

Mobile-first eller Desktop-first, en studie om utvecklingslösningar för responsive web design

Eduardo Castaneda

11 augusti 2013

Sammanfattning

I examensarbetet analyseras mobile-first och desktop-first vilka är två olika tillvägagångssätt för responsiv web design. Responsiv web design är ett koncept för webbsidor vars design anpassas utifrån skärmstorlek. Analysen utförs genom att samla in synpunkter kring metoderna och analysera användningsområdet för mobilt och desktop internet för att skapa en prototyp av en responsive webbplats där båda metoder utförs.

Den största skillnaden mellan metoderna är att grunden är avsedd för mobilen respektive skrivbordskärmen. Arbetet visar att mobile-first ger ett bättre resultat för mobilen, vilken tillskillnad från en dator har sämre prestanda, mindre skärm och sämre internetuppkoppling, därav en fördel för den enhet som behöver fokus för att upprätthålla snabb respons. Desktop-first tar inte det i akt, utan fokus sker för datorn där prestandan är så pass hög att skillnaden för responsen är liten. Resultat visar även det naturliga arbetssättet mobile-first tillför då designen för mobilen utvecklas utifrån en perspektiv som är standard inom webbutveckling. Desktop-first har fördelen att vara ett snabbt tillvägagångssätt för att skapa responsivt utifrån en existerande webbsida, men kommer däremot att sakna den stabila grund mobile-first kan uppnå.

Implementationen av prototypen har inte kunnat styrka alla påståenden kring metoderna. Men ändå kunnat styrka tillräckligt många för att lyfta mobile-first som den metod som gynnar utvecklingen av responsiva sidor och den utveckling som sker inom internetanvändning. En utveckling där mobilt internet ökar markant och fler webbsidor ses utifrån mobilen.

Innehåll

1	Inledning	5
1.1	Mobile-first eller Desktop-first	5
1.2	Valtech AB	6
1.3	Syfte	6
1.4	Avgränsningar	7
1.5	Disposition	7
2	Bakgrund	8
2.1	Mobilanvändare	8
2.2	Desktopanvändare	9
3	Teori	10
3.1	Verktyg	10
3.1.1	HTML	10
3.1.2	CSS	10
3.1.3	JavaScript	12
3.2	Responsive web design	13
3.2.1	Fluid Grid	13
3.2.2	Fluid Images	14
3.2.3	Media Queries	14
3.3	Mobile-first	17
3.4	Desktop-first	17
4	Metod	19
4.1	Litteraturstudie	19
4.1.1	Miljö	19
4.1.2	Målgrupp	19
4.1.3	Kontext	20
4.2	Implementation av webbsida	20
4.2.1	Prototyp	21
4.2.2	Webbsidan i Desktop-first	21
4.2.3	Webbsidan i Mobile-first	21
5	Resultat	22
5.1	Litteraturstudie	22
5.1.1	Samlade synpunkter kring Mobile-first och Desktop-first	22
5.1.2	Användningsområde	27
5.2	Implementation	30
5.2.1	Prototyp	30
5.2.2	Responstid	33
5.2.3	Kodstorlek	35
5.2.4	Implementeringstid	35

5.2.5	Kommentarer från dagboken	35
6	Diskussion	37
6.1	Mobile- vs Desktop-first	37
6.1.1	Responstid	38
6.1.2	Kodanalys	39
6.1.3	Implementeringstid	41
6.1.4	Skärmstorlek	42
6.2	För- och Nackdelar	43
6.3	När ska Desktop-first respektive Mobile-first användas	45
7	Slutsats	47

1 Inledning

Användningen av mobilt internet ökar för varje dag och nya webblösningar har krävts för att nå ut till mobilanvändarna. En av lösningarna har varit responsive web design. Responsive web design innebär att samma webbsida renderas olika beroende på enheten webbsidan ses från och kan på så sätt anpassas utifrån skärmstorleken. Tekniken för att skapa responsiva webbsidor existerar, med hjälp av utvecklingen av tekniker och verktyg så som CSS, HTML finns i nuläget möjligheten att skapa webbsidor som beroende på skärmstorlek har olika designlayouts med en och samma kodgrund. Det finns även kunskap om tillvägagångssätt för skapandet av responsiva webbsidor, där utvecklare med formler och riktlinjer kan lära sig att använda grundprinciperna fluid grid, fluid images och media queries på bästa sätt. Däremot är vägen till att skapa en responsiv webbsida olika. I bloggar och forum på webben diskuteras flitigt valet av utvecklingsmetod när det kommer till responsiva webbplatser och innan implementeringen av en webbsida är detta en fråga som med stor sannolikhet dyker upp hos webbutvecklare. Utvecklingsmetoderna som diskuteras är *Mobile-first* och *Desktop-first*. Mobile-first och Desktop-first är två olika metoder där det responsiva angreppssättet används. Båda strävar efter samma mål, men skiljer sig i implementeringen för responsive web design.

1.1 Mobile-first eller Desktop-first

Kännetecken för metoderna är grunden på strukturen och designen, där storleken på webb-läsaren avgör på vilket sätt den skall ändras för att anpassas för en annan enhet. Kunskapen om metoderna var för sig har med tiden blivit större i samband med utvecklingen av responsive web design. Däremot ställs webbutvecklare kring frågan om vilken metod som appliceras bäst till typen av webbsida som skall skapas. I dagsläget finns ingen kunskap om hur bra metoderna appliceras i jämförelse med varandra. Det finns tankar och spekulationer, därefter väljs metod utifrån dessa, komplikationer hanteras men dokumenteras inte vilket tillslut leder till en fungerande webbsida utan vetskapen ifall bästa lösning är applicerad. Eftersom webbsidan är färdig implementerad finns ingen anledning till att implementera om en fungerande lösning med en annan metod, därav finns ingen större kunskap om jämförelse av implementationen med respektive metod riktat mot ett och samma resultat.

Användning av mobilt internet ökar och användningen av internet via en dator kommer med stor sannolikhet bestå. Det gör att en grund för val av metod är högst passande, då det redan nu och i framtiden kommer att kräva mer kunskap än vad som idag kan erbjudas från bloggar. Båda metoder har sina fördelar samt nackdelar, vilket bör lyftas fram tydligare. Frågan som även bör ställas är när metoderna utnyttjas på bästa sätt och kan valet av metod leda till positiva faktorer vilka motsatta metod inte hade kunnat uppnå.



Figur 1: En responsive webbsida kan även anpassas till enheter mellan desktop och mobil, t.ex tablets och ipads. I arbetet kommer fokus endast ske för mobil och desktop.

1.2 Valtech AB

Examensarbetet kommer att utföras på Valtech AB i Stockholm. Valtech Sverige fokuserar främst på utveckling av användaranpassade webbsidor, webbapplikationer och intranät. Hos Valtech finns erfarna gränssnittsutvecklare som har stött på problem under utvecklingsprocessen i form av ett beslutstagande av tillvägagångssätt för en responsiv webbsida. Det finns ett behov hos utvecklare på Valtech att ha en grund för vilka faktorer som är viktigast vid beslutstagande för metod och som ger den bästa lösning för en effektiv utveckling av en responsiv webbplats. Gränssnittsutvecklarna besitter på mycket erfarenhet och kunskap, vilket kan samlas och användas som vägledning för att finna problem som uppstår i verkliga situationer.

1.3 Syfte

Syftet med arbetet är att kunna hitta riktlinjer till när en specifik metod appliceras bäst. Det vill säga beroende på faktorer kring webbsidan hitta den metod som tillför det mest optimala lösningen för implementationen av en webbsida. Att hitta en metod som fungerar bäst i alla lägen är inte nödvändigtvis målet med arbetet då det existera många olika vinklar, vilka gör det svårt att hitta en specifik metod som ger det optimala resultatet oberoende på situation.

Frågeställningar som besvaras i rapporten är:

1. I vilka lägen appliceras metoderna bäst?
2. Vad är för- och nackdelarna med Mobile-first?
3. Vad är för- och nackdelarna med Desktop-first?

Frågeställning 1 baseras på miljö, målgrupp och kontext. Syftet med frågeställningen är att kunna utifrån analys hitta situationer där metoderna visar sig vara fördelaktiga under utvecklingsfasen, för att hitta faktorer hos metoden vilka motsatta metod inte har och därmed inte kan tillföra samma lyckade resultat. Syftet med *Frågeställning 2 och 3* är att kunna lyfta fram de fördelar och nackdelar som finns vid implementation med de två metoderna. För att på så sätt hitta ramar för varje metod vilka en läsare kan relatera till med en egen situation och utifrån det använda det som grund vid val av metod.

1.4 Avgränsningar

Arbetet kommer att enbart fokusera på mobile-first och desktop-first (*se figur 1*). Ett mellanläge som existerar för Ipads, tablets, osv. kommer inte att tas med i arbetet utan kommer att ses som desktopläge. I dagsläget finns även andra lösningar för mobilt webb t.ex *Appar* och *separata mobilsidor*, jämförelse mellan dessa och responsive web design kommer inte att göras, utan jämförelsen som görs är mellan mobilalösningar inom responsive web design.

Termen *Mobile-first* kan tolkas på olika sätt. Det har förekommit tillfällen för webbutvecklare där mobile-first är prioriterad designmässigt men där implementeringen ändå har skett för desktop innan mobilen. Det är inte fallet i arbetet, utan mobile-first beskrivs som en tanke och en implementering avsedd för mobilen i första hand, och tvärtom för desktop-first.

1.5 Disposition

Kommande kapitler är indelade i *Bakgrund*, *Teori*, *Metod*, *Resultat* och *Diskussion*. I *bakgrund* beskrivs den ökning av användning som har skett för mobilt internet under de senaste åren och även användningen av desktop internet som har bestått. I bakgrund beskrivs även betydelsen dessa har för webbutvecklingen och varför responsive web design är en metod som kan gynna dem. I kapitlet *teori* förklaras tekniken responsive web design samt de verktyg och metoder som samverkar för att få en fungerande responsiv webbplats, på så sätt tillföra en teknisk förståelse för hur webbutvecklingen av mobile-first och desktop-first går till. Metoderna för arbetet är en litteraturstudie samt en prototyp av en webbsida som både implementeras i mobile-first och desktop-first. I kapitlet *metod* beskrivs valet av metoderna samt hur dessa samspelas för att få fram resultat som besvarar frågeställningarna för arbetet. I nästa kapitel, *resultat*, samlas resultatet från tillvägagångssätten som beskrivs i metod, vilket senare diskuteras i kapitlet *diskussion*, där även frågeställningarna besvaras.

2 Bakgrund

Mobilutvecklingen har under de senaste åren gått fort framåt, vilket har lett till att mobiler numera utvecklas till att fungera likt en handburen minidator. Huvudsyftet med mobiler är inte längre enbart kommunikation med andra, utan även att tillgå information från webben. Att en webbsida inte längre endast ses från en skrivbordsskärm har gjort att nya webblösningar krävs. En utvecklare måste ha i åtanke att webbdesignen bör fungera för en användare som sitter på kontoret och läser sidan från en skrivbordsskärm, likaväl för en användare som sitter på tåget och läser sidan från mobilen. *Responsive Web Design* är en lösning vilken beroende på skärmstorlek renderar samma sida på olika sätt med olika design. I responsive web design har det fokuserats på två olika metoder, *Mobile-first* och *Desktop-first*. Metoderna grundar sig på att utveckling sker för en typ av skärm först för att sedan utifrån det utveckla så att den sidan även passar för andra skärmar. Det finns mycket kunskap om metoderna var för sig, men inget underlag för valet av metod i olika situationer och inom olika områden. Mobilanvändarna ökar för varje dag medan skrivbordsanvändare behåller sina användare med en viss minskning. Det största antalet är användare av båda, vilket kräver kunskap om när metoderna appliceras bäst för att få effektivare lösningar inom webbutveckling.

2.1 Mobilanvändare

Sedan 2007 då Apple visade upp den första versionen av iPhones (Seitz 2009) har utvecklingen för *smartphones* eskalerat markant. Smartphones ger dig möjligheten att utföra datorliknande handlingar som att tillgå information från webben och använda nätbaserade applikationer. Tidigare mobiler har delvis haft den funktionen men fokus har mestadels riktat sig mot musikspelare och kamera, då surfande gick trögt och var icke praktiskt användbart. Smartphones innebar större pekskrärmar och gjorde åtkomsten till internet på mobilen mer användaranpassade, vilket har under de senaste åren utvecklats till en oundviklig funktion i dagens mobiler (Helmreich m. fl. 2009, s. 4). Även prisutvecklingen har varit en givande faktor till användningen av mobilt internet, då telekomföretag har under de senaste åren varit tvungna att sänka avgifterna samt erbjuda fasta priser för att kunna passa in i marknaden. Sedan 2007 fram tills 2012 har lägsta priser för mobilt bredband minskat med 60%. I dagens erbjudanden finns abonnemang med fri surf för 89kr i månaden med 18 månaders bindningstid. Samma erbjudande kostade år 2007 169kr i månaden (Fransén m. fl. 2012, s. 36). Numera satsas det på att även kunna erbjuda ett fast pris för att surfa utomlands, vilket tyder på att en breddning för användandet görs och antal svenskar som använder mobilt internet fortsätter att öka (Gustafsson 2012).

Av svenskar mellan 12-79 år är det cirka 54% som använder sig av mobilt internet, i jämförelse med 2008 har det ökat med ungefär 70%. Under tidiga åren under 2000-talet låg ökningen per år på 1%, vilket visar att stora genombrottet har skett under de två senaste åren (Findahl 2012, s. 24). Telefonoperatörer har anpassat sig till den nya marknaden och

erbjuder abonnemanger vilka innehåller en fast kostnad för surf på ett 3G nät. Då mobilsurf anses som en nödvändig funktion, sätts krav hos mobiloperatörer i form av att hög åtkomlighet och en hög hastighet för 3G nätet (Helmreich m. fl. 2009, s. 17). Effekten av att mobilanvändarnas åtkomst till internet ökar, leder till att mobilt internet används mer och äldre tjänster ersätts. En tidning i tunnelbanan är inte alls lika vanlig nu som för fem år sen, tidtabell vid busshållplatserna är inte enda sättet att få ut information om busstider och att checka-in inför en flygresor behöver inte nödvändigtvis ske via en check-in disk (Meeker m. fl. 2009).

