

an der HTWK Leipzig
Hochschule für Technik, Wirtschaft und Kultur (FH)

© Lars Häuser | lh84.de

Responsive und Adaptive Webdesign

Ein Überblick über aktuelle Ansätze für mobiles Webdesign

Erstellt von:	Lars Häuser (Matrikelnummer: 60395)
Eingereicht am:	29.11.2013
Prüfer:	Prof. Dr.-Ing. Jörg Bleymehl

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	II
Abkürzungsverzeichnis	III
1 Einleitung	1
2 Strategien für mobiles Webdesign	2
2.1 One Web	3
2.2 Desktop first	3
2.3 Mobile First	5
2.4 Separate mobile Websites	7
2.5 Progressive Enhancement / Graceful Degradation	7
3 Technische Ansätze	9
3.1 Herkunft	9
3.2 Adaptive Web Design (AWD)	10
3.3 Responsive Web Design (RWD)	11
3.4 Responsive versus Adaptive Webdesign	12
3.5 Allgemeine Vor- und Nachteile dieser Techniken	13
3.6 Beispiele für Responsive Webdesign	15
4 Grundlegende Techniken für Responsive Webdesign	17
4.1 CSS Media Queries	17
4.2 Flexible Grid	19
4.2.1 Responsive in 3 Schritten	20
4.2.2 Frameworks	22
4.3 Flexible Bilder und andere Medien	23
5 Fazit und Ausblick	26
Literaturverzeichnis	V
Selbstständigkeitserklärung	VIII

Abbildungsverzeichnis

Abb. 2-1: Schem. Darstellung des »Desktop First« Prinzips	4
Abb. 2-2: Schem. Darstellung des Mobile First Prinzips.....	6
Abb. 3-1: Bsp. RWD Volkswagen Showroom – Desktop ähnliche Auflösung	15
Abb. 3-2: Bsp. RWD Volkswagen Showroom – Tablet ähnliche Auflösung	16
Abb. 3-3: Bsp. RWD Volkswagen Showroom – Smartphone ähnliche Auflösung	16
Abb. 4-1 HTML Seitenaufbau für Responsive Webdesign Beispiel	20

Abkürzungsverzeichnis

API	Application Programming Interface (zu deutsch: Programmierschnittstelle)
AWD	Adaptive Web Design
CSS	Cascading Style Sheets
HCI	Human Computer Interface (zu deutsch: Mensch-Maschine-Schnittstelle)
HTML	Hypertext Markup Language
HTWK	Hochschule für Technik, Wirtschaft und Kultur Leipzig
IE	Internet Explorer von Microsoft
JS	Javascript
MWI	Mobile Web Initiative
RWD	Responsive Web Design
SEO	Search engine optimization (zu deutsch: Suchmaschinenoptimierung)
TV	Television (zu deutsch: Fernsehen)
Typo	Typographie
UX	User Experience (zu deutsch: Nutzererlebnis)
W3C	World Wide Web Consortium
XML	Extensible Markup Language

1 Einleitung

Von Tag zu Tag steigt die Anzahl der mobilen Nutzer des Internets. Fast täglich kommen neue Geräte, wie Smartphone und Tablets auf den Markt, mit denen Nutzer mobil im World Wide Web surfen können. Immer unterschiedlicher sind die Displaygrößen. Immer günstiger werden die Tarife und sind somit für mehr Menschen nutzbar.

All diese Faktoren ändern auch das Verhalten der Anwender. Hat man früher noch einmal am Tag im Internet die E-Mails an einem stationären PC abgerufen, sind die Nutzer heutzutage stündlich und meist mobil im Internet unterwegs. Eine Studie von Mary Meeker der Morgan Stanley Investmentbank aus New York hat belegt, dass 2014 die Anzahl der mobilen User des Internets erstmals die Anzahl der Desktop User übertreffen wird. (Ingram 2010)

Der immer stärker wachsende Markt der Smartphones, Tablets und anderer kleiner internetfähiger Geräte bedeutet auch, dass ein Umdenken bei dem Aufbau von Webseiten beachtet werden muss. Benutzer von Webangeboten aufgrund mobiler Verwendung vom Webauftritt auszuschließen, ist und kann keine Alternative sein, weswegen sich in den letzten Jahren unterschiedliche Strategien entwickelt haben, wie man Internetseiten für die mobile Nutzung optimieren kann.

Aufgrund der stetig wachsenden Anzahl von Geräten mit unterschiedlichen Displaygrößen, wird es für Ersteller von Internetseiten schwieriger Webseiten auf allen Geräten optimal darzustellen. Ein moderner Ansatz, Responsive Webdesign, zur Website-Erstellung wurde 2010 vom amerikanischen Webdesigner Ethan Marcotte vorgestellt und ist zur Zeit eines der meist diskutierten Themen zwischen Webentwicklern. (Boudreaux 2013)

Die vorliegende Hausarbeit soll einen Überblick über diese aktuelle Technik und Strategie zur Erstellung von mobilen Internetseiten geben. Ein besonderer Teil der Arbeit stellt die Zusammenfassung der Responsive Webdesign Technik mit allen konzeptionellen Ansätzen und technischen Lösungen dar.

Um den Rahmen dieser Arbeit nicht zu sprengen, kann nur eine Auswahl an technischen Lösungen vorgestellt werden. Es kann keine Garantie auf Vollständigkeit gegeben werden. Grundlegende Kenntnisse in HTML, CSS und Javascript sollten zum Verständnis der vorliegenden Arbeit vorhanden sein.

2 Strategien für mobiles Webdesign

Die Welt besteht bekanntermaßen nicht nur aus Schwarz und Weiß, was sich auch auf das Webdesign bzw. das Internet allgemein sehr gut übertragen lässt. Die Entwicklung des Internets und der Webentwicklung bzw. des Webdesigns in den letzten Jahren hat gezeigt, dass diese rasend schnell voran schreiten kann. Nicht nur die Techniken, die zur Umsetzung genutzt werden, sondern auch die Gedanken und Ansätze, die sich vorher entwickelt haben, werden stets weiterentwickelt.

So ist es auch mit der Entwicklung von mobilen Internetangeboten. Mit der Einführung des iPhones vor zirka sechs Jahren hat sich das Verhalten der Nutzer im Umgang mit Internetangeboten stark geändert und ändert sich stetig weiter. Man kann die mobile Nutzung des Internets jedoch nicht nur auf das iPhone beschränken, wie es teilweise von Unternehmen getan wird. (Maurice 2012, S. XIII) Sondern es benötigt Strategien, um möglichst viele Menschen mit den unterschiedlichsten Geräten im mobilen Bereich zu erreichen.

Für die Gestaltung von mobilen Webangeboten existieren unterschiedliche Strategien und Ansätze. Von kleineren Anpassungen der Desktop-Version einer Internetseite, über Responsive Webdesign bis hin zur nativen App für ein Smartphone ist alles möglich.

Im April 2012 erläuterte der Usability-Experte Jakob Nielsen in dem Blogbeitrag „Mobile Site vs. Full Site“, dass es immer zwei von einander getrennte Versionen einer Website geben sollte: zum einen die vollfunktionsfähige Desktop-Version und zum anderen die mobile Variante, die mit reduzierten Inhalten und gekappten Features auskommen muss. Er stellt diese Herangehensweise als einzig Richtige dar, wie folgendes Zitat zeigt:

“ *Summary: Good mobile user experience requires a different design than what's needed to satisfy desktop users. Two designs, two sites, and cross-linking to make it all work.*

JAKOB NIELSEN: 10. April 2012 (NIELSEN 2012)

Daraufhin wurde ein Sturm der Entrüstung losgetreten in der Web-Community, auch unter Vertretern des Responsive Webdesign. (Maurice 2012, S. XIII)

Die unterschiedlichen Ansätze und Vorgehensweise bei der Entwicklung von mobilen Internetangeboten soll daher in den folgenden Kapiteln näher erläutert werden.

2.1 One Web

Der Gedanke des »One Web« ist schon etwas älter und entsprang den Gedanken der W3C¹Arbeitsgruppe MWI². Diese Arbeitsgruppe setzt sich seit 2005 mit Strategien, Techniken und anderen „Best Practices“ für das mobile Internet auseinander. Der Leitspruch der Arbeitsgruppe „Combining the power of the Web with the strengths of mobiles devices.“ (W3C 2013) belegt dies.

