

RESPONSIVE DESIGN

Afstudeerscriptie Quentin Buiteman



Responsive Design

Eén website, meerdere apparaten

Auteur

Naam	Quentin Buiteman
Studentnummer	1563478
E-mailadres	quentin.buiteman@student.hu.nl
Opleidingsinstituut	Hogeschool Utrecht
Bachelor	Communicatiesystemen
Afstudeerrichting	Digitale Media & Communicatie
Faculteit	Communicatie & Journalistiek
Afstudeerdatum	Juli 2013

Afstudeerbegeleider

Naam	Eric Leltz
Organisatie	Hogeschool Utrecht
Faculteit	Communicatie & Journalistiek

Bedrijfsbegeleider

Naam	Rogier van Roon
Organisatie	Los Banditos
Functie	Bedrijfsleider

19 juni 2013

Voorwoord

Mijn afstudeerperiode heeft me in laten zien dat ik nu bezig ben met iets wat ik echt leuk vind. Het onderzoek wat ik heb gedaan, in combinatie met het praktijkwerk wat ik verricht heb, hebben mij dag en nacht beziggehouden. Als je “na werktijd” en in de weekenden nog bezig bent met je afstudeerproject “omdat je het leuk vindt”, dan kun je wel zeggen dat je je doel gevonden hebt. Deze periode heeft mij in laten zien dat ik twee jaar geleden de goede keuze heb gemaakt om mij te richten op programmeren.

Misschien is het saai om te vertellen dat je afstudeerperiode zonder slag of stoot is verlopen, maar dat is eigenlijk wel wat er gebeurd is. Van dag één tot de laatste dag heb ik netjes op schema gelegen en heb ik met plezier kunnen werken aan mijn afstudeerproject. Daarom wil ik graag Rogier, Stefan, Navarone, Leroy, Caine (MAND!), Lennard en de stabiele internetverbinding bedanken voor de fijne tijd die ik heb meegemaakt bij Los Banditos.

Natuurlijk zijn de laatste vier jaar studeren niet helemaal vlekkeloos verlopen. Er waren hoogtepunten, zoals het binnenslepen van talloze goede cijfers, maar ook dieptepunten, zoals de toets Concepting. Gelukkig waren daar dan de beide Jonathans (of Jonathannen, ik weet het niet), Reinout en Coen die altijd klaar stonden om me weer op te vrolijken met een lekker pilsje (of hilarische, nachtelijke voicemailen indien ze dat pilsje zonder mij opdronken).

Tot slot wil ik Iris nog bedanken omdat ze bijna mijn gehele studentenperiode mijn trouwe steun en toeverlaat is geweest.

Quentin Buiteman

10 juni 2013

Samenvatting

Responsive design is een trend van de laatste jaren en geeft de webontwikkelaar de mogelijkheid om één website te maken die zich aanpast aan alle mogelijke schermresoluties. Los Banditos wil op deze trend inspringen en alle toekomstige websites met een responsive design afleveren. Hiervoor wil Los Banditos een nieuwe werkwijze geïmplementeerd hebben die dit bewerkstelligt.

Om dit alles te realiseren is er een onderzoek verricht naar responsive design. Dit onderzoek geeft antwoord op de hoofdvraag *“Hoe ziet de werkwijze van Los Banditos eruit na het toevoegen van responsive design aan het proces?”*. Daarnaast is Los Banditos momenteel bezig met een projectmanagementtool genaamd MijnWerkers. Los Banditos wil deze tool op alle mogelijke apparaten aanbieden en wil deze dus responsive maken. Tijdens het gehele proces is er een interne analyse uitgevoerd om de huidige werkwijze en de toekomstvisie van Los Banditos te leren kennen.

Uit het theorieonderzoek is gebleken dat responsive design wordt herkend aan twee kenmerken: een vloeiende lay-out die gebruik maakt van een flexibele grid en afbeeldingen die in een flexibele context werken. Om dit te realiseren wordt er gebruik gemaakt van drie technieken: het gebruik van relatieve waardes, het gebruik van media queries en het gebruik van de viewport meta tag.

Het gebruik van deze technieken wordt vergemakkelijkt door de mobile first aanpak. Hierbij wordt er eerst voor de kleinere schermen ontwikkeld, waarna de website uitgebreid wordt voor grotere schermen. Van klein naar groot werken is een stuk makkelijker dan andersom. Daarnaast is het verstandig om gebruik te maken van content-out design. Dit zorgt ervoor dat de media queries gebaseerd worden op de inhoud van de website, in plaats van op de beschikbare apparaten.

Uit de interne analyse bleek dat de huidige werkwijze van Los Banditos structuur en specialisme tekort komt. Door middel van scrum wordt getracht dit gat te vullen. Scrum zorgt ervoor dat een project niet meer in handen is van één persoon, maar juist van meerdere personen. Hierdoor wordt er meer gekeken naar de specialiteit van de werknemer. Daarnaast biedt scrum een vast proces dat herhaaldelijk doorlopen wordt. Dit zorgt voor de structuur binnen de projecten. Rogier, bedrijfsleider van Los Banditos, functioneert als de scrummaster. Tevens zijn er twee frameworks ontwikkeld die door de werknemers van Los Banditos gebruikt kunnen worden om het ontwikkelproces te versnellen.

De case studies wezen uit dat de frameworks klaar zijn voor gebruik. Ook is de projectmanagementtool van Los Banditos, MijnWerkers, responsive gemaakt. Deze case studies hebben bewezen dat Los Banditos met gebruik van de nieuwe werkwijze op een snelle en effectieve manier websites met een responsive design kan leveren.

Inhoudsopgave

Begrippenlijst	vii
1. Inleiding.....	1
1.1. Aanleiding en hoofdvraag	1
1.2. Structuur en deelvragen.....	2
1.3. Leeswijzer.....	3
2. Responsive design	5
2.1. Definitie	5
2.2. Technieken	9
3. Apparaten.....	17
3.1. Smartphones	17
3.2. Tablets	17
3.3. Desktop & Laptop	18
3.4. Gebruiksverschillen.....	18
4. Relevante trends.....	25
4.1. Toekomstige trends	25
4.2. Relatie met responsive design.....	27
5. Werkwijze.....	29
5.1. Huidige werkwijze.....	29
5.2. Toekomstvisie.....	30
5.3. Nieuwe werkwijze	30
5.4. Voor- en nadelenvergelijk	34
6. Case studies.....	37
6.1. Frameworks	37
6.2. MijnWerkers.....	38
6.3. Leerpunten	43
7. Conclusie	45
Literatuurlijst.....	47
Bijlagen	53
Bijlage A. Meest gebruikte browsers	53
Bijlage B. Schermresoluties van verschillende apparaten.....	54
Bijlage C. Voorbeeldpagina met kolommenindeling	55
Bijlage D. Lijst met mogelijke media query waardes	56
Bijlage E. Overzicht van browsers die media queries ondersteunen	57

Begrippenlijst

4G	De vierde generatie van het mobiele internet.
Blogbericht	Een bericht op een online dagboek.
Boxmodel	Een model dat door CSS wordt gebruikt om de grootte van een element te meten.
Browse	Het bekijken van webpagina's op het internet.
CSS	CSS is een afkorting voor de programmeertaal Cascading Stylesheets. Een CSS bestand bevat de code die de opmaak van een webpagina genereert.
Emulator	Een programma dat een apparaat of een ander programma na kan doen.
Flash	Een computerprogramma waarmee animaties, filmpjes en webapplicaties gebouwd kunnen worden.
Full HD	Full HD wordt gebruikt om een schermresolutie van 1920 bij 1080 pixels aan te geven.
Grid	Een verzameling van hulplijnen die gebruikt kunnen worden om het indelen van een webpagina te vergemakkelijken.
HTML	De afkorting van de programmeertaal HyperText Markup Language. Dit is de code die de structuur van een webpagina genereert.
HTTP request	Een verzoek van een browser aan een server voor het opvragen van een HTML bestand.
Native	De aanduiding die gebruikt wordt om aan te geven dat een programma voor een specifiek apparaat ontwikkeld is (met gebruik van de bijbehorende ontwikkelingsmethode).
Plugin	Een aanvullende module om de functies van een programma uit te breiden.
Prefixes	Een voorvoegsel dat gebruikt wordt om sommige CSS styles zichtbaar te maken aan de bijbehorende browser.
Search Engine Optimization	Het optimaliseren van een website zodat hij makkelijker gevonden wordt in zoekmachines.
Subdomein	Een subdomein is een onderverdeling van een bestaande domeinnaam.
Tag	Een aanduiding die gebruikt wordt door een browser om te herkennen hoe de inhoud van een document weergegeven moet worden.
Viewport	Gedeelte van een webpagina dat zichtbaar is voor een gebruiker.

1. Inleiding

1.1. Aanleiding en hoofdvraag

Het online landschap verandert. Er komen steeds meer apparaten waarmee mensen online kunnen en willen komen. Elk apparaat heeft zo zijn eigen kenmerken; het heeft zijn eigen schermresolutie, schermgrootte, gebruikersinterface en besturingsmogelijkheden. Al deze mogelijkheden en talloze combinaties zorgen voor een probleem voor de webontwikkelaar. Hij kan niet meer uitgaan van een vast product waarmee een website bekeken wordt. Door deze recente ontwikkeling moest de webwereld komen met een oplossing zodat websites overal op bekeken kunnen worden.

Responsive design is de oplossing waarmee de webwereld is gekomen. Deze trend is de laatste jaren in opkomst. Het geeft de webontwikkelaar de mogelijkheid om één website te maken die zich aanpast aan alle mogelijke schermresoluties. Of de website nu bezocht wordt met een tablet, smartphone, laptop of desktop, de gebruiker krijgt altijd de optimale websiteweergave te zien.

Los Banditos is een groeiend internetbedrijf dat op deze trend in wil springen. Ze ziet de enorme groei in mobiele apparaten als een kans, in plaats van een bedreiging. Ze is er tevens van overtuigd dat een 'normale' website niet meer voldoende is om aan haar klanten te leveren. Los Banditos ziet dit als het beste moment om zich te specialiseren in responsive design. Om dit mogelijk te maken, wil ze een duidelijk onderzoek naar responsive design. Responsive design is namelijk de enige oplossing voor de groei in mobiele apparaten die toegepast kan worden op de core business van Los Banditos (het leveren van websites). De doelstelling van het onderzoek luidt: *"Binnen vijf maanden wil Los Banditos een nieuwe werkwijze geïmplementeerd hebben, waardoor alle toekomstige websites gebruik maken van een responsive design"*.

Het onderzoek geeft antwoord op de hoofdvraag *"Hoe ziet de werkwijze van Los Banditos eruit na het toevoegen van responsive design?"*.

Daarnaast is Los Banditos momenteel bezig met het ontwikkelen van een projectmanagementtool genaamd MijnWerkers. Deze tool wil Los Banditos op alle mogelijke apparaten beschikbaar maken. Het resultaat dat uit het onderzoek naar responsive design komt, wordt daarom ook toegepast op MijnWerkers.

Op basis van de onderzoeksresultaten is een adviesrapport voor Los Banditos opgesteld. Dit adviesrapport vormt de basis voor alle toekomstige websites van Los Banditos. Het adviesrapport is een los document voor intern gebruik bij Los Banditos. Dit adviesrapport is, na overleg, simpel van opbouw gemaakt. Hierdoor is het makkelijker om te lezen voor een nieuwe stagiair. Het adviesrapport dient als kennismaking met Los Banditos voor de nieuwe stagiairs.

1.2. Structuur en deelvragen

De scriptie is een onderzoeksdocument. Deze is dus gebaseerd op een hoofdvraag, met deelvragen als ondersteuning. Deze deelvragen zijn gericht op het vaststellen van de theorie en het praktijkonderdeel. De theorie richt zich op de informatie rondom responsive design, de verschillende apparaten die momenteel op de markt te vinden zijn en de verschillende trends die binnen de wereld van de webontwikkeling gaande zijn. Daarnaast is de werkwijze van Los Banditos op de schop genomen, zodat responsive design hierin geïmplementeerd kan worden. Het praktijkonderdeel richt zich op het responsive maken van MijnWerkers en het testen van de frameworks die gemaakt zijn voor de nieuwe werkwijze. Om de hoofdvraag te ondersteunen, zijn er deelvragen opgesteld voor elk van deze onderwerpen.

Responsive design

- Wat is de definitie van responsive design?
- Welke technieken zijn het beste toepasbaar voor het ontwikkelen van responsive design?
 - Welke technieken kunnen gebruikt worden voor responsive design?
 - Wat zijn de voor- en nadelen van deze technieken?
- Welke browsers ondersteunen responsive design?
- Welke relatie heeft de inhoud van een website met responsive design?
- Hoe wordt ervoor gezorgd dat de laadtijd optimaal blijft?
 - Wat voor relatie heeft dit met de 'vorm' waarin de website bekeken wordt?
 - Hoeveel impact heeft het apparaat waarop de website bekeken wordt hierop?
- Welke middelen zijn er nodig voor het testen van een responsive website?

Apparaten

- Wat zijn de apparaten waar rekening mee moet worden gehouden?
 - Welke apparaten ondersteunen responsive design?
 - Op wat voor resolutie draaien ze?
 - Wat voor schermgrootte hebben ze?
 - Hoe ziet het toekomstbeeld eruit?
- Wat is er bekend over de gebruiksmanier van elk van de apparaten?
 - Hoe wordt een website bekeken op de verschillende apparaten?
 - Wat betekent dit voor de lay-out van de website?

Relevante trends

- Met welke trends op het gebied van webontwikkeling moet rekening gehouden worden bij het ontwikkelen van responsive design?
 - Welke trends zijn er op het gebied van webontwikkeling?
 - Welke trends kunnen een impact hebben op responsive design?
 - Op welke manier kunnen deze trends in het voor- en nadeel werken van responsive design?

Werkwijze

- Hoe kan responsive design verwerkt worden in een werkwijze die aansluit op de visie van Los Banditos?
 - Wat is de huidige manier van werken?
 - Welke stappen moeten er genomen worden om over te gaan naar een werkwijze met responsive design?
- Hoe verhoudt een werkwijze met responsive design zich ten opzichte van de huidige manier van werken?
 - Wat zijn de voor- en nadelen van responsive design?
 - Wat zijn de voor- en nadelen van de huidige manier van werken?
 - Hoe verhouden deze zich ten opzichte van elkaar?
- Hoe ziet een werkwijze binnen Los Banditos eruit waarbij gebruik gemaakt wordt van responsive design?
- Hoe ziet de vernieuwde werkwijze eruit na het verwerken van de kritiekpunten van de case studies?

MijnWerkers

- Uit welke componenten bestaat MijnWerkers?
 - Wat is de prioriteit van elk component?
 - Hoe is de werking van deze componenten toe te passen op de gebruiksmanner van elk apparaat?
 - Welke componenten zijn toepasbaar op de verschillende apparaten?
- Welke kritiekpunten zijn voortgekomen tijdens het responsive maken van MijnWerkers?
 - In welke fase van het proces deed zich dit voor?
 - Wat voor impact had dit?
 - Hoe had dit voorkomen kunnen worden?
 - Op welke manier kan dit veranderd worden in de vernieuwde werkwijze?

1.3. Methodologie

Tijdens de afstudeerperiode wordt gebruik gemaakt van een aantal verschillende methodes:

- Een literatuuronderzoek om de theorie rondom responsive design, de relevante apparaten en de relevante trends vast te stellen.
- Een interne analyse om de huidige werkwijze en de toekomstvisie van Los Banditos in kaart te brengen. Deze interne analyse bestaat uit het opdoen van praktijkervaring en het afnemen van een interview. Op basis van deze analyse is de nieuwe werkwijze opgesteld.
- Het uitvoeren van case studies om de resultaten van het literatuuronderzoek en de interne analyse in de praktijk te brengen. Het responsive maken van MijnWerkers dient ook als case study.
- Voor het adviesrapport is een best practices analyse uitgevoerd. Hierbij is gekeken naar een aantal onderdelen (navigatie, tabellen, kolommen en formulieren) van praktijkvoorbeelden van responsive websites.

1.4. Leeswijzer

In hoofdstuk 2 (responsive design) wordt de theorie rondom responsive design besproken. In hoofdstuk 3 (apparaten) worden de verschillende apparaten besproken die momenteel toegang hebben tot het internet (en daarmee dus relevant zijn voor een responsive design). In hoofdstuk 4 (relevante trends) worden trends besproken die momenteel in de wereld van de webontwikkeling te vinden zijn en een eventuele impact (kunnen) hebben op responsive design. Hoofdstuk 5 (werkwijze) richt zich op de huidige en de vernieuwde werkwijze van Los Banditos. De verschillende case studies worden besproken in hoofdstuk 6 (case studies). Hierin wordt het responsive maken van MijnWerkers en het testen van de frameworks besproken. Afsluitend is in hoofdstuk 7 (conclusie) een samengevatte versie van het advies aan Los Banditos te vinden.

2. Responsive design

Voorafgaand aan mijn afstudeerperiode heb ik verschillende leraren, collega's en klasgenoten gesproken die ieder weer een eigen, unieke uitleg hadden over wat responsive design is. Er is daarom een duidelijke analyse van het begrip responsive design gemaakt, zodat hier geen verwarring meer over kan ontstaan. Hierbij wordt gekeken naar de verschillende kenmerken, de voor- en nadelen (en hoe hier op in te spelen is) en de te gebruiken technieken.

2.1. Definitie

In 2005 gaf het W3C (World Wide Web Consortium), een online gemeenschap die zich bezighoudt met het ontwikkelen van webstandaarden, aan dat ze voorzagen dat de verscheidenheid in mobiele apparaten de groei van het mobiele web tegen zou werken (Archer & Mitukiewicz, 2005). Hierna gaven ze hun toekomstvisie, de "One Web" visie:

"One Web' vision of a seamlessly integrated Internet remains the long-term objective of the Mobile Web Initiative ... Our immediate goal is to define a set of forward looking Best Practices guidelines that - when followed by authors and developers - are likely to make their content accessible with equal ease to users of desktop and mobile devices of certain assumed capabilities." (Archer & Mitukiewicz, 2005)

Het W3C zegt hiermee toe te willen werken naar een set webstandaarden waardoor een gebruiker altijd ten volste kan genieten van het web (ongeacht het apparaat waarmee hij op het web komt). In 2008 kwam het W3C terug op hun One Web visie. Ze gaven hierbij duidelijker aan dat de nadruk moet liggen op het aanbieden van dezelfde informatie en diensten, ongeacht het gebruikte apparaat (Rabin & McCathieNevile, 2008).

Ethan Marcotte, de grondlegger van responsive design, brengt in een [blogbericht](#) de complicaties van de hedendaagse webontwikkelaar aan het licht. Inconsistente schermgroottes, inconsistente schermresoluties, gebruikersvoorkeuren en de geïnstalleerde lettertypes van de gebruiker zijn slechts een aantal van de complicaties die hij noemt. Hij geeft hierbij aan dat de webontwikkelaar zich continu aanpast aan deze eisen, maar dat dit niet langer zo kan (Marcotte, Responsive Web Design, 2010). In dit blogbericht komt hetzelfde voorbeeld naar voren als gebruikt door het W3C. Marcotte geeft het resultaat van het Mobile Internet Report 2010 weer, waarin wordt geconcludeerd dat het mobiele [browsen](#) in zo'n drie tot vijf jaar vaker voor gaat komen dan het traditionele browsen vanaf de desktop PC (Morgan Stanley Research, 2010).

Een originele oplossing voor het steeds grotere mobiele web was de bekende "mobiele website". Zoals Marcotte (Responsive Web Design, 2010) aangeeft, wilden veel opdrachtgevers graag een "iPhone website". Dit werd dan opgelost door verschillende [subdomeinen](#) aan te maken, waardoor de gebruiker op de bijbehorende website terecht kwam. Maar wat is dan de volgende stap? Een iPad website? Een Nokia N90 website? Er kan geen aparte website gemaakt worden voor elke smartphone of tablet die op de markt komt.

Vanuit de webwereld is er behoefte aan een oplossing voor het continu veranderende “landschap”. Marcotte gaat in zijn blogbericht door met een oplossing vanuit de architectenwereld. Hier wordt gebruik gemaakt van een discipline die “responsive architecture” heet. Hierbij wordt gekeken hoe een fysieke ruimte kan reageren op de aanwezigheid van mensen. Door middel van een combinatie van verschillende technieken hebben architecten gebouwen gemaakt die uitbreiden naarmate er meerdere mensen binnenkomen (Fox & Kemp, 2009). Deze situatie is gelijk te stellen aan het web. Het web verandert continu en moet zich dus ook continu aanpassen. Marcotte stelt dat deze aanpak gebruikt moet worden in het web, om zo het probleem aan te pakken. Het web moet zich gaan richten op *responsive design*.

