung von Webflow kommt der Benutzer dagegen komplett ohne HTML- und CSS-Kenntnisse aus.

### **6.1.3 Einsatzmöglichkeiten**

Aufgrund der mangelnden Funktionalität und des in Teilen wenig intuitiv zu bedienenden Interfaces von Macaw sowie der starken Ausrichtung von Pinegrow auf CSS-Code und dessen Mangel an visuellen Einstellungsfeldern, kommt sowohl für den Einsatz im Designprozess als auch für die Erstellung von Click-Prototypen nur Webflow in Frage. Als einziges Programm bietet Webflow einen ausreichenden Funktionalitätsrahmen, um dem Designer genügend kreative Freiheit zu lassen, aber trotzdem ein weiterverwertbares Ergebnis zu produzieren. Durch die gut durchdachte Komponenten-Bibliothek und das intuitive, durch die In-App-Hilfe unterstützte Interface lassen sich sowohl komplexe als auch einfache Designs umsetzen. Da das Interface, im Gegensatz zu Photoshop, für die Erstellung von Webseiten optimiert ist, ist

hier nach einer Einarbeitungsphase ein großer Zeitvorteil zu erwarten. Da der von Webflow generierte Code von guter Qualität ist, ist die App besonders für die Erstellung von Click-Prototypen geeignet, da der hierfür generierte Code weiterverwendet werden kann.

Im Entwicklungsprozess selbst eignet sich nur Pinegrow für die direkte Verwendung, da es als einziges Programm bidirektionales Bearbeiten des generierten Codes ermöglicht. Sowohl Macaw als auch Webflow bieten keine Möglichkeit, bearbeiteten Code wieder zu importieren. Pinegrow selbst eignet sich hier hauptsächlich als Hilfswerkzeug zur Unterstützung des manuellen Code-Schreibens, da insbesondere die Eingabemaske für CSS-Eigenschaften eher einen Nachteil bildet. Es ist zu erwarten, dass ein erfahrener Entwickler des Teams hier schneller manuell zum gewünschten Ergebnis kommt. Als Hilfswerkzeug für den Aufbau von Seitenstrukturen und Inhalten sowie für das gleichzeitige Anzeigen der Webseite in verschiedenen Auflösungen ist das Programm jedoch geeignet.

## **6.2 EMPFEHLUNG FÜR DIE AGENTUR**

Auf Basis der vorliegenden Arbeit und vorangegangenen Diskussion der Ergebnisse wird zur Verwendung in der Agentur die Web-App Webflow empfohlen.

Webflow erzielte im Test vor allem in den als hoch wichtig eingestuften Kriterien Interface, CSS-Tools und -Frameworks sowie Validität und Semantik gute Ergebnisse. Speziell für die Erstellung von Click-Prototypen, ein Kriterium welches von den befragten Kundenberatern mit einer höchstmöglichen Wichtigkeit von 10 bewertet wurde, ist die App besonders gut geeignet. Da Strukturen schnell erstellt werden können und sowohl Verlinkungen zwischen Unterseiten als auch Animationen auf Einzelseiten möglich sind, gelingt die Erstellung eines Click-Prototypen deutlich schneller als durch einen Entwickler bei Programmierung von Hand.

Abgerundet wird der positive Eindruck des Programms vom ebenfalls verfügbaren Hosting und CMS. Insbesondere für die Erstellung von kleineren Webseiten wie der Heimatland-Seite oder im Team bisher durch Verwendung von WordPress umgesetzte Webseiten ist das Tool hervorragend geeignet. Da die App auch ohne HTML- und CSS-Kenntnisse gut benutzbar ist, kann sie auch von Designern ohne Code-Erfahrung verwendet werden.

Webflow besitzt alle nötigen Eigenschaften, um das Team sowohl im Design- als auch im Entwicklungsprozess bei verschiedenen Arten von Projekten zu unterstützen, daher hängt die erfolgreiche und produktive Verwendung vor allem von der Bereitschaft des Teams ab, sich auf das Programm einzulassen. Insbesondere die Limitierung der Kreativität im Designprozess wurde in der initialen Befragung des Teams oft als Risiko genannt. Da durch die limitierte Anzahl an verfügbaren Elementen die Kombinationsmöglichkeiten begrenzt sind, ist gerade dieses Risiko genau zu beobachten. Eine genaue Analyse des jeweiligen Projektes und der

Vision des Designers für das zu erstellende Design ist nötig, um vorab abschätzen zu können, ob eine Realisierung des Projektes in Webflow möglich und sinnvoll ist. Dieses Risiko bietet jedoch gerade für Projekte mit geringem Budget eine Chance, da die verfügbaren Elemente klar definiert sind und keine Zeit für die Programmierung von Elementen aufgebracht werden muss, die einfach zu gestalten aber in der Entwicklung schwierig umzusetzen sind.

