



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MINNA VIITANEN

**FULL-STACK -SUUNNITTELIJAN ROOLIN HYÖDYT JA
HAASTEET KETTERÄSSÄ OHJELMISTOKEHITYKSESSÄ**

Diplomityö

Tarkastajat: professori Kari Systä ja
professori Kaisa Väänänen

Tarkastajat ja aihe hyväksytty 31.
lokakuuta 2018

TIIVISTELMÄ

MINNA VIITANEN: Full-stack -suunnittelijan roolin hyödyt ja haasteet ketterässä ohjelmistokehityksessä

Tampereen teknillinen yliopisto

Diplomityö, 78 sivua, 9 liitesivua

Joulukuu 2018

Tietotekniikan diplomi-insinöörin tutkinto-ohjelma

Pääaine: Ohjelmistotuotanto

Tarkastajat: professori Kari Systä ja professori Kaisa Väänänen

Avainsanat: full-stack -suunnittelija, full-stack -suunnittelu, ohjelmistokehitys, ketterä kehitys

Diplomityön tavoitteena oli selvittää full-stack -suunnittelijan roolin hyötyjä ja haasteita ketterässä ohjelmistokehityksessä. Full-stack -suunnittelijan rooli oli työn toteuttamisen aikaan vielä melko tutkimaton, eikä roolin termi ja määritelmä olleet vielä vakiintuneet. Työn tarkoituksena oli myös kerätä mielipiteitä full-stack -suunnittelijan roolin hyödyistä ja haasteista sekä ratkaisuehdotuksia rooliin liittyviin haasteisiin. Lisäksi työssä oli tarkoitus selvittää työn aloitushetkellä roolista käytössä olleen termin ja määritelmän paikansapitävyyksiä.

Diplomityössä toteutettiin tutkimus kirjallisuusselvitysten, verkkoaineistoanalyysin ja kyselyn avulla. Kirjallisuusselvityksellä perehdyttiin aiempiin tutkimuksiin käyttäjäkokemussuunnittelijoiden työstä ohjelmistokehitysprojekteissa. Verkkoaineistoanalyysin avulla etsittiin tietoa full-stack -suunnittelijan roolista, sen termistä sekä määritelmästä. Verkkoaineisto piti sisällään ammattilaisten blogeja aiheesta sekä aiheeseen liittyviä muita keskustelupalstoja ja verkkosivuja. Kyselyn avulla kartoitettiin ketteriä menetelmiä käytävien suomalaisten ohjelmistoyritysten työntekijöiden mielipiteitä käyttäjäkokemussuunnittelu- ja ohjelmistokehitysprosessista tutkimushetkellä, yleisesti full-stack -suunnittelijan roolista, sen hyödyistä ja haasteista, sekä kerättiin ehdotuksia sille, kuinka esiin tulleita haasteita voitaisiin ratkaista.

Eräs tutkimuksessa esiin nousut asia oli se, että kyselyyn vastanneet työntekijät kokivat, että full-stack -suunnittelijan roolin määritelmään sopivia ihmisiä on vaikeaa löytää. Kuitenkin saman kyselyn tutkimuksessa myös selvisi, että suuri osa kyselyyn vastanneista toimi täysin tai osittain full-stack -suunnittelijan roolin määritelmän mukaisesti projekteissa.

Kirjallisuuden ja kyselyn pohjalta on esitetty tutkimuksessa selvinneet full-stack -suunnittelijan rooliin liittyvät hyödyt ja haasteet sekä yrityksen että työntekijöiden näkökulmasta. Lisäksi työ esittää ratkaisuehdotuksia tutkimuksessa olennaisimpiin löytyneisiin haasteisiin.

ABSTRACT

MINNA VIITANEN: Tampere University of Technology thesis template

Tampere University of Technology

Benefits and challenges of a role of full-stack designer in agile development, 78 pages, 9 Appendix pages

December 2018

Master's Degree Programme in Information Technology

Major: Software Engineering

Examiners: Professor Kari Systä and Professor Kaisa Väänänen

Keywords: full-stack designer, full-stack design, software development, agile development

The goal of the master's thesis was to find the benefits and challenges of a role of full-stack designer in agile development. The role of full-stack designer was yet rather unexamined during this study, and the term and definition of the role were not yet established. The purpose was also to gather opinions related to the benefits and challenges of the role of full-stack designer and proposed solutions for the challenges related to the role. In addition, the purpose of the work was to examine the validities of the term and definition of the role.

Master's thesis was conducted as a research that included a literature review, web content analysis and an online survey. The literature review was performed to study the previous research of work of a user experience designer in software development projects. The web content analysis was used to find information about the role, term and definition of full-stack designer. The web content analysis included related blogs by professionals in the field and as well as other related discussions and websites. Survey was used to map opinions of the employees working in Finnish software companies using agile methods about design and software development process, the role of full-stack designer, its benefits and challenges, as well as solutions how to resolve the challenges that came up during the survey.

A matter that came up during the study was that the employees that answered the survey felt that persons that would match the definition of full-stack designer are hard to find. However, from the same survey one could find out that the major part of the people that took part in the survey was already performing tasks fully or partially by the definition of full-stack designer.

The benefits and challenges related to the role of full-stack designer based on the literature and the survey are presented from both a company's and employees' point of view. Additionally, the proposed solutions for the the most essential challenges emerged from the study are also presented.

ALKUSANAT

Iso kiitos diplomityöohjaajilleni professori Kari Syställe sekä professori Kaisa Väänäselle rakentavasta palautteesta, sopivasta painostuksesta diplomityön saattamiseksi loppuun sekä etenkin ymmärryksestä nopeaan aikatauluun ja valmistumistavoitteeseen. Haluan kiittää kaikkia kyselyyn osallistuneita alan ammattilaisia sekä niitä yrityksiä, jotka välittivät tutkimuksen kyselyä eteenpäin työntekijöilleen. Kiitos myös Vincitin työntekijöille, jotka saivat minut uskomaan, että tutkimuksen valmistuminen ja diplomityön saattaminen loppuun on todella mahdollista.

Eryityisesti haluan kiittää vanhempiani kaikesta tuesta koko yliopisto-opintojeni aikana. Lisäksi haluan kiittää ystäviäni kaikesta saamastani tuesta, ymmärryksestä ja avusta koko tämän – välillä suuriakin tunteita herättäneen – diplomityöprosessin aikana. En olisi onnistunut tässä projektissa ilman teitä!

Tampereella, 19.11.2018

Minna Viitanen

SISÄLLYSLUETTELO

1.	JOHDANTO	1
1.1	Työssä käytettävät termit	1
1.2	Tutkimuksen tausta	2
1.3	Tutkimuksen tavoitteet, tutkimuskysymykset ja aiheen rajausta	3
1.4	Tutkimusmenetelmät ja –prosessi	4
1.5	Diplomityön rakenne	4
2.	KETTERÄ OHJELMISTOKEHITYS SCRUM-MENETELMÄLLÄ	5
2.1	Scrum lyhyesti	6
2.2	Taustateoria ja arvot	7
2.3	Roolit ja tehtävät	7
2.4	Prosessi ja tapahtumat	8
3.	KÄYTTÄJÄKESKEINEN SUUNNITTELU OHJELMISTOKEHITYKSESSÄ	10
3.1	Käyttäjakeskeinen suunnittelu lyhyesti	10
3.2	Suunnittelu integroituna ketteryteen	12
3.2.1	Etupainotteinen suunnittelu	12
3.2.2	BoB (Best of Both Worlds) -viitekehys	13
3.2.3	Kehittäjät ja suunnittelu	14
3.3	Käyttäjäkokeussuunnittelijan rooli ketterässä ohjelmistokehityksessä ...	15
3.4	Käyttäjäkokeussuunnittelijan yhteistyö eri roolien kanssa	19
4.	FULL-STACK -SUUNNITTELIJAN ROOLIN MÄÄRITTELY	24
5.	TUTKIMUSPROSESSI	29
5.1	Tutkimuksen tavoitteet	29
5.2	Tutkimuksen eteneminen	30
5.3	Tutkimuksen aineisto	30
5.4	Tutkimusmenetelmät	31
5.5	Tutkimusaineiston analysointi	34
6.	TUTKIMUSTULOKSET	35
6.1	Yleistä tietoa kyselyyn vastanneista	35
6.2	Työntekijöiden oma kiinnostus ja osaaminen suunnittelun osa-alueilla	38
6.3	Nykyinen kokemus suunnittelusta ja ohjelmistokehityksestä	39
6.4	Full-stack -suunnittelijan roolin tunnettavuus ja määritelmä	41
6.5	Full-stack -suunnittelijan roolin hyödyt, haasteet sekä ratkaisut haasteisiin	43
6.5.1	Roolin hyödyt	45
6.5.2	Roolin haasteet	46
6.5.3	Ratkaisut haasteisiin	48
6.6	Työntekijöiden kokemus roolin toimivuudesta yrityksessä	50
6.7	Työntekijöiden mielipiteet full-stack -suunnittelijasta	52
6.8	Työntekijöiden kiinnostus roolissa toimimiseen	53

6.9	Muuta kommentoitavaa full-stack -suunnittelijan roolista	54
6.10	Palaute kyselystä.....	54
7.	TULOSTEN TARKASTELU.....	56
7.1	Tutkimustulosten vertailu aiempiin tutkimuksiin	56
7.2	Full-stack -suunnittelijan hyödyt ja haasteet yritykselle.....	59
7.2.1	Prosessiin, ihmisiin sekä projekteihin liittyvät hyödyt	60
7.2.2	Prosessiin, ihmisiin sekä projekteihin liittyvät haasteet	61
7.3	Työntekijöiden näkemykset full-stack -suunnittelijan roolin hyödyistä ja haasteista	63
7.3.1	Hyödyt	63
7.3.2	Haasteet.....	65
7.4	Full-stack -suunnittelijan rooliin liittyvien haasteiden ratkaisu.....	66
7.4.1	Opiskelu ja koulutus	67
7.4.2	Full-stack -suunnittelijan roolin termin ja määritelmän tekeminen tunnetuksi.....	67
7.4.3	Full-stack -suunnittelijan resurssointi järkevästi	67
7.4.4	Projektin tehtävien tekemisen painottaminen osaamisen ja kiinnostuksen perusteella.....	67
7.4.5	Projektissa toimivien roolien ja niiden määrien järkevä valinta ...	68
7.4.6	Muiden työntekijöiden tuki.....	68
7.4.7	Ohjelmointia helpottavien menetelmien ja työkalujen käyttäminen	68
7.4.8	Omien vahvuuksiensa tunteminen ja halu kehittyä	68
7.5	Tulosten luotettavuus	68
8.	JOHTOPÄÄTÖKSET	70
8.1	Tutkimuksen keskeiset tulokset	70
8.2	Full-stack -suunnittelija terminä	71
8.3	Tutkimuksen onnistumisen arviointi.....	72
8.4	Jatkotutkimusmahdollisuudet	73
	LÄHTEET.....	75
	LIITE A: KYSELYRUNKO	79

KUVALUETTELO

Kuva 1.	<i>Esimerkki vesiputousmallista perustuen lähteeseen [18]</i>	5
Kuva 2.	<i>Scrum-prosessi perustuen lähteisiin [36, 33].....</i>	8
Kuva 3.	<i>Käyttäjakeskeinen suunnitteluprosessi perustuen lähteeseen [35]</i>	11
Kuva 4.	<i>Syn suunnittelumalli perustuen lähteeseen [37]</i>	13
Kuva 5.	<i>BoB-viitekehys perustuen lähteeseen [24]</i>	14
Kuva 6.	<i>Esimerkki käyttäjäkokeussuunnittelijan roolien tehtävien jakautumisesta eri iteraatioissa perustuen lähteeseen [34].....</i>	17
Kuva 7.	<i>Tehtävien jako eri roolien välillä perustuen lähteisiin [23, 24].....</i>	21
Kuva 8.	<i>Havainnollistava kuva full-stack -suunnittelijan roolin tehtävistä perustuen lähteisiin [6, 15, 26, 20, 41].....</i>	25
Kuva 9.	<i>Kyselyyn vastanneiden työntekijöiden ilmoittamien yritysten kokoja-kauma</i>	36
Kuva 10.	<i>Vastaajien projektitiimien henkilömäärä.....</i>	37
Kuva 11.	<i>Vastaajien roolit yrityksissä.....</i>	37
Kuva 12.	<i>Vastaajien työkokemukset ohjelmistokehityksestä, käyttäjäkokeussuunnittelusta sekä ketteristä mentelmistä</i>	38
Kuva 13.	<i>Vastaajien kiinnostuksen ja työkokemuksen korrelaatio UI-suunnittelun osa-alueella</i>	39
Kuva 14.	<i>Suunnittelijoiden ja kehittäjien kiinnostus käyttöliittymien toteutukseen</i>	39
Kuva 15.	<i>Vastaajien mielipiteet suunnittelijoiden ja kehittäjien poikkiteollisesta osaamisesta</i>	40
Kuva 16.	<i>Vastaajien mielipiteet väittämistä liittyen siihen, jos suunnittelija toteuttaa omat designinsa.....</i>	40
Kuva 17.	<i>Vastaajien mielipiteet suunnittelijan rooliin liittyvistä väittämistä</i>	41
Kuva 18.	<i>Full-stack -suunnittelijan tehtäviä tuotekehityksessä web-projektin mukaan esitettynä.....</i>	42
Kuva 19.	<i>Vastaukset kysymykseen: Olitko aiemmin kuullut termiä "full-stack -suunnittelija"(engl. full-stack designer)?.....</i>	42
Kuva 20.	<i>Vastaukset kysymykseen: Oletko samaa mieltä roolin määritelmän kanssa?</i>	43
Kuva 21.	<i>Vastaajien määrät samassa projektissa suoritettujen tehtävien mukaan</i>	44
Kuva 22.	<i>Vastaajien vastaukset kysymykseen: ”Oletko joskus työskennellyt projektissa, jossa joku tiimin jäsen on toiminut full-stack -suunnittelijan roolissa tai tehnyt roolia vastaavia tehtäviä?”</i>	45
Kuva 23.	<i>Vastaajien mielipiteet full-stack -suunnittelijan rooliin liittyvistä väittämistä</i>	52
Kuva 24.	<i>Suunnittelijoiden, kehittäjien sekä molemmissa rooleissa toimivien vastaajien kiinnostus full-stack -suunnittelijan rooliin</i>	53

LYHENTEET JA MERKINNÄT

Back-end -kehitys	Palvelinpuolen kehitys web-ohjelmistokehityksessä [13]
Back-end -kehittäjä	Ohjelmistokehittäjä, joka tekee back-end -kehitystä
Design	Tämän työn tutkimusosassa käytetty termi käyttöliittymäsuunnittelusta
Front-end -kehitys	Selainpuolen kehitys web-ohjelmistokehityksessä [13]
Front-end -kehittäjä	Ohjelmistokehittäjä, joka tekee front-end -kehitystä
Full-stack -kehittäjä	Kehittäjä, joka kehittää sekä front-end- että back-end -puolia web-ohjelmistokehityksessä
Full-stack -suunnittelija	Tässä työssä käytetty termiä käyttäjäkeskeisestä suunnittelijasta, joka tekee sekä käyttäjäkokemussuunnittelua, käyttöliittymäsuunnittelua, vuorovaikutussuunnittelua sekä käyttöliittymien toteutusta [6, 10]
Full-stack -suunnittelu	Tässä työssä käytetty termiä käyttäjäkeskeisestä suunnittelusta, jossa toteutetaan kaikkia tuotesuunnitteluun liittyviä osa-alueita (käyttäjäkokemussuunnittelu, käyttöliittymäsuunnittelu, vuorovaikutussuunnittelu sekä käyttöliittymien toteutus tai front-end-kehitys) [6, 10]
Kehittäjä	Tässä työssä lyhenne ohjelmistokehittäjästä
Ketterä ohjelmistokehitys	Ohjelmistokehitysmenetelmä, joka perustuu tiettyihin arvoihin ja periaatteisiin [2, 3]
Käyttäjäkokeumus-suunnittelu	Suunnittelun osa-alue, jossa suunnitellaan tuotetta käyttäjätarpeiden näkökulmasta [35]
Käyttöliittymäsuunnitelma	Tässä työssä käyttäjäkokemussuunnittelun lopputuote (engl. design)
Käyttöliittymäsuunnittelu	Suunnittelun osa-alue, jossa suunnitellaan kehitettävän tuotteen käyttöliittymää ja sitä, miltä tuote näyttää [35]
Suunnittelija	Tässä työssä yleisesti suunnittelija, joka tekee käyttäjäkeskeisestä suunnittelusta
Suunnittelu	Tässä työssä yleisesti käyttäjäkeskeisestä suunnittelusta käytettävä termi
UI-suunnittelu	Tämän työn tutkimusosassa käytetty termi käyttöliittymäsuunnittelusta
UX-suunnittelu	Tässä työn tutkimusosassa käytetty termi käyttäjäkokemussuunnittelusta
Vuorovaikutussuunnittelu	Suunnittelun osa-alue, jossa suunnitellaan se, miten käyttäjät suorittavat tehtäviä kehitettävän tuotteen avulla [35]

1. JOHDANTO

Olen työskennellyt ohjelmistoalalla muutaman vuoden ohjelmistosuunnittelijana pääasiassa web-projekteissa. Olen aina ollut kiinnostunut selainpuolen (engl. front-end) kehityksestä ja erityisesti kaikista sovelluksen ulkoasuun liittyvistä tehtävistä. Työskennellessäni ohjelmistosuunnittelijana tuntui luonnolliselta laajentaa omaa osaamista tekemällä myös palvelinpuolen (engl. back-end) kehitystä sekä mobiilikehitystä. Huomasin kuitenkin, että selainpuolen kehitys veti enemmän puoleensa visuaalisen luonteensa ja käyttäjäläheisyytensä vuoksi.

Tästä syystä olen opintojeni aikana suorittanut ohjelmistotuotannon kurssien lisäksi käyttöliittymäsuunnittelun kursseja. Sain nykyisessä työpaikassani Vincitillä mahdollisuuden kokeilla käyttäjäkokemussuunnittelijana toimimista. Halusin kuitenkin tämän lisäksi jatkaa myös ohjelmistokehittäjän roolissa.

Kuulin kollegan mainitsevan termin ”full-stack -suunnittelija” (engl. full-stack designer). Kiinnostuin etsimään tietoa käyttäjäkokemussuunnittelijoista, jotka myös toteuttavat suunnittelemiensa käyttöliittymiä. Vaikka roolin kuvaukseen sopivia toimijoita löytyy ohjelmistoalalta, on aihe suhteellisen vähän tutkittu. Rooli, siihen liittyvä osaaminen, sen vastualueet sekä hyödyt ja haasteet yritykselle ovat muutaman vuoden aikana olleet erilaisilla käyttäjäkokemussuunnittelun sekä ohjelmistokehityksen keskustelupalstoilla esillä [41, 7, 1]. Roolilla ei kuitenkaan vielä ole vakiintunutta termiä, vaikka kiinnostus rooliin on kasvanut. Tästä syntyi idea tälle diplomityölle.

1.1 Työssä käytettävät termit

Tässä työssä tutkitaan erästä käyttäjäkeskeiseen suunnitteluun liittyvää poikkitieteellistä roolia ”**full-stack -suunnittelija**”. Tutkittu rooli ”full-stack -suunnittelija” on tässä työssä osittainen käännös sen englanninkielisestä termistä ”full-stack designer”. Termi ei ole vielä vakiintunut alalla. Termi on tässä työssä käännetty ja muodostettu mukaillen ohjelmistokehityksen puolelta tuttua termiä ”full-stack -kehittäjä”. Full-stack -kehittäjällä tarkoitetaan erityisesti web-ohjelmistokehityksen puolella useammalla osa-alueella työskentelevää ohjelmistokehittäjää [10]. Full-stack -suunnittelijan termin nimeämisellä edellä mainitulla tavalla – full-stack -kehittäjän roolin termiä mukaillen – on pyritty tekemään full-stack -suunnittelijan termistä johdonmukainen full-stack -kehittäjän termin kanssa. Lisäksi termistä on tällä tavoin pyritty tekemään ymmärrettävämpi sellaisille lukijoille, jotka jo tietävät termin full-stack -kehittäjä. Englannin kielestä johdettujen full-stack -alkuisten, työntekijöiden rooleja kuvaavien termien on tarkoitus määritellä, että tällaisessa roolissa toimivat henkilöt työskentelevät useammalla tuotesuunnittelun tai -kehityksen

osa-alueella. Termillä ”full-stack -suunnittelija” tarkoitetaan siis sellaista käyttäjakeskeistä suunnittelijaa, joka työskentelee useammalla kuin yhdellä suunnittelun osa-alueella sekä jonkin verran ohjelmistokehityksen puolella tuotekehitysprosessia [6, 10]. Lisäksi full-stack -suunnittelulla viitataan tekstissä full-stack -suunnittelijan rooliin liittyvien tehtävien prosessiin. Full-stack -suunnittelijan roolin määritelmä on kerrottu tarkemmin luvussa 4.

Tässä työssä käytetään seuraavia käyttäjakeskeisen suunnittelun osa-alueiden termejä: käyttäjäkokeemussuunnittelu, käyttöliittymäsuunnittelu sekä vuorovaikutussuunnittelu. Käyttäjakeskeistä suunnittelijaa kutsutaan tässä työssä yleisesti termillä ”**käyttäjäkokeemus-suunnittelija**”. Termit ovat kuvattu seuraavassa aliluvussa 1.2 sekä luvussa 3.

Tutkimustuloksissa kahteen edellä mainittuun käyttäjakeskeisen suunnittelun osa-alueeseen viitataan termeillä ”UX-suunnittelu” tarkoitettaessa käyttäjäkokeemussuunnittelua, ”UX-suunnittelija” tarkoitettaessa käyttäjäkokeemussuunnittelijaa, ”UI-suunnittelu” tarkoitettaessa käyttöliittymäsuunnittelua sekä ”UI-suunnittelija” tarkoitettaessa käyttöliittymäsuunnittelija. Näiden termien käyttöä on perusteltu lisää luvussa 6.

Ohjelmistokehittäjästä käytetään tässä työssä lyhennettä ”kehittäjä”. Lisäksi tässä työssä käytetään englanninkielisestä termistä ”design” termiä ”**käyttöliittymäsuunnitelma**”. Käyttöliittymäsuunnitelma tarkoittaa käyttäjakeskeisen suunnittelun lopputuotetta eli sitä, miltä suunniteltu tuote näyttää.

1.2 Tutkimuksen tausta

Yritysten voimavaroja ovat työntekijä ja työntekijän osaaminen. Yrityksen kannalta eräs olennainen asia on se, miten henkilöstöresurssija voidaan hyödyntää tehokkaasti. Pelkäämään teknisen työn tehokkuutta lisäämällä on vaikeaa tasapainottaa puutteita suunnittelemisessa, organisoimisessa, henkilöstöhallinnossa, ohjaamisessa ja johtamisessa. Ohjelmistokehitys on sekä ihmis- että tietokeskeistä, mikä tekee ihmisistä tärkeimmän voimavaran ohjelmistokehityksessä. Jos oikeat ihmiset eivät tee oikeita työtehtäviä, se voi olla haitallista ohjelmistoprojektin onnistumiselle. [31] Yritykset ovat alkaneet ymmärtää, että tavoitteiden saavuttaminen vaatii tuotteen, joka vastaa asiakkaiden tarpeita. Käyttäjäkokeemussuunnittelu (engl. User Experience Design) on olennainen osa tuotteen kehitystä, mutta käyttäjäkokeemussuunnitteluprosessi voi olla pitkä ja kallis. [6]

Ohjelmistotuotannossa roolit on jaettu karkeasti käyttäjäkokeemussuunnittelijoihin sekä ohjelmistokehittäjiin [13]. Käyttäjäkokeemussuunnittelijat vastaavat käyttäjäkokeemuksen suunnittelusta eli siitä, minkälaisena käyttäjä kokee järjestelmän ja käyttäjän välisen vuorovaikutuksen [35, 13], ja kehittäjät vastaavat toteutuksesta [13].