Konkurrensen som finns i dagens mobilmarknad har tvingat ledande företag att skapa mobiler med ny teknik till allt lägre priser. Pris, tillgänglighet och användbarhet har gjort att användning av internet via mobilen i världen närmar sig antalet användare av internet via en dator (Stanley 2009). Analytiker som förutspådde mobilen till att slå i marknaden har i dagsläget fått det bekräftat och förutspår att användare av mobilt internet i världen kommer att passera antalet användare av Desktop internet under år 2014 (Stanley 2009). Detta kräver från webbplatser att följa målgruppen användare och anpassa webbsidor utifrån de enheter som webbsidor ses från, vilket till stor del är från mobilen.

2.2 Desktopanvändare

Att mobilanvändare ökar för varje dag har till viss del ett samband med minskningen av antalet *desktopanvändare*. Men datorer har i dagsläget funktioner som gör det osannolikt för mobiler att helt kunna ersätta datorer. Funktioner som kräver administration via en dator och därför sker informationssökandet i webben via desktop.

Webbdesignen på en mobil är kompakt och uppfyller den nödvändigaste användbarheten. På desktop är designen mer informationsrikt, vilket ger en större inblick till webbsidans innehåll och bidrar till simpel navigering på webbsidan. Det gör att en användare beroende på komplexitet och säkerhet av handlingen väljer att utföra den via en desktop (Singhal 2009), än via mobilen. En sådan handling kan t.ex. vara bankärenden eller webbshopping. Om bankärendet gäller en översikt av saldo i kontot eller överföring mellan egna konton, anses mobilen vara en smidig enhet att använda. Däremot om handlingen innefattar att betala räkningar eller överföra stora summor pengar finns behovet av att utföra dessa handlingar via desktop. Det känns säkrare och en mer simpel navigering på webbsidan minskar risken av att göra misstag. Även webbshop faller i samma kategori, då användare väljer att söka information om produkten via mobilen, men väljer sedan att utföra köpet via desktop eller i affären (Singhal 2009). Detta tyder på att användare av mobilt- eller desktop internet inte är endera, utan är användare av båda, beroende på situation, miljö och kontext på webbsidan.

Det finns ingen grundlig faktum som tyder på att mobilt internet kommer att ta över all desktop internetanvändande, endast att de kan bli fler. Därför krävs att vyer kring webbutveckling för mobil och desktop utvidgas. För även om mobilanvändare blir fler, går det inte att förbise desktop användare.

3 Teori

I nedanstående kapitel beskrivs den teori som krävs för att upprätthålla ett responsivt tillvägagångssätt för både *desktop-first* och *mobile-first*. Först beskrivs verktygen mer ingående för att sedan övergå till hur dessa samverkar med responsive web design.

3.1 Verktyg

Inom webbdesignsutveckling används främst verktygen HTML, CSS och JavaScript, vilket är det verktyg som kommer att användas under den tekniska delen i examensarbetet.

3.1.1 HTML

HTML står för *Hypertext Markup Language* och är ett format som definierar strukturen och logiken på en webbsida. HTML beskriver strukturen genom att märka upp olika delar av sidan med hjälp av *taggar* som beskriver typen av ett element. En webbläsare läser av HTML-koden och kan på så sätt rendera sidan med önskvärd layout.

Det finns olika sorters taggar inom HTML, stycke, rubrik, tabeller, länkar, listor, sektioner är en av dessa. Inom webbutveckling har användning av taggen `<div>` blivit vanlig för att strukturera upp en sida i olika sektioner (Rana 2009), vartefter kan en sektion innehålla andra element.

```
<div>
    <h1> Mobile-first eller Desktop-first?</h1>
    <p>Frågan en webbutvecklare ställer sig inför skapandet av
        en responsiv webbsida
    </p>
</div>
```

I ovanstående HTML-kod definieras en sektion på webbsidan med `<div>` taggen. Inom sektionen finns det rubrik som definieras med taggen `<h1>` och ett stycke som definieras med taggen `<p>`. Taggarna i html-koden bildar tillsammans strukturen på sidan. Html-koden har ingen kontroll över utseendet för taggarna, endast strukturen. Utseendet sköts av CSS.

3.1.2 CSS

CSS står för *Cascading Style Sheets* och är ett språk som beskriver utseendet på en webbsida. Med hjälp av CSS-kod kan olika element i html-koden få ett speciellt utseende i form av storlek, färg och position. CSSen ser till att ge element egenskaper, vilka kan definieras

direkt i taggen t.ex `<p style="color:blue; width:50px">Frågan en webbutvecklare ställer sig inför skapandet av en responsiv webbsida</p>`. Det betyder att elementet `<p>` kommer att ha bredden 50px och texten kommer att vara blå. Att definiera utseendet på det sättet kallas för *inline-styling*. Det går även att i html-filen, definiera utseendet i `<head>` taggen, vilket kallas för *internal styling* (w3schools 2009). Det betyder att alla taggar i html filen delar egenskapen som är utsatt, i det här fallet att **alla** `<p>` taggar i koden har blå text och bredden 50px:

```
<head>
    <style>
    p {
    color:blue;
    width:50px;
    }
    </style>
</head>
```

Tredje sättet är att definiera det i en egen css-fil och länka till det från `<head>` taggen i html-koden. Det är det vanligaste av de tre sätten och kallas för *external styling*.

```
<link rel="stylesheet" type="text/css" href="main.css">
```

I css-kod definieras utseendet av olika taggar. Det finns även möjlighet att definiera samma tagg med olika utseenden, vilket görs genom att sätta *Klasser* och *IDs* för varje element. Ett element kan ha olika klasser eller ids, vilket definieras med ett utseende i css-koden. Det gör att varje tagg som delar klassen, även delar utseendet.

I html-filen:

```
<div id='main-container'>
    <p class='paragraf'>Mobile-first vs Desktop-first</p>
</div>
```

I css-filen:

```
#main-container {
    width:50%;
    background-color:grey;
}
.paragraf{
    font-size: 125%;
    color:black;
}
```

I ovanstående kod kommer sektionen få utseendet som definieras som ID: *main* i css-koden, det vill säga 50% bredd och bakgrundsfärgen grå. Paragrafen får utseendet som definieras som Klass: *paragraf*, vilket är svart text i storlek 125%.

Dessa tre olika sätt syns i en browser enligt prioritering, inline-styling, internal-styling och sist external-styling, det betyder att utseende som skrivs i inline-styling kommer att skriva över utseende som har definierats i internal- och external-styling. Oftast används external-styling i första hand för att utseendet inte behövs definieras flera gånger, allt är samlat i en fil och det går snabbare att ladda sidans utseende (w3schools 2009). Därmed hålls det ur en webbutvecklarens synvinkel en bra struktur som upprätthåller teknikerna för responsive web design, vilka beskrivs i nästa delkapitel.

3.1.3 JavaScript

JavaScript är ett scriptspråk som inom webbutveckling främst används för att hantera dynamiska funktioner för beteendet hos en webbsida, beteenden som skapas från klientsidan. JavaScript läser en användarens klick i webbsidan och kallar på funktioner som utför ändringar på t.ex innehåll eller utseende. Eftersom JavaScript är ett skriptspråk behövs ingen kompilering och koden kan skrivas direkt i html-filen.

För att JavaScript skall fungera korrekt krävs det dock att webbläsaren stödjer det. I vissa fall har webbläsaren den funktionen men har den avstängd (Ingmar 2012, s.13). Tidigare fanns problemet att mobilwebbläsare inte haft stöd för JavaScript, vilket har lett till att menyer och pop-up fönster inte har fungerat korrekt när sidan besökts via mobilen. Tekniken har utvecklats och numera klarar mobilwebbläsare av JavaScript. Det finns fortfarande en del problem när det kommer till mer komplicerade JavaScript funktioner och bibliotek som mobilwebbläsare inte kan hantera (quirksmode.org 2009). Vid webbsidor som har helt separata mobilsidor kan det vara en fördel då det väljs att inte läsa in all JavaScript som finns på desktopsidan. Med responsiva webbsidor används samma webbsida för varje enhet, vilket leder till att en mobilwebbläsaren laddar all JavaScript och kan försämra webbsidans respons. En webbutvecklare måste ha i åtanke och redan från början utveckla en hemsida vars JavaScript inte kommer i vägen för den responsiva webbsidans funktionalitet.

3.2 Responsive web design

Responsive web design är ett koncept som innebär att gränssnittet på en webbsida ändras beroende på skärmstorlek, således ges möjligheten att ha olika layout på en och samma webbsida anpassat efter en enhet. Konceptet definierades av Marcotte 2011 i en artikel kallad "*A List Apart*", som sedan blev en del av boken "*responsive web design*". I artikeln delade Marcotte med sig av teorier och praktiska exempel för att förklara begreppet.

Syftet med Responsive Web Design är att kunna rendera olika delar av sidan beroende på skärmstorlek för att ge en optimal vy för den enhet webbsidan ses från. Om tre element renderas bredvid varandra på en desktop sida, så vore det optimala för en mobilsida att kunna rendera elementen under varandra och även skala ner de till en rimlig storlek. På så sätt försvinner inte artiklarna från webbläsarfönstret eller tar för stor plats på skärmen.

Då endast layouten på sidan ändras behövs inga särskilda versioner av webbsidan för varje enhet, vilket gör det möjligt för en webbutvecklare att på ett enkelt sätt utföra en ändring i en fil, istället för att ändra varje fil för respektive version av webbsidan. Däremot krävs en flexibel grundlayout för att responsive web design skall fungera. Enligt Marcotte fås det genom tre grundtekniker, *Fluid Grid*, *Fluid Images* och *Media Queries* (2011), vilka kommer att beskrivas mer ingående härnäst.

3.2.1 Fluid Grid

På en webbsida kan storleken på element definieras på olika sätt. Ett vanligt sätt att definiera element är genom bredd och höjd i pixlar. När det definieras i pixlar betyder det att storleken på elementet är förutbestämd vare sig upplösning eller storlek på skärm. Vid tidigare skede informerade webbutvecklare användaren vilken upplösning som renderade sidan på bästa sätt och sedan var upp till användaren att ändra upplösning på skärmen för att få den önskade layouten på sidan (Marcotte 2011, s. 6). I dagsläget finns många olika skärmar och många olika alternativ till upplösning vilket gör en förutbestämd storlek icke optimalt.

Fluid Grid är en teknik vilken använder sig av procentsatser istället för pixlar vid definiering av storlek på element. När procentsatser används förstoras eller förminskas ett elements storlek relativt till webbläsarfönstrets höjd och bredd, vilket gör att sidan istället anpassas efter användaren. Om ett element skall täcka hela bredden på skärmen, sätts bredden till 100%, en fjärde del av skärmen, 25% osv. För att kunna bestämma storleken på ett element i procentform härleder Marcotte ett sätt vilket innebär att, i pixelstorlek, dividera elementets storlek med behållarens storlek. Det ger ett resultat i procent, där elementet storlek ändras relativt till hållarens bredd och höjd (Marcotte 2011, s. 23).

3.2.2 Fluid Images

Fluid Images bygger på samma princip som Fluid Grid, däremot är lösningen mer komplex då det innefattar skalning av bilder. Om en bild skalas på fel sätt leder detta till att bilden ser för utsträckt ut eller för intryckt ut. Om skalning inte görs kan en bild i princip ta upp all plats på webbsidan och även försvinna ut i kanterna. Inom responsive web design används Fluid Images, vilket innebär en flexibel hållare med en önskad storlek, innehållande en bild anpassad till hållaren.

Lösningen som tas upp av Marcotte är användningen av egenskapen **"max-width: 100%"** för bilder i webbsidan (Marcotte 2011, s. 45). Med egenskapen **"max-width:100%"** förstår webbläsaren att bilden skall skalas efter hållaren men endast till max av den originala storleken på bilden. Egenskapen **"width:100%"** innebär att bilden bara får anpassa sig helt efter hållaren, samtidigt som den får en normal skalning när storleken på webbläsaren minskar och hållaren krymper. Egenskapen **Max** i **"max-width"** innebär att bilden aldrig blir större än bildens verkliga storlek. Om *"max"* inte sätts, finns risk att bilden skalas ut och gör pixelkanter synliga. Däremot krävs en bild som redan från början är av önskad storlek. En för liten bild kommer den inte att förstöras då den som störst kan vara 100% av bildens ursprungliga storlek. Detta är ett sätt att ta använda sig av Fluid Images som fungerar för responsiva webbsidor. Det kan uppstå andra komplikationer i form av att bilden inte skalas enligt önskemål, vilket kan kräva andra lösningar. Grundtanken för att upprätthålla fluid images är att bilder skalas efter webbläsarfönstrets storlek.

3.2.3 Media Queries

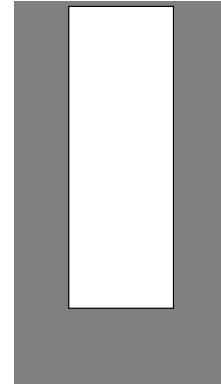
Fluid Grid och Fluid Images skapar tillsammans en del av en responsiv sida då storleken på element i sidan renderas utifrån storleken på skärmen. Minskningen eller förstoringen av elementen fungerar däremot endast till en viss gräns. Vid tillfällen då storleken på webbläsaren är liten finns risk att element kommer för nära varandra och skapar konflikt i layouten. Det leder till att element hamnar på felaktiga positioner och ger en vy som designmässigt ser förstörd ut. Blir storleken däremot för stor, finns risken att *max-width* på bilderna uppnås och stannar i storlek medan andra element följer förstoringen hos webbläsaren, blir för stora och skapar en osymmetri i layouten. Media Queries är en lösning som tillåter element att ha olika värden beroende på skärmstorlek (Marcotte 2011, s. 65).

Inom responsive web design är bredden och höjden viktiga egenskaper då de avgör hur mycket av en sida skall synas på webbläsarfönstret. Media queries är anrop som görs i CSS-filen, där antingen höjden eller bredden kollas för att utföra nödvändiga ändringar i layouten och upprätthålla en bra design. Tillsammans med Fluid Grid och Fluid Image skapar Media Queries responsive web design.

