

Eemil Pirinen

MONIALUSTAISET KÄYTTÖLIITTYMÄT: ADAPTIIVISUUS VS RESPONSIIVISUUS

Kandidaatintyö
Informaatioteknologian ja viestinnän tiedekunta
Elokuu 2023

TIIVISTELMÄ

Eemil Pirinen: Monialustaiset käyttöliittymät: adaptiivisuus vs responsiivisuus

Kandidaatintyö

Tampereen yliopisto

Tietotekniikan tutkinto-ohjelma

Elokuu 2023

Käyttöliittymä on se ohjelmiston rajapinta, jonka kanssa käyttäjä vuorovaikuttaa. Monialustaisella ohjelmistolla taas tarkoitetaan usealla eri alustalla toimivaa ohjelmistoa, joka perustuu yhteen koodipohjaan. Tällaisten ohjelmistojen käyttöliittymiä kutsutaan monialustaisiksi käyttöliittymiksi. Monialustaisten käyttöliittymien tulee kyetä sopeutumaan käyttöalustan ominaispiirteisiin, kuten näytön tai selausikkunan resoluutioon, käyttöjärjestelmään ja mahdollisesti muihin eri toiminnallisuuksiin. Kaksi yleistä suunnittelumenetelmää monialustaiselle käyttöliittymälle ovat responsiivisuus ja adaptiivisuus.

Responsiivisuudella tarkoitetaan yhden ja ainoan käyttöliittymän ominaisuutta skaalautua käyttöalustalle sopivaan muotoon. Responsiivinen käyttöliittymä tunnistaa päätelaitteen ominaisuudet sekä kykenee muokkaamaan näkymää sopiakseen kyseiselle alustalle. Tavoitteena on saavuttaa samat ominaisuudet sekä käyttökokemus jokaisessa eri kontekstissa ja alustalla. Adaptiiviselle käyttöliittymälle on sen sijaan luotu useita eri versioita, joista käyttökontekstista riippuen valitaan sopivin. Adaptiivisessa käyttöliittymässä paras näkymä valitaan päätelaitteen resoluutioon perustuen. Tällöin hajoitetaan ohjelmiston yhtenevyyttä eri alustojen välillä, mutta tavoitellaan parasta visuaalista esitystä alustasta riippumatta.

Tämän tutkielman tarkoituksena on analysoida adaptiivisen ja responsiivisen käyttöliittymäsuunnittelun vahvuuksia sekä heikkouksia. Tutkielma on toteutettu kirjallisuuskatsauksena ja aineistona on käytetty suunnittelumenetelmiä käsitteleviä tieteellisiä lähteitä, joita pääsääntöisesti ovat aiheutta tutkivat kirjat sekä tieteelliset artikkelit. Tutkimuskysymykseen pyritään vastaamaan toteuttamalla suunnittelumenetelmien välinen vertailu, joka perustuu aineistosta kerättyyn materiaaliin. Vertailun kategorioina ovat kaikki kirjallisuuskatsauksessa esille nousseet monialustaisen käyttöliittymäsuunnittelun ja käyttöliittymien ominaisuudet. Jokaisen kategorian kohdalla parempi suunnittelumenetelmä valitaan adaptiivisuudesta ja responsiivisuudesta löydettyjen vahvuuksien sekä heikkouksien perusteella. Työn tavoitteena on selvittää millaisissa tilanteissa kumpaakin toteutusmallia on perustellusti järkevää käyttää.

Tutkielmassa toteutetun vertailun perusteella suunnittelumenetelmillä on niiden toiminnallisuksien ja toteutustapojen myötä yksilölliset ja hyvin kehittäjän tavoitteisiin sidonnaiset vahvuudet. Adaptiivisuus sopii käyttöliittymän suunnittelumenetelmäksi projekteissa, joissa tavoitellaan mahdollisimman tehokasta, optimaalista sekä käyttäjäystävällistä lopputulosta päätelaitteesta huolimatta. Responsiivisen käyttöliittymäsuunnittelun todettiin olevan paras ratkaisu tavoiteltaessa käyttöliittymän optimointia ensisijaisesti yhdelle laitetypille tai etsittäessä toteutuksen ja ylläpidon kannalta edullisempaa vaihtoehtoa. Tutkielman tulos ei viittaa kummankaan suunnittelumenetelmän olevan yksiselitteisesti toistaan parempi, vaan oikean menetelmän valinta perustuu valtaosin kehittäjän tavoitteisiin ja kehitettävän ohjelmiston käyttökontekstiin.

Avainsanat: monialustaisuus, responsiivinen, adaptiivinen, käyttöliittymä, käyttöliittymäsuunnittelu

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

ABSTRACT

Eemil Pirinen: Multiplatform user interfaces: adaptive vs responsive
Bachelor's Thesis
Tampere University
Bachelor's Degree Programme in Information Technology
August 2023

The part of a software which a user can interact with is known as a user interface. A software that has a single codebase but can be run on multiple different platforms is called a multiplatform software. The user interface for such program is called a multiplatform user interface. A multiplatform UI needs to be capable of adapting based on the characteristics of the used platform, such as screen resolution, operating system and any other unique functionalities. Two common methods for creating multiplatform user interfaces are responsive and adaptive design.

Responsive design means that a single UI is able to re-scale itself based on the use context. A responsive UI recognizes the features of the end device and is then able to adapt accordingly. This drives to achieve the most similar user experience and functionalities on every platform. Adaptive user interface on the other hand has multiple variations of the same interface from which the most suitable one is selected for every platform. The most optimal layout is selected according to the end device's resolution. This breaks up the coherence of the interface across different platforms, yet aims to achieve the best possible visual representation of the software regardless of the use context.

The purpose of this study is to analyze the strengths and weaknesses of both responsive and adaptive design methods. This study is carried out as a literature review and the selected materials are mainly books and scientific articles about the topic. In order to answer the research question, a comparison between the design methods is executed which is based on the previously mentioned materials. The properties of both multiplatform design methods and user interfaces are chosen as the categories for the comparison. The better design method in context of each category is decided based on the strengths and weaknesses found in both responsive and adaptive design. The study aims to discover in what situations each of the methods are reasoned to be used.

When comparing the two methods it became clear that adaptive design is most effective when pursuing a performance efficient, most optimal and user friendly result for every desired device. Where as responsive design is best suited for situations where the interface is aimed mainly for one platform and lowering the costs of development and maintenance is high on the priority list. The results of this study does not indicate that one design method is necessarily better than the other, but rather the correct method of choice is largely dependant on what is the developer aiming for and the use context of the developed software.

