

Joonas Aarnio & Markku Tähtinen

Sosiaalinen verkkopalvelu mobiiliksi

Tapaus Wreckamovie.com

Tekijä(t) Otsikko	Joonas Aarnio & Markku Tähtinen Sosiaalinen verkkopalvelu mobiiliksi. Tapaus Wreckamovie.com
Sivumäärä Aika	43 sivua 25.1.2012
Tutkinto	Medianomi (AMK)
Koulutusohjelma	Viestinnän koulutusohjelma
Suuntautumisvaihtoehto	Digitaalinen viestintä
Ohjaaja(t)	lehtori Matti Rantala
<p>Opinnäytetyön aiheena on verkkopalvelun mobiiliversio suunnittelu. Työn toiminnallisena osana suunniteltiin ja toteutettiin yhteisöllisen elokuvatuotantopalvelun Wreckamovie.comin mobiiliversio, joka löytyy osoitteesta http://www.wreckamovie.com. Kirjallisessa osassa dokumentoidaan toiminnallisen osan työvaiheita ja ratkaisuja sekä työskentelyssä käytettyjä työmenetelmiä ja kehitysympäristöä. Siinä käydään läpi kohdelaitteet ja niiden selaimet sekä painotetaan mobiililaitteiden yleistymistä web-selainlaitteena ja kuinka siihen tulisi suhtautua.</p> <p>Työ tehtiin Star Wreck Studios Oy:n toimeksiantona. Yksityiskohtaisella tasolla se tukee Ruby on Rails -sovelluskehityksellä rakennettujen sivustojen mobiiliversioiden kehittämistä. Se on kuitenkin suunnattu kaikille mobiililaitteet huomioonottavasta web-kehityksestä kiinnostuneille.</p> <p>Työ pyrkii motivoimaan mobiililaitteiden huomioimiseen web-kehityksessä ja käy läpi erilaisia lähtökohtia ja ajattelutapoja mobiilisuunnitteluun. Työssä myös tarkastellaan mitä hyötyjä koko kehitettävän palvelun kannalta voidaan saavuttaa suunnittelemalla se mobiililaitteiden rajoitusten ehdoilla nk. mobiili edellä -suunnittelutavan mukaisesti.</p> <p>Lisäksi työ avaa responsiivisen web-suunnittelun käsitettä, jonka periaatteena on rakentaa web-sivun sommitelma mukautumaan sitä näyttävän laitteen ominaisuuksien kannalta parhaalla mahdollisella tavalla. Suunnittelutavasta annetaan perustavanlaatuisia kooditason esimerkkejä, jotka vaativat lukijalta ainoastaan HTML- ja CSS-ymmärrystä. Työssä käydään myös läpi mobiililaitteen tunnistaminen Ruby on Rails -sovelluskehityksen avulla ja kerrotaan, mitä eri tapoja on toteuttaa sivustolle mobiiliversio.</p> <p>Työn tärkeäksi johtopäätökseksi nousi se, että web-palvelu tulisi kehittää niin, että sen käytettävyys ja suorituskyky olisivat optimaalisia aina sitä selaavalle laitteelle. Tämä on päällimmäisenä ajatuksena responsiivisen suunnittelun ajattelutavassa, jota voidaan pitää tulevaisuuden mallina web-suunnittelulle.</p>	
Avainsanat	Responsiivinen web-suunnittelu, Ruby, käyttöliittymä, mobiili

Author(s) Title	Joonas Aarnio & Markku Tähtinen Optimizing Social Website for Mobile Devices. Case Wreckamovie.com
Number of Pages Date	43 pages 25 Jan 2012
Degree	Bachelor of Media
Degree Programme	Media
Specialisation option	Digital Media
Instructor(s)	Matti Rantala, Lecturer
<p>The topic of this thesis deals with designing a mobile optimized website. It was carried out as an assignment of Star Wreck Studios Oy. The functional part of the thesis consisted of designing and building a mobile and thus optimized version of the social filmmaking website Wreckamovie.com. The solutions made in different stages of the functional part are documented in the literary part. It also includes the practices used and the development environments, and takes a look at the target devices outlined in the assignment. It emphasizes the growth of mobile as a web access device and shows ways to react to it.</p> <p>In a detailed level, the thesis will support the development of web sites by giving basic code examples. The applied web application framework is called Ruby on Rails. However, it is aimed for everyone interested in mobile optimized web design.</p> <p>The aim of the thesis is to motivate the web designer to take the mobile devices into account in their design. It takes a look at different styles and views of designing for mobile such as designing Mobile First. It shows what kinds of benefits may be achieved by designing a website embracing the constraints of mobile devices, rather than fighting them. The thesis also covers the concept of Responsive Web Design and gives code examples of it. Its' principle is to design a web page to adapt to the media that renders it with the help of media queries and flexible grids.</p> <p>The main conclusion of the thesis is that the web service should be developed in such way, that both its usability and performance are always optimized to the device rendering it. Responsive Web Design is all about that and it could be seen as the way of web design of the future.</p>	
Keywords	Responsive Web Design, mobile, Ruby on Rails, UI, iPhone

Sisällys

1	Johdanto	1
2	Toimeksiannon rajaus ja tavoitteet	3
3	Wreckamovie	4
4	Mobiilimaisemat	5
4.1	Mobiilin yleistyminen	5
4.2	Laitteisto	7
4.2.1	Apple iOS	8
4.2.2	Google Android	9
4.2.3	Windows Phone 7	11
4.3	Mobiiliselaimet	12
5	Suunnittelu mobiililaitteille	13
5.1	Mobiili edellä -suunnittelutapa	13
5.2	App vai Web?	16
5.3	Responsiivinen web-suunnittelu	17
5.3.1	Staattisesta sivusta joustavaksi	18
5.3.2	Mediakyselyt	20
5.3.3	Laitteen tunnistus palvelin- ja asiakaspuolella	23
6	Työmenetelmät ja kehitysympäristö	25
6.1	Scrum-menetelmä	25
6.2	Ruby on Rails ja MVC	26
7	Toteutus	27
7.1	Suunnittelun ja toteutuksen lähtökohdat	27
7.1.1	Esimerkkinä YouTube	28
7.1.2	Esimerkkinä Facebook	29
7.2	Käyttöliittymän suunnittelu	29
7.2.1	Ominaisuuksien siirtäminen mobiiliversioon	29
7.2.2	Käyttöliittymäliitännäiset	36
7.3	Mobiililaitteen tunnistaminen	37

8 Jatkotoimenpiteet

39

Lähteet

41

1 Johdanto

Opinnäytetyömme aiheena on verkkopalvelun optimointi mobiililaitteille. Työ jakautuu toiminnalliseen ja kirjalliseen osaan. Toiminnallisen osan aiheena on yhteisöllisen elokuvatuotantopalvelun Wreckamovie.comin mobiiliversion suunnittelu ja toteutus. Kirjallisessa osiossa dokumentoimme toiminnallisessa osassa käytyjä työvaiheita ja ratkaisuja. Kirjallinen osio käsittelee myös yleisellä tasolla mobiililaitteiden huomioonottamista web-kehityksessä. Toiminnallinen osa on nähtävillä osoitteessa <http://www.wreckamovie.com>. Mobiiliversion pääsee tietokoneen selaimella sivun alalaidan linkistä, jos käytössäsi ei ole mobiililaitetta.

Olemme olleet kehittämässä Wreckamovieta vuoden 2009 syksystä ja toteutimme mobiilisivun työprojektina. Työtehtäviimme Wreckamoviessa kuuluu laaja-alaisesti web-palvelun kehittäminen aina graafisesta suunnittelusta tekniseen toteutukseen. Enemmän Wreckamovie.com-palvelusta kerromme kolmannessa luvussa. Olemme tehneet opinnäytetyömme parityönä. Siihen liittyvät ratkaisut ja päätelmät olemme tehneet yhdessä. Kirjallisessa osiossa on työnjako tehty niin, että Aarnio on kirjoittanut neljännen ja Tähtinen viidennen luvun. Muutoin kirjallinen osio on tehty yhteistyönä.

Digitaalisen viestinnän opiskelijoina olemme molemmat kiinnostuneita verkkopalveluiden teknisestä toteuttamisesta ja kasvavana kiinnostuksen kohteena on mobiililaitteille kohdistettu suunnittelu ja toteutus. Olemme tutkineet monia mobiililaitteille optimoituja verkkosivustoja (mm. Facebook ja YouTube) ja lukeneet alan opinnäytetöitä ja kirjallisuutta. Olemme myös huomanneet, että todella monia verkkosivustoja ja -palveluita ei ole mitenkään optimoitu eli paranneltu mobiililaitteita ajatellen.

Ennen opinnäytetyömme toiminnallisen osan aloittamista keväällä 2011 Wreckamovie.com-verkkopalveluakaan ei ollut optimoitu pienille näytöille. Tämä hankaloitti sivuston selaamista ja lukemista mobiililaitteilla. Tavoitteenamme oli luoda helposti saavutettava ja käytettävä versio, jota esimerkiksi av-tuotantojen johtajat voisivat käyttää kuvauspaikalla. Myös tuotantojen jäsenet voisivat mistä tahansa antaa oman panoksensa tuotantojen tehtäviin tai muuten vain selailta palvelun tarjontaa. Opinnäytetyön toisessa luvussa selvitämme tarkemmin mobiiliversion toteutukseen liittyvät toimek-

siantajan ohjeistukset sekä sovitut tavoitteet ja rajaukset. Seitsemännessä luvussa käymme läpi tarkasti työn vaiheita ja tehtyjä ratkaisuja sekä kerromme niitä koskevat toimeksiannossa määritellyt tekniset rajaukset. Sitä ennen käymme luvussa 6 läpi palvelun kehityksessä käytettävää Ruby on Rails -kehitysympäristöä ja sen käyttämää MVC-mallia sekä työskentelyssä käytettyjä menetelmiä.

Tehdessämme taustatutkimusta työllemme selvisi, että kosketusnäytöllisten mobiililaitteiden myynti on kasvanut viimeisen vuoden aikana yli 40 prosenttia (Gartner 2011a). Älypuhelinmyynti on odotettu ylittävän työpöytäkäyttöön tarkoitettujen tietokoneiden myyntiä. Vuoden 2011 kolmannen neljänneksen myyntilukuja tarkasteltaessa voidaan huomata, että näin on jo käynyt. Työpöytäkäyttöön tarkoitettuja tietokoneita myytiin maailmalla alle 92 miljoonaa kappaletta, kun saman ajanjakson aikana älypuhelimia myytiin yli 115 miljoonaa laitetta (Gartner 2011b).

Samalla kun mobiilien päätelaitteiden määrä kasvaa, kasvaa tarve niille optimoiduille verkkosivuille. Nykyään myös 3G-yhteyksien nopeus mahdollistaa sujuvamman sivujen selauksen mobiililaitteilla. Tähän perehdymme enemmän neljännessä luvussa, kun teemme katsauksen tämänhetkisiin mobiilimaisemiin. Lisäksi käymme läpi yleisimpiä mobiililaitteita ja selaimia.

Viidennessä luvussa tarkastelemme yleisellä tasolla erilaisia lähestymistapoja ja näkökulmia mobiililaitteiden huomioon ottamiseen web-kehityksessä. Tutkimme mitä hyötyjä voidaan saavuttaa koko kehitettävän sivuston näkökulmasta, kun suunnittelussa otetaan huomioon mobiililaitteiden edut ja rajoitukset. Käymme myös läpi natiivin eli suoraan laitteelle kehitetyn mobiiliohjelman ja web-ratkaisun eroja. Huomioimme responsiivisen web-suunnittelun näkökulman, jonka lähtökohtia tarkennamme koodiesimerkein, jotka vaativat lukijalta ainoastaan HTML¹- ja CSS²-osaamista.

Opinnäytetyömme on pääasiallisesti suunnattu mobiilisuunnittelusta kiinnostuneille ohjelmoijille ja web-suunnittelijoille. Sen tarkoitus on kuitenkin motivoida muutkin mobiililaitteet huomioivaan ajatusmaailmaan web-suunnittelussa. Myös muut aiheesta kiinnostuneet, kuten esimerkiksi graafikot hyötyvät tästä työstä.

¹ HTML = Hypertext Markup Language, web-sivustojen merkkäuskieli.

² CSS = Cascading Style Sheets, web-sivuston ulkoasun tyylimäärittelyt.

Yksityiskohtaisella tasolla työ tukee Ruby on Rails -verkkosovelluskehityksellä rakennettujen sivustojen mobiiliversioiden kehittämistä. Seitsemännessä luvussa käymme läpi toiminnallisessa osassamme tehtyjä ratkaisuja kooditasolla Ruby-kielellä.

2 Toimeksiannon rajaus ja tavoitteet

Wreckamovie.com-verkkopalvelun mobiiliversion suunnittelu ja toteutus aloitettiin maaliskuussa 2011. Palvelulla ei ollut aikaisemmin minkäänlaista mobiilioptimointia. Internetin selaamiseen soveltuvien mobiililaitteiden suosion merkittävän kasvun sekä käyttäjien lisääntyvien toiveiden myötä nähtiin mobiiliversion suunnittelu tarpeelliseksi. Rajaukset ja tavoitteet mobiiliversiolle sovittiin tiimipalaverissa tuoteomistajan kanssa, joka toimi opinnäytetyön toimeksiantajan asemassa. Työskentely tapahtui Scrummenetelmällä, jonka periaatteita avataan kuudennessa luvussa.

Pääasialliseksi kohdelaitteeksi optimointia varten otettiin Applen iPhoneen mallit 3G ja 4. Niiden yleisyys tuohon aikaan varsinkin Yhdysvalloissa ja poikkeuksellisen hyvät ominaisuudet web-selaukseen olivat päällimmäiset syyt valintaan. Toissijaiseksi kohdelaitteeksi otettiin Google Android -puhelimet, joiden web-selausominaisuudet nähtiin hyvin samankaltaisiksi iPhoneen kanssa.

Tavoitteenamme oli optimoida sivusto moderneille kosketusnäytöllisille mobiililaitteille siten, että sen ulkoasu olisi riippumaton työpöytäversiosta. Tämä mahdollistaisi versioiden ulkoasun ja käyttöliittymän erillisen päivittämisen. Lisäksi tavoitteenamme oli tutkia valmiiden liitännäisten (plugin) käyttömahdollisuuksia mobiiliversion suunnittelussa.