2.1.1. Kenmerken

In een later blogbericht geeft Marcotte drie kenmerken aan die essentieel zijn voor het maken van een responsive design (Marcotte, On being “responsive”, 2010):

- een vloeiende **lay-out** die gebruik maakt van een flexibele **grid**;
- afbeeldingen die in een flexibele context werken;
- het gebruik van media queries.

Persoonlijk ben ik het niet volledig eens met Marcotte. Ik ben van mening dat het derde punt, media queries, een oplossing (techniek) is in plaats van een kenmerk. Om deze reden kom ik in het subhoofdstuk technieken terug op het gebruik van media queries.

Een vloeiende lay-out met een flexibele grid

Voorheen werd er gebruik gemaakt van een statische paginawijdte. De pagina had een vast aantal pixels als breedte, welke niet flexibel aangepast werden. Hierbij werd dus totaal geen rekening gehouden met de gebruiker z’n schermgrootte en –resolutie. Meestal werd de website dan aangepast op basis van de toentertijd meest gebruikte schermresolutie (bijvoorbeeld 1024 bij 768 pixels). Alle kleinere schermresoluties hebben dan een probleem en moeten horizontaal gaan scrollen. Grotere schermresoluties hebben rechts en links ongebruikte ruimtes.

Het idee van een vloeiende lay-out is dat deze zich aanpast aan de schermgrootte en –resolutie van de gebruiker. Hierdoor krijgt de gebruiker altijd de lay-out te zien die voor zijn schermgrootte en –resolutie het beste is. Het maakt dan niet meer uit of je de website bezoekt op een tablet of smartphone, op een laptop of desktop, met een hoge of lage resolutie, je krijgt altijd de lay-out te zien die het beste geschikt is voor jouw specifieke combinatie.

Een populaire, nieuwe techniek is om een gridsysteem te gebruiken om lay-outs te maken (Gardner B. S., 2011). Een gridsysteem maakt gebruik van een set van basisstijlen voor kolommen, waardoor het makkelijker is om websites te ontwikkelen met gebruikersvriendelijke uitlijning van de inhoud.

De combinatie van deze twee technieken is essentieel voor het ontwerpen van een responsive design. Een flexibel gridsysteem zorgt ervoor dat de inhoud van een website altijd zichtbaar is op de juiste plek. Door middel van een vloeiende lay-out krijgt de gebruiker dus altijd de juiste lay-out te zien. Aangezien deze twee kenmerken onderdeel uitmaken van een responsive design, worden deze ook gebruikt bij de projecten van Los Banditos.

Afbeeldingen die in een flexibele context werken

Het gebruik van afbeeldingen is waarschijnlijk een van de grootste problemen die voorkomt bij het ontwikkelen van een responsive design. Doordat er gebruik gemaakt wordt van een vloeiende lay-out met een flexibele grid, zijn er veel verschillende groottes waaraan de inhoud zich aan moet passen. Met normale tekst is dit geen groot probleem. Afbeeldingen zijn echter een heel ander probleem. Je kunt een afbeelding niet zomaar vergroten of verkleinen, aangezien deze dan zijn kwaliteit verliest. Er moet dus gezorgd worden dat de plaatsing en grootte van afbeeldingen zich aanpassen aan de grootte van de pagina.

Dit punt heeft veel overeenkomst met de toepassing van de vloeiende lay-out die nodig is voor een responsive design. Een hogere schermresolutie wil niet zeggen dat alles groter moet zijn. Er moet veel nagedacht worden over de plaatsing van de inhoud. Daarnaast moet er altijd gelet worden op de behoefte van de gebruiker bij het plaatsen van deze inhoud. Welke inhoud is het belangrijkste en op welke plek moet deze dan staan? Onderstaand is een voorbeeld te zien van hoe responsive design er in theorie uit zou moeten zien. Duidelijk is te zien dat er gevarieerd wordt met de breedte en de indeling van de kolommen tussen desktop, tablet en mobiel.



Figuur 2.1: Voorbeeld van responsive design

Afbeeldingen brengen ook nog een apparaatgerelateerd probleem met zich mee. Sommige smartphones en tablets (voornamelijk de iPhone en iPad) hebben een hoge schermresolutie, maar maken gebruik van een device pixel ratio die hoger is dan 1 (Cant, 2012). Dit betekent dat CSS minder pixels herkent dan dat het scherm daadwerkelijk bevat. Het gevolg hiervan is dat een afbeelding niet op elk scherm even scherp is. De apparaten met een hogere device pixel ratio hebben een afbeelding met een hogere resolutie nodig. Een oplossing voor dit probleem met betrekking tot achtergrondafbeeldingen is te vinden in hoofdstuk 2.2.2. Om dit probleem op te lossen met betrekking tot afbeeldingen die in de structuur van de pagina staan, moet er gekeken worden naar JavaScript. Hier zijn verschillende plugins voor te gebruiken, zoals Retina.js (<http://retinajs.com/>).

Deze plugin zorgt ervoor dat de goede afbeelding geselecteerd wordt, voordat de pagina geladen wordt. Dit gaat dus niet ten koste van de laadtijd van de pagina.

2.1.2. Voor- en nadelen

Responsive design brengt zo zijn voor- en nadelen met zich mee (Design Lunatic, 2011). Opmerkelijk is dat de meeste nadelen een kwestie van tijd zijn, voordat ze teniet gedaan kunnen worden. Zo is er met de lancering van 4G mobiel internet al een groot probleem opgelost met betrekking tot de laadtijd van de pagina's (De Telegraaf, 2013). De browsersupport wordt tevens elk moment hoger.

Voordelen

- Eén [HTML](#) pagina betekent dat er minder onderhoud gepleegd hoeft te worden.
- Je bent altijd dezelfde [tags](#) aan het stijlen.
- Je krijgt nooit een horizontale scrollbar.
- Verschillende apparaten kunnen dezelfde link gebruiken om op de pagina te komen, in tegenstelling tot het gebruik van aparte subdomeinen voor mobiele websites.
- Je maakt je website 24/7 beschikbaar, doordat hij op elk apparaat toegankelijk is.
- Er wordt meer rekening gehouden met [Search Engine Optimization \(SEO\)](#) (Varsamis, 2012).

Nadelen

- Het is niet op alle gebruikte browsers toepasbaar. Internet Explorer 8 (6.4% van de gebruikers) en oudere versies van Firefox (7.3% van de gebruikers) ondersteunen het bijvoorbeeld nog niet (W3Schools, 2013).
- Het ontwikkelen van een responsive design kost tijd en geld.
- Het testen van een responsive design duurt langer.

2.1.3. Laadtijd

Het grootste probleem wat hierbij naar voren komt is de laadtijd van een responsive website. Dit is waarschijnlijk wel het grootste discussiepunt tussen voorstanders en tegenstanders van responsive design. Hoewel het maar de vraag is of dit een discussiepunt over responsive design is, of over de kennis van degene die het implementeert (Kadlec, 2012). Kadlec geeft al aan dat een techniek zo goed is als degene die hem implementeert. Als er een responsive website gemaakt wordt van 4MB, dan is het de schuld van de maker dat de laadtijd hoog is, niet van responsive design. Hij geeft een aantal technieken aan om de laadtijd te verlagen, zoals lazy loading en de anchor include pattern. Lazy loading zorgt ervoor dat de zwaarste onderdelen van een website (zoals plaatjes) pas later worden geladen, waardoor de belangrijkste functies van de website wel vroeg worden geladen (Coyier, 2012). Een anchor include pattern maakt gebruik van meerdere [includes](#), die allemaal verwerkt worden in één [HTTP request](#), om op die manier de verschillende onderdelen achter elkaar te laden, in plaats van tegelijkertijd. Zo kan er, net als bij lazy loading, gezorgd worden dat sommige onderdelen eerder laden dan andere onderdelen (Jehl, 2012). Door het implementeren van deze twee technieken, wordt de laadtijd van een responsive website drastisch verlaagd. Beide technieken hebben ongeveer hetzelfde effect. Voor Los Banditos zal gebruik gemaakt worden van lazy loading, aangezien deze het makkelijkste te implementeren is.

2.1.4. Testen

Een ander groot punt is het testen van een responsive design. Er zijn namelijk veel apparaten, browsers en besturingssystemen waar het responsive design op moet werken. De enige mogelijke manier om een responsive design uitvoerig te testen is om deze op zoveel mogelijk apparaten en in zoveel mogelijk browsers en besturingssystemen te testen. De initiële test kan het makkelijkste gebeuren door de browser te vergroten en te verkleinen, om te zien hoe de website reageert op verschillende schermgroottes. Er is echter pas een volledige test gedaan als de website op meerdere apparaten met verschillende eigenschappen bekeken is. Er zijn twee oplossingen voor het testen op meerdere apparaten (Verbeek, 2012). De meest simpele manier is om een [emulator](#) te gebruiken. Deze bootst de eigenschappen van verschillende apparaten na, zodat er getest kan worden. Een tweede oplossing is het gebruik van een openbaar testlab. Dit is een locatie waar een verzameling aan fysieke apparaten aanwezig is, waar iedereen gratis zijn responsive designs op kan testen. Netvlies heeft zo'n locatie beschikbaar gesteld in Breda.

Los Banditos heeft niet vaak langdurige projecten. Dit betekent dat de projecten over het algemeen niet meer dan één maand duren. Aangezien Los Banditos gelegen ligt in Rotterdam, is er gekozen om de korte projecten (minder dan een maand tijd) te testen met het gebruik van een emulator. Responsinator (<http://www.responsinator.com>) is gekozen, aangezien deze een goed overzicht geeft van een responsive design op de meestgekochte apparaten. Eventuele langdurige projecten kunnen getest worden in het openbare testlab van Netvlies in Breda.

2.2. Technieken

De benodigde technieken zijn terug te vinden in de kenmerken die Ethan Marcotte heeft gebonden aan responsive design: een vloeiende lay-out die gebruik maakt van een flexibele grid, afbeeldingen die in een flexibele context werken en media queries (Marcotte, Responsive Web Design, 2010). In principe zijn hier twee technieken te herkennen: een vloeiende layout met een flexibele grid (het gebruik van relatieve waardes) en media queries. Beide technieken worden gerealiseerd door het gebruik van HTML(5) en CSS(3). Ook is de [viewport meta](#) tag noodzakelijk (Yates, 2012), hoewel deze niet door Marcotte genoemd wordt. Responsive design bestaat dus uit de combinatie van deze drie technieken. Zonder een van deze technieken, zou een responsive design nooit volledig zijn.

2.2.1. Relatieve waardes

Designers vinden een statische grid fijner werken, omdat ze dan meer controle hebben. Bij een statische grid kan alles tenslotte op de pixel na uitgemeten worden. De eerste stap richting een flexibele grid is om deze controle wat verder los te laten. Een flexibele grid bestaat uit vier elementen: flexibele typografie, flexibele containers, flexibele margin en flexibele padding (Byers, 2011). We gebruiken hierbij de onderstaande formule (Marcotte, Responsive Web Design, 2011):

$$\text{target} \div \text{context} = \text{result}$$

Target: de grootte die we in willen stellen.

Context: de grootte op basis waarvan we het element af willen stellen: de basiswaarde.

Flexibele typografie

De techniek achter flexibele typografie zit in het gebruik van de eenheid voor de grootte van het lettertype. Tot nu toe drukken webontwikkelaars dit meestal in pixels (px) uit. Voor een flexibele typografie moet er echter gebruik gemaakt worden van em. Em is een eenheid die zich aanpast aan de basiswaarde voor de lettertypegrootte van de browser. Stel dat er een koptekst in het aangeleverde ontwerp staat met een grootte van 32px. De target zou in dit geval 32px zijn, de context 16px (1em is standaard 16px in de meeste browsers). Het result wat hieruit komt is 2em.

Als kanttekening hierbij geeft Ethan Marcotte nog een aantal tips. Het kan zijn dat het result op veel decimalen eindigt. Het is belangrijk om in dit geval zoveel mogelijk decimalen in te vullen. Ook geeft hij als tip aan om de gebruikte formule te vermelden achter de ingevulde waarde, zodat deze altijd als referentie gebruikt kan worden (Marcotte, Responsive Web Design, 2011).

```
font-size: 2em; /* 32px/16px */
```

Figuur 2.2: CSS code voor de grootte van het lettertype na het toepassen van de formule

Flexibele containers, margin & padding

Het meest gebruikte statische gridsysteem heeft een breedte van 960px (La, 2010). Voor dit voorbeeld wordt deze breedte dan ook aangehouden. Het verschil is dat er voor flexibele containers met procenten wordt gewerkt, niet met em's. Marcotte houdt voor de container van zijn responsive designs meestal een breedte van 90% van het venster aan. Het is noodzakelijk om dan de overige indeling van de pagina aan te passen aan dit percentage. Het verschil is dat er nu vermenigvuldigd moet worden met 100, zodat er een procentueel getal uit de formule komt. Voor het voorbeeld is er gebruik gemaakt van een lay-out met drie kolommen. De linkerkolom is 470px, de middelste kolom 300px en de rechterkolom is 150px. Op de linkerkolom zit een rechtse margin van 25px. Op de middelste kolom zit een rechtse padding van 15px. Elke kolom wordt omgezet naar een procentuele waarde, zodat de pagina altijd overeen komt met het scherm van de gebruiker. De context voor alle elementen is 960px. Deze waarde wordt daarna vermenigvuldigd met 100. In bijlage C is een voorbeeld van de pagina te zien. Deze techniek wordt ondersteund in alle browsers.

```
#container {  
    width: 960px;  
}  
#links {  
    width: 470px;  
    margin-right: 25px;  
}  
#midden {  
    width: 300px;  
    padding-right: 15px;  
}  
#rechts {  
    width: 150px;  
}
```

Figuur 2.3: CSS code van de oude situatie (voor het toepassen van de formule)

```
#container {
    width: 90%;
}
#links {
    width: 48.958333%; /* (470/960) * 100 */
    margin-right: 2.604166%; /* (25/960) * 100 */
}
#midden {
    width: 31.25%; /* (300/960) * 100 */
    padding-right: 1.5625%; /* (15/960) * 100 */
}
#rechts {
    width: 15.625%; /* (150/960) * 100 */
}
```

Figuur 2.4: CSS code van de nieuwe situatie (na het toepassen van de formule)

Hoewel deze techniek centraal staat bij de originele opzet van responsive design, is er wel voor een lichtelijk andere aanpak gekozen voor de websites van Los Banditos. Dit heeft te maken met de denkwijze om content-out te ontwerpen. Kortgezegd houdt dit in dat het uiterlijk van een website altijd gebaseerd is op de inhoud van de website. Content-out design wordt verder in detail besproken in hoofdstuk 3.4.1. In dit hoofdstuk wordt echter wel de techniek besproken die gekozen is voor Los Banditos. Allereerst wordt er gebruik gemaakt van een container met een maximale breedte. De pagina moet zich namelijk aanpassen aan de inhoud. Er is dan geen reden om de pagina verder te spreiden dan dat nodig is. Dit zorgt alleen maar voor een uitgerekte pagina, welke grafisch niet aantrekkelijk is. De maximale breedte van de inhoud wordt dus bepaald door de container, of beter wrapper te noemen. Daarnaast wordt de inhoud gecentreerd.

```
.wrapper {
    margin: 0 auto; /* Centreren van de inhoud */
    max-width: 960px; /* Maximale breedte instellen */
}
```

Figuur 2.5: CSS code voor het centreren van de inhoud met een maximale breedte

Tevens wordt hier gebruik gemaakt van een class (wrapper), in plaats van een id (container). Hierdoor kan deze stijl op meerdere locaties toegepast worden, bijvoorbeeld voor het centreren van de inhoud van de footer of de header van de pagina.

Vervolgens wordt er gebruik gemaakt van een nieuwe functie van CSS3: box-sizing. Deze functie zorgt ervoor dat het **boxmodel** anders ingedeeld wordt, wat perfect is voor het gebruik van procenten. Het traditionele boxmodel telt de **padding** en de **border** van een element op bij de ingestelde breedte (width) of hoogte (height) van dat element. Een element met een width van 50%, een padding van 20px en een border van 1px, wordt dus in totaal 50% van de pagina breed + 21px aan elke zijde erbij opgeteld. Door de box-sizing op border-box te zetten, worden de padding en border niet opgeteld bij de width van het element, maar er vanaf gehaald. Een element met width 50%, is dus ook daadwerkelijk 50% breed, ongeacht de padding en border van dat element. Dit zorgt ervoor dat de breedtes van de elementen veel makkelijker uitgerekend kunnen worden, zonder dat deze breder

worden dan het aangegeven percentage. Deze CSS-code wordt in elke “moderne” browser ondersteund (indien de [prefixes](#) toegevoegd zijn). Tevens werkt hij ook vanaf Internet Explorer 8 (Irish, 2012). Deze wordt ingesteld op elk element om het uitrekenen van de breedtes van de elementen te vergemakkelijken.

```
* {
    -webkit-box-sizing: border-box; /* Safari 5.0-, Chrome 9.0- */
    -moz-box-sizing: border-box; /* Firefox */
    box-sizing: border-box; /* Internet Explorer 8.0+, Safari 5.1+,
                                Chrome 10.0+, Opera */
}
.wrapper {
    max-width: 960px;
    margin: 0 auto;
}
#links {
    width: 48.958333%; /* (470/960) * 100 */
    margin-right: 2.604166%; /* (25/960) * 100 */
}
#midden {
    width: 32.8125%; /* (315/960) * 100 */
    padding-right: 15px;
}
#rechts {
    width: 15.625%; /* (150/960) * 100 */
}
```

Figuur 2.6: CSS code voor de pagina met het gebruik van de nieuwe technieken

2.2.2. Media queries

CSS stylesheets hebben altijd al een soort van apparaatherkenning gehad, in de vorm van mediatypes. Bij het laden van een stylesheet in de **head** van de HTML pagina, kon dus duidelijk gemaakt worden voor welk apparaat die specifieke stylesheet bedoeld was (Marcotte, Responsive Web Design, 2010).

```
<link rel='stylesheet' type='text/css' href='screen.css'
      media='screen' />
<link rel='stylesheet' type='text/css' href='print.css'
      media='print' />
```

Figuur 2.7: HTML code voor een stylesheet alleen voor op het scherm en alleen voor een geprinte pagina

Het W3C is met de ontwikkeling van CSS3 hier verder op door gaan bouwen en heeft de media queries ontwikkeld. Door middel van media queries kun je specifieke karakteristieken van een pagina inspecteren.

```
<link rel='stylesheet' type='text/css' href='screen.css'
      media='screen and (max-width: 480px)' href='phone.css' />
```

Figuur 2.8: HTML code voor een stylesheet met de media query max-device-width

De bovenstaande stylesheet werkt dus alleen als het scherm een maximale breedte van 480 pixels heeft. Dit betekent dat deze wel op een iPhone te zien is, maar niet op een normale desktop. Door middel hiervan kunnen er verschillende styles toegepast worden voor apparaten met verschillende kenmerken. In bijlage D is een lijst met mogelijke media query waardes te zien (Rivoal, 2012).

Voor een responsive design wordt meestal alleen gebruik gemaakt van de max-width en min-width media queries, aangezien deze kunnen beïnvloeden voor welke schermresolutie een specifieke opmaak gebruikt wordt. Dit is tenslotte de volledige gedachte achter responsive design. De gekozen maten worden momenteel vaak afgesteld op basis van de schermresolutie van een apparaat. Deze maten worden ook wel 'breekpunten' genoemd. Bij elk breekpunt wordt andere CSS voor de pagina ingeladen. Voor Los Banditos wordt er ook gebruik gemaakt van de max-width en min-width media queries. Er wordt echter geen gebruik gemaakt van breekpunten, maar van optimalisatiepunten (op basis van de inhoud van de pagina). Dit heeft opnieuw te maken met content-out design, wat verder wordt behandeld in hoofdstuk 3.4.1.