Insgesamt bietet Webflow für das Team die Möglichkeit, vor allem kleinere Projekte effizienter umzusetzen, besonders bei der Erstellung von Click-Prototypen schneller zum Ziel zu kommen sowie die Entwicklungszeiten insgesamt zu reduzieren.

### **6.3 AUSBLICK**

War die Qualität des WYSIWYG-Editoren generierten Codes vor wenigen Jahren noch sehr schlecht, so schneiden heute auf dem Markt verfügbare Tools deutlich besser ab. Die Untersuchung der Tools in dieser Arbeit zeigt jedoch, dass weiterhin Verbesserungsbedarf besteht. Selbst Programme wie Webflow, welche bereits hervorragenden Code generieren, haben z.B. im Bereich semantisches HTML noch Verbesserungspotenzial, was zweifelsohne ausgeschöpft werden sollte.

Ein Programm, welches sowohl volle Code-Kontrolle (wie Pinegrow) als auch ein intuitives, für nicht-Entwickler bedienbares Interface (wie Webflow) bietet, konnte im Rahmen dieser Arbeit nicht gefunden werden. Es bleibt abzuwarten, ob Pinegrow oder Webflow hier durch weitere Funktionalität nacharbeiten, oder ob ein anderes, noch nicht auf dem Markt erschienenenes Programm beide Anforderungen erfüllen kann.

Als potenzielles Programm, welches diese Lücke füllen könnte, ist hier Macaw Scarlet[58] zu nennen, welches sich ohne festes Veröffentlichungsdatum in der Entwicklung befindet. Macaw Scarlet verspricht volle Code-Kontrolle bei visuellem Arbeiten. Da die Macaw-Entwickler sich Anfang 2016 dem InVision-Projekt angeschlossen haben, ist ein Release von Macaw Scarlet derzeit allerdings nicht abzusehen[59].

Insgesamt bleibt es spannend zu beobachten, wie sich der Markt an WYSIWYG-Tools entwickelt. Es besteht sowohl Potenzial für die bestehenden Tools, als auch für neue Programme, sich im Markt zu etablieren.

## 7 LITERATURVERZEICHNIS

---

- 1 Home | G + P Glanzer + Partner Werbeagentur GmbH. Online verfügbar unter <http://www.werbeagentur-glanzer-und-partner.de/>, zuletzt geprüft am 24.07.2016.
- 2 Adobe: Foto-Editor | Adobe Photoshop CC. Online verfügbar unter <http://www.adobe.com/de/products/photoshop.html>, zuletzt geprüft am 19.06.2016.
- 3 Adobe: Software für Grafikdesign | Adobe Illustrator CC. Online verfügbar unter <http://www.adobe.com/de/products/illustrator.html>, zuletzt geprüft am 19.06.2016.
- 4 Sublime Text: The text editor you'll fall in love with. Online verfügbar unter <https://www.sublimetext.com/>, zuletzt geprüft am 19.06.2016.
- 5 Github: A hackable text editor for the 21st Century. Online verfügbar unter <https://atom.io/>, zuletzt geprüft am 19.06.2016.
- 6 Atlassian: Free Mercurial and Git Client for Windows and Mac | Atlassian SourceTree. Online verfügbar unter <https://www.sourcetreeapp.com/>, zuletzt geprüft am 19.06.2016.
- 7 Axosoft: Git GUI Client for Windows Mac and Linux. Online verfügbar unter <https://www.gitkraken.com/>, zuletzt geprüft am 19.06.2016.
- 8 Otto, Mark; Thornton, Jacob; contributors, and Bootstrap: Bootstrap · The world's most popular mobile-first and responsive front-end framework. Online verfügbar unter <http://getbootstrap.com/>, zuletzt geprüft am 24.07.2016.
- 9 Foundation | The most advanced responsive front-end framework in the world. (2016). Online verfügbar unter <http://foundation.zurb.com/>, zuletzt aktualisiert am 22.07.2016, zuletzt geprüft am 24.07.2016.
- 10 25 Useful WYSIWYG Editors Reviewed – Smashing Magazine (2008). Online verfügbar unter <https://www.smashingmagazine.com/2008/05/25-wysiwyg-editors-reviewed/>, zuletzt aktualisiert am 06.08.2008, zuletzt geprüft am 24.07.2016.
- 11 Fitzpatrick, Jason (2010): Five Best WYSIWYG HTML Editors. Online verfügbar unter <http://lifehacker.com/5579328/five-best-wysiwyg-html-editors>, zuletzt aktualisiert am 07.04.2010, zuletzt geprüft am 24.07.2016.
- 12 Brian Wood (2014): Next-Generation Responsive Web Design Tools: Webflow, Edge Reflow, Macaw. Online verfügbar unter <https://www.smashingmagazine.com/2014/05/next-generation-responsive-web-design-tools-webflow-edge-reflow-macaw/>, zuletzt geprüft am 19.06.2016.