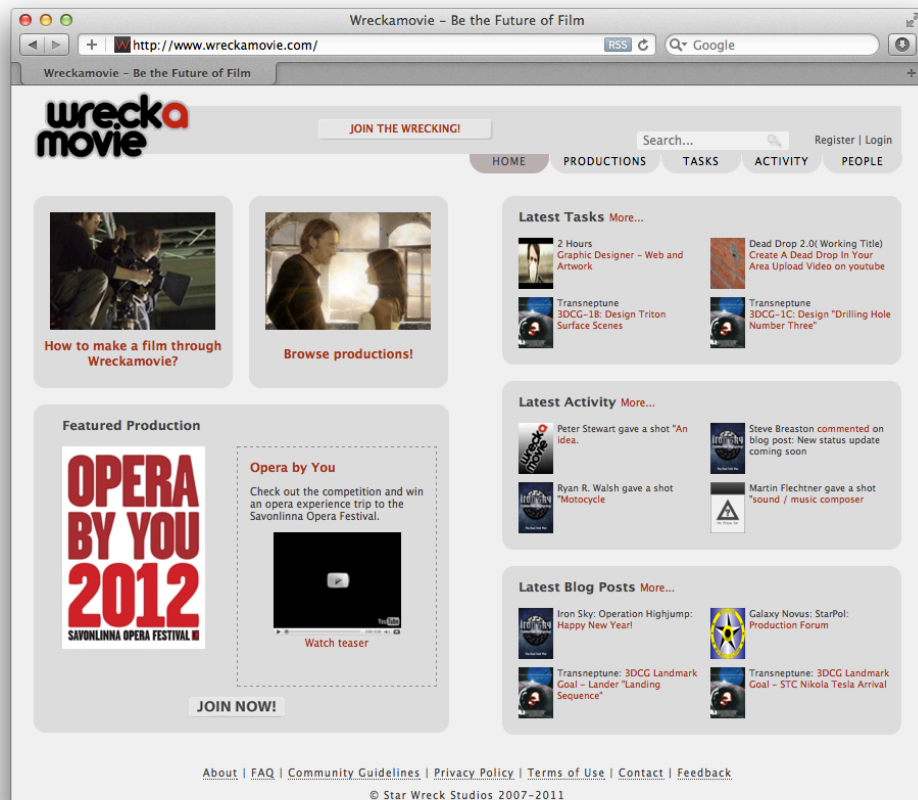
Henkilökohtaiset tavoitteemme opinnäytetyössämme olivat oppia nykyaikaisen mobiilisuunnittelun periaatteet ja parhaat mobiilimenetelmät Ruby On Rails -kehityksessä. Lisäksi halusimme oppia HTML5³- ja CSS3⁴-tekniikoiden käyttöä ja niiden tarjoamia mahdollisuuksia mobiiliverkkosivuja suunniteltaessa. Pyrimme myös oppimaan jQuery-javascript-kirjaston käyttöä dynaamisten verkkosivujen toteuttamisessa.

³ HTML5 = Viimeisin versio HTML:stä.

⁴ CSS3 = Viimeisin versio CSS:stä.

3 Wreckamovie

Wreckamovie.com on verkkopalvelu, joka mahdollistaa eri tyyppisten elokuvien tekemisen yhteisöllisesti suuren ihmismäärän yhteisellä työpanoksella. Käyttäjää on kertynyt jo yli 6000. Palvelu sisältää monia erilaisia tuotantoja aina ooppera-esityksistä sarjakuviin ja täyspitkiin elokuviin. Tuotannon perustaja voi asettaa yhteisön jäsenille erilaisia tehtäviä ja pyyntöjä, jotka auttavat tuotannon etenemisessä. Näitä tehtäviä kutsutaan palvelussa nimellä "task" ja niihin annettuja vastauksia nimellä "shot". Taskien aiheet voivat olla mitä tahansa tuotantoon liittyvää: ideoita, käsikirjoituksia tai sen osia, 3D-mallinnuksia tai elokuvien trailereita. Palvelun avulla kenen tahansa on mahdollista toteuttaa isoja tai pieniä tuotantoja yhteisön avustuksella.



Kuva 1. Kuvakaappaus Wreckamovien etusivun työpöytäversiosta. (Wreckamovie 2011.)

Palvelu sai alkunsa vuonna 2005 julkaistun elokuvan *Star Wreck: In the Pirkinning* jälkeen. Harrastelijavoimin tehdyn elokuvan mahdollisti suuri avustajajoukko, johon pidettiin yhteyttä internetin välityksellä. Elokuva julkaistiin ilmaiseksi internetissä ja aktiivi-

sen avustaja- ja faniyhteisön avustuksella se saavutti alle kahdessa kuukaudessa yli kolmen miljoonan latauksen rajan (Star Wreck).

Lähes kaikki Wreckamovie.comin sisältämät tiedot ja tuotannot ovat kaikkien luettavissa ilman rekisteröitymistä. Tietoa lisätäkseen tai muokatakseen sekä käyttäjäkohtaisten ominaisuuksien käyttöä varten tulee palveluun kuitenkin rekisteröityä.

4 Mobiilimaisemat

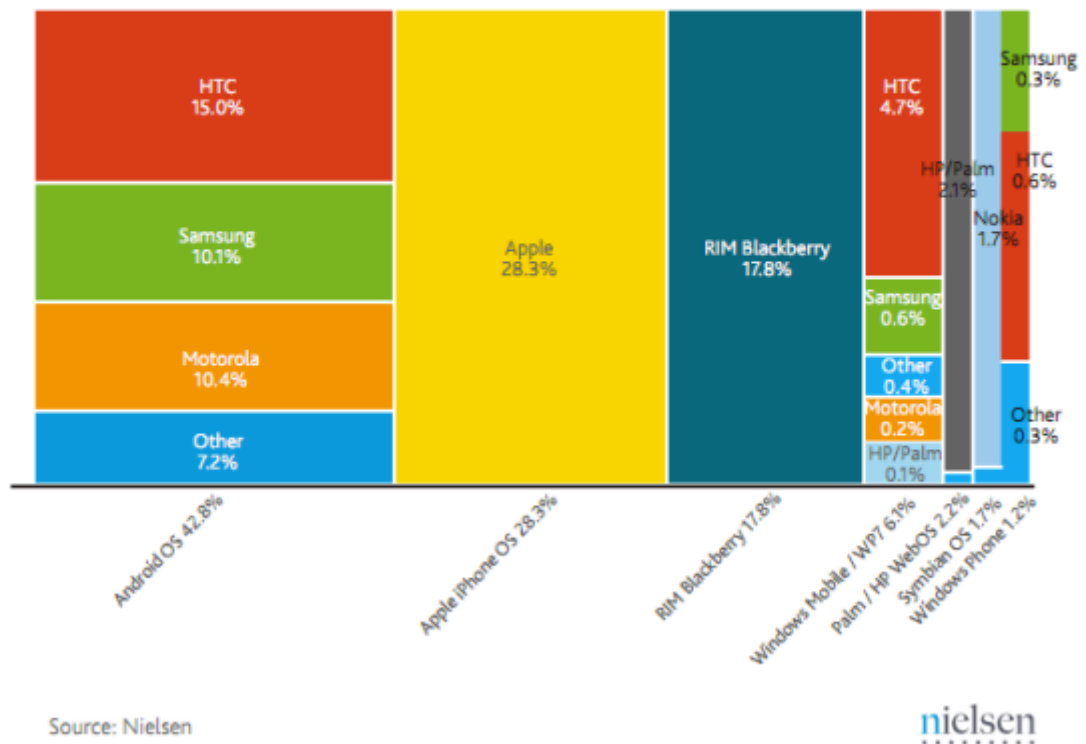
Mobiililaitteet puhuttavat tänä päivänä lähes jokaisessa mediassa. Niiden terveysriskejä tutkitaan, uusia malleja julkaistaan lähes viikoittain ja eri valmistajien välillä on lukuisia patenttikiistoja. Jokainen valmistaja haluaa olla markkinoiden valtiasta ja myydä eniten matka- ja älypuhelimia kuluttajille. Tämä luku käsittelee itse laitteiden lisäksi mobiililaitteiden yleistymistä ja niiden myyntiä ja käyttöä sekä eri älypuhelimien web-selainten ominaisuuksia.

4.1 Mobiilin yleistyminen

Tutkimusyhtiö Nielsenin uusimman Yhdysvalloissa tehdyn tutkimuksen mukaan Applella on valmistajien suurimmat myyntiluvut. Käyttöjärjestelmien myyntiluvuissa Androidilla on suurin markkinaosuus. Seuraavana on Applen iOS, jonka jälkeen Blackberry ja aiemmin älypuhelinmarkkinoita hallinneella Symbianilla on vain 1,7 prosentin osuus. Älypuhelimien osuus Yhdysvalloissa myydyistä puhelimista oli 56 %. (Nielsen 2011.)

Manufacturer operating system share-smartphones

Q3 '11; postpaid mobile subscribers, n=22,200



Kuva 2. Eri käyttöjärjestelmien ja valmistajien markkinaosuudet Yhdysvalloissa syksyllä 2011. (Nielsen, blog 2011.)

Kuvan 2 prosenttiosuuksista huomaa hyvin, että tutkimus on Yhdysvaltojen markkinoilta. Euroopassa tilanne on toinen. Taulukossa 1 esitettyjen heinäkuun 2011 myyntilukujen mukaan Euroopan suurimmissa maissa – Saksassa, Ranskassa, Italiassa, Espanjassa ja Iso-Britanniassa – markkinoita hallitsee edelleen Symbian, jota Nokia kehittää vieläkin, vaikka sen prosenttiosuus onkin laskemaan päin. Seuraavana on vuoden aikana Applen iOS-laitteiden myynnin ohittanut Android ja kolmantena iOS. (comScore 2011.)

Twitter kertoo, että sen viesteistä 14 prosenttia tulee mobiiliselaimen kautta, kun taas 8 prosenttia tulee iPhoneille tehdyn ja 7 prosenttia Blackberrylle tehdyn natiivin ohjelman kautta. (Twitter 2010.)

Top Smartphone Platforms in EU5 by Share of Smartphone Users* 3 Month Average Ending July 2011 vs. July 2010 Total EU5 (DE, FR, IT, ES and UK) Mobile Subscribers, Age 13+ Source: comScore MobiLens			
Smartphone Platform	Share (%) of EU5 Smartphone Users		
	Jul-10	Jul-11	Point Change
<i>Total Smartphone Users</i>	100.00%	100.0%	0.0
Symbian	53.9%	37.8%	-16.1
Google	6.0%	22.3%	16.2
Apple	19.0%	20.3%	1.2
RIM	8.0%	9.4%	1.5
Microsoft	11.5%	6.7%	-4.8

**MobiLens measures users above the age of 13 and reports on only primary handset usage.*

Taulukko 1. Eri käyttöjärjestelmien markkinaosuudet viidestä Euroopan maasta. (comScore 2011.)

Myös Facebookin käyttäjistä löytyy samanlaista статистиikkaa. Lähes 19 prosenttia Facebookin päivityksistä tehdään Facebookin mobiiliselaimella, kun natiiveilla ohjelmissa iPhoneille, Androidille ja Blackberrylle tehdään kullakin vain neljä prosenttia päivityksistä. Tulos näyttää hyvin suuntaa mobiilin yleistymiselle, vaikka tutkimuksen kohteena olikin vain 70 000 Facebookin käyttäjää. (Zarella 2011.)

4.2 Laitteisto

Älypuhelinvalmistajia on jo lukuisia. Tunnettuja valmistajia ovat mm. Nokia, Apple, Samsung, HTC ja Motorola. Apple on kehittänyt oman iOS-käyttöjärjestelmän, jota se käyttää sekä älypuhelimessaan iPhoneissa että multimediatoinin iPod Touchissa ja tablet-tietokoneessaan iPadissa. Samsung, HTC ja Motorola käyttävät älypuhelimissaan Googlen kehittämää Android-käyttöjärjestelmää ja lisäksi Samsungilta, HTC:lta ja Nokialta löytyy puhelimia, jotka käyttävät Microsoftin uutta Windows Phone 7 -käyttöjärjestelmää.

4.2.1 Apple iOS

Ensimmäinen Suomessa myyntiin tullut Applen puhelin, iPhone 3G (kuva 3) julkaistiin vuonna 2008. Puhelimen näytön resoluutio oli 320x480 pikseliä, joka oli tuohon aikaan hyvin yleisesti käytössä. (GSMarena 2008.)

Vuotta myöhemmin Apple päivitti 3G -mallin iPhone 3GS -malliin, joka oli ulkoisesti ja näytöltään identtinen edeltäjänsä kanssa, mutta sisälsi nopeamman prosessorin ja enemmän muistia. (GSMarena 2009.)



Kuva 3. Apple iPhone 3G vasemmalla ja Apple iPhone 4 oikealla (Apple 2012).

Vuonna 2010 Apple julkaisi iPhone 4:n (kuva 3), jossa oli uusi muotoilu, tarkempi näyttö ja nopeampi suoritin kuin edellisissä malleissa. Näytön resoluutio saatiin kasvatettua kaksinkertaiseksi eli 640x960 pikseliin näytön koon pysyessä kuitenkin samana. Tästä resoluutiosta Apple käyttää nimitystä retina display, eli normaalilta katseluetäisyydeltä ei enää pysty erottamaan yksittäisiä pikseleitä. (GSMarena 2010a.)

Apple iPhonen näytön fyysinen koko on kuitenkin sen verran pieni, että optimoimattomat sivustot ovat vaikeita selattavia. Jos sivuston käyttäjistä voi olettaa, että he sela-

vat sivustoa mobiililaitteella kuten iPhoneella tai Android-laitteella, on optimointi lähes välttämätöntä.

Apple muutti myös tablet-tietokoneiden markkinat vuonna 2009 julkaisemalla iPadin (kuva 4), joka sisälsi saman suorittimen ja käyttöjärjestelmän kuin iPhone 4. iPadissa on 9,7” kapasitiivinen kosketusnäyttö. iPad on suunnattu sekä matkustavalle bisnesmiehelle, että tavallisen talleajan arkiseen nettisurffailuun ja pelaamiseen.