Web-sovelluskehityksessä kehittäjien rooleja ovat front-end- sekä back-end -kehittäjät (ohjelmistokehitykseen) [13]. Henkilöä, joka työskentelee sekä front-end- että back-end -kehittäjänä kutsutaan full-stack -kehittäjäksi. Viime vuosina myös käyttäjäkokeemussuunnittelun puolella on alettu puhua full-stack -suunnittelusta (engl. Full-Stack Design) sekä

full-stack -suunnittelijasta. Näiden termien käyttö on kuvattu edellisessä aliluvussa 1.1 ja määritelmät tarkemmin luvussa 4.

Full-stack -suunnittelu on saanut verkossa paljon huomiota. Canzibanin mukaan työmarkkinoilla on kysyntää sellaiselle käyttäjäkokemussuunnittelijalle, joka työskentelee useammalla kuin yhdellä suunnittelun osa-alueella. [6] Suunnitteluprosessissa kommunikoidaan ja jaetaan paljon tietoa, kuten suunnittelun visiota tai ideaa. Visio saattaa menettää alkuperäistä tarkoitustaan, kun sitä välitetään erilaisten työkalujen kautta. Lisäksi projektissa eri rooleissa toimivat tekijät – kuten eri sidosryhmät, käyttäjäkokemussuunnittelijat ja kehittäjät – saattavat vaikuttaa visioon matkalla. Dashin mukaan kehittäjien on vaikea ymmärtää käyttöliittymäsuunnitelmia, ja se lisää iteraatioiden määrää ohjelmistoprosessissa. Full-stack -suunnittelijan rooli on noussut esiin vastaamaan tähän haasteeseen sujuvoitukseen käyttöliittymäsuunnitelmien toteuttamista. [10]

Useimmat käyttäjäkokemussuunnittelijat eivät toteuta niitä käyttöliittymiä, joita itse suunnittelevat. Käyttäjäkokemussuunnittelijat työskentelevät kehittäjien kanssa säännöllisesti ja varmistavat, että heidän tekemänsä suunnitelma säilyy myös toteutuksessa. On tärkeää, että käyttäjäkokemussuunnittelijat pysyvät mukana koko ohjelmistokehitysprosessin ajan, toteutusvaihe mukaan lukien. [27] Myös Kuusisen [22] mukaan teknologisten tekijöiden vaikutusten tutkimista käyttäjäkokemussuunnittelijan työssä on ollut vielä vähän ja tutkimusta tarvittaisiin lisää.

1.3 Tutkimuksen tavoitteet, tutkimuskysymykset ja aiheen rajaus

Tutkimuksen aiheena oli full-stack -suunnittelijan roolin hyödyt ja haasteet ketterässä ohjelmistokehityksessä sekä yrityksen että työntekijöiden näkökulmasta. Tutkimuksen tavoitteena oli selvittää, mitä hyötyjä ja haasteita full-stack -suunnittelijan roolista on yritykselle sekä työntekijöille verrattuna nykyiseen toimintamalliin ketterässä ohjelmistokehityksessä. Lisäksi työssä määritellään tarkemmin full-stack -suunnittelijan rooli erityisesti tämän työn kannalta.

Tutkimuskysymyksiä asetettiin kolme:

1. Mitä hyötyjä ja haasteita full-stack -suunnittelijan roolista on yritykselle?
2. Mitkä ovat työntekijöiden näkemykset ja kokemukset full-stack -suunnittelijan roolin hyödyistä ja haasteista?
3. Miten full-stack -suunnittelijan roolin haasteet voidaan ratkaista?

Tutkimus toteutettiin ketterän ohjelmistokehityksen viitekehityksessä. Ketterä ohjelmistokehitys otettiin huomioon erityisesti tässä työssä toteutetussa tutkimuksessa, jossa tutkimus kohdistui ohjelmistoyrityksiin, jotka toteuttivat ohjelmistoja ketterin menetelmin.

1.4 Tutkimusmenetelmät ja –prosessi

Full-stack -suunnittelijan rooli ja sen termi on vielä melko tutkimatton. Työssä selvitettiin kirjallisuusselvitysten, kyselyn sekä verkkoaineistoanalyysin avulla full-stack-suunnittelijan hyötyjä ja haasteita työntekijälle ja yritykselle ketterässä ohjelmistokehityksessä.

Tutkimuksessa hyödynnettiin tieteellistä aineistoa siltä osin kuin sitä oli saatavilla. Etenkin ohjelmistokehityksestä ja käyttäjäkokemussuunnittelun organisoinnista yrityksissä löytyi aiempaa tutkimusta, jota voitiin käyttää työn teoreettisena perustana. Lisäksi roolista ja määritelmästä etsittiin tietoa verkkoaineistoanalyysin avulla, jolla saatiin suuntaa antavaa teoriaa full-stack -suunnittelijan roolille. Teoreettisen tutkimuksen lisäksi työssä toteutettiin empiiristä tutkimusta. Tutkimuksessa toteutettiin kysely, joka suunnattiin alalla toimiville suunnittelijoille sekä kehittäjille ketteriä menetelmiä hyödyntävissä ohjelmistoyrityksissä.

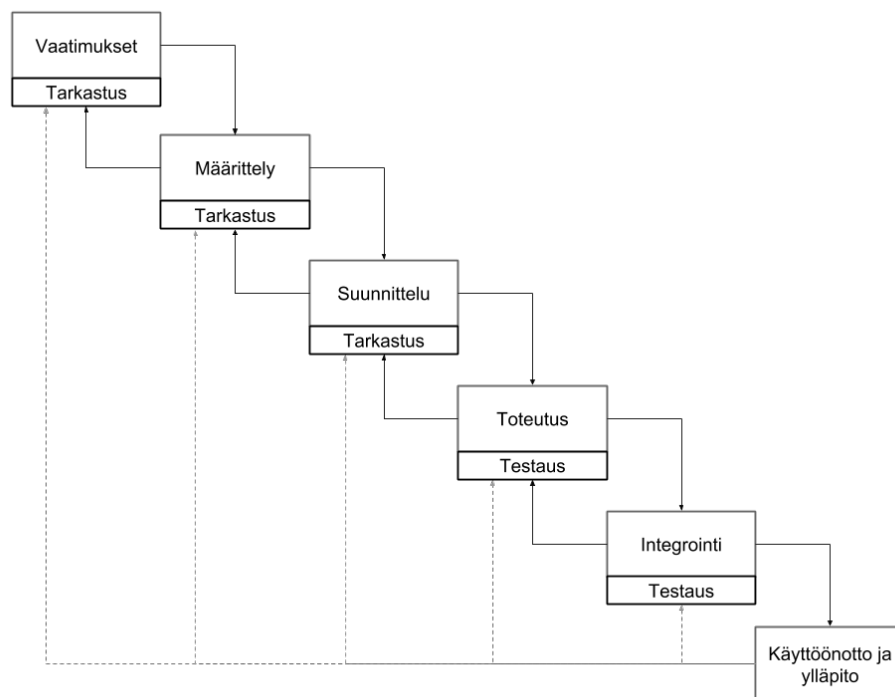
1.5 Diplomityön rakenne

Diplomityön luvussa 2 käsitellään työn aiheeseen liittyvää teoriaa ketterästä ohjelmistokehityksestä ja luvussa 3 käyttäjäkeskeisestä suunnittelusta sekä miten käyttäjäkeskeistä suunnittelua nykyisellään otetaan mukaan ohjelmistokehitykseen eli käyttäjäkokemussuunnittelijoiden ja ohjelmistokehittäjien (myöhemmin kehittäjät) yhteistyöstä. Lisäksi luvussa 4 määritellään vielä tässä työssä tutkittu rooli, roolin termi sekä mitä roolin termillä tarkoitetaan tässä työssä. Samassa luvussa toteutetaan myös verkkoaineistoanalyysia.

Luvussa 5 käydään läpi työn empiirinen osuus eli tutkimusmenetelmät ja -prosessi. Tämän jälkeen luvuissa 6 ja 7 kerrotaan tutkimuksen tulokset sekä tarkastellaan saatuja tuloksia. Lopuksi vielä esitellään työn johtopäätökset luvussa 8.

2. KETTERÄ OHJELMISTOKEHITYS SCRUM -MENETELMÄLLÄ

Nykyään yritykset suosivat ohjelmistokehityksessä ketterää ohjelmistokehitystä perinteisen ohjelmistokehitysmallin sijaan. Perinteisenä ohjelmistokehitysmalliksi mielletään vaihejakomallia, jossa kehitystyö on jaettu vaiheisiin [18, 36]. Tavallisin vaihejakomalli on niin kutsuttu vesiputousmalli (engl. waterfall) [18]. Vesiputousmalli perustuu tarkkoihin määrittelyihin sekä suunnitteluvaiheeseen, jossa lopputuote suunniteltiin ja dokumentoitiin huolellisesti. Tämän jälkeen edettiin mallin määräämien vaiheiden mukaan kohti lopputuotetta [18, 36], kuten on esitetty 1. Kaikkiin vaiheisiin liittyy laadunvarmistustoimenpiteitä, esimerkiksi tarkastuksia, katselmoiteja ja testausta [18, 36].



Kuva 1. Esimerkki vesiputousmallista perustuen lähteeseen [18]

Vaikka vesiputousmalli toimi loogisesti ja organisoidusti, vesiputousmallia on kritisoitu jäykkyydestä. Kun suunnitteluvaiheessa on määritelty tuotteen vaatimukset, vesiputousmalli ei salli muutoksia missään projektin vaiheessa. Tällöin uusien ideoiden ja innovaatioiden mukaan ottaminen kesken projektin ei ole mahdollista, millä voitaisiin kasvattaa lopputuotteen arvoa. [36] Myös Haikala et al. mainitsevat, ettei käytännön ohjelmistokehitys voi koskaan edetä täysin vesiputousmallin mukaisesti, sillä vaatimuksilla on tapana muuttua ja osa vaatimuksista selviää vasta projektin aikana [18].

Lisäksi vaatimusten dokumentointi on raskasta ja vaatimusedokumentit ovat isoja, mikä lisää mahdollisuutta sille, että dokumentti ei tule luetuksi. Väärinymmärryksen mahdollisuus on myös suuri, koska kriittiset asiat kommunikoidaan pääasiassa näiden isojen dokumenttien kautta. [36]

Ketterä ohjelmistokehitys on termi joukolle tiettyihin arvoihin ja periaatteisiin keskittyneitä menetelmiä ja käytäntöjä [2]. Termi ”ketterä” syntyi vuoden 2001 alussa, jolloin ryhmä ohjelmistokehityksen osajia kerääntyi jakamaan ideoita ja näkökulmia ohjelmistokehityksestä. Tämän tapaamisen myötä syntyi Ketterän ohjelmistokehityksen julistus (engl. Agile Manifesto [3]), joka keräsi yhteen ketterän kehityksen arvot ja periaatteet. [2, 3] Julistuksessa määritellyt arvot ovat:

- Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja
- Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota
- Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja
- Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa [3]

Ketterän ohjelmistokehityksen menetelmissä uskotaan, että ihmisiin keskittynyt menetelmä tuottaisi parempia tuloksia [36]. Ketterät periaatteet pyrkivät valmiiseen tuotteeseen nopeasti käyttämättä paljon aikaa vaatimusten määrittelyyn. [31, 36]

Ketterä kehitys keskittyy itseohjautuviin ja monitaitoisiin tiimeihin, jotka tekevät projektikohtaisia päätöksiä itsenäisesti isojen hierarkioiden sijaan, sekä nopeisiin iteraatioihin, joissa kehittäjät työskentelevät yhdessä asiakkaan tai muiden sidosryhmien kanssa. [2, 31, 36] Asiakkaat vastaavat lopputuotteen käyttäjien edustamisesta. Toteutettavista ominaisuuksista päättävät sekä asiakas että kehitystiimi. [31]

Tässä luvussa esitellään yksi yleisimmistä ketterän ohjelmistokehityksen menetelmistä [36], Scrum, esimerkkinä ketteriin menetelmiin. Seuraavaksi käydään läpi Scrumin määritelmä sekä siihen liittyviä periaatteita.

2.1 Scrum lyhyesti

Scrum on ketterässä ohjelmistokehityksessä maailmanlaajuisesti käytetty viitekehys, joka on suunniteltu monimutkaisten tuotteiden kehittämiseen, jakeluun ja ylläpitoon. Tarkoin määritellyn menetelmän sijasta Scrumin ajatellaan olevan viitekehys, jossa voidaan ratkaista monimutkaisia ongelmia kehitettäessä tuotetta tuottavasti ja luovasti mahdollisimman korkealla lisäarvolla. [33]

Scrum pyrkii tuomaan tuotehallinnan sekä -kehityksen menetelmät ja niiden vaikutukset näkyväksi, jotta tuotetta, tiimiä sekä työympäristöä voidaan jatkuvasti parantaa. Scrumin säännöt määrittelevät roolit, tapahtumat, tuotokset sekä ohjaavat näiden välistä vuorovaikutusta. Scrumin etuina ovat keveys ja yksinkertaisuus, mutta haasteena sen hallitseminen. [33]

2.2 Taustateoria ja arvot

Scrum käyttää toistavaa ja lisäävää eli iteratiivis-inkrementaalista lähestymistapaa, jolla optimoidaan ennustettavuutta ja kontrolloidaan riskejä prosessissa. Scrum perustuu empiiriseen prosessin hallintateoriaan, jossa tieto perustuu kokemukseen ja päätösten tekemiseen tunnettujen tosiasioiden pohjalta. [33]

Prosessin kolme periaatetta ovat läpinäkyvyys, tarkastelu sekä sopeuttaminen. Läpinäkyvyydellä tarkoitetaan esimerkiksi prosessiin liittyvien tekijöiden sopimista ja määrittämistä yhdessä tiiminä, jotta jokaisella projektissa mukana olevalla jäsenellä on prosessiin liittyvistä tekijöistä sekä ”valmiin” määritelmästä samanlainen käsitys. Tarkastelulla viitataan työn edistymisen seurantaan, jotta haitalliset poikkeamat voidaan havaita prosessissa aikaisessa vaiheessa. Mikäli poikkeamia havaitaan, toteutetaan tällöin sopeuttaminen säätämällä prosessia ja käytettäviä materiaaleja mahdollisimman nopeasti myöhempien poikkeamien minimoimiseksi. [33]

Jotta periaatteet onnistuisivat, tulisi jokaisen projektissa mukana olevan jäsenen toimia Scrumin arvojen mukaisesti. Näitä arvoja ovat sitoutuminen, rohkeus, fokus, avoimuus sekä kunnioitus. Näiden arvojen kautta luvataan Scrumin onnistuminen ja tavoitteiden saavuttaminen. [33]

2.3 Roolit ja tehtävät

Scrumin ydin on pieni, itsenäinen scrumtiimi, joka on itseohjautuva ja monitaitoinen. Tiimi päättää itse, kuinka työ tehdään, ja sillä on kaikki työn tekemiseen vaadittava osaaminen. [33, 36] Ulkopuoliset eivät osallistu tai ohjaa tiimin työskentelyä [33].

Scrumtiimi koostuu tuoteomistajasta (engl. product owner), kehitystiimistä (engl. development team) sekä scrummasterista. Tuoteomistaja on yksi henkilö, joka on vastuussa tuotteen ja kehitystiimin työn arvon eli ROI:n (engl. Return on Investment) [36] maksimoinnista. Yksi tuoteomistajan tärkeimmistä tehtävistä on priorisoida ja päivittää tuotteen kehitysjonon (engl. backlog) sekä varmistaa kehitysjonon avoimuus, läpinäkyvyys ja ymmärrettävyys, jotta siitä selviää scrumtiimin seuraavat tehtävät [33, 36].

Kehitystiimin tehtävänä on kehittää toimivia, julkaisukelpoisia tuoteversioita (engl. increment) kohti tuotteen ”valmiin” määritelmää. Scrumtiimin tavoin, kehitystiimi on itseohjautuva ja monitaitoinen. Vaikka kehitystiimistä löytyy erilaista osaamista, kehitystiimissä ei ole perinteisiä ohjelmistokehitysrooleja, kuten arkkitehtiä tai testaajaa, vaan tehtäviä suoritetaan tiimin sopimalla tavalla tiimin jäsenten osaamisesta ja kiinnostuksesta riippuen. [33] Kehitystiimissä voi kuitenkin olla mukana esimerkiksi kehittäjiä, testaajia, suunnittelijoita, arkkitehtejä ja dokumentoijia [36].

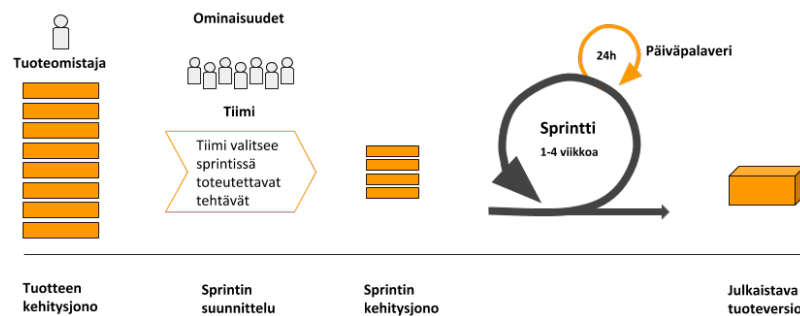
Kehitystiimin koko vaihtelee, mutta ideaalitalanteessa kehitystiimin koko on kolmesta yhdeksään jäsentä, jotta tiimiä on helppo hallita ja tuottavuus säilyy hyvänä. Mikäli kehitys-

tiimissä on vähemmän kuin kolme jäsentä, tuottavuus sekä jäsenten välinen vuorovaikutus vähenee. Tiimiltä saattaa puuttua tarvittavaa osaamista, jolloin julkaisukelpoista tuoteversiota ei välttämättä pystytä tuottamaan. Jos taas tiimissä on enemmän kuin 9 ihmistä, tiimiä voi olla hankala hallita sen suuren koon ja monimutkaisuuden vuoksi. [33]

Scrummaster on scrumtiimiä palveleva johtaja. Scrummasterin tulee olla eri henkilö kuin tuotteenomistaja [36]. Scrummasterin vastuulla on scrumin edistäminen ja tukeminen, jotta kaikki ymmärtävät scrumin teorian, toimintatavat, säännöt ja arvot. Scrummaster palvelee työssään kolmea eri tahoa: tuoteomistajaa, kehitystiimiä sekä organisaatiota. Scrummaster auttaa tuoteomistajaa tehostamaan kehitysjonon hallintaa ja ymmärtämistä sekä varmistaa, että kaikki scrumtiimissä ymmärtävät, mitä kohti prosessissa ollaan menossa. Scrummaster valmentaa kehitystiimiä itseohjautuvuudessa ja moniosaamisessa, kehityksen maksimoimisessa sekä poistaa esteitä, jotka vaikeuttaisivat kehitystiimin etenemistä. Scrummaster tukee organisaatiota scrumin johtamisessa, suunnittelussa, organisoinnissa ja ymmärtämisessä tuottavuuden kasvattamiseksi. [33, 36]

2.4 Prosessi ja tapahtumat

Scrum käyttää tiettyjä ennalta määriteltyjä, säännöllisesti toistuvia tapahtumia, joilla vähennetään tarvetta muille, scrumin ulkopuolisille palavereille. Scrumin tapahtumien tarkoitus on lisätä läpinäkyvyyttä ja antaa mahdollisuus tarkasteluille. [33]



Kuva 2. Scrum-prosessi perustuen lähteisiin [36, 33]

Scrumin päätapahtuma on sprintti (engl. sprint). Sprintiksi kutsutaan enintään kuukauden kestävästä ajanjaksoa, jonka aikana tuotetaan ”valmiin” määritelmää vastaava käyttökelpoinen ja mahdollisesti julkaisukelpoinen tuoteversio. Sprinttien kesto määritellään projektin aluksi ja niiden pituus on koko projektin ajan sama. Sprinttejä voi olla useampia, ja seuraava sprintti alkaa välittömästi edellisen päätyttyä. Jokaisella sprintillä on ennalta määritellyt vaatimukset ja tavoitteet, joita kohti sprintissä työskennellään. Näihin vaatimuksiin tai tavoitteisiin ei tulisi tehdä muutoksia enää sprintin aikana. [33]

Sprintti koostuu sprintin suunnittelusta, päivittäisistä scrumeista (engl. daily scrum) eli päiväpalavereista, kehitystyöstä, sprintin katselmoinnista sekä sprintin retrospektriivistä. Sprintin suunnittelussa määritellään sprintin tavoite, vaatimukset, valitaan tuotteen kehitysjonosta tehtävät, joita sprintin aikana suoritetaan, sekä sovitaan, kuinka nämä tehtävät suoritetaan. Sprintin suunnittelun tuloksena on sprintin kehitysjono. Suunnittelussa tärkeää on huomioida, kuinka monta tehtävää sprintin aikana ehditään tehdä niin, että päästään sprintin tavoitteeseen ja saadaan tuotettua toimiva tuoteversio. Sprinttiin valituista tehtävistä ja tehtävien suoritustavasta vastaa kehitystiimi. Sprintin suunnittelu voi kestää enintään kahdeksan tuntia. [33, 36]

Päiväpalaveri on lyhyt, enintään 15 minuuttia kestävä tapaaminen kehitystiimin kesken. Päiväpalaveri pidetään sprintin jokaisena päivänä samaan aikaan ja sen aikana käydään läpi jokaisen kehitystiimin jäsen kertoo lyhyesti, mitä on tehnyt edellisen päiväpalaverin jälkeen, mitä tulee tekemään ennen seuraavaa päiväpalaveria ja onko tekemiselle esteitä. Päiväpalaverin tarkoitus on parantaa kommunikaatiota kehitystiimin jäsenten välillä, ratkaista syntyneitä ongelmia ajoissa, tehostaa päätöksentekoa sekä lisätä kehitystiimin asiantuntemusta. [33, 36]

Sprintin katselmointi on sprintin lopussa pidettävä palaveri, jossa suoritetaan sprintin aikana kehitetyn tuoteversion tarkastelu ja tarvittaessa sopeutetaan tuotteen kehitysjonoa [33, 36], mikäli siihen tarvitaan muutoksia. Sprintin katselmoinnissa ovat mukana scrumitiimi sekä sidosryhmien edustajat [33], joita ovat kaikki ne, jotka vaikuttavat projektiin tai joihin projekti vaikuttaa [31]. Sprintin katselmointi pyrkii palautteen keräämiseen ja kommunikaation kasvattamiseen [33].