Keywords: multiplatform, responsive, adaptive, interface, design

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

SISÄLLYSLUETTELO

1. Johdanto	1
2. Tutkimusmenetelmä	3
3. Monialustaisuus	4
3.1 Monialustaisuuteen työntävät voimat	4
3.2 Monialustaisuus ja käyttöliittymät	5
4. Responsiivinen käyttöliittymäsuunnittelu	6
4.1 Responsiivisuuden vahvuudet	7
4.2 Responsiivisuuden heikkoudet	8
5. Adaptiivinen käyttöliittymäsuunnittelu	10
5.1 Adaptiivisuuden vahvuudet	11
5.2 Adaptiivisuuden heikkoudet	12
6. Adaptiivisen ja responsiivisen suunnittelun vertailu	13
7. Johtopäätökset	16
Lähteet	18

LYHENTEET JA MERKINNÄT

CSS	Verkkosivuille kehitetty tyylisivu (engl. Cascading Style Sheet)
HTML	Hypertekstiä sisältävä avoimesti standardoitu merkintäkieli (engl. HyperText Markup Language)
IoT	Esineiden internet (engl. Internet of Things)
RWD	Responsiivinen käyttöliittymäsuunnittelu (engl. Responsive Web Design)
SDK	Ohjelmistokehityspaketti (engl. Software Development Kit)
SEO	Hakukoneoptimointi (engl. Search Engine Optimization)
URL	verkkosivun osoite (engl. Uniform Resource Locator)

1. JOHDANTO

Markkinoille ilmaanuu jatkuvasti uusia, eri käyttökontekstiin ja erilaisilla ominaisuuksilla varustettuja laitteita. Tämä esittää haasteita ohjelmoijille, sillä ohjelmistojen halutaan kykenevän toimimaan laajalla valikoimalla erilaisia laitteita, eli eri alustoilla. (Mori et al. 2004) Ohjelmistoja ja erityisesti mobiililaitteille alunperin suunaattuja applikaatioita on tarve käyttää niin mobiililaitteilla ja tietokoneilla, kuin esimerkiksi sulautetuissa järjestelmissä ja muissa alkeellisissa tietokoneissa. Muun muassa musiikinsoittamiseen tarkoitettujen sovellusten integrointi auton ajotietokoneeseen toimii esimerkkinä mobiililaitteelle tutun ohjelmiston istuttamisesta uuteen ympäristöön.

Ohjelmistotuotannossa saman ohjelmiston kehittäminen ja ylläpitäminen erikseen monilla alustoille vaatii paljon ylimääräistä työtä ja resursseja (Shah et al. 2019). Tämän ongelman ratkaisemiseksi esille on noussut menetelmiä yhden ohjelman hyödyntämiseksi useilla alustoilla, jota yleisesti kutsutaan monialustalliseksi kehitykseksi (Vranić ja Staráček 2014). Periaatteen takana on WORA (Write Once, Run Anywhere), eli suomennettuna kirjoita kerran, aja missä vain. WORA yksinkertaisuudessaan tarkoittaa applikaation ohjelmoimista vain kerran sellaisella teknologialla, jolla se kyetään julkaisemaan useille eri alustoille. (Shah et al. 2019) Oleellinen osa monialustaisuutta on myös ohjelmiston käyttöliittymä, sillä se on käyttäjälle visuaalinen esitys ohjelmistosta, oli käytettävänä sitten mobiililaitte tai esimerkiksi äly-TV. Monialustainen käyttöliittymä sopeutuu käytössä olevan laitteen ominaisuuksiin, kuten resoluutioon. Menetelmiä tämän saavuttamiseksi on useita, ja erityisesti web-selainten käyttöliittymäsuunnittelussa yleisiä lähestymistapoja ovat responsiivisuus ja adaptiivisuus.

Tässä työssä tarkastellaan responsiivisen ja adaptiivisen käyttöliittymäsuunnittelun vahvuuksia ja heikkouksia sekä vertaillaan niitä keskenään. Tavoitteena on saavuttaa ymmärrys menetelmien eduista toisiinsa nähden ja täten mahdollistaa oikean menetelmän valinta suunniteltavasta ohjelmistosta sekä sen käyttökontekstista riippuen.

Tutkielmassa käsiteltävä aihe on ajankohtainen, sillä internetiä, verkkoselaimia ja applikaatioita hyödyntävien laitteiden monimuotoisuus kasvaa jatkuvasti. Tämän lisäksi jo pelkästään tietokoneen ja mobiililaitteen välisten käyttöliittymämuunnosten ongelmiin törmää lähes jokapäiväisessä arjessa, kun tietokoneelta tutun verkkosivun näkymä ei suostukkaan kääntymään käytettävään muotoon mobiililaitteella. Tilanteeseen sopivaa monialustallista

käyttöliittymäsuunnittelun menetelmää noudattamalla voidaan saavuttaa haluttu käyttäjäkokemus kohdealustoilla kustantamatta kohtuutonta määrää työtunteja ja resursseja. Tämän takia aihe on mielenkiintoinen myös erityisesti ohjelmistosuunnittelun näkökulmasta.

Tutkielman luvussa kaksi esitellään tutkielman toteutustapaa ja avataan muun muassa lähteiden haun prosessia. Luvussa kolme käsitellään monialustaisuutta, sen etuja sekä sitä, miksi yhä useampia alustoja tukevien ohjelmistojen kehittäminen on ajankohtaista. Luvussa esitellään myös käyttöliittymät ja erityisesti niiden merkitys monialustaisten ohjelmistojen toteutuksessa. Luvussa neljä esitetään responsiivinen ja luvussa viisi adaptiivinen käyttöliittymäsuunnittelu. Näissä luvuissa keskitytään menetelmien ominaisuuksiin, vahvuuksiin, heikkouksiin, sekä yleisiin käyttökohteisiin. Luvussa kuusi adaptiivisuudesta ja responsiivisuudesta löydetyt vahvuudet ja heikkoudet asetetaan vastakkain. Vertailulla kartoitetaan suunnittelumenetelmien etuja toisiinsa nähden. Viimeisenä on koko tutkielman yhteenveto, vertailun pohjalta tehtävät johtopäätökset sekä tutkimuskysymykseen vastaaminen johtopäätösten avulla toteutuu luvussa seitsemän.

2. TUTKIMUSMENETELMÄ

Tämä tutkielma on toteutettu kirjallisuuskatsauksena. Tutkimuskysymykseen on pyritty vastaamaan etsimällä ja vertailemalla tieteellisestä kirjallisuudesta löytyvää tietoa aiheesta. Lähteitä on haettu Google Scholar, ACM Digital Library, IEEE Electronic Library, sekä Tampereen yliopiston kirjaston Andor tietokannoista. Hakulausekkeissa käytettiin useimmiten seuraavia avainsanoja

"multiplatform", "cross-platform", "responsive", "adaptive", "web", "design", "UI".

Hakulausekkeissa erityisesti avainsanat **web** ja **design** tuottivat haluttuja hakutuloksia, sillä avainsanat kuten **adaptive**, **responsive** ja **multiplatform** sellaisenaan ilmenivät viittaavan myös moniin muiden tieteenalojen käsitteisiin.

Tutkimuskysymystä käsittelevää lähdemateriaalia oli kohtuullisen hankalaa löytää, sillä esimerkiksi laadukkaita informaatio adaptiivisuudesta ja responsiivisuudesta oli monessa tapauksessa sisällytettyinä laajemmin käyttöliittymäsuunnittelua käsitteleviin kirjoihin. Erityisen onnistuneesti lähdemateriaalia löytyi jo ennalta tutkielmassa käytettyjen teosten lähteistä. Responsiivisuuden osalta valikoituneiden lähteiden luotettavuutta tarkasteltiin vertaamalla vastaavista teoksista löytyvään informaation julkaisuvuosi huomioiden. Sen sijaan lähes kaikki adaptiivisuuden lähdemateriaaliksi valitut teokset viittasivat Aaron Gustafsonnin (2015) kirjaan *Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement*, jota on myös tässä työssä käytetty lähteenä. Vertailun kannalta responsiivisuudesta sekä sen vahvuuksista ja heikkouksista löytyi paljon laadukkaita ja luotettavia lähteitä, kun taas adaptiivisuudesta vastaavaa tietoa oli huomattavasti vaikeampi löytää. Tutkielmassa ominaisuuksien vertailua toteutettiin vain, jos kyseistä ominaisuutta puoltavia tai vastustavia argumentteja löytyi kummankin suunnittelumenetelmän kohdalta.

