**Diskussion**

**Mobile-first 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 tretton olika källor, vilka inläggsdatum för de flesta inte är mer än ett år gamla. Av dessa tretton källor är det elva som förespråkar för mobile-first, en som föredrar desktop-first och en som anser att tekniken är för ny för att förorda(inlägget skrevs 2011). Implementationsfasen bekräftar mycket av det som har sagts i litteraturstudien, både till fördel för desktop-first och mobile-first. Men i slutändan har implementationen utav webbsidorna visat sig vara kod mässigt relativt lika, mobile-first använder sig utav min-width, medan desktop-first använder sig utav max-width, mobile-first ser till att rendera om elementen när skärmen blir mindre och desktop-first när skärmen blir större. Mycket fokus har lagts på metoderna i sig, när ett större problem är att det existerar en enhet med sämre prestanda, mindre skärm och bandbreddsbegränsningar, som inte existerar i lika stor utsträckning på den andra. Nutidens mobiler har kommit långt i utvecklingen, men kan inte mäta sig med dagens datorer. Det visade sig att arbetet kom att handla om vilken metod passar bäst i dagens teknik, och inte vilken metod som är bäst och dagens teknik består av mobiler vilka presterar sämre än datorer.

Mobile-first tar sig an problemen 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 ifrån skrivbordskärmen då den läser både allt för mobil och även det som skall ändras för desktop. Desktop-first läser in det nödvändigaste för skrivbordskärmen, men när webbsidan ses ifrån en mobil, kommer den läsa in allt för desktop och sedan allt som den behöver ändra för att webbsidan skall passa mobilen. Detta betyder att mobil-first kommer se till att mindre läses in för den enhet vars prestanda är låg och mer läses från den enhet vars prestanda är hög och har möjligheten 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 kodalanys, 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, anledningen var att en användare skall ha samma möjligheter vare sig enhet webbsidan ses ifrån. I litteraturstudien ses detta som ett tillvägagångsätt 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 finns på ena enheten men inte den andra. Anledningen till att det ses som ett mobile-first princip är för att det är hos mobilen då element kan ha 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 på en större skärm. 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, vilket kommer sakna funktionalitet som finns för desktopvyn och där mobilvyn istället blir en komplement till desktopsidan. En användare som tidigare har vart 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 mobil designen så fick den inte finnas i desktopdesignen, följdes från start.

Respons tid, kodanalys, implementerings tid och skärmstorlek är en av de större faktorer som stärker de påståenden taget från litteraturstudien, nedan beskrivs och analyseras dessa i synnerhet till de praktiska och teoretiska tillvägagångssätten.

**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 där 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. När webbsidan ses utifrån ett desktopperspektiv hämtas css värderna 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 man skulle vilja jämföra metoderna 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, och desktop fungerar bra i desktopvyn av samma orsak. Men som tidigare poängterats i diskussionen handlar det inte om metoderna var för sig utan enheterna webbsidan ses från, där responstiderna för mobile-first gynnar 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 väldigt låga, ibland endast 2 millisekunder, för att prestandan och bandbredden som har använts för att hämta webbsidorna är ifrån en dator vilket har bra prestanda.

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 desktyopvyn 21ms. 21ms är relativt snabbt, vilket även 36ms. 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. Detta för att mobilen inte har samma prestanda och bredbandsmöjligheter som datorn som användes för hämtningar. Skillnaden mellan desktop-first och mobile-first blir inte längre 2ms, utan måste multipliceras med faktorerna, sämre prestanda och bredband. Vilket leder till att skillnaden blir mellan metoderna blir oerhört större än vad som har visats i tabellen för hämtningarna av css-filen(tabell nr 5). 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 has i åtanke då den egentligen bara speglar skalet av en fulländig webbsida. Det vill säga att hämtningarna som görs är för en webbsida som är relativt liten i jämförelse med en normal webbsida. Återigen är dessa 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 det punkter som diskuterats i litteraturstudien. Mobile-first ger robust webbsida för mobilen som fokuserar i första hand på situationer hos användaren där snabb respons krävs. Tack vare prestandan på dagens datorer utelämnar inte mobile-first desktop, utan låter den snarare ta hand om den större lasten, vilken den klarar av det.