Sprintin retrospektriivi on luotu työtapojen tarkastelua varten. Retrospektriivi on palaveri, joka antaa scrumitiimille mahdollisuuden omien toimintatapojensa arviointiin ja luomaan uusia, parempia toimintatapoja seuraavaan sprinttiin. Retrospektriivi sijoittuu katselmoinnin ja seuraavan sprintin suunnittelun väliin, ja sen kesto on noin kolme tuntia. Toisin kuin muut tapahtumat, retrospektriivi keskittyy tehtävien sijaan ihmisiin, suhteisiin, prosessiin sekä prosessia käytettyihin työkaluihin. Sen tarkoitus on parantaa prosessia, eli tunnistaa toimivia ja toimimattomia toimintatapoja. [33, 36] Scrummaster ohjaa retrospektriiviä siten, että se pysyy positiivisena ja tuottavana. [33]

3. KÄYTTÄJÄKESKEINEN SUUNNITTELU OHJELMISTOKEHITYKSESSÄ

Tässä luvussa kerrotaan tiivistetysti käyttäjakeskeisen suunnittelun periaatteita sekä kerrotaan lyhyesti menetelmistä, joilla käyttäjakeskeistä suunnittelua toteutetaan ketterässä kehityksessä. Lisäksi luvussa esitellään aiempia tutkimuksia käyttäjakeskeisen suunnittelun integroimisesta ketterään kehitykseen. Luvussa kuvataan menetelmiä ja tutkimuksia siltä osin kuin ne liittyvät tutkittavaan aiheeseen, jotta saadaan selkeä yleiskäsitys käyttäjakeskeisestä suunnittelusta ohjelmistokehityksiprojekteissa.

Koska suoraan tämän diplomityön aiheesta ei löytynyt aiempia tutkimuksia, tästä luvusta löytyviä tutkimuksia voidaan pitää tärkeänä pohjana tämän diplomityön tutkimukselle. Tässä luvussa pyritään tuomaan esiin käyttäjäkokeussuunnittelijoiden ja kehittäjien perinteinen yhteistyötapa, johon tässä diplomityössä tehtyä tutkimusta voidaan myöhemmin verrata.

3.1 Käyttäjakeskeinen suunnittelu lyhyesti

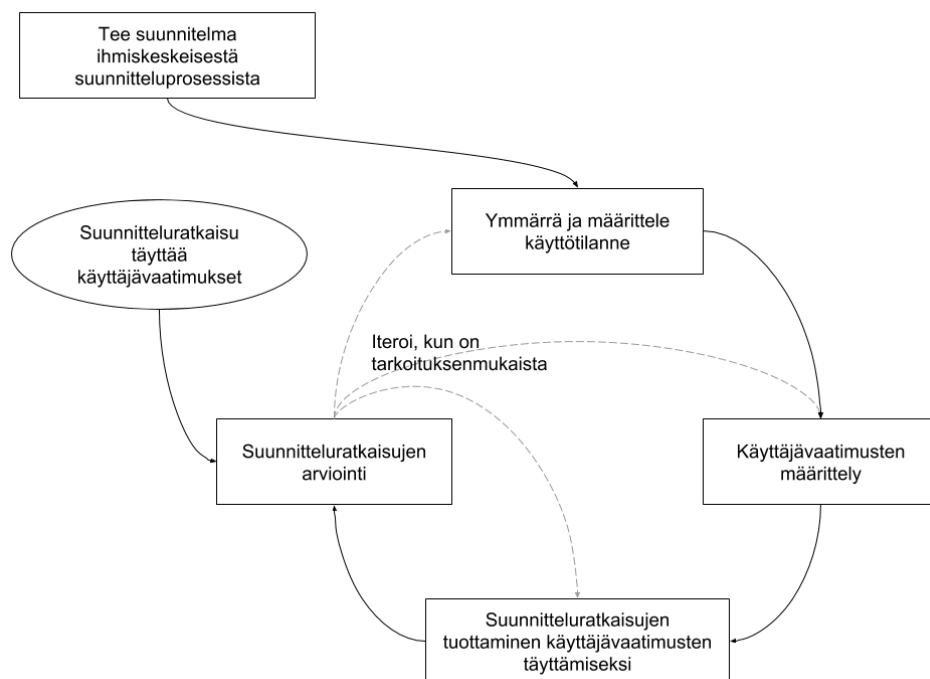
Käyttäjakeskeinen suunnittelu tai ihmiskeskeinen suunnittelu on järjestelmäsuunnittelun ja -kehityksen lähestymistapa. Käyttäjakeskeisen suunnittelun tavoitteena on parantaa järjestelmien käytettävyyttä. Standardin ISO 9241 osa 210 antaa kehikon käyttäjakeskeiselle suunnittelulle. Tämän standardin mukaan käyttäjakeskeisen suunnittelun tulisi noudattaa seuraavia periaatteita:

- suunnittelu perustuu käyttäjien, tehtävien ja ympäristöjen selkeään ymmärtämiseen
- käyttäjät ovat mukana koko suunnittelun ja kehityksen ajan
- käyttäjakeskeinen arviointi ohjaa ja tarkentaa suunnittelua
- prosessi on iteratiivinen
- suunnittelu kohdistuu käyttäjäkokeemukseen kokonaisuutena
- suunnittelutiimillä on moninaisia taitoja ja näkökulmia. [35]

Käyttäjien, tehtävien ja ympäristöjen selkeällä ymmärtämisellä tarkoitetaan esimerkiksi sitä, että tuotteiden, järjestelmien sekä palveluiden kaikki oleelliset käyttäjä- ja sidosryhmät olisi tunnistettava. Käyttäjakeskeisellä arvioinnilla painotetaan sitä, että suunnitteluratkaisut tulisi arvioida käyttäjillä ja niitä tulisi parantaa käyttäjiltä saadun palautteen perusteella. Tällä tavoin minimoidaan riskiä siitä, että järjestelmä ei täytä käyttäjän tai organisaation tarpeita. Iteratiivisella prosessilla tarkoitetaan vaihejaksojen toistamista kunnes haluttu lopputulos saavutetaan. Iterointia käytetään vuorovaikutteisten järjestelmien

kehityksen aikana epävarmuuksien eliminointiin. Lisäksi käyttäjakeskeisten suunnittelu- tiimien monialaisuus auttaa tulemaan tietoisiksi eri alojen rajoituksista ja realiteeteista, kuten teknisistä rajoitteista. [35]

Käyttäjakeskeistä suunnittelua varten tulee tehdä suunnitelma, joka sisältää menetelmien ja resurssien tunnistamisen, menettelytapojen määrittelyn, suunnittelutoiminnasta vastuussa olevien henkilöiden tai organisaatioiden ja niiden osaamisalueiden tunnistamisen, menettelytavat esimerkiksi kommunikointiin, välitavoitteiden sopimisen sekä aikamääreiden sopimisen esimerkiksi iteroinnin sisällyttämiseksi projekti aikatauluun. Käyttäjakeskeistä suunnittelua koskevan suunnitelman tulisi sisältyä kaikkiin elinkaaren vaiheisiin eli luonnosvaiheeseen, analyysiin, suunnitteluun, toteutukseen, testaukseen ja ylläpitoon. Sen on oltava osa koko järjestelmän projektisuunnitelmaa. Projektin suunnittelussa tulisi tällöin varata aikaa sekä resursseja käyttäjakeskeisen suunnittelun aktiviteeteille. [35]



Kuva 3. Käyttäjakeskeinen suunnitteluprosessi perustuen lähteeseen [35]

Käyttäjakeskeisen suunnittelun prosessi aktiviteetteineen on kuvattu kuvassa 3. Suunnitteluprosessi koostuu neljästä aktiviteetista: käyttötilanteen ymmärtäminen ja määrittely, käyttäjävaatimusten määrittely, suunnitteluratkaisujen tuottaminen sekä suunnitteluratkaisujen arviointi. Suunnitelman tekemisen jälkeen, käyttötilanteen ymmärtämisen ja määrittelyn vaiheessa pyritään ymmärtämään ja määrittämään järjestelmän käyttötilanne. Käyttötilanteeseen vaikuttavat käyttäjien ominaisuudet, tehtävät sekä organisatorinen ja fyysinen ympäristö. Käyttäjävaatimusten määrittely -vaiheessa tunnistetaan käyttäjätarpeet sekä määritellään tuotteen tai järjestelmän toiminnalliset ja muut vaatimukset. Lisäksi tässä vaiheessa määritellään vaiheen nimen mukaisesti käyttäjävaatimukset suhteessa käyttötilanteeseen ja järjestelmän liiketoiminnan tavoitteisiin. [35]

Suunnitteluratkaisujen tuottamisen vaihe jakaantuu neljään ala-aktiiviteettiin: käyttäjätehtävien, käyttäjän ja järjestelmän välisen vuorovaikutuksen (myöhemmin vuorovaikutussuunnittelu) ja käyttöliittymän suunnittelu, suunnitteluratkaisujen konkretisointi esimerkiksi prototypoimalla, suunnitteluratkaisujen muuttaminen arvioinnin ja palautteen pohjalta ja suunnitteluratkaisujen välittäminen toteutuksesta vastaaville henkilöille. Vuorovaikutussuunnittelussa pyritään tekemään päätöksiä siitä, miten käyttäjät saavat suoriteksi tehtäviä järjestelmän avulla. Suunnittelun aktiiviteetit eivät ole tiukka lineaarinen prosessi, mutta aktiiviteetit käyttävät toisen aktiiviteetin tuotoksia. [35]

Käyttäjäkokeussuunnittelu (UX) noudattaa käyttäjakeskeisen suunnittelun prosessia [6]. Käyttäjäkokeussuunnitteluun sisältyy käyttäjäpolkujen luominen tai vuorovaikutussuunnittelu, näkymien luominen ja käyttöliittymän (engl. User Interface, UI) komponenttien valitseminen [22].

Käyttäjäkokeuksella tarkoitetaan henkilön havaintoja ja vasteita, jotka ovat seurausta tuotteen, järjestelmän tai palvelun käytöstä ja/tai ennakoitusta käytöstä. Se sisältää esimerkiksi kaikki käyttäjän tunteet ja uskomukset, jotka ilmenevät ennen järjestelmän käyttöä, järjestelmän käytön aikana ja järjestelmän käytön jälkeen. [35]

3.2 Suunnittelu integroituna ketteryyteen

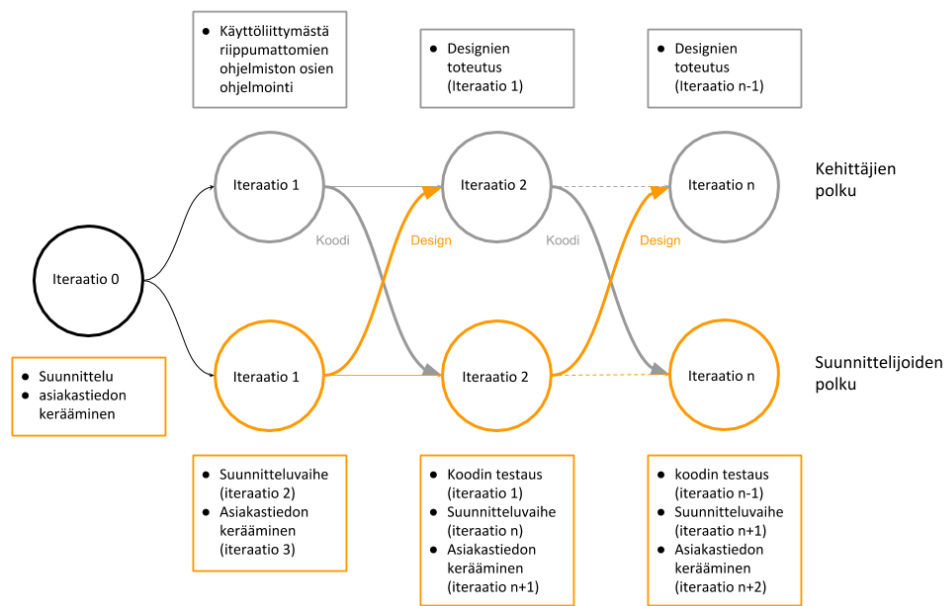
Suunnittelun integroimista ketterään ohjelmistokehitykseen on tutkittu paljon. Koska ketterät menetelmät eivät määrittele käyttäjäkokeussuunnittelijan roolia, on projekteissa usein epäselvää se, kuinka käyttäjäkokeussuunnittelija toimii kehitystiimissä. [23]

3.2.1 Etupainotteinen suunnittelu

Vuonna 2007 Sy esitteli artikkelissaan nykyisin laajalti käytössä olevan [5] mallin, jolla käyttäjakeskeinen suunnittelu integroidaan osaksi ketterää ohjelmistokehitystä. Tässä mallissa on kaksi rinnakkaista polkua (engl. dual track) kuvan 4 mukaisesti. [37]

Syn mallissa käyttäjäkokeussuunnittelua tehdään etupainotteisesti. Suunnittelu aloitetaan 1-2 iteraatiota ennen ensimmäistä toteutusiteraatiota (iteraatio 1). Tämä vaihe on nimeltään iteraatio 0 (eng. sprint 0) tai niin kutsuttu nollaiteraatio. Ennen ensimmäisen toteutusiteraation aloittamista, eli iteraatiossa 0, käyttäjäkokeussuunnittelijat pyrkivät ymmärtämään tuotteen kokonaiskuvan sekä määrittelemään sen vaatimukset ja tavoitteet. [37]

Nollaiteraation jälkeen käyttäjäkokeussuunnittelu ja kehitys jatkuu rinnakkain kahdella eri polulla: tekninen toteutus (kuvassa kehittäjien polku) ja käyttäjäkokeus (kuvassa suunnittelijoiden polku). Ensimmäisessä iteraatiossa kehittäjät ohjelmoivat sellaisia osia, jotka eivät vaikuta käyttöliittymään. [37] Tällaisia tehtäviä voivat olla esimerkiksi kehitysympäristön konfiguraatio sekä back-end -kehityksen tehtävät. Käyttäjäkokeussuunnittelijat ovat projektissa aina yhden iteraation toteutusta edellä. Tällöin käyttäjäkokeus-



Kuva 4. Syn suunnittelumalli perustuen lähteeseen [37]

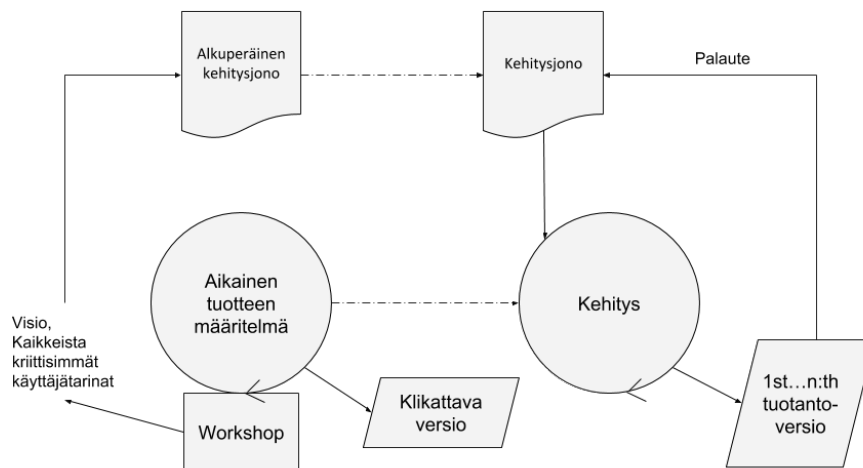
suunnittelijat työskentelevät kehittäjien seuraavan iteraation tehtävien parissa ja arvioivat edellistä. [37].

3.2.2 BoB (Best of Both Worlds) -viitekehys

Kuusinen esitteli vuonna 2015 julkaisemassaan väitöskirjassaan BoB (Best of Both Worlds) -viitekehysten käyttäjäkokemussuunnittelun integroimiseksi ketterään kehitykseen vaihtoehtoksi edellä esiteltyyn etupainotteiseen suunnittelumalliin. Kuusisen esittämä viitekehys pyrkii käyttäjäkokemussuunnitteluun ketterässä kehityksessä siten, että toteutusi-teraatioita edeltävä suunnittelu ei olisi enää tarpeellinen. [24]

BoB-viitekehyksessä kehittäjät ja käyttäjäkokemussuunnittelijat muodostavat monitieteellisen tiimin, joka saman iteraation aikana työskentelee yhdessä tiettyjen tehtävien parissa yhdistäen käyttäjäkokemussuunnittelun ja ohjelmistokehityksen. [24]

Käyttäjäkokemussuunnittelijoiden ja kehittäjien välinen yhteistyö voidaan aloittaa muutamilla lyhyillä työpajoilla, joissa käsitellään esimerkiksi käyttäjätarinoita ja luodaan tuotteen kehitysjono. Näiden työpajojen välillä tiimi työskentelee pienin askelin kohti kosketeltavaa ja näkyvää tuotetta. Alussa tuote voi olla pelkkä prototyyppi tai läpiklikkailtava malli. Mikäli tiimi ei pysty ratkaisemaan jotain käyttäjäkokemussuunnitteluongelmaa, kehittäjät siirtyvät kehitysjonon seuraavaan tehtävään, ja käyttäjäkokemussuunnittelijat pyrkivät ratkaisemaan ongelman. Näin jatketaan, kunnes ongelma on saatu ratkaistua tai



Kuva 5. BoB-viitekehitys perustuen lähteeseen [24]

tehtävän suoritusta on siirretty myöhemmäksi. Käyttäjäkokeussuunnittelija joko itse toteuttaa käyttöliittymän tai työskentelee yhdessä front-end -kehittäjän kanssa. [24]

Kuusisen esittelemän BoB-viitekehityksen etuja ovat esimerkiksi se, että kehitystyö voidaan aloittaa heti, kun tiimillä on kehitettävästä tuotteesta käsitys. Täydellisiä, kokonaisia käyttäjäkokeussuunnitelmia ei tuoteta, vaan mallin myötä syntyvä tekninen velka mahdollistaa käyttäjäkokeussuunnittelun tekemisen pienissä osissa. [24]

Kuusinen toteaa väitöskirjassaan BoB-viitekehityksen haasteina olevan esimerkiksi se, että se soveltuu parhaiten pieniin, samassa tilassa työskenteleviin tiimeihin ja viitekehityksen käyttäminen vaatii, että käyttäjäkokeussuunnittelija työskentelee kehitystiimissä. Suuremmissa projekteissa, joissa työskentelee useampi tiimi, voi olla, että jokaisessa tiimissä tulisi olla oma käyttäjäkokeussuunnittelija. [24]

Kuusinen totesi tutkimuksessaan, että käyttäjäkokeussuunnittelijan roolia kaivataan osana kehitystä, jotta voidaan varmistaa haluttu käyttäjäkokeus. Ketterän kehityksen metodi ei kuitenkaan tarjoa ohjeita sille, kuinka käyttäjäkokeussuunnittelu tuodaan osaksi projektia tai kuinka yhteistyö käyttäjäkokeussuunnittelijoiden ja ohjelmistokehittäjien välillä toteutetaan. [24]

3.2.3 Kehittäjät ja suunnittelu

Käyttäjäkokeussuunnittelua on yritetty integroida ketterään kehitykseen paljon myös siten, että pyritään opettamaan ohjelmistokehittäjille käyttäjäkokeussuunnittelutaitoja. Esimerkiksi Øvad et al. tekivät tutkimuksen siitä, kuinka yritykset voivat toteuttaa käyttäjäkokeussuunnittelun tehtäviä, kun organisaatiossa ei ole käyttäjäkokeussuunnitte-

lijoita tai niitä on vain muutama. He tutkivat, kuinka hyvin ohjelmistokehittäjät suoriutuivat käyttäjäkokemussuunnittelun tehtävistä järjestämällä ohjelmistokehittäjille koulutus-tilaisuuksia käyttäjäkokemussuunnittelun tehtävistä. Tilaisuuksissa mitattiin ohjelmistokehittäjien suoritusta heidän oppiessaan ja soveltaessaan oppimaansa itsenäisesti ja heille tarjottiin kirjallisia materiaaleja, kuten suosituksia hyvistä toimintatavoista, malleja, ja "lunttilappuja". [25]

Tutkimus osoitti, että kehittäjät pystyvät oppimaan taidot ja keinot tiettyjen käyttäjäkokemussuunnittelun tehtävien suorittamiseksi sprintin aikana, erityisesti etukäteen valmisteltujen mallien avulla. Øvad et al. kuitenkin huomauttavat, ettei tapa sovellu ainoaksi ratkaisuksi, eikä siksi korvaa käyttäjäkokemussuunnittelijaa ohjelmistoprojektissa. Ohjelmistokehittäjien oli hankala analysoida kvalitatiivista dataa, jolloin laajat tutkimukselliset menetelmät, kuten etnografiset tutkimukset, ovat poissuljettuja tehtäviä. [25] Myös Boivie et al. kokemusten mukaan kehittäjä, joka on kiinnostunut käytettävyydestä, ei kuitenkaan pysty tuottamaan käyttäjien tarpeisiin ja käytettävyysoongelmiin vastaavaa ratkaisua ohjelmistokehitysprojektissa [4]. Ketterä ohjelmistokehitys vaatii nopeaa ja epämuodollista arviointia [25]. Kuusisen tutkimus osoittaa, että kehittäjät ymmärtävät kehitettävän ohjelmiston käyttäjäkokemuksen käytännön puolta, mutta he korostavat käytännön puolta liikaa hedonisen puolen kustannuksella. Käyttäjäkokemussuunnittelijan kannattaa osallistua erityisesti hedonisen puolen ymmärtämiseen ja suunnitteluun ohjelmistoprojektissa. [22]

3.3 Käyttäjäkokemussuunnittelijan rooli ketterässä ohjelmistokehityksessä

Käyttäjäkokemussuunnittelijan erilaisia rooleja ohjelmistokehitysprojekteissa on myös tutkittu yrittäen ymmärtää käyttäjäkokemussuunnittelijan työtä projekteissa sekä selvittää, mitä tehtäviä käyttäjäkokemussuunnittelijan erilaisiin rooleihin kuuluu.