Der erste Ansatz bzw. die erste Definition des Begriffs »One Web« besagt, dass es wünschenswert ist, dem Nutzer sämtliche Inhalte und Dienste des Webs unabhängig vom Gerät bereitzustellen (Rabin/McCathieNevile 2005). Dieser Gedanke ist keine konkrete Technik oder Strategie, als vielmehr eine allgemeine Empfehlung oder Philosophie wie mit Webseiten umzugehen ist, um alle Nutzer zu erreichen.

Aufgrund verschiedener Geräte und Nutzerbedürfnisse erwies sich diese Vorstellung als eher unpraktisch, weshalb sie bis 2008 mehrmals überarbeitet wurde. Die aktuellste Fassung der Definition lautet:

“ *One Web means making, as far as is reasonable, the same information and services available to users irrespective of the device they are using. However, it does not mean that exactly the same information is available in exactly the same representation across all devices.*

(Rabin/McCathieNevile 2005)

Mit dieser Änderung des Leitsatzes, bleibt der Grundgedanke des einheitlichen Webs zwar erhalten, aber es wird besser spezifiziert, wie sich die Inhalte auf unterschiedlichen Geräten präsentieren sollten.

2.2 Desktop first

Wohingegen der »One Web« Gedanke eher einem groben Leitfaden entspricht, ist der Ansatz »Desktop First« konkreter und beschreibt ein Vorgehen, wie bei der Entwicklung neuer Websites agiert werden sollte.

Wie der Name schon sagt, beginnt man bei dieser Technik mit der Planung bzw. Gestaltung der Desktop-Variante einer Internetseite. Zu Beginn wird von

¹ Abkürzung für World Wide Web Consortium, <http://www.w3.org/>

² Abkürzung für Mobile Web Initiative, <http://www.w3.org/Mobile/>

viel Platz (auf die Bildschirmgröße bzw. -Auflösung bezogen) und einer hohen Performance der Geräte ausgegangen, auf dem die Seite angezeigt werden soll. Die Layouts und Designs sind daher oft sehr aufwendig gestaltet und die Features einer Website verwenden oft die neuesten Techniken, die moderne Browser bieten.

Nachdem das Design und die Inhalte der Desktop-Version gestaltet sind, werden diese dann nach und nach „abgespeckt“ um so die mobile Version der Seite zu gestalten. Dieses Vorgehen wird in manchen Zusammenhängen auch als »Graceful Degradation« bezeichnet siehe Kap. 2.5. (Hellwig 2013)

Die folgende Grafik zeigt eine schematische Darstellung dieses Prinzips.

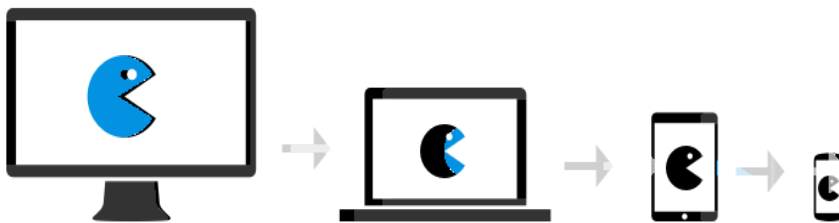


Abb. 2-1: Schem. Darstellung des »Desktop First« Prinzips³

Dieser Ansatz wurde bis vor ein paar Jahren noch von jedem Entwickler angewendet, weil es der Gewohnheit entspricht, in dieser Reihenfolge zu arbeiten. In Zeiten, in denen Geräte noch nicht so leistungsfähig waren und die Mehrheit der Nutzer mit dem stationären PC online gegangen sind, war dieser Ansatz durchaus praktikabel und sinnvoll. Doch mit den immer leistungsfähigeren Mobilgeräten haben sich auch durchaus Schwächen herausgestellt. Im Folgenden werden Vor- und Nachteile dieser Technik aufgezeigt: (Hellwig 2013)

Vorteile:

- Entspricht unserer Gewohnheit
- Moderne Systeme werden voll ausgereizt. Neuste Technologien können eingesetzt werden.

Nachteile:

- Schwache Systeme werden oft nicht oder nur schlecht berücksichtigt, weshalb Performance-Probleme entstehen und die Systeme überlastet werden können.

³ Eigene Darstellung in Anlehnung an (Frost, 2012)

- Informationen sind nicht für alle Menschen gleich zugänglich, wenn schwache Systeme und/oder kleine Displays bestraft werden. Das widerspricht der Grundidee des Internets.
- Schwierige Erweiterung der Website. Änderungen müssen in allen schwächeren Systemen getestet und ggf. nachgerüstet werden.

2.3 Mobile First

Ein weiterer Ansatz stellt die »Mobile First« Idee dar. Wie auch hier der Name vermuten lässt, beginnt man zuerst mit der Konzeption für mobile Geräte. Der amerikanische Webdesigner Luke Wroblewski hat diese Herangehensweise im November 2009 das erste Mal vorgestellt und somit den Begriff geprägt. In seinem Internetartikel »Mobile First«⁴ nennt Wroblewski drei Gründe, warum es Sinn macht bei der Erstellung von neuen Projekten mit der Planung der mobilen Version einer Internetseite zu beginnen (Zillgens 2013, S. 62):

1. Growth - The growth of mobile is a huge opportunity to reach more people than ever.

Sein erster Grund zielt auf das sehr große Wachstum des mobilen Marktes ab. Immer mehr internetfähige mobile Geräte kommen auf den Markt und sind nicht mehr wegzudenken, weswegen man diese nicht ignorieren kann. Sein Ansatz sieht diesen Markt als Chance, wenn man von Beginn an, diese schnell wachsende Nutzerschaft gewinnt, ist man für die Zukunft vorbereitet. (Wroblewski 2009)

2. Constraints - The constraints of the mobile medium force us to focus on what really matters.

In seinem zweiten Punkt sagt er, dass man durch den wenigen Platz, den man auf mobilen Geräten zur Verfügung hat, dazu gezwungen ist, sich auf das Wesentliche i.S. von Inhalten zu konzentrieren. „Es ist einfach kein Platz, um fragliche Inhalte [...] unterzubringen.“ (Zillgens 2013, S. 63)

3. Capabilities - The capabilities of mobile create opportunities to innovate.

Der dritte und abschließende Punkt seines Artikels besagt, dass man mit den Plattformen von Google und Android auch neue Möglichkeiten für innovative Applikationen nutzen kann. Er nennt z.B. Funktionen wie Kompass, Ortsbestimmungen oder Bewegungssensoren, die im Desktop-Bereich nicht so einfach zu verwenden oder gar nicht erst sinnvoll sind. (Wroblewski 2009)

⁴ Mobile First (3.11.2009) - <http://www.lukew.com/ff/entry.asp?933>

Zu diesen Punkten hat Wroblewski im Jahr 2011 zusätzlich einen Artikel veröffentlicht, der mit dem folgenden Zitat zusammengefasst werden kann, welches auch seinen Artikel von 2009 bestätigt:

“ „*WHY MOBILE FIRST? Here's the elevator pitch: designing for mobile first not only prepares you for the explosive growth and new opportunities on the mobile internet, it forces you to focus and enables you to innovate in ways you previously couldn't. Of course there's a lot of detail behind that statement, which is what this part is all about.*“ (Wroblewski 2011)

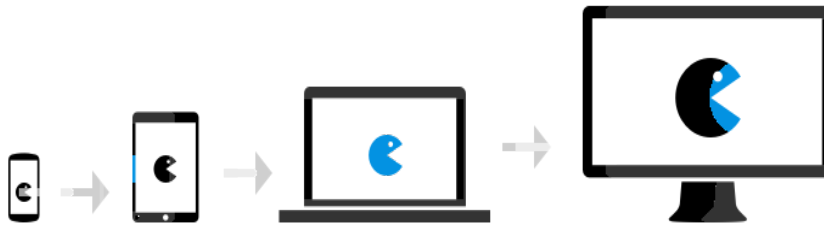


Abb. 2-2: Schem. Darstellung des Mobile First Prinzips⁵

Im Folgenden werden wieder die Vor- und Nachteile dieser Herangehensweise aufgezeigt (Hellwig 2013):

Vorteile:

- Sehr prägnante Inhalte, da sich auf diese bei der Erstellung konzentriert wird.
- Optimale Darstellung auf allen Geräten
- Bestmögliche Performance auf allen Geräten, da der Quellcode “umgedreht” wird. Es wird demnach erst für die mobile Version der Quellcode erstellt und dann stets erweitert.
- Unkomplizierte Erweiterbarkeit auf größere Displays oder modernere Systeme

⁵ Eigene Darstellung in Anlehnung an (Frost, 2012)

Nachteile:

- Aufwändige Konzeption und/oder technische Umsetzung
- Starkes Umdenken notwendig (aus heutiger Sicht)

2.4 Separate mobile Websites

Ein weiterer Ansatz ist es, für die mobile Version einer Internetseite, eine separate Version zu erstellen, die sich technisch wie inhaltlich von der Desktop-Version unterscheidet und auf die im Falle eines mobiles Zugriffs umgeleitet wird.