- 13 Smashing Magazine – for professional Web Designers and Developers. Online verfügbar unter <https://www.smashingmagazine.com/>, zuletzt geprüft am 30.07.2016.
- 14 Macaw: The Code-Savvy Web Design Tool. Online verfügbar unter <http://macaw.co/>, zuletzt geprüft am 17.07.2016.
- 15 Responsive Webdesign | Download Adobe Edge Reflow CC (Preview). Online verfügbar unter <http://www.adobe.com/de/products/edge-reflow.html>, zuletzt geprüft am 30.07.2016.
- 16 Web design tool, CMS, and hosting platform | Webflow. Online verfügbar unter <https://webflow.com/>, zuletzt geprüft am 24.07.2016.
- 17 Macaw: The Code-Savvy Web Design Tool. Online verfügbar unter <https://www.kickstarter.com/projects/macaw/macaw-the-code-savvy-web-design-tool>, zuletzt geprüft am 30.07.2016.
- 18 W3C HTML (2016). Online verfügbar unter <https://www.w3.org/html/>, zuletzt aktualisiert am 31.03.2016, zuletzt geprüft am 30.07.2016.
- 19 Cascading Style Sheets (2016). Online verfügbar unter <https://www.w3.org/Style/CSS/>, zuletzt aktualisiert am 29.07.2016, zuletzt geprüft am 30.07.2016.
- 20 Team, The Core Less (2016): Getting started | Less.js. Online verfügbar unter <http://lesscss.org/>, zuletzt aktualisiert am 13.07.2016, zuletzt geprüft am 30.07.2016.
- 21 Sass: Syntactically Awesome Style Sheets (2016). Online verfügbar unter <http://sass-lang.com/>, zuletzt aktualisiert am 29.03.2016, zuletzt geprüft am 30.07.2016.
- 22 JavaScript (2016). Online verfügbar unter <https://developer.mozilla.org/en-US/docs/Web/JavaScript>, zuletzt aktualisiert am 03.07.2016, zuletzt geprüft am 30.07.2016.
- 23 PHP: Hypertext Preprocessor (2016). Online verfügbar unter <http://php.net/>, zuletzt aktualisiert am 30.07.2016, zuletzt geprüft am 30.07.2016.
- 24 Foundation, Node js (2016): Node.js. Online verfügbar unter <https://nodejs.org/en/>, zuletzt aktualisiert am 23.07.2016, zuletzt geprüft am 30.07.2016.
- 25 Adobe: DTP, DTP-Software | Adobe InDesign CC. Online verfügbar unter <http://www.adobe.com/de/products/indesign.html>, zuletzt geprüft am 19.06.2016.
- 26 Adobe: Software und Services für Kreative | Adobe Creative Cloud. Online verfügbar unter <http://www.adobe.com/de/creativecloud.html>, zuletzt geprüft am 19.06.2016.
- 27 InVision: Free Web & Mobile Prototyping (Web, iOS, Android) and UI Mockup Tool. Online verfügbar unter <https://www.invisionapp.com/>, zuletzt geprüft am 19.06.2016.