Vuonna 2008 Fox et al. määrittelivät kolme yleisintä lähestymistapaa käyttäjäkeskeisen suunnittelun (engl. user-centered design) toteuttamiseen: spesialisti (engl. the specialist), generalisti (engl. the generalist) ja generalisti-spezialisti (engl. the generalist-specialist). Spesialistin lähestymistavassa on kolme erilaista ryhmää: käyttäjät/asiakkaat, käyttäjäkokemussuunnittelijat sekä kehittäjät. Tiimissä on useampia kehittäjiä ja yksi käyttäjäkokemussuunnittelija, joka vastaa kaikesta käyttäjäkokemussuunnitteluun liittyvästä projektissa. Generalistin lähestymistavassa on kaksi pääroolia: käyttäjät/asiakkaat sekä kehittäjät, jotka työskentelevät myös käyttäjäkokemussuunnittelijan roolissa. Kehittäjät eivät ole virallisesti kouluttautuneet käyttäjäkeskeiseen suunnitteluun vaan ovat itseoppineita. Spesialistin tavasta poiketen, tiimissä oli siis useampi jäsen, joka suoritti käyttäjäkokemussuunnittelun tehtäviä. Kolmas lähestymistapa, generalisti-spezialisti, on kahden edellä mainitun lähestymistavan yhdistelmä, risteymä (engl. hybrid). Generalisti-spezialistilähestymistavassa tiimissä on mukana jäsen, jolla on sekä virallinen koulutus käyttäjäkeskeiseen suunnitteluun että ohjelmistokehityskokemusta. Tämä jäsen toimii sekä generalistina että specialistina tiimissä. Tiimissä toimi lisäksi useampia spesialisteja. Ero-

na aikaisempiin lähestymistapoihin on se, että generalisti-spesialisti ei varsinaisesti toimi kummassakaan roolissa vaan toimii eri tiimin jäsenen (kehittäjän, käyttäjäkokemussuunnittelijoiden ja asiakkaiden) yhteyshenkilönä. [14]

Silva et al. ovat tutkineet käyttäjäkokemussuunnittelijan rooleja ketterissä ohjelmistokehitysprojekteissa tarkoituksenaan tarjota parempi ymmärrys käyttäjäkokemussuunnittelijan rooliin kuuluvista tehtävistä. He mainitsevat tutkimuksessaan, että käyttäjäkokemussuunnittelun integroimisesta ketterään kehitykseen on tutkittu, mutta ne eivät kerro tarkemmin käyttäjäkokemussuunnittelijan roolista projekteissa. Silva et al. myös huomioivat, että rooli voi muuttua projektin aikana, ja rooli voi riippua paljon kontekstista. [34]

Taulukko 1. Käyttäjäkokemussuunnittelijan rooleja ketterässä ohjelmistokehityksessä [34]

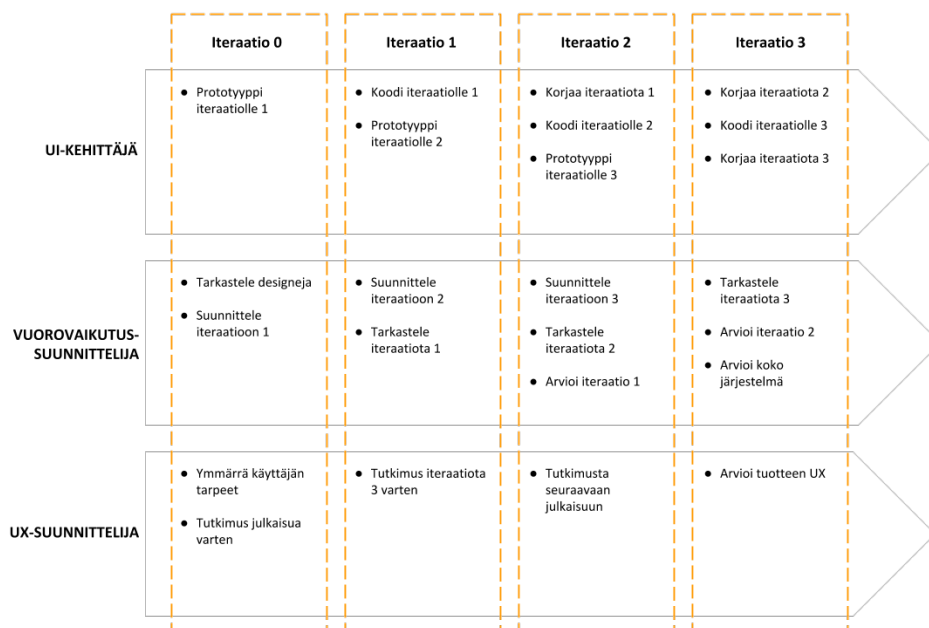
Rooli	Vastuu	Taidot
Käyttäjäkokemussuunnittelija	Käyttäjien ymmärtäminen	Käyttäjätutkimus, etnografia eli kansantiede, käyttäjäkokemuksen suunnittelu, käyttäjäprofilointi, ideointi, kilpailija-analyysi, suunnitteluajattelu, asiakaspolun suunnittelu (engl. customer journey mapping)
Vuorovaikutussuunnittelija	Käyttäjän vuorovaikutuksen tuotteiden ja palveluiden kanssa suunnittelu ja arviointi, sekä prototyypeissä että kehitetyssä järjestelmässä	Vuorovaikutussuunnittelu, nopea prototyyppiointi (engl. rapid prototyping), käyttäjäkokemussuunnittelu, tuotesuunnittelu, testaus, luonnostelu (engl. sketching), käytettävyydestaus, ideointi, yhteissuunnittelu (engl. collaborative design), prosessin kulut, tietoarkkitehtuuri, palvelumuotoilu (engl. service design), suunnitteluajattelu
Käyttöliittymäkehittäjä (UI-kehittäjä)	Graafisen käyttöliittymän (engl. Graphical User Interface, GUI) kehitys ja graafisten elementtien suunnittelu	Nopea prototyyppiointi, yhteissuunnittelu, tietoarkkitehtuuri, visuaalinen suunnittelu, GUI-suunnittelu, palvelumuotoilu, suunnitteluajattelu

Tutkimuksessaan Silva et al. esittelevät kolme erilaista yksinkertaistettua käyttäjäkokemussuunnittelijan roolia: käyttäjäkokemussuunnittelija (engl. user experience designer), vuorovaikutussuunnittelija (engl. interaction designer) sekä UI-kehittäjä (engl. UI developer eli käyttöliittymäkehittäjä) kuten ne ovat esitetty tehtävineen ja taitoineen taulukossa 1. Käyttäjäkokemussuunnittelijan vastuulla on käyttäjien ymmärtäminen, jolloin erityi-

sesti käyttäjätutkimuksen tekeminen on tärkeässä osassa käyttäjäkokemussuunnittelijan työtä. Käyttäjäkokemussuunnittelijan tulisi työskennellä yhdessä liiketoiminta-analyytikon kanssa (engl. business analyst) vision ja suunnan luomiseksi. [34]

Vuorovaikutussuunnittelijan vastuulla ovat vuorovaikutuksen suunnittelu ja arviointi. Käyttäjäkokemussuunnittelija on voinut ylläpitää käyttäjäkokemussuunnittelun kehitysjonoa, jota vuorovaikutussuunnittelija voi käyttää näkymien suunnitteluun ja prototypointiin. Vuorovaikutussuunnittelija voi myös työskennellä kehittäjien kanssa selvittääkseen, kuinka käyttöliittymäsuunnitelmat voidaan toteuttaa. [34]

UI-kehittäjä on vastuussa graafisesta käyttöliittymästä ja siihen liittyvistä elementeistä. UI-kehittäjät toimivat usein käyttäjäkokemussuunnittelijoiden ja front-end -kehittäjien välissä, sillä he ymmärtävät sekä suunnittelu- että kehitysprosessia. Vuorovaikutussuunnittelijoiden tapaan, UI-kehittäjät työskentelevät kehittäjien kanssa saadakseen selville, miten käyttöliittymä voidaan toteuttaa. Silva et al. havaitsivat, että UI-kehittäjän rooli oli muita rooleja harvinaisempi, sillä useimmat käyttäjäkokemussuunnittelijat eivät kehittäneen suunnittelemaa käyttöliittymää. Kaikille suunnittelijoille käyttöliittymien toteuttaminen ei ole yksinkertaista. Silva et al. kuitenkin tutkimuksessaan löysivät myös käyttäjäkokemussuunnittelijoita, jotka toteuttivat prototyyppjä käyttäen HTML-merkkausta. [34]



Kuva 6. Esimerkki käyttäjäkokemussuunnittelijan roolien tehtävien jakautumisesta eri iteraatioissa perustuen lähteeseen [34]

Silva et al. mukaan käyttäjäkokemussuunnittelijan suuren työmäärän vuoksi on tärkeää, että käyttäjäkokemussuunnittelija on kehitystiimin jäsen. Esimerkki siitä, kuinka käyttäjäkokemussuunnittelijan eri roolit, roolien tehtävät ja niiden vaiheet voitaisiin toteuttaa ketterissä ohjelmistoprojekteissa, on esitetty kuvassa 6. Käyttäjäkokemussuunnittelijan ei

myöskään tulisi työskennellä samaan aikaan liian monessa projektissa, mikä tosin usein on kiinni yrityksen – ei käyttäjäkokemussuunnittelijan – tekemistä valinnoista. [34]

Ratcliffe ja McNeill määrittelevät useamman erilaisen roolin, joissa käyttäjäkokemussuunnittelijat voivat työskennellä: käyttöliittymä-, vuorovaikutus- ja käytettävyyssuunnittelija, kokemussuunnittelija (engl. experience designer), UI-kehittäjä ja front-end -kehittäjä, tietoarkkitehti tai tutkija [30]. Silva et al. kritisoivat tämän roolijaon olevan liian yksityiskohtainen eikä välttämättä vastaa todellisuutta projekteissa. Käyttäjäkokemussuunnittelija on yleensä yksi ihminen, joka toimii useammassa roolissa. [34]

Myös Ibragimova et al. toteuttivat tutkimuksen käyttäjäkokemussuunnittelijoiden rooleista ketterissä ohjelmistokehitysprojekteissa seuraamalla kymmentä eri kokoista ohjelmistoprojektia eri kokoisissa yrityksissä. Ibragimova et al. tutkimuksessaan määrittelevät näiden eri projektien käyttäjäkokemussuunnittelijoiden tehtäviin perustuen yleisimmät käyttäjäkokemussuunnittelijoiden profiilit, jotka saattoivat koostua yhdestä tai useammasta roolista, kuten on esitetty taulukossa 2. [21]

Taulukko 2. Käyttäjäkokemussuunnittelijan profiilit [21]

Rooli	Vastuualue	Esiintyvyys
Käyttöliittymäsuunnittelija (UI)	Näkymien luominen ja linjausten tekeminen projektin tyyliohjeiden ja periaatteiden mukaisesti	0/10
Käyttäjäkokemussuunnittelija (UX)	Rautalankamallien ja siirtymien luominen käyttäjätarpeisiin perustuen	2/10
Käyttäjätutkija (UR, engl. User Researcher)	Ongelman ja käyttäjän tarpeiden kartoitus sekä rautalankamallien tai näkymien validointi loppukäyttäjien kanssa	0/10
UI/UX	Vastaa molempien UI- ja UX-suunnittelijoiden rooleista	2/10
UX/UR	Vastaa molempien UX- ja UR-suunnittelijoiden rooleista	1/10
UI/UR	Vastaa molempien UI- ja UR-suunnittelijoiden rooleista	0/10
UI/UX/UR	Monitaitoinen käyttäjäkokemussuunnittelija, joka vastaa kaikista tehtävistä koko luovan prosessin aikana	5/10

Tutkimuksessa selvisi, että UI/UX/UR-profiili (UR, engl. User Researcher eli käyttäjätut-

kija) oli mukana tiimeissä, jossa oli ainoastaan yksi käyttäjäkokemussuunnittelija. Tässä profiilissa työskentelevä käyttäjäkokemussuunnittelija oli myös tyytyväisin työhönsä verrattuna muihin profileihin. Ibragimova et al. olettavat, että mitä vähemmän projektissa on käyttäjäkokemussuunnittelijoita, sitä enemmän käyttäjäkokemussuunnittelijoilla on vastuualueita. Tästä johtuen, mitä enemmän suunnitteluprosessi on käyttäjäkokemussuunnittelijan hallinnassa ja omistuksessa, sitä tyytyväisempi käyttäjäkokemussuunnittelija on. Profiilin haittapuolena tutkijat mainitsevat profiilin mahdollisen negatiivisen vaikutuksen tuotteen laatuun, sillä yhden ihmisen on vaikeaa suorittaa tehtäviä erittäin hyvin jokaisella osa-alueella. [21]

Profiilit ja niistä saadut hyödyt riippuvat myös projektin koosta. Pienissä projekteissa koetaan etuna se, että palvelumuotoilu ja vuorovaikutussuunnittelu on erotettu toisistaan, koska tällöin käyttäjäkokemussuunnittelijat voivat keskittyä omiin vahvuuksiinsa omissa rooleissaan. Pienissä projekteissa käyttäjäkokemussuunnittelijat usein erikoistuvat visuaaliseen suunnitteluun. Kuitenkin pienissä projekteissa koettiin hyödylliseksi myös käyttäjäkokemussuunnittelijan monipuolisuus, koska tämä yksinkertaistaa prosessia ja mahdollistaa tiiviimmän kommunikaation kehittäjien kanssa. Keskikokoisissa ja suurissa projekteissa käyttäjäkokemussuunnittelijan rooli keskittyy rautalankamallien luomisesta käytettyyystestaukseen. Lisäksi suurissa projekteissa, jossa voi olla monta käyttäjäkokemussuunnittelijaa, käyttäjäkokemussuunnittelija voi keskittyä yhteen rooliin, vaikka ymmärtää myös muiden roolien toimintaa. [21] Boivie et al. mainitsevat myös tutkimuksessaan, että käyttäjäkokemussuunnittelijoilla tulisi olla jonkin verran ymmärrystä ohjelmistokehityksestä [4]. Vaikka Ibragimova et al. tutkimus ei ottanut kantaa siihen, osallistuvatko käyttäjäkokemussuunnittelijat myös käyttöliittymäsuunnitelman toteuttamiseen, mainitsevat tutkijat joillain osallistujilla olevan myös muita toissijaisia rooleja ja tehtäviä projekteissa. Näistä rooleista yksi oli esimerkiksi ohjelmointi, jotta käyttöliittymäsuunnitelman siirtämistä kehittäjille saadaan nopeutettua. [21]

3.4 Käyttäjäkokemussuunnittelijan yhteistyö eri roolien kanssa

Kuusinen tutki tehtävien jakamista käyttäjäkokemussuunnittelijoiden ja kehittäjien kesken. Kuusisen tutkimuksessa seurattiin kuutta ketterää ohjelmistokehitysprojektia viidesssä suomalaisessa yrityksessä, jotka kehittivät yritysjärjestelmiä. Kuusinen raportoi tutkimuksestaan yleisimmiksi suunnittelutehtäviksi vaatimusten selkeyttämisen, uuden ominaisuuden suunnittelun sekä käyttöliittymäsuunnitelmien luomisen ja katselmoinnin. Kehittäjät tekivät seuraavia suunnittelu tehtäviä yhdessä käyttäjäkokemussuunnittelijan kanssa: käyttäjätutkimus, keskustelu käyttöliittymäsuunnitelmista tai teknologia-asioista ja muuhun käyttäjäkokemukseen liittyvä toiminta, kuten palaveri, dokumentointi ja testitapausten luonti. Käyttäjäkokemussuunnittelijat osallistuivat vähiten arkkitehtuurisuunnitelman luomiseen, kehitysjonon päivittämiseen sekä toteutuksen katselmointiin.[23]

Kuusinen lisäksi julkaisi tutkimuksestaan seuraavat kolme yhteistyötyyppiä: minimaalinen, tuoteomistaja-käyttäjäkokemussuunnittelija sekä kehittäjät-käyttäjäkokemussuunnittelija.

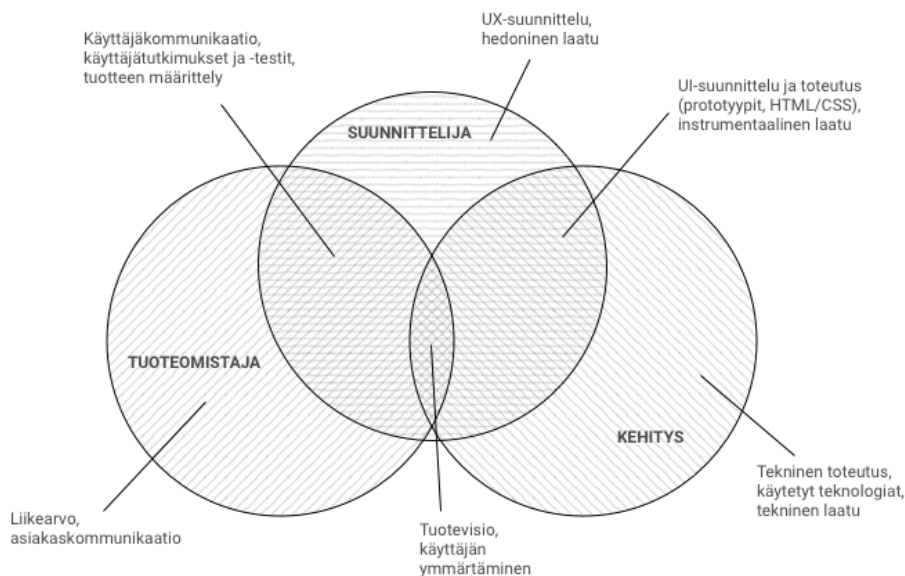
Taulukko 3. Yhteistyötyyppien hyödyt ja haasteet perustuen lähteeseen [23]

Tyyppi	Hyödyt	Haasteet
Minimaalinen	Käyttäjäkoke- muussuunnittelija voi keskittyä käyttäjäkoke- muussuunnit- telun tehtäviin ilman häiriöitä	Työn synkronointia käyt- täjäkoke- muussuunnittelijan ja kehittäjien välillä. Projektin ison kuvan ylläpitäminen. Voi johtaa epäsopivaan käyttöliitty- mäsuunnitelmaan ja tuplatyöhön. Käyttäjäko- ke- muussuunnittelun budjetin käyttäminen liian aikaisin
Tuoteomistaja-Käyttäjäkoke- muussuunnittelija	Käyttäjäkoke- muussuunnittelun asiat projektin tasolla. Helpottaa projektin ison kuvan ylläpitämistä	Korkeat yleiskustannukset erityisesti projekteissa, jota tehdään eri paikoissa
Kehittäjä-Käyttäjäkoke- muussuunnittelija	Kehittäjät tietävät, miksi käyttöliittymäsuunnitelma- päätökset ovat tehty. Keskustelu käyttöliitty- mäsuunnitelmasta on helpompaa. Edesauttaa järkevempiin kompromisseihin käyttö- liittymäsuunnitelman ja teknisten rajoitteiden välillä	Taipumus allokoida liian vähän tai ei yhtään aikaa suunnittelulle ja käyttäjätutkimuksille. Riski työn palastelulle ja käyttäjäkoke- muksen liiallinen sovittelu teknisten syiden vuoksi

Minimaalisessa yhteistyötyypissä joko yksi henkilö oli vastuussa käyttäjäkoke-
muussuunnittelun tehtävien tekemisestä ilman muiden projektin jäsenten avustusta tai useat ih-
miset suorittivat käyttäjäkoke-
muussuunnittelun tehtäviä erikseen. Tuoteomistaja-käyttäjä-
koke-
muussuunnittelija -tyypissä tuoteomistaja ja käyttäjäkoke-
muussuunnittelija tekivät käyt-
täjäkoke-
muussuunnittelun tehtäviä yhdessä. Kehittäjä-Käyttäjäkoke-
muussuunnittelija -tyypissä
kehittäjät sekä käyttäjäkoke-
muussuunnittelija tekivät suurimman osan käyttäjäkoke-
muustehtävistä yhdessä, ja tuoteomistaja on vähemmän mukana käyttäjäkoke-
muukseen liitty-
vien tehtävien tekemisessä. [23] Näiden tyyppien hyödyt ja haasteet on kuvattu taulu-
kossa 3. Kuusinen raportoikin tutkimuksessaan, että näistä kolmesta tyyppistä Kehitys-
käyttäjäkoke-
muussuunnittelija -yhteistyötyyppi oli osallistujien kesken suosituin, vaikka-
kin se vaatii sekä käyttäjäkoke-
muussuunnittelijalta että kehittäjiltä kokemusta ja halua
työskennellä yhdessä. Kun kommunikaatio käyttäjäkoke-
muussuunnittelijan ja kehittäjien
välillä ei ole sujuvaa, kehittäjien tarvitsee luoda oma ymmärrys käyttäjän tarpeista ja
käyttöliittymäsuunnitelman ratkaisusta. Heidän näkemyksensä voi kuitenkin olla käyttä-
jäkoke-
muussuunnittelijan näkemyksiin verrattuna hyvin erilainen. Käyttäjäkoke-
muussuun-
nittelijoiden ja kehittäjien tulisi työskennellä yhdessä kehitettävissä olevien käyttöliitty-
mäsuunnitelmien luomiseksi. Tämä tarkoittaa sitä, että käyttäjäkoke-
muussuunnittelijoiden
tulisi ottaa huomioon tekniset rajoitteet, ja kehittäjien tulisi oppia käyttöliittymäsuunnit-

telun periaatteita. [23]

Kuusisen mukaan yhden ihmisen ollessa vastuussa suuresta työmäärästä, työ toteutetaan osissa, mikä tarkoittaa projektissa odotteluaikaa. Hän ehdottaa, että useampi ihminen tiivistä tekisi käyttäjäkokemussuunnittelua. [24] Kuusinen esittelee tutkimukseensa [23] perustuen, kuinka tehtäviä voitaisiin jakaa eri roolien kesken (kuva 7). Kuvassa ”UX-suunnittelu” viittaa käyttäjäkokemussuunnitteluun, ”UI-suunnittelu” viittaa käyttöliittymäsuunnitteluun ja ”suunnittelija” viittaa käyttäjäkokemussuunnittelijaan.