De media query is tevens toepasbaar binnen de stylesheet, als toevoeging van een `@media` of een `@import` regel.

```
@media screen and (max-width: 480px) { /* inhoud */ }
@import url('iphone.css') screen and (max-width: 480px);
```

Figuur 2.9: Een media query toegevoegd aan een @media en een @import regel

Aangezien media queries een CSS3 techniek zijn, worden deze nog niet ondersteund door alle browsers (Summer, 2012). Dit is dan ook het enige probleem waar tegenaan wordt gelopen in het geval van browserondersteuning. In bijlage E is een overzicht te zien van de browsers die media queries ondersteunen (Can I Use, 2013). Hierin is te zien dat veel browsers het al ondersteunen. Het grootste struikelblok zit nog bij Internet Explorer, die pas vanaf versie 9 media queries ondersteunt. Samen staan Internet Explorer 8 en lager nog gelijk voor zo'n 12% van de markt van webbrowsers (februari 2013), waar dit vorig jaar nog zo'n 24% was. Alle mobiele browsers ondersteunen media queries. Responsive design wordt hedendaags ondersteund door ongeveer 88% van de webbrowsers en door alle mobiele browsers. Om media queries toch in Internet Explorer 6, 7 en 8 te laten werken, kan er gebruik gemaakt worden van de JavaScript plugin Respond.js (<https://github.com/scottjehl/Respond>).

Afbeeldingen

Zoals besproken in hoofdstuk 2.1.1, zijn er apparaten die een afbeelding met een hogere resolutie nodig hebben, om te zorgen dat deze volledig scherp is op het beeldscherm. Media queries kunnen hierbij helpen, aangezien zij ook de device pixel ratio kunnen herkennen. Dit werkt echter alleen met een achtergrondafbeelding, niet met afbeeldingen die in de structuur van de website staan (HTML).

```
#image {
  width: 50px;
  height: 50px;
  background-image: url('../images/image.png');
  background-repeat: no-repeat;
  background-position: center;
}
```

Figuur 2.10: CSS code voor een achtergrondaafbeelding.

```
@media
only screen and (-webkit-min-device-pixel-ratio : 2),
only screen and (min-device-pixel-ratio : 2) {
  #image {
    background-image: url('../images/image-high-res.png');
    background-size: 50px 50px;
  }
}
```

Figuur 2.11: CSS code voor een achtergrondaafbeelding voor schermen met een hoge device pixel ratio

Mobile First

Het gebruik van media queries ligt dichtbij het gebruik van de mobile first ideologie. Deze denkwijze houdt in dat er eerst voor mobiel ontwikkeld wordt (de meest simpele weergave), om vervolgens uit te gaan breiden naar de grotere weergaves (Wroblewski, Mobile First, 2011). Het grootste voordeel hieraan is dat er geen extra code voor styles (en dus extra laadtijd) bijkomt als de minimale weergave geladen wordt. De minimale weergave van een responsive website wordt grotendeels geladen op een mobiel apparaat, welke meestal toegang heeft tot een mobiele internetverbinding. Deze verbinding is een stuk langzamer dan de gemiddelde internetverbinding waarmee een desktop of laptop verbonden is (Artman, 2011).

Door het gebruik van mobile first, wordt er getracht om zo min mogelijk data in te laden op de minimale weergave. De media queries worden gebruikt om styles toe te voegen bij de grotere weergaves, in plaats van styles te overschrijven voor kleinere weergaves van de website. Vaak gebeurt dit nog andersom, aangezien veel websites responsive gemaakt worden, in plaats van dat ze responsive opgezet worden. Ook bij veel nieuwere websites worden vaak nog steeds de grotere weergaves als eerste gemaakt, terwijl de meest optimale methode voor de laadtijd van een website juist andersom is. Het enige nadeel wat mobile first met zich meebrengt is de gewinningsfactor. Aangezien het een andere aanpak is dan dat veel ontwikkelaars gewend zijn, is er een lichte leercurve aanwezig. Aangezien Los Banditos over gaat op een andere werkwijze, is dit het perfecte moment om ook mobile first mee te nemen in deze werkwijze.

2.2.3. Viewport meta tag

De viewport **meta** tag zorgt ervoor dat de browser herkent met welke viewport je de website bekijkt. Als deze niet ingesteld wordt, geeft hij standaard de viewport van de desktopversie. Dit wil zeggen dat hij de breedte van de pagina (viewport) niet op het juiste aantal pixels instelt. Om dit te

voorkomen moet altijd de viewport meta tag toegevoegd worden aan de head tag van de HTML pagina. Deze tag is vereist in het geval van een responsive design. Het flexibele grid werkt anders niet optimaal en past zich niet aan het apparaat aan. Opmerkelijk is dat Marcotte deze tag niet heeft vermeld in zijn blogbericht over de eisen voor een responsive design. Deze is echter wel terug te vinden in zijn gepubliceerde boek (Marcotte, Responsive Web Design, 2011). Voor websites van Los Banditos wordt deze tag ook gebruikt.