Kuva 4. Applen ensimmäinen versio iPad-tabletista, Wi-Fi-malli (The Handheld Blog)

Apple iPad on sekä Wi-Fi-, että Wi-Fi + 3G -mallina 16GB-, 32GB- tai 64GB-kokoisina. Laitteen näytön resoluutio on 1024x768 pikseliä, joka on tuttu myös vanhemmista tavallisista näytöistä. Monet nykyaikaisistakin nettisivuista on optimoitu tuolle näyttöresoluutiolle, joten välttämätöntä tarvetta mobiilioptimoinnille ei ole, vaikkakin se on suotavaa.

4.2.2 Google Android

Android on Googlen kehittämä mobiililaitteille suunnattu Linux-ytimeen pohjautuva käyttöjärjestelmä. Androidia käyttäviä puhelinvalmistajia on useita, mm. HTC, LG, Samsung ja Motorola. Android on kasvattanut suosiotaan viime vuosien aikana, kun

älypuhelimet ovat tehneet tuloaan. Suosittuja puhelinmalleja ovat mm. HTC Desire ja Samsung Galaxy S.



Kuva 5. HTC:n suosittu Android-puhelimen Desiren uudempi versio HTC Desire S. (Geeky-Gadgets)

Nämä ovat laitteistoltaan hyvin samankaltaisia kuin Applen valmistamat iPhone-laitteet. Kapasitiivisten kosketusnäyttöjen koot ovat 3,7 tuumasta (kuten HTC Desire S:ssä, ks. kuva 5) 4,3 tuumaan (kuten HTC Desire HD:ssa) ja resoluutiot ovat 480x800 pikseliä. Androidin internetselain pohjautuu samaan WebKit-moottoriin kuin iPhoneen käyttämä Mobile Safari, joten sen pitäisi pystyä näyttämään sivustot samalla tavalla. (GSMarena 2010b; GSMarena 2011; Android, Wikipedia.)

Aikaisemmin mainitsimme iPadin resoluution riittävän normaalin nykyaikaisen verkkosivuston selaamiseen ilman mobiilioptimoinnin tarvetta. Androidille on kuitenkin mm. kaksi suosittua 7 tuuman näytöllä varustettua tablet-tietokonetta, kuten uusi Amazon Kindle Fire ja Samsung Galaxy Tab. Näissä molemmissa resoluutio on 1024x600, joten pystykytöissä saattaa tulla optimoimattoman sivun leveyden kanssa ongelmia (Amazon 2012; Samsung 2012.)

4.2.3 Windows Phone 7

Microsoft julkaisi Windows Phone 7:ää edeltäneen käyttöjärjestelmän Windows Mobile 6:n jo vuonna 2007 (Microsoft 2007). Windows Mobilea tarjottiin työpöytäversion tavoin kolmena eri versiona: Mobile 6 Standard, Mobile 6 Professional ja Mobile 6 Classic. Windows Mobile 6 oli hyvin samankaltaisia kuin Microsoftin Windows Vista, eikä ollut nykyisten käyttöjärjestelmien tavoin suunniteltu kosketusnäytölle. Kuutosversiota päivitettiin versioon 6.1, jota seurasi versio 6.5 sekä pienemmät päivitykset 6.5.1, 6.5.3 ja 6.5.5.

Kun Microsoft Windows 7 julkaistiin, alkoivat huhupuheet myös uudesta Microsoftin Windows-mobiilikäyttöjärjestelmästä. Loppuvuodesta 2010 julkaistiinkin Microsoft Phone 7, joka oli Microsoftin ensimmäinen kosketusnäytölle suunniteltu käyttöjärjestelmä. Windows Mobilen ja nykyään Phonen etuna ovat aina olleet sisäänrakennetut Microsoft Office -ohjelmat, mukaan lukien Word, Excel ja Powerpoint.



Kuva 6. Nokian uusin Windows-käyttöjärjestelmällä varustettu puhelin Lumia 800. (Mobiiliblogi)

Windows Phone 7 vaatii laitteistoltaan enemmän kuin kilpailijoiden käyttöjärjestelmät. Puhelimen resoluutioksi suositellaan aina vähintään 480x800 pikseliä. Puhelimesta on myös löydyttävä nykyään jo yleistyneet GPS-paikannus sekä laitteen asennon tunnistavat kiihdytysensorit. Nämä laitteistovaatimukset varmistavat puhelinten saumattoman ja sulavan käytettävyyden. (Microsoft 2011.)

Tällä hetkellä Windows Phone 7 on yleisesti käytössä kolmella suurella laitevalmistajalla: Nokialla (ks. kuva 6), HTC:llä ja Samsungilla.

4.3 Mobiiliselaimet

Jokaisella valmistajalla on mobiililaitteissaan käytössä mobiiliversiot työpöytäkäyttöön suunnitelluista internetselaimista: Applen Safaria, Microsoftin Internet Exploreria ja Googlen Chromea vastaava selain. Myös muista tunnetuista selaimista, kuten Operasta ja Firefoxista on tehty mobiiliversiot.

Mobiiliselaimet mahdollistavat paljon. Ne osaavat ladata vain tarpeellisen tai jättää kuvat pois, jos datayhteys on hidas. Lisäksi ne mahdollistavat sivustojen mobiilioptimoitujen versioiden näyttämisen pienellä näytöllä user agentin mukaan. User agent sisältää yleiset tiedot selaimen versiosta, käyttöjärjestelmästä ja sivun selainmoottorista.

Esim. 1.

```
Mozilla/5.0 (iPad; U; CPU OS 3_2_1 like Mac OS X; en-us)
AppleWebKit/531.21.10 (KHTML, like Gecko) Mo-
bile/7B405
```

Esimerkissä 1 mainittu user agent sisältää tiedot Mobile Safari -selaimesta Applen iPad-laitteella. Ensimmäisenä oleva Mozilla/5.0 kertoo selaimen olevan yhteensopiva Mozilla-selainmoottorin kanssa. Toinen osio kertoo laitteen ja käyttöjärjestelmän tiedot. Seuraavaksi kerrotaan mitä alustaa selain käyttää. Yleisimmät alustat mobiiliselaimissa ovat WebKit (Applen Safari Mobile ja Androidin internetselain), Gecko (Firefox Mobile), Trident (Internet Explorer Mobile) ja Presto (Opera Mobile). Viimeisenä kerrotaan jos selaimessa on käytössä laajennuksia tai erikoisominaisuuksia, kuten onko kyseessä

mobiiliselain. Tämän tiedon avulla on helppo kohdistaa sivustojen mobiiliversioita eri laitteille käytettäväksi, kun user agent kertoo suoraan onko käytössä mobiiliselain.

Uudet selaimet osaavat myös kehittää itselleen uuden user agentin, jolla voi huijata sivustoa luulemaan, että käytössä on työpöytäselain. Tämä on hyödyllinen ominaisuus, jos sivuston mobiilioptimointi on kesken tai ei toimi oikein tietyillä päätelaitteilla. Samoin jos sivustolla on jätetty tiettyjä ominaisuuksia pois mobiiliversiosta, pääsee user agentia vaihtamalla helposti palvelun työpöytäversioon käsiksi. Sivustoilta yleensä löytyy myös linkki, jolla sen voi pakottaa toimimaan täysikokoisena, jolloin user agentin vaihtamista ei tarvita. (Mobile Browsers, Wikipedia; Germain 2007.)

5 Suunnittelu mobiililaitteille

Tässä luvussa käymme yleisesti läpi näkökulmia mobiililaitteet huomioivaan web-suunnitteluun. Luku pyrkii myös motivoimaan nykyisten internetin selaamiseen kykenevien laitteiden erilaiset ominaisuudet huomioivaan ajattelumaailmaan.

Alussa tutkimme mitä hyötyjä voidaan saavuttaa koko kehitettävän sivuston näkökulmasta, kun suunnittelussa otetaan huomioon mobiililaitteiden edut ja rajoitukset ja käytetään mobiili edellä -suunnittelutapaa. Käymme läpi natiivin mobiiliohjelman ja web-ratkaisun eroja ja selvitämme kumpaa tapaa tulisi milloinkin suosia. Lisäksi otamme esille responsiivisen web-suunnittelun näkökulman ja lähtökohdat.

5.1 Mobiili edellä -suunnittelutapa

Edellisessä luvussa kävi ilmi mobiililaitteiden huima yleistyminen web-selainlaitteena. Kun pöytätietokone on menettämässä asemaansa johtavana internetilaitteena, on web-kehittäjien huomioitava tapahtuvat muutokset työssään. Kuten Wreckamoviellekin, mobiililaitteet ovat olleet useiden kehittäjätiimien prioriteettijärjestyksessä vasta pöytäkoneen ja kannettavan jälkeen, silloinkin vain mukavana lisänä sivun työpöytäversiolle, jos sitäkään. Olihan internetin käyttö mobiililaitteilla vielä muutama vuosi sitten vähintäänkin haastavaa huonojen yhteyksien ja mobiiliselainten vuoksi. Nyt kun asiat ovat

toisin, kehittäjien kohdelaitteiden prioriteettijärjestys voisi olla toisin päin. Mitä hyötyjä voitaisiin saavuttaa suunnittelemalla web-sivusto ajatellen ensin mobiilia?

Wreckamovien mobiiliversiota suunnitellessa kävi selväksi, että pienempi näyttö vaatii ominaisuuksien karsimista. Suunnittelun alkaessa keväällä 2011 keskiverto älypuheli-
men näytön resoluutio oli 320x480 pikseliä. Kun standardiksi muodostunut 1024x768 pikselin resoluution näytölle optimoitu sisältö siirretään noin 80 prosenttisyksikköä pienempään tilaan on selvää, että kaikkea ei saa mahtumaan mukaan. Vain tärkein sisältö siis kelpuutettiin. Lopulta oli mielenkiintoista huomata, kuinka paljon mukavampi ja selkeämpi oli käyttää mobiiliversiota kun turha ja ylimääräinen ison näytön pikselintäyte oli poistettu. Prosessi auttoi todella hahmottamaan mikä sisältö palvelulle on kaikkein oleellisinta.

Suunnittelija Luke Wroblewski on mobiili edellä -web-suunnittelutavan (mobile first) aktiivinen puolestapuhuja. Tapa pakottaa kehittäjän priorisoimaan palvelunsa mobiililaitteille luontaisten rajoitusten ehdoilla. Se valmistaa kehittäjät mobiiliin kasvussa nouseviin mahdollisuuksiin ja kannustaa innovoimaan mobiililaitteiden uusien ominaisuuksien innoittamana ja näkemään sitä kautta palvelunsa uudessa perspektiivissä (Wroblewski 2011a, 1). Mitä nämä mobiililaitteiden rajoitukset sitten ovat, jotka auttavat web-palvelun suunnittelussa? Kuten edellä totesimme, voi näytön pienempi koko auttaa sisällön hahmottamisessa ja ominaisuuksien priorisoimisessa.

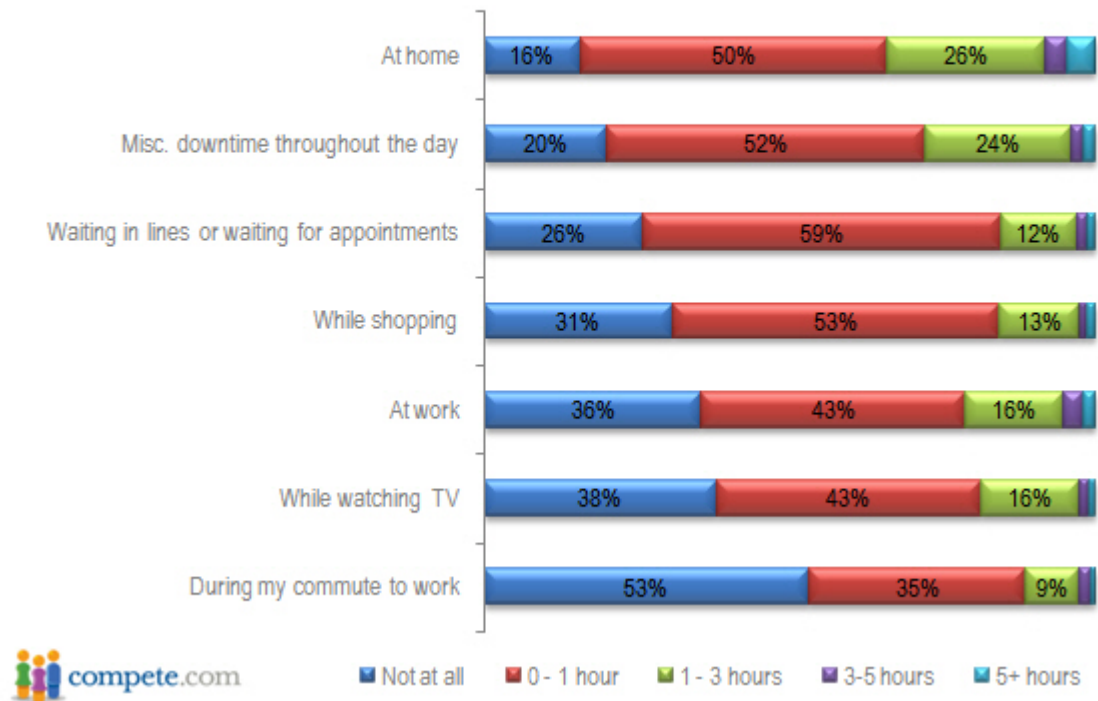
Suorituskyvyn maksimointi on tärkeä osa web-suunnittelua. Mobiili edellä -suunnittelutavalla suunniteltaessa sen arvo nousee yhä korkeammalle. Vaikka 3G-yhteyksien nopeudet ja katealueet ovatkin kasvaneet, ne eivät vedä vertoja kiinteille laajakaistayhteyksille. Samoin on otettava huomioon erilaiset puhelinoperaattoreiden tiedonsiirtosopimukset. Kaikilla käyttäjillä ei ole käytettävissään rajatonta kiinteää so-
pimusta, vaan operaattori veloittaa käyttäjää siirretyn tiedon mukaan. Jos sivuston suorituskyky kehitetään näiden syiden pakottamana tarpeeksi tehokkaaksi mobiiliverkolle, käyttö tietokoneella on luonnollisesti salaman nopeaa. (Wroblewski 2011a, 22–24.)