//

**Kod**

**Storlek**

Kod storleken 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 storlek. Main.css består utav grundkoden, där elementen definieras och får det egenskaper för att ritas ut på webbläsaren, och mediaqueries koden, där dessa värden ändras för att anpassas efter webbläsarfönstret. Totalt antal rader är ungefär detsamma mellan desktop-first och mobile-first. Det som skiljer sig mer är antal rader av kod som metoderna har i grund och media queries. I mobile-first är grundkoden avsedd för mobilen, vilket betyder att den endast läser grundkoden när webbsidan ses i mobilvy, vilket är 193 rader. I desktop-first är grundkoden avsedd för desktop och läser endast in den vid desktopvy, vilket består av 221st. Om det kommer sig att webbsida ses utifrån den vy vars metoden inte är avsedd för, betyder det att den måste läsa koden från media queries, vilket är grundkoden och all kod upp till media queries för respektive storlek på webbläsarfönstret. I desktop-first måste mobilwebbläsaren läsa totalt 381st rader för att kunna visa webbsidan, i mobile-first måste desktopwebbläsaren läsa totalt 374 rader för att kunna visa webbsidan. Båda metoder har sina fördelar och inriktar sig för att fungera bästa var sin enhet. Men återigen måste enheterna has i åtanke, mobilen är inte alls lika kraftig som datorn. Antingen väljer man att läsa in lite rader kod i mobilen och mycket rader kod i desktop, eller mycket rader i mobilen och lite rader i desktop. Återigen visar detta fördelarna som diskuterades i litteraturstudien, låta mobilen läsa så lite som möjligt på grund av prestandaskäl och datorn mer än nödvändigt då den har kapaciteten att kunna göra det.

**Komplexitet**

Main.css skiljer sig ändå i vissa lägen mellan mobil-first och desktop-first. 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 bredvid varandra. I desktop-first måste elementen redan från början definieras att de skall ligga bredvid varandra eftersom 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 syns under varandra. Med andra ord behövs en rad kod i mobil-first som ändrar elementens position när den syns i desktop, 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. Det är endast fåtal element i protoypen där detta förekommer, men existerar fler gånger i större webbsidor, speciellt nyhetssidor, där många element går från att vara bredvid varandra till att vara under. Det medför att mobilsida blir naturligt att påbörja en responsiv sida med.

Mobile-first

.articles{

color: white;

widht: 200px

height:200px

}

@media all and (min-width: 768px){

float:left;

}

Desktop-first

.articles{

float:left;

color: white;

widht: 200px

height:200px;

}

@media all and (min-width: 768px){

float: none;

}

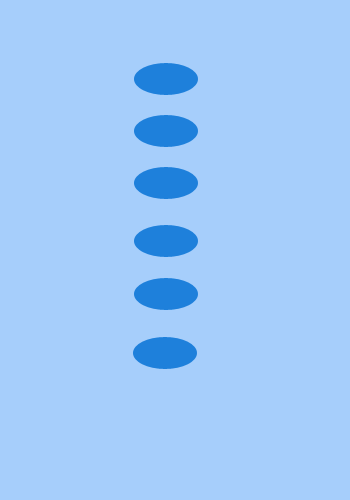
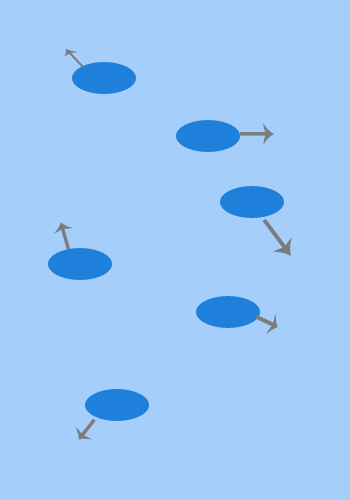
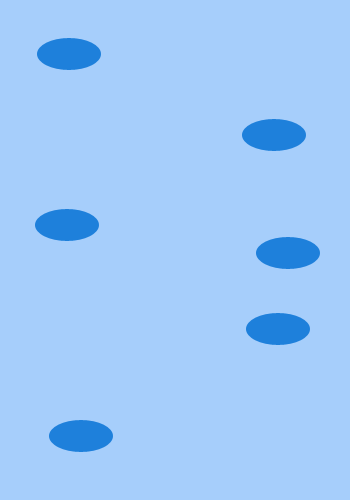
**Implementeringstid**