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
```

Figuur 2.12: De viewport meta tag

De code is opgebouwd uit drie onderdelen. Allereerst geeft het aan dat de inhoud van de meta tag bedoeld is voor de viewport van de pagina, door middel van het name-element. Vervolgens geeft het nog drie kenmerken mee door middel van het content element. Allereerst zorgt het dat de breedte van de pagina ingesteld wordt op de breedte van het apparaat waarop de pagina bekeken wordt (width=device-width). Vervolgens zorgt hij ervoor dat de pagina altijd met een schaal van 1:1 geopend wordt (initial-scale = 1). Daarna zorgt hij ervoor dat gebruikers niet kunnen inzoomen op de pagina (maximum-scale = 1). Deze laatste optie is gebonden aan de manier van het implementeren van responsive design, welke dus optioneel is. Het gevaar is dat het slechtziende mensen geen toegang geeft tot de site, aangezien zij een inzoommogelijkheid nodig hebben. Om deze reden wordt deze optie niet verwerkt bij de responsive websites van Los Banditos.

Slotbeschouwing

Responsive design is een groeiende techniek in de wereld van webontwikkeling om te voldoen aan het almaar uitbreidende aanbod van apparaten met internettoegang. Door middel van deze techniek kunnen websites geschaald worden, waardoor ze goed te gebruiken zijn op alle denkbare apparaten, met alle denkbare schermresoluties en -groottes. Een vloeiende lay-out met een flexibele grid en flexibele afbeeldingen zijn de twee kenmerken van een responsive design.

De twee grootste nadelen van responsive design zijn de laadtijd en het testen van een responsive design.

Het gebruik van responsive design wordt gerealiseerd door het gebruik van drie verschillende technieken: het toepassen van een vloeiende lay-out met een flexibele grid, het gebruik van media queries en het toepassen van de viewport meta tag. Een flexibele grid wordt bereikt door het gebruik van relatieve waarden voor de opbouw van de pagina. Media queries geven aan wat er moet gebeuren als er een bepaalde status van een pagina behaald wordt (bijvoorbeeld een bepaalde paginabreedte). De viewport meta tag zorgt ervoor dat de browser rekening houdt met de kenmerken van het apparaat (zoals de breedte van de pagina gemeten in pixels (viewport)).

In dit hoofdstuk zijn de volgende deelvragen behandeld:

- Wat is de definitie van responsive design?
- Welke technieken zijn het beste toepasbaar voor het ontwikkelen van responsive design?
 - Welke technieken kunnen gebruikt worden voor responsive design?
 - Wat zijn de voor- en nadelen van deze technieken?
- Welke browsers ondersteunen responsive design?
- Hoe kan ervoor gezorgd worden dat de laadtijd optimaal blijft?
 - Wat voor relatie heeft dit met de 'vorm' waarin de website bekeken wordt?
 - Hoeveel impact heeft het apparaat waarop de website bekeken wordt hierop?
- Welke middelen zijn er nodig voor het testen van een responsive website?

3. Apparaten

De verschillende apparaten zijn een essentieel onderdeel van responsive design. Zij zijn de reden dat er responsive design nodig is. De relatie met responsive design en de schermgrootte en –resolutie van deze apparaten is daardoor erg hoog. Dit hoofdstuk geeft duidelijkheid over de relatie van de apparaten met een responsive design en tot in hoeverre Los Banditos hier rekening mee moet houden. Dit gebeurt door te kijken naar de verschillende kenmerken van de apparaten. Ook wordt er gekeken naar de gebruiksverschillen tussen de verschillende apparaten. Begeleidend bij dit hoofdstuk kan bijlage B gebruikt worden met informatie over de gebruikte schermresoluties.

3.1. Smartphones

Er zijn talloze verschillende smartphones te krijgen. Er zijn daardoor ook een oneindig aantal mogelijkheden met betrekking tot de schermgrootte en –resolutie. Om deze reden wordt er een algemeen beeld van de gemiddelde waardes geschept. De trends van de laatste jaren zijn steeds grotere schermen en hogere resoluties. Hierdoor is er voor de smartphones dus geen duidelijke maatstaaf te trekken over de resolutie en grootte. Er is simpelweg teveel variatie.

Schermgrootte

De meest succesvolle smartphones van 2012 hebben een schermgrootte die varieert van 4 tot 5 inch (Shanklin, 2012). Echter zijn er nog oudere modellen waar rekening mee gehouden moet worden. De eerdere smartphones hebben een schermgrootte die rond de 3,5 inch reikt (Lutz, 2012). Een goed voorbeeld hiervan is de succesvolle iPhone 4.

Schermsresolutie

De hogere schermsresoluties van smartphones hebben een speciale eigenschap. Zij maken gebruik van de device pixel ratio. Zo heeft de iPhone 5 een schermsresolutie van 1136 x 640 pixels, maar wel een device pixel ratio van 2. Dit betekent dat CSS de **device-width** instelt op 320 pixels. Door dit terugschalen worden de meeste schermsresoluties herkend als 320 of 480 pixels breed (zie bijlage B), waardoor ze makkelijker te selecteren zijn met CSS3 media queries (Koch, 2010). In werkelijkheid is er een enorme variatie aan schermsresoluties, van een simpel 480 x 320 scherm tot een **Full HD** scherm op de allernieuwste smartphones (Goodwin, 2013).

3.2. Tablets

Net zoals bij de smartphones is er ook een oneindige selectie aan tablets te krijgen. Opmerkelijk bij de tablets is dat ze sinds kort onderverdeeld kunnen worden in drie categorieën: klein, medium en groot (Franklin, 2012). De grote categorie is op het moment nog erg leeg.

Schermgrootte

Door de scheiding van tablets in de drie categorieën wordt het een stuk makkelijker om hier schermgroottes aan te koppelen. De kleine tablets bevinden zich tussen de 7 en 7.9 inches, de medium tablets tussen de 8.9 en 10.1 inches en de grote tablets boven de 10.1 inches.

Schermeresolutie

Ook bij de tablets wordt gebruik gemaakt van de device pixel ratio. Zo heeft The New iPad een schermresolutie van 2048 x 1536 pixels, maar wordt de device-width uitgedrukt als 1024 x 768. Bij de kleine tablets speelt dit ook op, zo krijgen de meeste tablets een [viewport](#) van ongeveer 600 pixels, terwijl ze een schermresolutie van gemiddeld 1280 x 800 hebben (Wroblewski, Data Monday: Physical Device & Viewport Sizes, 2012). Het opvallende hierbij is dat de iPad mini geen device pixel ratio heeft, waarschijnlijk om fragmentatie in de markt te voorkomen (Wroblewski, Targeting devices that don't account for physical size in Viewport, 2012). Over het algemeen hebben de mini tablets een resolutie van ongeveer 1280 x 800, de medium en grotere tablets hebben meestal een Full HD resolutie, waarbij er een paar uitschieters naar boven zijn (Ødegård, 2012). Opnieuw dus erg veel variatie, waardoor er geen duidelijke lijn getrokken kan worden.

3.3. Desktop & Laptop

Gezien de aard van deze beide producten kunnen deze samen in een categorie terecht komen. De desktop en de laptop zijn de klassieke apparaten waarmee men het web bezoekt. Websites zijn dan ook gemaakt voor deze beide apparaten. Er zijn desktops met een scherm van 55 inch, maar ook laptops met een scherm van 13 inch. Responsive design heeft als kern om rekening te houden met smartphones en tablets. Websites werken al goed op desktops en laptops. Het enige verschil wat gemaakt kan worden is dat de website aangepast kan worden voor een scherm met een erg hoge resolutie. Door middel van het ontwerpen voor de medium en grotere tablets, pak je ook gelijk de laptops met een klein scherm en lage resolutie aan. De meest gebruikte schermresolutie ligt echter nog steeds erg laag, namelijk 1366 x 768. De meest gebruikte hogere resoluties zijn gelijk aan of liggen boven de 1600 pixels in breedte (1920 x 1080, 1600 x 900 en 1680 x 1050). De gemiddelde desktop en/of laptop ligt in breedte tussen de 1000 en 1400 pixels (zie bijlage B).

3.4. Gebruiksverschillen

3.4.1. Relatie met inhoud

Om te beginnen moet er een belangrijke mythe uit de weg geruimd worden. Vaak wordt er gesuggereerd dat gebruikers hun mobiel alleen voor kleinere zaken (bijvoorbeeld snel een e-mail lezen) gebruiken. 40% van de mensen geeft tenslotte aan zijn/haar mobiel op het toilet te gebruiken (Wroblewski, Josh Clark: Busting Mobile Myths, 2012). Het is echter een fabeltje dat mobiele gebruikers hun mobiel alleen voor kleinere zaken gebruiken. Ook de grotere onderhandelingen worden op de mobiel gedaan. Iemand heeft zelfs een keer een volledig vliegtuig gekocht via zijn smartphone (Wroblewski, Mobile First, 2011). Niemand wil naar een website gaan om er vervolgens achter te komen dat bepaalde functies niet beschikbaar zijn op jouw apparaat (Frost, 2012).

Bij de redesigns van de verschillende technologiewebsites (Mashable, The Next Web) is dit gelukkig ook terug te zien. Alle functies zijn zowel op mobiel als op desktop te verkrijgen. Meestal wordt dit opgelost door functies samen te voegen onder een instellingenknop. Bij het klikken op deze knop, wordt een accordeon met meer informatie zichtbaar. Daarnaast wordt er veel gebruik gemaakt van mouse-overs en een vaste menubalk bovenaan de pagina. Opmerkelijk is dat er hierdoor wel

gebruiksgemak op de desktop wordt ingeleverd. Vooral door de verschillende scrollbare gebieden en mouse-overs is de desktopversie minder natuurlijk dan verwacht (Fadeyev, 2012).

Dezelfde inhoud moet gebracht worden op verschillende apparaten. De meeste inhoud staat tegenwoordig vast. Het is gemaakt voor één website, voor één apparaat. In dit systeem staan de websites centraal. De ideale wereld heeft echter flexibele inhoud, welke overal te verkrijgen is. Hierbij staat de gebruiker centraal. De inhoud kan op verschillende media aangeboden worden. Dit wordt content shifting genoemd (Koczon, 2011).

Het design van een pagina moet gebaseerd zijn op de inhoud, niet andersom. Deze basis zorgt ook voor een andere toepassing van de media queries. Er kan niet uitgegaan worden van de pixels op de apparaten voor de verschillende 'standen' van een responsive design, zoals uit de apparaatanalyses geconcludeerd is. In plaats daarvan moet het design gebaseerd worden op de inhoud (content-out design). Tevens moet er niet gekeken worden naar 'breekpunten' (zie hoofdstuk 3.2.2.), maar naar 'optimalisatiepunten' (Callahan, 2013). Bij elk breekpunt wordt vaak een volledig nieuw design aangebracht, terwijl dit niet nodig is. Content-out design stelt ons in staat om aparte onderdelen van onze pagina aan te passen, aan de hand van de optimalisatiepunten (Boulton, 2013). Er hoeft geen hele nieuwe pagina aangemaakt te worden, er moet gezorgd worden dat elk element voor zichzelf bepaald wanneer het moet veranderen. Hierdoor wordt de oude manier van ontwikkelen (voor elk apparaat een aparte website) voorkomen.

Om dit te bereiken wordt er een nieuwe wending gegeven aan de toepassing van media queries. Zoals in het hoofdstuk *Technieken* is aangegeven, maakte Marcotte gebruik van pixels voor het aangeven van de max-width en min-width in zijn media queries. Een Pixel is geen relatieve eenheid, waardoor het geen goede maatstaaf is. In plaats van pixels, is het verstandiger om gebruik te maken van **em** (zoals Marcotte al wel doet bij het gebruik van lettertypes). De voornaamste reden hiervoor is omdat deze wel rekening houdt met inzoomen. De em is een relatieve eenheid en gaat uit van de standaardwaarde van elke browser. Voor de meeste browsers geldt dat 1em ongeveer 16px is. Bij het inzoomen van de pagina wordt deze waarde (en dus de tekstgrootte) veranderd. Indien er gebruik gemaakt wordt van een em-gebaseerde media query, past deze zich aan het inzoomen van de pagina aan (de waarde van de em is tenslotte veranderd door het inzoomen). Bij een pixelgebaseerde media query gebeurt dit echter niet, aangezien de waarde van de pixel niet veranderd is (Gardner L., 2012). Dit zorgt voor een pagina die zich niet aanpast aan de tekstgrootte en daardoor dus vervormd raakt.

Bij de em-eenheid moet wel altijd rekening gehouden worden dat hij zich aanpast aan de **font-size** van het **parent** (bovenliggende) element (Snook, 2011). Dit betekent dat **geneste** elementen een steeds grotere **font-size** krijgen. Denk hierbij aan list elementen met font-size 1.5em, met hieronder nieuwe list elementen. De tweede verdieping aan list elementen is dan 1.5 keer zo groot als de bovenliggende verdieping, aangezien em zich altijd aan het **parent** element aanpast. Een alternatief hierop is de **rem** eenheid, welke zich aanpast aan het **html** element. Er komt dan dus geen probleem voor met afwijkende font-sizes. Deze eenheid wordt echter niet door veel browsers ondersteund (Lazaris, 2012). Daarom wordt deze alleen gebruikt bij projecten van Los Banditos waarbij de doelgroep van de website uitsluitend gebruik maakt van "moderne" browsers (Internet Explorer 9 of hoger, Google Chrome, Mozilla Firefox of Apple Safari).

Stukken tekst met regels tussen de 45 en 75 karakters zijn voor een mens het fijnste om te lezen (Bringinghurst, 2004). Dit komt neer op zo'n 10 woorden per regel (Müller-Brockmann, 1996). Door middel van deze gegevens kan de minimale en maximale breedte van een (tekst)kolom berekend worden (oftewel een optimalisatiepunt). Belangrijk hierbij zijn het lettertype en de lettertypegrootte. Ook is de taal van de tekst een factor. Duitse woorden zijn over het algemeen bijvoorbeeld langer dan Engelse woorden. Een Engelse tekst met lettertype Georgia heeft bijvoorbeeld een tekstkolom met een maximale breedte van 22 em (352 pixels) als grootte, terwijl een Duitse tekst in Verdana een tekstkolom met een maximale breedte van 38.5 em (616 pixels) als grootte heeft (Gemert, 2013).

Dit is echter alleen voor tekstuele inhoud toepasbaar. Bij andere vormen van inhoud kan er vaak een andere waarde als minimale of maximale breedte nodig zijn. Het belangrijkste is dat het hoofdelement (met de belangrijkste inhoud) als basis wordt genomen voor de rest van de website. Op basis van het hoofdelement kunnen de andere elementen aangepast worden. Hierbij kan bijvoorbeeld gebruik gemaakt worden van de golden ratio voor de breedte van andere kolommen (Remick, 2008). Ook wil de breedte van de tekstkolom niet specifiek de breedte van het omhullende element aangeven. De kolom kan altijd verder opgevuld worden met witruimte. De ontwerper krijgt bij Los Banditos hier het laatste woord over. Hij kan zijn expertise laten gelden om de minimale en maximale breedte van een kolom te bepalen.

Het makkelijkste is om altijd te beginnen met de meest simpele lay-out (mobile first). Meestal is dit een lay-out waarbij alleen het hoofdelement bij de eerste oogopslag zichtbaar is. De andere elementen zijn in dit geval vaak onder het hoofdelement geplaatst of worden pas zichtbaar na het gebruik van een knop. Vanuit hier moet het scherm vergroot worden. Als het hoofdelement te breed wordt, kunnen de andere kolommen weergegeven worden naast het hoofdelement. Deze stap kan een aantal keer herhaald worden, totdat er geen extra informatie meer is om weer te geven. Hierbij hebben alle kolommen een minimale en maximale breedte, ze bepalen dus samen wanneer er geschoven gaat worden met de elementen, dit is niet puur afhankelijk van het hoofdelement. Vanaf hier kan gewerkt worden met witruimtes om de pagina verder te vullen. Doordat de inhoud het uiterlijk van de pagina bepaalt, is dit op elk apparaat ook goed zichtbaar. Er wordt tenslotte gewerkt met alleen maar relatieve waardes.

3.4.2. Relatie met apparaatverschillen

In zijn boek *Mobile First* geeft Luke Wroblewski duidelijkheid over vier essentiële onderdelen voor een mobiele strategie (Wroblewski, *Mobile First*, 2011). In het adviesrapport worden deze onderdelen meegenomen in de richtlijnen, omdat ze de ontwerper inzicht geven over de apparaatverschillen.

Organisatie

Denk vanuit de gebruiker, zodat je weet hoe de website gebruikt moet worden. Hier hangt nauw bij samen dat er eerst op de inhoud moet worden gericht (zoals hierboven besproken). De inhoud moet vindbaar zijn door middel van een relevante en goedgeplaatste navigatie. Hierbij wordt op mobiel vaak gebruik gemaakt van een knop welke lijdt naar het menu. Dit wordt gedaan om te voorkomen dat er teveel ruimte op het kleine scherm wordt ingenomen door de navigatie. Op desktop en tablet is er echter meer dan genoeg ruimte om het menu altijd zichtbaar te hebben bovenaan de pagina. Er moet wel altijd gezocht worden naar een zo minimalistisch mogelijke navigatie. Dit zorgt ervoor dat

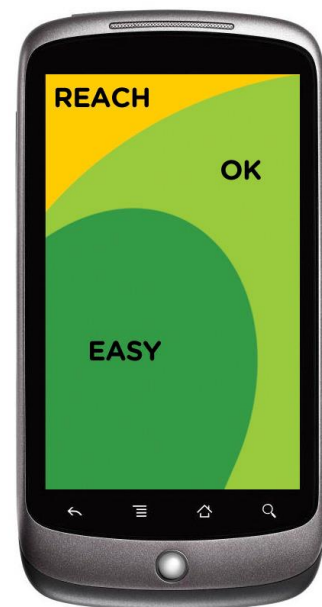
gebruikers zich altijd goed kunnen richten op hun behoeften. Tevens wordt de eenvoud van het ontwerp gewaardeerd.

Actie

De acties die uitgevoerd moeten worden op een mobiel verschillen van die van de desktop. Zo is er het grote verschil dat een smartphone en een tablet een touchscreen hebben. Hierdoor is er dus geen pixelperfecte selectie mogelijk, zoals wel mogelijk is met de muis op een desktop. Zorg er dus voor dat alle klikbare elementen groot genoeg zijn voor het aanraken met de vinger. Een goed voorbeeld wordt gegeven door Microsoft, welke zijn richtlijnen heeft ingesteld op basis van de gemiddelde vingerdikte van een mens (11 mm). Als minimale grootte van het aan te raken object raadt Microsoft 7 mm aan, met als adviesgrootte 9 mm. Daarnaast geeft ze een ruimte van 2 mm tussen de elementen aan (Microsoft, sd). Deze waarden gelden voor zowel smartphones als tablets.

Ook wordt een smartphone vaak met één hand beetgehouden, waardoor de plaatsing op het scherm essentieel is. Hiernaast is een voorbeeld te zien van de bereikbare plekken op een smartphone. Zo is voor een rechtshandig persoon de linkerbovenhoek van het scherm moeilijk te bereiken. Hier moeten dus niet veel klikbare objecten geplaatst worden.