Kuva 7. Tehtävien jako eri roolien välillä perustuen lähteisiin [23, 24]

Vuonna 2011, Esteves ja Andrade esittelivät toimintamallin, jossa käyttöliittymäsuunnitelman toteutus oli käyttäjäkokemustiimin, erityisesti visuaalisen käyttäjäkokemussuunnittelijan, vastuulla. Käyttäjäkokemussuunnittelija toteuttaa käyttöliittymäsuunnitelman prototyypin, eli sovelluksen front-end -puolen, joko projektiin valituilla teknologioilla (kuten HTML/CSS-merkkausta ja JavaScript-kieltä käyttäen) tai käyttäen jotain ulkopuolisen toteuttamaa viitekehystä tai ohjelmistoa. Tällöin front-end- ja back-end -ohjelmoinnin tehtäviä suoritetaan rinnakkain, mikä mahdollistaa sen, että suunniteltavan sovelluksen käyttöliittymä vastaa suunnitteluspesifikaatioita. [12]

Esteves et al. kannattavat tiivistä yhteistyötä käyttäjäkokemussuunnittelijoiden ja kehittäjien välillä toteutusprosessissa. Verrattuna perinteiseen ohjelmistokehitysmalliin, jossa käyttäjäkokemussuunnittelijat antavat kehittäjille spesifikaatiot ja myöhemmin esittävät korjauksia kehitettyyn koodiin, Esteves et al. esittelemässä toimintamallissa käyttäjäkokemussuunnittelijat antavat kehittäjille selainpuolen koodia ja ovat osa kehitystiimiä. Kun käyttäjäkokemussuunnittelijat ja kehittäjät ohjelmoivat yhdessä, toteutus vastaa luotuja käyttäjäkokemusspesifikaatioita. Yhteistyöprosessia voidaan parantaa, jos kehittäjä toimii ”integroiijan” roolissa ja auttaa visuaalista käyttäjäkokemussuunnittelijaa toteutuksen aikana. Integroija yhdistää käyttäjäkokemussuunnittelijan tekemän front-end -koodin

back-end -koodiin ja lisäksi auttaa käyttäjäkokemussuunnittelijaa vaikeissa käyttöliittymän toteutukseen liittyvissä tehtävissä, jotka vaativat enemmän osaamista ohjelmoinnista. [12]

Käyttöliittymäsuunnitelman toteuttaminen prototyyppiksi vaatii erilaisen mentaalisen mallin kuin käyttöliittymäsuunnitelman suunnittelu. Visuaalinen suunnittelu on käsitteellinen tehtävä, ja suunnittelun toteuttaminen on käytännöllinen tehtävä. Visuaalisesta käyttäjäkokemussuunnittelijasta tulisi sekä käyttäjäkokemussuunnittelu- että kehitystiimin jäsen, ja tämä vaatisi visuaaliselta käyttäjäkokemussuunnittelijalta myös teknistä osaamista. RIA (Rich Internet Application) viitekehysten tarjoamat kehitystyökalut auttavat käyttäjäkokemussuunnittelijaa prototyyppin luomisessa, mutta tämä saattaa saada asiakkaan luulemaan, että suunnitteluprosessi tarvitsee vähemmän työtä kuin se oikeasti tarvitsee. [12] Prosessi tosin säästää aikaa ja rahaa, sillä kehitystiimi on vapautettu käyttöliittymäsuunnitelman spesifikaatioiden kääntämisestä front-end -koodiksi ja voivat keskittyä enemmän back-end-ohjelmointiin. Prosessi näin vähentää front-end -puolen toteutuksen vaatimien kierrosten määrää. [12] Prosessi vaatii käyttäjäkokemussuunnittelijalta täysin uusien, mahdollisesti käyttäjäkokemussuunnittelijalle ei niin kiinnostavien taitojen opettelua, joko ohjelmoinnin opettelua tai siinä avustavien ohjelmistojen käytön opettelua. Lisäksi Esteves et al. lisäävät, että vaikka sopivia työkaluja on tarjolla, vuorovaikutuksen toteuttamista voi hallita kokonaan ainoastaan koodin kirjoittamisen kautta. Tästä syystä Esteves et al. esittelemä toimintamalli riippuu visuaalisen käyttäjäkokemussuunnittelijan kyvystä oppia toteutukseen tarvittavat mentaaliset mallit ja tekniset taidot. Tässä prosessissa kuitenkin visuaalista käyttäjäkokemussuunnittelijaa tukee kokeneempi kehittäjä, joka ratkaisee front-end -koodiin liittyviä ongelmia niiden ilmaantuessa. Ideaalissa tilanteessa visuaaliselle käyttäjäkokemussuunnittelijalle löytyisi työkalu, jolla konsepti voitaisiin muuttaa front-end-koodiksi. Tämä poistaisi tarpeen toteuttaa suunnittelua kehitystyökalulla tai luoda käyttöliittymäsuunnitelmaa uudelleen toiseen työkaluun. [12]

Esteves et al. esittelemä ohjelmistokehityksen ja käyttäjäkokemussuunnittelun toimintamalli mahdollistaa sujuvan yhteistyön käyttäjäkokemussuunnittelijoiden ja kehittäjien välillä. Tämä johtaa optimoituun toteutusprosessiin pienemmillä kustannuksilla ja lyhyemmällä kestolla. Tämä optimointi voi onnistua vain, jos käyttäjäkokemustiimi, ja erityisesti visuaalinen käyttäjäkokemussuunnittelija, ottaa vastuulleen sekä käyttäjäkokemussuunnittelun että käyttöliittymäsuunnitelman toteuttamisen. Jotta tämä onnistuu, visuaalisen käyttäjäkokemussuunnittelijan tulisi oppia lisäksi teknistä puolta, ylläpitää selvää näkemystä käyttöliittymästä sekä ymmärtää kehitystiimin tarpeita toteutukseen. Esteve et al. ohjelmistokehitysmalli pyrkii parantamaan käyttäjäkokemussuunnittelijan suunnitelman käyttöliittymäsuunnitelman välitystä kehittäjille. [12]

Yksi suurimmista haasteista ketterässä käyttäjäkokemussuunnittelussa on käyttäjäkokemussuunnittelijoiden ja kehittäjien työn synkronointi sekä erityisesti vaikeudet kommunikaatiossa [32]. Kuusinen huomasi toisessa tutkimuksessaan, että kommunikaatiosta tekee vaikeampaa kehittäjien ja käyttäjäkokemussuunnittelijoiden ajatusten hetkelliset eroavaisuudet [22]. Yhteistyö käyttäjäkokemussuunnittelijoiden ja kehittäjien välillä on helpom-

paa, jos tiimin jäsenet ymmärtävät toisten tiimin jäsenten työstä jonkin verran. Esimerkiksi Gulliksen et al. [17] ehdottavat, että käyttäjäkokemussuunnittelijan tulisi olla jonkin verran ymmärrystä ohjelmistokehityksestä kommunikaation parantamiseksi. [24] Kuusinen esittää, että käytettävistä teknologioista ja alustoista riippuen käyttöliittymäsuunnitelma voidaan toteuttaa ja kommunikoida toimivana ohjelmistona/prototyypinä, esimerkiksi käyttäen HTML- ja CSS-merkkausta. Jos käyttäjäkokemussuunnittelija käyttää tällaisia työvälineitä, käyttäjäkokemussuunnittelija pystyy tekemään käyttöliittymäsuunnitelman vähintään samassa ajassa tai jopa nopeammin kuin piirtäessään käyttöliittymäsuunnitelman. [22, 24]

Saatavilla olevat toteutustyökalut jonkin verran vähentävät tarvetta front-end -koodin luomiselle. Asiakkaat voivat tosin tällöin luulla, että työkalun käyttäminen vaatii vähemmän työtä käyttäjäkokemussuunnittelijalta. Käyttäjäkokemussuunnittelijoiden kannattaa opettaa asiakasta ymmärtämään, että suunnittelu ei rajoitu vain toteutustyökalujen käyttämiseen. Nämä työkalut tarjoavat hyvännäköisiä elementtejä ja toiminnallisuutta, mutta se ei takaa hyödyllistä ja käytettävää sovellusta. [12]

Kuusisen tekemässä tutkimuksessa selvisi, että käyttäjäkokemussuunnittelijoiden ja kehittäjien välisessä kommunikaatiossa oli ongelmia. Tämä saattoi aiheuttaa saman työn tekemistä kahdesti: käyttäjäkokemussuunnittelija ja kehittäjät voivat molemmat erikseen määritellä vaatimukset ja suunnitella näkymiä. Kehittäjät voivat tehdä muutoksia käyttöliittymäsuunnitelmiin eivätkä ilmoita muutoksista käyttäjäkokemussuunnittelijalle. Tämä tuotti erilaisia konsepteja ja heikensi kehitettävän tuotteen laatua. Kuusinen havaitsi tutkimuksessaan, että käyttäjäkokemussuunnittelijan kannattaa oppia myös muilta rooleilta. Hänen mukaansa kuitenkin käyttäjäkokemussuunnittelun tehtävien jakaminen useammalle ihmisille auttaa tuotteen vision ylläpitämisessä, vastualueiden sopimisessa sekä helpottaa käyttäjäkokemussuunnittelijan työtä ja mahdollistaa sen, että käyttäjäkokemussuunnittelija voi keskittyä ainoastaan varsinaiseen käyttäjäkokemussuunnittelutyöhön. Kuitenkin nopeassa mobiilikehityksessä matalatarkkuuksiset rautalankamallit voivat hidastaa kehitystä, sillä korkeatarkkuuksisia grafiikoita voi taas käyttää uudelleen soveluksessa ja lisäksi kehittäjät saattavat ymmärtää niistä enemmän. Käyttäjäkokemussuunnittelija voi myös tehdä varsinaisen käyttöliittymäsuunnitelman toteutuksen, mikäli kommunikaatio kehitystiimin ja käyttäjäkokemussuunnittelijan välillä on jatkuvaa. Kehittäjät saattavat tietää käyttäjäkokemussuunnittelijaa enemmän eri alustoihin liittyvistä tyyleistä. Kehittäjien vastuulla tulisikin olla teknisiin rajoitteisiin, alustan tyyleihin ja suorituskykyyn liittyvät tehtävät. [22]

4. FULL-STACK -SUUNNITTELIJAN ROOLIN MÄÄRITTELY

Full-stack -suunnittelijan rooli on saanut paljon huomiota viime vuosina. Internetissä erilaisilla keskustelupalstoilla on ollut keskustelua full-stack -suunnittelijan roolista. Keskustelua on käyty roolin määritelmästä [41, 7, 1, 26, 20, 15, 11], roolin hyödyllisyydestä - sekä puolesta että vastaan [41, 7, 1, 26, 28, 9, 20, 15, 11] - ja kysynnästä työmarkkinoilla [1, 9].

Suoraan tästä roolista tai roolissa työskentelevistä osaajista ei myöskään löytynyt aiempia tutkimuksia. Tähän rooliin sopivia työnkuvia löytyy voi löytyä eri tyyppisistä ja kokoisista yrityksistä. Vastaavia tehtäviä tekevät työntekijät voivat toimia esimerkiksi tavallisen käyttäjäkokemussuunnittelijan tai ohjelmistokehittäjän rooleissa [22, 25, 14].

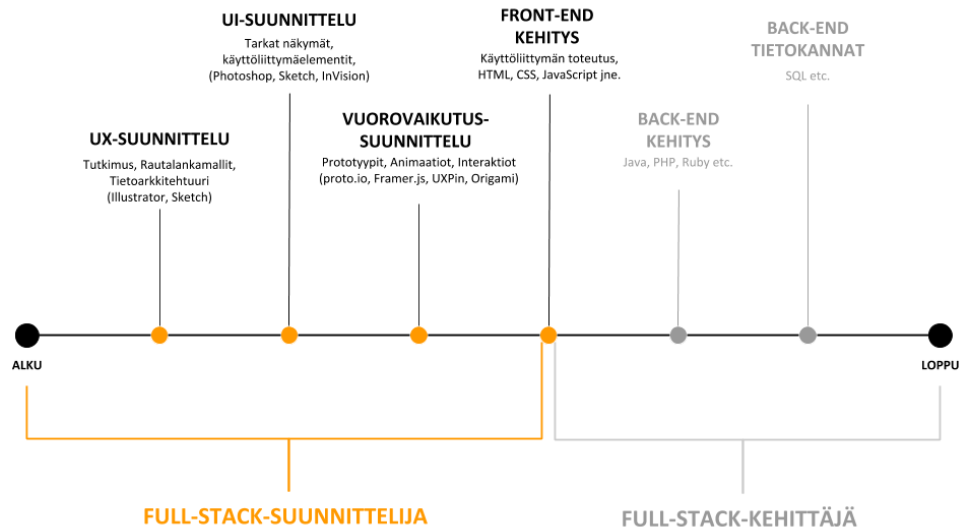
Termi ”full-stack -suunnittelija” (myöskään englanninkielinen termi full-stack designer) tai sen tarkoitus ei ole vielä täysin vakiintunut, mutta roolista on muodostettu yhtenäinen käsitys, mitä full-stack -suunnittelijan roolilla tarkoitetaan. Tässä luvussa määritellään se, mitä full-stack -suunnittelijan roolilla tarkoitetaan tässä diplomityössä. Lisäksi käydään läpi aiempia tutkimuksia, jotka liittyvät full-stack -suunnittelijan roolin työnkuvaan.

Canzibanin mukaan full-stack -suunnittelija tarjoaa tyyliohjeet verkkoon, animaatiot, graafiset materiaalit, vuorovaikutukseen liittyvät tiedot ja käyttöliittymäelementit, joita käytetään alustalla. Canzibanin mukaan monet yritykset rakentavat omia mukautettuja viitekehysjä (engl. frameworks), joiden avulla tietävät, mitä heidän tuotteensa käyttöliittymä sisältää. Viitekehysten luominen alusta loppuun eli käyttäjäkokemustutkimuksesta valmiisiin käyttöliittymäkomponentteihin, jotka edustavat tuotteessa monen eri osa-alueen konsepteja, ja tämä on full-stack -suunnittelua. [6]

Canziban painottaa, etteivät monet roolin tehtäviin kuuluvat osa-alueet välttämättä tarkoita, että roolissa työskentelevä työntekijä tekisi yksin kaiken suunnittelusta ohjelmointiin, vaan työntekijä on kehittänyt itselleen useampia taitoja näillä osa-alueilla. Nämä antavat työntekijälle valmiudet työskennellä itsenäisesti jokaisella projektin osa-alueella, sekä käyttäjäkokemussuunnittelun että kehityksen puolella. Full-stack -suunnittelija voi työskennellä koko suunnitteluprosessin ajan. [6]

Canzibanin mukaan full-stack -suunnittelijan ei tarvitse olla yhtä hyvä ohjelmoimaan kuin on suunnittelemaan, mutta full-stack -suunnittelijan tulee ymmärtää, kuinka front-end -ohjelmakoodi toimii ja minkälaisia teknologioita front-end -puolen ohjelmistokehitykseen sisältyy. Käyttäjäkokemussuunnittelijoiden ja kehittäjien tehtävien jakoon front-end -kehityksessä Canziban selventää, että kehittäjien tehtävänä on täydentää full-stack

-suunnittelijan toteuttamaa viitekehystä oikealla datalla ja yhdistämällä tarjottu viitekehys back-end-puoleen ja tietokantaan. Pääasia full-stack -suunnittelijan roolissa ei kuitenkaan ole ohjelmointi vaan kaiken käyttäjäkokemussuunnittelussa aikaansaadun viitekehyyksen, työvälineiden ja materiaalien yhdistäminen kokonaisuudeksi, josta kehittäjät voivat jatkaa työntekemistä. [6]



Kuva 8. Havainnollistava kuva full-stack -suunnittelijan roolin tehtävistä perustuen lähteisiin [6, 15, 26, 20, 41]

Englannin kielen sana ”stack” eli pino voi olla tuttu roolista full-stack -kehittäjä (engl. full-stack developer), missä pinolla viitataan yleensä web-teknologiakerrokseen: front-end ja back-end. Full-stack -suunnittelijassa pino viittaa suunnittelun eri kerrokseen seuraavasti:

- Käyttäjäkokemussuunnittelu (kuvassa UX-suunnittelu: tutkimus, rautalankamallit, prototyypit)
- Käyttöliittymäsuunnittelu (kuvassa UI-suunnittelu: näkymien suunnittelu, mockupit, ikonit ja materiaalit)
- Vuorovaikutussuunnittelu (UI-prototyypit, animaatio, ihmisvuorovaikutus)
- Front-end -kehitys, jossa kaksi eri osa-aluetta:
 - HTML ja CSS käyttöliittymän ulkoasun toteutukseen
 - Javascript sekä edistyneemmät CSS-ominaisuudet käyttöliittymän vuorovaikutusten ja animaatioiden rakentamiseen [6]

Yleensä full-stack tai full-stack -kehittäjä siis viittaa henkilöön, joka työskentelee hyvin jokaisella ohjelmistokehityksen kerroksella [10]. Full-stack -suunnittelijan ja -kehittäjän rooleihin liittyvät tehtävät ja yllä kuvatut vaiheet on esitetty myös kuvassa 8.

Full-stack -suunnittelijalla tarkoitetaan siis sellaista henkilöä, joka osaa tehdä tutkimustyötä, käyttäjäkokemussuunnittelua ja prototypointia sekä toteuttaa käyttöliittymäsuunnitelmia ohjelmoimalla. Nämä henkilöt tietävät kaikki eri suunnittelun osat mukaan lukien grafiikka, web, mobiili, ohjelmisto ja muita tuotekehityksen erityisosalueita. Full-stack -suunnittelija on siis käyttäjäkokemussuunnittelija, jolla on useampaa erilaista osaamista. Tällainen käyttäjäkokemussuunnittelija ymmärtää, suunnittelee ja ylläpitää koko tuotteen rakenteen. Hän myös voi mukautua erilaisiin projekteihin. [6]

Canzibanin mukaan full-stack -suunnittelijan etuja on juurikin se, että full-stack -suunnittelijan on mahdollista toteuttaa sujuvasti koko tuotekehitysprosessi. Lisäksi tämä voi hänen mukaansa säästää aikaa ja rahaa. Canzibanin mukaan full-stack -suunnittelijasta ei kuitenkaan hyödy ainoastaan suunnittelutiimi vaan myös muut tiimit yrityksessä, jopa työnantaja. Sen lisäksi, että full-stack -suunnittelija tietää ja osaa useita erilaisia asioita, full-stack -suunnittelijan etu on se, että hän tietää, mitä odottaa käyttöliittymäsuunnitelmilta. Full-stack -suunnittelija tietää, mikä on mahdollista toteutusvaiheessa ja mikä ei, mikä on hyvä tai huono lähestymistapa ja toimiiko se tietyssä käyttöliittymässä vai ei ja niin edelleen. [6]

Dashin mukaan full-stack -suunnittelijasta kasvaa poikkitieteellinen ammattilainen, joka voi työskennellä niin vuorovaikutussuunnittelun, käyttöliittymäsuunnittelun kuin käyttöliittymän toteutuksen ja prototypoinnin parissa. [10]

Full-stack -suunnittelijan rooli vastaa suurelta osin esimerkiksi Fox et al. määrittelemää generalisti-spesialisti-lähestymistapaa siltä osin, että käyttäjäkokemussuunnittelijalta löytyy kokemusta ja osaamista sekä käyttäjakeskeisen suunnittelun että ohjelmistokehityksen puolelta. Myöskään generalisti-spesialisti ei välttämättä osallistu suunnitteluun ja kehitykseen, mutta toimii apuna tiimin eri jäsenille. [14] Myöskin Ibragimova et al. määrittelemä UI/UX/UR-profiili sopii suunnitteluprosessin osalta full-stack -suunnittelijan rooliin. Ibragimovan et al. tutkimus mainitsi joillakin olevan myös ohjelmistokehitystaustaa, mutta tutkimus ei tarkemmin tuonut esiin, liittyykö tuo tausta esimerkiksi UI/UX/UR-profiiliin. [21]

Roolista on käyty keskustelua verkossa erilaisilla keskustelupalstoilla. Keskustelu on keskittynyt pääasiassa pohtimaan roolin määritelmää sekä rooliin liittyviä hyötyjä ja haasteita.

Verkkokeskustelut ja -artikkelit tukevat Dashin ja Canziban määritelmää full-stack -suunnittelijan roolista ja sen termistä. Näissä lähteissä roolin määritelmälle on ollut muutamaa erilaista näkökulmaa:

- Full-stack -suunnittelija ymmärtää prosessia sekä web-kehitystä teknisellä tasolla ja toteuttaa käyttöliittymäsuunnitelmansa toimivaksi prototyyppiksi [41, 28, 11].
- Full-stack -suunnittelija on henkilö, jolla on taitoja viedä suunnitteluprojekti loppuun eri osat toteuttaen (käyttäjäkokemussuunnittelu ja toteutus) [7, 20].

- Sama kuin edellinen väite, mutta korostetaan sitä, ettei rooli välttämättä tarkoita kaiken tekemistä [26, 15].

Määritelmää pohditaan erityisesti siitä näkökulmasta, kuinka pitkälle rooli ulottuu ohjelmistokehityksen puolella. Koska ohjelmointi voi nykyään tapahtua visuaalisella tavalla, tämä tarkoittaa sitä, että web-käyttäjäkokelemussuunnittelijat voivat kaikki olla full-stack -orientoituneita. [28] Vidal kommentoi, että prototypointi on helpompaa käyttämällä työkaluja kuin tekemällä sen itse [41]. Vidal lisäksi kommentoi startup-yritysten olevan paras paikka full-stack -rooleille. [41] Myös muualla [26, 11] pohditaan sitä, että monet pienet kehitystiimit eivät voi täyttää jokaista roolia eri työntekijällä, mikä pakottaa tiimin jäseniä toimimaan useammassa roolissa samaan aikaan.

Verkossa käydyn keskustelun pohjalta on myös mahdollista määrittää suuntaa antavia näkemyksiä siitä, minkälaisia hyötyjä ja haasteita full-stack -suunnittelijasta on. Alla on tiivistelmät verkkokeskusteluissa käydyistä hyödyistä sekä haasteista.

Roolin hyödyt

- Roolin monipuolisuus [1]
- Omien taitojen ja osaamisen kehittäminen [1, 20]
- Kokonaisvaltainen ajattelu [1, 26, 28, 20]
- Ymmärrys kehitys- ja suunnitteluprosessista [26, 28, 20, 11]
- Vähentää töiden tekemistä uudelleen ja odottamattomia tilanteita [26, 20]
- Kehitysprosessin ja sen rajoitteiden ymmärtäminen [28]
- Realististen prototyyppien / käyttöliittymäsuunnitelmien luominen [28, 20, 26]
- Sujuva tiimityöskentely [28, 20, 26, 11]
- Käyttöliittymäsuunnitelmista kestävämpää arvoa, koska huomioidaan useampi tuotekehitykseen liittyvä osa-alue [20]
- Käyttäjäkokelemussuunnittelu aina mielessä [20]
- Yhteensopivuus Leanin ja ketterän kehityksen menetelmiin [20]
- Yritys voi säästää kustannuksissa ja resursseissa, jos yksi ihminen hoitaa useamman ihmisen tehtävät [11]

Roolin haasteet

- Työnantajan joutuu maksamaan työntekijälle enemmän palkkaa kuin tavalliselle käyttäjäkokelemussuunnittelijalle [7]
- Osaamisen ylläpitäminen [1, 28, 11]
- Jokaisen roolin osa-alueen omaksuminen [28, 11]
- Ei voi korvata muita rooleja [28, 11]

- Ohjelmointi vaatii pitkää harjoittelua [28]
- Suunnittelun rajoittaminen, koska rajoittunut ohjelmointitaito [28, 9] tai ei halua itse tehdä muutoksia [9]
- Ymmärrys teknologiasta voi estää full-stack -suunnittelijaa luomasta parasta käyttöliittymäsuunnitelmaa [28]
- Henkinen kuormitus [28, 9]
- Työntekijän resurssit (työntekijällä rajallisesti tunteja käytettävissä ja mahdollista keskittyä yhteen tehtävään) [9, 11]

Yllä mainitut verkkolähteet koostuvat eri ihmisten mielipiteistä. Vaikka osa näistä mielipiteistä voi perustua kirjoittajien todellisiin kokemuksiin, laajempi tutkimus osoittamaan näiden kokemusten yleisyyttä on tarpeen. Koska aiempia tutkimuksia suoraan full-stack -suunnittelijan roolista ei diplomityön tekovaiheessa löytynyt, näitä eri verkkolähteistä löytyviä näkemyksiä pidetään tämän työn osalta suuntaa antavana aineistona.

Yllä esiteltyjen lähteiden ja määritelmien avulla on määritelty full-stack -suunnittelijan rooli tämän työn kannalta. Tässä työssä full-stack -suunnittelijana tarkoitetaan sellaista henkilöä, jolla on osaamista kaikilta tuotesuunnittelun osa-alueilta (käyttäjäkokesuunnittelu, käyttöliittymäsuunnittelu, vuorovaikutussuunnittelu sekä käyttöliittymien toteutus), kuten esitetty aiemmin kuvassa 8. Lyhyesti määriteltynä full-stack -suunnittelija on siis käyttäjäkokesuunnittelija, joka osaa tarvittaessa viedä itsenäisesti tuotesuunnitteluprosessin alusta loppuun tutkimuksesta käyttöliittymäsuunnitelman toteuttamiseen.

5. TUTKIMUSPROSESSI

Tässä työssä tehtiin empiiristä tutkimusta full-stack -suunnittelijan roolista. Tämä luku kuvaa diplomityön tutkimustavoitteet, jossa käydään läpi vielä johdannossa esitetyt tutkimuskysymykset. Tämän jälkeen kuvataan tutkimusprosessia. Sitten perustellaan tutkimukseen valitut tutkimusmenetelmät eli niiden edut ja haitat tutkimuksen kannalta. Lisäksi kuvataan aineiston käsittely ja analyysimenetelmät.