- 28 Google: Chrome Browser. Online verfügbar unter <https://www.google.com/chrome/browser/desktop/index.html>, zuletzt geprüft am 19.06.2016.
- 29 Mozilla: Choose the independent browser. Online verfügbar unter <https://www.mozilla.org/de/firefox/new/>, zuletzt geprüft am 19.06.2016.
- 30 Browsersync - Time-saving synchronised browser testing. Online verfügbar unter <https://www.browsersync.io/>, zuletzt geprüft am 24.07.2016.
- 31 gulp.js - the streaming build system. Online verfügbar unter <http://gulpjs.com/>, zuletzt geprüft am 24.07.2016.
- 32 Grunt: The JavaScript Task Runner. Online verfügbar unter <http://gruntjs.com/>, zuletzt geprüft am 24.07.2016.
- 33 Vagrant by HashiCorp (2016). Online verfügbar unter <https://www.vagrantup.com/>, zuletzt aktualisiert am 21.06.2016, zuletzt geprüft am 24.07.2016.
- 34 Slim Framework (2016). Online verfügbar unter <http://www.slimframework.com/>, zuletzt aktualisiert am 29.06.2016, zuletzt geprüft am 24.07.2016.
- 35 PHP Template Engine | Smarty. Online verfügbar unter <http://www.smarty.net/>, zuletzt geprüft am 24.07.2016.
- 36 RequireJS (2016). Online verfügbar unter <http://requirejs.org/>, zuletzt aktualisiert am 17.03.2016, zuletzt geprüft am 24.07.2016.
- 37 TYPO3 - The Enterprise Open Source CMS (2016). Online verfügbar unter <https://typo3.org/>, zuletzt aktualisiert am 24.07.2016, zuletzt geprüft am 24.07.2016.
- 38 Fluid Powered TYPO3: Welcome. Online verfügbar unter <https://fluidtypo3.org/>, zuletzt geprüft am 24.07.2016.
- 39 XAMPP Installers and Downloads for Apache Friends. Online verfügbar unter <https://www.apachefriends.org/index.html>, zuletzt geprüft am 24.07.2016.
- 40 Help for The W3C Markup Validation Service. Online verfügbar unter [https://validator.w3.org/docs/help.html#validation\\_basics](https://validator.w3.org/docs/help.html#validation_basics), zuletzt geprüft am 24.07.2016.
- 41 Editor, Pinegrow Web (2016): Pinegrow Web Editor | Website Builder for Professionals. Online verfügbar unter <http://pinegrow.com/>, zuletzt aktualisiert am 08.07.2016, zuletzt geprüft am 24.07.2016.
- 42 Adobe: Website erstellen, HTML-Editor | Adobe Dreamweaver CC. Online verfügbar unter <http://www.adobe.com/de/products/dreamweaver.html>, zuletzt geprüft am 19.06.2016.

- 43 Jetstrap: The Twitter Bootstrap Builder Tool. Online verfügbar unter <https://jetstrap.com/>, zuletzt geprüft am 24.07.2016.
- 44 Getting Started with Macaw – Macaw. Online verfügbar unter <https://help.macaw.co/hc/en-us/sections/201740333>, zuletzt geprüft am 17.07.2016.
- 45 Macaw Videos - The code-savvy web design tool. Online verfügbar unter <http://macaw.co/videos/>, zuletzt geprüft am 17.07.2016.
- 46 Designing and Building Websites With Macaw - Envato Tuts+ Web Design Tutorials. Online verfügbar unter <http://webdesign.tutsplus.com/series/designing-and-building-websites-with-macaw--cms-556>, zuletzt geprüft am 17.07.2016.
- 47 Schonne Eldridge: Responsive Web Design with Macaw. Online verfügbar unter [http://schonne.com/macaw/Responsive\\_Web\\_Design\\_with\\_Macaw.pdf](http://schonne.com/macaw/Responsive_Web_Design_with_Macaw.pdf).
- 48 Chellman, Joe; Rainey, Rex (2015): Getting started with MACAW. 2. Aufl. San Francisco, CA: Peachpit Press.
- 49 Pinegrow Web Editor | Documentation & Tutorials. Online verfügbar unter <http://docs.pinegrow.com/>, zuletzt geprüft am 17.07.2016.
- 50 Documentation - Materialize. Online verfügbar unter <http://materializecss.com/>, zuletzt geprüft am 24.07.2016.
- 51 Pinegrow Web Designer (2016). Online verfügbar unter <http://pinegrow.com/edit.html>, zuletzt aktualisiert am 08.07.2016, zuletzt geprüft am 24.07.2016.
- 52 Help center, documentation, and tutorials | Webflow. Online verfügbar unter <https://help.webflow.com/>, zuletzt geprüft am 24.07.2016.
- 53 Can I import websites, HTML, or other projects into Webflow? | Article. Online verfügbar unter <https://help.webflow.com/article/can-i-import-websites-html-or-other-projects-into-webflow>, zuletzt geprüft am 21.07.2016.
- 54 SASS/LESS Export feature. Online verfügbar unter <http://forum.webflow.com/t/sass-less-export-feature/68/3>, zuletzt geprüft am 21.07.2016.
- 55 Webflow on Twitter. Online verfügbar unter <https://twitter.com/webflowapp/status/569566582214504448>, zuletzt geprüft am 21.07.2016.
- 56 Percentages and breakpoints. Online verfügbar unter <http://forum.webflow.com/t/percentages-and-breakpoints/710>, zuletzt geprüft am 21.07.2016.