Daarnaast zijn er ook de touch gestures die bekend zijn op de smartphones en tablets. Er kan onderscheid gemaakt worden tussen drie verschillende touch gestures: tap (het scherm eenmalig aanraken), drag (over het scherm heen bewegen zonder los te laten) & swipe (een vliegende beweging over het scherm heen maken). Deze touch gestures worden op meerdere elementen van een webpagina toegepast. Zo is er de bekende "pull down to refresh". Voor een volledige lijst, kan de touch gesture guide gebruikt worden (<http://static.lukew.com/TouchGestureGuide.pdf>).

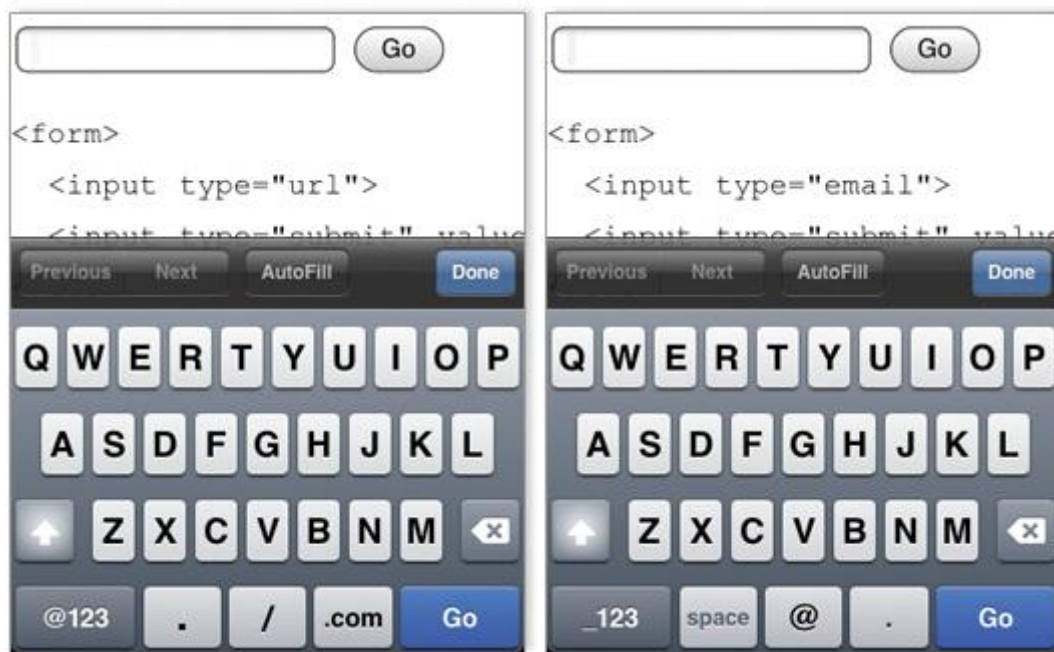


Figuur 3.1: Voorbeeld van de bereikbare plekken op een smartphone

Ook is de bekende CSS `:hover` techniek niet toepasbaar voor touchscreens. Deze techniek activeert wanneer een gebruiker boven een element "zweeft". Denk hierbij bijvoorbeeld aan een muis die op een element staat. Aangezien touchscreens niet beschikken over een aanwijzer met deze functionaliteit (je gebruikt je vingers), kan de `:hover` dus niet geactiveerd worden. Deze techniek moet dus zoveel mogelijk worden vermeden, zeker als het werkt in combinatie met een belangrijke functionaliteit van de website.

Invoer

Invoer is één van de belangrijkste onderdelen van webontwikkeling. Er is echter een groot verschil in de manier van invoeren tussen smartphones, tablets en desktops/laptops. Door het kleinere scherm van een smartphone kan er beter gekozen worden voor labels die in de invoervelden staan, zodat deze geen extra ruimte innemen. Daarnaast moet er veel gebruik gemaakt worden van de `type`, `attribute` en `mask` attributen, zodat er verschillende toetsenborden zichtbaar worden bij verschillende invoervelden.



Figuur 3.2: Het verschil in toetsenbord tussen een invoerveld met type url (links) en een invoerveld met type email (rechts)

Lay-out

De complexiteit van de lay-out van de website moet zo laag mogelijk liggen. Dit zorgt namelijk ook voor een makkelijkere implementatie van een responsive design. Ook moet er rekening mee gehouden worden dat het mobiele landschap continu verandert. Er kan dus geen vaste set aan regels gemaakt worden. Dit heeft veel relatie met het centreren van de website rondom de inhoud, in plaats van de implementatie van de apparaten. Dit betekent niet dat er compleet geen rekening gehouden moet worden met de apparaten. Ze hebben namelijk nog steeds hun eigen kenmerken, zoals bijvoorbeeld de pixeldichtheid voor afbeeldingen. Dit soort verschillen moeten altijd in de gaten worden gehouden en hier moet altijd op worden ingespeeld. Als beste tip geeft Luke Wroblewski mee dat je als webontwikkelaar moet accepteren dat de snelheid van verandering niet gaat veranderen. Wees dus altijd klaar voor nieuwe ontwikkelingen.

Slotbeschouwing

Door de enorme variatie in apparaten is er eigenlijk geen echte lijn meer te trekken. Er komen continu tussenmaten, wat het onmogelijk maakt om verschillende soorten apparaten bij elkaar onder te delen. Dit betekent ook dat er niet gekeken kan worden vanuit de apparaten, maar dat er een algemene oplossing nodig is. De apparaten geven geen garantie voor de toekomst en zijn geen maatstaaf waarop gebouwd kan worden. Er kunnen geen breekpunten onderscheiden worden op basis van “veelvoorkomende schermgroottes”, aangezien deze simpelweg niet bestaan.

De gebruiksverschillen tussen de apparaten worden veroorzaakt door het verschillende gebruik van inhoud en de algemene apparaatkenmerken. Om hier synergie in te verwerken, moet er altijd content-out worden ontworpen; dit betekent dat de inhoud altijd centraal staat. Hierbij moet er dus niet gezocht worden naar aparte breekpunten op basis van de apparaatkenmerken, maar eerder optimalisatiepunten op basis van de inhoud. De apparaatverschillen zelf komen neer op organisatie (van de website), actie, invoer en lay-out. De verschillen hierin kunnen overbrugd worden door het verantwoordelijk programmeren van de website.

In dit hoofdstuk zijn de onderstaande deelvragen behandeld:

- Welke browsers ondersteunen responsive design?
- Welke relatie heeft de inhoud van een website met responsive design?
- Wat zijn de apparaten waar rekening mee gehouden moet worden?
 - Welke apparaten ondersteunen responsive design?
 - Op wat voor resolutie draaien ze?
 - Wat voor schermgrootte hebben ze?
 - Hoe ziet het toekomstbeeld eruit?
- Wat is er bekend over de gebruiksmanier van elk van de apparaten?
 - Hoe wordt een website bekeken op de verschillende apparaten?
 - Wat betekent dit voor de lay-out van de website?

4. Relevante trends

Er moet altijd gekeken worden naar de andere trends in webontwikkeling die op het moment spelen. Zo kunnen trends elkaar versterken, of misschien juist een gevaar voor elkaar zijn. Ook bij responsive design is dit het geval. De toekomstige trends moeten duidelijk geanalyseerd worden om te kijken wat voor invloed ze hebben op responsive design.

4.1. Toekomstige trends

Aangezien dit een toekomstgerichte opdracht is, wordt er gekeken naar de verwachte trends. Op deze manier is het beter om in te spelen op de trends die er nog aan gaan komen.

- Typografie neemt de overhand (Bos, 2012);
- Geen [Flash](#) meer (Turner, 2012);
- Grote afbeeldingen voor een grote visuele impact (Turner, 2012);
- Meer witruimte (Turner, 2012);
- Meer delen (sharen) (de Bruin, 2012);
- Gebruik van één kleur (Ghosh, 2013);
- Mobiele applicaties & HTML5 (Gartner, 2012);
- King Content blijft eerste prioriteit (Casel, 2013).

Aangezien responsive design het onderwerp van dit rapport is, wordt deze trend achterwege gelaten in deze analyse. Deze wordt echter wel door de meeste bronnen (Bos, Casel, Ghosh en Turner) genoemd als één van de meestbelovende trends in webontwikkeling van de toekomst.

Typografie neemt de overhand

Door de introductie van Typekit is het legale gebruik van niet-standaard lettertypes enorm toegenomen (Bos, 2012). Het is door deze nieuwe technologie mogelijk om meer lettertypes toe te voegen aan een website, zonder dat het probleem optreedt dat deze niet werkt bij een gebruiker van de website. Er wordt ook verwacht dat er meer manieren naar voren komen om lettertypes op websites beschikbaar te stellen (Turner, 2012). Het meest aantrekkelijke aan deze trend is dat het makkelijker is om een website een eigen uiterlijk te geven. Elk denkbaar lettertype (zelf gemaakt of gekocht), kan nu gebruikt worden op alle websites, zonder verdere complicaties. Daarnaast zijn lettertypes op alle groottes altijd volledig scherp, in tegenstelling tot afbeeldingen. Dit geeft de ontwikkelaar dus meer mogelijkheden om de website een grafisch aantrekkelijk uiterlijk te geven, zonder beeldkwaliteit te verliezen.

Geen Flash meer

Waar deze trend de laatste jaren al redelijk op gang is gezet, gaat deze in de toekomst alleen maar door. Flash heeft altijd de mogelijkheid geboden voor het weergeven van animaties op websites. Echter heeft het veel nadelen met zich meegebracht. Het is slecht voor de SEO van een website, veel mobiele apparaten ondersteunen het niet meer en het is niet makkelijk te updaten. Er zijn steeds meer technieken die kunnen wat Flash kan, maar dan beter (Turner, 2012). Steeds meer bedrijven nemen afstand van Flash. Dit begon allemaal bij Apple (Schroeder, Steve Jobs: Flash Is No Longer

Necessary, 2010). Vervolgens hebben steeds meer bedrijven dit voorbeeld gevolgd, waardoor het nu niet meer beschikbaar is op de hedendaagse mobiele browsers (Schroeder, Adobe Won't Support Flash in Android 4.1, 2012).

Grote afbeeldingen voor een grote visuele impact

Er worden momenteel steeds meer websites gemaakt die een grote afbeelding op de achtergrond gebruiken om een grote visuele impact achter te laten op de eindgebruiker (Spooner, 2012). Vaak worden deze grote afbeeldingen alleen aangeboden bij gebruikers die de website bezoeken op een groot scherm (Turner, 2012). Het nadeel aan deze techniek is dat het niet werkt voor mobiele websites. De laadtijd van een grote afbeelding is immers veel te groot voor het mobiele internet.

Meer witruimte

In 2012 zijn er steeds meer websites uitgekomen welke gebruik maken van een grotere witruimte tussen de elementen. Witruimte op een pagina zorgt voor een makkelijker te begrijpen website, waar sneller op geïnterpreteerd kan worden. Het grootste probleem is dat het moeilijk is om klanten te overtuigen om iets te gebruiken wat je niet kan zien. Over het algemeen heerst de gedachte dat een website een container vol met inhoud moet zijn. Wat altijd over het hoofd wordt gezien is dat witruimte de gebruiker een bepaalde route af kan laten leggen over de website. Daarnaast zorgt het ervoor dat de gebruiker minder snel afgeleid wordt. Het grootste voordeel is misschien wel dat het ervoor kan zorgen dat bepaalde elementen echt uit de pagina springen (zoals bijvoorbeeld gesponsorde links), waardoor je de gebruiker meer kan beïnvloeden om op links, knoppen en banners te klikken (Boman, 2012).

Meer delen (sharen)

Social Media hebben in de laatste jaren een flinke opmars gemaakt. Facebook heeft in 2012 een verhoging van 25% van de actieve gebruikers meegemaakt, om het totaal op 1.1 biljoen maandelijks actieve gebruikers te zetten (Facebook, 2013). Twitter heeft zo'n 500 miljoen gebruikers, waarvan er 170 miljoen actief zijn (Lunden, 2012) en Google+ stormt al naar de 343 miljoen actieve gebruikers (Vaughan-Nichols, 2013). Het is dus simpel om te stellen dat social media (en daarmee het delen) alleen nog maar belangrijker wordt en vaker voor gaat komen. Er moet echter wel correct gebruik gemaakt worden van social media, anders kan het juist tegenstrijdig werken in een marketingstrategie (Besse, 2012).

Gebruik van één kleur

Er wordt steeds vaker gebruik gemaakt van één kleur (en de bijbehorende tinten) voor een website, in tegenstelling tot de toepassing van meerdere kleuren. Daarnaast wordt er dan gebruik gemaakt van wit en grijs tinten voor het aanvullen van de kleuren op de rest van de pagina. Door het gebruik van een simpele set aan kleurtonen en verschillende grijs tinten, kun je de gebruiker meer laten concentreren op de tekst en inhoud van de website (Ghosh, 2013).

Mobiele applicaties & HTML5

1.1 biljoen mensen wereldwijd hebben een smartphone, waarvan er 9 miljoen uit Nederland komen (Yarow, 2012). Dit gebruik van de smartphone zorgt voor een wens van de gebruiker om diensten te kunnen gebruiken op zijn/haar smartphone. Om deze reden worden mobiele applicaties alsmaar belangrijker om rekening mee te houden (wat zowel een nadeel als een voordeel is). Gartner (2012) herkent een verschuiving van *native* apps naar web apps. Dit komt door de opkomst van de programmeertaal HTML5. HTML5 wordt steeds beter en krijgt meer functies, waardoor web apps alsmaar aantrekkelijker worden om te maken. Wel stelt Gartner dat native apps altijd de beste gebruikerservaring bieden.

King Content blijft eerste prioriteit

De term “King Content” (de inhoud altijd als eerste prioriteit hebben) is altijd erg belangrijk geweest voor webontwikkelaars. Dit werd nogmaals benadrukt door Google toen ze hun zoekmachine hadden geoptimaliseerd om zich meer te richten op de inhoud van een pagina (Casel, 2013). Mede hierdoor is deze trend weer helemaal terug van weggeweest. Deze trend ligt tevens ook dichtbij de aanpak om content-out te ontwerpen.

4.2. Relatie met responsive design

De trends hebben bijna allemaal een relatie met de implementatie van responsive design. De volgende richtlijnen heb ik voor Los Banditos op kunnen stellen op basis van de trends en de eerder vastgestelde definitie van responsive design. Deze zijn ook verwerkt in het adviesrapport.

- Het gebruik van veel typografie (in plaats van afbeeldingen) is een *must*. De combinatie van typografie met responsive design is erg gunstig. Het werkt voor alle gebruikers en is tevens schaalbaar. Daarnaast kan het ook de laadtijd van een pagina verbeteren. Afbeeldingen eisen simpelweg te veel van een website.
- Het gebruik van Flash is een *no go*. Het is niet schaalbaar en er zijn nieuwere technieken die wel goed samen werken met responsive design.
- Het gebruik van grote afbeeldingen wordt grotendeels afgekeurd. Hoewel dit wel toepasbaar is voor de grotere versie van een website, is dit in principe niet optimaal als de website geschaald gaat worden. Het mobiele internet is hier nog niet klaar voor.
- Het gebruik van witruimte is een goede techniek. Waar het zowel de eerder besproken voordelen met zich meebrengt, is het ook erg goed toepasbaar voor het maken van meerdere lay-outs voor een website. De toepassing van witruimte kan ervoor zorgen dat een pagina goed zichtbaar wordt bij een specifieke paginabreedte.
- Social Media zijn een goede toevoeging op elke website, maar laten hun kracht pas echt zien op responsive websites. Het delen van content is een veel voorkomende bezigheid op de mobiele apparaten, waardoor het dus perfect aansluit op een responsive design.
- King Content binnen responsive design is heel belangrijk. Door middel van responsive design moet er bepaald worden welke content de hoogste prioriteit heeft. Deze twee trends liggen erg dicht bij elkaar, aangezien de één niet kan werken zonder de ander. King Content is pas “king” als het op een goede manier geïmplementeerd wordt.

Slotbeschouwing

De volgende trends worden herkend in de huidige webwereld:

- Typografie neemt de overhand;
- Geen **Flash** meer;
- Grote afbeeldingen voor een grote visuele impact;
- Meer witruimte;
- Meer delen (sharen);
- Gebruik van één kleur;
- Mobiele applicaties & HTML5;
- King Content blijft eerste prioriteit.

Deze trends kunnen ieder op hun eigen manier responsive design versterken. Door op deze trends in te spelen, kunnen de responsive designs van Los Banditos versterkt worden. In dit hoofdstuk zijn de volgende deelvragen behandeld:

- Met welke trends op het gebied van webontwikkeling moet rekening gehouden worden bij het ontwikkelen van responsive design?
 - Welke trends zijn er op het gebied van webontwikkeling?
 - Welke trends kunnen een impact hebben op responsive design?
 - Op welke manier kunnen deze trends in het voor- en nadeel werken van responsive design?

5. Werkwijze

Los Banditos werkt op het moment volgens hun eigen werkwijze. Om responsive design succesvol te implementeren, wordt deze werkwijze eerst in kaart gebracht. Dit is gebeurd door het opdoen van praktijkervaring. Dit is de beste manier om de werkwijze te leren kennen, omdat ik er zo direct mee in aanraking kwam. Daarnaast wordt er gekeken naar de toekomstvisie van Los Banditos. Dit is gedaan door middel van een interview met bedrijfsleider Rogier van Roon. De keuze voor het interview is gemaakt omdat Rogier de enige is die beslist over de toekomst van Los Banditos, hij is tenslotte de bedrijfsleider. De toekomstvisie speelt een grote rol in het ontwikkelen van de nieuwe werkwijze. Samen met de informatie uit de voorgaande hoofdstukken brengt dit een nieuwe werkwijze teweeg, welke vervolgens besproken wordt.

5.1. Huidige werkwijze

Los Banditos bestaat uit bedrijfsleider Rogier van Roon, aangevuld met stagiairs en freelancers. Er worden drie verschillende producten geleverd: websites, webshops en maatwerk. De core business van Los Banditos is het leveren van websites. Deze worden gemaakt met het gebruik van de content management systemen (CMS) Joomla! en WordPress. Voor de webshops wordt de techniek vaak uitbesteed, het grafische ontwerp wordt echter wel zelf gedaan. Het maatwerk is voor de complexere projecten, welke een volledig eigen systeem willen hebben. Dit wordt ook volledig binnenshuis ontwikkeld. Zo'n 80% van de opdrachten wordt momenteel 'via via' verkregen.

De werkwijze van Los Banditos is grotendeels gebaseerd op de voorloper van MijnWerkers: Studio. Deze projectmanagementtool zorgde voor de structuur in de projecten van Los Banditos. Hierbij is een simpel waterfall-model te herkennen dat bij elk project opnieuw herhaald wordt.



Figuur 5.1: Procesverloop dat bij elk project van Los Banditos voorkomt

De taken die in dit bovenstaande proces worden aangegeven zijn meestal globale taken. Voorbeelden hiervan zijn grafische vormgeving, templating of techniek. Het gevolg hiervan is dat de medewerker erg vrij wordt gelaten in zijn invulling hiervan, wat Rogier ook graag wil. Meestal krijgt een medewerker van tevoren een aanwijzing van Rogier over het verwachte resultaat. Hierbij komt het regelmatig voor dat één medewerker een eigen project heeft, wat beslist wordt op basis van de kennis van de werknemer. De volgorde van de taken staat nooit vast en kan per project variëren. Dit hangt ook nauw samen met Rogier's visie om Los Banditos te centreren rondom het gebruik van stagiairs. Hij probeert altijd om twee stagiairs met een grafische achtergrond en één met een technische achtergrond te hebben. Door zijn ruime ervaring met het gebruik van stagiairs is Rogier tot de conclusie gekomen dat zij het meeste leren door vrijheid te hebben. Hiermee probeert hij ze ook te motiveren om hun eigen wending aan een project te geven. Rogier probeert zich dan ook te richten op het gebruik van stagiairs met een specialisatie, in plaats van 'allrounders'.

De afsluiting en oplevering van een project gebeurt door middel van een optionele training voor de klant in het gebruik van het gebruikte CMS. Hierbij gaat Rogier bij de klant langs om het systeem uit te leggen en aanwijzingen te geven over de manier van het onderhouden van het product. Dit is echter een optionele dienst, die niet door elke klant verlangd wordt. Het gevolg hiervan is dat niet alle klanten deze dienst nemen, maar vervolgens wel vaak via de telefoon of e-mail contact opnemen met Rogier voor vragen. Dit zorgt regelmatig voor tijdsproblemen aan Rogiers kant, waardoor hij zich niet makkelijk aan de planning kan houden.

5.2. Toekomstvisie

Rogier geeft aan dat de huidige werkwijze structuur tekort komt. Er is geen duidelijke planning voor projecten. Zo wordt er soms vroeg in het project inhoud aangeleverd, waar dit bij andere projecten pas veel later gebeurt.