Vid användning av fluid grid och fluid image sätts egenskaper hos element i procentform, t.ex 50%. Med en centrerung på elementet, blir resultatet en ruta i mitten av skärmen som täcker 50% och har 25% utrymme på var sida. När webbläsarfönstret minskas till storleken av en mobilskärm anpassas elementet relativt till förändringen, då fluid grid tekniken används för bredden. Det ger oss nedanstående vyer, ena med webbläsarfönstret i storlek av en skrivbordsskärm och den andra i storlek av en mobilskärm:



Figur 2: Utifrån desktop



Figur 3: Utifrån mobilen

Denna design har endast ett element med egenskaperna:

```
div {  
    margin: 0 auto; //centrerad  
    width:50%; //bredd  
    height:500px; //höjd  
}
```

Designen fungerar bra för skrivbordsskärmar (*Figur 2*), då rutan är centrerad och symmetrisk. 25% av utrymmet på var sida av rutan gör designen läsvänlig och användbart då fokusering sker i mitten av skärmen där information samlas. Däremot inte lika optimalt när skärmen blir mindre (*Figur 3*). Rutan är fortfarande 50% av skärmen, vilket gör att designen blir kompakt i ett utrymme där förmedling av mycket information sker. 50% procent av skärmen går till spillo åt utrymme mellan rutan och kanter, vilket får rutan att se ihoptryckt ut. Därav en försämring på design och användbarhet när webbsidan ses från en mobilskärm.

Med media anrop i CSSen, kallat för media queries, tillåter vi att värden hos "div" elementet skrivs över när bredden på webbläsaren underskrider 380px, en bredd som förekommer hos mobilskrmar.

```
@media all and (max-width: 380px){  
  div {  
    margin:0;  
    width:100%;  
  }  
}
```

I media anropet kommer "*width*" och "*margin*" skrivas över när skärmstorleken är 380px eller mindre, "*height*" förblir detsamma. Utrymmet mellan kanterna och rutan ändras till 0 och bredden till 100%, vilket innebär att rutan täcker hela skärmbredden. Det leder till att sidan får andra designvärden när den ses från mobiler och andra enheter där skärmen har som störst 380px i bredden. Det ger en optimal lösning i både ett användbarhets- och designperspektiv, då mer plats ges åt information och gör att navigeringen blir simplare för en användare.



Figur 4: Resultatet vid användning av media queries.

På så sätt kan samma CSS-fil låta element ha olika värden på egenskaper beroende på webbläsarstorlek. Responsive web design fyller sin funktion genom att tillföra en användbar design oavsett enhet som används för att se webbsidan.

3.3 Mobile-first

Mobile-first metoden bygger på att webbsidan anpassas för mobilskärmen först, för att sedan med hjälp av media queries och en flexibel layout renderera elementen på webbsidan desto större webbläsarfönstret blir. På så sätt är grundlayouten designad för mobilen. Mobile-first anses som en rimlig utvecklingsmetod då hänsyn tas till antalet mobilanvändare och det behov som finns vid navigering och informationsintagelse från mobilskärmar. Det betyder att innehåll prioriteras då bristen på plats är ett faktum och fokus läggs på de delar som informationsmässigt är de viktigaste (Lande 2009). Det behöver inte nödvändigtvis betyda att hänsyn till design för mobilen inte tas i desktop-first, utan snarare att implementationen för mobilskärm sker vid ett senare tillfälle i utvecklingen än vid mobile-first.

Kodmässigt är grunden anpassad för webbsidans mobilvy. Layout och struktur är helt designat till mobil där förändringen sker vid ett media anrop som reagerar på förstoringen av webbläsarfönstret. Media query anropet ser till att skriva om värden hos valda element så att de anpassas till skrivbordsskärmen.

```
@media all and (min-width: 980px){  
    .main {  
        margin:0 auto;  
        width:50%;  
    }  
}
```

I fallet ovan sker ett anrop när bredden på webbsidan är som minst 980px, vilket gör att elementen med klassen main får "*margin*" och "*width*" överskrivs med nya värden. I de fall där mobiler inte klarar av att läsa media queries (Clark 2009), vilket i dagsläget är få, är detta en optimal lösning då grunden redan är skrivet för mobilen. Läsning av media queries krävs endast när webbsidan ses från en skrivbordsskärm, vilket de flesta webbläsare klarar av.

3.4 Desktop-first

Desktop-first metoden bygger på att utveckling av webbsidan sker för skrivbordsskärmen först och därefter rendererar sidan desto mindre webbläsarfönstret blir. Det behöver inte nödvändigtvis betyda att en färdig sida för desktop i efterhand designas om till mobil, utan tanken för mobil finns redan från början men implementeringen sker i synnerhet för skrivbordsskärmen först. Webbстор som i efterhand skapas till mobilen, brukar innebära en separat mobilside då refaktoreringen av kod i samband med att förvandla sidan till responsiv kan innebära en del komplikationer (Clark 2009).

Kodmässigt är grunden anpassad för skrivbordsskärmen. Det vill säga att media anrop sker

när skärmbredden når en minimum gräns. Där ser media queries till att skriva om värden på valda element för anpassa de efter enheter med mindre skärmar.

```
@media all and (max-width: 380px){  
  .main {  
    margin:0 ;  
    width:100%;  
  }  
}
```

I fallet ovan sker ett anrop när bredden är som max 380px. Anropet ger "*margin*" och "*width*" i klassen "*main*" nya värden, som är anpassade till mobilen. Desktop-first används oftast i samband med webbsidor som skall förmedla en upplevelse och uppnå ett design maximum när den ses från skrivbordskärmen. Upplevelsen från en mobilskärm är inte lika högt prioriterad, men bör innebära att webbsidan uppnår den nödvändigaste funktionaliteten.

4 Metod

För att kunna svara på frågeställningarna är tillvägagångssättet uppdelat i en praktisk och en teoretisk del. Den teoretiska delen är en litteraturstudie bestående av en analys av den erfarenhet som finns hos profiler inom webbutveckling och en analys av användningsområden för mobilt och desktop internet. Den praktiska delen består av en implementering för vardera metod gentemot samma responsiva webbsida. Tillsammans tillför tillvägagångssätten resultat som besvarar frågeställningarna. Den praktiska delen vilket innebär gränssnittsutveckling, är det som tar mest tid och vägt tyngst i arbetet. Litteraturstudien har använts för att besvara de områden som resultatet av implementationen inte har kunnat nå fram till, samt styrka de påståenden som samlats från implementationsfasen.

4.1 Litteraturstudie

I litteraturstudien har bland annat 13 källor i form av bloggar samlats och analyserats, där skribenterna består av gränssnittsutvecklare, interaktionsdesigners, användbarhets experter och profiler inom området. Dessa har kommit till användning för att få en bild av de problem som kan uppstå, samt samla synpunkter och erfarenhet om för- och nackdelar gentemot vardera metod. Tekniken är så pass ny att information kring metoderna mestadels finns i bloggar. Bloggar är en vanlig källa där profiler inom området delar med sig erfarenhet kring ny teknik, inte nödvändigtvis för att det skall ses som vetenskapligt, men för att förmedla sin åsikt. Analysen av skribenternas synpunkter används även för att styrka påståenden som har tagits från implementationsfasen av metoderna. Litteraturstudien består även av 3-5 artiklar och statistik som har analyserats för att få en klarhet till användningsområdena för mobilt och desktop internet. Det visade sig i första delen av litteraturstudien att när ett responsivt tillvägagångsätt utförs är användningsområdet en viktig aspekt att analysera då det kan vara en avgörande faktor för val av metod. Artiklarna och statistik har analyserats utifrån enheternas användningsområde beroende på webbsidans målgrupp, kontext och miljö.

4.1.1 Miljö

Med miljö menas den miljö som existerar hos användaren vid besök av en webbsida och det sammanhang för besöket. En miljö kan t.ex. vara kontor eller resande, vilket leder till att olika enheter används beroende på den plats användaren befinner sig på.

4.1.2 Målgrupp

I dagsläget är internetsurfandet utspritt i många olika åldrar, vilket även gör målgrupp till en viktig faktor då användningen av enheterna är olika bland olika åldrar. Det ger en bild av

hurdan internetanvändningen ser ut för det de åldersgrupp webbsidan är tänkt att inrikta sig mot.

4.1.3 Kontext

Kontext handlar om webbsidans innehåll och den information som vill förmedlas till användaren. Webbssidor har olika mycket innehåll och olika typer av information, vilka kräver olika stora ytor för att kunna förmedla informationen till användaren på bästa sätt. Därav är valet av prioriteringen för enhet beroende av webbsidans kontext och handlingen användaren skall genomföra på webbsidan.

4.2 Implementation av webbsida

Den praktiska och tekniska delen av examensarbetet utgörs av en implementation av en webbsida som konstrueras i både Mobile-first och Desktop-first. Syftet med implementationen är att jämföra vardera av metod genom att applicera dem för en och samma webbsida, med andra ord, nå samma mål och resultat med två olika lösningar. Innan implementeringen har en prototyp designats och ritats utifrån det ramar som anses rimliga för att webbsidan skall spegla ett riktigt gränssnittsprojekt. Prototypen skall även vara en webbsida vars besökare är både via mobilen och via desktop, vilket fås genom litteraturstudien. Sedan har utifrån en design för mobil och desktop, en webbsida i mobile-first och en webbsida i desktop-first skapats med hjälp av HTML-, CSS- och javascriptskod för att uppnå den gränssnittsfunktionalitet som eftersträvas, vilket i det här fallet är en responsiv webbsida. Under implementationens gång har en daglig rapport förts för varje metod, för att på så sätt kunna jämföra data gentemot samma period av implementeringsfasen för vardera av metoderna. Under implementationsfasen har kodstorlek på css-filen, responstid för webbsidorna samt implementeringstid använts som data att jämföras. Genom utförandet av en dagbok för respektive metod har även kommentarer antecknats och samlats in tillsammans med nedskriven implementationsstid. Kommentarererna beskriver svårigheter och speciella utmaningar för varje metod. Implementationstiden är mestadels för att jämföra totala tiden, samt tiden som har tagits för att göra grunden för webbsidan och webbsidan responsivt. Responstid och kodstorlek jämfördes efter implementationen, där kodstorlek jämfördes i både filstorlek och antal css rader. Antalet css rader har tagits med som data för att kunna jämföra hur mycket av elementens utseende har skrivits i grundkoden och sedan förändrats i media queries. Responstiden har mätts genom att utföra tre olika hämtningar av webbsidornas css-fil och html-fil i både desktop- och mobilvy, för att få fram den metod som laddar sidan snabbast för varje vy.

4.2.1 Prototyp

Prototypen är det första momentet i den praktiska delen. Rent designmässigt handlar det om att hitta en bra struktur på webbsidan så att den kan spegla ett verkligt gränssnittsprojekt. Designen och kontexten av webbsidan som implementerades bestämdes med hjälp av en litteraturstudie där syftet var att hitta en typ av webbsida där besökarna är både mobilanvändare och desktopanvändare. Anledningen till att besökarna skall bestå av både parter är för att det då speglar en svår situation för en gränssnittsutvecklare som måste på förhand välja en implementeringsmetod. Därför krävdes det av ritningarna att de uppfyllde de nödvändigaste kraven i form av struktur och design. När sidtypen valdes ut, jämfördes den med andra webbsidor av samma typ och ritades ut med hjälp av *photoshop*. Ritningarna avser en hel webbsida, det vill säga, mer än vad skärmen kan visa. I fall där menyer öppnas och stängs har ritningarna gjorts för vardera av fallen för att få en helhetsbild. Ritningarna har granskats av en gränssnittsutvecklare på arbetsplatsen, vilket har lett till att ritningarna, efter korrigering, har använts som designbas för kodningen.

4.2.2 Webbsidan i Desktop-first

Desktop-first är första implementeringsmetoden som utfördes utifrån ritningarna. Valet att implementera i desktop-first var slumpmässigt. Under implementeringen utvecklades webbsidan med en responsiv grund i form av fluid grids och fluid images. När webbsidan var klar för desktop, påbörjades utvecklingen med media queries för att få en mobilvy av de element som fanns i desktopvyn. I slutändan jämfördes gränssnittsutvecklingens resultat med skisserna, då skisserna är de mål webbsidan förväntas uppnå. Vid sidan av utvecklingen genomfördes en rapport som vid ett senare tillfälle jämfördes med rapporten för mobile-first. Webbsidan kodgranskades av gränssnittsutvecklare för att upprätthålla en bra kvalitet i arbetet.

4.2.3 Webbsidan i Mobile-first

Mobile-first är andra implementeringsmetoden och utvecklades efter desktop-first. Implementationen genomfördes direkt efter implementationen för desktop-first, då analysen av implementeringsfasen skulle göras efter att båda rapporterna för utförandet var slutförda. Under implementeringen utvecklades webbsidan med en responsiv grund i form av fluid grids och fluid images. När webbsidan var klar för mobilen, påbörjades utvecklingen av media queries för att få en desktopvy av de element som har utvecklats för mobilvyn. Webbsidan kodgranskades av gränssnittsutvecklare för att upprätthålla en bra kvalitet i arbetet.

5 Resultat

I detta kapitel har resultat samlats från litteraturstudien samt implementationen av prototypen. Resultatet från litteraturstudien är uppdelat i två delar. Ena består av samlade synpunkter kring mobile- och desktop-first utifrån webbutvecklare samt interaktionsdesigners perspektiv. Andra delen av litteraturstudien består av en studie kring användningsområden för mobilt och desktop internet, som visar hur dessa används i nuläget beroende på målgrupp, typ av webbsida samt den miljön där webbsidan används. Studien kring användningsområdet för mobilt och desktop internet har legat till grund för typen av webbsida som har använts som prototyp och implementerats i mobile- och desktop-first. Resultat från implementationen har samlats i form av responstid, kodgranskning och implementeringstid, vilket har jämförts mellan metoderna. Mer ingående vad resultatet har för betydelse diskuteras i nästkommande kapitel.

5.1 Litteraturstudie

I nedanstående del har resultat samlats från litteraturstudien som gjordes uppdelat i ”synpunkter angående mobile-first och desktop-first” samt ”användningsområden för mobilt och desktop internet”.