- 57 Marketing – Create your stunning website. Online verfügbar unter <https://wordpress.com/create/>, zuletzt geprüft am 25.07.2016.
- 58 Macaw Scarlet | The Live Design Environment (2016). Online verfügbar unter <http://scarlet.macaw.co/>, zuletzt aktualisiert am 08.02.2016, zuletzt geprüft am 25.07.2016.
- 59 Macaw is joining InVision! (2016). Online verfügbar unter <http://macaw.co/invision/>, zuletzt aktualisiert am 27.01.2016, zuletzt geprüft am 25.07.2016.
- 60 UX-Design, Prototyping | Adobe Experience Design CC. Online verfügbar unter <http://www.adobe.com/de/products/experience-design.html>, zuletzt geprüft am 30.07.2016.
- 61 Atomic: Interface design software for professionals (2016). Online verfügbar unter <https://atomic.io/>, zuletzt aktualisiert am 29.07.2016, zuletzt geprüft am 30.07.2016.
- 62 Collaboration between designers and developers made easy (2016). Online verfügbar unter <https://avocode.com/>, zuletzt aktualisiert am 27.07.2016, zuletzt geprüft am 30.07.2016.
- 63 Google Web Designer (2016). Online verfügbar unter <https://www.google.com/webdesigner/>, zuletzt aktualisiert am 03.06.2016, zuletzt geprüft am 30.07.2016.
- 64 Pixate (2015). Online verfügbar unter <http://www.pixate.com/>, zuletzt aktualisiert am 06.11.2015, zuletzt geprüft am 30.07.2016.
- 65 Pablo (2016): WYSIWYG Web Builder. Online verfügbar unter <http://www.wysiwygwebbuilder.com/index.html>, zuletzt aktualisiert am 21.05.2016, zuletzt geprüft am 30.07.2016.

## ANHANG

### A TEAM-BEFRAGUNG ZU DEN ANFORDERUNGEN AN EIN WEBDESIGN-TOOL

Im Rahmen der Befragung konnten die Team-Mitglieder eine Reihe an Freitextfragen beantworten (Tabelle 3) sowie die Wichtigkeit von Kriterien auf einer Skala von 1 (unwichtig) bis 10 (sehr wichtig) beurteilen (Tabelle 5). Zusätzlich wurden die Befragten vor der Beurteilung der Wichtigkeiten zu ihrem Tätigkeitsfeld befragt (Tabelle 4), woraufhin die darauffolgenden Fragen auf das angegebene Tätigkeitsfeld angepasst wurden.