5.1 Tutkimuksen tavoitteet

Tutkimuksessa tutkittiin full-stack -suunnittelijan roolia sekä sen hyötyjä ja haasteita yrityksen sekä työntekijöiden näkökulmasta. Tavoitteena oli ymmärtää paremmin full-stack -suunnittelijan rooliin kuuluvia ominaispiirteitä ja tehtäviä, sekä sitä, miten rooli toimii yhdessä kehittäjien kanssa ketterässä ohjelmistokehityksessä. Koska roolista ei löytynyt tällä termillä aiempia tutkimuksia tämän tutkimuksen toteutusvaiheessa, tämä tutkimus rajattiin pääasiassa roolin yleiskuvan ymmärtämiseen jonkun tietyn roolin osan tutkimisen sijaan.

Tutkimustehtävä muuttui ja tarkentui jonkin verran tutkimuksen aikana. Hirsjärvi et al. [19] mainitsevatkin, että kvalitatiivisessa tutkimuksessa tutkimusongelma tai -tehtävä saattaa muuttua tutkimuksen edetessä. Tutkimuskysymyksiksi valikoituivat seuraavat kolme kysymystä, kuten esitetty luvussa 1.3:

1. Mitä hyötyjä ja haasteita full-stack -suunnittelijan roolista on yritykselle?
2. Mitkä ovat työntekijöiden näkemykset ja kokemukset full-stack -suunnittelijan roolin hyödyistä ja haasteista?
3. Miten full-stack -suunnittelijan roolin haasteet voidaan ratkaista?

Tämän diplomityön tutkimuksella oli kolme erilaista tarkoitusta: tutkimuksen on tarkoitus olla sekä kartoittava että selittävä tutkimus. Kartoittavalla tarkoitetaan esim. uusien näkökumien ja ilmiöiden löytämistä ja selvittää vähän tunnettua ilmiötä. Selittävä tarkoitus etsii selitystä tilanteelle. [19] Tämä tutkimus siis pyrkii selvittämään full-stack -suunnittelijan rooliin liittyviä näkökulmia, hyötyjä ja haasteita, selvittää, miksi roolista on hyötyä tai haasteita ja kuinka näitä haasteita voidaan ratkaista.

Tutkimuksessa pyritään myös varmistamaan tälle työlle määritellyn termin ”full-stack -suunnittelija” ja sen määritelmän paikkansa pitävyys. Itse tutkimus sijoittui lokakuulle ja marraskuulle vuonna 2018.

5.2 Tutkimuksen eteneminen

Diplomityö aloitettiin aiheen valitsemisella ja rajauksella. Aiheen valintaprosessi alkoi keskustelemalla alalla työskentelevien käyttäjäkokemussuunnittelijoiden ja kehittäjien kanssa. Aiheen valintaan vaikuttivat esimerkiksi Hirsjärven [19] mainitsevat perusteet, kuten aiheen kiinnostavuus ja sopivuus tieteenalalle ja se, että aiheesta löytyy tarpeeksi tietoa tutkimustyön tekemiseksi. Kun ajatus sopivasta diplomityöaiheesta oli selvinnyt, aihetta pyrittiin rajaamaan erilaisin käsitekarttojen avulla. Tällä tavoin voitiin hahmottaa aiheen laajuutta sekä tarvittaessa rajata sitä pienemmäksi. Aihe rajautui ja tarkentui lisää myös työn edetessä.

Varsinaisen työn aloitus alkoi alustavan aikataulun laatimisella, aiheeseen perehtymisellä sekä tutkimussuunnitelman laatimisella. Tutkimussuunnitelmaa laatiessa pohdittiin sopivat tutkimuskysymykset, vastausten saamiseksi sopivia tutkimusmenetelmiä sekä tutkimuksen kohderyhmää. Kun alustava tutkimussuunnitelma oli valmiina, työtä jatkettiin kirjallisuuskatsauksella eli kerättiin talteen aiheeseen liittyvää teoriaa ja selvitettiin aiheeseen liittyviä aiempia tutkimustuloksia.

Sillä työn tutkimuskohteena on rooli, josta ei suoraan löytynyt tutkimusvaiheessa aiempia tutkimuksia ja jonka termi on suhteellisen uusi, yksi tärkeimmistä vaiheista työn alussa oli termin ”full-stack -suunnittelija” määrittäminen tämän diplomityön kannalta. Tutkimuskohteena oleva rooli kuvattiin tarkemmin 4.

Teoriakatsauksen ja vaadittavien taustatietojen selvitysten jälkeen suunniteltiin tarkemmin tässä työssä tehty tutkimus. Tutkimusmenetelmien yhteydessä hyödynnettiin jo olemassa olevaa tietoa lähteistä sekä aiheeseen liittyvistä aiemmista tutkimustuloksista. Kun tutkimus oli suunniteltu ja kartoitettu työn tutkimuskohteet, toteutettiin tutkimus seuraavissa luvuissa tarkemmin kuvatuilla menetelmillä.

Kun tutkimus oli suoritettu ja tarvittavat tutkimustulokset saatu kerättyä, tutkimustulokset dokumentoitiin ja analysointiin tämän luvun lopussa kuvatulla tavalla. Tutkimustulosten dokumentoinnin ja analysoinnin jälkeen, viimeisteltiin tämä tutkimusraportti.

5.3 Tutkimuksen aineisto

Työssä toteutetussa tutkimuksessa yhdistyy sekä kvalitatiivinen eli laadullinen sekä kvantitatiivinen eli määrällinen tutkimus [19]. Kvantitatiivisella tutkimuksella pyritään löytämään yleispäteviä syyn ja seurauksen lakeja. Tähän sisältyy esimerkiksi johtopäätösten tekeminen aiemmista tutkimuksista, teorioiden selvittäminen sekä käsitteiden määrittely. Tutkimuksessa tutkittavien valinta voidaan tehdä satunnaisotannalla ja kohdia perusjoukosta valittuun otokseen. [19] Tutkimuksen kohderyhmä eli perusjoukko on kuvattu myöhemmin tässä luvussa.

Kvalitatiivisessa tutkimuksessa taas pyritään tutkimaan kohdetta kokonaisvaltaisesti. Kvalitatiivisessa tutkimuksessa on pyrkimyksenä löytää tai paljastaa tosiasioita jo olemassa

olevien väittämien todentamisen sijaan. Laadullisessa tutkimuksessa tutkimuksen kohdejoukko valitaan tarkoituksenmukaisesti. Kvantitatiivisessa tutkimuksessa määritellään perusjoukko ja poimitaan perusjoukosta edustava otos. satunnaisotanta, systemaattinen otanta, ositettu otanta, ryväotanta. [19]

Kvalitatiivinen tutkimuksessa aineiston kooksi riittää yksi tapaus. Aineiston riittävyteen, kylläisyyteen viittaavaa saturaation käsitettä (kylläntyminen). Tällä tarkoitetaan että tutkija alkaa kerätä aineistoa päättämättä etukäteen miten monta tapausta hän tutkii, jatkaa niin kauan kuin uutta tietoa tulee ja lopettaa kun aineisto alkaa kertaantua. Ongelma tässä on kuitenkin se, että kuinka paljon voi löytää tai huomata jatkuvasti uusia näkökulmia. [19] Laadullisen tutkimuksen yleisempiä aineistonkeruumenetelmiä ovat esimerkiksi haastattelu ja kysely. Menetelmiä voidaan käyttää joko vaihtoehtoisesti, rinnan tai eri tavoin yhdisteltynä tutkittavan ongelman ja tutkimusresurssien mukaan. [40]

Tutkimuksen kohderyhmänä eli perusjoukkona olivat erilaisissa ohjelmistoyrityksissä työskentelevät ohjelmistojen käyttäjäkokemussuunnittelijat ja kehittäjät sekä tutkittavassa roolissa työskentelevät työntekijät. Kohdeyrityksiltä, joissa käyttäjäkokemussuunnittelijat ja kehittäjät työskentelivät, edellytettiin ketterien menetelmien hyödyntämistä osana ohjelmistokehitystä.

5.4 Tutkimusmenetelmät

Tutkimusmenetelmiksi valikoituivat sekä laadulliseen että määrälliseen tutkimukseen sopeva kysely [19]. Tutkimuksen lisäksi tehtiin kirjallisuusselvitystä. Kyselyn avulla toivottiin saada kerätyksi laajemmin tietoa tutkittavasta roolista sekä rooliin liittyviä yleistyksiä. Valitut tutkimusmenetelmät sekä niiden toteutus tavat on kuvattu tarkemmin alla olevissa luvuissa.

Kysely valikoitui tutkimusmenetelmäksi siksi, että se soveltuu laajemman tutkimusaineiston keräämiseen. Kyselyn avulla voidaan tavoittaa useampia henkilöitä sekä kysyä paljon kysymyksiä. Kyselyn muita etuja ovat esimerkiksi se, että se säästää tutkijalta aikaa ja vaivaa, ja se, että kyselyä voidaan käsitellä ja analysoida nopeasti, mikäli kysely on huolellisesti suunniteltu. [19] Kyselyn haasteita ovat vastaajiin liittyvien haasteiden kontrollointi, kuten vastaajien perehtyneisyys tai suhtautuneisuus tutkittavaan asiaan, sekä väärinymmärrysten kontrollointi kyselytilanteessa. Lisäksi hyvän lomakkeen laatiminen vie paljon aikaa, sekä vaatii kyselyn laatijalta tietoa ja taitoa kyselyn tekemisestä. Myöskin vastauskato eli kyselyyn vastaamattamuus voi nousta suureksi. Kyselyn avulla voidaan kerätä laajasti erilaista tietoa esimerkiksi vastaajan uskomuksista, käsityksistä ja mielipiteistä sekä perusteluja näihin tietoihin. [19]

Kyselyn määrittely oli pitkä prosessi. Kyselyn kysymysten määrittely liian aikaisessa vaiheessa tutkimusprosessia viivästytti tutkimuksen etenemistä. Kyselyn kysymysten määrittely ilman kattavaa lähteisiin tutustumista ja tutkimussuunnitelman määrittelyä aiheutti

sen, että kysymyksiä tuli keksittyä liikaa ja osa kysymyksistä ei kysynyt aiheen kannalta tärkeitä kysymyksiä. Tästä oppineena ennen lopullisen kyselyn muodostamista työssä syvennyttiin ensin paremmin teoriaan sekä aloitettiin kyselyn kysymysten keksimisen valitsemalla ensin vastaukset, joita tutkimuksella haluttiin saada. Tämä sai aikaan paremmin valikoituja kysymyksiä ja rajatun kyselyn.

Kyselystä pyrittiin saamaan kattava, mutta siten, että se olisi helposti ja nopeasti täytettävä. Tästä syystä kyselyyn valittiin enimmäkseen suljettuja kysymyksiä, joissa oli etukäteen määritellyt vastausvaihtoehdot. Vastausvaihtoehdot, esimerkiksi kyselyssä esitetyt väittämät, valittiin kirjallisuusselvitysten ja verkkoaineistojen pohjalta. Suljettujen kysymysten lisäksi kyselyssä pyrittiin kysymään myös muutamia avoimia kysymyksiä uusien näkemysten saamiseksi.

Kun kyselyn kysymykset oli määritelty, itse kysely toteutettiin verkossa Google Forms -työkalulla [16]. Google Forms valikoitui kyselypohjan tekoon siksi, että sen käytöstä oli aiempaa kokemusta, kyselyn jakaminen on helppoa verkon kautta ja se tarjoaa työkaluja tutkimusaineiston analysointiin. Lisäksi tutkija voi helposti aloittaa ja päättää vastausten keräämisen.

Ennen kyselyn lähettämistä eteenpäin laajemmin tutkimuksen kohderyhmälle, kyselyn luoki läpi ja teki useampi ihminen, jotta varmistuttiin siitä, että kyselyn kysymykset ja niiden vastausvaihtoehdot olivat selkeitä. Kysely annettiin luettavaksi kahdelle käyttäjäkokemussuunnittelijalle, joilla oli aiempaa kokemusta kyselyiden tekemisestä. Kyselyn oikoluku kahden käyttäjäkokemussuunnittelijan toimesta tapahtui siten, että käyttäjäkokemussuunnittelijat saivat kyselyn pdf-muodossa tarkasteluun, he saivat rauhassa aikaa tutustua kyselyn runkoon ja kysyä tarkentavia kysymyksiä ja lopuksi esittää korjausehdotuksia.

Tämän jälkeen kyselyä testattiin kahdella kehittäjällä. Linkki toteutettuun kyselyyn jaettiin näille kahdelle eri kehittäjälle, ja he täyttivät kyselyn heille sopivana ajankohtana. Toinen kyselyyn vastanneista kehittäjistä antoi palautteensa kyselyn palaute-osioon ja toinen antoi palautteen viestillä jälkikäteen.

Kyselyyn tehtiin korjauksia käyttäjäkokemussuunnittelijoiden ja kehittäjien antamien palautteiden perusteella. Koska joitain korjauksia jouduttiin tekemään myös kyselyn rakenteeseen ja vastausvaihtoehtoihin, kyselyyn osallistuneiden testihenkilöiden, kahden kehittäjän, vastauksia ei voitu ottaa mukaan varsinaiseen tutkimukseen.

Kyselyn huolellisen testaamisen jälkeen kysely välitettiin eteenpäin varsinaiselle kohderyhmälle. Kyselyyn valittiin perusjoukosta sopiva otos ensin selvittämällä verkosta alalla toimivia ohjelmistoyrityksiä, jotka käyttävät ohjelmistojen kehityksessä ketteriä menetelmiä. Selvitys tehtiin verkossa Google-hakukoneen avulla ja yritysten verkkosivuilla ilmoitettujen tietojen perusteella. Perusjoukkoa vastaava otos valikoitui pääosin satunnaisesti, mutta otokseen valittuja yrityksiä jaettiin lisäksi yrityksen koon mukaan siten, että jokaisesta kokoryhmässä olisi mukana useampia yrityksiä. Tässä työssä yrityksen koko määriteltiin yrityksessä työskentelevien työntekijöiden lukumäärän mukaan seuraavasti:

- mikroyritys (alle 10 työntekijää)
- pieni yritys (10 - 49 työntekijää)
- keskisuuri yritys (50 - 249 työntekijää)
- suuri yritys (250+ työntekijää) [38, 39].

Työntekijöiden lukumäärää selvitettiin yritysten verkkosivuilta sekä yritysrekistereistä, jotta saatiin tarkkaa työntekijämäärää vastaava luku tai mahdollisimman tarkka arvio siitä, mille yllä mainituista lukumääräväleistä työntekijöiden lukumäärä sijoittuu. Otoksen lajittelu yrityksessä työskentelevien työntekijöiden mukaan tehtiin siksi, jotta saataisiin kasvatettua vastausten saamisen todennäköisyyttä eri kokoisilta yrityksiltä. Linkki Google Forms -työkalulla tehtyyn kyselyyn lähetettiin yrityksille sähköpostilla. Taulukossa 4 on esitetty sähköpostien lähetysmäärät eri kokoisille yrityksille.

Taulukko 4. Kyselyn vastaanottaneet yritykset

Yrityksen koko	Yritysten määrä
Mikroyritys	2
Pieni yritys	5
Keskisuuri yritys	3
Suuri yritys	4

Kyselyä pidettiin auki viikon ajan, minkä jälkeen kyselyyn ei hyväksytty enää uusia vastauksia. Tämän jälkeen kyselystä saatu data dokumentoitiin ja analysoitiin. Kyselyn tutkimustulokset ja näiden pohjalta tehty analyysi on nähtävissä seuraavassa luvussa.

Kyselyn kysymykset oli teemoitettu. Alussa kyselyssä kysyttiin vastaajan esitiedot, minkä jälkeen kysely jakaantui 7 aihealueeseen:

- Työntekijöiden oma kiinnostus ja osaaminen suunnittelun osa-alueilla
- Nykyinen kokemus suunnittelusta ja ohjelmistokehityksestä
- Full-stack -suunnittelijan roolin tunnettavuus ja määritelmä
- Full-stack -suunnittelijan roolin hyödyt, haasteet sekä ratkaisut haasteisiin
- Työntekijöiden kokemus roolin toimivuudesta yrityksessä
- Työntekijöiden mielipiteet full-stack -suunnittelijasta
- Työntekijöiden kiinnostus roolissa toimimiseen

Kyselyn lopussa kysyttiin vielä vastaajan omat tiedot. Yrityksille lähetetyn kyselyn runko löytyy liitteestä A.

5.5 Tutkimusaineiston analysointi

Kysely toteutettiin siten, että niistä ei voida tunnistaa yksittäisiä henkilöitä tai yrityksiä, joissa henkilöt työskentelevät. Kyselyssä kysyttiin erilaisin teemojin kysymyksiä, joiden vastauksia oli mahdollista analysoida sekä määrällisin että laadullisin menetelmin. Kysely sisälsi sekä suljettuja että avoimia kysymyksiä.

Kyselyn sisältämien strukturoitujen eli suljettujen kysymysten (niiden kysymysten, joihin vastausvaihtoehdot oli valmiiksi määritelty) [19] vastaukset taulukoitiin, ja niistä pyrittiin muodostamaan johtopäätöksiä taulukoinnin ja kaavioiden avulla. Aineistosta litteroitiin eli kirjoitettiin puhtaaksi [19, 40] avoimiin kysymyksiin tulleet vastaukset, joita esiintyi seuraavissa kyselyn teemoissa: full-stack -suunnittelijan roolin tunnettavuus ja määritelmä, full-stack -suunnittelijan roolin hyödyt, haasteet sekä ratkaisut haasteisiin, työntekijöiden kokemus roolin toimivuudesta yrityksessä sekä työntekijöiden kiinnostus roolissa toimimiseen. Myös kyselystä saadut palautteet litteroitiin.

Litteroinnin jälkeen hyödyt, haasteet ja ratkaisut sekä teemoiteltiin että luokiteltiin. Teemoittelu on laadullisen aineiston pilkkomista ja ryhmittelyä. [40] Teemoittelulla keskityttiin vastauksien yhteisten teemojen löytämiseen. Lisäksi hyödynnettiin luokittelua. Luokittelua pidetään kvantitatiivisena analyysinä, jonka avulla datasta määritellään luokkia ja lasketaan, kuinka monta kertaa luokka esiintyy aineistossa [40]. Tutkimuksessa löydetyt luokat (teemat) esitettiin taulukkona. Lisäksi vielä selvitettiin, mitä teemoista oltiin tarkemmin sanottu. Lopuksi löydetyt teemat vielä ryhmiteltiin tietyiksi tyypeiksi. Tämä tarkoitti yhteisten ominaisuuksien etsimistä teemojen sisältä [40]. Kyselyn tutkimustulokset on esitetty seuraavissa luvuissa.

6. TUTKIMUSTULOKSET

Verkkokysely lähetettiin 13:lle eri suomalaiselle ohjelmistoyritykselle, jotka toimivat Pääkaupunkiseudulla, Pirkanmaalla sekä Varsinais-Suomessa, ja yrityksiä pyydettiin välittämään kysely eteenpäin heidän työntekijöilleen. Vastauksia kyselyyn tuli yhteensä 96 kappaletta. Seuraavissa alaluvuissa on käyty läpi kyselyn tulokset luvussa 5 esiteltyjen kyselyn teemojen mukaan.

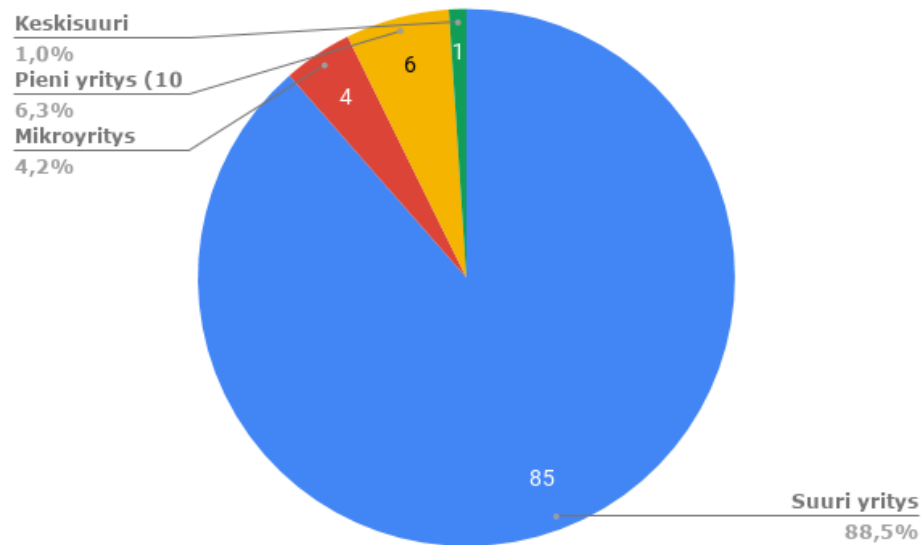
Kyselyssä käytettiin tekstien lyhentämisen ja lukemisen helpottamisen vuoksi seuraavia termejä: **UX-suunnittelu** (käyttäjäkokenussuunnittelu), **UI-suunnittelu** (käyttöliittymäsuunnittelu) sekä vuorovaikutussuunnittelu. Kyselyssä käytettiin myös englanninkielistä termiä ”**design**” käyttöliittymäsuunnittelusta, jotta saataisiin lyhyempi ja vastaajien ammattisanaston kannalta ymmärrettävämpi termi. Käyttäjäkokenussuunnittelijaan viitataan termillä ”**suunnittelija**”. Tässä luvussa käsitellään kyselyn tuloksia näiden kyselyssä mainittujen termien mukaan siitä syystä, jotta kuvaajien ja taulukoiden tulkitseminen tekstin lukemisen ohella olisi lukijalle nopeampaa ja johdonmukaisempaa.

6.1 Yleistä tietoa kyselyyn vastanneista

Kyselyyn vastasi joukko eri ikäisiä, erilaisilla rooleilla eri kokoisissa yrityksissä työskenteleviä ohjelmistoalan ammattilaisia. Vastaajien joukossa oli sekä miehiä että naisia. Kyselyyn vastanneiden syntymävuodet sijoituivat vuosien 1966 - 1998 välille. Suurin vastaajajoukko oli syntynyt vuosien 1980 - 1989 välillä (54 vastaajaa). Vastaajien koulutustausta myös vaihteli. Vastaajien koulutustaustoista eniten oli ylempää korkeakoulututkintoa (51,1% vastaajista) tai alempaa korkeakoulututkintoa (26%).

Kyselyssä ei identifioitu sitä, minkä yrityksen työntekijöiltä vastaukset tulivat. Sen sijaan kyselyssä vastaajat määrittelivät itse, minkä kokoisessa yrityksessä he työskentelivät. Tämä valinta tehtiin siksi, jotta vastaaminen säilyisi täysin anonymina, erityisesti pieniä, muutaman henkilön yrityksiä kohtaan. Kyselyn tuloksista voidaan todeta, että suurista yrityksistä (250 työntekijää tai enemmän) vastauksia oli eniten (88,7%), toiseksi eniten pienistä yrityksistä (6,2%), kolmanneksi eniten mikroyrityksistä (4,1%) ja vähiten keskikokoisista yrityksistä (yksi vastaus, 1%). Jakauma yritysten koosta on esitetty lisäksi kuvassa 9.