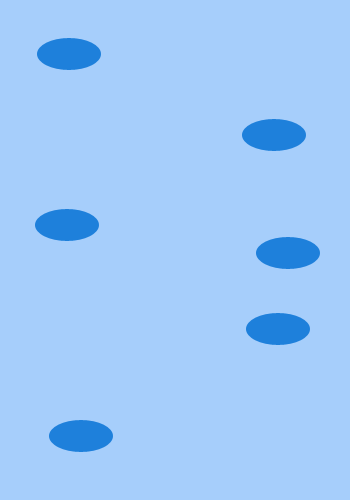
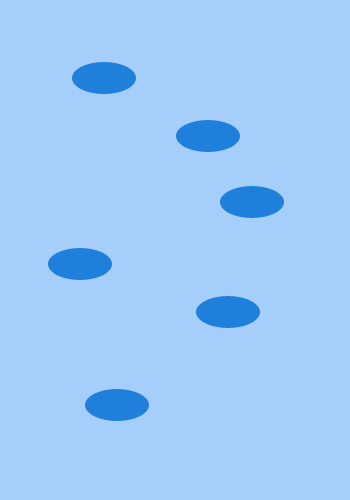
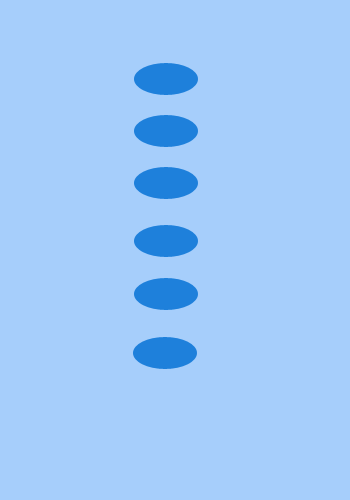
Implementeringstiden för webbsidorna skiljer sig med ungefär med 5 timmar, mobil-first tog totalt 16,5 timmar och desktop-first 22h. Det är svårt att dra slutsatsen att mobile-first tar mindre tid att implementera än desktop-first, på grund av att 1. Desktop-first implementerades före mobile-first, för en junior gränssnittsutvecklare, kan desktop-first implementeringen varit en bra uppvärmning, 2. För att erfarenheten utav utvecklaren inte är tillräckligt stor, inlärningen räknas med i tiden.

Däremot finns en teori till implementeringstiden som grundar sig på kommentarerna tagna från dagboken.

Grundsidan för desktop-first tog mest tid, vilket enligt dagboken krävdes många element att sätta ut och dekorera 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 webbsida, allt är i en kolumn och man arbetar uppifrån neråt utan att tappa bort sig någonstans, eller ändra något som påverkar andra element. Att göra webbsidan responsiv tog mindre tid för desktop-first än mobile-first. I desktop-first har alla element dekorerats och lagts ut på rätt plats, det enda som behövs göras nu är att minska elementen och samla ihop dessa till en kolumn, vilket fluid grid, fluid images och media queries tar hand om utan att gränssnittsutvecklaren behöver justera för mycket. I mobile-first finns elementen i kolumnen, det som skall göras är att låta fluid grid, fluid images göra så att dessa förstoras. I mobile-first krävs det däremot fler justeringar i media queries då inte alla element skall samlas i en kolumn som i reponsivt för desktop-first, utan nu skall varje element till en speciellt plats, vilket måste justeras mer än vad man behöver i desktop-first. Det kan vara anledningen till att responsivdelen för mobile-first tog längre tid än responsivdelen i desktop-first.

Mobile-first har en fördel då i början gör det jobbiga på ett simpelt sätt. Istället för att rita ut alla element sprida på en sida, dekoreras detta i en kolumn och låter responsiv ta hand så att dessa element hamnar på rätt plats i desktop, med ett antal justeringar med media queries. I desktop-first, sprids och dekoreras elementen för att sen låta responsiv samla in dessa i en kolumn, men då har redan mycket tid lagts åt att rita ut elementen i desktopvy. Teoring säger att mobile-first är ett mer effektivt tillvägagångsätt, men att om man snabbt vill göra en existerande webbsida responsivt, finns fördelen med desktop-first för att det går fort då grunden redan är lagd.