5.1.1 Samlade synpunkter kring Mobile-first och Desktop-first

Utifrån de 13 olika källor som har samlats och analyserats säger 11 att mobile-first kommer att vara, om inte redan är, nästa stora teknik inom responsiv web design. Den främsta anledningen som poängteras är användandet av mobilt internet som börjar bli så oerhört stor i utsträckning. Webbutvecklingen måste finna ett anpassningssätt för det behov som finns hos mobilanvändare utan att gå miste om användare för desktop, vilka har funnits under en längre tid. Anpassningen bör inriktas mot att vid ett tidigt skede ta sig över barriärer som förekommer hos mobilwebb. Mobilen har mindre skärmar, sämre prestanda och begränsad internethastighet (Zee 2012) och statistik visar att om en användare finner mobilvyn slö, finns även risken att desktopvyn tappar användarens intresse (zurbword 2012). Mobilanvändare kräver i dagsläget snabb respons från mobilsidor, då många aktiviteter sker i tidspressade situationer (Wroblewski 2010). Med desktop-first existerar risken att webbsidan blir stor, innehållande många element som även laddas i mobilvyn, oavsett ifall de visas eller inte. Det försämrar mobilvyns respons och skapar även en kompakt, komplicerad design (zurbword 2012) eller en nerskalad version av desktopvyn.

Av 13 källor påpekar 6 av dessa på funktioner hos mobilen som kräver en annan typ av fokus, då med dessa finns möjligheten att webbsidan tas till en helt ny nivå funktionsmässigt. Funktionerna är GPS, Touchscreen och Multitouch. Dessa tre funktioner kan anpassa en mobilsida till att fungera på användningsområden som i dagsläget inte fungerar lika bra för desktop, men som utvecklas åt det hållet (Dimestam 2012). Dessa funktioner är nya

tekniker vilka fortfarande utvecklas för att nå optimal funktionalitet. Ett tidigt fokus på dem ger en stabil grund för funktionaliteten hos mobilsidan, vilket gör mobile-first till ett bra alternativ (Wroblewski 2010, Chapman 2012).

Webbutveckling har sin grund hos desktopsidor, vilka har funnits i flera år. Flera källor påpekar detta i mån om att poängtera att ett säkert och välprövat alternativ ibland kan kännas som den bästa startpunkten (Houx 2013, Chan 2013). Mobile-first är en ny teknik som kräver ett nytt tänk i jämförelse med implementeringen av en desktopsida och kan därför ta längre tid att implementera än desktop-first, således kosta mer (Chan 2013, Pope 2012). Däremot anser alla förutom en källa (Houx 2013) att argumentet inte håller, då det leder till en ond cirkel där ny och bra teknik inte används eller utvecklas (Johnson 2013, Pope 2012). Webbutvecklare får även ha i åtanke att vanliga funktioner hos desktop som har existerat under en längre tid inte fungerar på mobilsidor (Foster 2012, Johnson 2013, Mellas 2011). Vid desktop-first finns risken att använda funktionaliteter, t.ex hover, vilka kan leda till komplikationer när mobilvyn börjar implementeras (Chan 2013). Med det menar källan att funktioner och innehåll kan av misstag försvinna utan att de har varit avsedda att tas bort (Chan 2013). *Exemplet med hover görs i samband med desktop-first, där en ruta med text i desktopvy blir större när muspekaren hålls över, rutan visar då all text istället för en liten del av texten. I implementeringen av mobilvyn går inte samma kod att använda då "hover" inte fungerar. Det leder till att elementet antingen är stor hela tiden (tar mycket plats) eller liten och endast visar en del av texten. Vid sen påkommenhet kräver detta att utvecklare får be interaktionsdesigners att finna en lösning så att texten får plats i mobilvyn eller att elementet får prioriteras om.*

En nackdel med mobile-first är att Internet Explorer 6, 7 och 8 inte stödjer media queries (Pope 2012, Hansson 2012) och där lösningen fås genom javascript som av prestanda skäl inte är bra för ett mobilgränssnitt (Foster 2012). Fördelen med desktop-first blir att media queries inte läses utifrån en desktopvy (Hansson 2012), utan det görs via mobilens webbläsare vilka de flesta klarar av tack vare den ständiga uppdateringen som görs för webbläsare i mobiler (Mellas 2011).

Av 13 källor var det 2 som ansåg att webben och tekniken för responsive web design inte var redo för mobile-first (Grigsby 2011, Houx 2013). Enligt en studie hade flertal mobile-first sidor granskats och nått slutsatsen att mobile-first sidor blev större än desktop (Grigsby 2011). Enkelheten med desktop-first gentemot mobile-first tas återigen upp med hänvisning till att desktop har funnits under en längre tid, vilket medför att det är simpelt för användbarhetsexperten att ta sig an, samt är en säker grund för gränssnittsutvecklare att utveckla responsivt ifrån (Houx 2013). Skapandet av en responsiv webbsida efterhand innebär mycket refaktorering med mobile-first (med antagandet att alla sidor som inte är responsiv är desktopsidor). Resterande källor säger inte emot påståendet (Hansson 2012), men syftar på att en webbsida i desktop-first har tendensen att få mobilvyn förmedla en känsla av en nödlösning snarare än en genomtänkt produkt (Johnson 2013, Chapman 2012). I det fallet anses en bättre lösning vara att påbörja en ny design för mobilt, istället för att från desktop designa om till en responsiv webbsida.

7 av 13 anser mobile-first vara en optimal lösning då den enkla principen säger att om element får plats i mobilvyn får de alltid plats i desktopvyn, vilket inte gäller tvärtom. Med desktop-first kan element av misstag försvinna vid nedskallning till mindre skärmar då de anses vara överflödiga och inte får plats (Bull 2012, Foster 2012). Dessa hänvisar även till att webbdesign har tagit ett nytt steg mot användbarhet där webbsidan skall vara simpel för användaren att förstå och innehålla relevant information snarare än att den skall fånga användaren genom "överflödig" design (Bull 2012).

3 källor anser att användningsområde för webblösningen är en viktig aspekt att analysera inför implementeringen av den responsiva webbsidan (Hansson 2012, Pope 2012, Johnson 2013). Fördelarna i de flesta källor bygger på att en aning om målgruppen i stort existerar (zurbword 2012). En vanlig siffra som förekommer är att 25 procent av USA's befolkning använder sig endast av mobilt internet och där ökningen som sker för användning av mobilt internet tyder på att mobile-first är ett bra tillvägagångsätt till en responsiv webbplats. Men 25 procent av användare som enbart använder mobilt internet tyder även på att 75 procent antingen använder båda eller endast desktop och är därför inte tillräckligt övertygande (Pope 2012). Källorna menar då att en bra riktlinje till val av metod är att analysera användarna som webbplatsen kommer att inrikta sig mot (Hansson 2012).

5.1.1.1 Nämnda för- och nackdelar

Desktop-first

Fördelar	Nackdelar
Går fort att redesigna en existerande desktopsida till responsive, om elementen är definerade i procentform(fluid grid och fluid images).	Mycket kod som inte används läses in när sidan ses från mobilen, leder till sämre respons när sidan ses i mobilvy.
Enkelt koncept att förstå sig på, finns mycket erfarenhet från både kund, designers och utvecklare.	Lätt att funktioner för desktop försvinner i mobilt, kan leda till att mobilvyn tvingas redesignas och element får prioriteras om.
Fokus på enheten den större delen av befolkningen använder. Även om antalet mobilanvändare växer består antalet som använder desktop.	Mycket fokus på desktop leder till nerskalad mobilvy design- och funktionsmässigt, eftersom mobilvyn har fler begränsningar.
Bra för webbsidor om internet explorer 6,7 och 8 används, dessa kan inte läsa media queries, desktop blir då kodgrunden och media queries läses endast av mobilwebbläsaren.	Mindre fokus på speciella funktioner för mobilwebb t.ex hardware acceleration, touch, location awareness.
Säker punkt att utveckla responsivt ifrån, mer erfarenhet leder till mindre implementeringstid således blir kostnaden mindre.	All innehåll från desktopvyn får inte alltid plats i mobilvyn, samma element på en stor yta får inte alltid plats på en mindre yta.
	Problemen som finns hos mobilvy, som prestanda och sämre bandbredd, åtgärdas i efterhand.

Tabell 1: Desktop-first, nackdelar och fördelar som samlades i litteraturstudien.

Mobile-first

Fördelar	Nackdelar
Användbarhet i fokus , att börja i en begränsad yta tvingar fram en design där det nödvändigaste prioriteras.	Ny teknik, nya utmaningar , kan ta längre tid att implementera.
Fokus på innehåll , innehåll prioriteras först, sedan presentation och sist animation.	Internet explorer 6,7,8 klarar inte av media queries.
Skapar anpassningsbara och framtidsvänliga hemsidor , varje enhet får ut det bästa från en och samma webbsida.	Liten skärm att påbörja designen för , svårt att få en helhetsbild för hur desktopvyn skall renderas.
Robust grund för arbetssätt och design , implementering sker där det viktigaste är i centrum och allt byggs utifrån det.	Komplicerat om det redesignas från en existerande desktopsida.
Robust grund för komplikationer , svårigheter tas i början (prestanda, design, prioritering).	Svårigheter möts vid ett tidigt skede , kan uppfattas som trist och jobbigt.
Läser in mindre kod i mobilvyn , behöver inte läsa in allt för desktop, utan bara det för mobilen.	Svår koncept att sälja , inte lika populärt bland kunder.
Anpassning efter dagens marknad , användningen av mobilt internet ökar för varje dag.	Simpelt är svårt, prioritering är svårt , speciellt med olika investerare som tidigt vill få in varumärket i en produkt.
Fokuserar på kontext som fungerar för plats, tid och sociala egenskaper , situationer då snabb respons krävs.	
Tidig fokus på speciella funktioner för mobilen , multi-touch, gps, hardware acceleration, location awareness.	
Allt från mobilvyn tas med i desktopvyn , samma element i en liten yta får plats i en stor yta.	
Känns naturligt att lägga till desto större fönstret blir , gentemot att ta bort desto mindre skärmen blir.	

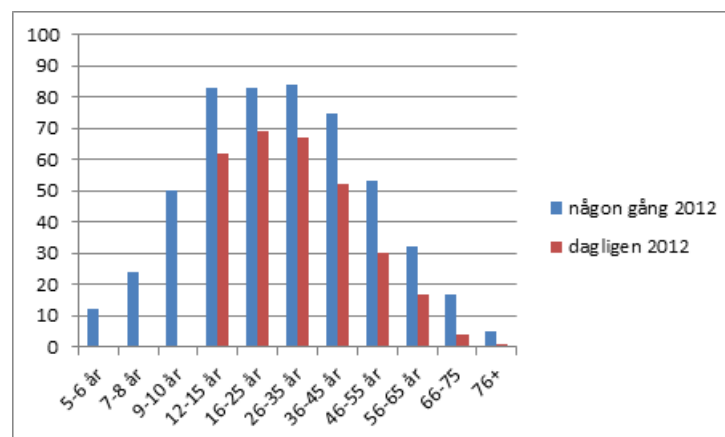
Tabell 2: Mobile-first, nackdelar och fördelar som samlades i litteraturstudien.

5.1.2 Användningsområde

Mobil- och desktopsidor har olika besökare beroende på typ av webbsida och under vilka förhållanden den ses från. Även åldern spelar roll då det visar sig i statistik att ungdomar är de största användarna av mobilt internet (Findahl 2012, s.24). I Sverige utförs varje år en statistikredovisning på svenskar och deras internetanvändning (Findahl 2012). Trots distansen ser användningen av mobilt internet ungefär likadan ut i Sverige som i USA. I detta kapitel visas resultat i en litteraturstudie där fokus läggs på användning av webbsidor beroende på användarnas åldersgrupp, kontext på webbsidan samt miljön webbsidan ses från.

5.1.2.1 Målgrupp

I de vetenskapliga artiklar som togs med i litteraturstudien gällande användning av mobilt internet har dessa inriktat sig till användare där målgruppen är från 19 år uppåt, med en medelålder mellan 20-29 år. Detta tyder på att en stor del av användarna för mobilt internet består av nämnda målgrupp. I statistik som utförts för att ta reda på svenskarna dagliga användning av mobilt internet sträcker sig lägsta ålder ner till 12 år. Trots detta stämmer målgruppen som har valts i de vetenskapliga artiklar bra då statistik visar att den stora målgruppen i Sverige av mobilt internet består av användare mellan 16-25 år där 69% av målgruppen använder sig av mobilt internet dagligen. Tätt efter är målgruppen från 26-35 år med 68% och 12-15 år med 67%. Den stora förändringen sker inte förens målgrupp 46-55 år då endast 30% använder sig av mobilt internet dagligen (Findahl 2012, s. 24). Lägst är målgruppen från 76 år uppåt som ligger på 1% men där internet användningen i överhuvudtaget inte är stor.



Figur 5: Andel procent av befolkningen som använder sig av mobilt internet, statistik visar att den stora målgruppen är mellan 12-45 år gamla (Findahl 2012, s. 24).

5.1.2.2 Kontext

Med kontext menas sidans typiska innehåll och huvudsyftet för webbsidan. Beroende på webbsidans kontext kan användarna finna det simplare att använda antingen desktop, mobilen eller den enhet närmast till hands. De vetenskapliga artiklar som har analyserats i litteraturstudien har alla gemensamt att mobilt internet används flitigt för:

- Sköta e-post
- Sociala nätverk
- Kolla nyheter, väder
- Söka information

Andel procent av användarna skiljer sig beroende på artikel och statistik, då antalet användare för studierna har varit olika. Analysen visar ändå vad fokus läggs på när mobilt internet används. I Sverige uppskattas 50% använda sig av mobilt internet för att sköta e-post, 43% för socialt nätverk, 40% för att kolla vädret, nyheter och 18% för att söka fakta (Findahl 2012, s. 25). I en annan studie där användningen av mobilt internet analyserades med hjälp av 109 deltagare visade den att 70% av deltagarna använder mobilt internet för att kolla nyheter, väder och söka fakta, 13% av deltagarna använder mobilt internet för sociala nätverk och 17% för att kolla på film och lyssna på musik (Schmiedl m. fl. 2009). I en liknande studie där 18 aktiva mobilanvändare studerades varje dag under en 4 veckors period, bestod den dagliga användningen av mobilt internet 27% för sociala nätverk, 24% för koll på nyheter och vädret, 18.8% för email, 14.9% för vanlig surf, 10% för mobilsök och 5.1% för kartor och navigation (Church m. fl. 2011).