Oft wird hierbei auch nur eine Version für mobile Geräte erstellt. Es muss schon vorher überlegt werden, für welche Gerätegröße die Seite optimiert werden soll. Der Nachteil dieser Variante ist, dass man entgegen moderner Ansätze wie Responsive Webdesign, wieder nur eine kleinere Gruppe von Nutzern optimal erreicht.

Vorteile hingegen bestehen darin, dass die Inhalte und Strukturen optimal für diese eine Version erstellt worden sind. Bilder, HTML-Code und andere Elemente können gezielt für diese eine Gerätegröße verbessert werden. Das sich dann auf die Performance der Website positiv auswirkt.

2.5 Progressive Enhancement / Graceful Degradation

Die beiden folgenden Konzepte sind vorrangig grundlegende Herangehensweisen. Sie sind etwas älter, als die bereits vorgestellten Konzepte, weswegen auch gewisse Gemeinsamkeiten festgestellt werden können. Der Vollständigkeit halber sollen Sie hier kurz erläutert werden.

Progressive Enhancement (PE)

Erstmals erwähnt wurde diese Methodik im Jahr 2003 von Steve Champeon und Nick Finck (Champeon/Finck 2003). PE beschreibt eine Herangehensweise im Webdesign, eine Internetseite für alle Geräte nutzbar zu machen, und beginnt mit der Entwicklung für beschränkt fähige Geräte (sog. z.B. Featurephones oder eBook-Reader, die wenig bis gar kein Javascript, CSS oder Flash beherrschen). Erst nach und nach wird eine komplexere Version der Internetseite für fähigere Geräte erstellt, ohne jedoch weniger fähige Geräte auszuschließen. Die Philosophie dahinter ist, jedem Webbrowser mit jeglicher Internetverbindung den Zugang zur Website zu ermöglichen, wie es ursprünglich auch vom Erfinder des WWW, Tim Burners-Lee, gedacht war, ein allgemein zugängliches Internet zu schaffen.

Der Grundgedanke nach Gustafson ist jedoch, dass der Fokus bei dieser Methode auf dem Inhalt einer Seite liegt, und nicht erst der Browser und deren Fähigkeiten betrachtet wird. (Gustafson 2008)

Graceful Degradation

Graceful Degradation ist wesentlich älter als Progressive Enhancement und kommt ursprünglich nicht aus dem Bereich Web, sondern hat seinen Ursprung in der Systementwicklung und wird dort »Fault tolerance« genannt, welche schon in den 70er und 80er Jahren erwähnt wurde. (Fault tolerance 2013)

Wie der Name vermuten lässt, handelt es sich bei »Graceful Degradation« um das Gegenteil von »Progressive Enhancement«. Bei diesem Ansatz wird sich darauf konzentriert, eine Website für den fähigsten und aktuellsten Browser zu erstellen. Erst kurz vor Ende der Entwicklung, meist im letzten Viertel, einer Website wird in älteren Browsern und auf älteren Geräten getestet. Bei dieser Methode wird in Kauf genommen, dass Nutzer mit älteren Browsern ein schlechteres aber noch passables Nutzererlebnis (User Experience) haben. (Gustafson 2008) Die wichtigsten Funktionen sollten funktionieren, aber mehr auch nicht.

3 Technische Ansätze

Wie das vorherige Kapitel gezeigt hat, existieren einige Strategien, wie man mobile Internetseiten konzipieren kann. Die folgenden zwei technischen Ansätze sind konkrete Herangehensweisen zur Umsetzung von mobilen Websites, um eine optimale Darstellung auf allen Geräten, von Smartphones bis hin zu Desktop- oder sogar großen TV-Geräten zu erreichen. Alle Nutzer einer Internetseite sollen ein gleichbleibend qualitativ hochwertiges, stringentes Nutzererlebnis auf allen Geräten erhalten.

Im Folgenden werden die beiden Techniken »Responsive« und »Adaptive Webdesign« näher erläutert. Es wird auf Vor- und Nachteile eingegangen und am Ende dieses Kapitels wird ein gelungenes Beispiel in Form von Grafiken gezeigt.

3.1 Herkunft

Die Techniken, die Responsive und Adaptive Webdesign nutzen, existieren schon länger als angenommen. Die Voraussetzungen zur Nutzung eines solchen Ansatzes sind seit der CSS Version 2 gegeben. Im April 2001 hat das W3C erste Entwürfe in einem Working Draft zu Media Queries (Lie/Çelik 2001) spezifiziert, die Grundvoraussetzung für Responsive & Adaptive Webdesign sind. Jedoch haben die Hersteller diese Spezifikationen erst später in ihre Browser implementiert (z.B. Firefox in Version 3.5, Juni 2009⁶)

Wie es sich jedoch schon mit Ajax verhalten hat, brauchte es einen aussagekräftigen Begriff unter dem diese Techniken zusammengefasst und erläutert werden. Der Begriff Responsive Webdesign wurde erstmals im Mai 2010 von dem Webdesigner und Entwickler Ethan Marcotte in einem Artikel auf „A List Apart“⁷ erwähnt und somit auch geprägt. (Marcotte 2010)

In seinem Artikel stellt er nicht nur die CSS Technik der Media Queries vor, die es ermöglicht auf unterschiedliche Displaygrößen mit einfachen Mitteln zu reagieren, sondern er versuchte der Webgemeinde die Augen zu öffnen, wieder umzudenken bzw. sich zu erinnern, dass Webseiten nicht nur starre Gebilde sind. (Zillgens 2013, S. 4) Marcotte bezog sich hierbei auf einen neuen Ansatz in der Architektur, das starre architektonische Strukturen zu reaktionsfähigen Gebilden werden, in dem sich Gebäude den äußeren Strukturen und Gegebenheiten anpassen sollen. (Zillgens 2013, S. 7) Dieses Prinzip hat er auf das

⁶ Firefox 3.5 Release Notes, 30.6.2009 - <http://www.mozilla.org/en-US/firefox/3.5/releasesnotes/>

⁷ A List Apart Artikel „Responsive Web Design“ von Ethan Marcotte (25.5.2010) - <http://alistapart.com/article/responsive-web-design>

Web übertragen, in dem sich Websites besser an die äußeren Gegebenheiten anpassen sollten. Im Falle des Webs sind das die Browser, in denen Websites dargestellt werden.

Schon immer war es möglich Prozentangaben, statt feste Pixel für Div-Container o.Ä. zu definieren. Im Webdesign hat man sich die letzten zehn Jahre jedoch eher am Printdesign orientiert. Man hat feste Breiten für die gesamte Internetseite definiert und diese aufwendigen Designs in Layout-Programmen wie Photoshop entwickelt und diese dann 1:1 mit HTML und CSS umgesetzt.

Der Ursprung des Webs nach Tim Berners-Lee, war jedoch anders. Die ersten Internetseiten bestanden nur aus Text und Hyperlinks. Fügt man keine CSS Angaben hinzu, die zu dieser Zeit noch nicht existierten, haben die ersten Browser den Text mit dem Browserfenster skaliert. Und auch heute verhält es sich ähnlich. Öffnet man ein „nacktes“ HTML Dokument ohne CSS-Formatierungen (Zillgens 2013, S. 8), reagiert dieses auf die Browserfensterbreite. Der Ursprung des Webs war also schon immer „responsive“.