**Skärmstorlek**

Storleken på mobilskärmen 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 utav riktiga nyhetssidor, där en svårighet av att förstå sig på prioriteringen uppkommer. I prototypen fanns det inte lika många element att prioritera som i en riktig webbsida, vilket förklarar den svårighet som webbutvecklare finner när i desktop-first finns möjligheten att inte behöva prioritera alls om plats och element. Däremot visar detta återigen att det finns risk för att mobilvyn blir en förenklad och nerskalad version utav desktop-sidan, vilket inte är ett bra resultat. 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 samtidigt skapar en bra grund, för både arbetssätt och användbarhet. Användbarhet är något som har växt inom webbområdet, vilket av egen erfarenhet anses vara högt prioriterande. I litteraturstudien delas denna åsikt med härledning till att webben börja ses med ett annat perspektiv, där funktionalitet och användbarhet uppskattas mer än mäktig design som kan kräva mycket prestanda, speciellt för en mobil. Desktop-first är inte nödvändigtvis ett hinder för att skapa en användarvänlig webbsida, men mobil-first försäkrar att detta upprätthålls, detta tack vare mobilens skärmstorlek som kräver att fokus läggs på innehåll och funktionalitet.

**För- och nackdel**

För och nackdelarna som kommer att diskuteras fram är dessa som både speglar det som har fåtts utav litteratursstudien och av implementeringsfasen, vilket betyder att fördelar så som att endast finns i litteraturstudien men som inte kan stärkas utav implementeringen av prototypen inte kommer att tas med, även om det verkligen är en fördel eller nackdel.

Fördelarna med mobile-first kan sammanfattas i tre punkter, Innehåll och funktionalitet i fokus, anpassad efter dagens teknik och Vägledning för ett bra arbetssätt.

**Innehåll och funktionalitet i fokus** betyder dessa prioriteras framför .t.ex. 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 det viktiga är hur lättsamt en användare kan förstå sig på webbsidan och kunna enkelt navigera på den. Med ett fokus på innehåll och funktionalitet undviker man även att skapa webbsidor som blir subventionerade versioner beroende på vy, eftersom som innehållet och funktionaliteten är i centrum det skapar det även en robust grund där allt annat byggs utifrån det.

**Anpassning för dagens teknik** har varit en viktig punkt som beskrevs i stor utav resultatet från responstiden och kodgranskningen. Självklart kan tekniken ändras, mobiler kan i framtiden ha lika hög prestanda som datorer, men tekniken är inte där än, mobile-first tar har detta i beaktande och är därför anpassad efter dagens teknik. . Mobile-first läser in mindre kod i mobilvyn, därav snabbare responstid för webbsidor som ses från mobilen. Mobiltinternet användning ökar för varje dag, och kräver att webbsidor har snabbrespons vid flexibla situationer. Mobile-first är anpassad för de prestanda komplikationer som existerar för dagens mobiler gentemot dagens datorer.

Mobile-first **tillför ett naturligt arbetssätt,** här designas elementen på ett sätt som är standard först och som sedan ändras sedan desto större skärmen blir. Positionerna för elementen som formar webbsidan behöver inte definieras då dessa redan är i en kolumn. Under implementationens gång påpekades fördelen med att börja utveckla efter en design där alla element är under varandra, det medför en struktur som är lätt att följa och som koncentrerar på att ta sig an de svåra problemen först, därav en robust grund att följa därefter. Arbetssättet som mobile-first medför, påminner mycket om det agila och kan därför fungera väldigt bra, element prioriteras, och man arbetar efter en robust grund där det viktigaste är i centrum och resten byggs utifrån det.

De flesta nackdelar som togs upp i litteraturstudien är inte relevanta till implementationsfasen, det är svårt att stärka 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. Det är ett av de större problemen med mobile-first eftersom det är webbläsare som fortfarande används i en mindre utsträckning och så länge windows XP finns kommer även möjligheten för användning utav dessa. Lösningen är inte nödvändigtvis att övergå till desktop-first, utan fördelarna för mobile-first är så pass många i jämförelse med nackdelen vilket får då lösas på annat sätt, t.ex en annan css för just dessa webbläsare.