För desktop anses dessa fyra områden vara minst lika populära. I en studie som gjordes hos The Harris Polls Research (Shannon-Missal 2013) samlades 2400 vuxna varav 991 mobilt internet användare för att se hur enheterna används i jämförelse beroende på kontext. I studien hamnade *Sköta e-post*, *sociala nätverk* och *söka information* längst upp som de typer av hemsida vars desktopanvändare är ungefär lika stora om mobilanvändare.

	Dator(desktop/laptop)	Mobil(Smartphone)
Sköta E-post	90%	72%
Sociala Nätverk	69%	64%
Söka information	81%	45%
Kolla nyheter, väder	44.2%	52.1%

Tabell 3: Områden i webben där användarna utnyttjar både mobil och desktop.

I studien ingick inte läsning av nyheter, men i Q1 Research report för nyhetsläsning (Fidler 2013) visades det sig, med undantag från målgruppen 65 år uppåt, att majoriteten av

läsarna(59% till 77% i varje målgrupp) använder både mobil och desktop för att läsa nyheter. Studien visar även att nyhetsläsare från mobilen numera är större än läsarna från desktop.

I The HarrisPole Research fanns även siffror för områden där användaren föredrar en enhet mer än den andra.

	Dator(desktop/laptop)	Mobil(Smartphone)
Fylla enkäter	86%	24%
Handla produkter	78%	23%
Navigation/Kartor	56%	73%

Tabell 4: Områden i webben där användarna föredrar en enhet mer än de andra.

5.1.2.3 Miljö

En användares miljö har förändras tack vare mobilens handburna storlek. Det har gjort att miljön inte är längre är den vanliga sederliga hemma eller på kontor, utan kan vara i vilket sammanhang som helst och även vart som helst. Ett vanligt antagande är att mobilen endast används vid rörliga situationer (Wroblewski 2011, s.12). Vilket vetenskapliga artiklar och statistik visar tvärtemot (Church m. fl. 2011, Wroblewski 2011).

Författaren av "*mobile first*" hänvisar(Wroblewski 2011, s.12) till en enkätundersökning vilket visar användningen av internet genom smartphones:

- 84% använder det hemma.
- 80% använder det under diversa dötid genom dagen.
- 74% använder det i kötid och i väntan på möten.
- 69% använder det under shopping.
- 64% använder det på jobbet.
- 62% använder det under TV tittande.
- 47% använder det under pendling.

En stor andel av smartphonebärare använder sig av mobilt internet i hemmet, vilket styrks av fler vetenskapliga artiklar (Church m. fl. 2011, Wroblewski 2011). Studierna pekar på bekvämligheten som existerar vid användande av mobilen istället för desktop. Mobilen tillåter 1 minuts intervaller för användning av internet vid brådska tillfällen, vilket kan anses vara omständliga via desktop (Church m. fl. 2011). I studien där 18 deltagare skulle föra dagböcker (Church m. fl. 2011) under en period av 4 veckor var över 70% av inläggen om aktiviteter gjorda vid ett stationärt perspektiv, det vill säga, antingen på jobbet eller i hemmet. Bara 17% av inläggen handlade om aktiviteter som gjordes vid resor, utomhus

eller vid pendling.

- 49,6% av inläggen handlade om aktiviteter i hemmet.
- 21,8% av inläggen handlade om aktiviteter på jobbet/universitetet.
- 11,4% av inläggen handlade om aktiviteter inomhus.
- 9,6% av inläggen handlade om aktiviteter under pendling eller transit.
- 7,2% av inläggen handlade om aktiviteter utomhus.
- 0,5% av inläggen handlade om aktiviteter under resa utomlands.

5.2 Implementation

5.2.1 Prototyp

Efter resultatet av litteraturstudien blev prototypen en nyhetssida. Grunden till valet gjordes med hänvisning till resultatet från litteraturstudien där statistik visar att andelen desktop- och mobilbesökare hos nyhetssidor är jämnt fördelat och även en av de största inom mobilt internet. För att skapa sidan gjordes en jämförelse mellan stora nyhetssidor så som aftenbladet.se, metro.co.uk, globalnews.ca och time.com, för att upprätthålla typiska element som finns hos nyhetssidor. Allt som finns i ena vyn finns även i den andra.

5.2.1.1 Banner

Banner visar tydligt sidan som besöks och innehåller namnet och en slogan som beskriver typen av webbsida. Både i mobilt och i desktop går dessa inte att undgå för att det skapar ett igenkännande hos användaren. Däremot är bannern mindre på mobilsidan i förhållande till artiklarna tillskillnad från desktopvyn (*figur 6*), för att undvika att bannern ta för stor plats på mobilskrmen (*figur 7*).



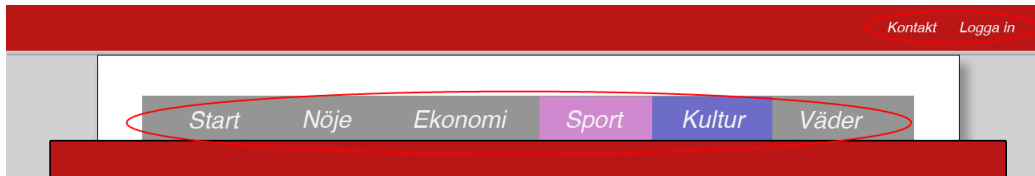
Figur 6: Banner för desktopvyn.



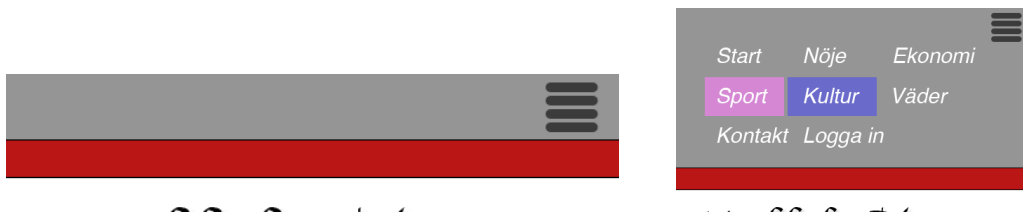
Figur 7: Banner för mobilvyn.

5.2.1.2 Menyer

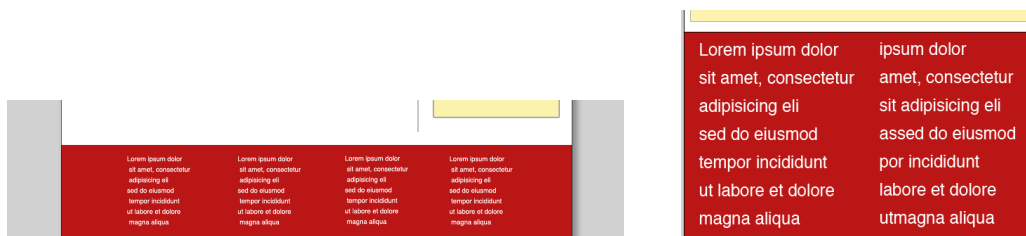
Nyhetssidorna som användes som referens för prototypen hade alla minst tre olika sorters menyer. En huvudmeny, en mindre meny bestående av typiska ”skapa konto, logga in” länkar(*figur 8*), samt en nedre meny längst ner på sidan med diverse länkar(*figur 10*). I mobilvyn behölls nedersta menyn i en ihopsamlad version och de två översta sattes ihop och gömdes under en menyknapp för att spara plats på mobilskärmen(*figur 9*).



Figur 8: Huvudmenyn i desktopvy.



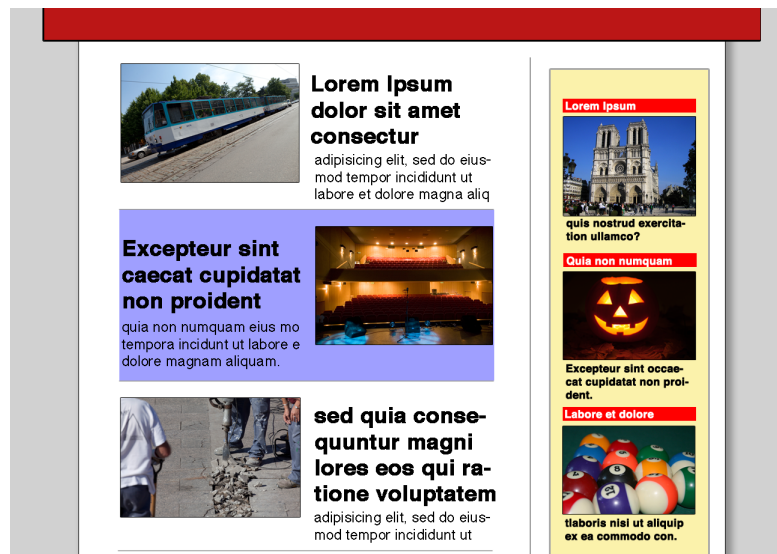
Figur 9: Huvudmenyn i mobilvyn.



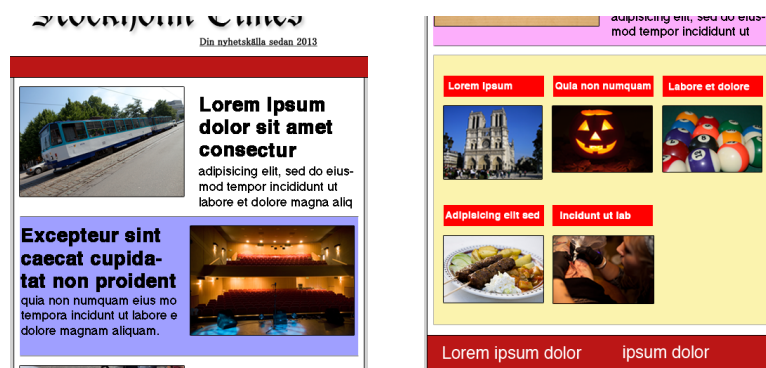
Figur 10: Nedremenyn i desktopvyn och mobilvyn.

5.2.1.3 Artiklar och Annonser

Artiklarna och annonserna prioriterades ungefär lika högt i desktopvyn (*figur 11*) för att annonser anses vara en viktig faktor till nyhetssidors intäkter. Dessvärre inte samma prioritering i mobilläge, då ytan är mindre och artiklarna är huvudelementen på nyhetssidan. Därför hamnade annonserna i mobilvyn efter artiklarna (*figur 12*). Artiklarna har alla samma storlek men skiftar i plats på bild och text för att skapa en mer levande känsla hos webbsidan. Annonsernas design härstammar från *aftonbladet.se*, där färgen på bakgrunden samt texten får läsaren att förstå att det är annonser.



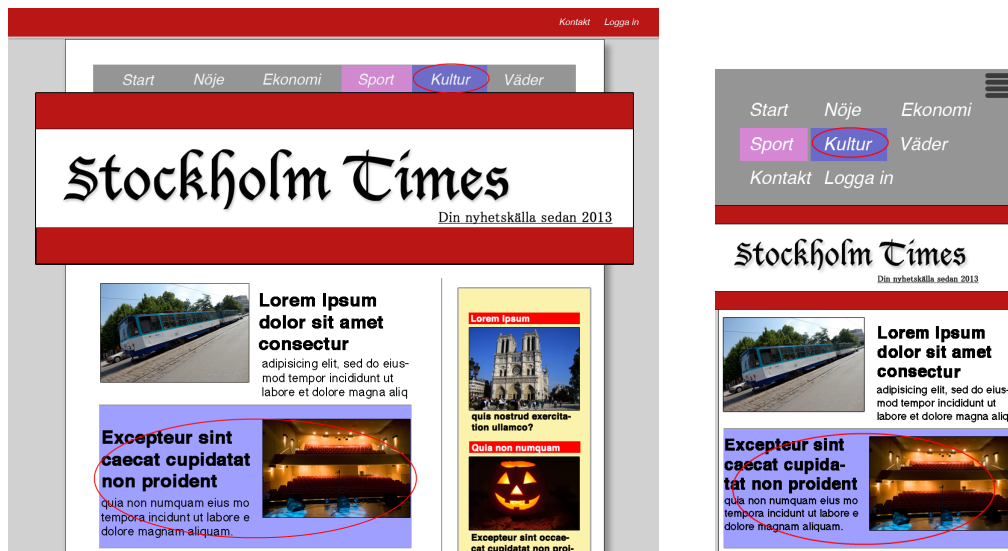
Figur 11: Artiklar och annonser i desktopvyn.



Figur 12: Artiklar och annonser i mobilvyn.

5.2.1.4 Färgmarkeringar

Färgmarkeringen görs i webbsidan för att markera olika typer av artiklar. På nyhetssidor görs detta för att markera artiklar som tillhör en annan sektion av tidningen, t.ex sport, ekonomi, kultur osv. I prototypen markeras sport och kultur med en rosa och blå färg vilken även blir färgen för respektive länk (figur 13).



Figur 13: Blå färgmarkering för kultursektionen.

5.2.2 Responstid

Nedan visas tre olika hämtningar av bägge webbsidor, både i desktopmiljö samt mobilmiljö. Värdena som har antecknats är responstiden för hämtning av HTML-filen samt CSS-filen. Sambandet mellan de olika hämtningar visar att mobilvyn laddas snabbare i mobile-first och desktopvyn laddas snabbare i desktop-first.

Hämtning 1

	.html mobilvy	.css mobilvy	.html desktopvy	.css desktopvy
Desktop-first	73ms	36ms	36ms	21ms
Mobile-first	36ms	19ms	39ms	23ms

Tabell 5: Värden från första hämtningen av prototypsidan. Tabellen visar tydligt snabbare responstid för mobilvyn med mobile-first och snabbare responstid för desktopvyn med desktop-first. Anmärkningsvärt är värdena för laddningen av css-filen i mobilvyn, där det tar mobile-first 19ms, men desktop-first 36ms.

Hämtning 2

	.html mobilvy	.css mobilvy	.html desktopvy	.css desktopvy
Desktop-first	19ms	22ms	46ms	20ms
Mobile-first	33ms	20ms	22ms	22ms

Tabell 6: Värden från andra hämtningen av prototypsidan. Ungefär samma resultat som tabell 5, men med mindre skillnad mellan responstiderna. Tabellen visar ändå en snabbare hämtning av css-filen för mobile-first i mobilvyn respektive desktop-first i desktopvyn.