Frage	Anzahl Antworten	Antworten
Wo siehst du in der Benutzung eines speziellen Tools Vorteile und Chancen im Vergleich zum jetzigen Workflow?	6	<p>Definierte Prozesse und bekannter Funktionsumfang in gleichbleibender und garantierte Qualität</p> <p>Die Erstellung von Prototypen würde deutlich schneller gehen</p> <p>Schneller und effizienter auch in kleinen Projekten zum Ziel zu kommen. Auch kleinere Projekte bedienen zu können, ohne sofort den Budgetrahmen zu sprengen. Schnelle Produktion von funktionsfähigen Click-Dummies.</p> <p>Einerseits bei der Erstellung von Klickdummies wäre es super, wenn das schneller und einfacher ginge, aber je nachdem wie gut die Tools sind, wäre die initiale Umsetzung des Screendesigns in HTML auch eine Stelle, an der ein Tool unsere Geschwindigkeit erhöhen könnte. Dabei wäre interessant, ob man im Tool auch nur einzelne Elemente umsetzen kann. Da wir zurzeit im Sinne von BEM die Module ja sehr abgekapselt bauen, wäre das eine super Sache :) Vielleicht gibt es sogar im Tool eine Möglichkeit Styleguides daraus zu erstellen?</p> <p>Geeigneteres Tool/Workflow, effizientere Arbeitsweise, mehr "Automatisierung", kürzere Entwicklungszeiten</p> <p>Einfache Websites schneller und unkomplizierter umsetzen zu können</p>
Wo siehst du in der Benutzung eines speziellen Tools Nachteile und Risiken im Vergleich zum jetzigen Workflow?	6	<p>Eventuell weniger Spielraum für kreativere Ansätze</p> <p>Wahrscheinlich wird es am Ende darauf hinauslaufen, dass alles zweimal gebaut werden muss, da die Struktur für diverse JS-Anpassungen nicht passt, die Wiederverwertbarkeit von Modulen nicht gegeben ist, etc. Ich glaube nicht, dass man am Ende wirklich schneller ist.</p> <p>Mangelnde Qualität im Code und Verlust von Individualität in der Gestaltung auf Grund technischer Einschränkungen der Editoren.</p>

		<p>Risiko ist natürlich, dass wir nach der Nutzung des Tools viel Arbeit haben, um den Code tatsächlich nutzen zu können. Das würde den Vorteil dann wieder zunichte machen.</p> <p>Mangelhafte Code-Qualität, initialer Einarbeitungsaufwand, letztendliche Untauglichkeit könnte womöglich erst nach langer Praxis-Phase erkannt werden, höhere Einstiegshürde für neue Mitarbeiter, evtl. unzureichende Export-Formate für verschiedene Zwecke (Vorschau-Screenshots o.ä.)</p> <p>* Qualität des Screendesigns und des Codes hängt stark von den Möglichkeiten des Tools ab * Wir machen uns selber etwas obsolet, weil jeder Depp unter Umständen das Tool erlernen kann * Limitierung der Kreativität, Annäherungen der Websites aneinander, da immer nach Baukastenprinzip (noch schlimmer als bisher in Sachen Responsive / Flat Design)</p>
Weitere Anmerkungen?	0	keine

Tabelle 3: Freitextfragen in der Team-Befragung

Du arbeitest hauptsächlich als...	Anzahl Antworten
Entwickler	3
Designer	1
Entwickler + Designer	1
Kundenberater	2

Tabelle 4: Frage nach dem Tätigkeitsfeld in der Team-Befragung

Frage (Auf einer Skala von 1-10, als wie wichtig erachtest du...)	Anzahl Antworten	Durchschnittliche Bewertung
---	------------------	-----------------------------

#### Entwickler

Validität des generierten Codes nach Standards des W3C	4	8
semantisch korrekte Auszeichnung der Elemente im HTML	4	7,8
sinnvolle Klassenbezeichnungen nach BEM o.ä.	4	7
Verwendung von Variablen	4	5,3

Verwendung von LESS/SASS	4	7,8
volle Kontrolle über den generierten Code bereits während des Erstellens im WYSIWYG-Modus	4	8,3
Preview in verschiedenen Browsern	4	5,3
Preview von verschiedenen Auflösungen	4	8,5
einfaches Erstellen von media queries	4	8,5
Integration eines Build-Systems (ES6, Imagemin, ...)	4	4,8
Auswahl des Frameworks (Bootstrap, Foundation, ...)	4	7,5

### Designer

Hilfe beim Erstellen von Designs für verschiedene Auflösungen	2	6,5
Ähnliches Interface zu bisherigen Programmen (z.B. Photoshop)	2	8,5
Benutzung ohne Programmierkenntnisse	2	7,5
Hilfe zum Erstellen von Designs über mehrere Unterseiten hinweg	2	7
Hilfe zum Erstellen von oft verwendeten Content-Elementen (Navbar, Formulare, Slider...)	2	7,5
Bibliotheken zum Wiederverwenden von häufig benutzten Elementen	2	6,5