3.2 Adaptive Web Design (AWD)

Der Begriff »Adaptive Web Design« (zu deutsch: „Anpassungsfähiges Webdesign“) wurde vom amerikanischen Webdesigner Aaron Gustafson in dem gleichnamigen Buch⁸ 2011 geprägt. In seinem Buch nutzt er viele Komponenten des Progressive Enhancements und definiert mit ihnen eine Möglichkeit von Design-Methoden, die den Nutzer einer Internetseite und nicht den Browser in den Mittelpunkt stellen.

Die drei Ebenen des Progressive Enhancements bestehen aus

1. der Inhaltsschicht = Semantisches HTML Markup,
2. der Präsentationsschicht = CSS und Stilangaben und
3. der Client-Seitigen Script Schicht = Javascript- und jQuery-Verhaltensweisen.

Auf diesen drei Ebenen baut auch Gustafson im Adaptive Webdesign Ansatz auf. In seinem Buch beschreibt er auf diesen drei Ebenen viele technische Lösungen, die dazu führen sollen, dem Anwender einer Website das best mögliche Nutzererlebnis zu bieten. Er beschränkt sich nicht nur auf einzelne Punkte, sondern sieht es als komplexen Lösungsansatz für eine Multi-Screen-Strategie im Webdesign.

⁸ ADAPTIVE WEB DESIGN - Crafting Rich Experiences with Progressive Enhancement von Aaron Gustafson, 2011, <http://easy-readers.net/books/adaptive-web-design/>

In der Literatur und in vielen Internetartikeln wird die von Gustafson beschriebene Methode jedoch oft auf einen Punkt reduziert, dass eine Website auf vordefinierte unterschiedliche Displaygrößen angepasst dargestellt wird. In der Praxis werden häufig drei bis vier unterschiedliche Größen für z.B. eine Desktop- (z.B. 1200px Breite), Tablet- (768px Breite) und Smartphone-Variante (z.B. 480px Breite) definiert.

Auf die Abgrenzung des Begriffs wird in Kapitel 3.4 noch einmal näher eingegangen.

Wenn man von der Ansicht der meisten Entwickler und Designer ausgeht, ergeben sich daraus gewisse Vor- und Nachteile gegenüber anderen Methoden. Diese werde im Folgenden kurz zusammengefasst:

Vorteile:

- Es ist eine einfache Entwicklung und Gestaltung mit klassischen Mockups möglich, da fest definierte Größen für die unterschiedlichen Layouts existieren.
- Für jedes Design ist aufgrund der festgelegten Größen ein starres Raster vorhanden, was gut zur Entwicklung von Layouts genommen werden kann.
- Diese Technik ist nicht extrem komplex und kann mit einfachen Erweiterungen der CSS-Regeln umgesetzt werden.
- Inhalte müssen nicht hoch flexibel sein
- Durch die geringere Komplexität, ist eine schnellere Umsetzung möglich und somit kann Zeit gespart werden.

Nachteile:

- Es werden nur bestimmte Viewports unterstützt bzw. für diese optimiert. Wenn Anwender nicht das exakte Gerät mit der exakten Displaygröße besitzen, kann die Darstellung suboptimal oder fehlerbehaftet ausfallen.
- Zu Beginn sollte eine aufwendige Zielgruppen-Analyse stattfinden, um die relevanten Viewports der Nutzer zu ermitteln. (Hellwig 2013)

3.3 Responsive Web Design (RWD)

Im Gegensatz zum Adaptive Web Design Ansatz, setzt Responsive Web Design (zu deutsch: „reaktionsfähiges Webdesign“) auf „flüssige“ Techniken in jeder Hinsicht. Ziel dieser Technik ist es, eine Internetseite auf jedem erdenklichen Display optimal darzustellen. Das Hauptelement dieses Ansatzes ist ein flüssiges Gestaltungsraster (fluid grid), was sich mit der Größe des Browsers

verändert. Umgesetzt wird es mit CSS Media Queries, worauf in Kapitel 4 näher eingegangen wird.

Wenn ein Design auf allen Geräten optimal dargestellt werden soll, bedeutet dies einen höheren technischen und gestalterischen Aufwand bei der Erstellung. Da „flüssige“ Spalten bzw. Raster schwer in Layout-Programmen dargestellt werden können, wird oft schon in einer frühen Entwicklungsphase direkt mit der Implementierung begonnen. Die Weiterentwicklung des Designs in Layout-Programmen wird dann parallel zur Implementierung fortgeführt. (Hellwig 2013)

Auch die Inhalte müssen konzeptionell anders betrachtet werden. Um eine gute Integration der Inhalte in die Website zu erzielen, müssen diese auch flexible und anpassungsfähig bzw. ebenfalls flüssig sein. Diese Tatsache lässt das Konzept und die Umsetzung des Responsive Webdesigns technisch oft sehr aufwendig und komplex werden.

Im Folgenden werden auch hier die Vor- und Nachteile dieser Technik kurz zusammengefasst:

Vorteile:

- Jede Displaygröße wird optimal berücksichtigt, da sich die Internetseite automatisch „flüssig“ an die Browsergröße anpasst.
- Es findet keine Platzverschwendung, wie beim AWD statt. Der Platz wird immer optimal ausgenutzt.
- Idealerweise steht der Inhalt im Vordergrund.

Nachteile:

- Mockups bzw. die typische Gestaltung in der Designphase sind schwer möglich, da ein flüssiges Layout nicht in Standard-Layout-Programmen abgebildet werden kann.
- Die Anpassung und technische Umsetzung ist oft komplexer als andere Methoden, da auch Inhalte sehr flexibel sein müssen und sich den äußeren Gegebenheiten anpassen müssen. Dadurch steigt natürlich der Zeit- und somit auch der Kostenaufwand in der Gestaltungs- und Umsetzungsphase.

3.4 Responsive versus Adaptive Webdesign

Der Webdesigner Zillgens behauptet in seinem Buch „Responsive Webdesign“, dass diese Technik eine Unterart von Adaptive Webdesign ist. (Zillgens 2013, S. 14)

Darüber lässt sich sichtlich streiten, wie mehrere Blog-Artikel und Diskussionsforen belegen. Viele Blogger und Webdesigner beschränken die Methode von Adaptive Webdesign auf die Auslieferung der Internetseite auf verschiedene vordefinierte Viewports, wie schon in Kapitel 3.2 beschrieben. Fakt ist, dass beide Techniken darauf ausgelegt sind, eine Internetseite optimal auf mehreren Geräten darzustellen. Beide Techniken arbeiten auch mit modernen CSS-Techniken wie z.B. Media Queries, was zusätzlich darauf schließt, dass beide sich ähneln.

Doch der Adaptive Ansatz von Gustafson geht darüber hinaus, wie er in dem Blogbeitrag »On Adaptive vs. Responsive Web Design«⁹ erklärt:

“*“Responsive web design”, as coined by Ethan Marcotte, means “fluid grids, fluid images/media & media queries.” “Adaptive web design,” as I use it, is about creating interfaces that adapt to the user’s capabilities (in terms of both form and function). To me, “adaptive web design” is just another term for “progressive enhancement” of which responsive web design can (and often should) be an integral part, but is a more holistic approach to web design in that it also takes into account varying levels of markup, CSS, JavaScript and assistive technology support.*

(Gustafson 2011)

Er selbst sagt, dass sein Ansatz eher mit Progressive Enhancement gleichzusetzen ist, und dass Responsive Webdesign ein Teil dieses Konzepts sein kann.

3.5 Allgemeine Vor- und Nachteile dieser Techniken

Im Folgenden werden allgemeine Vor- und Nachteile moderner Webtechniken wie Responsive und Adaptive Webdesign im Gegensatz zu herkömmlichen und älteren Techniken der Internetseiten-Erstellung mit nur einem Design vorgestellt.

⁹ On Adaptive vs. Responsive Web Design, Aaron Gustafson, 16.11.2011 - <http://blog.easy-designs.net/archives/on-adaptive-vs-responsive-web-design/comments/>

Vorteile:

- **Erhöht die Reichweite bei mobilen Geräten**

Eine Website für alle Geräte und User bedeutet, dass zweifellos mehr Geräte als mit einer „starrten“ Website erreicht werden. Somit steigt auch die Anzahl der Nutzer, die man erreicht.

- **Erhöht die Sichtbarkeit in Suchmaschinen (SEO)**

Responsive Design erlaubt die Verwaltung der Website mit einem Set von Links und Inhalten. Dies erlaubt den Fokus auf die Reichweite der Links und kann für eine solide SEO-Strategie genutzt werden. Außerdem müssen nicht unterschiedliche Systeme gepflegt und verwaltet werden, und somit kann sich auf Inhalte konzentriert werden, was der SEO zu Gute kommt.