Yrityksen työntekijöiden määrää ja sen luotettavuutta tulee tarkastella kriittisesti. Yritysten arvioidut työntekijämäärät perustuivat verkosta löytyviin (yritysten verkkosivut, vuosikatselmukset) tietoihin, jotka ovat saattaneet muuttua työntekijöiden vastatessa kyselyyn. Työntekijät myös itse saattavat arvioida yrityksen koon väärin. Nämä saattavat



Kuva 9. Kyselyyn vastanneiden työntekijöiden ilmoittamien yritysten kokojakauma

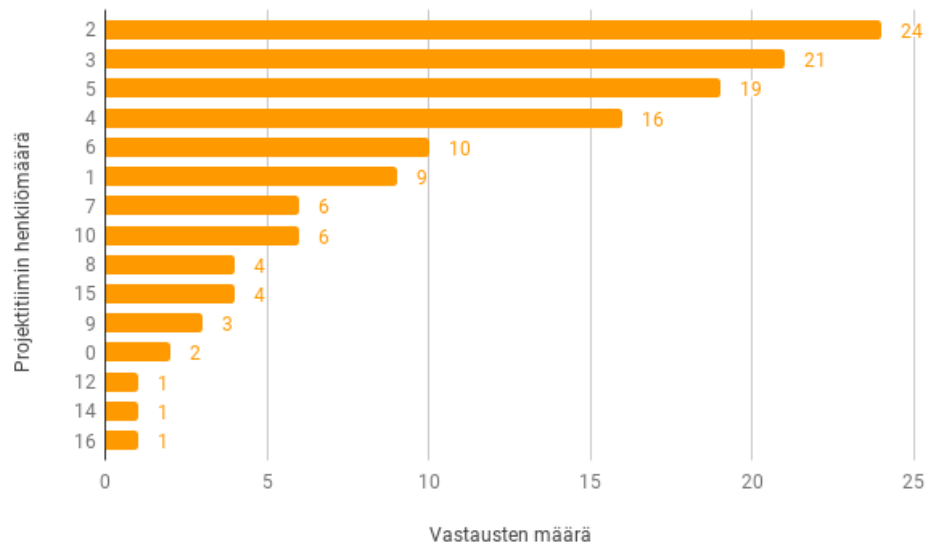
aiheuttaa vääristymää yrityksen koon vastauksissa. Suuri vastauslukumäärä suurista yrityksistä voi kuitenkin johtua juurikin siitä, että kysely on suurissa yrityksissä tavoittanut suuremman määrän ihmisiä, mikä lisää sitä mahdollisuutta, että näistä yrityksistä vastaan kyselyyn helpommin.

Vastausten perusteella web-sovellusprojektit olivat yleisimpiä projekteja vastaajien joukossa (86,5% mainituista projekteista) ja toiseksi yleisimpiä olivat mobiilisovellusprojektit (28,1%). Muut kyselyssä etukäteen määritellyt projektit olivat yleisyys vastaajien keskuudessa vähemmän yleisiä: palvelumuotoilu (14,6%), sulautetut järjestelmät (13,5%) ja työpöytäsovellukset (10,4%). Muita näiden lisäksi mainittuja yksittäisiä projekteja olivat eri alustoille kehittäminen, SaaS (Software as a Service), data-analyysi sekä järjestelmäintegraatiot ja tiedonhallinta.

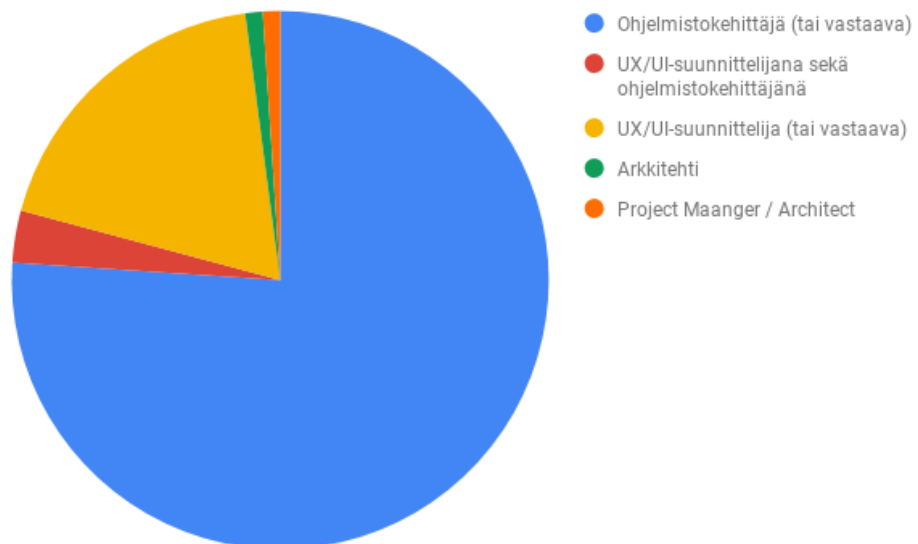
Kuvassa 10 on esitetty tiimien koot, joissa vastaajat työskentelivät vastaushetkellä. Eniten esiintyi pienempiä tiimejä, alle 5 henkilön tiimejä, mutta myös isompia yli 10 henkilön oli jonkin verran. Kokonaismäärä on suurempi kuin kyselyyn vastanneiden määrä, sillä 17 vastaajaa (17,7% vastaajista) työskenteli useammassa kuin yhdessä projektissa samaan aikaan, ja heistä seitsemän oli suunnittelijoita.

Vastaajien roolit jakaantuivat kuvan 11 mukaisesti. Suurin osa vastaajista (76%) toimi kehittäjän roolissa ja toiseksi suurin osa vastaajista (18,8%) käyttäjäkokemussuunnittelijan roolissa. Vastaajista kolme työskenteli molemmissa näissä rooleissa. Muita mainittuja rooleja olivat arkkitehti sekä projektipäällikkö/arkkitehti.

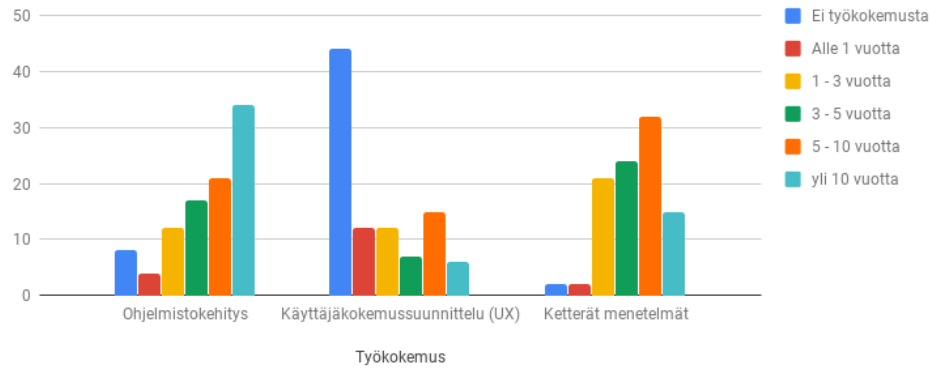
Kaikki vastaajista olivat työskennelleet vähintään kaksi vuotta IT-alalla. Pisimpään IT-alalla työskennelleet olivat työskennelleet alalla 25 vuotta. Kuvassa 12 on esitetty vastaajien työkokemus eri aihealueissa: ohjelmistokehityksessä, käyttäjäkokemussuunnittelussa



Kuva 10. Vastaajien projektitiimien henkilömäärä



Kuva 11. Vastaajien roolit yrityksissä



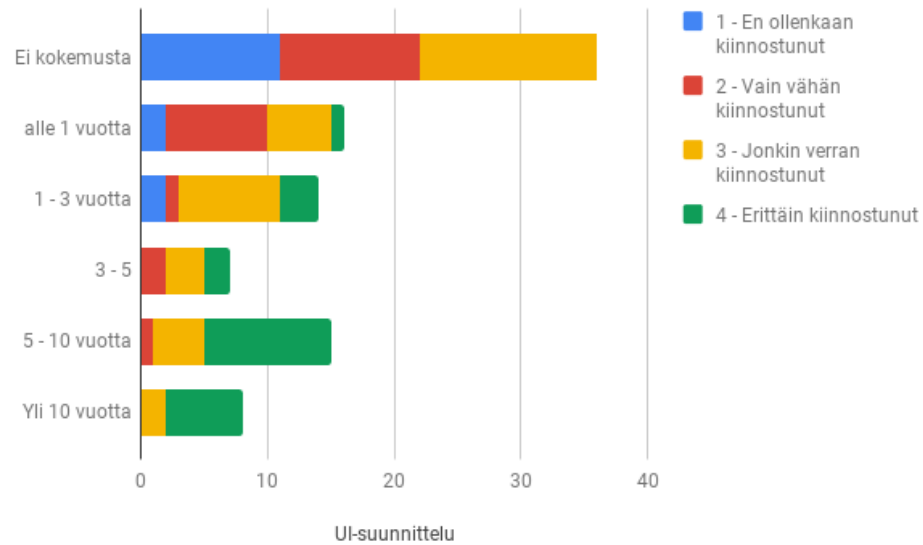
Kuva 12. Vastaajien työkokemukset ohjelmistokehityksestä, käyttäjäkokeussuunnittelusta sekä ketteristä menetelmistä

sekä ketterään ohjelmistokehityksessä. Useammalla vastaajalla oli monen vuoden työkokemus ohjelmistokehityksestä. Muutamalla vastaajalla työkokemus oli ohjelmistokehityksestä oli lyhyempi. Käyttäjäkokeussuunnittelussa työkokemusta ei ollut kaikilla vastaajilla yhtä paljon kuin ohjelmistokehityksestä. Suuri piikki ei työkokemusta -vastauksen kohdalla johtuu siitä, että suuri osa vastaajista työskenteli kehittäjänä. Myös ketterissä menetelmissä työkokemusta oli vastaajilla useampia vuosi, ainoastaan yhdellä vastaajalla ei ollut työkokemusta ketteristä menetelmistä.

6.2 Työntekijöiden oma kiinnostus ja osaaminen suunnittelun osa-alueilla

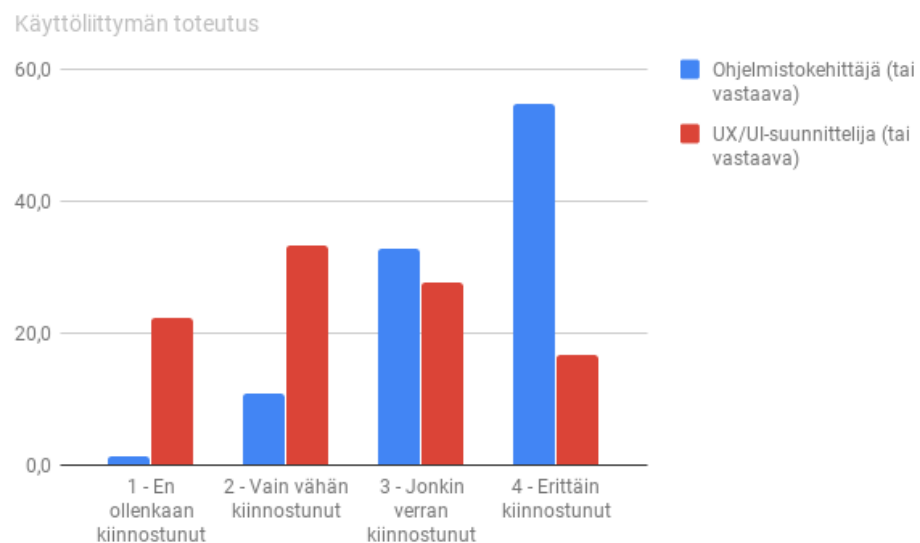
Työntekijöillä oli full-stack -suunnittelun osa-alueilla (UX-suunnittelu, UI-suunnittelu, vuorovaikutussuunnittelu, käyttöliittymän toteutus) kiinnostusta ja kokemusta vaihtelevasti. Vastauksista oli huomattavissa, että kiinnostus eri osa-alueisiin kasvoi työkokemuksen lisääntyessä. Tämä oli jokseenkin odotettu tulos. Huomionarvoinen seikka tuloksissa oli kuitenkin se, että suunnitteluun liittyvät osa-alueet (UX-suunnittelu, UI-suunnittelu ja vuorovaikutussuunnittelu) olivat hyvin kiinnostava osa-alue myös niiden vastaajien joukossa, joilla ei ollut niihin kuuluvien tehtävien tekemisestä aiempaa kokemusta. Käyttöliittymien toteutuksessa korrelaatio ei ollut yhtä selkeä, vaan myös pidempään työskennelleiden joukossa oli edelleen kiinnostuksen vähäisyyttä. Kuvassa on 13 on esitetty esimerkkinä kiinnostuksen ja kokemuksen korrelaatio UI-suunnittelun suhteen.

Pelkästään suunnittelijoiden ja kehittäjien vastauksia arvioitaessa eri full-stack -suunnittelun osa-alueilla, työntekijöiden antamat vastaukset vastasivat odotettuja tuloksia: suunnittelijat olivat pääasiassa enemmän kiinnostuneita UX- ja UI-suunnittelusta sekä vuorovaikutussuunnittelusta ja kehittäjät puolestaan käyttöliittymien toteutuksesta. Kuitenkin, kuten kuvasta 14 voidaan todeta, jopa 44,5% kyselyyn vastanneista suunnittelijoista oli jonkin verran tai erittäin kiinnostuneita toteuttamaan käyttöliittymiä. Kehittäjien keskuudessa vähäinen kiinnostus käyttöliittymien tekemiseen on voinut johtua siitä, että nämä kehittäjät



Kuva 13. Vastaajien kiinnostuksen ja työkokemuksen korrelaatio UI-suunnittelun osa-alueella

työskentelevät pääasiassa back-end-kehityksessä eikä heidän työnkuvaansa kuulu käyttöliittymien toteuttaminen.

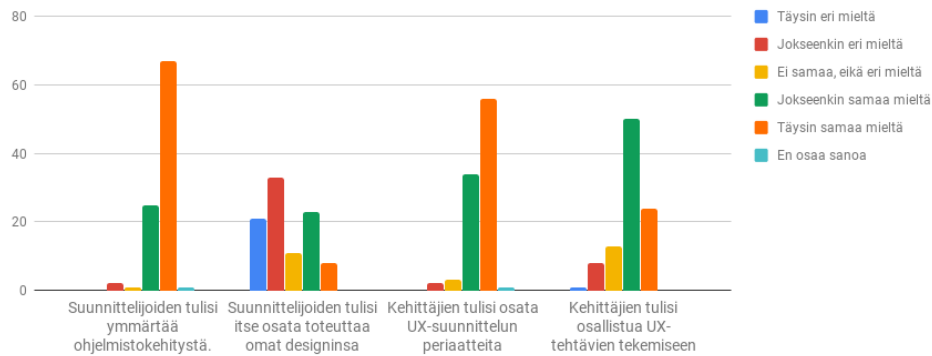


Kuva 14. Suunnittelijoiden ja kehittäjien kiinnostus käyttöliittymien toteutukseen

6.3 Nykyinen kokemus suunnittelusta ja ohjelmistokehityksestä

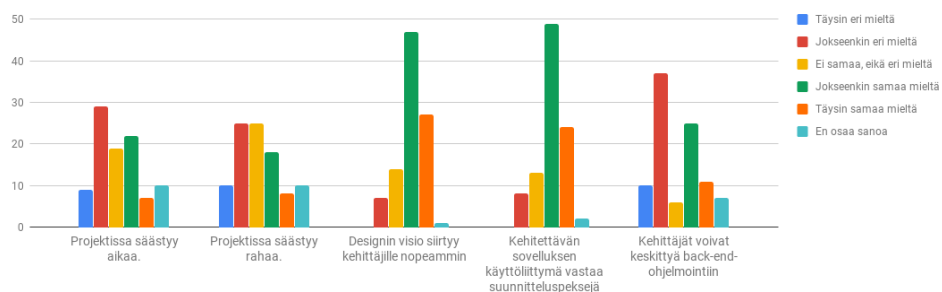
Työntekijöiltä kysyttiin, mitä mieltä he olivat erilaisista suunnitteluun ja ohjelmistokehitykseen liittyvistä väittämistä. Kyselyssä esitettyjen väittämien tarkoituksena oli kartoit-

taa vastaajien kokemuksia nykyisestä ohjelmistokehitysprosessista sekä mielipiteitä nykyisistä suunnittelijoiden ja kehittäjien rooleista.



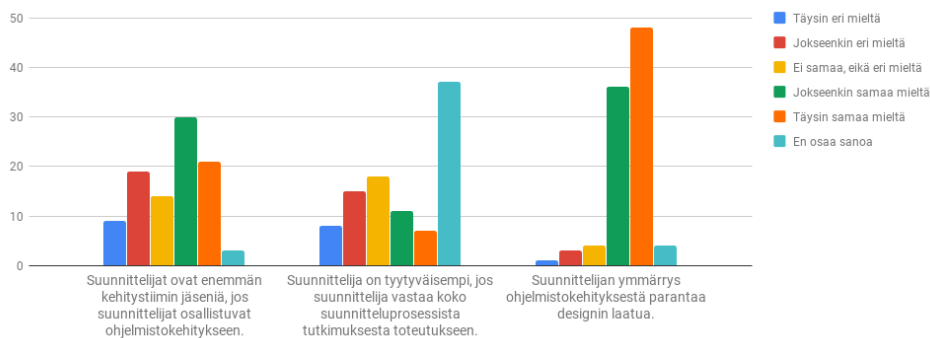
Kuva 15. Vastaajien mielipiteet suunnittelijoiden ja kehittäjien poikkitieteellisestä osaamisesta

Kuvassa 15 on esitetty vastaajien mielipiteet liittyen roolien poikkitieteellisyteen. Poikkitieteellisyydellä tarkoitetaan tässä sitä, että suunnittelija osallistuisi kehittäjien tehtävien tekemiseen (omien käyttöliittymäsuunnitelmien toteutus) ja kehittäjä osallistuisi suunnittelijoiden tehtävien tekemiseen (UX-tehtävien tekeminen). Kuvasta voidaan päätellä, että vastaajat pääosin kokevat erityisesti poikkitieteellisen tiedon olevan hyväksi kummallekin roolille (suunnittelija ja kehittäjä). Vastaajat kuitenkin kokevat tärkeämmäksi sen, että kehittäjät osallistuisivat UX-tehtävien tekemiseen sen sijaan, että suunnittelijat osallistuisivat kehittämiseen.



Kuva 16. Vastaajien mielipiteet väittämistä liittyen siihen, jos suunnittelija toteuttaa omat designinsa

Työntekijöitä pyydettiin seuraavaksi kuvittelemaan tilanne, jossa suunnittelija itse toteuttaa suunnittelemansa designin, ja kertomaan omat mielipiteensä tilannetta koskevista väittämistä. Vastaukset ovat näkyvillä kuvassa 16. Vastaajien mielipiteet siitä, että projektissa säästyisi tällaisessa tilanteessa aikaa tai rahaa, vaihtelivat suuresti. Designin vision siirtymisestä kehittäjille nopeammin sekä siitä, että kehitetty käyttöliittymä vastaisi suunnitteluspeksejä, oltiin enemmän samaa mieltä. Väite siitä, että kehittäjät voisivat keskittyä back-end-ohjelmointiin, jakoi paljon mielipiteitä. Enemmistö vastaajista kuitenkin koki, että kehittäjät eivät voisi keskittyä back-end-ohjelmointiin silloin, kun suunnittelija itse toteuttaa suunnittelemansa designin.



Kuva 17. Vastaajien mielipiteet suunnittelijan rooliin liittyvistä väittämistä

Viimeisissä nykytilannetta koskevissa väittämässä, työntekijöiltä kysyttiin mielipiteitä suunnittelijan rooliin liittyvistä väittämistä. Väittämät ja niihin annetut vastaukset on esitetty kuvassa 17. Molemmat ensimmäiset väittämät jakoivat mielipiteitä puolesta ja vastaan. Viimeisestä väittämästä ”suunnittelijan ymmärrys ohjelmistokehityksestä parantaa designin laatua” vastaajat olivat enimmäkseen jokseenkin tai täysin samaa mieltä.

6.4 Full-stack -suunnittelijan roolin tunnettavuus ja määritelmä

Kyselyssä esitettiin roolin määritelmäksi tässä työssä luvussa 4:

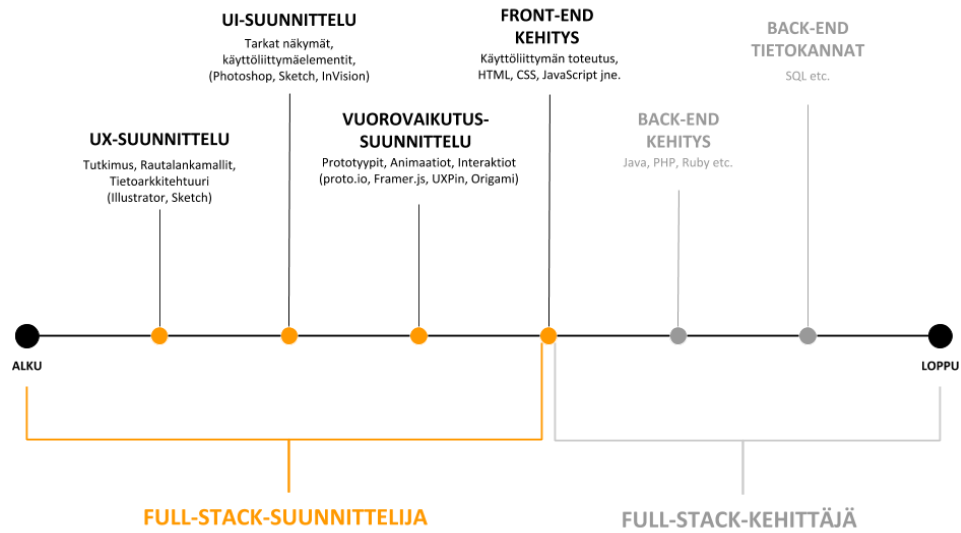
Full-stack -suunnittelijan määritelmä

Full-stack -suunnittelijalla tarkoitetaan sellaista henkilöä, jolla on osaamista kaikilta tuotesuunnittelun osa-alueilta. Full-stack -suunnittelija on siis suunnittelija, joka osaa tarvittaessa viedä itsenäisesti koko tuotesuunnitteluprosessin alusta loppuun, käyttäjätutkimuksesta designin toteutukseen. Roolin tehtävät/osaaminen on kuvattu tiivistetysti alla olevassa kuvassa.

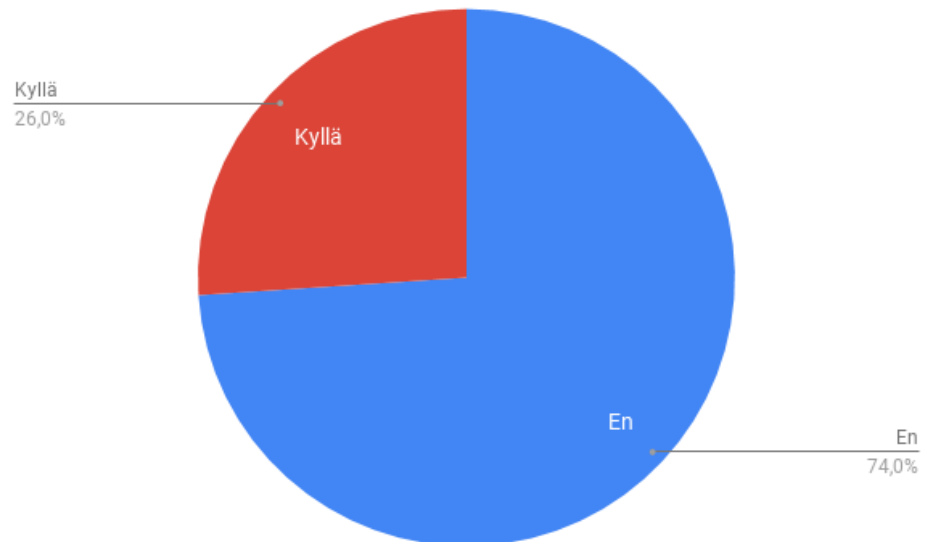
Kuvalla viitattiin jo luvussa 4 esitettyyn kuvaan 8. Alla on esitetty roolin määritelmän mukainen havainnollistava kuva uudelleen kuvassa 18.