Hämtning 3

	.html mobilvy	.css mobilvy	.html desktopvy	.css desktopvy
Desktop-first	37ms	22ms	20ms	19ms
Mobile-first	16ms	18ms	32ms	20ms

Tabell 7: Värden för tredje hämtningen av prototypsidan. Hämtning av css i mobilvyn är snabbare för mobile-first med 4ms skillnad, däremot är css hämtningen i desktopvyn snabbare för desktop-first, men med bara 1ms i skillnad.

Hämtningarna visar tydligt att css-filen laddas snabbare för mobile-first i mobilvyn och snabbare för desktop-first i desktopvyn. Gällande hämtning av html-filen följer det samma mönster förutom i en av hämtningarna (*tabell 6*). Där är resultatet tvärtom från resultatet i tabell 5 och 6. Alla hämtningar har gjorts från en stationär dator med en emulator som renderar om webbläsarens storlek för att passa vald enhet. Utförligare analys angående responstiden diskuteras i diskussionavsnittet.

5.2.3 Kodstorlek

Tabellen visar storleken på css-filen i både antal rader och filstorlek. Den visar att CSSen för mobil-first är större än CSSen för desktop-first i filstorlek. Däremot har den i desktop-first fler rader än den i mobile-first. I mobile-first är uppdelningen i CSSen mellan rader i grundkod och rader i media queries, 193 mot 180. I den för desktop-first består den större delen av rader i grundkoden, där är uppdelningen 221 mot 160.

main.css	Desktop-first	Mobile-first
Antal rader i grundkod	221st	193st
Antal rader i media queries	160st	180st
Totalt antal rader	381st	373st
Filstorlek	5.2kb	5.6kb

Tabell 8: Värden från main.css.

5.2.4 Implementeringstid

Implementeringstiden redovisas utifrån dagboken som skrevs under implementationsfasen. Resultatet visar skillnaden mellan implementeringstiden för grundsidan och tiden som krävdes för att göra webbsidan responsiv med hjälp av media queries. I mobile-first tog grundsidan mindre tid att implementera än att göra webbsidan responsiv. I desktop-first tog det däremot längre tid att implementera grundsidan. Desktop-first tog totalt längre tid att implementera än mobile-first. Ta i beaktande att implementeringen gjordes av en junior inom området gränssnittsutveckling, samt att desktop-first implementerades före mobile-first.

	Desktop-first	Mobile-first
Tid för grundsidan	14.5h	6.5h
Tid för responsive	7.5h	10h
Totalt tid	22h	16.5h

Tabell 9: Värden på implementationstid, hämtade från dagboken.

5.2.5 Kommentarer från dagboken

Desktop-first

"Mycket småpill i början, många element att finslipa, göra fina och sättas på plats."

"Oerhört lätt att göras till responsive så länge fluid grid och fluid layout används från början. Allt renderas av sig självt och så fort det ser konstigt ut, modifieras CSSen i media queries."

"Lång tid i början, i jämförelse med responsive delen. Förvånad över att responsivedelen gick så smidigt. Endast lite småpill lite här och där för att fixa storleken på text, margins och paddings. Det mesta hade fixats under desktoputseendet."

"Inte så många media queries, sidan ser bra ut mellan media queries. Mycket kod i grunden som är avsedd för desktop, inte lika mycket i media queries."

Sammanfattning: Skapandet av grundsidan var det som tog mest tid, då det innebar många element att designa och positionera på rätt plats. Den responsiva delen gick smidigt till då media queries, fluid grid och fluid images enkelt renderar om elementen till att vara mindre och positioneras under varandra i en kolumn anpassad till mobilskärmen.

Mobile-first

"Enkelt sätt att påbörja webbsidan, följer skissen uppifrån neråt, går inte att tappat bort sig åt kanterna, utan elementen är så pass små att det känns som att de finns i en smal led att följa."

"Behövs mycket småpill mellan media queries. Media queries kommer i början hyfsat tät inpå varandra enbart för att fixa små fel som margins och padding så att elementen håller sig på ett och samma ställe. Samt för förstoring av text, som är liten i början och måste förstöras i media queries då det inte kan använda sig av fluid grid."

"Gäller att hitta en struktur redan från början mellan skisserna. Det blir några refaktoreringar för att få webbsidan, då bredden är 768px, att se perfekt ut, eftersom mycket ändras just under den vyn."

"Från 768px till desktop-size gick väldigt smidigt."

"Väldigt lätt i början, enkel struktur att lägga upp för mobilen. Responsivt höll det till 480px, till och med 768px var det mycket att fixa med hjälp av media queries, då många element går från att vara under varandra till att vara bredvid varandra."

"Viktigt att planera och lägga en struktur på hur den ska övergå till desktop. Mycket kod i media queries sektionen, lite kod i grunden för mobilvyn. Ser bra ut mellan mobilvyn och desktopvyn."

Sammanfattning: Skapandet av grundsidan gick fort, var stabil och väl strukturerad. Däremot krävdes det fler media queries i början av den responsiva delen då mycket förändras tidigt. Det är viktigt att lägga upp en bra planering till hur webbsidan skall övergå till desktopvyn, då element övergår till att positioneras utspjutt över webbplatsen.

6 Diskussion

6.1 Mobile- vs Desktop-first

Mobile-first kontra desktop-first har varit den stora frågan i arbetet. Efter litteraturstudien och implementationen har tydligheten i metodernas områden blivit allt klarare och har under arbetets gång även blivit mer populärt bland bloggar. I resultatet för litteraturstudien samlades synpunkter från 13 olika källor, vilka inläggsdatum för de flesta inte är mer än ett år gamla. Av dessa källor är det 11 som förespråkar för mobile-first, en som föredrar desktop-first och en som anser att tekniken är för ny för att förorda. Implementationsfasen bekräftar mycket av det som har sagts i litteraturstudien, både till fördel för desktop- och mobile-first. Men i slutändan har implementationen av webbsidorna visat sig vara kodmässigt relativt lika. Mobile-first använder sig av min-width, medan desktop-first använder sig av max-width, mobile-first ser till att rendera om elementen när skärmen blir större och desktop-first när skärmen blir mindre. Mycket fokus har lagts på metoderna var för sig, när ett större problem visar sig vara skillnaden mellan mobilen och desktopens kapacitet. Mobiler har sämre prestanda, mindre skärm och bandbredds begränsningar. Begränsningar som inte existerar i samma utsträckning hos datorer. Dagens mobiler har kommit långt i utvecklingen men kan inte mäta sig med datorn. Det visade sig att arbetet kom att även handla om vilken metod **passar** bäst i dagens teknik och inte endast vilken metod som **är** bäst rent generellt. Dagens internet användning utgörs mestadels ifrån mobiler och datorer, där mobilen är den enhet som presterar sämre.

Mobile-first tar sig an problem som mobilen har och ser till att det utför det simplaste och läser in så lite som möjligt av webbsidan, med andra ord endast den kod som är avsedd för mobilen. Det finns det ingen metod som tillför samma sorts fördel för båda enheter, utan mobile-first kommer att behöva läsa in mer när webbsidan ses från skrivbordskärmen då den läser allt som krävs för mobil och även det som skall ändras för desktop. Desktop-first läser in det nödvändigaste för skrivbordskärmen, men när webbsidan ses från en mobil, kommer den att läsa in allt för desktop och alla ändringar för att webbsidan skall passa mobilen. Det betyder att mobil-first ser till att mindre läses in för den enhet vars prestanda är låg och mer för den enhet vars prestanda är hög och har kapacitet för att hantera det.

I litteraturstudien nämndes att mobile-first bidrar till en robust grund för komplikationer som prestanda, design, och prioritering av element. Prestanda visas nedan i diskussionen genom responstid och kodanalys, men design och prioritering är lite svårare att visa. I arbetet designades både mobilvyn och desktopvyn samtidigt, sida vid sida och vid ett tidigt skede bestämdes att båda vyerna skulle innehålla samma funktioner. Med motivet att användarna skall ha samma möjlighet oberoende på enhet webbsidan ses från. I litteraturstudien ses detta som ett tillvägagångssätt som existerar i mobile-first där meningen istället översätts till *"Får inte någonting plats i mobilvyn, så ska den inte heller få finnas i desktopvyn"*. Detta används för att undvika att funktionalitet endast finns på en av enheterna. Anledningen till att det är ett mobile-first princip är för att det är hos mobilen element har svårigheter att få plats, på grund av den begränsade skärmstorleken. Prioriteringen i mobilvyn sker då vid

ett tidigt skede och allt som får plats i mobilskärmen kommer att få plats i en större vy. I desktop-first gäller inte samma princip då element i en större skärm inte alltid får plats i en mindre yta. Det leder till att prioriteringen kommer i efterhand och sker endast för mobilvyn. Mobilvyn kommer då att sakna funktionalitet som finns hos desktopvyn och blir istället en komplement till desktopsidan. En användare som tidigare har varit på desktopsidan kommer att sakna dessa funktioner i mobilvyn och då har webbsidan plötsligt tappat sitt syfte för en mobilanvändare. I arbetet dök aldrig problemet upp för att principen "*om inte ett element fick plats i mobildesignen så får den inte finnas i desktopdesignen*" följdes från start.

Responstid, kodanalys, implementerings tid och skärmstorlek är en av de större faktorer som styrker påståenden tagna från litteraturstudien, nedan beskrivs och analyseras dessa i synnerhet till de praktiska och teoretiska tillvägagångssätt.

6.1.1 Responstid

Responstid är den tid det tar mellan att en begäran har lagts till dess att begäran har utförts. Det vill säga i webbsammanhang den tid det tar att hämta och visa elementen på webbsidan. De tre hämtningar visar klart att när webbsidan ses utifrån ett mobilperspektiv hämtas css-värden för att rita ut elementen snabbare med mobile-first än med desktop-first (tabell 5, 6, 7). När webbsidan ses utifrån ett desktopperspektiv hämtas css värdena snabbare med desktop-first än med mobile-first. Det vill säga, mobil-first ger bra siffror för hämtningar från mobilvyn och desktop-first bra siffror för hämtningar från desktopvyn, båda lika bra i sitt område om metoderna jämförs strikt gentemot varandra. Att metoderna är bäst för var sin vy är förståeligt eftersom mobile-first i mobilvyn endast läser grundkoden utan att behöva ändra något värde i media queries, desktop fungerar bra i desktopvyn för att den endast läser grundkoden för desktopvyn. Men som tidigare poängterats i diskussionen handlar det inte om metoderna var för sig utan enheterna webbsidan ses från. Där gynnar responstiderna för mobile-first den enhet med sämre prestanda vilket desktop-first inte gör. Båda responstiderna för mobil-first och desktop-first är hämtade från en dator där en emulator som minskar webbläsaren till en iPhones storlek har använts. Skillnaden mellan responstiderna är låga, ibland endast 2 millisekunder, för att prestandan och bandbredden som har använts för att hämta webbsidorna avser datorns.

Ändå visar responstiden på en skillnad, men i verkligheten är inte omständigheterna likadana som i testfallen. En desktop-first sida som ses i mobilvyn tog enligt en av hämtningar 36ms och desktopvyn 21ms. 21ms är relativt snabbt, vilket även 36ms är. När desktop-first sidan ses på en skrivbordskärm kommer hämtningen av css-en ta ungefär 21ms, men när vi ser en desktop-first sida på mobilen kommer den inte nödvändigtvis att ta 36ms som det står i tabellen (tabell 5). Detta för att mobilen inte har samma prestanda och bandbandsmöjligheter som datorn som användes för hämtningar har. Skillnaden mellan desktop-first och mobile-first blir inte längre 2ms, utan måste multipliceras med faktorerna, sämre prestanda och bredband. Det leder till att skillnaden mellan metoderna blir oerhört mycket

större än vad som visas i tabellen för hämtningarna av css-filen. Mobile-first visar genom tabellerna (*tabell 5, 6, 7*) att enheter där webbsidan ses i mobilvy inte behöva ladda mer än vad som behövs för att visa webbsidan och låter de enheter med bättre omständigheter ta på sig att hämta mer än nödvändigt.

Förutom att hämtningarna är gjorda med en dator så måste prototypen ha i åtanke då den egentligen bara speglar skalet av en fulländad webbsida. Det vill säga att hämtningarna som görs tillhör en webbsida som är relativt liten i jämförelse med en normal webbsida. Återigen är 2ms i skillnad mellan metoderna i en hämtning av mobilvyn små i jämförelse med vad skillnaden skulle vara om prototypen hade varit för ett större projekt.

Resultat av responstiden visar tydligt de punkter som diskuterats i litteraturstudien. Mobile-first ger en robust webbsida för mobilen som fokuserar i första hand på situationer hos användaren där snabb respons krävs. Tack vare prestandan hos dagens datorer utelämnar inte mobile-first desktop, utan låter den snarare ta hand om den större lasten vilken den har kapacitet för.

6.1.2 Kodanalys

6.1.2.1 Storlek

Kodstorleken spelar en stor roll till resultatet av responstiden. Beroende på metod läser webbsidorna olika mycket av css-koden för att renderera sidan beroende på webbläsarens bredd. Main.css består av grundkoden, där elementen definieras och får de egenskaper för att ritas ut på webbläsaren och media queries koden, där värdena ändras för att anpassas till webbläsarfönstret. Totalt antal rader är ungefär samma mellan desktop-first och mobile-first (*tabell 8*). Det som skiljer sig är antal rader av kod som metoderna har i grunden och i media queries. I mobile-first är grundkoden avsedd för mobilen, vilket betyder att den endast läser grundkoden när webbsidan ses i mobilvy, som är 193 rader. I desktop-first är grundkoden avsedd för desktop och läser endast in den vid desktopvy, vilket består av 221 rader. Om webbsidan ses från den vy vars metoden inte är avsedd för, betyder det att den måste läsa in all grundkod och sedan koden i media queries för att ändra egenskaperna för varje element. I desktop-first måste mobilwebbläsaren totalt läsa in 381 rader för att kunna visa webbsidan. I mobile-first måste desktopwebbläsaren totalt läsa in 374 rader för att kunna visa webbsidan. Båda metoder har sina fördelar och inriktar sig för att fungera bäst för var sin enhet. Men återigen måste enheterna ha i åtanke. Mobilen har inte samma prestanda som en dator, antingen läses det in få rader kod i mobilen och många rader kod i desktop, eller många rader i mobilen och få rader i desktop. Det visar fördelarna som diskuterades i litteraturstudien, vilka innebär att låta mobilen läsa in så lite som möjligt på grund av prestandaskäl och datorn mer än nödvändigt då den har kapaciteten att kunna hantera det. Det vill säga låta mobilen läsa in 193 rader och desktop 373 rader för att uppnå bästa resultat, vilket görs genom mobile-first.