Tutkimuksessa käytettyä lähdemateriaalia pyrittiin rajaamaan vuonna 2015 julkaistuihin sekä uudempiin teoksiin. Myös vanhempaa materiaalia on käytetty lähteinä tilanteissa, joissa käytettävän informaation ajankohtaisuus kyettiin varmentamaan tuoreemmista teoksista. Aineistoa tutkiessa vanhemmista ja uudemmista lähteistä löytynyt tieto oli hyvin yhtenevää, eikä tutkimuskysymyksen kannalta törmätty ristiriitaiseen informaatioon. Lopulliseen lähdemateriaaliin valikoitiin laajasti samoja aiheita käsitteleviä teoksia, jotta tutkimuskysymykseen vastaamiseksi vertailuun saatiin riittävän laaja näkökulma.

3. MONIALUSTAISUUS

Perinteisesti tietylle alustalle kohdennettujen natiivien ohjelmistojen kehitys perustuu kyseisen teknologian edellyttämiin työkaluihin. Käyttöjärjestelmien tuottajat tarjoavat omalle alustalleen suunnitellun ohjelmistokehityspaketin tai lyhyesti SDK:n, jonka työkalujen avulla ohjelmistokehittäjät voivat kehittää ohjelmistoja haluamallaan teknologialla. (Nagy 2022) Esimerkiksi yleisin ohjelmointikieli Android-applikaatioiden kehittämiseen on Java, mutta päästäkseen käsiksi kaikkiin vaadittaviin työkaluihin on ohjelmoijan otettava osaksi kehitysympäristöään myös Androidin tarjoama ohjelmistokehityspaketti. Android SDK mahdollistaa muun muassa Javalle ominaisten .jar-tiedostojen kokoamisen Androidille yhteensopivaksi ohjelmistoksi. (Griffiths 2015 - 2015) Monialustaisessa ohjelmistokehityksessä sen sijaan käytetään erillistä SDK:ta, jonka avulla kytetään referoimaan jokaiselle kohdelaustalle ominaisen SDK:n työkaluja. Täten voidaan kirjoittaa ohjelmisto, joka on mahdollista koota usealle eri käyttöalustalle. (Nagy 2022)

3.1 Monialustaisuuteen työntävät voimat

Monialustaisten ohjelmistojen kehitys on viimevuosina yleistynyt merkittävästi. Esimerkiksi Google Play Storen kaiken kaikkiaan noin viidestä miljoonasta applikaatiosta yli 200 000 on kehitetty monialustaisten ohjelmistojen kehitykseen tarkoitetulla Flutterilla. Suurimmat syyt monialustaisuuden yleistymiselle ovat ohjelmoinnin yksinkertaistaminen ja kehittäjän kulujen vähentäminen. (Nagy 2022)

Monialustaisuus tarjoaa ratkaisua ohjelmoinnin yksinkertaistamiseksi, sillä muutokset natiiveissa ympäristöissä kehitetyissä ohjelmistoissa vaativat muutosten tekemistä jokaisessa kehitysympäristössä (García-Moya Herrera 2013). Täten kehittäjän tulisi osata kehittää samalla laadulla yhtä ohjelmistoa usealla eri teknologialla. Yhden, puhumattakaan useamman teknologian hallitseminen vaatii ohjelmoijalta toistuvaa osaamisen päivittämistä kyseisten teknologioiden jatkuvan kehityksen myötä (Nagy 2022). Monialustaisuutta tukevalla teknologialla samat muutokset voidaan tehdä tutussa kehitysympäristössä ja täten vähentää yksittäisen ohjelmoijan taakkaa sekä mahdollisesti tiettyihin kehitysvaiheisiin vaadittavaa työvoimaa. Toisaalta on myös oleellista mainita, että monimutkaisuudesta huolimatta osa ohjelmoijista saattavat jopa suosia monialustaisen ohjelmiston kehittämistä natiiveissa ympäristöissä niiden tarjoaman nopeuden ja resurssien takia (García-Moya

Herrera 2013).

Kustannusten kannalta monilla eri teknologioilla kehitettävässä ohjelmistossa yksittäisten ohjelmojen mieltymykset sekä natiivien kehitysympäristöjen ominaispiirteet saattavat aiheuttaa eroavaisuuksia kehitysvaiheessa. Tätä tarvetta eri versioiden yhtenevyyden ylläpidolle kyetään vähentämään monialustaisen kehityksen jaetun koodin avulla, joka puolestaan laskee kehityksen kokonaiskuluja. (Nagy 2022) Ohjelmoinnin yksinkertaistaminen siis jo itsessään puhuu myös ohjelmiston tuotannon kulujen vähenemisestä.

Monialustaisuuteen työntävien käytännöllisten ja taloudellisten voimien ymmärtäminen auttaa hahmottamaan, miksi kyseisten teknologioiden yleisyys on jatkuvassa kasvussa. Tämä yleistymisen puolestaan kertoo myös tarpeesta ymmärtää käyttöliittymiä osana monialustaisia teknologioita sekä sitä, miten käyttöliittymät saadaan mukautettua onnistuneesti useille eri alustoille.

3.2 Monialustaisuus ja käyttöliittymät

Vuonna 2022 toteutetun kyselyn mukaan yksi yleisimmistä määritelmistä käyttöliittymälle on tila, jossa tapahtuu vuorovaikutus ihmisen ja koneen välillä. Toisen määritelmän mukaan käyttöliittymä on käyttäjän ja tietokoneohjelman risteys, kun taas kolmas ehdottaa käyttöliittymäksi komentojen ja valikoiden yhdistelmää, joiden avulla käyttäjä kommunikoi ohjelmiston kanssa. (Morales Díaz 2022) Kiistelystä määritelmästä huolimatta käyttöliittymän voidaan väittää olevan kriittisin osa ohjelmistoa, sillä huono käyttöliittymä laskee käyttäjän kykyä hyödyntää ohjelmiston tarjoamia ominaisuuksia. Jopa muilta osin erinomaisesti suunniteltu ja toteutettu ohjelmisto voi täysin epäonnistua huonosti toteutetun käyttöliittymän takia. (Sridevi 2014)