De kern van Rogiers toekomstvisie is specialisme. Rogier ziet de markt verschuiven en wil hier op in gaan spelen. Zo is hij er van overtuigd dat iedereen binnenkort zijn eigen website kan maken door de grote toename in open source projecten. De webontwikkelaar moet zich dan onderscheiden door specialisme te tonen in het vak. Een voorbeeld hiervan is dat Rogier ervoor wil zorgen dat zijn stagiairs alleen nog maar taken binnen hun specialisme uitvoeren. Zo krijgt elk project een aparte ontwikkelaar en een aparte vormgever, in plaats van één volledig project aan een medewerker te koppelen.

Ook wil Rogier meer van het werk gaan verplaatsen naar zijn medewerkers, zodat hij zelf aan de slag kan met meer innovatieve projecten. Hiermee probeert hij de standaard dienst te blijven leveren (websites), terwijl hij ook op de toekomst in probeert te spelen. Dit gaat echter pas gebeuren als MijnWerkers volledig klaar is en voor regelmatige inkomsten zorgt. Hierbij ziet hij een ideale situatie voor zich met drie vaste medewerkers (een ontwikkelaar, een vormgever en een commercieel medewerker), aangevuld met drie stagiairs. Het verkrijgen van opdrachten wil Rogier ook gaan doen via AdWords, zodra de nieuwe Los Banditos website gelanceerd wordt.

Een laatste punt van verandering is de overschakeling van Studio naar MijnWerkers. In principe heeft dit niet heel veel met de werkwijze te maken. MijnWerkers is een verbeterde versie van Studio op het gebied van techniek en ontwerp. Het houdt daarom ook gewoon dezelfde stappen aan als Studio (zoals te zien is in figuur 6.1).

5.3. Nieuwe werkwijze

De twee kenmerken die in de nieuwe werkwijze terug moeten komen, zijn specialisme en structuur. Bij de scrumwerkwijze staan deze twee kenmerken centraal (Guldemon, 2011). De huidige waterfallmethode zorgt voor weinig structuur. Deze methode is niet ver genoeg uitgewerkt om ook tijdens het ontwikkelen structuur te bieden. Er wordt 'maar wat gedaan' tijdens het ontwikkelingsproces. Scrum kan hier de oplossing bieden. Bij scrum krijgt iedereen zijn eigen functie (specialiteit) en wordt er via een continu herhalend proces gewerkt (structuur) (Schwaber & Beedle, 2001).

De keuze voor scrum is gebaseerd op de kennis die is opgedaan uit het werken met verschillende werkwijzen tijdens de bachelor Communicatiesystemen. Zo heb ik daar voornamelijk gewerkt met

Prince2 en scrum. Uit deze twee werkwijzen vind ik scrum de beste oplossing voor dit probleem. Scrum biedt structuur en is redelijk makkelijk om mee te beginnen. Er is geen verdere uitgebreide analyse gedaan naar verschillende werkwijzen, omdat er intern is besloten dat het beter is om meer tijd in responsive design te steken dan in een nieuwe werkwijze. Het adviseren omtrent responsive design is namelijk waar mijn echte specialisatie ligt. Het achterwege laten van deze analyse werkte tevens tijdsbesparend, waardoor er meer tijd over was voor de case studies.

Scrum is een Agile softwareontwikkelmethode. Dit betekent dat er geconcentreerd wordt op het ontwikkelen in korte, overzichtelijke perioden, waarbij meerdere mensen tegelijkertijd werken aan één project. Bij scrum gebeurt dit door het project op te delen in 'sprints'. Een sprint is een vaste, herhalende periode (of iteratie) van meestal één tot vier weken, waarvan de lengte en inhoud van tevoren zijn vastgesteld (Schwaber & Beedle, 2001). Gedurende elke sprint levert het scrumteam een functionaliteit van het eindproduct op.

Elk project heeft zijn eigen scrumteam. Dit team bestaat uit de producteigenaar, de scrummaster en het ontwikkelingsteam (VersionOne, 2013).

5.3.1. Product backlog

De producteigenaar (de klant) is verantwoordelijk voor het samenstellen van de *product backlog*. De product backlog is een geordende lijst met eisen voor een product. Deze lijst bestaat uit functionaliteiten, op te lossen systeemfouten en niet-functionele eisen. Al deze eisen worden door de producteigenaar geordend op basis van prioriteit. De prioriteit wordt bepaald door het toekennen van de geschatte meerwaarde en het bepalen van de ontwikkelingstijd. De geschatte meerwaarde wordt bepaald door de producteigenaar. Dit gebeurt doorgaans door het toekennen van een aantal punten. De ontwikkelingstijd wordt door het ontwikkelingsteam aangegeven. Dit gebeurt tevens door het toekennen van punten, maar het is ook mogelijk om dit in uren uit te drukken. De eisen worden in verhaalformaat weergegeven (Szalvay, 2009). De structuur die hiervoor gebruikt wordt is: "Als een [gebruikertype] wil ik [een bepaalde actie] zodat ik [het gewenste resultaat]". In deze structuur zijn de variabelen tussen de vierkante haakjes geplaatst. Een voorbeeld voor een eis genoteerd in verhaalformaat is: "Als een medewerker van Los Banditos wil ik een takenlijst hebben in MijnWerkers, zodat ik snel kan zien welke taken er volbracht moeten worden voor een project". De scrummaster kan helpen bij het opstellen van deze eisen.

5.3.2. Sprint backlog

Volgens Schwaber & Beedle (2001) is het ontwikkelingsteam verantwoordelijk voor de op te leveren onderdelen van het eindproduct tijdens elke sprint. Gezien de situatie bij Los Banditos, bestaat het team hier doorgaans uit minimaal één ontwerper en één ontwikkelaar. Samen stellen zij de *sprint backlog* samen. De sprint backlog is een lijst met eisen die afgewerkt moet worden tijdens de eerstvolgende sprint. Deze eisen worden geselecteerd uit de product backlog. Aan het begin van elke sprint wordt bepaald hoe lang de sprint gaat duren (dit varieert tussen minimaal een week en maximaal een maand). Het ontwikkelingsteam voegt vervolgens eisen uit de product backlog toe aan de sprint backlog, totdat er genoeg eisen zijn geselecteerd om de sprint backlog volledig te vullen. De toegekende waarde voor de ontwikkelingstijd wordt hierbij als leidraad genomen. Elke eis wordt vervolgens opgedeeld in taken. Deze taken worden bij Los Banditos ingevoerd in MijnWerkers. Dit is

het centrale punt voor het ontwikkelingsteam om hun taken op te slaan. Voorheen werd een stagiair ingezet voor meerdere specialiteiten (ontwerp, ontwikkeling, etc) binnen een project. De stagiairs krijgen nu een eigen specialiteit toegekend binnen een project.

5.3.3. Scrummaster

De scrummaster is verantwoordelijk voor het begeleiden van het scrumproces (Jongerius, 2010). Dit betekent dat de scrummaster verantwoordelijk is voor het verloop van elk project. Daarnaast is de scrummaster verantwoordelijk voor het begeleiden van de verschillende bijeenkomsten die plaats vinden tijdens het scrumproces. Dit wordt de functie van Rogier. In deze functie kan Rogier alle projecten in de gaten houden zonder zelf te hoeven ontwikkelen. Het begeleiden van projecten neemt minder tijd in dan het begeleiden én het ontwikkelen van projecten. Dit ligt in lijn met Rogiers wens om het werk meer van zich af te schuiven, zodat hij zelf meer tijd overhoudt voor innovatieve projecten.

5.3.4. Scrumproces

Tijdens het scrumproces vinden er verschillende bijeenkomsten plaats: sprintplanning, de dagelijkse scrum, *scrum of scrums*, *backlog grooming*, projectreview en sprintreview. Schwaber & Beedle (2001) omschrijven deze bijeenkomsten als volgt:

Sprintplanning

Tijdens de sprintplanning wordt de eerstkomende sprint samengesteld. Dit proces duurt maximaal acht uur. Tijdens de eerste helft van dit proces wordt de product backlog geprioritiseerd en de duur van de sprint bepaald. Dit gebeurt door het gehele scrumteam. Het tweede deel wordt door het ontwikkelingsteam gedaan, dat de sprint backlog samenstelt en indeelt.

Dagelijkse scrum

Aan het begin van elke werkdag komen de scrummaster en het ontwikkelingsteam samen voor een korte bespreking van maximaal 15 minuten. In deze tijd geeft elk aanwezig lid antwoord op de volgende drie vragen:

- Wat heb je gedaan?
- Wat ga je doen?
- Ben je nog problemen tegengekomen?

Door deze sessie blijft iedereen op de hoogte van de voortgang binnen het project. Eventuele problemen worden genoteerd door de scrummaster en worden buiten deze sessie om behandeld.

Scrum of scrums

Na de dagelijkse scrum is er ruimte voor de scrum of scrums. Hier worden taken besproken waarbij er een overlap aanwezig is tussen de verschillende specialiteiten.

Backlog grooming

Backlog grooming vindt plaats tijdens de sprints. Deze bijeenkomst duurt maximaal een uur. Tijdens deze bijeenkomst wordt de product backlog gevormd. Hierbij zetten de producteigenaar en het ontwikkelingsteam de eisen om naar verhaalvorm. Ook worden ze gesorteerd op prioriteit. Backlog grooming kan meerdere malen per sprint plaatsvinden.

Projectreview en scrumreview

Deze bijeenkomsten vinden beide aan het einde van een sprint plaats. De projectreview bestaat uit het beoordelen van het gemaakte werk en duurt maximaal 4 uur. De scrumreview bestaat uit het beoordelen van de sprint en duurt maximaal 3 uur. Bij beiden is het de bedoeling om de focus te leggen op verbetering. Wat ging er fout en hoe kan dit verbeterd worden?

5.3.5. Frameworks

Aangezien er vaak met stagiairs gewerkt wordt, is er gekeken naar een oplossing om deze makkelijk kennis te laten maken met het ontwikkelen van responsive websites. Een vereiste voor de oplossing van dit probleem is dat de stagiairs een praktisch voorbeeld hebben van het gebruik van de technieken van responsive design. Na een intern overleg, heb ik twee verschillende frameworks aangemaakt: een framework voor Joomla! (<http://demo.losbanditos.nl/joomla>) en een framework voor WordPress (<http://demo.losbanditos.nl/wordpress>). Beide frameworks zijn een schil waarmee alle projecten van Los Banditos gemaakt kunnen worden. De toepasbare omgeving (Joomla! of WordPress) wordt bij elk nieuw project gekopieerd, zodat de basis voor een responsive design al aanwezig is (wat tevens de ontwikkeltijd verlaagt). Tevens dragen deze frameworks weer bij aan de structuur die momenteel ontbreekt binnen de projecten.

Open source frameworks, zoals Twitter Bootstrap en Foundation, zijn bewust niet gekozen om te gebruiken. Deze frameworks zijn gemaakt voor een uitgebreid scala aan use cases, die nooit allemaal door Los Banditos gebruikt zullen worden. Deze overbodige use cases genereren onnodige code en verhogen daarmee de laadtijd van de website. Door het gebruik van de op maat gemaakte frameworks, wordt er veel beter omgegaan met de laadtijd van de pagina.



Figuur 5.2: Voorbeeld van het Joomla! framework, hier is duidelijk de rijen- en kolommenverdeling te zien

De frameworks geven een goed voorbeeld van het gebruik van content-out design en mobile first ontwikkeling. Beide frameworks houden dan ook geen rekening met de verschillende apparaten die op de markt te krijgen zijn, maar letten puur op de inhoud van de website.

Beide frameworks zijn gebaseerd op het gebruik van kolommen in rijen die over de volle breedte van het scherm spannen. Zo kan de ontwikkelaar zelf kiezen hoeveel rijen aan inhoud hij wil hebben en hoe de kolommenindeling binnen deze rijen eruit ziet.

De tekst die in de kolommen staat is voorbeeldinhoud. Hier kunnen ook tabellen, afbeeldingen of andere inhoud in terecht komen. Deze kolommen worden ieder op hun eigen optimalisatiepunt aangepakt, zodat de website goed responsive werkt. Dit werkt voor de stagiair als voorbeeld voor het toepassen van responsive design op een (complexe) inhoudsgevulde website.

5.4. Voor- en nadelenvergelijk

Om een duidelijk verschil tussen de oude en de nieuwe werkwijze te laten zien, zijn de voor- en nadelen van beide werkwijzen opgesteld. Allereerst de voor- en nadelen van de oude werkwijze.

Voordelen	Nadelen
Vrijheid voor de werknemer om zijn eigen draai aan een project te geven	Rogier moet teveel tijd steken in elk project en komt niet aan zijn eigen dingen toe
Werknemer heeft veel variatie in z'n werk	Ontbreken van specialisme
	Ontbreken van structuur

Tabel 5.1: Voor- en nadelen van de oude werkwijze

De huidige werkwijze heeft vooral voor de werknemer veel voordelen. Zo heeft deze veel variatie in zijn projecten, aangezien hij vaak alle specialisaties in een project voor zijn rekening neemt. Daarnaast kan hij makkelijk zijn eigen draai aan een project geven. Dit heeft echter tot gevolg dat er structuur en specialisme ontbreken in de projecten. Aangezien er grotendeels met stagiairs gewerkt wordt, moet veel van het werk nog gecontroleerd worden door Rogier. De nieuwe werkwijze probeert deze nadelen aan te pakken.

Voordelen	Nadelen
Mogelijkheid voor Rogier om afstand te nemen van het ontwikkelen	Het vele samenwerken kan tot frustratie leiden
Elke werknemer is bezig met zijn specialisme	Projecten duren langer
Elke website is responsive	Rekening houden met meer browsers (mobiele browsers komen erbij)
Structuur in elk project	

Tabel 5.2: Voor- en nadelen van de nieuwe werkwijze

Het invoeren van de nieuwe werkwijze lost de problemen van de oude werkwijze op: er komt structuur en specialisme in de werkwijze en Rogier krijgt meer tijd om handen (alledrie als gevolg van de implementatie van scrum). Dit brengt echter wel een nieuw probleem met zich mee. Als mensen niet goed samen kunnen werken, kunnen er onderlinge frustraties ontstaan. De invoering van responsive design brengt ook nog een aantal nadelen met zich mee. Zo duren projecten langer, aangezien responsive design extra werk met zich meebrengt. Daarnaast komen er meer browsers die ondersteund moeten worden (zoals alle mobiele browsers).

Slotbeschouwing

In de huidige werkwijze van Los Banditos ontbreekt structuur en specialisme. Rogier wil deze twee kernbegrippen in zijn werkwijze verwerken. Ook kost de huidige werkwijze Rogier zelf teveel tijd. Om dit te bereiken wordt er een scrumwerkwijze geïmplementeerd, naast het responsive maken van alle websites. Om de opstap naar een responsive website makkelijker te maken voor de stagiairs, zijn er twee frameworks ontwikkeld. Tevens versnelt dit het ontwikkelproces van een responsive website. In dit hoofdstuk zijn de volgende deelvragen behandeld:

- Hoe kan responsive design verwerkt worden in een werkwijze die aansluit op de visie van Los Banditos?
 - Wat is de huidige manier van werken?
 - Welke stappen moeten er genomen worden om over te gaan naar een werkwijze met responsive design?
- Hoe verhoudt een werkwijze met responsive design zich ten opzichte van de huidige manier van werken?
 - Wat zijn de voor- en nadelen van responsive design?
 - Wat zijn de voor- en nadelen van de huidige manier van werken?
 - Hoe verhouden deze zich ten opzichte van elkaar?
- Hoe ziet een werkwijze binnen Los Banditos eruit waarbij gebruik gemaakt wordt van responsive design?

6. Case studies

Responsive design is een praktisch onderwerp. Daarom zijn er verschillende case studies verricht om de theorie uit het literatuuronderzoek op de proef te stellen. Deze case studies worden opgedeeld in twee grote onderdelen: het testen van de frameworks en het responsive maken van de projectmanagementtool MijnWerkers. De Frameworks zijn getest om te kijken of deze bruikbaar zijn voor de (toekomstige) stagiairs. MijnWerkers is een project met een grote omvang. Deze bevat veel componenten die een creatieve aanpak vereisen. Dit is daarom een goed voorbeeld om te laten zien wat er allemaal bij responsive design komt kijken.

De scrumwerkwijze is niet gebruikt bij de case studies, aangezien deze pas toegepast kan worden als MijnWerkers klaar is. Deze werkwijze is namelijk gebaseerd op de toekomstvisie van Rogier, die nog niet gerealiseerd kan worden zolang MijnWerkers nog veel tijd inneemt. Vooralsnog heeft elke werknemer dus nog zijn eigen project. De scrumwerkwijze is wel verwerkt in het adviesrapport. Na de case studies zijn de hieruit voortgekomen leerpunten opgesomd en verder verwerkt in het adviesrapport.

6.1. Frameworks

Het Joomla! framework is, op het moment van schrijven, toegepast op de websites voor twee klanten van Los Banditos: Masters of Lxry en Weijenbarg. Beide websites zijn ontwikkeld door stagiairs van Los Banditos die nog geen ervaring hadden met het framework en met het ontwikkelen van een responsive website. Dit is opzettelijk gedaan, zodat de frameworks getest werden in soortgelijke omstandigheden zoals ze ook gebruikt gaan worden (door stagiairs zonder ervaring in responsive websites). Het testen van de frameworks ging van start op het moment dat de stagiair het gemaakte ontwerp overzette naar code.

6.1.1. Uitleg

Het grootste struikelblok is waarschijnlijk de uitleg voorafgaand aan het gebruik van het framework. Bij beide projecten werd dit aangepakt door de stagiairs het adviesrapport te laten lezen. Hierin worden de 'ins and outs' van responsive design besproken. Dit zou de stagiair dus ook een goed stuk op weg moeten helpen. Echter werd er niet goed begrepen dat er met een mobile first aanpak gewerkt moest worden. Hierdoor werden alle styles buiten de media queries geplaatst, waardoor de mobiele weergaves een overbodige hoeveelheid aan styles binnenkregen. Dit heeft er ook mee te maken dat het adviesrapport een aantal weken voor het begin van het project gelezen werd, in plaats van vlak ervoor. Deze punten moeten dus iets duidelijker overgebracht worden. Mobile first is geen optie, maar een vereiste.

6.1.2. Fouten in de code

Zoals bij elk systeem dat gebaseerd is op code, zijn er bij de eerste tests altijd wel een paar fouten te vinden. Zo hield de module voor het menu van het Joomla! framework geen rekening met eventuele submenu's. Hierdoor vond er een vervorming van het menu plaats bij de responsive versie van de

website voor Weijenbarg. Deze module is aangepast zodat hij wel goed omgaat met een eventueel submenu.

Voor de kolommenverdeling werd er gebruik gemaakt van procenten. Zo is er de mogelijkheid voor een kolommenindeling van drie kolommen van één-derde breedte, maar ook voor een kolommenindeling met één kolom van twee-derde en één kolom van één-derde. Hierbij kwam naar voren dat de verticale uitlijning van deze twee rijen niet goed uitkwam, doordat de procentuele waarde van de kolommen niet goed verdeeld was.