### Kundenberater

geringe Kosten	2	4,5
einfachere Erstellung von Click-Prototypen	2	10
Entlastung von Entwicklern	2	7,5
Entlastung von Screendesignern	2	4

### Alle

intuitives Benutzerinterface	7	9
gute Dokumentation	7	7,7
Tutorials	7	6,3

Video-Tutorials	7	4,7
Möglichkeit, auch bereits bestehende Projekte zu bearbeiten	7	6,1
Einsatz für möglichst viele verschiedene Projekttypen und -größen	7	7,7

Tabelle 5: Fragen nach der Wichtigkeit von Kriterien in der Team-Befragung

## B VORANALYSE UND -AUSWAHL VON WEBDESIGN-TOOLS

Im Vorlauf der vorliegenden Arbeit wurden auf dem Markt verfügbare Webdesign-Tools anhand der von den Herausgebern der Tools zur Verfügung gestellten Informationen auf ihre mögliche Eignung zur Verwendung in der Agentur überprüft. Grundkriterien für die Eignung waren dabei die grundsätzliche Eignung als Webdesign-Tool und Beschaffenheit als WYSIWYG-Editor (Zeichnen von Elementen auf eine Arbeitsfläche und anschließende Generierung von Quellcode), die mögliche Verwendung auf Windows- und Mac-Computern sowie rudimentäres Vorhandensein von Funktionalität in den Bereichen Responsive Design und Seitenverwaltung.

Die folgende Tabelle bietet eine Übersicht über die untersuchten Tools sowie ihre Aufnahme- bzw. Ablehnungskriterien.

Webdesign-Tool	Eignung möglich	Aufnahme-/Ablehnungskriterium
Adobe Dreamweaver[42]	nein	Tests der Entwickler in der Vergangenheit, keine erkennbare Verbesserung
Adobe XD[60]	nein	kein WYSIWYG-Verhalten, Anwendung nur für Mac
Atomic[61]	nein	kein WYSIWYG-Verhalten
Avocode[62]	nein	kein WYSIWYG-Verhalten, Code kann nur in Teilen aus vorhandenen Designs generiert werden
Google Web Designer[63]	nein	Webdesign theoretisch möglich, jedoch zur Erstellung von Werbeanzeigen gedacht
Macaw[14]	ja	WYSIWYG-Verhalten, Windows- und Mac-Anwendung, Responsive Design, Seitenverwaltung
Pinegrow[51]	ja	WYSIWYG-Verhalten, Windows- und Mac-Anwendung, Responsive Design, Seitenverwaltung
Pixate[64]	nein	kein WYSIWYG-Verhalten

Webflow[16]	ja	WYSIWYG-Verhalten, Web-App, Responsive Design, Seitenverwaltung
WYSIWYG Web Builder[65]	nein	Anwendung nur für Windows

Tabelle 6: Voruntersuchung von Webdesign-Tools

## C ABKÜRZUNGSVERZEICHNIS

BEM	Block Element Modifier
bzw.	beziehungsweise
ca.	circa
CMS	Content-Management-System
CSS	Cascading Style Sheets
DOM	Document Object Model
ES6	ECMAScript 6
FAQ	Frequently Asked Questions
ggf.	gegebenenfalls
GmbH	Gesellschaft mit beschränkter Haftung
HTML	HyperText Markup Language
i.d.R.	in der Regel
Lotto BW	Lotto Baden-Württemberg
PHP	PHP: Hypertext Preprocessor
px	Pixel
SASS	Syntactically Awesome Style Sheets
SSL	Secure Sockets Layer
u.a.	unter anderem
W3C	World Wide Web Consortium
WYSIWYG	What You See Is What You Get
XAMPP	XAMPP Apache + MariaDB + PHP + Perl
z.B.	zum Beispiel



## D GLOSSAR

Die im folgenden Glossar zu findenden Erklärungen beziehen sich auf die Bedeutung der Begriffe und Ausdrücke im Kontext dieser Arbeit, des Webdesigns und der Webentwicklung. Andere Bedeutungen, die außerhalb dieses Kontextes relevant sind, werden nicht erläutert.