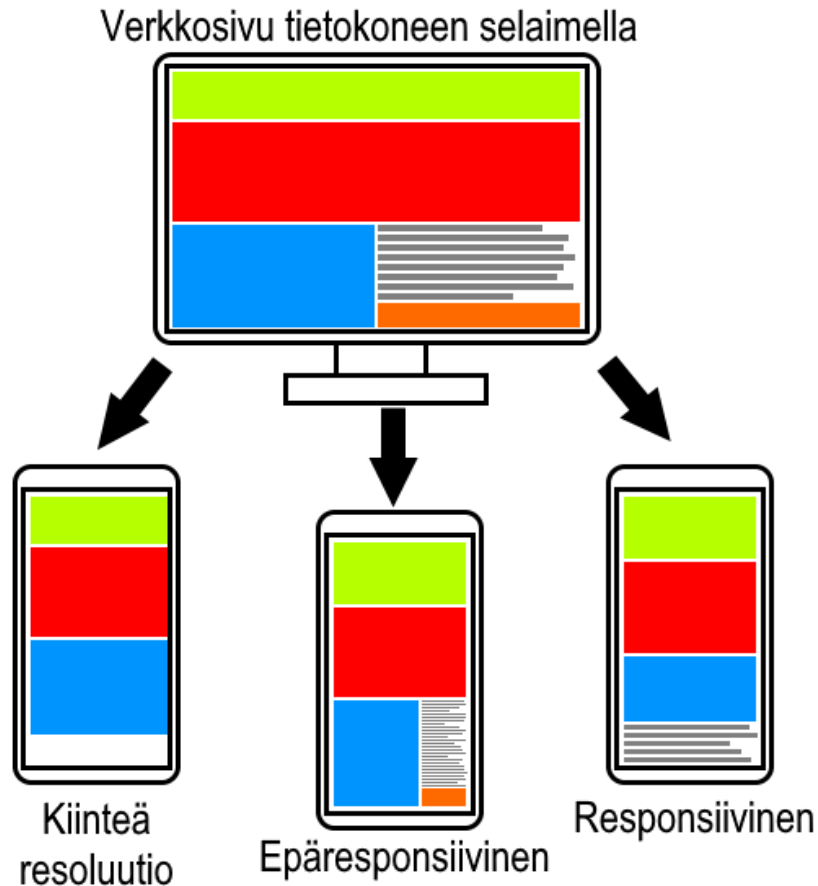
Kuten aiemmin on todettu, monialustainen ohjelmisto on mahdollista julkaista usealla eri alustalla ilman sen ohjelmoimista monella eri teknologialla (Shah et al. 2019; Bishop ja Horspool 2006). Käyttöliittymän kohdalla tämä sopeuttaminen on kuitenkin vielä huomattavasti hankalampaa, sillä suunnittelussa tulee ottaa huomioon kaikki ihmisen ja ohjelmiston väliseen vuorovaikutukseen vaikuttavat tekijät. Yksinkertaisimmillaan tämä saattaa tarkoittaa vain käytettävän laitteen näytön resoluutiota, mutta esimerkiksi IoT-ratkaisuissa yleiset epästandardiset symboliset näytöt saattavat tehdä käyttöliittymän sopeutuksesta monimutkaista. (Chmielewski 2015) Tavoitteena on tarjota kaikki ohjelmiston sisältö ja ominaisuudet jokaiselle alustalle sekä niiden esitys käytännöllisessä ja visuaalisesti miellyttävässä muodossa (Dominguez et al. 2021). Käyttöliittymä edustaa käyttäjän ikkunaa ohjelmistoon ja antaa kuvan sen laadusta. Käyttäjä saattaa siis totaalisesti hylätä valitsemansa sovelluksen tai verkkosivun, mikäli käyttöliittymän sopeuttaminen käytössä olevalle alustalle epäonnistuu ja johtaa sekavaan tai jopa täysin rikkiin näkymään. (Sridevi 2014)

4. RESPONSIIVINEN KÄYTTÖLIITTYMÄSUUNNITTELU

Responsiivisen käyttöliittymäsuunnittelun ajatuksena on huomioida käytettävän laitteen ja näytön ominaisuudet. Responsiivisuus mahdollistaa käyttöliittymän sovittamisen käyttökontekstiin sisältöä venyttämällä sekä näkymää muuttamalla. (Clark 2015) Responsiivinen käyttöliittymäsuunnittelu eli lyhyesti ilmaistuna RWD edustaa monialustaista käyttöliittymäsuunnittelua, sillä jokaiselle käyttöalustalle sovellettava käyttöliittymä luodaan yhdestä, esimerkiksi HTML, CSS ja JavaScript teknologioilla toteutetusta koodikannasta (Ward 2017). RWD:n peruseriaatteet ovat ruudukkoihin perustuvan näkymän säädeltävyys, joustavat objektit sekä mediakyselyt. CSS:n mediakyselyjen avulla kyetään selvittämään käyttöalustan ominaispiirteet, kuten esimerkiksi näytön ja sisällön keskeytyskohdat. Tämän jälkeen näkymä ja sen sisältämät objektit on mahdollista sovittaa selvitettyjen vaatimusten mukaan. (Clark 2015)

Ennen responsiivisuuden yleistymistä verkkosivut suunniteltiin kiinteällä resoluutiolla, jonka seurauksena esimerkiksi suunniteltua suuremmalla näytöllä verkkosivun reunoille jää tyhjiä kaistaleita. Kuitenkin entistä huonompi lopputulos saadaan ennakoitua pienemmällä käyttölaitteella, jolloin osa sisällöstä jää ruudun ulkopuolelle, eli täysin käyttäjän ulottumattomiin. (Clark 2015) Nykyään täysin skaalautumattomia käyttöliittymiä näkee hyvin harvoin jos lainkaan, sillä modernit verkkosivut ovat jo luonnostaan jokseenkin joustavia. Verkkosivujen sisältö siis asettuu automaattisesti siten, että se on tavalla tai toisella nähtävissä käytettävän näytön koosta ja selaimesta huolimatta. Tämä voi kuitenkin johtaa käytettävyysoongelmiin, sillä esimerkiksi mobiililaitetta käytettäessä tekstisisältö saattaa skaalautua niin pieneksi, että sitä on käytännössä mahdotonta lukea (Frain 2020). Responsiivisuus tarjoaa ratkaisua nimenomaan tähän ongelmaan hyödyntämällä muokattavaa ruudukkonäkymää yhdessä joustavien objektien kanssa (Clark 2015). Mediakyselyt kykenevät muokkaamaan riveihin ja sarakkeisiin perustuvaa ruudukkonäkymää siten, että rivien ja sarakkeiden kokoonpano muuttuu riippuen käytettävän päätelaitteen näytön koosta. Ruudukkonäkymä jakautuu osioihin, jotka ovat prosentuaalisessa suhteessa ikkunan kokoon ja täten muuttaa kokoaan oikeaoppisesti esimerkiksi tietokoneella selaimen ikkunakokoa muutettaessa. Ollakseen aidosti responsiivinen verkkosivun tulee skaalautuessa ottaa huomioon myös näkymän sisältö, kuten tekstielementit ja kuvat. Erityisesti tekstin kohdalla responsiivinen verkkosivu ottaa huomioon tekstin piirteitä, kuten esimerkiksi fonttikoon, rivivälit ja tasauksen säilyttääkseen käytettävyyden skaalautuessaan uu-

den näkymän dimensioihin. (Ward 2017) Kuvassa 4.1 on esitetty tietokoneen verkkoselaimelle asettuvan verkkosivun muunnos mobiililaitteen ruudulle tilanteissa, joissa verkkosivu on suunniteltu ensiksi kiinteällä resoluutiolla, sitten epäresponsiiviseksi ja lopuksi responsiiviseksi.



Kuva 4.1. Esimerkki verkkosivun skaalautumisesta mobiililaitteelle.

Kuvasta voidaan nähdä miten kiinteän resoluution kohdalla osa sisällöstä katoaa täysin mobiililaitteelle luodun näkymän ulkopuolelle. Nykyaikaisen luonnostaan joustavan epäresponsiivisen verkkosivun (Frain 2020) kohdalla kaikki elementit skaalautuvat uuden päätelaitteen ruudulle, mutta esimerkiksi tekstin luettavuus laskee huomattavasti pienentyneen näytön takia. Responsiivisen verkkosivun kohdalla kaikki elementit skaalautuvat uuden näkymän dimensioihin ja myös tekstielementin käytettävyys säilyy. Tässä esityksessä responsiivisessä näkymässä kaikki elementit eivät ole välittömästi nähtävillä, mutta niihin pääsee käsiksi vierittämällä näkymää alaspäin.