- **Erhöht die Verkaufs- und Conversion-Raten**

Verbesserte Inhalte und die Konzentration auf diese, ist oft der Erfolg von Internetseiten. Wenn die Website eine konsistente User Experience (UX) bietet, die keine Weiterleitungen oder Ähnliches enthält, hat dies eine positive Auswirkung auf die Nutzer und somit kann davon ausgegangen werden, dass dies auch die Conversion-Raten ankurbeln wird. (Onbible.com 2013)

- **Zeit- und Kostenersparnis bei der Pflege**

Eine einzige Website zu pflegen, bedeutet weniger Zeit- und Kostenaufwand diese zu betreuen und zu aktualisieren. Es müssen sich keine Gedanken mehr um unterschiedliche Geräte etc. gemacht werden. Zwar entsteht in der Anfangsphase bei der Entwicklung ein etwas höherer Aufwand, dieser gleicht sich jedoch bei der Pflege wieder aus. (Onbible.com 2013)

- **Konsolidierung von Webtracking Analysen und Berichten**

Wenn nur eine Seite betrieben wird, müssen sich auch keine Gedanken über unterschiedliche Analyse Methoden, Nutzer-Klick-Verläufe und Weiterleitungen gemacht werden. Das beliebte Analyse-Tool „Google Analytics“ wurde erst Mitte 2012 für den mobilen Gebrauch aufbereitet und optimiert. (Craver 2012) Google selbst empfiehlt in seinen Guidelines für mobile Seiten, die Anwendung von Responsive Webdesign für eine bessere Listung in Suchmaschinen. (Google Inc. 2013)

Nachteile:

- **Ladezeiten**

Nutzer bzw. die Geräte, die eine mobile Internetseite nutzen, laden eventuell unnötigen HTML und CSS-Code herunter, da in den mitgelieferten Dateien, oft auch der Code der „Desktop-Variante“ enthalten ist. Auch mit Bildern wird oft nicht optimal verfahren. (Vugts 2012) Zwar werden die Bilder skaliert und somit aus Usability-Sicht und UX gut dargestellt, aber nicht verkleinert. Dies erhöht demnach auch die Ladezeiten. Moderne Ansätze mit Javascript können bei diesen Problemen aber eventuell Abhilfe schaffen, welche im Kapitel 4 näher erläutert werden.

- **Höhere initiale Entwicklungszeit**

Die Entwicklungszeit, wenn eine neue Responsive Website erstellt werden soll, ist oft länger im Vergleich zu einer Seite, die nur für ein Gerät optimiert werden muss. Viele Entwickler und Designer sind auch der Meinung, dass es einfacher und zeitersparender ist eine Responsive Website von Grund auf neu zu entwickeln als eine bestehende Seite zu konvertieren. (Vugts 2012)

3.6 Beispiele für Responsive Webdesign

Die Internetseite <https://www.awwwards.com/>, die es sich zur Aufgabe gemacht hat, die Besten für Design, Innovation und Kreativität im Internet regelmäßig zu küren, haben seit einiger Zeit, eine Kategorie für die besten Responsive Designs. Im Folgenden wird die Internetseite Volkswagen Showroom, welche ebenfalls von awwwards gekürt wurde, mittels Screenshots dargestellt.

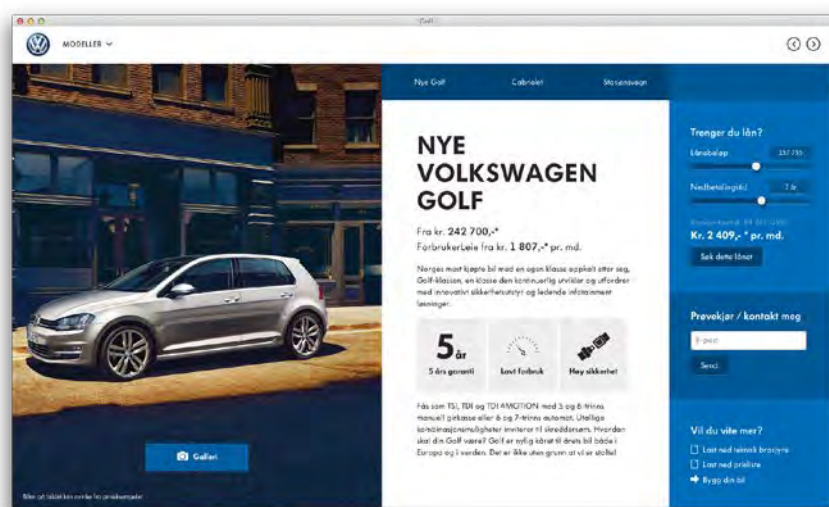


Abb. 3-1: Bsp. RWD Volkswagen Showroom – Desktop ähnliche Auflösung

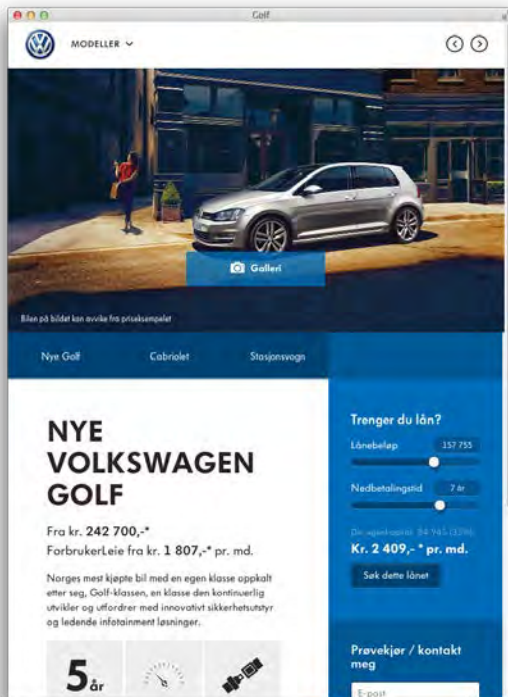


Abb. 3-2: Bsp. RWD Volkswagen Showroom – Tablet ähnliche Auflösung

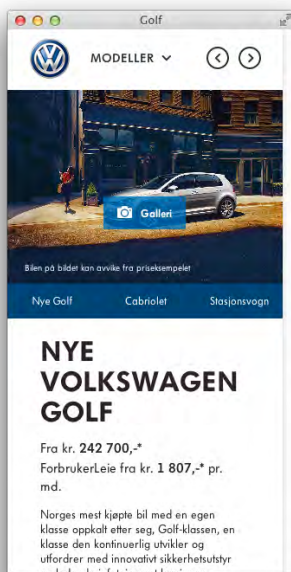


Abb. 3-3: Bsp. RWD Volkswagen Showroom – Smartphone ähnliche Auflösung

Es existieren noch mindestens hundert andere gelungene Beispiele. Jedoch ist es im Rahmen dieser Arbeit nicht möglich näher auf andere bzw. Beispiele allgemein einzugehen. Wie schon erwähnt, können jedoch auf <https://www.awwwards.com> eine Menge gute mobile responsive Websites, angesehen werden.

4 Grundlegende Techniken für Responsive Webdesign

Die Techniken, die es für einfache reaktionsfähige Websites benötigt, sind nicht sehr umfassend und schon länger in der Webgemeinde bekannt. Aus Komplexitätsgründen wurden sie jedoch lange nicht für diese Zwecke eingesetzt. Laut Ethan Marcotte, dem Erfinder des Begriffs »Responsive Webdesign«, sind nur drei zentrale Elemente für die Erstellung von reaktionsfähigen Internetseiten nötig:

1. Media Queries, ein Element aus den CSS3-Spezifikationen.
2. Ein flexibles Gestaltungsraster – Flexible Grid
3. Flexible Bilder und andere Medien

In den folgenden Kapiteln werden diese drei Hauptelemente kurz mit Code-Beispielen vorgestellt, die es ermöglichen aus einer „starr“ Seite eine reaktionsfähige zu erstellen.

4.1 CSS Media Queries

Wenn man einem flexiblen Raster keinen Maximalwert übergibt und das Browserfenster über die gesamte Breite eines aktuellen 24“ Monitors zieht, so kann dies eine Breite von 1920 Pixeln oder mehr aufweisen. Wenn sich in einem HTML-Element Text befindet und das Element über die gesamte Breite des Fensters dargestellt wird, kann dies erheblich die Lesbarkeit von Texten beeinflussen.