Koska mobiililaitte voi olla lähes aina mukana, aika ja paikka ovat voimakkaasti sidoksissa sen käyttötilanteisiin. Siksi kehittäjän on mobiililaitteelle suunnitellessaan ajateltava

eri tavalla sitä, miten asiakkaat sivustoa käyttävät päivän aikana. On kehitettävä tuote, jota voi käyttää missä vain ja milloin tahansa.

How much time, throughout a typical day, do you spend using your mobile device?

(Compete's Quarterly Smartphone Intelligence, Jan-Feb 2010, n = 1246)



Kuva 7. Kuinka paljon ihmiset käyttävät aikaa mobiililaitteen ääressä päivän erilaisissa tilanteissa (Complete.com 2010.)

Kuvassa 7 on compete.comin tekemän tutkimuksen tulokset. Tutkimuksessa kysyttiin 1246 ihmiseltä, kuinka paljon he käyttävät mobiililaitteitaan päivän aikana erilaisissa ajan ja paikan määräämissä konteksteissa. Taulukon kontekstit ovat ylhäältä alas: kotona, erilaisina häiriöaikoina päivän aikana, odottaessa jonossa tai odottaessa tapaamista, ostoksilla, töissä, TV:tä katsellessa ja työmatkalla. Vaikka otanta on pieni, voidaan tuloksista tehdä suhteellisen osuvia oletuksia.

Mobiililaitteita käytetään lähes koko päivän aikana ja mitä erilaisimmissa tilanteissa. Näitä tilanteita yhdistää kuitenkin oletettavasti se, että laite harvoin saa osakseen käyttäjän täyttä huomiota. Mobiililaitteille kehitettäessä on siis huomioitava tällaiset tilanteet ja pyrittävä suunnittelemaan tuote niin, että se on aina helposti ymmärrettävissä. (Wroblewski 2011a, 24–29.)

Mobiililaitteella käyttö tapahtuu yleensä myös lyhyissä sarjoissa päivän aikana, jolloin on oltava saatavilla nopeasti ajantasaista tietoa ja relevantteja päivityksiä. Palvelut kuten Twitter ja Facebook ovatkin saavuttaneet huomattavaa suosiota mobiiliratkaisuillaan.

Mobiili edellä -suunnittelutavassa ei tarvitse kuitenkaan pelkästään keskittyä mobiililaitteiden rajoitusten pakottamaan yksinkertaistamiseen. Uusien laitteiden kapasiteetit ja ominaisuudet sekä käyttökontekstit avaavat täysin uusia ovia web-kehitykseen. Paikantimet, kamerat, kiihdytysensorit ja kosketusnäytöt ovat vain esimerkkejä siitä, mitä mobiililaitteet pystyvät tarjoamaan.

5.2 App vai Web?

Mobiilipalvelun voi tarjota web-selaimella käytettäväksi, tai sitten natiiviksi ohjelmaksi mobiililaitteelle. Esimerkiksi Facebook tarjoaa palvelustansa natiivin ohjelman kaikille yleisimmille älypuhelimien käyttöjärjestelmille ja yli 2500 mobiililaitteelle selaimella käytettävän mobiiliversionsa lisäksi (Facebook 2012).

Natiivilla ohjelmalla on käytössään laitteen systeemiresurssit, joihin web-aplikaatiot eivät pääse käsiksi. Sillä pystyy siis tuottamaan käyttöliittymään sulavampia siirtymiä ja vuorovaikutuksia. Tällaisten käyttöjärjestelmän efektien imitoiminen selaimessa tuottaa usein huomattavaa jumiintumista ja laahaavan käyttökokemuksen. Pääsy laitteen ominaisuuksiin, kuten puhelinmuistioon, tekstiviesteihin ja kameraan on myös mahdollista natiiville ohjelmalle, mutta nykyisillä mobiiliselaimilla se ei onnistu. (Wroblewski 2011a, 14–15.)

Jos mobiilipalvelu tarvitsee käyttöönsä pääsyn puhelimen laitteistoon, mahdollisuuden tausta-ajoon tai esimerkiksi pääsyn ohjelmistokauppaan kuten Applen App Storeen, on natiivi ohjelma tarpeellinen.

Facebook tarjoaa palvelunsa useimmille mobiilialustoille. Tapaus on kuitenkin harvinaisen. Aivan jokainen ei siihen pysty. Eri laitevalmistajien käyttöjärjestelmät tukeutuvat yleensä erilaiseen kehitysympäristöön. Esimerkiksi Applen iOS vaatii Objective C:n ja Googlen Android Javan. Windows Phone 7 taas vaatii Silverlightin. On siis hyvin vaike-

aa ja resursseja kuluttavaa kehittää jokaiselle alustalle oma applikaationsa puhumatta-
kaan niiden ylläpitämisestä (Wroblewski 2011a, 15–16). Vaikka Facebook tähän pys-
tyykin, tarjoaa se silti myös web-pohjaisen mobiilipalvelun.

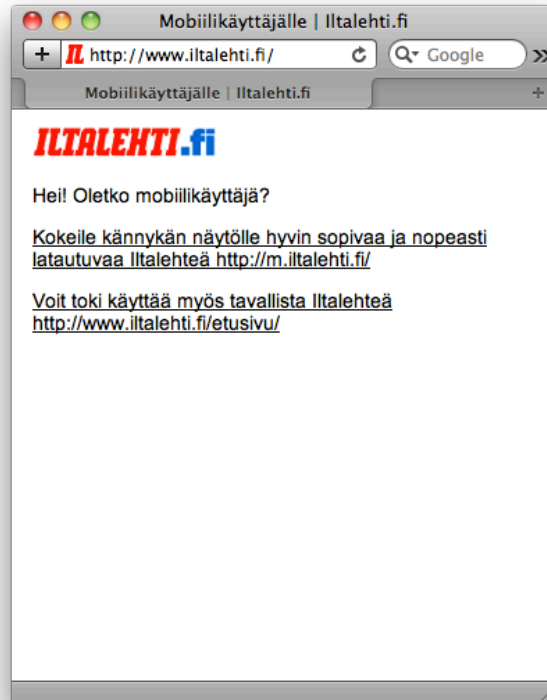
Jos palvelu tarjoaa materiaalia verkossa, ihmiset voivat jakaa linkkejä sinne vaikka säh-
köpostilla tai sosiaalisessa mediassa. Hakukoneet ja web-sivut tarjoavat linkkejä palve-
luun. Linkit eivät kuitenkaan avaa ohjelmia vaan web-sivuja (Grispy 2011). Jos palvelu
ei tarjoa mobiili web-ratkaisua, käyttäjä ei saa toivomaansa käyttökokemusta seurates-
saan linkkiä mobiililaitteellaan. Vähin mitä voidaan tarjota on mahdollisuus ladata natiivi
mobiiliohjelma web-palvelusta. Verkkopalvelun käyttäjän ei myöskään tarvitse ladata
ohjelmistopäivityksiä, sillä ne tapahtuvat palvelinpuolella. (Wroblewski 2011a, 15–16.)

5.3 Responsiivinen web-suunnittelu

Testasimme opinnäytetyötämme varten kahden suosituksen ilta- ja iltapäivälehdet, Iltalehden ja
Iltasanomien verkkosivuja mobiililaitteella. Otimme nämä kaksi sivua esimerkiksi, sillä
niiden mobiilikäyttö on oletettavasti suosittua, mutta kokemuksesta tiesimme niiden
mobiiliversioiden saavutettavuudessa olevan ongelmia. Käytettävä laite oli Samsung
Galaxy 2, jonka 4,3 tuuman näyttö on suhteellisen suurikokoinen puhelimelle, mutta
kuitenkin liian pieni tietokoneen näytölle optimoidun sisällön selaamiseksi.

Iltalehti ja Iltasanomat tarjoavat molemmat verkkopalveluaan alidomainilla (subdo-
main) työpöytäversiosta eristettynä mobiiliversiona (esim. m.iltasanomat.fi). Jos Iltas-
anomien sivulle hakeutuu kirjoittamalla selaimen osoitekenttään "www.iltasanomat.fi",
käyttäjä ei ohjautu automaattisesti mobiiliversioon. Näyttää siis siltä, että Iltas-
anomien palvelua ei ole ollenkaan suunniteltu reagoimaan sitä selaavaan laitteeseen.
Käyttäjän on joko tiedettävä tai arvattava "m" -alidomaini tai ladattava raskas etusivu
kokonaan ja löydettävä mobiiliversioon johtava linkki, mikäli haluaa laitteellensa opti-
min lukukokemuksen.

Iltalehti ohjaa vastaavassa tilanteessa laskeutumissivulle, jossa käyttäjän on päätettävä haluaako käyttää mobiilisivua vai tavallista Iltalehteä (ks. kuva 8). Tämä yksi välietappi mobiiliversioon ohjaukseen ei ole mikään katastrofi käytettävyyden kannalta, mutta se herättää kuitenkin kysymyksen: Jos palvelu on jo todennut asiakkaan käyttävän mobiililaitetta, miksi asiaa tarvitsee tässä vaiheessa enää varmistaa?



Kuva 8. Iltalehden mobiiliversioon laskeutumissivu. Kuvan tilanteessa Safarin user agentiksi on muutettu iPhone sen develop-lisäosan avulla. (Iltalehti 2011.)

Alidomainilla eristetty mobiililaitteelle optimoitu sivu tarjoaa varman mobiilikokemuksen, mikäli osoite on tiedossa. Ongelmana on kuitenkin linkkien jakaminen. Esimerkiksi sosiaalisessa mediassa jaettu linkki Iltalehden uutiseen ei tarjoa hyvää käyttö- ja lukukokemusta kuin ainoastaan sellaisella laitteella, jolle kyseinen osoite on tarkoitettu.

5.3.1 Staattisesta sivusta joustavaksi

Ethan Marcotte on artikkeleissaan ja kirjallisuudessaan puhunut responsiivisesta web-suunnittelusta (responsive web design). Tämän nimen hän on lainannut arkkitehtuurin puolelta. Responsiivisesta arkkitehtuurista rakenteet ja pinnat reagoivat ympäristönsään tapahtuviin muutoksiin, kuten ääniin tai ihmisten liikkeisiin. Responsiivisessa web-

suunnittelussa ideana on kehittää sivu niin, että se reagoi sitä näyttävän laitteen ominaisuuksiin, kuten näytön kokoon ja asentoon, ja muuttaa muotoaan venymällä ja elementtejä siirtelemällä laitteen ominaisuuksien vaatimalla tavalla ja luo näin hyvän käyttökokemuksen laitteelle kuin laitteelle. Käyttäjää ei siis tarvitse ohjata uuteen osoitteeseen, jossa sivun sommittelu on optimoitu käyttäjän laitteelle, vaan sama HTML-tiedosto voidaan tarjota lukemattomille erilaisille laitteille ja juuri niiden ominaisuuksia vastaavalla, parhaalla mahdollisella sommitelmalla. (Marcotte 2011.)

Kuvassa 9 on Ethan Marcotten esimerkkisivu responsiivisesta sommitelmasta. Sivun toteutus on näin, että selaimen kokoa eli viewportia muutettaessa sivun rakenne mukautuu koolle relevantiksi.



Kuva 9. Responsiivinen sommitelma muuttaa muotoaan selaimen kokoa muutettaessa.

Teknisesti responsiivinen suunnittelu perustuu kolmeen perustekijään: joustavaan runkoon (fluid grid), joustaviin kuviin ja mediaan sekä mediakyselyihin (media query, tekijän suomennos). Joustava runko saadaan aikaiseksi määrittämällä sivun elementtien koot relaatiivisesti prosenttiarvoilla staattisten pikseliarvojen sijasta. Haluttu suhde saadaan jakamalla kohde sen ympäröivällä elementillä. On tärkeää, että kaikki leveysarvot, mukaan lukien padding- ja margin-arvot määritellään relaatiivisesti, jotta joustava runko saadaan aikaiseksi. (Marcotte 2011, 13–42.)

Joustavat kuvat voidaan toteuttaa monella tapaa. Marcotte suosii responsiivisessa suunnittelussa CSS-tyylissä kuvan 100 %:n min-width-arvoa ja sitä kehystävää div-elementtiä, joka määrittää kuvan koon. Kehyksen voi määrittää myös leikkaamaan (crop) kuvan niin, että kehyksen reunojen ylimenevä osa jää piiloon. Tällöin kehyksen overflow-määre saa CSS:ssä arvokseen "none". Kuvien tarkkuus kuitenkin kärsii alkuperäiskokoa venytettäessä ja yksityiskohtaiset kuvat, kuten tarkat informaatiografiikat,

voivat olla hyvä tarjota erilaisina ja erikokoisina versioina laitteen ruudun koosta riippuen. On olemassa myös valmiita scriptejä, jotka vähentävät kuvien sumentumista koon vaihtuessa. (Marcotte 2011, 42–64.)

5.3.2 Mediakyselyt

Pelkkä venyvä rakenne ei riitä responsiivisessa web-suunnittelussa, sillä sivun venyessä tiettyyn pisteeseen sivurakenne hajoaa ja elementtien asettelu ei enää toimi. Kuvat suttaantuvat ja yleinen käyttökokemus huononee.

Varsinaisen responsiivisuuden web-sommitteluun tekee CSS3-määritelmän mediakysely-ominaisuus. Se laajentaa CSS2:n mediatyypin tarkastus-ominaisuutta, joka antoi mahdollisuuden määrittellä CSS-tyylit erikseen esimerkiksi tulostimelle ja näytölle. Mediakyselyissä voidaan määrittellä tyylit myös kyseisen mediatyypin tarkempien ominaisuuksien, kuten näytön leveyden tai rotaation mukaan. (W3C 2010.)

Mediakyselyt toimivat useimmilla moderneilla selaimilla. Desktop-selaimet kuten Safari 3+, Chrome, Firefox 3.5+ ja Opera 7+ ja mobiiliselaimet kuten Opera mobile ja mobile WebKit, kaikki tukevat sitä natiivisti. Microsoft on luvannut sisällyttää sen Internet Exploreriin, joka ei vielä tue mediakyselyitä (Marcotte 2011).