4.1 Responsiivisuuden vahvuudet

Monialustaisen kehityksen periaatteiden mukaisesti (Nagy 2022) yksi responsiivisen suunnittelun suurimmista hyödyistä on ohjelmistotuotannon taloudelliset edut (Clark 2015). RWD:n avulla yksi versio verkkosivusta saadaan adaptoitumaan käyttökontekstiin ja tä-

ten resursseja ei tarvitse käyttää laitekohtaisten versioiden kehittämiseen. Lisäksi responsiivisuus tukee verkkosivun hakukoneoptimointia (SEO), sillä sivu toimii samalla URL-osoitteella päätelaitteesta huolimatta. Responsiivinen verkkosivu ei siis vaadi laitekohtaista uudelleenohjausta, ja tämä nostattaa todennäköisyyttä verkkosivun löytymiselle hakurobottien toimesta. SEO:ta parantaa myös hakukoneiden sisäiset mittarit, jotka suosivat erityisesti responsiivisuuteen perustuvia monialustaisia verkkosivuja. (Clark 2015)

Toinen responsiivisuuden etu on sen käyttöliittymälle tuoma johdonmukaisuus. Kuten aiemmin on todettu, responsiivisen käyttöliittymän adaptoituminen perustuu säädeltävän ruudukkonäkymän mahdollistamaan objektien venyttämiseen sekä uudelleensijoittamiseen. Käyttöliittymän tarjoama sisältö pysyy siis pääpiirteittäin samana ja täten käyttäjät voivat odottaa hyvin samanlaista käyttäjäkokemusta päätelaitteesta huolimatta. Responsiivinen suunnittelu myös itsessään ajaa kohti yksinkertaisempaa ja puhtaampaa verkkosivua sen asettamien rajoitusten seurauksena. Sivun optimointi mobiililaitteille ja muille pienille näytöille vaatii erityistä keskittymistä pääominaisuuksiin ja täten mahdollisesti vähentää ylimääräistä koodia sekä parantaa käyttäjäkokemusta. (Clark 2015)

4.2 Responsiivisuuden heikkoudet

Responsiivisella suunnittelulla on vahvuuksiensa ohella myös heikkouksia erityisesti verrattaessa muihin monialustaisen verkkosuunnittelun lähestymistapoihin. Yksi näistä on responsiivisuuden implementointi jo olemassa oleviin verkkosivuihin. Mediakyselyjen pohjalta säädeltävä ruudukkorakenne vaatii hyvin spesifiä arkkitehtuuria koko ohjelmistolta, joten sen sisällyttäminen olemassaolevaan verkkosivustoon edellyttää merkittävää uudelleensuunnittelua. (Nebeling ja Norrie 2013)

Toinen responsiivisuuden haasteista on suorituskykyyn ylläpitäminen ja syynä tähän on ensisijaisesti suuret tiedostokoot. Verkkosivun oikeoppinen skaalautuminen vaatii suuria ja laajenevia CSS tiedostoja, kun taas laajoja HTML tiedostoja saatetaan käyttää vain tiettyissä näkymissä. Seurauksena on pullonkaulamaisesti rajoittunut suorituskyky ja esimerkiksi hidas latausnopeus. Ratkaisuja tämän ongelman lievittämiseksi on kehitetty, mutta suuret tiedostokoot toimivat silti merkittävänä rajoitteena responsiivisuudelle. (Clark 2015)

Lisää haasteita responsiivisuudelle tuottaa mainonnan, erityisesti banneri- ja videomainosten toteuttaminen osana responsiivisista käyttöliittymästä. Selainten standardisoidut mainoskoot vaativat niiden erillistä implementointia osaksi joustavasti skaalautuvaa näkyä. Mainostajat ja selainohjelmat pyrkivät tukemaan responsiivisuutta, kuten esimerkiksi Googlen AdSense -ohjelmisto kontrolloi mainosten esitystapaa reagoimalla käytettävän näytön kokoon. Tämä ei kuitenkaan tarkoita sitä, etteikö mainonnan toteuttaminen responsiivisella käyttöliittymällä vaatisi erityishuomiota toteutusvaiheessa. (Mohd Shukran et al. 2014)

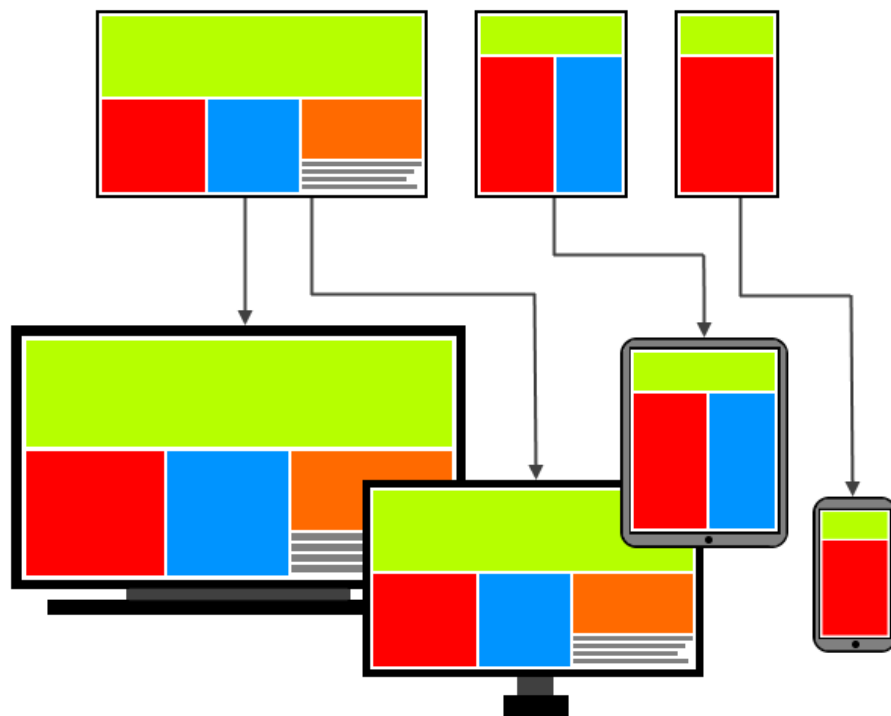
Maininnan arvoista on myös responsiivisuuden pakollinen ehto aloittaa suunnittelu jonkin tietyn päätelaitteen näkymästä, josta se kyetään muuntamaan muille käyttöalustoille. Seurauksena erityisesti mobiilinäkymien optimointi kärsii, mikäli suunnittelu on aloitettu esimerkiksi suuremmalle näyölle optimoidusta verkkoselaimen näkymästä. Parhaimmaksi ratkaisuksi tähän on todettu olevan niinsanottu Mobile-First -menetelmä, jossa suunnittelu aloitetaan nimenomaan mobiilille olennaisimmista ominaisuuksista ja niiden esityksestä käyttöliittymässä. (Margea et al. 2017) Mobile-First -mallista huolimatta responsiivinen verkkosivu on hyvin hankalaa tai kutakuinkin mahdotonta optimoida täydellisesti jokaiselle päätelaitteelle, koska pohjimmillaan perustana on sätää jollekin alustalle optimoitua käyttöliittymää. Responsiivisella käyttöliittymällä on siis hyvin hankalaa tyydyttää koko asiakaskuntaa, kun jokaisella käyttöalustalla on omat tarpeet ja jokaisen alustan käyttäjillä omat alustakohtaiset odotukset käyttöliittymältä. (Ward 2017)

5. ADAPTIIVINEN KÄYTTÖLIITTYMÄSUUNNITTELU

Alunperin Aaron Gustafsonin vuonna 2011 esittelemä adaptiivisen käyttöliittymäsuunnittelun malli (Hing 2018) perustuu näkymän ja sen käyttäytymisen lisäksi myös koko käyttöliittymän muuttamiseen (Edge 2022). Toisin kuin responsiivisuudessa, adaptiivinen käyttöliittymäsuunnittelu keskittyy ensisijaisesti alustakohtaiseen käyttöliittymän optimointiin luomalla sopivan näkymän jokaiselle tyypilliselle päätelaitteelle. Adaptiivinen ohjelmisto tunnistaa käytettävän laitteen ja ohjaa käyttäjän ikäänkuin juuri kyseiselle laitteelle suunniteltuun verkkosivustoon. (Hing 2018) Päätelaitteesta tunnistetaan näytön koko, jolloin kyseiseen resoluutioon valitaan sopivin vaihtoehto olemassaolevista käyttöliittymän näkymistä (Heath 2017). Näkymän laitekohtainen valinta perustuu progressive enhancement menetelmään, jossa JavaScriptillä ohjelmoitu scripti toteuttaa ennaltamääritellyt muutokset verkkosivun HTML marginaaliin, ja lopputuloksena käyttäjä kohtaa juuri päätelaitteelle optimoidun käyttöliittymän (Gustafson 2015).