Hierbei hilft die in CSS3 präzierte Technik der Media Queries, die es ermöglicht, bestimmte Attribute wie Bildschirm- bzw. Viewportgröße des Browsers abzufragen und dementsprechend angepasstes CSS-Stile an den Client auszuliefern. Somit kann man in Grenzbereichen das Layout einer Seite sehr gut kontrollieren.

Wie eingangs erwähnt ist es schwierig einen Text über eine Breite von 1920px zu lesen. Genauso verhält es sich bei sehr kleinen Bildschirmgrößen wie etwa Smartphones. Man kann ein Layout nicht beliebig strecken oder stauchen. Mit Hilfe der Media Queries können für unterschiedliche Displaygrößen komplett neue Layouts geschaffen werden.

Die Anwendung von Media Queries ist einfach und erinnert an die mit CSS2 eingeführten Media-Types (W3C 2002), mit denen es möglich war, für unterschiedliche Geräte unterschiedliche CSS-Stile auszuliefern. So war es beispielsweise mit dem `media`-Attribut innerhalb eines HTML `link`-Elements im

Head einer HTML-Seite möglich, CSS-Stile für `print` (Druck) oder `screen` (Bildschirm) zu definieren.

```
<link rel="stylesheet" media="screen" href="screen.css" />
<link rel="stylesheet" media="print" href="print.css" />
```

Alternativ können diese Angaben auch direkt in einer CSS-Datei angegeben werden:

```
@media print {
    ...
}
```

An dieser Stelle knüpfen die CSS Media Queries an und schaffen eine Möglichkeit die Media-Typen näher einzugrenzen. Die zwei wichtigsten Attribute für eine Responsive Website stellen `min-width` und `max-width` dar, mit denen es möglich ist, CSS-Stile bezogen auf die Browserbreite auszuliefern.

So ist es z.B. mit der folgenden CSS-Angabe möglich, alle CSS-Regeln innerhalb der geschweiften Klammern, auf Fensterbreiten größer als 480px anzuwenden.

```
@media screen and (min-width: 480px) {
    ...
}
```

Der Vollständigkeit halber sollen hier auch die weiteren Eigenschaften des `@media` Elements kurz aufgezählt werden: `width` / `height` (Breite / Höhe des Viewports/Browserfensters), `device-width` / `-height` (Breite / Höhe des physischen Ausgabemediums – Display, Papiergröße), `orientation` portrait oder landscape (z.B. Ausrichtung des Smartphones), `aspect-ratio` (Seitenverhältnis des Viewports, color (Farbtiefe in Bit), `color-index` (Absolute Anzahl der Farben in der Farbtabelle des Ausgabemediums), `monochrome` (Farbtiefe bei monochromen Ausgabegeräten), `resolution` (Auflösung, z.B. in dpi), `scan` (progressive oder interlace für TV-Bildschirme), `grid` (für Grid-basierte Ausgabegeräte wie TTY-Terminals) (Lie et al. 2012)

Media Queries auf Smartphones

Wie man sieht, existiert ein Unterschied zwischen `width` und `device-width`. Warum? Als 2007 das iPhone auf den Markt kam, gab es für diese Gerätegröße keine optimierten Websites. Die Smartphones bzw. die Browser renderten Internetseiten auf einer imaginären Breite von 980px, obwohl die Displaygröße nur 320px breit war. Wenn man nun die CSS-Regel `@media screen and (max-width: 320px)` definieren würde, würde diese nicht zur Anwendung kommen. Aus diesem Grund wurde von Apple die HTML Meta-Eigenschaft `viewport` definiert:

```
<meta name="viewport" content="width=device-width">
```

Somit ist die Breite des Bildschirms gleich der Fläche, die zum Rendern für die Internetseite zur Verfügung steht (Zillgens 2013, S. 47) und die Seite wird entsprechend der CSS-Regeln und Media Queries dargestellt. Das eingeführte `meta`-Tag gilt als Quasi-Standard und wird von allen bekannten Browsern unterstützt. (Zillgens 2013, S. 48)

4.2 Flexible Grid

Das wohl wichtigste Element im RWD stellt das sog. Fluid oder Flexible Grid (flexibles/flüssiges Raster) dar, was auf Media Queries aufbaut. Ohne dieses zentrale Element würde diese Technik nicht existieren und ist somit Grundvoraussetzung.

Wie schon in Kapitel 3.1 kurz erwähnt, sind HTML-Seiten von Haus aus flexibel, wenn man die HTML-Tags nicht mit festen, in Pixeln angegebenen, CSS-Regeln versieht, wie es bei den meisten Internetseiten der Fall sein dürfte. In der Vergangenheit des Webdesigns hat man sich stark an dem Printdesign orientiert, da es leichter fällt für eine feste Breite eines Mediums ein Layout zu erarbeiten. In den Anfängen des Webs hat man sich stets an aktuellen durchschnittlichen Bildschirmbreiten der Nutzer orientiert. So betrug die durchschnittliche Auflösung eines Bildschirms im Jahr 2000 noch 800 x 600 Pixel. Dies hat sich natürlich mit den größer werdenden Displays stets angepasst. In den letzten Jahren haben sich viele Entwickler dafür entschieden ein 960 Pixel breites Raster für die Gestaltung zu verwenden.

Die Technik des flüssigen Rasters bezieht sich jedoch nicht auf die HTML-Seiten und deren Tags, sondern arbeitet mit prozentualen CSS-Angaben, die es ermöglichen, dass sich die HTML-Elemente einer Seite abhängig zu deren übergeordneten Elementen verhalten. Das oberste Element, an was sich ein HTML-Element orientieren kann, ist das Browserfenster. Damit die Lesbarkeit von Inhalten auf Seiten jedoch nicht leidet, wird die obere Grenze, also die maximale Breite eines Layouts oft trotzdem beschränkt.

Möchte man eine Internetseite flexibel gestalten, benötigt es letztendlich nicht viel. Dies soll an einem kleinen Praxisbeispiel im Folgenden aufgezeigt werden.

4.2.1 Responsive in 3 Schritten

Das folgende Beispiel basiert auf einem Tutorium von Web Designer Wall und kann unter der URL <http://webdesignerwall.com/tutorials/responsive-design-in-3-steps> vollständig eingesehen werden.

Voraussetzung soll eine einfache HTML-Seite mit folgendem Aufbau sein:



Abb. 4-1 HTML Seitenaufbau für Responsive Webdesign Beispiel

Schritt 1 – Meta Tag

Damit die Änderungen, die in den folgenden Schritten durchgeführt werden sollen, auch auf einem Mobilgerät sichtbar werden, muss wie schon in Kap. 4.1 erwähnt worden ist, der meta-Tag in den `<head>` der HTML Seite eingefügt werden.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Schritt 2 – HTML Struktur

Als HTML sieht das Beispiel folgendermaßen aus:

```
<!doctype html>
<html lang="de">
<head>
  <meta charset="utf-8">
  <title>Demo: Responsive Webdesign in 3 Schritten</title>
</head>
<body>
  <div id="pagewrap">
    <div id="header">
      <h1>Header</h1>
```



```
</div>
<div id="content">
  <h2>Content</h2>
</div>
<div id="sidebar">
  <h3>Sidebar</h3>
</div>
<div id="footer">
  <h4>Footer</h4>
</div>
</div>
</body>
</html>
```

Der Aufbau besteht aus einem Kopfbereich (Header), einem Inhaltscontainer (Content), einer Seitenleiste (Sidebar) und einem Fußbereich (Footer). Der header-`div` hat eine feste Höhe von 180 Pixeln, der Content-`div` eine feste Breite von 600 Pixeln und der Sidebar-`div` ist 300 Pixel breit. Die CSS-Regeln sehen dazu folgendermaßen aus:

```
#pagewrap {
  padding: 5px; width: 960px; margin: 20px auto;
}
#header {
  height: 180px;
}
#content {
  width: 600px; float: left;
}
#sidebar {
  width: 300px; float: right;
}
#footer {
  clear: both;
}
```

Schritt 3 – CSS-Anpassungen

Im dritten Schritt wird nun mit einfachen CSS-Regeln und den schon vorgestellten CSS Media Queries, das Layout so angepasst, dass es mit Vergrößerung oder Verkleinerung des Browserfensters auf dieses reagiert und das Layout somit responsive (reaktionsfähig/flüssig) wird.

```
/* für 980px oder weniger */
@media screen and (max-width: 980px) {
  #pagewrap { width: 94%; }
```

```
#content { width: 65%; }
#sidebar { width: 30%; }
}

/* für 700px oder weniger */
@media screen and (max-width: 700px) {
  #content { width: auto; float: none;}
  #sidebar { width: auto; float: none; }
}

/* für 480px oder weniger */
@media screen and (max-width: 480px) {
  #header { height: auto; }
  h1 { font-size: 24px; }
  #sidebar { display: none; }
}
```

Wie man an dem sehr einfachen Beispiel erkennen kann, sind die Grundzüge für Responsive Webdesign nicht schwer. Mit recht wenigen Mitteln, kann das gewünschte Ergebnis erreicht werden. Klar ist jedoch auch, dass mit komplexeren Websites auch die Komplexität der HTML und CSS-Struktur steigt. Das Beispiel soll lediglich die Basis der Vorgehensweise demonstrieren.