Fördelarna med desktop-first är att det är smidigt att övergå till responsivt från en existerande webbsida. Det kan komma till användning om man snabbt vill visa en prototyp av en responsiv webbplats eller vill förbättra något litet när webbsidan ses i smalare vyer. Däremot är den inte lika optimal som mobile-first för att skapa genomtänkta större webbsidor. Anledningen till att det går snabbt är att grundsidan och elementen redan är på plats och är designade, med hjälp av media queries går dessa element att snabbt förminskas och anpassas i en liten skärm. Nackdelen blir däremot att mobilvyn har tendensen att bli en nerskalning av desktopsidan, allt innehåll från en desktopvy kan vara svårt att få in i en mobilvy och funktionalitet får istället prioriteras bort. Mycket kod som inte används ändå läses även in när webbsidan ses i mobilvy, vilket leder till att prestandaproblem är åtgärdas i efterhand.

Mycket av mobile-first fördelar blir nackdelar vid desktop-first då desktop-first fokuserar på att en optimal responsivt webbplats för desktop och inte mobilen som är den enhet som behöver mest fokus på grund av sämre prestanda och begränsad skärmyta.

**När ska man använda desktop-first respektive mobile-first?**

I resultatet beskrivs användningsområdet vara väldigt stor för mobilen, vilket två källor från litteraturstudien beskriver som 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, ett steg högre i åldern , 45-55 år är användningen utav mobilt internet 30% dagligen. I resultatet påpekas även det felaktiga antagandet att mobilt internet endast används vid rörliga situationer, vilket istället visar att större delen använder mobilt internet vid stationära situationer t.ex hemmet eller på kontoret. Gällande kontext av webbsida har de fyra större områderna har visat sig vara, E-post skötande, 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 däremot där mobilt internet inte har lyckats nå upp till desktop och där siffran är väldigt låg jämfört med desktop är ”fylla enkäter” och ”handla produkter”, det är väldigt 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 ett desktop, vilket betyder att om webbsidan ändå ses ofta från en mobil, så skall även möjligheten att kunna köpa produkten finnas, och med en robust och säker mobilsida kanske detta till och med sker. Mobilt internet ökar så pass mycket i användning i alla områden för målgrupp, kontext på webbsida och miljö, vilket gör att det är svårt att hitta ett fall där det räcker att mobilsidan kan vara en nerskalad version utav desktopsidan, där prestandan inte är lika viktig när webbsidan ses från mobilen.

Detta leder till att mobile-first bör användas i de flesta fallen, speciellt för webbsidor som avser stora och informationsrika, vilket då kan vara en stor fördel för att främja mobilens prestanda. Det är även ett effektivt sätt att arbeta med CSS-filen, 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 uppfylls och webbsidan uppfyller sitt syfte i både mobilvyn och desktopvyn.

Desktop-first är däremot en snabb lösning till en responsiv webbplats utifrån en existerande webbsida. Den existerande webbsidan har redan designats och med hjälp av media queries modifieras elementen till att förminskas och passa in i en mobilvy. Områderna där detta skulle hjälp är ifall man vill visa en prototyp för en responsivsida och hur denna skulle tänkas se ut utifrån ett responsivt perspektiv. Prototypen medför funktionalitet vilket en vanlig designsketch inte kan tillföra. Däremot är det viktigt att förstå att protypen inte kan tillföra den robusta grund, då grunden är avsedd för en desktopsida och inte en responsiv webbsida. Desktop-first kan även ses som ett bra alternativ när webbsidan skall uppfylla funktioner som helst görs i ett stationärt perspektiv och inte kräver samma användbarhet och funktionalitet från en mobilvy, men att hitta sådana området är i nuläget väldigt svårt.

Ett alternativ är att webbsidan in nödvändigtvis har samma funktionalitet på desktop så som på mobil, t.ex att utföra administrativa åtgärder från desktop, men endast kunna se ändringar från mobilen.

Med tanke på hur användningsområdena för mobilt expanderar anses inte det vara en bra lösning för ett webbgränssnitt. En mobilanvändare kommer ändå vilja ha möjligheten att utföra samma åtgärder som från desktop, vilket i längden blir lättare med mobil-first.