Joomla! maakt gebruik van modules en componenten. Er heerste wat onduidelijkheid over het verschil in toepassing tussen deze onderdelen. Dit had puur met voorlichting over het framework te maken. Een korte uitleg gaf de stagiair gelijk genoeg duidelijkheid. Dit probleem moet dus aangepakt worden door vooraf uit te leggen hoe het framework hiermee omgaat.

6.2. MijnWerkers

MijnWerkers heeft een aparte aanpak nodig, die niet volledig volgens het adviesrapport kan gaan. Dit komt omdat MijnWerkers voor de desktop al volledig klaar is. Er kan dus geen mobile first aanpak gehanteerd worden. MijnWerkers moet dus letterlijk responsive gemaakt worden, in plaats van responsive opgebouwd worden. Hiervoor worden echter nog steeds dezelfde technieken gebruikt, alleen de aanpak is anders. Er wordt ook nog steeds gebruik gemaakt van de mobile first denkwijze. Er wordt continu gekeken of bepaalde functies wel echt nodig zijn. *Less is more* staat centraal bij het responsive maken van MijnWerkers. De optimalisatiepunten zijn per onderdeel bepaald. Er is dus geen algemene set van media queries opgesteld om aan bepaalde apparaten te voldoen. MijnWerkers houdt zich daarmee aan het principe van content-out design.

Acties	Factuur	Nr.	Betalingstermijn	Status	Geen filter	Actief	Deelfactuur	Bedrag
	RET	130003	<div></div>	Openstaand	RET VOF	Ja	1 / 100%	€ 147,62
	Zomaar een maandelijks factuurtje	130006	<div></div>	Openstaand	RET VOF	Ja	2 / 100%	€ 363,00
Openstaand:								€ 422,00
21% BTW								€ 88,62
Totaal								€ 510,62

© 2013 MijnWerkers (Versie 1.0)

Figuur 6.1: Factuurpagina van MijnWerkers

Het responsive maken van MijnWerkers moest volgens een gestructureerde aanpak gaan. MijnWerkers bestaat uit 71 pagina's, waarvan sommige pagina's met richting de 10.000 regels aan code. Veel van de pagina's bestaan echter uit soortgelijke onderdelen, hierdoor moest er dus een duidelijke analyse komen van de responsive oplossing voor dit soort onderdelen. MijnWerkers bevat vier onderdelen die door het hele systeem hergebruikt worden. Deze onderdelen zijn tevens gesorteerd op prioriteit. De prioriteit geeft in dit geval aan welk onderdeel het belangrijkste is voor het goed functioneren van MijnWerkers. De prioriteit loopt van hoog naar laag.

1. Tabellen
2. Een statisch linkermenu
3. Formulieren
4. Een statische rechterkolom

Door het analyseren konden de probleempunten van deze onderdelen op een rij gezet worden. Vervolgens zouden de oplossingen hiervoor één voor één doorgevoerd worden. Na het analyseren van MijnWerkers kwamen er een aantal probleempunten naar voren:

- De tabellen nemen heel veel ruimte in, deze zijn alleen goed zichtbaar bij een breedte van 1200 pixels of hoger.
- Doordat het menu vaststaat aan de linkerkant van het scherm (met een breedte van 250 pixels), is er geen ruimte voor de overige inhoud van de pagina bij een lage resolutie.
- De formulieren maken gebruik van tabbladen, die niet meer netjes naast elkaar kunnen staan bij een lage resolutie. Dit zorgt voor een probleem in gebruiksgemak.
- De veelgebruikte statische rechterkolom brengt hetzelfde probleem met zich mee als het menu. Er blijft geen ruimte meer over voor de overige inhoud van de pagina bij een lage resolutie.
- Veel knoppen bevatten zowel een icoon als een tekst, waardoor een rij knoppen te veel ruimte inneemt, vooral als deze knoppen rechts uitgelijnd worden in combinatie met een links uitgelijnde titel.

6.2.1. Tabellen

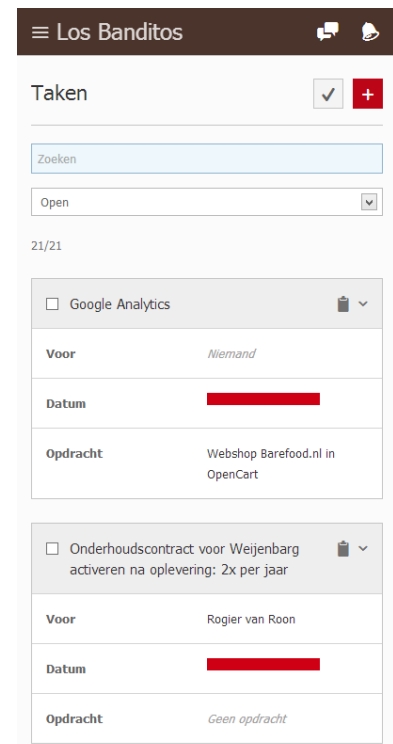
De tabellen zijn de grootste uitdaging. Zoals in het adviesrapport staat vermeld, zijn er niet veel oplossingen voor responsive tabellen. De tabellen van MijnWerkers zijn daarnaast ook erg data-intensief. Sommige tabellen zijn zelfs zo'n 10 kolommen breed. Als eerste is er mobile first gedacht om dit op te lossen. Er is gekeken welke kolommen echt nodig zijn in de tabellen en welke kolommen eigenlijk opvulling waren. Zo zijn de tabellen flink in breedte teruggebracht. Waar veel pagina's met tabellen voorheen een minimale breedte hadden van 1200 pixels, is dit nu teruggebracht naar ongeveer 1000 pixels. Door simpele prioriteiten te stellen zijn er zo'n 200 pixels uitgespaard.

De volgende keuze lag tussen twee verschillende componenten. De tabellen pasten niet meer in het inhoudsvlak, aangezien het menu links ook ruimte innam. Er moest een keuze op basis van prioriteit gemaakt worden. Voor MijnWerkers is besloten dat de tabellen een hogere prioriteit hebben dan het menu. De tabellen zorgen voor veel gebruiksgemak die Los Banditos tot de meest minimale breedte wil aanbieden. Met deze achterliggende gedachte is er gekozen om het menu in te klappen (dit staat

in details uitgelegd bij het kopje menu). Dit zorgt ervoor dat de tabel 250 pixels extra ruimte krijgt. Hierdoor zijn de tabellen tot een minimale breedte van 750 pixels zichtbaar.

Bij de breedte van 750 pixels moet er toch echt een verandering optreden. De tekst wordt slecht leesbaar in de kolommen, waardoor het grote pluspunt van de tabellen (het overzicht) verloren gaat. Op dit punt is er besloten om de rijen los te koppelen en elke rij een eigen blok te geven (zie figuur 7.2). De informatie van elke rij is nog duidelijk zichtbaar. Echter is het vergelijken van rijen (het grootste voordeel van de tabellen) niet meer makkelijk te doen. De zoekfunctie neemt een groot gedeelte van het zoeken naar een specifieke rij op zich. Hierdoor wordt een gedeelte van de functionaliteit van de tabel verplaatst naar de zoekfunctie.

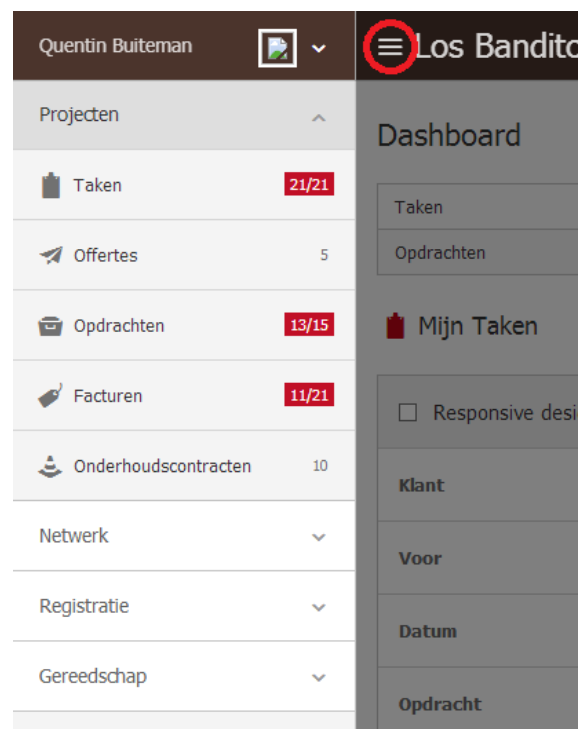
Het grootste nadeel dat hierbij komt kijken, is dat de pagina erg lang wordt. Het gebruik van de zoekfunctie is daarom erg belangrijk. De **:focus** van de pagina wordt daardoor ook gelijk gericht op de zoekfunctie, zodat de gebruiker z'n aandacht hier gelijk op gericht wordt. Ondanks dit nadeel was dit de beste keuze om het overzicht te behouden. De overige oplossing (zoals deze te vinden is in het adviesrapport), is niet goed toepasbaar voor de tabellen van Los Banditos, aangezien deze daarvoor te data-intensief zijn. Deze oplossing is mogelijk gemaakt door een aanpassing aan de HTML en CSS.



Figuur 6.2: Voorbeeld van een responsive tabel

6.2.2. Menu

Het ruimte maken voor de tabellen is ten koste gegaan van het menu. Deze is ingeklapt op het moment dat er te weinig ruimte over was voor de tabellen. Het inklappen van het menu betekende dat er een oplossing moest komen zodat het menu weer zichtbaar gemaakt kon worden. Op basis van de gebruiksverschillen tussen de apparaten (hoofdstuk 3.4) en de analyse van andere responsive websites, zijn er in het adviesrapport een aantal opties opgesomd voor het weergeven van het menu. In het originele ontwerp van Los Banditos is het menu een statische balk aan de linkerkant van het scherm. Met dit in het achterhoofd was er slechts één optie die gebruikt kon worden voor het menu van Los Banditos: een knop welke het menu laat zien. Deze knop is linksbovenaan de pagina geplaatst, zoals ook bij veel moderne applicaties en websites voorkomt (Facebook, Twitter, Mashable, The Next Web).



Figuur 6.3: Voorbeeld van het uitgeklapte menu. In de rode cirkel staat de knop die het menu in- en uitklapt.

Door het klikken op deze knop “duwt” het menu de rest van de pagina uit het scherm. Tevens wordt de rest van de pagina verduisterd, zodat de aandacht volledig op het menu gericht wordt. In figuur 7.3 is goed te zien hoe dit eruit ziet. De knop die gebruikt wordt om het menu in en uit te klappen is aangegeven met een rode cirkel. Deze knop wordt ook bij de meestgebruikte applicaties gebruikt (zoals bijvoorbeeld Facebook). De gebruiker weet dus wat hij kan verwachten bij het klikken op deze knop. Deze oplossing is gemaakt door een aanpassing aan de HTML en CSS en door het aanmaken van een nieuwe JavaScriptfunctie.

6.2.3. Formulieren

De formulieren zelf zijn niet heel erg moeilijk om responsive te maken. Er wordt gebruik gemaakt van een **label** gecombineerd met een **input** of **select**. Deze kunnen op een gegeven moment simpel onder elkaar gaan staan, mocht er te weinig ruimte zijn. Het grootste probleem bij de formulieren ligt in het gebruik van de tabbladen. Aangezien de formulieren over het algemeen veel velden bevatten, zijn ze opgedeeld in verschillende secties. Deze secties worden ingedeeld door het gebruik van tabbladen. Deze tabbladen zijn echter niet makkelijk meer in gebruik bij lagere resoluties. Doordat ze niet meer naast elkaar kunnen staan, zoals in het ontwerp voor de desktop, gaat het overzicht verloren.

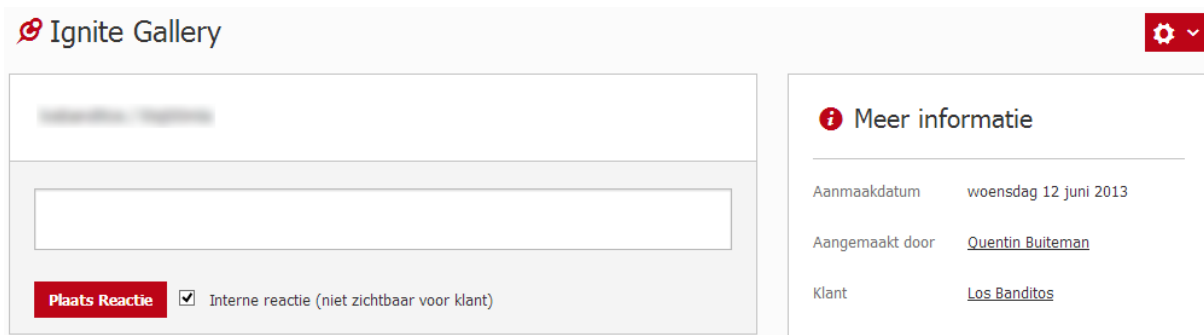
Figuur 6.4: Voorbeeld van het formulier om een taak toe te voegen. Hier wordt gebruik gemaakt van de tabbladen.

De tabbladen kunnen in hoeveelheid variëren. Daarnaast staat de lengte van een tabblad ook niet vast. Er moet dus een relatief hoog aantal pixels gekozen worden als minimale breedte om er zeker van te zijn dat de tabbladen altijd naast elkaar passen. Op basis van de bestaande formulieren is gekozen voor een minimale breedte van 560 pixels. Bij een lager aantal pixels verdwijnen de tabbladen en worden ze vervangen door knoppen onderaan het formulier. Deze oplossing is vooral bedoeld voor kleinere schermen. In verband met de minimale breedte op een klein scherm, kan het formulier erg lang worden. Daarom is er gekozen voor het plaatsen van de knoppen onderaan het formulier. De knoppen (en daarmee de verschillende stappen) vallen dan op tijdens het invullen van het formulier. In de knoppen zijn de teksten van dezelfde teksten als in de tabbladen te vinden, zodat de gebruiker nog steeds weet in welke stap hij zich bevindt tijdens het invullen van het formulier. Het genereren van de knoppen

Figuur 6.5: Responsive formulier

gebeurt door middel van het JavaScript. De opbouw en opmaak worden geregeld door een aanpassing in de HTML en CSS.

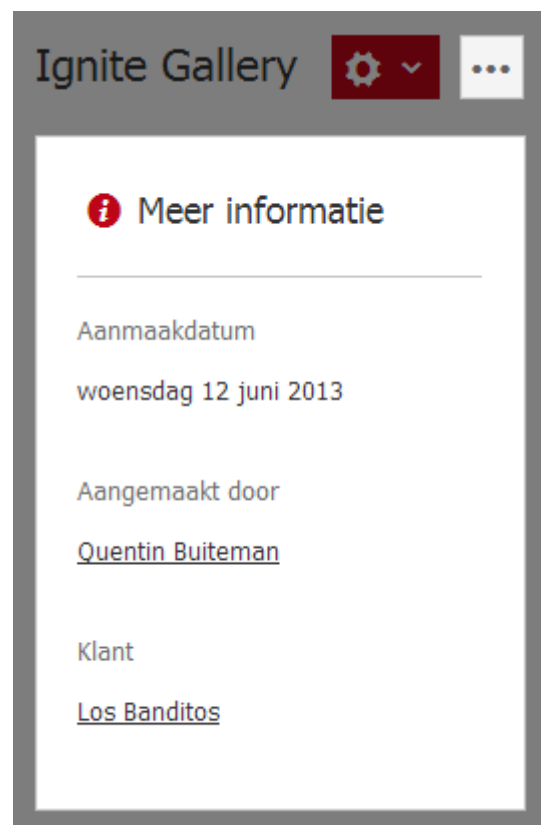
6.2.4. Rechterkolom



Figuur 6.6: Een voorbeeld van de rechterkolom op één van de show-pagina's.

De rechterkolom wordt gebruikt op pagina's waar een specifiek onderdeel bekeken wordt. Hierbij kan gedacht worden aan het bekijken van één factuur of het bekijken van één taak. Deze pagina's worden bij Los Banditos de "show-pagina's" genoemd. In vergelijking met het menu, is de rechterkolom een statisch onderdeel van MijnWerkers. Dit betekent dat dit ook complicaties op begon te leveren bij lagere resoluties.

Voor het menu is gekozen voor een oplossing met een knop. Om consistentie te bewaren, is dit ook voor de rechterkolom gedaan. Op deze manier wordt er getracht om de gebruiker te leren wat hij kan verwachten van het systeem. Dit zorgt ervoor dat het systeem als een geheel aanvoelt. Het enige verschil met de rechterkolom is dat deze over de rest van de pagina heen komt te staan, in plaats van dat hij de rest van de pagina "wegduwt", zoals bij het menu gebeurt. Hier is voor gekozen omdat de rechterkolom deel uitmaakt van de rest van de pagina (hij bevat ondersteunende inhoud), terwijl het menu een los onderdeel is. Door de kolom over de rest van de pagina heen te plaatsen wordt deze gedachte ook gewekt bij de gebruiker. Het icoon wat voor de rechterkolom gebruikt wordt, bestaat uit drie puntjes. Dit icoon is gekozen op basis van een intern overleg. Een voorbeeld van dit icoon is te zien in figuur 6.7. Opnieuw is er gebruik gemaakt van het donker maken van de rest van de pagina, zodat de aandacht volledig gericht wordt op de zojuist ingeschoven inhoud. Door het nogmaals klikken op het icoon (welke daarom dus ook goed zichtbaar blijft) verdwijnt de inhoud weer.



Figuur 6.7: Een voorbeeld van de ingeschoven rechterkolom.

6.3. Leerpunten

6.3.1. Mobile first

Het centrale thema tijdens de case studies is mobile first geweest. Bij de frameworks werd duidelijk dat het gebruik hiervan niet goed duidelijk was gemaakt. Toen dit echter wel gebeurde, vonden beide stagiairs het erg fijn om op deze wijze te werken. Voor mijzelf werd dit helemaal duidelijk tijdens het responsive maken van MijnWerkers. Doordat MijnWerkers niet mobile first is gebouwd, werd het hierbij heel goed duidelijk hoe moeilijk responsive design is. Door de mobile first aanpak werden er veel problemen van tevoren al opgelost bij de frameworks. Er is nooit ruimtegebrek, omdat je ontwerpt met de ruimte die je hebt. Bij MijnWerkers werd hier echter wel veel tegenaan gelopen. Het gebrek aan ruimte door de rechterkolom en het menu zijn hier de twee beste voorbeelden van. De case studies hebben mij daardoor de echte effectiviteit van een mobile first aanpak doen inzien. Deze is veel groter dan dat ik op basis van de theorie dacht.

6.3.2. Apparaten

Het is heel verleidelijk om toch te gaan letten op de apparaten, in plaats van dat er gebruik gemaakt wordt van content-out design. Rogier is bijvoorbeeld in het bezit van een iPhone en een iPad. Bij het testen van MijnWerkers merkte hij op dat hij toch graag het menu aan de linkerkant wilde hebben in de *landscape* stand van de iPad. Doordat er content-out design was toegepast, was dit menu al verborgen bij deze resolutie. Al zou Rogier in het bezit zijn van een Google Nexus 10 (welke een hogere resolutie bevat dan de iPad, maar wel een net zo groot scherm heeft), zou dit nooit opgevallen zijn. Je denkt als ontwikkelaar van een website toch snel aan jouw eigen apparaten. Zo denkt de klant ook aan haar eigen apparaten. Een leerpunt dat ik hieruit kan halen is dat content-out design nooit volledig toegepast kan worden zoals de theorie doet geloven. Je wilt je klant niet teleurstellen dat zijn website net niet helemaal naar wens staat op het apparaat wat hij bezit. In plaats van aan het grotere geheel te denken (namelijk alle duizenden apparaten waarop de website bekeken wordt), wil je toch zorgen dat de website exact staat zoals de klant dat wil op de verschillende apparaten waarop de klant de website bekijkt. In praktijk betekent dit dat de media queries soms met een aantal pixels (tussen de 50 en 150, zoals bleek uit het aanpassen van MijnWerkers) aangepast moeten worden. Content-out design is een mooi beginpunt, maar levert waarschijnlijk niet de definitieve media queries op.

Slotbeschouwing

De case studies bestaan uit twee onderdelen: de frameworks en MijnWerkers. De frameworks zijn door twee stagiairs getest voor twee verschillende projecten van Los Banditos. Tijdens deze tests kwam naar voren dat vooral de uitleg over de werking van de frameworks beter moest. Zo werd er niet goed begrepen dat er mobile first gewerkt moest worden. Ook zaten er een aantal fouten in de code. Niet alle elementen waren goed uitgelijnd, het verschil tussen een module en een component was niet duidelijk en het menu hield geen rekening met een eventueel submenu.

MijnWerkers is responsive gemaakt (in plaats van responsive opgebouwd), waardoor het dus geen mobile first aanpak kon hanteren. Wel is er content-out design toegepast voor het gebruik van media queries. Binnen MijnWerkers worden vier hoofdonderdelen onderscheiden: tabellen, het linkermenu, formulieren en de rechterkolom. De rijen van de tabellen worden bij de bijbehorende breedte overgezet naar overzichtelijke blokken. Zowel het menu als de rechterkolom verdwijnen uit beeld als er geen ruimte meer voor ze is. Deze zijn weer zichtbaar te maken door middel van een knop die in het ontwerp is geplaatst. Bij het klikken op de knop schuift het menu het beeld in. De rechterkolom komt bij het klikken op de knop over de pagina heen te staan. De formulieren maken gebruik van tabbladen voor de verschillende secties waaruit ze bestaan. Als deze niet meer in het beeld passen, worden de tabbladen vervangen door knoppen onderaan het formulier. Zo kan er gebladerd worden tussen de verschillende secties.

Uit de case studies zijn twee leerpunten te halen. Allereerst dat mobile first het maken van een responsive design nog veel meer vergemakkelijkt dan van tevoren gedacht. Dit maakt het hanteren van deze aanpak een grotere prioriteit dan dat origineel de bedoeling was. Ten tweede is te concluderen dat content-out design niet altijd volledig toe te passen is. Zowel de klant als de webontwikkelaar heeft toch altijd een voorkeur voor zijn eigen apparaten, waardoor de media queries vaak toch iets aangepast worden.

In dit hoofdstuk zijn de volgende deelvragen behandeld:

- Uit welke componenten bestaat MijnWerkers?
 - Wat is de prioriteit van elk component?
 - Hoe is de werking van deze componenten toe te passen op de gebruiksmanner van elk apparaat?
 - Welke componenten zijn toepasbaar op de verschillende apparaten?
- Welke kritiekpunten zijn voortgekomen tijdens het responsive maken van MijnWerkers?
 - In welke fase van het proces deed zich dit voor?
 - Wat voor impact had dit?
 - Hoe had dit voorkomen kunnen worden?
 - Op welke manier kan dit veranderd worden in de vernieuwde werkwijze?

7. Conclusie

Responsive design is een nieuwe techniek die voor een verschuiving in de webwereld zorgt. Met de komst van de vele verschillende apparaten waarmee men het internet bezoekt, wordt deze techniek als de 'holy grail' van webdesign gezien. Het geeft de webontwikkelaar de mogelijkheid om een website te maken die zich aanpast aan alle mogelijke schermresoluties. Of de website nu bezocht wordt met een tablet, smartphone, laptop of desktop, de gebruiker krijgt altijd de optimale weergave te zien. De twee kenmerken van responsive design zijn:

- een vloeiende lay-out die gebruik maakt van een flexibele grid;
- afbeeldingen die in een flexibele context werken.