4.2.2 Frameworks

Verschiedene Tools und Frameworks können bei der Erstellung von flexiblen Grids / Layouts helfen, wenn das Rad von Grund auf nicht neu erfunden werden soll. Der Vorteil ist, dass man einen einfachen Einstiegspunkt in das Thema hat, somit Arbeit und Zeit im Vorfeld sparen kann.

Die meisten Tools bestehen aus fertigen CSS-Dateien, die man nur noch in seine HTML-Seite einbinden muss. Um dann das vordefinierte Raster anzuwenden, ist es nur nötig sein HTML-Markup an das jeweilige Tool anzupassen. Im Internet existieren unzählige Tools dieser Art. Um jedoch kurz darauf einzugehen, sollen im Folgenden beispielhaft zwei dieser Tools vorgestellt werden:

Gridpak (www.gridpak.com)

Gridpak bietet einen Online-Generator für ein flüssiges Raster an, was man zusätzlich seinen persönlichen Vorstellungen anpassen kann. Mit dem Tool ist es möglich, die Anzahl der Spalten im Raster, die Breakpoints (also bei welchen Breiten, das Layout umbrechen soll) und das Gutter zwischen den Spalten definiert werden. Hat man alles seinen Vorstellungen angepasst, lädt man das Grid, bestehend aus CSS und Javascript-Dateien herunter und bindet es in die Internetseite ein.

Responsive Grid System (www.responsivegridsystem.com)

Responsive Grid System beschreibt sich selbst wie folgt:

“ *The Responsive Grid System isn't a framework. It's not a boilerplate either. It's a quick, easy & flexible way to create a responsive web site.*

www.responsivegridsystem.com

Im Gegensatz zu Gridpak besteht dieses Tool aus fertigen CSS-Dateien und man kann zwischen einem 2 bis 12-spaltigem Layout wählen.

4.3 Flexible Bilder und andere Medien

Möchte man ein Bild in einer Website flexibel darstellen, benötigt es lediglich eine kleine CSS-Regel damit das Bild sich mit dem umliegenden Container (z.B. ein `div`-Element) verkleinert oder vergrößert.

```
img {  
  width: 100%;  
}
```

Diese kleine CSS-Anweisung kann bedeutend sein für das Layout, es verbergen sich jedoch zwei Gefahren dahinter. Zum einen wenn der umliegende Container größer ist, als die native Auflösung des Bildes, wird dieses verpixelt dargestellt. Dem kann man Abhilfe schaffen, indem man folgende CSS-Regel definiert:

```
img {  
  max-width: 100%;  
  height: auto;  
}
```

Das zweite und vielleicht größere Problem, was sich auch hinter dem Begriff »Responsive Images« verbirgt, ist, dass Bilder und andere Medien mit den o.g. CSS-Regeln nicht an mobile Geräte und deren Bandbreite angepasst werden. Sie werden immer in voller Auflösung und somit mit maximaler Dateigröße an die Geräte ausgeliefert.

Einige Webentwickler sind der Meinung, dass es eine technische Lösung benötigt, die unabhängig von HTML und CSS funktioniert, in dem abhängig von Bildschirmauflösung und Netzwerkstatus des Geräts, Medien mit bestimmter Dateigröße geladen werden. Diese Variante wäre technisch die Einfachste. Doch bis sich solch ein technischer Standard durchgesetzt hat, haben sich eventuell schon komplexere Probleme entwickelt. (Hannemann 2013)

Deswegen haben sich in der Praxis zwei Ansätze durchgesetzt, die dieses Problem schon heute lösen.

Die erste Variante ist die serverseitige Lösung adaptive-images.com, die von dem englischen Webentwickler Matt Wilcox¹⁰ entwickelt wurde. Diese Methode, basierend auf PHP, Javascript, ein paar Anweisungen in der `.htaccess`-Datei für den Apache-Server und Cookies, erkennt und erstellt automatisch richtig skalierte Bilder anhand der `img`-Tags einer HTML-Seite und der Bildschirmgröße des Nutzers. Mit dieser Technik ist es nicht nötig, den HTML Markup zu ändern, diese Aufgabe übernimmt automatisch die Bibliothek. (Willcox 2012) Somit bekommt der mobile Nutzer die richtige Bildgröße geladen und es kann Bandbreite bei der Auslieferung gespart werden.

Der zweite Ansatz basiert auf reinen Javascript-Bibliotheken. Es existieren mehrere Javascript-Skripte, die dieses Problem lösen möchten, jedoch bauen einige auf jQuery auf, was den Nachteil hat, dass unnötiges Javascript geladen werden muss. Eine interessante Javascript-Methode, die ohne jQuery auskommt und komprimiert nur 0,5 Kilobyte groß ist, nennt sich `picture-polyfill` (<https://github.com/scottjehl/picturefill>). Der Ansatz, wie man Bilder flexibel an Nutzer mit unterschiedlichen Geräten ausliefern kann, orientiert sich an den CSS Media Queries und nutzt einfache `div`-Container mit `data`-Attributen, wie folgendes Beispiel zeigt:

```
<div data-picture data-alt="Beispiel Text">
  <div data-src=" small.jpg"></div>
  <div data-src=" medium.jpg" data-media="(min-width: 400px)"></div>
  <div data-src=" large.jpg" data-media="(min-width: 800px)"></div>
  <div data-src=" extralarge.jpg" data-media="(min-width: 1000px)"></div>
  <noscript>
    <!-- Fallback ohne JavaScript. -->
    
  </noscript>
</div>
```

Wie man erkennen kann, werden die unterschiedlichen Größen von Bildern mittels unterschiedlicher `div`-Container erstellt. Im `data-src`-Attribut gibt man den Pfad zum Bild an, wie man es von dem `src`-Element im `img`-Tag gewohnt ist und im `data-media`-Attribute wird angegeben, für welche Displaygröße das einzelne Bild genutzt werden soll. Den Rest, also die Auslieferung, übernimmt dann die JS-Bibliothek. Dieser Ansatz erzeugt eine Menge HTML-Code, orientiert sich jedoch an dem Ansatz, den auch das W3C mit dem `srcset`-Attribut für

¹⁰ Persönliche Internetseite von Matt Wilcox - <http://mattwilcox.net/>

[img](#)-Tags vorschlägt (Hickson 2013), wie man es schon vom [video](#)-Tag aus HTML5 kennt.

5 Fazit und Ausblick

Das wohl größte Buzzword zur Zeit im Internet unter Webentwicklern heißt »Responsive Webdesign«, wie es in dieser Arbeit vorgestellt wurde. Diese relativ neue Herangehensweise bei der Entwicklung von Internetseiten bietet eine Möglichkeit, mit einfachen CSS-, HTML- und Javascript-Techniken, Websites für alle möglichen Geräte- bzw. Displaygrößen optimaler als bisher zu erstellen. Bei der wachsenden Anzahl an mobilen Nutzern und unterschiedlichen Geräten, mit denen man das Internet nutzen kann, ist dies ein guter Ansatz, um möglichst viele Menschen mit einer Internetseite optimal zu erreichen.