Esim. 2.

```
<link rel="stylesheet" href="tyyli.css" media="screen and  
  (min-width: 769px)>  
<link rel="stylesheet" href="mobiili.css" media="screen  
  and (max-width: 768px)>  
<link rel="stylesheet" href="vaakataso.css" media="screen  
  and (max-width: 540px and orientation: landscape)>
```

Esimerkissä 2 määritellään CSS-tyylit erikseen isolle ja pienelle näytölle HTML-tiedostossa. Ensimmäisellä rivillä määritetään sivulle tyyli.css-tiedostossa määritelty tyylit sillä ehdolla, että näytön leveys on vähintään 769 pikseliä. Toisella rivillä taas määritetään tyyliksi mobiili.css, mikäli näytön leveys on korkeintaan 768 pikseliä. Viimeisellä rivillä taas määritetään vaakatasossa olevalle maksimissaan 540 pikseliä leveälle laitteelle omat tyylimääritelmät.

Esim. 3.

```

@media screen and (max-width:768px) {
    div.laatikko{
        float:none;
    }
}

```

Esimerkissä 3 mediakysely tapahtuu @media-kentän arvossa. Ensimmäisenä ehtona määritellään mediatyypiksi screen eli näyttö ja sen jälkeen annetaan ehto sen maksimiksi tai minimileveydelle. Mediakysely on sijoitettu suoraan CSS-tiedostoon. Lohkossa määritetty tyyli otetaan huomioon vain, jos mediakyselyn ehto toteutuu.

Mediakyselyjä hyödynnettäessä on kuitenkin otettava huomioon WebKit-selaimissa esiintyvä meta-elementti viewport. Vaikka esimerkiksi iPhone 3G:n näytön resoluutio on 480 x 320 pikseliä, sen web-selain Mobile Safari näyttää oletuksena sivun 980 pikselin leveydellä ja kutistaa sen laitteen näytön leveyttä vastaavaksi (Marcotte 2011, 80). Wreckamovien sommitelman keskitetty sisältö on määritetty 1020 pikselin levyiseksi. Mobile Safari kutistaa sivun niin, että se mahtuu melkein kokonaan 320 pikselin levyiselle näytölle. Luonnollisesti 980 pikselin levyinen sommitelma mahtuisi näytölle kokonaan.

Mikäli haluamme mediakyselymme ehtolauseen toimivan haluamallamme tavalla, on asetettava viewport-tagin kumoamaan Mobile Safarin 980 pikselin oletusleveys. Voimme esimerkiksi pakottaa viewportin leveyden 320 pikseliin esimerkin 4 tavalla.

Esim. 4.

```

<content="width=320" />

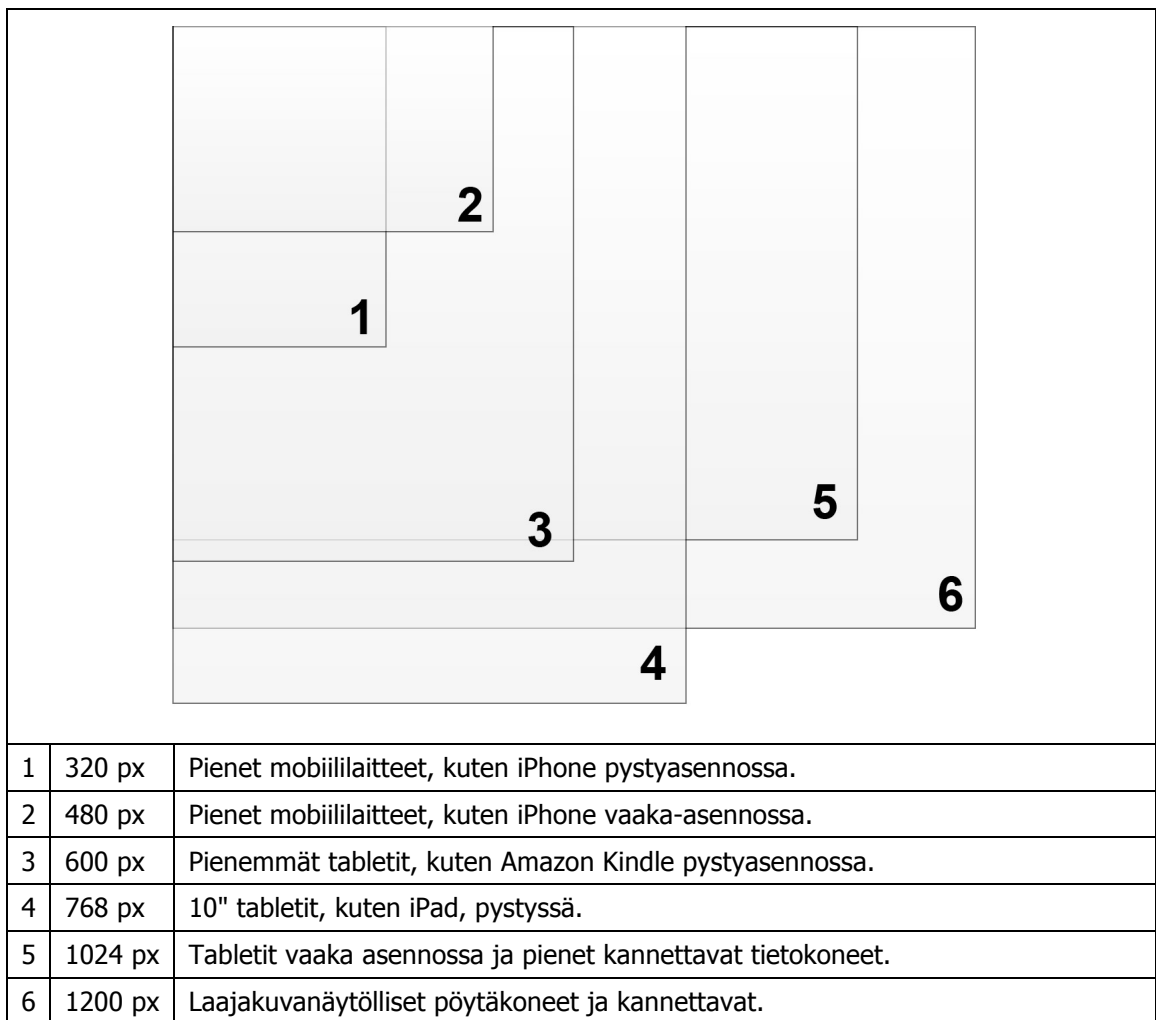
```

Viewport voidaan määritellä kuitenkin joustavammallakin tavalla vastaamaan sivua näyttävän laitteen omaa resoluutiota, kuten esimerkissä 5.

Esim. 5.

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

Esimerkin 5 initial-scale merkitsee sivun zoom-tasoa. Tässä se on määritetty täyteen kokoon. Viewportin leveys saa arvon: "device-width". Tämä merkitsee sitä, että esimerkiksi iPhone 3G näyttää sivun pystyasennossa 320 pikselin levyisenä ja kallellaan 480 pikselin levyisenä. iPhone 4 käyttäytyy täsmälleen samalla tavalla, sillä vaikka sen näytön tarkkuus on 640x960, sen device-width-arvo on määritelty samoin kuin iPhone 3G:ssä 320x480. (Firtman 2011.)

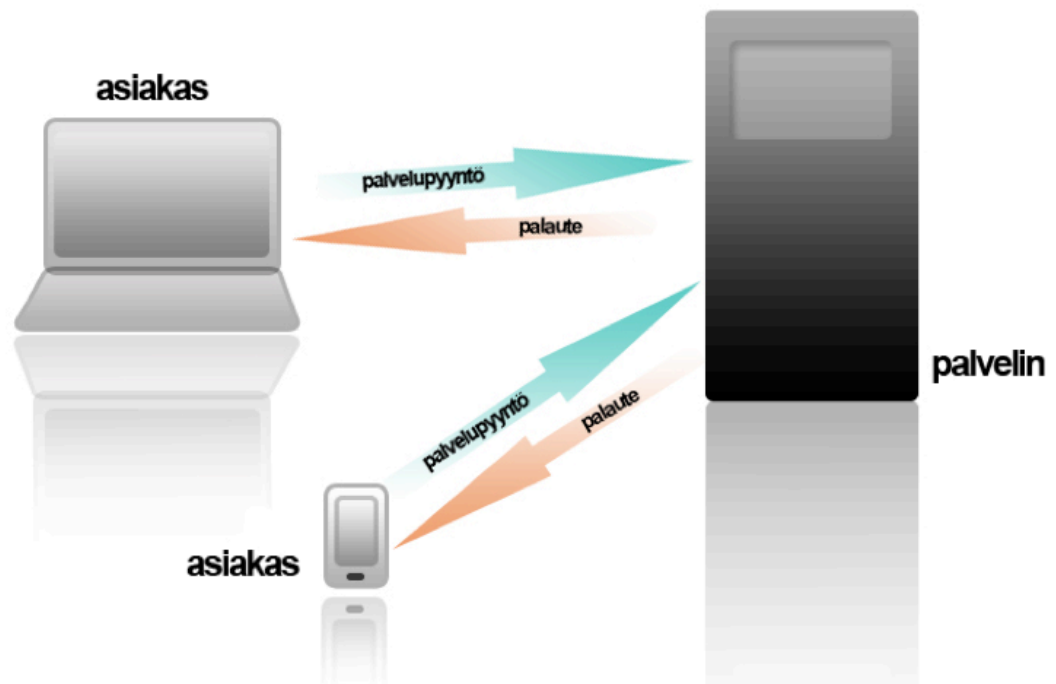


Kuva 10. Erilaisten laitteiden näyttöjen leveyserot havainnollistettuna ja laiteryhät, joita koot koskevat.

Tämänhetkisten yleisimpien mobiililaitteiden ja tietokoneiden ominaisuuksien perusteella voitaisiin toimia esimerkiksi kuvassa 10 havainnollistettujen näytön leveyttä koskevien rajojen mukaisesti. Joustava runko takaa sen, että rajojen väliin jäävät laitteet eivät jää huomioimatta.

5.3.3 Laitteen tunnistus palvelin- ja asiakaspuolella

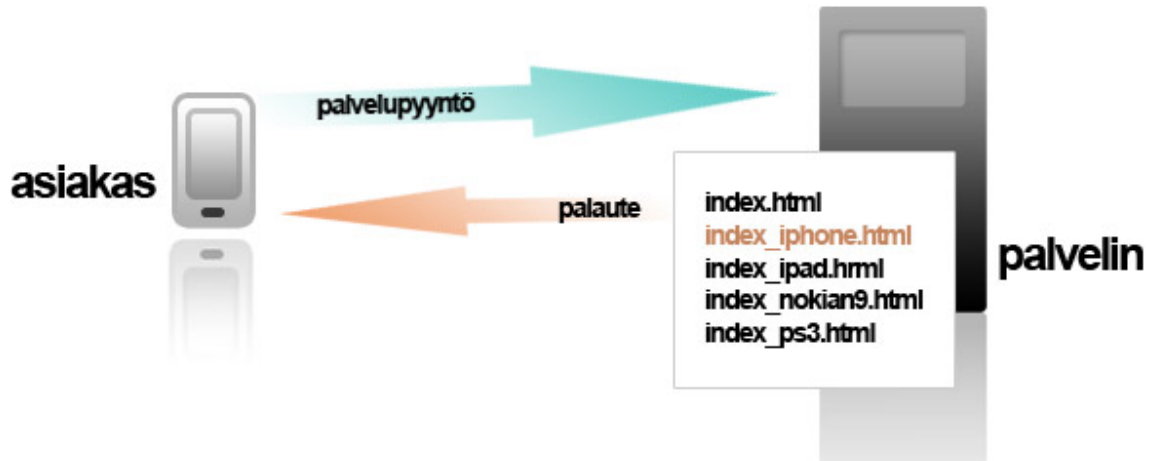
Kuva 11 esittää asiakkaan (client) ja palvelimen (server) suhdetta. Web-sovellus sijaitsee palvelimella, joka toimii palveluntarjoajana. Asiakkaana toimii palvelua käyttävä selainohjelma, joka ottaa yhteyden palvelimelle. Palvelin hyväksyy yhteyden ja käsittelee palvelupyynnön. Palautteeksi se lähettää web-sivun asiakkaalle.



Kuva 11. Asiakas ja palvelin.

Responsiivinen web-suunnittelu mediakyselyiden avulla perustuu siis siihen, että sivua näyttävän laitteen ominaisuudet tunnistetaan käyttäjän selaimella eli asiakaspuolella. Kuva 12 havainnollistaa asiakaspuolen ratkaisua. Asiakkaalta saatu palvelupyynnö sisältää laitteen tunnistetiedot (user agent), joiden perusteella suoritetaan uudelleenohjaus laitetta vastaavaan näyttöpohjaan (template). Tästä on etuna se, että yhdellä hyperlinkillä voidaan tarjota hyvä käyttökokemus lukuisille erilaisille laitteille yhdellä HTML-

pohjalla, mutta toisaalta saattaa olla, että käyttäjä joutuu lataamaan ylimääräistä piilotettua sisältöä, laajoja CSS-tyylitiedostoja tai JavaScript-kirjastoja. Tämä voi kehkeytyä suoritusongelmaksi. (Wroblewski 2011b.)



Kuva 12. Palvelinpuolen ratkaisu.

Toinen lähestymistapa mobiilille suunnitteluun on käyttää laitteen tunnistamiseen palvelinpuolen ratkaisua. Palvelimella sijaitseva mediakyselyn sisältävä sivupohja lähetetään asiakkaalle, joka tulkitsee mediakyselyn omien laiteominaisuuksiensa perusteella (ks. kuva 13).



Kuva 13. Asiakaspuolen ratkaisu.