6.1.2.2 komplexitet

I jämförelse mellan metodernas css-fil visar sig mobile-first vara ett naturligt sätt att implementera för att det utgår från en design som inom webbutveckling är standard. I Mobile-first är grundkoden avsedd för mobilen där element ligger i en kolumn under varandra. Eftersom det är standard att element visas under varandra behövs ingen kod för detta, utan kod läggs istället till i media queries när webbsidan börjar övergå till desktop och element skall visas vid sidan om varandra. I desktop-first måste elementen redan från början definieras så att de skall ligga bredvid varandra då detta inte är standard. När webbsidan börjar övergå till mobilläge och webbläsaren blir mindre, måste i media queries se till att det blir ogjort så att elementen hamnar under varandra i en kolumn. Med andra ord kräver detta en rad kod i mobil-first där positionen på elementet ändras när den syns i desktop (*figur 14*), men två rader kod i desktop-first, en som ändrar elementens position från början och en som ser till att det återgår till standard när sidan syns i mobilen (*figur 15*). Det är endast fåtal element i prototypen där detta förekommer, men tyder på att fallet kan komma att existera fler gånger för större webbsidor. Speciellt nyhetssidor, där många element går från att vara vid sidan om varandra till att vara under varandra i en kolumn. Det medför att mobilsidan blir ett naturligt sätt att påbörja implementationen av den responsiva webbplatsen, då det börjas från standard.

<pre> . articles { color: white; width: 200px; height: 200px; } @media (min-width: 768px) { float: left; } </pre>	<pre> . articles { float: left; color: white; width: 200px; height: 200px; } @media (max-width: 768px) { float: none; } </pre>
(a) Mobile-first	(b) Desktop-first

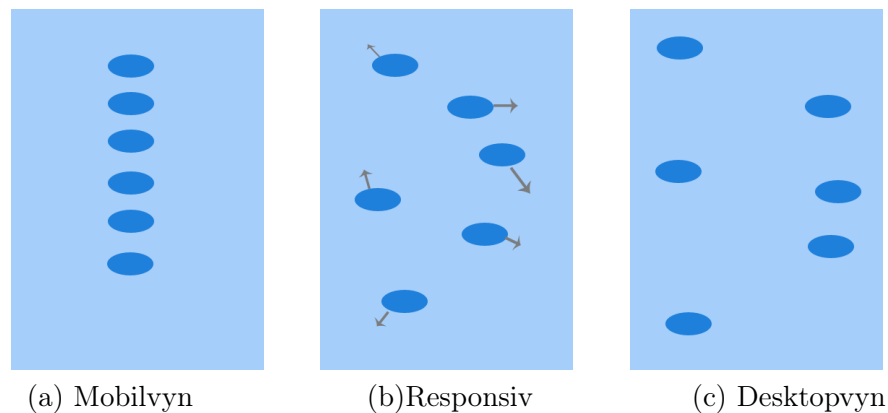
Figur 14: I Desktop-first(b) måste elementet först definieras så att de ligger vid sidan om varandra, sedan med media queries tas detta bort när det går över till mobilvyn. I Mobile-first(a) är det standard och inget behöver definieras förens webbsidan övergår till desktopvy, då sätts elementen vid sidan om varandra.

6.1.3 Implementeringstid

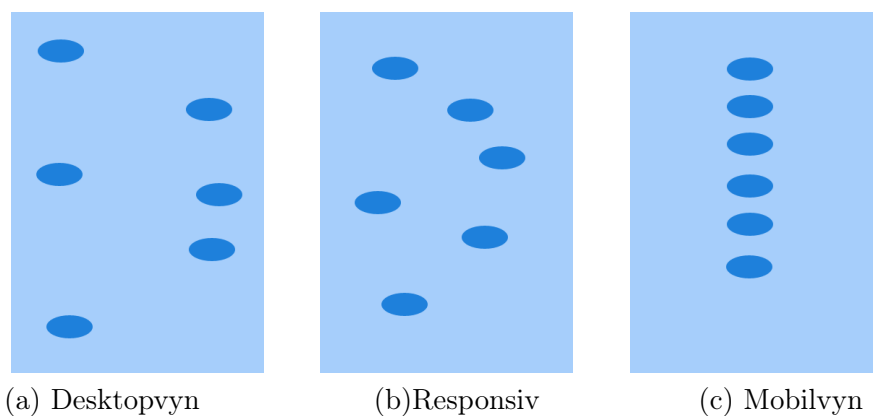
Implementeringstiden för webbsidorna skiljer sig med ungefär med 5 timmar, mobil-first tog totalt 16,5 timmar och desktop-first 22h. Det går inte att dra slutsatsen att mobile-first tar mindre tid att implementera än desktop-first, på grund av att: 1. Desktop-first implementerades före mobile-first, för en junior gränssnittsutvecklare, kan desktop-first implementeringen varit en bra uppvärmning, 2. För att erfarenheten av utvecklaren inte är tillräckligt stor, vilket kan medföra inläring inräknat i tiden utan att det har märkts av. En teori kan ändå skapas av kommentarerna från dagboken och tiden som redovisades under implementationsfasen.

Grundsidan för desktop-first tog längst tid att implementera, vilket enligt dagboken krävde många element att dekorera och positionera utspritt på en stor yta. Saknas erfarenhet inom området kan det vara svårt att finna en struktur på hur en webbsida skall formas. Grundsidan för mobile-first tog minst tid, vilket i dagboken beskrivs som ett enkelt sätt att forma en webbsida på. Innehållet samlas i en kolumn där element implementeras uppifrån och ner utan att ha möjligheten att tappa bort sig åt sidorna eller ändra något som påverkar andra element. Att göra webbsidan responsiv tog mindre tid för desktop-first än mobile-first. I desktop-first har elementen dekorerats och lagts ut på rätt plats, det enda som behövs göras är att minska elementen och samla ihop dessa till en kolumn. Det tar fluid grid, fluid images och media queries hand om utan att gränssnittsutvecklaren behöver justera för mycket. I mobile-first börjar elementen i en kolumn, det som skall göras är att låta fluid grid, fluid images göra så att dessa förstöras, men förändringar på webbsidan sker tidigt. I mobile-first krävs det då fler justeringar i media queries då elementen skall positioneras till sina utspridda platser i desktopvyn istället för en kolumn där elementen finns under varandra. Det kan vara anledningen till att responsivdelen för mobile-first tog längre tid att implementera än responsivdelen i desktop-first.

Mobile-first har fördelen att grundsidan implementeras på ett simplare sätt än i desktop-first. Istället för att rita ut alla element utspridda på en sida, dekoreras dem i en kolumn och låter den responsiva delen göra så att elementen hamnar på rätt plats i desktopvyn, med ett antal justeringar med media queries (*figur 15*). I desktop-first sprids och dekoreras elementen för att sen låta den responsiv delen samla in dessa i en kolumn, men då har redan mycket tid lagts på att rita och positionera ut elementen i desktopvyn (*figur 16*). Teoring säger att mobile-first är ett effektivare tillvägagångsätt, men om behovet istället är en snabb lösning för en existerande desktopsida responsiv, finns fördelen tidsmässigt hos desktop-first då grunden redan är lagd.



Figur 15: Responsiv i Mobile-first. Elementen börjar designas och placeras i en kolumn på skärmen, responsive ser till att med en del justeringar placera dessa rätt i desktopvyn.



Figur 16: Responsiv i Desktop-first. Elementen måste designas och placeras rätt på skärmen, responsiv ser till att de samlas en kolumn för mobilvyn.

6.1.4 Skärmstorlek

Storleken på mobilskrmen har i litteraturstudien setts som ett hinder, en barriär som kräver fokus där en design tas fram efter en prioritering av de nödvändigaste funktionerna. Under implementeringen var skärmstorleken en lättnad, en kompakt design som ledde till struktur i arbetssättet. Prototypen av webbsidan var en förenklad version av riktiga nyhetssidor, vilket kan vara anledningen till att prioriteringsvärigheten aldrig uppkom under implementationen.

Prototypen innehöll inte lika många element som en verklig nyhetssida, vilket gör det svårt att relatera till prioriteringssvärigheten som har beskrivits i litteraturstudien. I desktop-first

är skärmstorleken stor, där prioritering av funktionalitet inte har samma betydelse som i mobile-first, då det mesta får plats. Däremot visar detta återigen risken för att mobilvyn istället blir en förenklad och nerskalad version av desktopsidan. Skärmstorleken bör ses som ett hjälpmedel istället för ett hinder. Ett hjälpmedel för utvecklaren och interaktionsdesignen som försäkrar att funktionalitet inte försvinner mellan vyer och istället skapar en bra grund för både arbetssätt och användbarhet. Användbarhet har eskalerat inom webbutveckling, vilket av egen erfarenhet anses vara högt prioriterande. I litteraturstudien delas denna åsikt med härledning till att webben börjar ses utifrån ett annat perspektiv, där funktionalitet och användbarhet uppskattas mer än mäktig design som kräver för hög prestanda, i synnerhet för en mobil. Desktop-first är inte nödvändigtvis ett hinder för att skapa en användaranpassad webbsida. Tack vare mobilens kompakta skärmstorlek som kräver fokus på innehåll och funktionalitet, är mobile-first en försäkran på att användbarhet i praktiken upprätthålls.

6.2 För- och Nackdelar

För- och nackdelarna som diskuteras i nedanstående sektion är de som förekommer i både litteraturstudien och i implementeringsfasen. De fördelar så som att endast finns i litteraturstudien men som inte kan styrkas av implementeringen av prototypen tas inte med i detta avsnitt, vare sig det är en fördel eller nackdel.

Fördelarna med mobile-first kan sammanfattas i tre punkter: **Fokus på innehåll och funktionalitet, anpassad till dagens teknik och ett naturligt arbetssätt.**

Innehåll och funktionalitet i fokus betyder att dessa prioriteras framför presentation och animation. Prioriteringen sker just för att grunden avser mobilsidan vilket innebär en begränsad yta att designa för. Detta tvingar fram bra användbarhet då betydelsen för hur lätt samt en användare kan förstå och navigera på webbsidan prioriteras högt. Med ett fokus på innehåll och funktionalitet undviks skapandet av webbsidor som blir subventionerade versioner beroende på vy, det vill säga, funktioner och innehåll som tas bort på grund av platsbrist. Eftersom innehållet och funktionaliteten är i centrum skapar det även en robust grund där allt annat byggs utifrån den.

Anpassning för dagens teknik är en viktig punkt som visades tydligt av resultatet från responstiden, kodanalysen och som även styrks av litteraturstudien. Tekniken utvecklas ständigt och möjligheten för mobiler att hinna ikapp datorerna i prestanda existerar, men är inte där än. Mobile-first har det i beaktande då metoden är anpassad för dagens teknik där mobilen har sämre prestanda än datorn. Mobile-first läser in mindre kod i mobilvyn, därav snabbare responstid för webbsidor som ses från mobilskrämen. Användningen av mobilt internet ökar för varje dag där användarna kräver snabb respons vid flexibla situationer där tidspress existerar. Mobile-first är anpassad för de prestandasvårigheter som existerar hos mobiler, men som inte är lika kritiska för datorer.

Mobile-first tillför ett **naturligt arbetssätt**, här designas elementen på ett sätt som är

standard först och som sedan ändras desto större skärmstorleken blir. Positionerna för element som formar webbsidan behöver inte definieras då dessa redan befinner sig en kolumn. Under implementationsfasen påpekades fördelen med att börja utveckla efter en design där alla element är under varandra. Det medför en struktur som är lätt att hålla sig efter och där fokus finns för att ta sig an de svåra problemen först, därav en robust grund att bygga på. Arbetssättet som mobile-first medför påminner mycket om agila metoder och kan därför fungera bra tillsammans med dem. Element prioriteras och det arbetas utifrån en stabil grund där funktioner byggs på.

Flera nackdelar som togs upp i litteraturstudien är inte relevanta till implementationsfasen. Det är svårt att styrka ett påstående som bygger på känslor snarare än något tekniskt t.ex. *"det kan uppfattas som tråkigt och jobbigt eftersom det utmaningarna möts i början"*. Däremot uppmärksammades att media queries inte fungerar för internet explorer 6,7 och 8. Prototypen valdes därefter att inte vara avsedd för internet explorer 8, vilket tyder på att det är ett stort problem samt en nackdel för mobile-first. Internet explorer 8 används fortfarande i en mindre utsträckning och så länge windows xp finns installerat som operativsystem kommer även möjligheten för internet explorer 8 som primär webbläsare att finnas. Lösningen är inte nödvändigtvis att övergå till desktop-first då detta medför nackdelarna desktop-first har gentemot mobilen, men istället använda en annan css för denna webbläsare. Det kommer att medföra en del omständigheter för webbutvecklaren som måste skriva en separat css-fil, men kan vara värt med tanke på de fördelar som mobile-first bidrar med. Förutom media queries existerar andra tekniker som internet explorer 8 inte kan hantera och finns därför risken att webbsidan måste modifieras för webbläsaren oavsett metod.

Fördelen med desktop-first är att det är ett simpelt sätt att från en existerande desktopsida övergå till en responsiv. Det kan komma till användning vid visning av en prototyp av en responsiv webbplats eller vill förbättra något litet när webbsidan ses i smalare vyer, eftersom det går snabbt att applicera. Däremot är den inte lika optimal som mobile-first vid skapandet av större responsiva webbsidor. Fördelen beror på att grundsidan och elementen redan är designade och rätt positionerade. Med hjälp av media queries går elementen att enkelt anpassa till mindre vyer. Nackdelen blir däremot att mobilvyn har tendensen att bli en nerskalning av desktopsidan. All innehåll från en desktopvy kan vara svårt att få in i en mobilvy och funktionalitet får istället prioriteras bort. Mycket kod som inte används läses även in när webbsidan ses i mobilvy, vilket leder till att om prestandaproblem åtgärdas så sker det i efterhand. Mycket av fördelarna hos mobile-first blir strikt nackdelar hos desktop-first då desktop-first fokuserar på en optimal responsiv webbplats för desktop istället för mobilen. Tillskillnad från mobilen är desktop en enhet med bra prestanda och en skärmyta som inte innebär samma begränsningar.