Die Komplexität dieser Technik kann für einfache Seiten recht niedrig sein, sie steigt jedoch mit anspruchsvolleren Webangeboten. Die in Kapitel 2 vorgestellten Strategien können bei der Konzeption und bei der späteren Umsetzung mit den in Kapitel 3 vorgestellten Techniken helfen.

Es lässt sich nicht sagen, ob die vorgestellten Techniken des Responsive und Adaptive Webdesigns, dass Allheilmittel sind. Jedoch wird das Problem, eine Internetseite auf unterschiedlichen Bildschirmgrößen optimal darzustellen, mit diesen Techniken mit relativ wenig Mehraufwand im Gegensatz zum „normalen“ Webdesign gut gelöst.

Das mobile Internet entwickelt sich seit ein paar Jahren rasant fort und keiner weiß genau, wo die Reise hinführen wird. Das Verhalten der Internetnutzer hat sich seit Einführung des iPhones vor zirka sechs Jahren drastisch geändert. Erstmals war es möglich, mit einem Mobiltelefon, „normale“ Websites auf solch einem kleinen Display angenehm zu betrachten. Doch schnell wuchsen die Wünsche, das Internetangebot optimaler auf kleineren Displays darzustellen, woraus letztendlich die oben vorgestellten Techniken entstanden.

Auch die Entwicklung des mobilen Internets und damit der Techniken für das Webdesign ist noch lange nicht beendet und es muss beobachtet werden, wohin sie führt. Das Verhalten könnte sich mit zukünftigen Technologien und weiteren Geräten zur Nutzung des WWW weiter stark verändern und somit müssen sich auch die Webentwickler immer wieder auf diese neuen Situationen einstellen.

Literaturverzeichnis

BOUDREAUX, Ryan (2013): „What is the difference between responsive vs. adaptive web design?“.
⌊<http://www.techrepublic.com/blog/web-designer/what-is-the-difference-between-responsive-vs-adaptive-web-design/>⌋

[Stand: 11. April 2013. Zugriff: 05. November 2013]

CHAMPEON, Steven / FINCK, Nick (2003): „Inclusive Web Design for the Future“.
⌊<http://www.hesketh.com/thought-leadership/our-publications/inclusive-web-design-future>⌋

[Stand: 11. März 2003. Zugriff: 22. November 2013]

CRAVER, Thom (2012): „Google Analytics Finally Gets Mobile App“.
⌊<http://searchenginewatch.com/article/2188314/Google-Analytics-Finally-Gets-Mobile-App>⌋

[Stand: 29. Juni 2012. Zugriff: 12. November 2013]

FROST, Brad (2012): „The Many Faces of 'Mobile First'“.
⌊<http://bradfrostweb.com/blog/mobile/the-many-faces-of-mobile-first/>⌋

[Stand: 13. August 2012. Zugriff: 06. November 2013]

GOOGLE INC. (2013): „Building Smartphone-Optimized Websites“.
⌊<https://developers.google.com/webmasters/smartphone-sites/details#separateurls>⌋

[Stand: 12. April 2013. Zugriff: 2. November 2013]

GUSTAFSON, Aaron (2008): „Understanding Progressive Enhancement“.
⌊<http://alistapart.com/article/understandingprogressiveenhancement>⌋

[Stand: 07. Oktober 2008. Zugriff: 20. November 2013]

GUSTAFSON, Aaron (2011): ADAPTIVE WEB DESIGN Crafting Rich Experiences with Progressive Enhancement. Chattanooga : easy-readers.net

GUSTAFSON, Aaron (2011): „On Adaptive vs. Responsive Web Design“.
⌊<http://blog.easy-designs.net/archives/on-adaptive-vs-responsive-web-design/>⌋

[Stand: 16. November 2011. Zugriff: 30. Oktober 2013]

HANNEMANN, Anselm (2013): „Responsive Images: So setzt du flexible Bilder schon jetzt ein“.
⌊<http://t3n.de/magazin/praktische-losungen-flexible-bilder-responsive-images-232734>⌋

[Stand: 30. September 2013. Zugriff: 20. November 2013]

HELLWIG, Jonas (2013): „Adaptive Website vs. Responsive Website“.
⌊<http://blog.kulturbanause.de/2012/11/adaptive-website-vs-responsive-website/>⌋

[Stand: 17. September 2013. Zugriff: 30. Oktober 2013]

HELLWIG, Jonas (2013): „Mobile First + Progressive Enhancement“.
⌊<http://blog.kulturbanause.de/2013/08/mobile-first-progressive-enhancement/>⌋

[Stand: 30. September 2013. Zugriff: 01. November 2013]

HICKSON, Ian (2013): „The srcset attribute - An HTML extension for adaptive images“.
⌊<http://www.w3.org/html/wg/drafts/srcset/w3c-srcset/>⌋

[Stand: 25. November 2013. Zugriff: 25. November 2013]

INGRAM, Mathew (2010): „Mary Meeker: Mobile Internet Will Soon Overtake Fixed Internet“.
⟨<http://gigaom.com/2010/04/12/mary-meeker-mobile-internet-will-soon-overtake-fixed-internet/>⟩

[Stand: 12. April 2010. Zugriff: 28. Oktober 2013]

LIE, Håkon Wium / ÇELİK, Tantek (2001): „Media queries“.
⟨<http://www.w3.org/TR/2001/WD-css3-mediaqueries-20010404/>⟩

[Stand: 04. April 2001. Zugriff: 01. November 2013]

LIE, Håkon Wium et al. (2012): „Media Queries“.
⟨<http://www.w3.org/TR/css3-mediaqueries/>⟩

[Stand: 19. Juni 2012. Zugriff: 30. Oktober 2013]

MARCOTTE, Ethan (2010): „Responsive Web Design“.
⟨<http://alistapart.com/article/responsive-web-design>⟩

[Stand: 25. Mai 2010. Zugriff: 04. November 2013]

MAURICE, Florence (2012): Mobile Webseiten. München : Carl Hanser Verlag

NIELSEN, JAKOB (2012): „Mobile Site vs. Full Site“.
⟨<http://www.nngroup.com/articles/mobile-site-vs-full-site/>⟩

[Stand: 10. April 2012. Zugriff: 30. Oktober 2013]

ONBILE.COM (2013): „Responsive Web Design Advantages“.
⟨<http://www.onbile.com/info/responsive-web-design-advantages/>⟩

[Stand: 22. März 2013. Zugriff: 28. Oktober 2013]

RABIN, Jo / MCCATHIENEVILE, Charles (2005): „Mobile Web Best Practices 1.0 - W3C Working Draft 20 December 2005“.

⟨<http://www.w3.org/TR/mobile-bp/#OneWeb>⟩

[Stand: 20. Dezember 2005. Zugriff: 03. November 2013]

VUGTS, Ralph (2012): „Responsive Web Design: Pros & Cons“.
⟨<http://engagingcomms.com/s/48/Responsive-Web-Design-Pros-Cons.html>⟩

[Stand: 07. August 2012. Zugriff: 01. November 2013]

W3C (2002): „Media types“.
⟨<http://www.w3.org/TR/2002/WD-CSS21-20020802/>⟩

[Stand: 2. August 2002. Zugriff: 29. Oktober 2013]

W3C (2013): „Home“.
⟨<http://www.w3.org/Mobile/>⟩

[Stand: 2013. Zugriff: 2013]

WIKIPEDIA CONTRIBUTORS (2013): „Fault tolerance“.
⟨http://en.wikipedia.org/wiki/Fault_tolerance⟩

[Stand: 23. November 2013. Zugriff: 25. November 2013]

WILLCOX, Matt (2012): „Adaptive Images“.
⟨<http://adaptive-images.com/>⟩

[Stand: 16. April 2012. Zugriff: 20. November 2013]

WROBLEWSKI, Luke (2009): „Mobile First“.
〈<http://www.lukew.com/ff/entry.asp?933>〉
[Stand: 3. November 2009. Zugriff: 10. November 2013]

WROBLEWSKI, Luke (2010): „Mobile First“.
〈<http://www.lukew.com/presos/preso.asp?26>〉
[Stand: 5. April 2010. Zugriff: 10. November 2013, Präsentation auf An Event Apart Seattle, WA]

WROBLEWSKI, Luke (2011): Mobile First. New York City : A Book Apart, New York

ZILLGENS, Christoph (2013): Responsive Webdesign. München : Carl Hanser Verlag

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Leipzig, 29.11.2013

Lars Häuser