Adaptiivisen käyttöliittymän vaihtoehtoiset näkymät ovat valikoima erilaisia kiinteitä aseteluita. Usein tämä valikoima koostuu yleisimmistä laitteiden ruutujen leveyksistä, jotka ovat 320, 480, 760, 960, 1200 ja 1600 pikseliä. Tällöin valikoimasta löytyy sopiva vaihtoehto täsmäämään useimpien päätelaitteiden resoluutioon. Adaptiivisuus ei kuitenkaan rajoitu vain kohdennettuihin resoluutioihin, vaan tarjoaa käyttäjälle pääsyn sisältöön käytettävästä laitteesta tai selaimesta huolimatta. Adaptiivinen käyttöliittymä nimittäin tarjoaa käyttäjälle aina parhaimman olemassaolevan vaihtoehdon, vaikka päätelaitteen resoluutiota ei täydellisesti vastaisi mikään olemassaolevista näkymistä. (Gustafson 2015) Täten adaptiivisuus köyhimmilläänkin edustaa monialustaisen käyttöliittymän periaatetta tarjota sama ohjelmisto laitteelle kuin laitteelle yhdestä koodikannasta (Shah et al. 2019). Olennaista adaptiivisuudessa on se, että tunnistettava näytön koko viittaa nimenomaan fyysisesti käytettävään laitteeseen, eikä esimerkiksi verkkoselaimen senhetkiseen ikkunakokoon. Tästä syystä adaptiivinen käyttöliittymä ei muuta näkymäänsä selaimen ikkunakokoa muutettaessa. (Hing 2018)

Kuvassa 5.1 on esitetty miten adaptiivisen verkkosivuston käyttöliittymän kolme, eri resoluutioille kohdennettua näkymää valikoituu kullekin päätelaitteelle. Käyttöalustoina kuvassa ovat äly-TV, tietokone, tabletti, sekä mobiililaitte.



Kuva 5.1. Verkkosivun eri näkymien valikoituminen päätelaitteille.

Kuvasta voidaan nähdä miten mobiililaitteelle ja tabletille valikoituu juurikin kyseisten näyttökokojen omaaville laitteille suunniteltu verkkosivun näkymä. Tietokoneelle ja äly-TV:lle sen sijaan valikoituu sama näkymä, koska niiden näyttöjen leveyksien ja korkeuksien suhde on sama.

5.1 Adaptiivisuuden vahvuudet

Monialustaisuuden ydinajatuksena on saman verkkosivuston tai applikaation tarjoaminen useille eri käyttöalustoille (Mori et al. 2004). Tästä syystä adaptiivisuuden tarjoama valikoima näkymiä eri alustoille on sen suurin vahvuus. Resoluutioille kohdennetut näkymät mahdollistavat optimoidun käyttäjäkokemuksen laitteesta huolimatta, jolloin esimerkiksi mobiilikäyttäjät eivät kärsi liiallisesta sisällöstä pienellä ruudulla. (Yousaf et al. 2018)

Toinen adaptiivisuuden vahvuuksista on suhteellisen nopeat latausajat. Eri päätelaitteille optimoidut käyttöliittymät eivät vaadi kaikkien elementtien ja resurssien lataamista jokaisella alustalla. Sen sijaan ladataan vain kyseisellä päätelaitteella käytettävän näkymän tarvitsemat HTML-tiedostot. (Hing 2018)

Kuten aiemmin on todettu, myös verkkosivuilla esiintyvät mainokset saattavat tuottaa ongelmia, koska ne tuodaan osaksi käyttöliittymää lähtökohtaisesti ennaltamääritellyissä koluokissa (Mohd Shukran et al. 2014). Adaptiivisessa suunnittelussa tämä ei kuitenkaan

ole ongelma, koska jokainen näkymä suunnitellaan erikseen (Hing 2018). Tällöin jokaisen alustan näkymälle voidaan valita juuri sille sopivat mainosten koot niiden tarjoomasta. Näin yhden kiinteän mainoskoon sovittamisesta usealle eri näkymälle ei tarvitse huolehtia. Mainonnan optimointi on siis sulavaa adaptiivisessa käyttöliittymässä, sillä mainosten sijainti sekä koot kytetään yksilöimään jokaiselle päätelaitteelle.

5.2 Adaptiivisuuden heikkoudet

Adaptiivisen käyttöliittymän ohjelmointi on huomattavasti työläämpää verrattuna moniin muihin monialustaisen käyttöliittymäsuunnittelun menetelmiin. Syynä tähän on se, että käyttöliittymän jokaisen eri version toteuttaminen vaatii huomattavan määrän lisää CSS ja HTML -ohjelmointia. Samalla tavoin myös adaptiivisen käyttöliittymän ylläpito vaatii suuria määriä resursseja. Tuotantoprosessin yksinkertaistamisen ja resurssien säästämisen ollessa yksi monialustaisuuden kulmakivistä (Nagy 2022) nämä adaptiivisuuden haittapuolet korostuvat entisestään.

Tulevaisuudenkestävyyden varmistaminen on myös yksi adaptiivisuuden heikkouksista. Vaikka adaptiivinen suunnittelu mahdollistaa olemassaolevien ohjelmistojen käyttöalustojen laajentamisen, vaatii uuden kohdealustan tunnistaminen aina lisää työtä. Esimerkiksi siinä missä responsiivinen ohjelmisto kykenee skaalautumaan automaattisesti käytännössä jokaiselle päätelaitteelle (Ward 2017), niin adaptiivisuudessa on jokaisen uuden uniikin resoluution kohdalla tarve kehittää sille uusi kohdennettu näkymä (Hing 2018).

6. ADAPTIIVISEN JA RESPONSIIVISEN SUUNNITTELUN VERTAILU

Responsiivista ja adaptiivista suunnittelumenetelmää tutkittaessa käsiteltiin kummankin vahvuuksia sekä heikkouksia. Tässä kappaleessa vertaillaan näitä ominaisuuksia ja pyritään löytämään suunnittelumenetelmien etuja toisiinsa nähden. Taulukossa 6.1 on listattu kaikki merkittävät ominaisuudet, jotka nousivat esille adaptiivisuutta ja responsiivisuutta tutkittaessa. Lisäksi taulukossa on jokaisen ominaisuuden kohdalla se suunnittelumenetelmä, joka kyseinen ominaisuuden näkökulmasta nähdään parempana vaihtoehtona.