Tällä ratkaisutavalla voidaan varmistaa, että selainlaite ottaa vastaan ainoastaan sille oleellista dataa jo ennen kuin sitä lähetetään eteenpäin. Palvelinpuolen ratkaisut perustuvat kuitenkin usein user agentin eli selaimen yksilöllisen tunnistetiedon tarkastami-

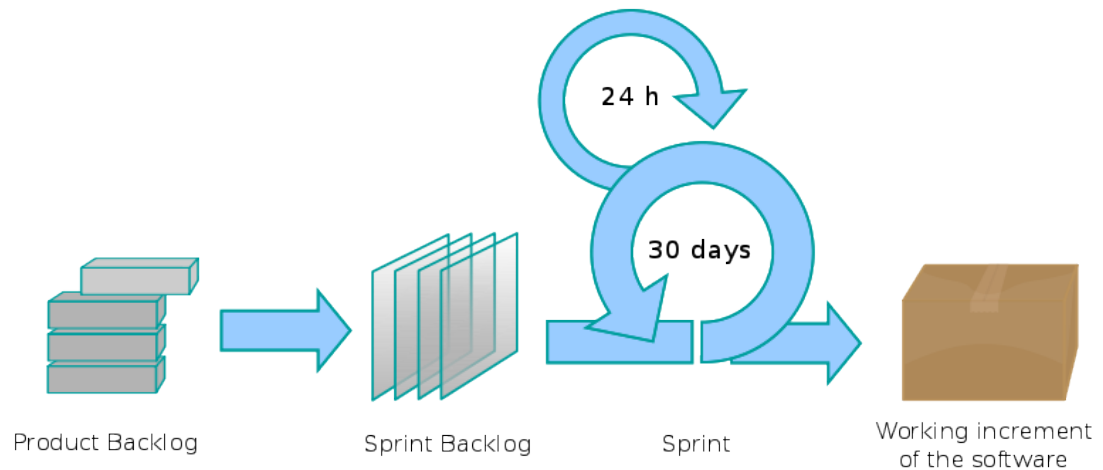
seen ja uudelleen ohjaamiseen kunkin user agentin yksilöityyn sivupohjaan. Nämä pohjat joudutaan luomaan jokaiselle optimointia vaativalle laiteryhmälle erikseen, ja ne saattavat sisältää paljon samaa koodia. (Wroblewski 2011b.)

6 Työmenetelmät ja kehitysympäristö

Tässä luvussa käsittelemme työskentelymenetelmiä, joita käytimme mobiiliversion toteutuksen aikana. Työmme koostui jo aikaisemmin hyvin pitkälti samoista aiheista, joten menetelmien opiskeluun ja totutteluun ei kulunut aikaa. Ruby on Railsin kanssa käytettynä Scrum-menetelmä on toimiva ja sopiva yhdistelmä ketterälle kehitykselle.

6.1 Scrum-menetelmä

Scrum on suhteellisen uusi projektinhallinnan menetelmä, jota käytetään yleisimmin ketterässä ohjelmistokehityksessä (agile development).



Kuva 14. Scrum-prosessin kuvaus (Torak 2012).

Scrumin tavoitteena oleva tuote rakentuu pikku hiljaa täydellisemmäksi ja valmiimmaksi, useiden toteutuskierrosten aikana, kuten kuvassa 14 näytetään. Tällaista toteutuskierrosta kutsutaan sprintiksi (sprint). Sprintti kestää yleensä kahdesta viikosta kuukauteen. Jokaisen sprintin alussa on palaveri, jossa määritellään sprintin tavoitteet ja teh-

täväksi valitaan vain niitä asioita, jotka ovat välttämättömiä projektin onnistumisen kannalta. Tuotteella on työlista (product backlog), josta valitaan jokaiseen sprinttiin erikseen sprintin työlista. Sprintin lopussa tuotteesta on olemassa toimiva versio sprintin aikana työstetystä ominaisuudesta (working increment).

Scrum-tiimiin kuuluu yleensä noin 7 jäsentä. Yksi jäsenistä on Scrum-mestari (Scrum master), jonka tehtävänä on työryhmän työrauhan säilyttäminen ja työskentelyn valvominen. Tiimin ulkorajapinnassa toimii tuoteomistaja (product owner), joka on tuotteen asiakkaiden edustaja. Hän määrittelee tuotteen vaatimukset ja aikataulun, vastaa projektin kannattavuudesta sekä ohjaa sprinttien alussa pidettävät suunnittelupalaverit. Scrum-tiimit ovat itseohjautuvia ja voivat työskennellä parhaaksi katsomallaan tavalla.

Myös tämä opinnäytetyö toteutettiin hieman sovelletulla Scrum-menetelmällä. Työryhmässämme oli meidän lisäksi mukana tuoteomistaja, jonka kanssa käytiin läpi projektin tavoitteet ja ominaisuudet, joita otetaan mukaan. Projektin toteutuksen aikana kävimme viikoittain läpi tiimin palaverissa mitä tuli menneen viikon aikana tehtyä ja mitä oli vielä tekemättä.

6.2 Ruby on Rails ja MVC

Ruby on Rails on avoimen lähdekoodin sovelluskehys (framework), joka on kehitetty ohjelmoijan työn helpottamiseksi ja tuottavuuden kasvattamiseksi. Rails pohjautuu MVC-arkkitehtuuriin (Model-View-Controller), joka mahdollistaa sivuston toiminnallisuuden erottamisen käyttöliittymästä ja selkeyttää sovelluksen koodin sijoittamista.

Malli (model) edustaa sovelluksen informaatiota ja määrittelee sen, miten sovellus voi muokata tietokannan tietoja. Railsissa käytetään usein yhdessä mallissa yhtä tietokannan taulua.

Näkymässä (view) määritellään sovelluksen graafinen käyttöliittymä. Railsissa useimmiten käytössä on HTML-pohjaiset näkymät, joissa käytetään upotettua Ruby-koodia. Näkymät siirtävät tarvittavat tiedot selaimelle tai muulle työkalulle, jolla käytetään sovellusta.

Ohjaimet (controller) yhdistävät mallin ja näkymän. Railsissa ohjaimet käsittelevät selaimelta tulevat kutsut, hakevat tarvittavat tiedot mallilta ja siirtävät ne näkymälle, joka näyttää tiedot selaimelle.

7 Toteutus

Tämä luku kertoo vaiheista, joita kävimme läpi työskennellessämme ja toteuttaessamme Wreckamovie.com-palvelun mobiiliversiota. Lähtökohdat koostuivat työnantajan kanssa läpikäydyistä ja kehitellyistä ohjeistuksista.

7.1 Suunnittelun ja toteutuksen lähtökohdat

Wreckamovie.com-palvelun mobiiliversion suunnittelussa lähdettiin liikkeelle rajaamalla pääasialliseksi kohdelaitteeksi Applen iPhone ja sen selain Mobile Safari. Sen lisäksi otettiin huomioon Android-laitteet, joiden web-ominaisuudet todettiin hyvin samankaltaisiksi kuin iPhonessa. Pelkistetysti siis päätettiin rakentaa palvelusta versio, joka tarjoaisi hyvän käyttökokemuksen kosketusnäytölliselle web-selainlaitteelle, joka olisi resoluutioltaan lähimain 320x480 pikseliä.

Wreckamovien kehityksessä käytetään Ruby on Rails -sovelluskehystä. Sen käyttämän MVC-mallin yhtenä teemana on erottaa käyttöliittymä ja toiminnallisuus toisistaan. Mobiiliversiota suunnitellessamme päätimme siis ottaa lähestymistavaksi keskittyä näkymiin ja jättää toiminnallisuudet pääosin samaksi.

Käyttöliittymän suunnitteluun haettiin esimerkkiä Facebookin ja YouTubeen toimivista mobiiliratkaisuista. Nämä suositut ja laajalti mobiililaitteilla käytössä olevat palvelut ovat luoneet käyttöliittymäratkaisuita, jotka ovat tulleet useille käyttäjille tutuiksi. Tällaista käyttöliittymän opittavuutta parantavaa seikkaa ei haluttu sivuuttaa.



Kuva 15. Vasemmalla Facebookin ja oikealla YouTube:n mobiiliversion näkymä iPhoneella. Kuva-kaappaukset ovat vuodelta 2011.

Suunnittelussa päätettiin myös tutkia valmiita vapaan lähdekoodin käyttöliittymäliitännäisiä, jotka on kehitetty nimenomaan iPhoneen käyttöön soveltuviksi ja joista osa muikailee vahvasti iOS:n natiivia käyttöliittymää. Mobiiliversiosta haluttiin karsia täysversion toimintoja ja keskittyä palvelun olennaisimpien toimintojen helppoon käyttöön mobiililaitteella.

7.1.1 Esimerkinä YouTube

Kuvassa 15 näkyy YouTube:n iPhone-version ylänavigaatio, joka on jaettu kolmeen osaan ikonein. Hakuun pääsee suurennuslasista, etusivulle YouTube-logosta ja päänavigointisivulle ruudukkoikonista. Päävalikkoon pääsee siis jokaiselta sivulta yhdellä näpäytyksellä. YouTube ei yritä ahtaa laajaa päänavigointia kokonaisuudessaan jokaiselle sivulle. Ratkaisu on toimiva ja yksinkertaistaa käyttöä pienellä näytöllä. Vastaavaa päänavigaation siirtämistä omalle sivulle on käytetty myös Wreckamovien mobiilisivun toteutuksessa. YouTube:n hakupalkki aukeaa samalle sivulle javascriptin avulla, eikä turhaa latausta uudelle sivulle siirtymisestä tapahdu.

YouTuben päänavigointisivulla haku- ja valikkoikonit katoavat ylänavigaatiosta ja tilalle tulee kirjautumispainike. Päävalikon navigointi-ikonit ovat isoja ja grafiikaltaan kosketukseen kutsuvia. Ikonit ovat jo itsessään toimintaansa kuvaavia, mutta käytettävyyttä on helpotettu seliteteksteillä. Päänavigaationsivu poikkeaa sommitelmaltaan YouTuben normaalista pystysuuntaisesta navigoinnista ja ikonit on aseteltu sivulle ruudukkoon. Poikkeama helpottaa muistettavuutta ja erottaa päänavigaationsivun tehokkaasti muusta sisällöstä.

7.1.2 Esimerkkinä Facebook

Kuvassa 15 näkyy myös, kuinka Facebookin ylänavigaatio on jaettu kolmeen linkkiin: notifiikaatiot, sivun otsikko, joka vie pääsivulle sekä haku. Näiden alta löytyy neljä päänavigaation linkkiä: koti, oma profiili, kaverit ja postilaatikko. Jokaiselta sivulta löytyy myös alavalikko, josta löytyy kolme linkkiä. Ylänavigaation linkit pysyvät aina samassa paikassa otsikon, notifiikaatio-linkin ja haku-linkin kanssa. Päänavigaation elementit ovat hieman liian ahtaasti ollakseen eniten käytettyjä linkkejä, minkä takia päädyimme käyttämään YouTuben käyttämää ratkaisua: päänavigaatiota omalla sivullaan.

7.2 Käyttöliittymän suunnittelu

Safarin develop-lisäosa antaa mahdollisuuden simuloida kehitettävää sivua muilla selaimilla ja laitteilla. Tarjolla on myös simulointi iPhoneille ja iPadille. Tämä helpottaa kehitystyötä antamalla mutkattoman tavan vaihdella user agentista toiseen käyttöliittymää testatessa. Toinen käyttämämme testausympäristö oli Applen oma iPhone-simulaattori Macille, joka pystyy simuloimaan iPhoneen eri versioita ja iPadia. Se antaa mahdollisuuden kääntää simuloitavan laitteen orientaatiota. Nämä työkalut tarjoavat hyvät testausmahdollisuudet käyttöliittymäsuunnitteluun, jos kehityksen kohdelaitetta ei ole käytettävissä.

7.2.1 Ominaisuuksien siirtäminen mobiiliversioon


Mobiililaitteiden pienempi näyttötila ei pysty tarjoamaan yhtä paljoa ominaisuuksia ja linkkejä kuin suuri laajakuvanäyttö ilman, että käytettävyys huononee merkittävästi. Päätimme siis karsia tiettyjä toimintoja ja sisältöjä, joita ei mobiililaitteella sen näytön





koon takia välttämättä pystyttäisi näyttämään. Oli syytä pohtia niitä käyttötilanteita, joissa käyttäjä haluaa nopean pääsyn palveluun, vaikka ei ole tietokoneen ääressä.

Rajauksen lähtökohtana pidimme, että mobiiliversioon säilytettävien toimintojen tulisi ennen muuta tukea palvelun pääasiallista aktiviteettiä: shottien ja taskien antamista ja selaamista. Tämän lisäksi halusimme säilyttää palvelun sosiaalista aktiviteettiä ja käyttäjien välistä viestintää tukevia ominaisuuksia. Palvelun normaalia käyttöä ei haluttu unohtaa, vaan palvelussa työstettävien tuotantojen helppo selaaminen ja esittelyt trailerivideoineen haluttiin tarjota myös mobiilikäyttäjille. Sisällön yksinkertaistamisessa päädyttiin ratkaisuun, että tuotantojen ja käyttäjätilien laajat hallintaominaisuudet jätettiin pois. Poisjätettävien ominaisuuksien käyttö haluttiin kuitenkin mahdollistaa mobiililaitteillekin tarjoamalla sivuston alalaitaan linkki palvelun täyteen versioon.

Yhtenä isoimmista ominaisuuksista, joita jätettiin pois mobiiliversiosta oli tuotantojen luominen. Ominaisuuden käyttötilanne nähtiin selvästi harkittuna toimintana, joka ei vaadi välitöntä saavutettavuutta. Tuotannon luominen vaatii paljon käyttäjäsyötteistä dataa, joka koettiin raskaaksi suorittaa mobiililaitteella. Näistä syistä johtuen ominaisuus päätettiin jättää pois ainakin mobiiliversiion ensimmäisestä versiosta. Tuotantojen tietojen hallintaominaisuudet jätettiin samalla pois. Tuotantojen hallitsijoille jätettiin kuitenkin mahdollisuus välittömään tiedottamiseen, joten blog-toiminto jätettiin käyttöön. Tilanne, jossa tuotannon johtaja tarvitsee nopeasti käyttäjien panosta uudella taskilla nähtiin mahdolliseksi, siksi uuden taskin luominen mahdollistettiin myös mobiiliversiossa.