Akkordeon	Aus mehreren Überschriften und zugehörigen Texten bestehendes Element, welches die platzsparende Unterbringung von Inhalten ermöglicht, indem die entsprechenden Texte erst bei Klick auf die Überschrift eingeblendet werden
BEM-Methodik	Bei der Erstellung von CSS-Code eingesetzte Methodik, die es u.a. ermöglicht, Elemente gestalterisch vom Rest der Webseite zu lösen und damit Interferenzen mit anderen Elemente zu vermeiden
Click-Prototyp	Klickbarer Prototyp einer Webseite, der i.d.R. auf Screenshots basiert, die über Link-Flächen miteinander verknüpft werden, um z.B. Navigation zwischen Seiten zu ermöglichen
Content-Management-System	System, in dem Inhalte einer Webseite von meist nicht technisch versierten Benutzern gepflegt werden können
Framework	Funktions- und Komponentenbaukasten, der als Rahmen zur schnelleren Entwicklung von Webanwendungen eingesetzt wird
Keyvisual	Element mit i.d.R. 100% Breite, welches über den oberen Bereich einer Webseite gespannt ist, um den Inhalt der Seite visuell zu unterstützen
Mobile-First-Ansatz	Ansatz, nachdem responsive Webseiten entwickelt werden. Dabei wird die Seite zuerst für den kleinsten verfügbaren Breakpoint gestaltet, um dann das Aussehen der Seite nach und nach für größere Auflösungen anzupassen
Navbar	Auf vielen Webseiten eingesetzte Kombination aus meist Logo, Navigation und weiteren Elementen wie Suchfeld oder Sprachauswahl-Feld, die am oberen Rand der Seite angedockt wird und auch beim Herunterscrollen meist am oberen Rand des Viewports angeheftet bleibt

Off-Canvas-Navigation	Navigations-Pattern, das meist für Mobilgeräte verwendet wird. Dabei wird die Navigation aus dem Viewport geschoben und z.B. durch Klick eines Buttons von der linken oder rechten Seite zurück in den Viewport gefahren, um dem Benutzer die Navigation zu ermöglichen
One-Pager	Webseite, die hauptsächlich aus einer Seite besteht, die i.d.R. in verschiedene Abschnitte aufgeteilt ist
Responsive Design	Auch Responsive Webdesign; Optimierung von Designs für die Anzeige auf Geräten unterschiedlicher Bildschirmgrößen
Syntax-Highlighting	Einfärben von Code, um die Leserlichkeit zu verbessern und Funktionalität der Code-Teile zu unterstreichen

## **E ABILDUNGSVERZEICHNIS**

<i>Abbildung 1: Screendesign Lotto BW "Kabinenaktion" .....</i>	<i>14</i>
<i>Abbildung 2: Screendesign Heimatland Startseite .....</i>	<i>15</i>
<i>Abbildung 3: Screendesign Fichtner .....</i>	<i>16</i>
<i>Abbildung 4: Generische HTML-Elemente und Klassennamen in Macaw .....</i>	<i>27</i>
<i>Abbildung 5: Nummerierung von Elementen und Positionierungsschwierigkeiten in Macaw.....</i>	<i>28</i>
<i>Abbildung 6: Framework-Auswahl in Pinegrow .....</i>	<i>31</i>
<i>Abbildung 7: Media Query Wizard in Pinegrow .....</i>	<i>32</i>
<i>Abbildung 8: Element hinzufügen Dialog mit In-App-Hilfe in Webflow .....</i>	<i>36</i>
<i>Abbildung 9: Einstellungs-Dialog, normaler Modus in Webflow.....</i>	<i>37</i>
<i>Abbildung 10: Einstellungs-Dialog, Advanced-Modus in Webflow .....</i>	<i>37</i>

## **F TABELLENVERZEICHNIS**

<i>Tabelle 1: Anforderungskatalog.....</i>	<i>22</i>
<i>Tabelle 2: Zusammenfassung der Tool-Analyse .....</i>	<i>41</i>
<i>Tabelle 3: Freitextfragen in der Team-Befragung .....</i>	<i>52</i>
<i>Tabelle 4: Frage nach dem Tätigkeitsfeld in der Team-Befragung .....</i>	<i>52</i>
<i>Tabelle 5: Fragen nach der Wichtigkeit von Kriterien in der Team-Befragung .....</i>	<i>54</i>