Taulukko 6.1. Adaptiivisuuden ja responsiivisuuden vahvuudet

OMINAISUUS	SUUNNITTELUMENETELMÄ
IMPLEMENTOINNIN KUSTANNUKSET	Responsiivisuus
TEHOKKUUS / SUORITUSKYKY	Adaptiivisuus
HAKUKONEOPTIMOINTI	Responsiivisuus
TULEVAISUUDENKESTÄVYYS	Responsiivisuus
OPTIMOINTI	Adaptiivisuus
MAINONTA	Adaptiivisuus
JÄLKI-IMPLEMENTOINTI	Adaptiivisuus
JOHDONMUKAISUUS	Responsiivisuus

Responsiivisuuden vahvuudet adaptiiviseen käyttöliittymäsuunnitteluun verrattuna painottuvat vahvasti ohjelmistokehityksen kustannuksiin. Vaikka erilaisille päätelaitteille luontevasti uudelleenmuotoutuvan käyttöliittymän toteuttaminen voi sellaisenaan olla moni-

mutkaista, on monen vaihtoehtoisen näkymän luominen usein silti työläämpää. Responsiivisuuden taloudelliset edut adaptiivisuuden rinnalla korostuvat entisestään, kun huomioidaan kaikki ohjelmiston ylläpitämiseen ja mahdolliseen jatkokehitykseen vaadittavat resurssit. Siinä missä responsiivisessa ohjelmistossa tulee huolehtia vain yhdestä käyttöliittymästä, vaatii adaptiivisen käyttöliittymän jokainen näkymä yksilöllistä ylläpitoa. Ohjelmiston kehittäjän taloudellisesta näkökulmasta myös responsiivisuuden tarjoama tulevaisuudenkestävyys on yksi sen suurimmista valttikorteista. Uuden päätelaitteen ilmaantua markkinoille tarvitsee responsiivinen ohjelmisto hyvin vähän, jos lainkaan jatkokehitystä. Adaptiivinen käyttöliittymä sen sijaan voi hyvin vaatia täysin uuden näkymän suunnittelua kohdatakseen uuden päätelaitteen asettamat vaatimukset. (Ward 2017; Hing 2018)

Responsiivisuuden ollessa selvästi vahvempi suunnittelumenetelmä kehittäjän resurssien osalta, kirii adaptiivisuus rinnalle loppukäyttäjän kohtaaman optimoinnin ja suorituskyvyn avulla. Adaptiivisessa suunnittelussa ladataan vain kyseiselle päätelaitteelle oleelliset resurssit kaikenkattavien CSS ja HTML -tiedostojen sijaan, joka takaa huomattavasti sulavammat käyttäjän toimintojen vasteajat, kuin mitä responsiivisuus kykenee tarjoamaan. Tämän vahvuuden merkitys kasvaa sitä myötä, mitä suuremmasta ja yhä laajemman kirjon eri komponentteja sisältävästä käyttöliittymästä puhutaan. (Clark 2015; Hing 2018) Samalla tavoin responsiivinen käyttöliittymä kohtaa hankaluuksia laitekohtaisessa optimoinnissa, koska kuten aiemmin on todettu, responsiivisuus edellyttää näkymän alkupeleistä suunnittelua jollekin tietylle päätelaitteen resoluutiolle (Margea et al. 2017). Adaptiivisuus sen sijaan mahdollistaa näkymän optimoinnin jokaiselle päätelaitteelle erikseen, jonka kanssa responsiivisen käyttöliittymän on hyvin vaikea kilpailla (Yousaf et al. 2018). Lisäksi on tilanteita, joissa jo olemassa olevan verkkosivuston käyttöliittymä halutaan modernisoida tukemaan monialustaisia toimintoja. Myös näissä tapauksissa adaptiivisuus on vahvemmillä, koska vaihtoehtoisten näkymien luominen ei vaadi responsiivisuudelle kriittistä koko ohjelmiston arkkitehtuurin uudelleenrakentamista (Nebeling ja Norrie 2013). Viimeinen adaptiivisuuden selkeä vahvuus responsiiviseen suunnitteluun verrattuna on mainonnan implementointi osana käyttöliittymää. Siinä missä mainosten toteuttaminen responsiivisena on yleinen haaste, on adaptiivisuuden voimavarana näkökohtaisen mainonnan tukeminen jokaiselle päätelaitteelle (Mohd Shukran et al. 2014).

Suunnittelumenetelmien ominaispiirteistä johtuvien suurimpien erojen ohella adaptiivisen ja responsiivisen suunnittelun välillä on myös vähemmän itsestäänselviä eroja. Responsiivisuuden vahvuuksia analysoitaessa esille yhtenä sen vahvuutena nousi potentiaallinen hakukoneoptimointi (Clark 2015). Tämä ei kuitenkaan tarkoita sitä, että adaptiivisuudella ei olisi mahdollista saavuttaa samoja etuja. Esimerkiksi käyttöliittymien mobiiliystävällisyys parantaa merkittävästi verkkosivustojen hakukoneoptimointia, koska usein suuri osa kokonaiskäyttäjäkannasta on mobiilikäyttäjiä (Enge 2015). Adaptiivisen käyttöliittymän mobiiliresoluutioille optimoidut näkymät mahdollistavat hyvin mobiilille kohdennetun

käyttäjäkokemuksen ja täten myös samat hyödyt SEO:n kannalta kuin responsiivisuuden mobile-first suunnittelu. On kuitenkin olennaista mainita, että laitekohtaisen optimoinnin vaatima suurempi työmäärä vaikuttaa epäsuorasti myös tehokkaan SEO:n saavuttamiseen vaadittavaan työmäärään adaptiivisuudessa. Tästä syystä responsiivisuus usein nähdäänkin hakukoneoptimoinnin kannalta tehokkaampana vaihtoehtona. Samanlaisena sivutuotteena responsiivisen suunnittelun ajatus esittää konkreettisesti sama sisältö jokaisella päätelaitteella parantaa myös käyttöliittymän johdonmukaisuutta monialustaisessa ympäristössä (Clark 2015). Vaihtoehtoisesti adaptiivisuus toimii lähes päinvastaisesti, mahdollisesti jopa kannustaen johdonmukaisuuden rikkomiseen edellyttämällä useiden eri versioiden suunnittelua käyttöliittymän näkymästä.

7. JOHTOPÄÄTÖKSET

Tutkielman tavoitteena oli analysoida responsiivista ja adaptiivista käyttöliittymäsuunnitelua monialustaisessa ympäristössä sekä saavuttaa ymmärrys siitä millaisissa tilanteissa kumpaakin menetelmää on perustellusti järkevä käyttää. Menetelmänä näiden tavoitteiden saavuttamiseksi oli molempien suunnittelumenetelmien sekä niiden vahvuuksien ja heikkouksien analysointi. Lopuksi löydettyjä vahvuuksia ja heikkouksia vertailtiin keskenään.

Vertailusta ilmeni, että molemmilla menetelmillä on selvät, omiin toteutustapoihin ja funktionaalisuuksiin perustuvat vahvuudet toisiinsa nähden. Tieteellisestä kirjallisuudesta vertailuun nousi yhteensä kahdeksan eri ominaispiirrettä joista neljän todettiin olevan responsiivisuuden kannattavuutta edustavia, kun taas loput neljä nähtiin vahvuuksina adaptiivisessa suunnittelussa. Menetelmien tarjoamien etujen jakautuminen tasan ei itsessään ole kovin yllättävää, sillä muulloin olisi perusteltua todeta toisen suunnittelumenetelmän olevan huomattavasti parempi vaihtoehto. Tällöin niiden vertailu ei olisi tieteellisessä kirjallisuudessa tai tässä tutkielmassa erityisen oleellista.