DASHBOARD MY PROFILE INBOX TODO FAVORITES MY SHOTS MY COMMENTS



-  Upload Picture
-  Edit Profile
-  Change Password
-  Invite

Recent Stuff

You have:

- Blog post from 2 productions
- 8 new productions to check out
- 9 new shots in your productions
- 1 new task in your productions

Settings

Open links in new window

Mark All As Viewed

2 new blog posts

Production	Title	Released	Viewed
Iron Sky: Operation Highjump	New status update coming soon	16 Nov 10:33	✗
Iron Sky: Operation Highjump	Status update: Current status & First proper in-game screenshot	21 Nov 16:15	✗

8 new productions in Wreckamovie

Production	Created	Viewed
Findie - Hikeä, Tekoverta ja Kynneliä	23 Nov 19:23	✗
Crooner	26 Nov 16:05	✗
Please delete	26 Nov 15:49	✗
At Your Command	26 Nov 17:41	✗
Iron Sky - The Exhibition	29 Nov 08:56	✗
Terry Pratchett's Troll Bridge	29 Nov 10:49	✗
Grandmonster	29 Nov 19:23	✗
Wrath	30 Nov 09:24	✗

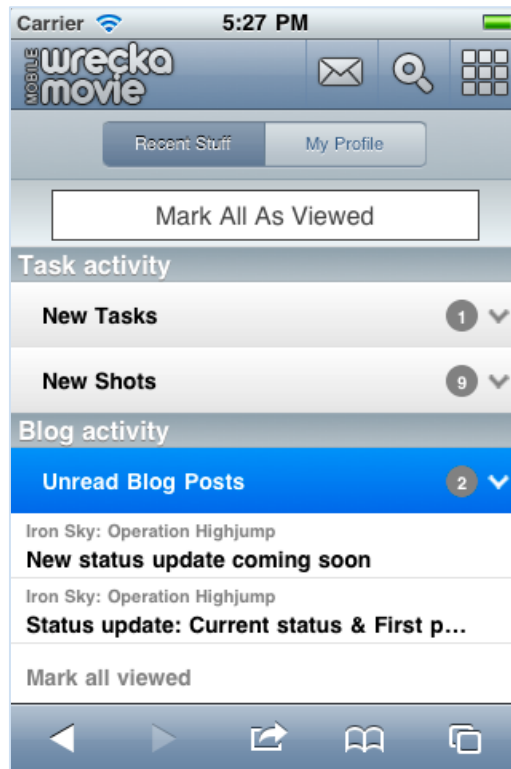
Latest Blog Posts

Iron Sky: Operation Highjump

21 Nov 16:15
 Status update: Current status & First proper in-game screenshot

Kuva 16. MyWreck-näkymä työpöytäselaimella tarkasteltuna (Wreckamovie 2011).

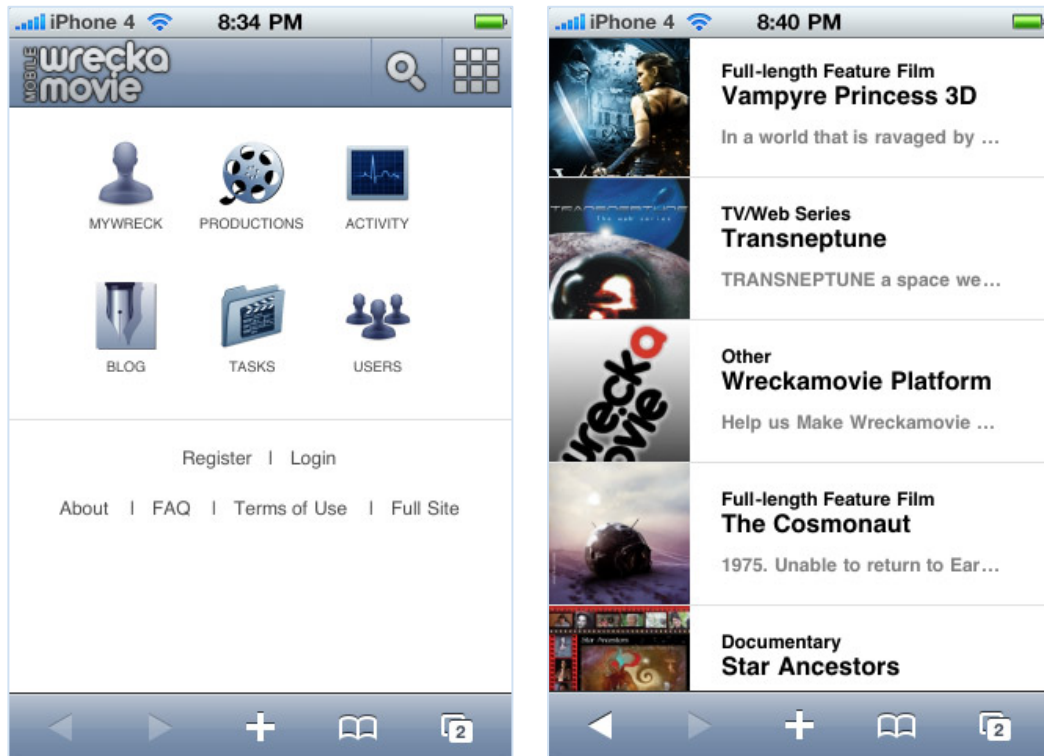
Sivuston käyttäjien kotitila on MyWreck, joka päätyi suurimman ominaisuuskarsinnan kohteeksi. Wreckamovien täydessä versiossa käyttäjä hallitsee MyWreckistä tiliään ja profiilitietojaan ja saa parhaiten tietoonsa kaikki sivulla tapahtunut aktiviteetti, joka erityisimmin koskee juuri häntä (ks. kuva 16). Uudet ilmoitukset näkyvät kaikki omien otsikoidensa alla jatkuvasti. Tämä saattaa aiheuttaa ongelmia jos ilmoituksia on paljon. Muita ominaisuuksia MyWreckissä ovat pääsy privaattiviesteihin, suosikkeihin, todolistaan, sekä omiin kommentteihin ja shotteihin.



Kuva 17. Mobiiliversion MyWreck-sivun recent stuff -näkyvä (Wreckamovie 2011).

Tehdäksemme MyWreckin käytön suoraviivaiseksi ja helppokäyttöiseksi jaoimme sen kahteen osioon: uusimpiin tapahtumiin (recent stuff) ja omaan profiiliin, josta löytyy omat perustiedot ja tuotantojäsenyydet (ks. kuva 17). Tilin ja profiilin hallinta jätettiin pois mobiiliversiosta. Vaikka ominaisuus on toki tärkeä, sitä ei nähty kuitenkaan sellaiseksi toiminnoksi jota käyttäjä tarvitsee välittömästi.

Seuraavaksi esittelemme tärkeimpiä käyttöliittymäratkaisuja kuvakaappauksin. Kuvassa 18 on mobiilisivun päävalikko, joka toimii myös palvelun etusivuna sekä tuotantolistaus. Verratessa päävalikkoa kuvan 15 YouTube:n vastaavaan, voidaan huomata selkeitä yhtymiä.



Kuva 18. Vasemmalla mobiilisivun päävalikko ja oikealla tuotantolistaus (Wreckamovie 2011).

Kuvassa 19 on verrattu työpöytä- ja mobiiliversion tuotantosivun navigaatiota ja tapahtumavirtaussivua. Huomioitavaa on, kuinka sivun sininen päävalikko ja linkki päänavigaatioon pysyy ylälaidassa YouTuben tavoin. Tuotantosivun navigaatio on siirretty sivun ylälaitaan.

Kuvissa 20 ja 21 verrataan palvelun oleellisinta ominaisuutta: taskien näyttämistä ja shottien antamista työpöytä- ja mobiiliversioissa. Shottien antamien ja kommentoimien mobiiliversiossa on pyritty tekemään helpoksi ja samalla logiikalla kuin työpöytäversiossakin. Mobiililaitteen selainten rajoitukset estävät kuitenkin liitetiedostojen lisäämisen. Tykkääminen eli peukun antaminen on tehty helpoksi kosketusnäytöllä isolla napilla.

The image displays two versions of the Wreckamovie website. The top portion shows the desktop website interface for the 'Vampyre Princess 3D' production page. It features the Wreckamovie logo, a navigation menu with 'HOME', 'PRODUCTIONS', 'TASKS', 'ACTIVITY', and 'PEOPLE', and a sub-menu with 'INFO', 'TASKS', 'ACTIVITY', 'MEMBERS', and 'BLOG'. The main content area is titled 'Latest Activity' and contains three activity items:

- Closed task: Story Development**: A message from Philipp Edner stating that a clearer definition of the story's cornerstones has been reached, and a new task has been opened.
- Kalle Max Hofmann commented a shot**: A comment on a concept artwork shot by Philipp Edner, expressing excitement about the production starting in 2012.
- Philipp Edner gave a shot "Vampire Princess Concept" for Vampyre Princess 3D**: A message from Philipp Edner announcing the start of concept artwork for the production.

Below the activity feed are buttons for 'Join This Production' and 'Follow Production'. The bottom portion of the image shows a mobile app interface for 'Vampyre Princess 3D' on an iPhone 4. The app header includes the title, a 'Join Production' button (showing 88 members), and navigation tabs for 'Info', 'Activity', 'Tasks', and 'Blog'. The 'Latest Activity' section in the app mirrors the desktop version, showing the same three activity items in a condensed format.

Kuva 19. Tuotantosivun päävalikko ja tapahtumavirtaus työpöytä- ja mobiiliversiossa. (Wreckamovie 2011).

INFO TASKS ACTIVITY MEMBERS BLOG ANALYTICS

Iron Sky > [Tasks](#) > Nazi fleet radio communication

Follow Production

Share

Nazi fleet radio communication

Created at January 17, 2012

Created by [Timo Vuorensola](#)

Deadline January 20, 2012: **1 day left**

Shots given 24

Wreckupations Actor

Reference media

Thumbs ranking

Marcel Drechsler	real German option	<div style="width: 80%;"></div>	8/17%
Dirk Mengel	what I think sounds best (and is corr...	<div style="width: 60%;"></div>	6/12%
Stephan Invis Weinberger	Language...	<div style="width: 50%;"></div>	5/10%
Thorsten von Eicken	My German Radio Sounds	<div style="width: 50%;"></div>	5/10%
Martin Gehring	Another option:	<div style="width: 30%;"></div>	3/6%
Dominic Guhl	My Contribution	<div style="width: 30%;"></div>	3/6%
Tobias Langguth	Fleet Communications	<div style="width: 20%;"></div>	2/4%

Description

This one goes to fluent German-speaking Iron Sky people – listen up!

We need some radio communication for the Nazi fleet as they are approaching the Earth orbit. Your task is to record the following lines (all of them, preferably, and we will pick the best) in German, in strict but calm military style.

The lines are:

"Hier spricht das Siegfried Flotten-Flagschiff Tannhäuser. Wir haben den Orbit erreicht und sind bereit für den Start der Walküren"

"Kriegsschiff Wolfram bereit für Angriff"

"Kriegsschiff Herrmann bereit für Angriff"

"Kriegsschiff Heinrich bereit für Angriff"

Delivery in mp3 (or other commonly used audio format) either as a shot to this task, or directly to me (timo@ironsky.net)

Because this is a Nazi fleet, we only need male voices for this one.

Danke Schön!!

[Give it a Shot!](#) [Add to Todo list](#)

Shots

Order by [latest update](#) [Show All Comments](#) | [Hide Comments](#)

« Previous **1** [2](#) [3](#) Next »

David Mulder

January 18, 2012 16:24

0 Thumb-ups

Nazi Fleet Jibba Jabba [Add to Favorites](#) [Destroy](#)

My Nazifleet Radio Communication-

[Download Walküren.mp3](#)

[Download Hermann.mp3](#)

[Download Heinrich.mp3](#)

[Download Biterolf.mp3](#)

[Download Walther.mp3](#)

[Download Wolfram__2_.mp3](#)

[Jump to comment form](#)

Comments

This shot doesn't have comments.

My Contribution [Add to Favorites](#) [Destroy](#)

In Nazi Germany "klar zum Gefecht" has been used instead "bereit zum Angriff". This is why I spoke it so.

Kuva 20. Työpöytäversion task-sivu ja shotit sekä kommentit (Wreckamovie 2011).

The screenshot displays a mobile application interface with a task and a comment section.

Task: Nazi fleet radio communication

- Created at: January 17, 2012
- Created by: [User]
- Deadline: January 20, 2012: 1 day left
- Shots given: 23
- Wreckupations: Actor
- Reference media: [Image]

This one goes to fluent German-speaking Iron Sky people - listen up!

We need some radio communication for the Nazi fleet as they are approaching the Earth orbit. Your task is to record the following lines (all of them, preferably, and we will pick the best) in German, in strict but calm military style.

The lines are:

- "Hier spricht das Siegfried Flotten-Flagschiff Tannhäuser. Wir haben den Orbit erreicht und sind bereit für den Start der Walküren"
- "Kriegsschiff Wolfram bereit für Angriff"
- "Kriegsschiff Herrmann bereit für Angriff"
- "Kriegsschiff Walther bereit für Angriff"
- "Kriegsschiff Biterolf bereit für Angriff"
- "Kriegsschiff Heinrich bereit für Angriff"

Delivery in mp3 (or other commonly used audio format) either as a shot to this task, or directly to me (timo@ironsky.net)

Because this is a Nazi fleet, we only need male voices for this one.

Danke Schön!!

Give a shot!

Shots

[User] January 17, 2012 19:05

Comment

[User] January 17, 2012 17:18 | 5 Thumb-ups

My German Radio Sounds

Hey there...
I totally agree with Dirk Mengel. His correction sounds miles better so I have recorded his version and a special Tannhäuser classic edition with background sound... enjoy :)

- Download Tannhäuser_1.mp3
- Download Wolfram_1.mp3
- Download Herrmann_1.mp3
- Download Walther_1.mp3
- Download Biterolf_1.mp3
- Download Heinrich_1.mp3
- Download Tannhäuser_Sound.mp3

Add comment

Comment

Dirk Mengel
Ha - that sounds really cool !!! :)
I just had a discussion with my colleague if the ships should all give the same status report (which then must be following military terms 'Klar zum Gefecht') - this would be more correct in a military way - and more straight, or if they should give their report in a more free way like in my posting which I think sounds better... I am not sure at the moment...
...but the recording / voice itself is still my favorite...!

January 17, 2012 17:34 | 0 Thumb-ups

Kuva 21. Mobiilisivun task ja shot sekä kommentit (Wreckamovie 2011).

7.2.2 Käyttöliittymäliitännäiset

Tutkimme kahta eri valmista liitännäistä (plugin), jotka on suunniteltu mobiililaitteille optimoitujen sivustojen toteutuksen avuksi. iWebkit sisältää tyyliiedostot ja valmiit javascript-kirjastot, jotka tekevät sivustosta hyvin samankaltaisen kuin Applen laitteiden natiivisovellukset. Efekteistä mukana ovat mm. siirtymäanimaatiot sivujen välillä ja listaelementtien ja kytkinten animoinnit. Tyylejä on mukana lähes kaikille sivustolla tarvittaville elementeille; napeille, listoille, ylä- ja alapalkeille sekä erilaisille valmiille sivupohjille. Toinen kokeilemamme sovelluskehys oli jQTouch, joka on hyvin samankaltainen iWebkitin kanssa, mutta sisälsi erilaisia teemoja mobiilisovellusten käytettäväksi.