Daarnaast wordt er gebruik gemaakt van een aantal verschillende technieken: het gebruik van relatieve waarden, het gebruik van media queries en het toevoegen van de viewport meta tag. Alledrie deze technieken worden gebruikt bij de responsive websites van Los Banditos. De case studies laten blijken dat deze technieken ook werken zoals werd verwacht.

Mobile first

Tijdens de case studies is er tevens gebruik gemaakt van een mobile first aanpak. Dit wil zeggen dat er eerst voor de meest minimale weergave ontwikkeld wordt. Dit heeft twee voordelen. Allereerst vermindert dit de laadtijd aanzienlijk voor alle mobiele apparaten. Daarnaast is het makkelijker om van een minimale weergave naar een grote weergave te werken, in plaats van andersom. Vooral tijdens de case studies kwam dit laatste goed naar voren

Zowel tijdens het ontwerp- als het ontwikkelproces is mobile first erg belangrijk. De frameworks zijn mobile first opgezet. Hierbij werd duidelijk hoeveel makkelijker het is om iets mobile first op te zetten in plaats van responsive te maken. MijnWerkers daarentegen was responsive gemaakt en kon dus niet mobile first opgezet worden. Vooral bij een product als MijnWerkers, dat heel erg data-intensief is, komt goed naar voren hoe moeilijk het is om dezelfde inhoud als op het grote scherm in te delen op een kleiner scherm. Terwijl juist uit het gebruik van de frameworks bleek dat het andersom wel makkelijk is. Mobile first ontwikkelen is een vereiste voor Los Banditos.

Content-out design

Het is heel verleidelijk om je volledig op de apparaten te richten om de media queries in te stellen. In dit geval zullen er breekpunten gemaakt worden die aangeven welke styles toepasbaar zijn op welke apparaten. Hier zit echter één groot nadeel aan: er zijn geen algemene breekpunten meer. Er zijn ondertussen al zoveel verschillende apparaten met zoveel verschillende resoluties op de markt gebracht, dat het onmogelijk is om de apparaten onder te verdelen. Content-out design is hier een goede oplossing voor. Hierbij wordt gekeken naar de inhoud van een pagina, om op basis daarvan de media queries in te stellen. Uit de case studies blijkt dat dit een goede manier is om de initiële media queries op te stellen. Echter heeft de klant toch altijd een voorkeur voor zijn eigen apparaten. Als de klant het niet eens is met het uiterlijk van de website op één van zijn apparaten, moeten de media queries toch aangepast worden. Dit bleek bijvoorbeeld al bij MijnWerkers. Hierbij wilde Rogier toch net een klein verschil in de media queries, omdat hij een kleine verandering wilde op de iPad. Meestal

gaat dit om kleine verandering van een tiental pixels. Dit heeft daarom ook geen hele grote impact op de responsive websites.

Nieuwe werkwijze

Al de informatie die verkregen is over responsive design is gebruikt om een nieuwe werkwijze op te stellen. Uit de case studies is te concluderen dat er structuur en specialisme ontbreekt binnen Los Banditos. Dit werd van tevoren ook al aangegeven door Rogier. Stagiairs weten soms niet wat ze precies moeten doen. Bij de huidige werkwijze krijgt de stagiair een opdracht voor een nieuwe website. Hij moet eerst een ontwerp maken, om deze daarna om te zetten naar code. Vooral het omzetten naar code is een onderdeel dat veel structuur vereist. Er kan niet zomaar wat gedaan worden. De stagiair moet allereerst mobile first werken en daarnaast gebruik maken van content-out design voor het aanmaken van de media queries. Deze structurele aanpak moet bij elke website doorlopen worden.

Door scrum te implementeren in de werkwijze van Los Banditos, weet de stagiair altijd wat er moet gebeuren. Daarnaast weet Rogier altijd waar de stagiair mee bezig is. In de scrumbijeenkomsten kunnen eventuele afspraken gemaakt worden over de vordering van een project. Zo zit er dus veel meer controle op het goede verloop van het ontwikkelen van een responsive design.

Rogier wil meer gebruik gaan maken van stagiairs met een specialisme in plaats van 'all-rounders'. Dit zorgt ervoor dat dezelfde stagiair niet meer én aan het ontwerp én aan de ontwikkeling werkt. Met scrum vormen meerdere stagiairs samen het ontwikkelingsteam. Rogier functioneert als de scrummaster en de klant als de product owner. Voordat een project is aangenomen zijn dus eigenlijk alle functies binnen een project al gevuld.

Aan de hand van het gemaakte adviesrapport is Los Banditos in staat om elke nieuwe website af te leveren met een responsive design.

Literatuurlijst

- Apple Inc. (2012, December 17). *iOS Human Interface Guidelines*. Opgeroepen op Maart 12, 2013, van Apple Developer: <http://goo.gl/v4EgD>
- Archer, P., & Mitukiewicz, E. (2005, December 20). *Scope of Mobile Web Best Practices*. Opgeroepen op Februari 21, 2013, van W3C: <http://goo.gl/aZt4a>
- Artman, J. (2011, May 12). *3G Speed Vs. WiFi*. Opgehaald van eHow: <http://goo.gl/mUapb>
- Besse, J. (2012, December 13). *Disadvantages of Social Media Marketing*. Opgeroepen op Februari 21, 2013, van Empower Network: <http://goo.gl/WYxel>
- Boman, E. (2012, November 21). *The importance of whitespace in web design*. Opgeroepen op Februari 21, 2013, van .net magazine: <http://goo.gl/lsebk>
- Bos, R. (2012, September 22). *Trends en ontwikkelingen in web design*. Opgeroepen op Februari 21, 2013, van Marketingfacts: <http://goo.gl/xsJK1>
- Boulton, M. (2013, Februari 14). *The In-Between*. Opgeroepen op Maart 5, 2013, van The Personal Disquiet of Mark Boulton: <http://goo.gl/yhNNM>
- Bringhurst, R. (2004). *The Elements of Typographic Style*. Hartley & Marks Publishers.
- Byers, J. (2011, November 29). *Mobile Responsive Design: The Flexible Grid*. Opgeroepen op Februari 22, 2013, van Studiopress: <http://goo.gl/THAhx>
- Callahan, B. (2013, Februari 7). *There Is No Breakpoint*. Opgeroepen op Maart 5, 2013, van Sparkbox: <http://goo.gl/Puqs3>
- Can I Use. (2013, Januari 1). *Can I Use CSS3 Media Queries*. Opgeroepen op Februari 25, 2013, van Can I Use: <http://goo.gl/5GbSO>
- Cant, E. (2012). *Optimising for High Pixel Density Displays*. Opgeroepen op Februari 27, 2013, van Menacing Cloud: <http://goo.gl/7w8Cs>
- Casel, B. (2013, Januari 4). *Keep an Eye on These Web Design Trends in 2013*. Opgeroepen op Februari 21, 2013, van Mashable: <http://goo.gl/G82p7>
- Coyier, C. (2012, Maart 27). *Lazy Loading Images*. Opgeroepen op April 9, 2013, van CSS-Tricks: <http://goo.gl/bCfJg>
- de Bruin, M. (2012, November 22). *5 social media-trends 2013 met Trendstrateeg @LiekeLamb*. Opgeroepen op Februari 21, 2013, van Marketingfacts: <http://goo.gl/V24W4>
- De Telegraaf. (2013, Februari 4). *KPN lost startschot 4G in Amsterdam*. Opgeroepen op Februari 19, 2013, van De Telegraaf: <http://goo.gl/1TliF>

- Design Lunatic. (2011, November 15). *Pros and Cons of Responsive Design*. Opgeroepen op Februari 19, 2013, van Design Lunatic: <http://goo.gl/gX4yd>
- Facebook. (2013). *Facebook Reports Fourth Quarter and Full Year 2012 Results*. Facebook.
- Fadeyev, D. (2012, December 5). *Redesign Trend in Tech News Sites: Big, Responsive and Content Heavy*. Opgeroepen op Maart 5, 2013, van UsabilityPost: <http://goo.gl/v19am>
- Fox, M., & Kemp, M. (2009). *Interactive Architecture*. Princeton Architectural Press.
- Franklin, E. (2012, Oktober 30). *Tablets buying guide*. Opgeroepen op Februari 27, 2013, van CNET: <http://goo.gl/RhdEV>
- Frost, B. (2012, Maart 3). *Content Parity*. Opgeroepen op Maart 5, 2013, van Brad Frost Web: <http://goo.gl/ihckT>
- Gardner, B. S. (2011). Responsive Web Design: Enriching the User Experience. In M. K. Corrigan, *Sigma: Inside the Digital Ecosystem* (pp. 13-19). Falls Church: Noblis.
- Gardner, L. (2012, Maart 28). *The EMs have it: Proportional Media Queries FTW!* Opgeroepen op Maart 5, 2013, van Cloud Four Blog: <http://goo.gl/z5izd>
- Gartner. (2012, Oktober 23). *Gartner Identifies the Top 10 Strategic Technology Trends for 2013*. Opgeroepen op Februari 21, 2013, van Gartner: <http://goo.gl/1brJO>
- Gemert, V. v. (2013, Maart 1). *Logical Breakpoints For Your Responsive Design*. Opgeroepen op Maart 5, 2013, van Smashing Magazine: <http://goo.gl/c2c4v>
- Ghosh, R. (2013, Januari 2). *Top 5 Web Design Trends in 2013*. Opgeroepen op Februari 21, 2013, van Technorati: <http://goo.gl/zSr3y>
- Goodwin, R. (2013, Februari 21). *HTC One beats Sony Xperia Z on pixel density and viewing angles*. Opgeroepen op Februari 27, 2013, van Know Your Mobile: <http://goo.gl/gLGKP>
- Guldemon, J. (2011, April 6). *Wat is Scrum? Uitleg over gebruik van Scrum methodologie bij Insyde*. Opgeroepen op Maart 25, 2013, van Insyde: <http://goo.gl/QgzJc>
- Irish, P. (2012, Februari 1). **{Box-sizing: Border-box} FTW*. Opgeroepen op April 3, 2013, van Paul Irish: <http://goo.gl/d27QI>
- Jehl, S. (2012, Maart 28). *An Ajax-Include Pattern for Modular Content*. Opgeroepen op April 9, 2013, van Filament Group: <http://goo.gl/kEbbk>
- Jongerius, P. (2010, Mei 19). *Een website ontwerpen met agile design en scrum (3): Teams en overleg*. Opgeroepen op Maart 25, 2013, van Frankwatching: <http://goo.gl/azPo4>
- Kadlec, T. (2012, Oktober 4). *Blame the Implementation, Not the Technique*. Opgeroepen op April 9, 2013, van TimKadlec.com: <http://goo.gl/Pogsk>

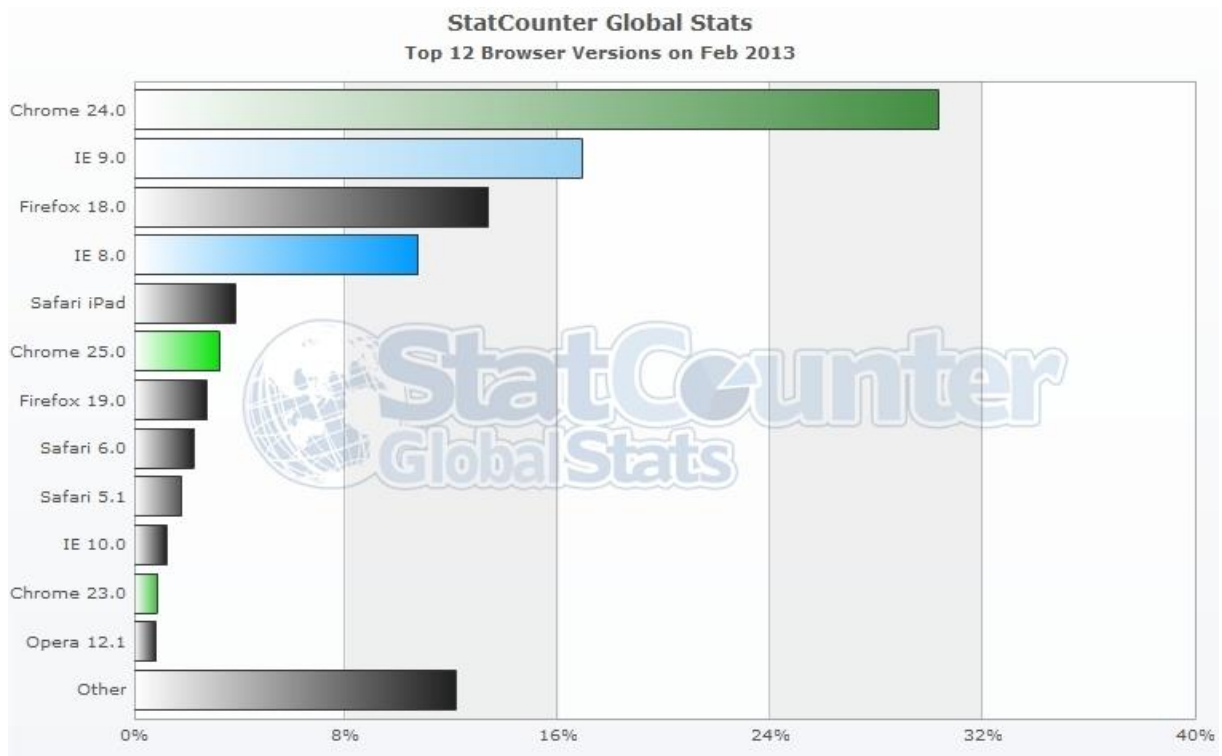
- Koch, P.-P. (2010, April 20). *A pixel is not a pixel is not a pixel*. Opgeroepen op Februari 27, 2013, van QuirksMode: <http://goo.gl/EtLgZ>
- Koczon, C. (2011, April 19). *Orbital Content*. Opgeroepen op Maart 5, 2013, van A List Apart: <http://goo.gl/2u1ro>
- La, N. (2010, September 21). *960 Grid System is Getting Old*. Opgeroepen op Februari 22, 2013, van Web Designer Wall: <http://goo.gl/EYSqy>
- Lazaris, L. (2012, Maart 14). *Understanding em Units in CSS*. Opgeroepen op April 3, 2013, van Impressive Webs: <http://goo.gl/LmcVI>
- Lunden, I. (2012, Juli 31). *Twitter May Have 500M+ Users But Only 170M Are Active, 75% On Twitter's Own Clients*. Opgeroepen op Februari 21, 2013, van TechCrunch: <http://goo.gl/cqP7n>
- Lutz, Z. (2012, April 2). *Engadget's smartphone buyer's guide: spring 2012 edition*. Opgeroepen op Februari 27, 2013, van Engadget: <http://goo.gl/kFZhC>
- Marcotte, E. (2010, September 14). *On being "responsive"*. Opgeroepen op Februari 19, 2013, van Unstoppable Robot Ninja: <http://goo.gl/yGiZI>
- Marcotte, E. (2010, Mei 25). *Responsive Web Design*. Opgeroepen op Februari 19, 2013, from A List Apart: <http://goo.gl/dj2ox>
- Marcotte, E. (2011). *Responsive Web Design*. A Book Apart.
- Microsoft. (sd). *Touch interaction design (Windows Store Apps) (Windows)*. Opgeroepen op Maart 13, 2013, van Windows: <http://goo.gl/3Fv6m>
- Morgan Stanley Research. (2010). *The Mobile Internet Report*. Morgan Stanley & Co. Incorporated. Opgehaald van Morgan Stanley.
- Müller-Brockmann, J. (1996). *Grid Systems in Graphic Design*. Niggli Verlag.
- Ødegård, A. (2012, Februari 13). *Tablets and screen resolution*. Opgeroepen op Februari 27, 2013, van Pocketables: <http://goo.gl/WYiLg>
- Rabin, J., & McCathieNevile, C. (2008, Juli 29). *Mobile Web Best Practices 1.0*. Opgeroepen op Februari 21, 2012, van W3C: <http://goo.gl/X7riR>
- Remick, J. (2008, December 29). *The Golden Ratio in Web Design*. Opgeroepen op Maart 5, 2013, van Nettuts+: <http://goo.gl/XOqi>
- Rivoal, F. (2012, Juni 19). *Media Queries*. Opgeroepen op Februari 25, 2013, van W3C: <http://goo.gl/ovLtj>
- Schroeder, S. (2010, April 29). *Steve Jobs: Flash Is No Longer Necessary*. Opgeroepen op Februari 21, 2013, van Mashable: <http://goo.gl/mGPYY>

- Schroeder, S. (2012, Juni 29). *Adobe Won't Support Flash in Android 4.1*. Opgeroepen op Februari 21, 2013, van Mashable: <http://goo.gl/QoJM8>
- Schwaber, K., & Beedle, M. (2001). *Agile Software Development with Scrum*. Pearson Education.
- Shanklin, W. (2012, November 7). *2012 Smartphone Comparison Guide*. Opgeroepen op Februari 27, 2013, van Gizmag: <http://goo.gl/uRRra>
- Snook, J. (2011, Mei 1). *Font sizing with rem*. Opgeroepen op April 3, 2013, van Snook.ca: <http://goo.gl/85fhM>
- Spooner, C. (2012, December 14). *Line25 Sites of the Week for December 14th 2012*. Opgeroepen op Februari 21, 2013, van Line25: <http://goo.gl/YAVVz>
- StatCounter Global Stats. (2013, Februari 27). *Top 10 Screen Resolutions on Feb 2013*. Opgeroepen op Februari 27, 2013, van StatCounter Global Stats: <http://goo.gl/9Wt2y>
- StatCounter Global Stats. (2013, Februari 27). *Top 12 Browser Versions on Feb 2013*. Opgeroepen op Februari 27, 2013, van StatCounter Global Stats: <http://goo.gl/o6YxK>
- StatCounter Global Stats. (2013, Februari 27). *Top 14 Mobile Screen Resolutions on Feb 2013*. Opgeroepen op Februari 27, 2013, van StatCounter Global Stats: <http://goo.gl/Kl2hu>
- StatCounter Global Stats. (2013, Februari 27). *Top 9 Mobile Browsers on Feb 2013*. Opgeroepen op Februari 27, 2013, van StatCounter Global Stats: <http://goo.gl/EV6A7>
- Summer, G. (2012, Februari 27). *How many web browsers support responsive design?* Opgeroepen op Februari 25, 2013, van Gabe Summer: <http://goo.gl/TTCvv>
- Szalvay, L. (2009, Oktober 24). *Scrum User Stories*. Opgehaald van Scrum Methodology: <http://goo.gl/ipKFt>
- Turner, A. L. (2012, December 30). *10 Web design trends you can expect to see in 2013*. Opgeroepen op Februari 21, 2013, van The Next Web: <http://goo.gl/vc44o>
- Varsamis. (2012, Juli 17). *Advantages of Responsive Web Design*. Opgeroepen op Februari 19, 2013, van InstantShift: <http://goo.gl/oGj55>
- Vaughan-Nichols, S. J. (2013, Januari 26). *Google+ moves up to second place in social networks*. Opgeroepen op Februari 21, 2013, van ZDNet: <http://goo.gl/w63v4>
- Verbeek, P. (2012, Juli 23). *Mobiel testen, een openbaar testlab*. Opgehaald van Netvlies: <http://goo.gl/qa75V>
- VersionOne. (2013). *Agile Methodologies for Software Development*. Opgehaald van VersionOne: <http://goo.gl/eTs28>
- W3Schools. (2013). *Browser Statistics*. Opgeroepen op Februari 19, 2013, van W3Schools: <http://goo.gl/7xZ1m>

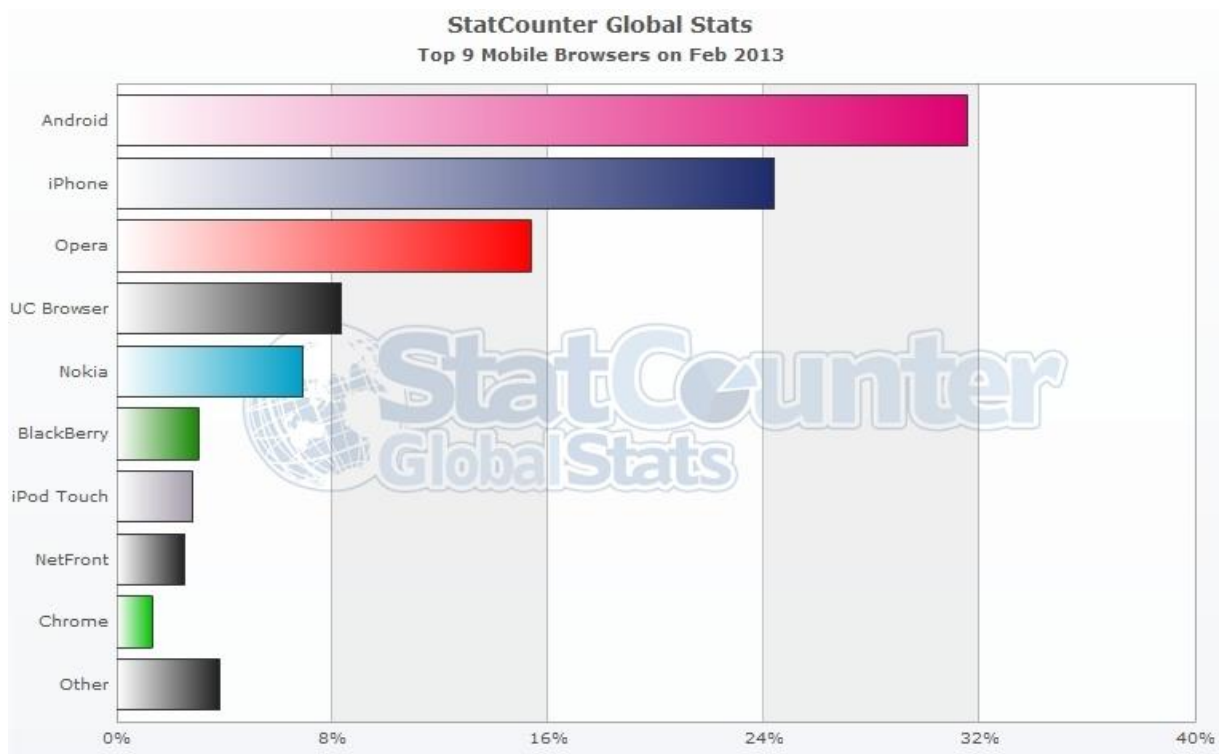
- Wroblewski, L. (2011). Mobile First. In L. Wroblewski, *Mobile First* (pp. 47-119). New York: A Book Apart.
- Wroblewski, L. (2012, November 18). *Data Monday: Physical Device & Viewport Sizes*. Opgeroepen op Februari 27, 2013, van LukeW: <http://goo.gl/gW4LV>
- Wroblewski, L. (2012, Februari 9). *Josh Clark: Busting Mobile Myths*. Opgeroepen op Maart 5, 2013, van LukeW: <http://goo.gl/g2Gto>
- Wroblewski, L. (2012, Mei 3). *Targeting devices that don't account for physical size in Viewport*. Opgeroepen op Februari 27, 2013, van Storify: <http://goo.gl/ba8aZ>
- Yarow, J. (2012, December 3). *Mary Meeker 2012 Internet Trends Year-End Update*. Opgeroepen op Februari 21, 2013, van Business Insider: <http://goo.gl/oDIBn>
- Yates, I. (2012, Februari 13). *Quick Tip: Don't Forget the Viewport Meta Tag*. Opgeroepen op Februari 22, 2013, van Webdesigntuts+: <http://goo.gl/rhPaf>

Bijlagen

Bijlage A. Meest gebruikte browsers

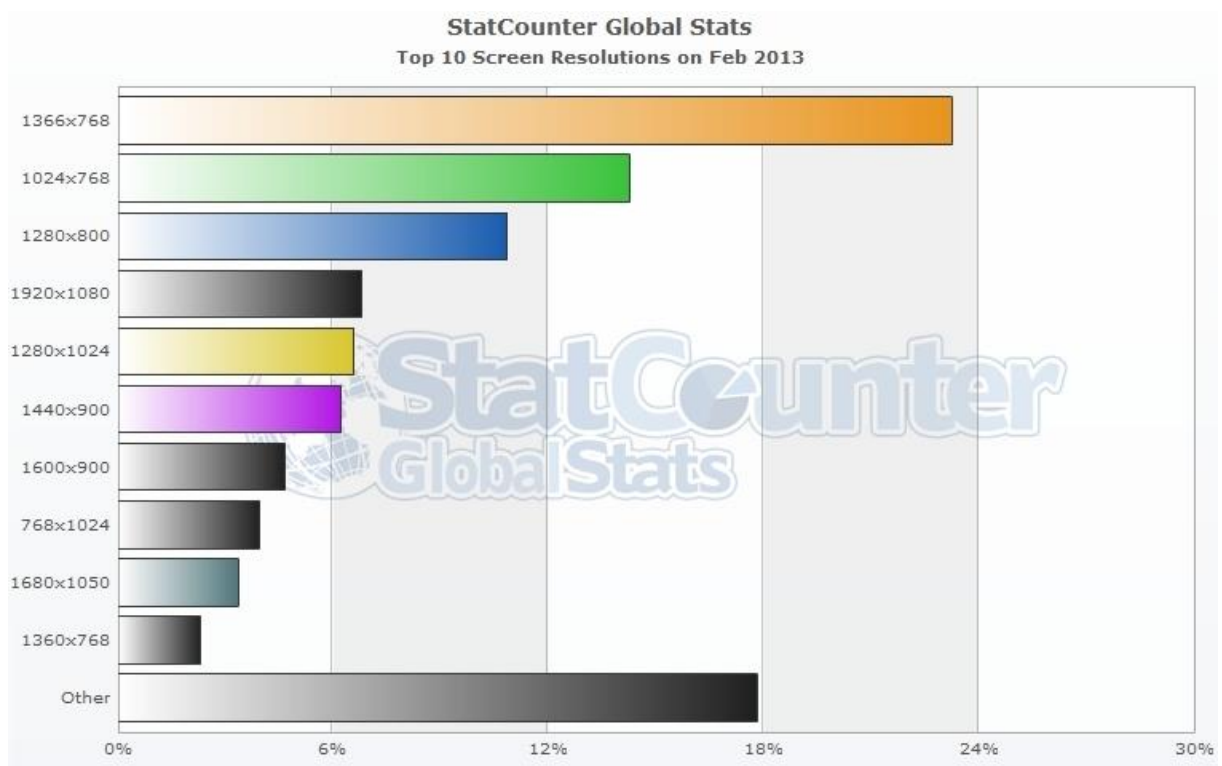


De meest gebruikte browsers gedurende februari 2013 (StatCounter Global Stats, 2013).

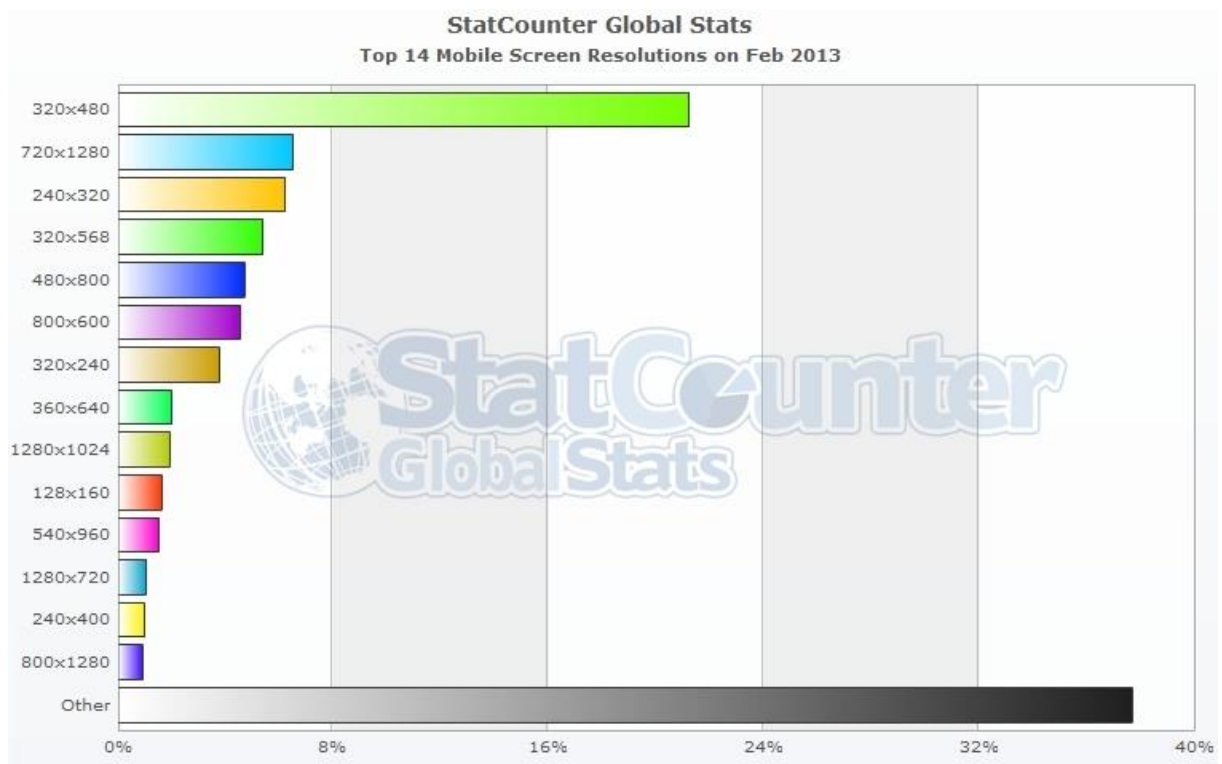


De meest gebruikte mobiele browsers gedurende februari 2013 (StatCounter Global Stats, 2013).

Bijlage B. Schermresoluties van verschillende apparaten

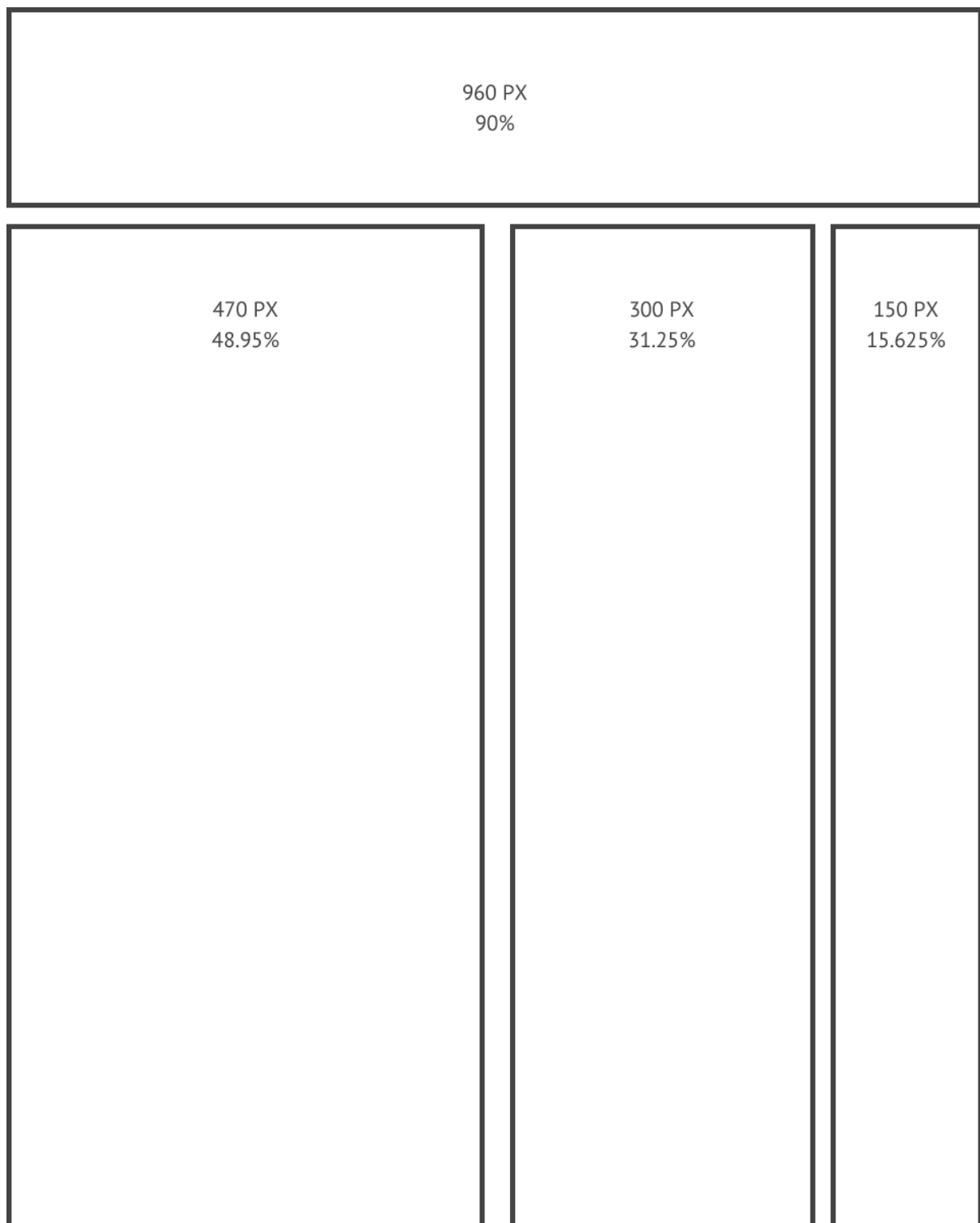


De meest gebruikte schermresolutie gedurende februari 2013 (StatCounter Global Stats, 2013).



De meest gebruikte schermresolutie van smartphones gedurende februari 2013 (StatCounter Global Stats, 2013).

Bijlage C. Voorbeeldpagina met kolommenindeling



Bijlage D. Lijst met mogelijke media query waardes

width	De breedte van de browser waarin de pagina bekeken wordt uitgedrukt in CSS pixels.
height	De hoogte van de browser waarin de pagina bekeken wordt uitgedrukt in CSS pixels.
device-width	De breedte van het scherm van een apparaat uitgedrukt in CSS pixels.
device-height	De hoogte van het scherm van een apparaat uitgedrukt in CSS pixels.
orientation	Als de height groter is dan de width, dan is dit 'portrait'. Als de width groter is dan de height, dan is dit 'landscape'.
aspect-ratio	De ratio tussen de width en de height waardes.
device-aspect-ratio	De ratio tussen de device-width en de device-height waardes.
color	De waarde van het aantal bits per kleurcomponent.
color-index	De hoeveelheid kleuren dat een apparaat bevat.
monochrome	Het aantal bits per pixel in een monochrome apparaat. Indien het apparaat niet monochroom is, is de waarde 0.
resolution	De dichtheid van de resolutie uitgedrukt in punten per inch (dpi) of punten per centimeter (dpcm)
scan	De scanningsmanier van TV's.
grid	Geeft aan of het apparaat grid of bitmap is. Grid wordt aangegeven met waarde 1, bitmap met waarde 0.

Bijlage E. Overzicht van browsers die media queries ondersteunen

	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry browser	Opera mobile	Chrome for Android	Firefox for Android
20 versies terug			4.0									
19 versies terug		2.0	5.0									
18 versies terug		3.0	6.0									
17 versies terug		3.5	7.0									
16 versies terug		3.6	8.0									
15 versies terug		4.0	9.0									
14 versies terug		5.0	10.0									
13 versies terug		6.0	11.0									
12 versies terug		7.0	12.0									
11 versies terug		8.0	13.0									
10 versies terug		9.0	14.0		9.0							
9 versies terug		10.0	15.0		9.5 – 9.6							
8 versies terug		11.0	16.0		10.0 – 10.1							
7 versies terug		12.0	17.0		10.5							
6 versies terug		13.0	18.0		10.6			2.1				
5 versies terug	5.5	14.0	19.0	3.1	11.0			2.2		10.0		
4 versies terug	6.0	15.0	20.0	3.2	11.1	3.2		2.3		11.0		
3 versies terug	7.0	16.0	21.0	4.0	11.5	4.0 – 4.1		3.0		11.1		
2 versies terug	8.0	17.0	22.0	5.0	11.6	4.2 – 4.3		4.0		11.5		
Vorige versie	9.0	18.0	23.0	5.1	12.0	5.0 – 5.1		4.1		12.0		
Huidig	10.0	19.0	24.0	6.0	12.1	6.0	5.0 – 7.0	4.2	7.0	12.1	18.0	18.0
Nabije toekomst		20.0	25.0		12.5				10.0			
Verre toekomst		21.0	26.0									

Niet ondersteund



Gedeeltelijk



Wel ondersteund