Vertailun tulosten avulla voidaan kuitenkin monialustaisen ohjelmistokehityksen määrittelmä ja tavoitteet mielessä pitäen tehdä johtopäätöksiä siitä, missä tilanteissa tietty suunnittelumenetelmä on mahdollisesti parempi vaihtoehto. Responsiivisen ratkaisun voidaan esimerkiksi todeta olevan järkevämpi ratkaisu kehitysprojekteissa, joissa tavoitellaan monialustaista tuotetta pienemmällä budjetilla. Tämä johtuu adaptiivisuuden merkittävästi suuremmista tuotanto- ja ylläpitokuluista. Lisäksi responsiivisuus sopii erityisesti monialustaisille ohjelmistoille, joiden käyttäjäkunnan odotetaan valtaosin koostuvan tiettyjen päätelaitteiden käyttäjistä. Erityisesti mobiililaitteille kohdennettu verkkosivu voidaan toteuttaa responsiivisena mobile first -menetelmällä jolloin optimointi keskitetään valtaosaan käyttäjästä, mutta muillekin päätelaitteille tarjotaan mahdollisuus päästä käsiksi samaan sisältöön.

Ohjelmisto on puolestaan perusteltua toteuttaa adaptiivisena mikäli prioriteettilistan kärjessä on erityisesti käyttöliittymän optimointi, tehokkuus sekä parhaan mahdollisen käyttäjäkokemuksen tarjoaminen jokaisella kohdealustalla. Saavuttaakseen tämän kehittäjän on kuitenkin oltava valmis uhraamaan enemmän resursseja kehitysvaiheessa sekä mahdollisesti myöhemmin laajentaakseen tuettujen päätelaitteiden valikoimaa. Vertailun pe-

rusteella adaptiivisuuden voidaan myös todeta olevan parempi vaihtoehto lisättäessä jo olemassa olevaan ohjelmistoon monialustallisia ominaisuuksia.

Käyttäjien tavoittaminen on mille tahansa julkiselle ohjelmistolle olennaista, ja täten verkkosivujen kohdalla hyvän hakukoneoptimoinnin saavuttaminen on kriittistä. Vertailun perusteella responsiivinen käyttöliittymä mahdollistaa paremman SEO:n, mutta maininnan arvoista on tuloksen kiistallisuus. Responsiivisuuden SEO:n puolesta puhuvia ja varteenotettavia lähteitä löytyi vain yksi vuodelta 2015, mutta adaptiivisuuden hakukoneoptimoinnista ei löytynyt lainkaan mainintoja. SEO:n osalta ei tässä tutkielmassa voida siis tehdä sen suurempia johtopäätöksiä.

Kaiken kaikkiaan tutkielmassa toteutetun vertailun sekä tehtyjen johtopäätösten perusteella voidaan todeta, että kehitettäessä monialustaista käyttöliittymää adaptiivinen suunnittelu mahdollistaa nimenomaan hyvän käyttöliittymän määritelmän täyttämisen jokaisella päätealustalla. Toisaalta monialustaisuuden ajatus ohjelmoi kerran, aja missä vain, eli WORA on huomattavasti enemmän keskiössä responsiivisessa suunnittelumallissa. Kaikenkattavan ratkaisun puuttuessa on oikean suunnittelumenetelmän valinta siis täysin riippuvainen käyttöliittymän toteutukseen asetetuista rajoitteista sekä siitä mitä ensisijaisesti tavoitellaan.

LÄHTEET

- Bishop, J. ja N. Horspool (2006). "Cross-Platform Development: Software that Lasts". *Computer* 39.10, s. 26–35. DOI: 10.1109/MC.2006.337.
- Chmielewski, Jacek (2015). "Run-time UI adaptation in the context of the Device-Independent Architecture". Teoksessa: *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*. Polish Information Processing Society (PIPS), s. 1157–1162.
- Clark, Jason A. (2015). *Responsive web design in practice*. Library technology essentials. Lanham: Rowman ja Littlefield.
- Dominguez, Ana, Julian Florez, Alberto Lafuente, Stefano Masneri, Inigo Tamayo ja Mikel Zorrilla (2021). "A Model for User Interface Adaptation of Multi-Device Media Services". *IEEE transactions on broadcasting* 67.3, s. 606–618.
- Edge, Ryan. (2022). *Cross-Platform UIs with Flutter : Unlock the Ability to Create Native Multiplatform UIs Using a Single Code Base with Flutter 3*. Birmingham: Packt Publishing, Limited.
- Enge, Eric (2015). *The art of SEO : mastering search engine optimization*. 3rd ed. Sebastopol, California: O'Reilly Media, Inc.
- Frain, Ben. (2020). *Responsive Web Design with HTML5 and CSS - Third Edition*. Lanham: Packt Publishing.
- García-Moya Herrera, Álvaro (2013). *Multiplatform mobile software development*. Tietotekniikan laitokset - Department of Pervasive Computing.
- Griffiths, Dawn (2015 - 2015). *Head first android development : a brain-friendly guide*. First edition. Head first Head first Android development. Beijing, [China: O'Reilly.
- Gustafson, Aaron (2015). *Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement*. Hoboken: Pearson Education, Limited.
- Heath, Martin (2017). *The effect of multi-device design on website efficiency and user preference*. Viestintätieteiden tiedekunta - Faculty of Communication Sciences.
- Hing, Philippe (2018). *Practical Web Design. 1st edition*. Packt Publishing.
- Margea, Romeo, Camelia Margea, Bogdan Veche ja Călin Hurbean (2017). "Mobile First. Current Trends and Practices in Website Design". *Analele Universității "Dunărea de Jos" Galați. Fascicula I, Economie și informatica aplicata* 23.3, s. 164–173.
- Mohd Shukran, Mohd Afizi bin, Chee Kong Wong ja Fong Yee Lee (2014). "Responsive Web Design for Mobile Device Screen Optimization". *Applied Mechanics and Materials* 548-549. Achievements in Engineering Sciences, s. 1460–1464.

- Morales Díaz, Leonel Vinicio (2022). "What is a User Interface, Again? A Survey of Definitions of User Interface: Our Shared and Implicit Understanding of the Concept of User Interface". Teoksessa: *Proceedings of the 9th Mexican International Conference on Human-Computer Interaction*. MexIHC '22. Virtual Event, Mexico: Association for Computing Machinery. DOI: 10.1145/3565494.3565504.
- Mori, Giulio, Fabio Paterno ja Carmen Santoro (2004). "Design and development of multi-device user interfaces through multiple logical descriptions". *IEEE transactions on software engineering* 30.8, s. 507–.
- Nagy, Robert (2022). *Simplifying application development with Kotlin multiplatform mobile : write robust native applications for IOS and android efficiently*. Birmingham, England ; Packt Publishing.
- Nebeling, Michael ja Moira C. Norrie (2013). "Responsive Design and Development: Methods, Technologies and Current Issues". Teoksessa: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 7977. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, s. 510–513.
- Shah, Kewal, Harsh Sinha ja Payal Mishra (2019). "Analysis of Cross-Platform Mobile App Development Tools". Teoksessa: *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, s. 1–7. DOI: 10.1109/I2CT45611.2019.9033872.
- Sridevi, S (2014). "User interface design". *International Journal of Computer Science and Information Technology Research* 2.2, s. 415–426.
- Ward, Chris. (2017). *Jump Start Responsive Web Design*. Victoria, Australia: Sitepoint.
- Vranić, Valentino ja L'uboš Staraček (2014). "MDA based multiplatform mobile application modeling with platform compliant user interfaces". *INFOCOMP Journal of Computer Science* 13.2, s. 34–43.
- Yousaf, Nazish, Aleena Arshad, Muhammad Nouman ja Umar Arshad (2018). "Towards adaptive and responsive web design: A systematic literature review". *Language* 1, s. 40.