Kummallakin liitännäisellä, joita kokeilimme, oli hyvät ja huonot puolensa. Sekä iWebKit että jQTouch olivat hyvin valmiita kokonaisuuksia yksinkertaisen sivustokokonaisuuden

rakentamiseen ja sisälsivät laajan valikoiman erilaisia tyylejä ja scriptejä sekä animoituja. Molemmat olivat kuitenkin pääsääntöisesti suunniteltu vasta rakenteilla olevan sivuston mobiiliversion tekemiseen sekä pienempien sivukokonaisuuksien hallintaan. Wreckamovien ohjelmistorakenne ei myöskään sopinut kummallekaan liitännäiselle, joten päätimme ottaa vain toisesta esimerkkiä mm. nappien ja ylävalikon tyyliin.

7.3 Mobiililaitteen tunnistaminen

Kuten luvussa 4.3 totesimme, selaimen user agent mahdollistaa mobiililaitteen tai -selaimen tunnistamisen. Aina kun selain lähettää kutsun sivustolle, sen mukana kulkee user agent, jonka voi käydä läpi Ruby on Railsissa esimerkissä 6 näytetyllä tavalla.

Esim. 6.

```
def mobile_device?
  if session[:mobile_param]
    session[:mobile_param] == "1"
  else
    request.user_agent =~ /Mobile/
  end
end
```

Esim. 7.

```
def prepare_for_mobile
  session[:mobile_param] = params[:mobile] if
  params[:mobile]

  request.format = :mobile if mobile_device?
end
```

Esimerkissä 7 application controllerissa määritetty action `prepare_for_mobile` tarkistaa ennen sivun lataamista, onko kutsussa annettu parametri "mobile" määritetty todeksi. Mikäli parametriä ei ole määritetty, metodi tarkastaa käyttäjän käytössä olevan selaimen tyyppin ja versionumeron (user agent) ja asettaa käytetyksi formaatiksi (request format) mobile-nimisen MIME-tyypin, joka luodaan `environment.rb` -tiedostoon esimerkissä 8.

Esim. 8.

```
Mime::Type.register_alias "text/html", :mobile
```

Kontrollerit tulostavat selaimelle mobile-päätteisen näkymän, joka on siis mobiiliversion näkymä. Jokaisesta näkymästä, jotka halutaan mukaan mobiiliversioon, on siis tehtävä esimerkin 9 mukainen mobile-päätteinen versio.

Esim. 9.

```
show.html.erb  
show.mobile.erb
```

Jokaiselle päätelaitteelle ja selaimelle voitaisiin siis käytännössä luoda oma näkymänsä esimerkin 10 tavalla luomalla uusia MIME-tyyppejä.

Esim. 10.

```
show.iphone.erb  
show.android.erb
```

Käyttäjälle annetaan mahdollisuus siirtyä mobiili- ja täysversion välillä sivun alalaitaan sijoitetulla linkillä, joka lähettää application controllerille mobile-parametrin (?mobile=0), jota esimerkissä 7 tarkistettiin.

Esimerkissä 11 lisätään Apple-spesifisiä meta-tietoja parantamaan käyttäjäkokemusta Mobile Safarilla ja Applen laitteilla. Kotinäkymään (home screen) lisättyyn pikakuvakeeseen voidaan liittää sivuston ikoni link-tageilla. Ensimmäisessä linkissä määritetään ikoni laitteelle, jonka näytön resoluutio on 163 pistettä tuumaa kohden, kuten iPhone 3G- ja 3GS- sekä Android-puhelimet. Toisessa linkissä määritetään sama, mutta näytölle, jonka resoluutio on 326 pistettä tuumaa kohden, kuten iPhone 4 ja iPod Touch 4G.

Esim. 11.

```
<link rel="apple-touch-icon" media="screen and (resolution: 163dpi)"
      href="/images/mobile/wam_icon_57x57.png">
<link rel="apple-touch-icon" media="screen and (resolution: 326dpi)"
      href="/images/mobile/wam_icon_114x114.png">
```

Esim. 12.

```
<meta content="no" name="apple-mobile-web-app-capable">
<meta content="minimum-scale=1.0, width=device-width,
              maximum-scale=0.6667, user-scalable=no"
      name="viewport">
```

Esimerkin 12 ensimmäisellä rivillä määritetään avautuuko Applen laitteiden kotinäky-
mään laitettu linkki koko ruudulle vai avautuuko se selaimessa, kuten tässä esimerkis-
sä. Toisella rivillä määritetään Mobile Safarin näkymän skaalautuvuus ja estetään käyt-
täjän suorittama zoomaus.

8 Jatkotoimenpiteet

Wreckamovien kehitys jatkuu edelleen ja tulee jatkossa keskittymään myös mobiilipuoleen. Vaikka toimeksiannossa asetetut tavoitteet mobiilisivulle saavutettiin, on tarve jatkokehitykselle selkeä. Uusien mobiililaitteiden markkinoille tulon myötä sivustoa on jatkuvasti kehitettävä niiden ominaisuuksille sopivaksi. Palvelun mahdolliset uudet mobiiliominaisuudet voivat riippua palvelun yhteisön toiveista, jotka ovat tärkeässä asemassa sivuston kehityksessä. Valmistunut versio ei siis todellakaan ole valmis, mutta sen kehitys antoi hyvän kuvan siitä, miten kehitystä tulisi jatkaa.

MyWreck on sivuston osa, joka on mobiilikäytössä ehkä tärkeimmässä roolissa, sillä sen kautta käyttäjälle tarjotaan nopeimmin hänelle kaikkein eniten relevanttia ja ajan tasalla olevaa tietoa. Tämä osa mobiiliversiosta vaatii paljon kehitystä, sillä nyt MyWreckin kautta käyttäjälle tarjottua dataa on usein liian paljon, ja osa siitä on epärelevanttia tärkeimpiä päivityksiä selaavalle ripeälle mobiilikäyttölle.

Mobiiliversion kehitys tulisi tukea myös työpöytäversion kehitystä. Nyt kehittäjätiimillä on oivallinen tilaisuus nähdä käytännössä, miten mobiili edellä -kehittämistavasta voisi hyötyä koko palvelun suhteen. Käyttäjän kannalta oleellisimman sisällön esille tuonti ja suorituskyvyn maksimointi eivät ole pelkästään mobiilikehityksen teesejä.

Palvelua selaavan laitteen tunnistus olisi hyvä suunnitella käyttämällä sekä palvelinpuolen että asiakaspuolen ratkaisuja. User agentin tarkastus auttaa karsimaan palvelun ominaisuuksia mobiililaitteilta jo ennen kun asiakaslaitteelle lähetetään dataa. Näyttöpohjat saadaan muokattua asiakaspuolella laitteen ominaisuuksia vastaaviksi mediakyselyjä ja joustavaa sivurunkoa hyödyntäen.

Tärkeimpänä lähtökohtana jatkosuunnittelulle ja koko opinnäytetyön keskeisenä johtopäätöksenä voidaan kuitenkin pitää sitä, että palvelu tulisi suunnitella niin, että sen käytettävyys ja suorituskyky olisivat aina optimaalisia sitä selaavalle laitteelle. Sen sijaan, että keskityttäisiin yksittäisiin kohdelaitteisiin, kuten iPhoneen, tulisi ajattelumaailma kehityksessä olla lähempänä responsiivista suunnittelutapaa ja sen tulisi ottaa huomioon erilaisten laitteiden ominaisuudet. Tätä on syytä pitää tulevaisuuden suuntana web-suunnittelussa.

Lähteet

Amazon 2012. Kindle Fire - More than a Tablet. Saatavuus

<http://www.amazon.com/gp/product/B0051VVOB2/ref=famstripe_kf> (luettu 23.1.2012).

Android, Wikipedia. Saatavuus <<http://fi.wikipedia.org/wiki/Android>> (luettu 15.1.2012).

Apple 2012. Vertaile iPhone-malleja. Saatavuus

<<http://www.apple.com/fi/iphone/compare-iphones/>> (luettu 8.1.2012).

Complete.com 2010. Smartphone Owners: A Ready and Willing Audience. Saatavuus:

<<http://blog.compete.com/2010/03/12/smartphone-owners-a-ready-and-willing-audience>> (luettu 10.11.2011).

comScore 2011. Android Captures #2 Ranking Among Smartphone Platforms in EU5. Saatavuus

<http://www.comscore.com/Press_Events/Press_Releases/2011/9/Android_Captures_number_2_Ranking_Among_Smartphone_Platforms_in_EU5> (luettu 15.1.2012).

Facebook 2012. Facebook for Mobile. Saatavuus <<http://www.facebook.com/mobile/>> (luettu 9.1.2012).

Firtman, Maximiliano 2010. iPhone 4 and iOS 4 Safari detection & behavior. Saatavuus:

<<http://www.mobilexweb.com/blog/iphone4-ios4-detection-safari-viewport>> (luettu 12.12.2011).

Gartner 2011a. Gartner Says Sales of Mobile Devices Grew 5.6 Percent in Third Quarter of 2011; Smartphone Sales Increased 42 Percent. Saatavuus

<<http://www.gartner.com/it/page.jsp?id=1848514>> (luettu 8.1.2012).

Gartner 2011b. Gartner Says Worldwide PC Shipments Grew 3.2 Percent in Third Quarter of 2011. Saatavuus <<http://www.gartner.com/it/page.jsp?id=1821731>> (luettu 8.1.2012).

Geeky-Gadgets, HTC Desire S. Saatavuus <<http://www.geeky-gadgets.com/wp-content/uploads/2011/02/HTC-Desire-S1.jpg>> (luettu 11.1.2012).

Grigsby, Jason 2011. Links Don't open Apps. Cloud For Blog. Saatavuus:

<<http://www.cloudfour.com/links-do-not-open-apps/>> (luettu 10.12.2011).

Germain, Jack M. 2007. KHTML vs. Gecko vs. Trident vs. Presto: Behind the Browser. Saatavuus:

<<http://www.technewsworld.com/story/59309.html>> (luettu 17.1.2012).

GSMarena 2008. Apple iPhone 3G. Saatavuus

<http://www.gsmarena.com/apple_iphone_3g-2424.php> (luettu 15.1.2012).

GSMarena 2009. Apple iPhone 3GS. Saatavuus
<http://www.gsmarena.com/apple_iphone_3gs-2826.php> (luettu 15.1.2012).

GSMarena 2010a. Apple iPhone 4. Saatavuus
<http://www.gsmarena.com/apple_iphone_4-3275.php> (luettu 15.1.2012).

GSMarena 2010b. HTC Desire HD. Saatavuus
<http://www.gsmarena.com/htc_desire_hd-3468.php> (luettu 15.1.2012).

GSMarena 2011. HTC Desire S. Saatavuus <http://www.gsmarena.com/htc_desire_s-3776.php> (luettu 15.1.2012).

Iltalehti 2011. Suomen suurin uutispalvelu. Saatavuus <<http://m.iltalehti.fi>> (luettu 15.11.2011).

Marcotte, Ethan 2011. Responsive Web Design. New York: A Book Apart.

Microsoft 2007. Microsoft Reveals New Windows Mobile 6 Smartphone Software. Saatavuus <<http://www.microsoft.com/presspass/press/2007/feb07/02-11WM6SoftwarePR.mspx>> (luettu 12.1.2012).

Microsoft 2011. Hardware Specifications for Windows Phone. Saatavuus
<[http://msdn.microsoft.com/en-us/library/ff637514\(v=vs.92\).aspx](http://msdn.microsoft.com/en-us/library/ff637514(v=vs.92).aspx)> (luettu 8.1.2012).

Mobiiliblogi 2011. Nokia Lumia 800 yhdistää Windows Phonen kaarevaan kosketusnäyttöön. Saatavuus <<http://www.mobiiliblogi.com/2011/10/26/nokia-lumia-800-yhdistaa-windows-phonon-kaarevaan-kosketusnayttoon/>> (luettu 8.1.2012).

Mobile Browser, Wikipedia. Saatavuus <http://en.wikipedia.org/wiki/Mobile_browser> (luettu 17.1.2012).

Nielsen 2011. Android Phones and iPhones Dominating App Downloads in the US. Saatavuus <<http://blog.nielsen.com/nielsenwire/?p=30235>> (luettu 15.1.2012).

Nielsen, blog 2011. Manufacturer operating system share-smartphones. [Kuva]. Saatavuus <<http://blog.nielsen.com/nielsenwire/wp-content/uploads/2011/07/june-2011-smartphone-share.png>> (luettu 15.1.2012).

Samsung 2011. GALAXY Tab - seitsemän tuumaa vapautta. Saatavuus
<<http://www.samsung.com/fi/consumer/mobile/mobilephones/mobilephones/GT-P1000MSANEE>> (luettu 23.1.2012).

Torak 2012. Scrum. Saatavuus <<http://www.torak.com/Scrum>> (luettu 8.1.2012).

Twitter 2010. The Evolving Ecosystem. Saatavuus
<<http://blog.twitter.com/2010/09/evolving-ecosystem.html>> (luettu 15.1.2012).

Star Wreck. Distribution. Saatavuus <<http://www.starwreck.com/distribution.php>> (luettu 15.1.2012).

W3C 2010. Media Queries. Saatavuus: <<http://www.w3.org/TR/css3-mediaqueries>> (luettu 1.9.2011).

Wreckamovie 2011. Be the Future of Film. Saatavuus <<http://www.wreckamovie.com/>> (luettu 15.11.2011).

Wroblewski, Luke 2011a. Mobile First. New York: A Book Apart.

Wroblewski, Luke 2011b. RESS: Responsive Design + Server Side Components. Saatavuus: <<http://www.lukew.com/ff/entry.asp?1392>> (luettu 15.11.2011).

Zarella, Dan 2011. New Data: 33% of Facebook Posting is Mobile. Saatavuus <<http://danzarella.com/new-data-on-mobile-facebook-posting.html>> (luettu 15.1.2012).