6.3 När ska Desktop-first respektive Mobile-first användas

I resultatet beskrivs användningsområdet för mobilt internet vara stor och utspritt. Två källor från litteraturstudien beskriver användningsområdet vara en viktig aspekt att analysera när metod väljs. I analysen av användningsgruppen har målgruppen mellan 12-45 år visats sig ha mer än 50% användare av mobilt internet dagligen. I målgruppen 45-55 år är användningen av mobilt internet 30% dagligen. I resultatet påpekas även det felaktiga antagandet att mobilt internet endast används vid rörliga situationer, vilket istället visar tvärtom. Större delen av mobilt internet används vid stationära tillfällen t.ex hemma eller på kontoret. Gällande kontext av webbsida har de fyra större områdena visat sig vara, hantering av e-post, sociala nätverk, söka information och kolla nyheter, vädret. Dessa fyra områden har växt markant inom mobilt internet och sannolikheten att dessa överkommer desktop visar sig vara stora. De områden där mobilt internet inte har lyckats nå upp till desktop och där siffran är låg, är "fylla enkäter" och "handla produkter". Det är svårt av resultatet att se varför det är så och ifall en förändring kommer att ske. Det som statistik visar däremot är att mobilen oftast används för att leta upp produkter innan köpet sker från desktop. Det betyder att om webbsidan ändå ses ofta från en mobil, så bör möjligheten att kunna köpa produkten finnas. Med en robust och säker mobilsida kanske det till och med sker. Mobilt internet ökar så pass mycket i användning i alla områden för målgrupp, miljö och kontext på webbsida, vilket gör att det är svårt att hitta ett fall där mobilsidan har möjligheten att vara en nerskalad version av desktopsidan. En nerskalad version där prestandan inte är lika viktig när webbsidan ses från mobilen och där mobilsidan agerar mer som bonus för desktopsidan.

Detta leder till att mobile-first bör användas i de flesta fallen. Speciellt för webbsidor som avser stora och informationsrika webbplatser, vilket då kan vara en stor fördel för att främja mobilens prestanda. Det är även ett effektivt sätt att arbeta med CSS-filen. Det ger en bra struktur för arbetet och försäkrar att webbsidornas vyer uppfyller samma kriterium. Användningen av mobilt internet är så pass stor och fördelat i så många områden att det är svårt att hitta ett scenario som tillåter en sämre mobilsida där information och funktionalitet inte nödvändigtvis behöver vara på samma nivå som den på desktopsidan. Mobile-first är motsatsen, det är en försäkran på att dessa nackdelar inte förekommer och att webbsidan uppfyller sitt syfte i både mobilvyn och desktopvyn, vilket visar sig i litteraturstudien för användningsområden att vara högst nödvändigt.

Desktop-first är däremot en snabb lösning till en responsiv webbplats utifrån en existerande webbsida. Den existerande desktopsida har redan designats och med hjälp av media queries modifieras elementen till att förminskas och passa in i en mobilvy. Områdena där detta skulle hjälpa är vid skapandet av en prototyp för en responsiv webbplats och hur denna skulle tänkas se ut utifrån ett responsivt perspektiv. Prototypen tillför funktionalitet vilket en vanlig designsketch inte kan tillföra. Däremot är det viktigt att förstå att prototypen inte kan tillföra den robusta grund som krävs för att fortsätta implementationen på, då grunden är avsedd för en desktopsida och inte en responsiv webbsida. Desktop-first kan även vara ett bra alternativ när webbsidan skall uppfylla funktioner som helst görs i ett stationärt

perspektiv och inte kräver samma användbarhet och funktionalitet från mobilvyn. Men att finna sådana områden är i nuläget svårt.

Ett alternativ är att webbsidan inte nödvändigtvis har samma funktionalitet på mobilen som på desktop, t.ex att utföra administrativa åtgärder från desktop, men endast kunna se ändringarna från mobilen. Med tanke på hur mobilt internet ökar i alla användningsområden tyder detta på att en mobilanvändare kommer att spendera mycket av tiden på internet via mobilen och därmed vilja ha samma möjligheter hos mobilen som på desktop. Därför är det ett tillvägagångssätt som inte gynnar dagens webbgränssnitt. Istället är mobile-first det tillvägagångssätt som i längden ger ett bättre resultat.

7 Slutsats

Arbetet har med hjälp av litteraturstudien och implementationen av prototypen kunnat påpeka viktiga punkter kring mobile-first och desktop-first vid en strikt jämförelse. Dels att koden är mer naturlig hos mobile-first då designen för mobilvyn speglar en standard positionering för element, desktop-first tillför ett enkelt och snabbt sätt att implementera den responsiva delen med och att mobile-first ger en bättre arbetsstruktur då det byggs utifrån en robust grund vilken är lätt att följa upp med ändringar. Dessa är skillnader som fås fram när ena metoden ställs mot den andra utan att ha andra faktorer i åtanke. Men resultatet och diskussionen av arbetet visar att valet av metod inte endast ligger i förmån för webbutvecklaren, utan även för användarna av den responsiva webbplatsen och hur väl metoderna passar in i dagens teknik. Dagens webbsidor ses från både mobiler och datorer, vilka är två enheter som förutom skärmstorlek skiljer sig i prestanda. I arbetet visas tydligt fördelarna för mobile-first och desktop-first för respektive enhet och varför det kan vara till fördel att prioriteringen sker för mobilen före desktop.

Desktop-first främjar användaren när webbsida ses från desktop. För mobilvyn tillkommer inte samma fördel då mer kod behöver läsas in och element från desktopvyn inte alltid får plats i mobilvyn på grund av den begränsade skärmytan. Mobile-first främjar användaren när webbsidan ses från mobilen. Den behöver endast läsa grundkoden och inga ändringar för när webbsidan övergår till desktopvy. Den begränsade skärmstorleken tillför prioritering hos element för att förbättra navigeringen för en användare vilket ger en stabil och genomtänkt grund för webbsidan. När webbsidan övergår till desktopvy är nackdelarna däremot inte lika drastiska som när webbsidan i desktop-first övergår till mobilvyn. Desktop har större yta än mobilskärmen vilket tyder på att alla element och funktionaliteter tas med i desktopvyn. Webbsidan läses in från en dator vilken hanterar de prestandakomplicationer som mobiler inte lika lätt hanterar. Med andra ord hanterar datorn nackdelar som förekommer med mobil-first bättre än vad mobilen hanterar nackdelar som förekommer med desktop-first.

Detta tyder på att skillnaderna mellan mobile-first och desktop-first blir stora när det kommer till ett tekniskt perspektiv, där tekniken för mobilen inte är lika avancerad som den för datorn. Mobile-first ser till att enheten med sämre förutsättningar läser in så lite kod som möjligt medan den med bättre får ta på sig ansvaret att läsa in mer än nödvändigt. För datorn blir uppladdningstiden omärkbar, men för mobilen hade detta betytt en längre uppladdningstid, i värsta fall mer än vad som accepteras av en användare. Dagens användare av mobilt internet finns numera i alla områden och det blir vanligare att webbsidor besöks lika mycket via mobilen som via desktop. Detta kräver mer fokus på mobilen för att kunna skapa samma förutsättningar som när surfandet sker från en dator. Med mobile-first åtgärdas komplicationer som prestanda och begränsad yta i första hand och tillför ett tillvägagångssätt som inte bara gynnar webbutvecklaren, men även dagens teknik och användandet av mobilt internet.

Referenser

- Bull, Matthew (2012): *Mobile comes first: responsive web design in a mobile world*. Senast besökt: 11 juli 2013. URL: <http://blogs.kent.ac.uk/webdev/2012/07/22/mobile-comes-first-responsive-web-design-in-a-mobile-world/>.
- Chan, Michael (2013): *The better approach to rwd: Mobile first design*. Senast besökt: 11 juli 2013. URL: <http://www.readyartwork.com/2013/05/15/mobile-first-design/>.
- Chapman, Katy (2012): *Interview by Edd Gold, Why design for mobile before desktop?* Senast besökt: 11 juli 2013. URL: <http://www.othermedia.com/blog/why-design-for-mobile-before-desktop,424,TA.html>.
- Church, Karen och Nuria Oliver (2011): "Understanding Mobile Web and Mobile Search Use in Today's Dynamic Mobile Landscape". I: *Telefonica Research*.
- Clark, Barry (2009): *Adaptive, Responsive or Mobile First*. Senast besökt: 14 maj 2013. URL: <http://barryclark.co/adaptive-responsive-or-mobile-first/>.
- Dimestam, Andreas (2012): *Acer släpper multi touch-skärmar för windows 8*. Senast besökt: 11 juli 2013. URL: <http://www.sweclockers.com/nyhet/16010-acer-slapper-multi-touch-skarmar-for-windows-8>.
- Fidler, Roger (2013): *2013 Q1 Research Report 1: News consumption on mobile media surpassing desktop computers and newspapers*. Senast besökt: 12 juli 2013. URL: <http://www.rjionline.org/research/rji-dpa-mobile-media-project/2013-q1-research-report-1>.
- Findahl, Olle (2012): *Svenskarna och Internet 2012*. .se. ISBN: 978-91-979411-7-4.
- Foster, Aidan (2012): *Mobile First: A key ingredient in great responsive design*. Senast besökt: 11 juli 2013. URL: <http://responsivedesign.ca/blog/mobile-first-a-key-ingredient-in-great-responsive-design>.
- Fransén, Karin och Oscar Holmström (2012): *Prisutevecklingen på mobiltelefoni och bredband*. PTS. ISBN: PTTS-ER-2012:26.
- Grigsby, Jason (2011): *Where are the mobile first responsive web designs?* Senast besökt: 11 juli 2013. URL: <http://blog.cloudfour.com/where-are-the-mobile-first-responsive-web-designs/>.
- Gustafsson, Marcus (2012): *Telia sänker surfpriser i USA*. Senast besökt: 18 juni 2013. URL: <http://www.telekomidag.se/nyheter/artikel.php?id=38201>.
- Hansson, Daniel (2012): *Responsive websites - mobile vs. desktop first*. Senast besökt: 11 juli 2013. URL: <http://www.neocreo.net/en/articles/responsive-websites-mobile-vs-desktop-first>.
- Helmreich, Doug och Philip Doriot (2009): "Smartphones, Providers and the Customers Who Love (and Loathe) Them". I: *CFI Group Smartphone Satisfaction Study 2009*.
- Houx, James (2013): *Choosing between Mobile-first, Desktop-first or Independent hemes and Sites*. Senast besökt: 11 juli 2013. URL: <http://www.armstrongtheme.com/appendix/choosing-between-mobile-first-or-desktop-first/>.
- Ingmar, Sara (2012): "Responsive web design: En studie i hur designkonceptet påverkar användbarheten hos en webbplats". I: *Uppsala Universitet*.

- Johnson, Joshua (2013): *Mobile First Design: Why It's great and why it sucks*. Senast besökt: 11 juli 2013. URL: <http://designshack.net/articles/css/mobilefirst/>.
- Lande, Simon (2009): *Mobile First, Desktop Later*. Senast besökt: 15 maj 2013. URL: <http://www.cmswire.com/cms/customer-experience/mobile-first-desktop-later-019716.php>.
- Marcotte, Ethan (2011): *Responsive web design*. Jeffrey Zeldman. ISBN: 978-0-09844425-7-7.
- Meeker, Mary och Liang Wu (2009): *2012 Internet Trends*. Senast besökt: 14 maj 2013. URL: <http://www.kpcb.com/insights/2012-internet-trends>.
- Mellas, Caleb (2011): *Why should I design for mobile-first?* Senast besökt: 11 juli 2013. URL: <http://www.webinsation.com/why-should-i-design-for-mobile-first/>.
- Pope, Marcus (2012): *Mobile-first, kind of like the secret*. Senast besökt: 1 juli 2013. URL: <http://www.marcuspope.com/blog/mobile-first,-kind-of-like-the-secret>.
- quirksmode.org (2009): *Mobile compatibility tables*. Senast besökt: 14 maj 2013. URL: <http://quirksmode.org/m/table.html>.
- Rana, Pranay (2009): *Div and table*. Senast besökt: 14 maj 2013. URL: <http://www.codeproject.com/Articles/47934/DIV-TABLE>.
- Schmiedl, Grischa, Markys Siedl och Klaus Temper (2009): "Mobile Phone Web Browsing - A Study on Usage and Usability Of The Mobile Web". I: *University of Applied Science St. Pölten, Institute of Media Informatics*.
- Seitz, Patrick (2009): *Inventors - Apple's Iphone Turns 5 'At The Top Of Its Game'*. Senast besökt: 14 maj 2013. URL: <http://news.investors.com/technology/062712-616302-apple-iphone-turns-5-faces-rivals-from-samsung-google.htm>.
- Shannon-Missal, Larry (2013): "Different Priorities in Smartphone vs Computer Use, But Some Common Ground". I: *The Harris Poll*.
- Singhal, Rohit (2009): *Analyzing mobile vs Desktop user behavior*. Senast besökt: 14 maj 2013. URL: <http://www.designer-daily.com/analyzing-mobile-vs-desktop-user-behavior-29380>.
- Stanley, Morgan (2009): *Internet trends 2010*. Senast besökt: 14 maj 2013. URL: <http://www.slideshare.net/fred.zimny/morgan-staneley-internet-trends-ri041210>.
- w3schools (2009): *CSS How To...* Senast besökt: 14 maj 2013. URL: http://www.w3schools.com/css/css_howto.asp.
- Wroblewski, Luke (2010): *Mobile web design tips: Mobile should come first*. Senast besökt: 11 juli 2013. URL: <http://www.techradar.com/news/internet/mobile-web-design-tips-mobile-should-come-first-719677/1#articleContent>.
- (2011): *Mobile First*. Jeffrey Zeldman. ISBN: 978-937557-02-7.
- Zee, Theo van der (2012): *Responsive web design and mobile first*. Senast besökt: 11 juli 2013. URL: <http://www.themepartner.com/blog/100/responsive-web-design-and-mobile-first/>.
- zurbword (2012): *Mobile first - An adaptive, future-friendly solution for website design*. Senast besökt: 11 juli 2013. URL: <http://zurb.com/word/mobile-first>.