Vastaajista vain 26,8% eli 26 vastaajaa oli aiemmin kuullut termin ”full-stack -suunnittelija”. Näiden vastaajien kesken mielipide annetusta määritelmästä roolille vaihteli. Kuten kuvasta 20 voidaan todeta, suurin osa vastaajista (76%) kuitenkin oli samaa mieltä roolin määritelmästä, 15,4% vastaajista oli eri mieltä ja loput, 7,7%, vastaajista eivät osanneet sanoa. En osaa sanoa -vastaukset ovat voineet johtua siitä, että vastaaja on kuullut aiemmin termin, mutta ei varsinaista termin määritelmää.

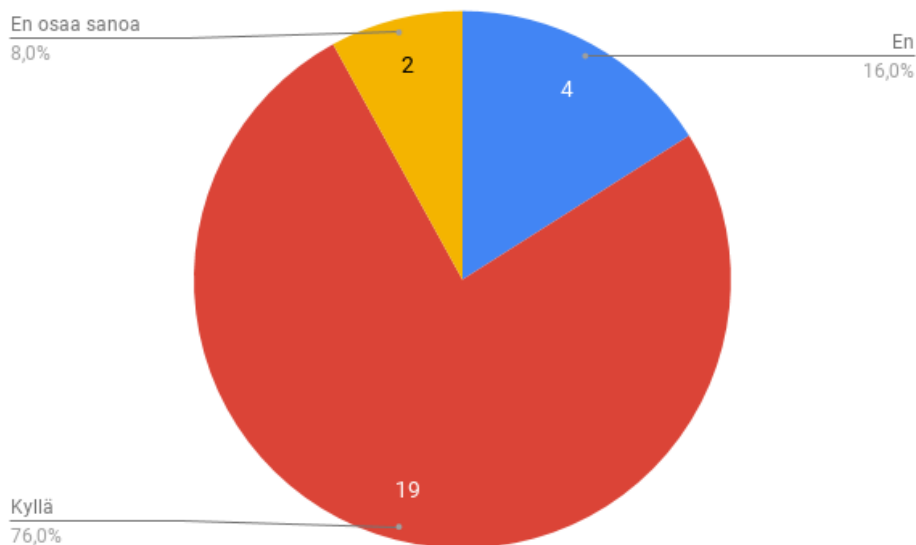
Määritelmä sai muutamia erilaisia korjausehdotuksia. Eräs kommentti oli esimerkiksi se, että full-stack -suunnittelijan tulisi ymmärtää ohjelmistokehitysprosessia kokonaisuudessaan – sekä front- että back-kehitystä – vaikkei full-stack -suunnittelijalla olisikaan kehitysvastuuta näiltä alueilta. Full-stack -suunnittelijaksi hyväksyttäisiin myös sellainen henkilö, jolla on kokemusta kaikista muista suunnittelun osa-alueista paitsi ohjelmoinnista.



Kuva 18. Full-stack -suunnittelijan tehtäviä tuotekehityksessä web-projektin mukaan esitettynä.



Kuva 19. Vastaukset kysymykseen: Olitko aiemmin kuullut termiä "full-stack -suunnittelija"(engl. full-stack designer)?



Kuva 20. Vastaukset kysymykseen: Oletko samaa mieltä roolin määritelmän kanssa?

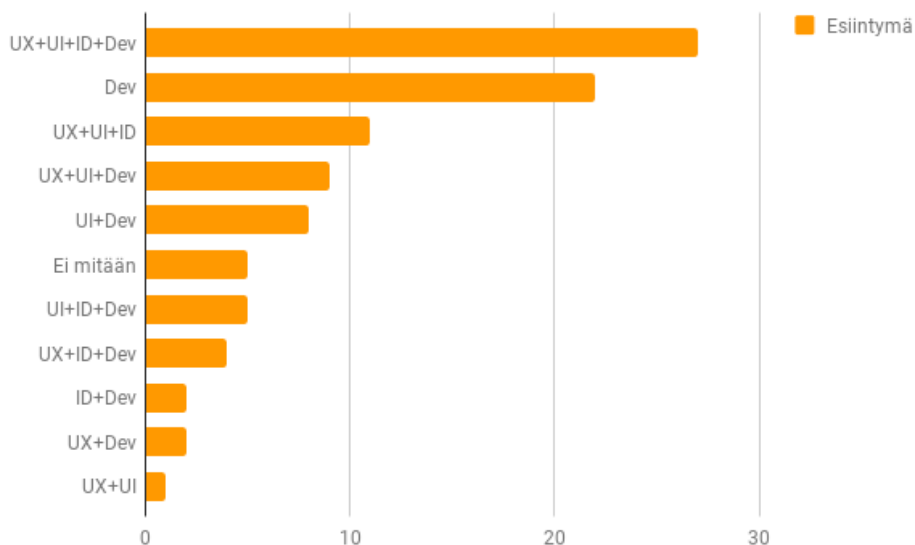
Kuitenkin koettiin myös, että full-stack -suunnittelija on sellainen henkilö, joka on lisäksi HTML/CSS-taitoinen.

Eräs vastaaja, joka oli kommentoinut olevansa samaa mieltä roolin määritelmän kanssa, kommentoi myös olevansa jokseenkin samaa mieltä, mutta koki, että kyselyssä annettu määritelmä oli työkalukeskeinen. Toinen vastaaja koki, että full-stack-termin jakaminen kahteen eri osaan (full-stack -suunnittelija ja full-stack-kehittäjä) ei ole järkevää, koska suunnittelijan tulisi joka tapauksessa osata kaikkea suunnittelun puolelta ja kehittäjän kehityksen puolelta. Full-stack tarkoittaisi tällöin koko tuotesuunnittelu ja -kehitysprosessia.

6.5 Full-stack -suunnittelijan roolin hyödyt, haasteet sekä ratkaisut haasteisiin

Ennen full-stack -suunnittelijan roolin hyötyjen ja haasteiden selvittämistä, työntekijöiltä kysyttiin kyselyssä, mitä suunnittelun tehtäviä he olivat tehneet samana projektin aikana. Kysymykseen tuli vastauksia 91. Vastauksista laskettiin eri yhdistelmien määrät ja nämä esiintymät on esitetty kuvassa 21. Tyhjät vastaukset otettiin mukaan laskuihin ”ei mitään” vaihtoehtona.

Kuvassa 21 termit tarkoittavat seuraavia: ”UX” tarkoittaa käyttäjäkokemussuunnittelua, ”UI” tarkoittaa käyttöliittymäsuunnittelua, ”ID” tarkoittaa vuorovaikutussuunnittelua ja ”DEV” tarkoittaa ohjelmistokehitystä. Yli neljäsosa (28,1%) koko kyselyyn vastaajista teki kaikkia kyselyssä mainittuja tehtäviä. Seuraavaksi suurin ryhmä oli pelkkä kehitys (22,9%). Kaikkia suunnittelun tehtäviä (UX-suunnittelu, UI-suunnittelu, Vuorovaikutussuunnittelu) teki 11,5% vastaajista. 5,2% vastaajista ei tehnyt mitään mainituista tehtävistä. Erilaisia suunnittelun tehtäviä ja kehitystä teki vajaa kolmasosa vastaajista (31,3%).



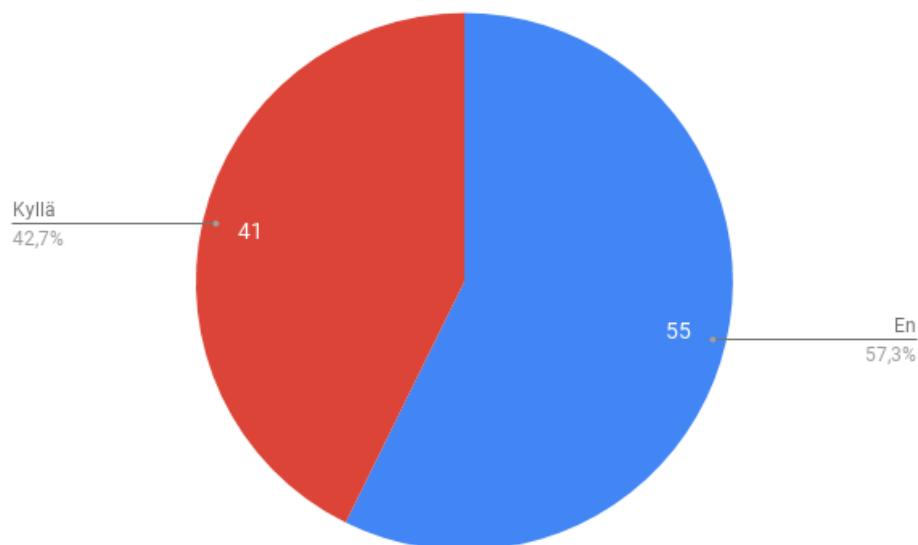
Kuva 21. Vastaajien määrät samassa projektissa suoritettujen tehtävien mukaan

Se, että suuri määrä vastaajista teki pelkästään kehitystehtäviä johtuu todennäköisemmin siitä, että suuri osa itse kyselyyn vastaajista oli kehittäjiä. Myös se, mitkä koetaan suunnittelutehtäviksi, voi vaihdella vastaajien mukaan. On myös mahdollista, että kehittäjät tekevät todellisuudessa enemmän suunnittelutehtäviä kuin mitä kyselystä kävi ilmi.

Yksikään vastaajista ei projekteissaan tehnyt pelkästään yhtä suunnittelun osa-aluetta (UX-suunnittelu, UI-suunnittelua tai vuorovaikutussuunnittelua). Myöskään sellaisia yhdistelmiä, joissa tehtäisiin pelkästään vuorovaikutussuunnittelua ja UI-suunnittelua tai vuorovaikutussuunnittelua ja UX-suunnittelua, ei tullut vastauksissa esiin yhtäkään.

Kyselyssä kysyttiin myös, olivatko työntekijät joskus työskennelleet projektissa, jossa joku tiimin jäsen on toiminut full-stack -suunnittelijan roolissa tai vastaavassa roolissa (kuva 22). Vastaukset jakautuivat lähes tasan, mutta suurempi osa vastaajista ei ollut työskennellyt full-stack -suunnittelijan roolissa toimivan tiimin jäsenen kanssa.

Vastaajien esittämiä kommentteja full-stack -suunnittelijan hyödyistä ja haasteista litteroitiin, jolloin poistettiin myös tutkimuksen kannalta epäolennaiset vastaukset. Lopuksi vastaajien kommentteista etsittiin yhteneviä teemoja eli vastaukset teemoiteltiin. Kommentit koskien haasteisiin liittyviä ratkaisuja käytiin läpi yhdessä haasteiden kanssa samojen teemojen säilyttämiseksi. Ratkaisut pyrittiin pitämään haasteiden kanssa samassa kontekstissa sen vuoksi, jotta kommentteissa ehdotetut ratkaisut kohdistuisivat juuri niihin haasteisiin, joihin vastaaja oli alunperin ratkaisunsa tarkoittanut. Alla olevissa luvuissa esitellään yleiskatsaus kyselyn kautta saatuihin hyötyihin, haasteisiin sekä niiden ratkaisuihin.



Kuva 22. Vastaajien vastaukset kysymykseen: ”Oletko joskus työskennellyt projektissa, jossa joku tiimin jäsen on toiminut full-stack -suunnittelijan roolissa tai tehnyt roolia vastaavia tehtäviä?”

6.5.1 Roolin hyödyt

Vastaajat olivat hyvin aktiivisia kertomaan oman mielipiteensä full-stack -suunnittelijan hyödyistä. Vastaajista yli puolet, 62,5%, vastasi tähän avoimeen kysymykseen. Kommentoiduista hyödyistä löydettyjen erilaisten toistuvien teemojen esiintyvyydet on esitetty taulukossa 5. Selkeästi suurimmiksi hyödyiksi koettiin realistiset designit (64,6% löydettyistä teemoista), ja tämän jälkeen saman vision säilyminen koko projektissa (8,9%), designin ja vision toisiaan vastaavuus (7,8%) sekä monipuolinen osaaminen (7,8%).

Vastaajien mielestä ohjelmistokehityssymmärrys auttaa full-stack -suunnittelijaa tuottamaan realistisia designeja. Vastaajat kokivat myös alunperin kehitetyn vision säilyvän koko projektin ajan, kun full-stack -suunnittelija vastaa sekä tuotteen suunnittelusta että kehityksestä. Vastaajista moni piti monipuolista osaamista hyvänä asiana. Monipuolinen osaaminen koettiin hyödyksi, koska tässä roolissa työskentelevä työntekijä voi osallistua moneen suunnittelun vaiheeseen. Tällainen henkilö työntekijöiden mukaan ymmärtää tuotesuunnittelua ja -kehitystä kokonaisvaltaisesti. Vastaajat myös arvelivat, että tuotteen kehitys nopeutuisi, suunnittelijan vision kommunikointi kehittäjille olisi helpompaa ja kehittäjien työ helpottuisi, mikäli full-stack -suunnittelijan roolissa toimiva henkilö suunnittelisi ja toteuttaisi itse designinsa. Roolin hyödyksi nähtiin myös se, että henkilö ymmärtää projektin kokonaisuudessaan. Tästä voi olla projektille hyötyä, sillä tällöin projektissa on joku, joka tietää ja ymmärtää koko projektin tilanteen.

Full-stack -suunnittelijan osaaminen koettiin parantavan myös lopputuotteen laatua. Kuten taulukossa 6 on esitetty, eräs vastaajista kommentoi työskennelleensä joskus full-stack -suunnittelijan kanssa, ja lopputuote oli tällöin erittäin viimeistelty. Lisäksi koettiin hyvä-

Taulukko 5. Hyötyjen esiintymät vastauksissa

Hyöty (teema)	Esiintymä
Realistisemmat designit	19
Saman vision säilyminen koko projektin ajan	8
Design ja visio vastaavat toisiaan	7
Moniosaaja	7
Tuotteen kehitys on nopeampaa	6
Projektin tunteminen kokonaisvaltaisesti	6
Vision kommunikointi helpompaa	6
Kehittäjien työn helpottuminen	5
Lopputuotteen parempi laatu	4
Rooli toimii hyvin pienissä projekteissa	4
Vähemmän kommunikaatio-ongelmia suunnittelijan ja kehittäjän välillä	4
Budjetissa pysyminen	2
Designilla on yksi omistaja	2
Parempi suunnittelijoiden ja kehittäjien yhteistyö	2
Pienempi tiimin koko	2
CSS- sekä saavutettavuus-asiantuntija	1
Ei mitään	1
Helpommat ketterän kehityksen iteraatiot	1
Responsiivisten näyttöjen suunnittelu	1
Rooli on lähempänä toteutusta ohjelmistoarkkitehdin rooliin verrattuna	1
Suunnitteluongelmien ratkaiseminen ohjelmoidessa	1

nä asiana se, että kehittäjät voisivat jatkaa suunnittelijan toteuttamasta käyttöliittymästä, sillä tällöin ei tarvitsisi tehdä samaa asiaa moneen kertaan. Erityisesti designin vision kommunikointi tällä tavoin koettiin hyvänä asiana niissä tilanteissa, joissa backend-kehitys on ulkoistettu. Kun full-stack -suunnittelija toteuttaa itse designinsa, työntekijöiden mukaan tämän hyötynä on se, että design vastaa juuri sitä visiota, joka suunnittelijalla oli. Toteutuksessa on tällöin mukana ne kaikki tärkeät elementit, jotka suunnittelija halusi pitää designissa. Myös suunnittelijan vision koetaan säilyvän koko projektin ajan, mikäli suunnittelija on mukana toteutuksessa. Visio ei myöskään muuttuisi toteutusvaiheessa ilman, että suunnittelija tietäisi tästä. Eräs esimerkki vastaajilta oli projekti, jossa samaa tyyliä halutaan viedä useisiin tuotteisiin samalla kertaa. Vision koettiin myös säilyvän yhtenäisenä, kun designin suunnittelee yksi henkilö.

6.5.2 Roolin haasteet

Vastaajat olivat aktiivisia myös kertomaan oman mielipiteensä full-stack -suunnittelijan haasteista. Vastaajista 64,6%, vastasi haasteita koskevaan avoimeen kysymykseen. Kommentoiduista haasteista löydettyjen, tämän työn kannalta tärkeimpien erilaisten toistuvien teemojen esiintyvyydet on esitetty taulukossa 7.

Taulukko 6. Vastaajien kommentteja full-stack -suunnittelijan roolin hyödyistä

Kommentteja full-stack -suunnittelijan roolin hyödyistä
Jos suunnittelijalla on osaamista käyttöliittymien toteutuksesta, UI:n suunnittelussa tulisi paremmin huomioida sen toteutuksen mahdolliset tekniset haasteet.
Helpon näköiset käyttöliittymät voivat olla todella monimutkaisia toteuttaa. Kun suunnittelija ymmärtää käytössä olevat teknologiat, suunnitelman mukaiseen lopputulokseen on helpompi päästä budjetin rajoissa.
Yhtenäinen visio säilyy alusta loppuun.
Suunnittelija pystyy siirtämään implisiittistä informaatiota interaktioista ja käyttöliittymästä toteutukseen helpommin. Suunnitteluartefaktit (rautalangat, designit jne.) eivät koskaan välitä kokonaista kuvaa jolloin suunnittelijan mukana olo käyttöliittymän koodauksessa paikkaa suunnitteluartefaktien puutoksia. Koodausvaiheessa voi myös ilmetä softan tuomia rajoitteita joita ei osattu ottaa aikaisemmin huomioon ja suunnittelijan ollessa koodauksessa mukana voidaan ongelmat ratkoa ad hoc. Suunnittelija pystyy siirtämään implisiittistä informaatiota interaktioista ja käyttöliittymästä toteutukseen helpommin.
Full-stack -suunnittelija pystyy osallistumaan hyvinkin moneen suunnittelun vaiheeseen, mikä on aina hyvä juttu. Suunnittelija voisi pitää huolen siitä, että kaikki käyttöliittymän suunnittelun osa-alueen otetaan huomioon, mikä nostaa lopputuloksen laatua huomattavasti. Etenkin jos suunnittelijalla on osaamista käyttöliittymien koodauksesta, hänen on paljon helpompi kommunikoida visionsa niille jotka lopulta kirjoittavat koodin. Usein toteutuksen ja suunnitelman väliset vivahde-erot voivat olla teknisesti suuntautuneelle ihmiselle näkymättömiä, joten jos suunnittelija osaa toteuttaa suunnitelmansa itse, hänen on helppo osallistua käyttöliittymän viimeistelyyn.
Helppo toteuttaa pieniä projekteja kokonaisuudessaan. Yksi henkilö olisi kartalla koko projektista kokonaisuudessaan. Designin siirtyminen toteutukseen jouhevasti.
Selvä tehtävän jako että on suunnittelija joka hoitaa suunnittelupuolen hommat ja ohjelmistokehittäjä voi paneutua funktionaaliseen toteutukseen.
En ole hyvä full stack suunnittelija, mutta olen työskennellyt erittäin taitavan full stack suunnittelijan kanssa ja kyllä se näkyi toistuvasti erittäin hyvin viimeistellyn lopputuotteen muodossa. Kuitenkin tässäkin tapauksessa full stack suunnittelija hallitsi paremmin ketjun UI suunnittelusta eteen päin.

Suurimmat haasteet full-stack -suunnittelijan roolissa koettiin olevan roolin generalistisuus (14,4% löydettyistä teemoista), laaja osaamisvaatimus (12,6%), ohjelmointiosaamiseen puutteellisuus (9%) sekä osaamisen ylläpitäminen (9%). Full-stack -suunnittelijan roolin koettiin olevan liian ”generalisti”, jolla vastaajat tarkoittivat sitä, että tällaisessa roolissa toimiva henkilö osaa paljon erilaisia asioita, mutta epäsopuuhaisesti. Vastaajien kokemus oli, että kaikkia roolin määritelmän mukaisia osa-alueita olisi vaikeaa osata yhtä hyvin. Roolin koettiin vaativan laajaa osaamista, mikä tekee roolista erittäin vaativan. Myös osaamisen ylläpitäminen oli tästä syystä vastaajien mielestä hankalaa erityisesti ohjelmistokehityksen puolella, sillä teknologiat kehittyvät ja muuttuvat kokoajan.

Myös ohjelmointiosaamisen puutteellisuus nousi useammassa vastaajien kommentteissa esiin. Ohjelmointiosaamisessa tuotiin esille ohjelmointitaitojen ja ohjelmointikielten käytön lisäksi se, että toimiva ohjelmointikoodi ei välttämättä takaa sitä, että se olisi tehty hyviä ohjelmointitapoja noudattaen. Lisäksi vastaajat arvelivat full-stack -suunnittelijalta kuluvaan ohjelmointiin myös huomattavan kauemmin aikaa kuin kehittäjältä, jonka pääasiallinen työ on ohjelmointi. Otanta vastaajien esittämiä kommentteja full-stack -suunnittelijan haasteista on esitetty taulukossa 8.

Taulukko 7. Haasteiden esiintymät vastauksissa

Haaste (teema)	Esiintymä
Generalisti	16
Laaja osaamisvaatimus	14
Ei tarpeeksi ohjelmointiosaamista	10
Osaamisen ylläpitäminen	10
Ajankäyttö	7
Toteutuksen huono laatu	7
Paljon vastuualueita	6
Muiden ihmisten näkökulmien puute	4
Vaikeus löytää rooliin sopivia henkilöitä	4
Kehitys hidastuu	3
Ristiriita	3
Yksin jääminen	3
Ei tarpeeksi ymmärrystä ohjelmistokehityksestä	2
Henkinen kuormitus	2
Ketterien menetelmien toteutumattomuus	2
Kontekstin vaihtaminen vähentää tehokkuutta	2
Rooliin kehittyminen vie kauan aikaa	2
Suunnittelijoita ei kiinnosta ohjelmointi	2
Ylityöllistyminen	2
Ei voi tehdä useampia projekteja päällekkäin	1
Full-stack-suunnittelijoita tarvitaan lukumäärällisesti enemmän	1
Itseä kiinnostamattomien asioiden tekeminen	1
Kommunikaation painopiste siirtyy enemmän frontendin ja backendin -rajapintaan	1
Liian valmiin näköiset suunnitelmat antavat asiakkaalle väärän kuvan toteutuksen tilanteesta	1
Sokeutuminen omalle työlle	1
Tiimityö kärsii	1
Vaikeus fokuoittaa olennaiseen	1
Yhden projektin työstäminen voi syödä motivaatiota	1

6.5.3 Ratkaisut haasteisiin

Edellisessä alaluvussa esitellyt vastauksissa toistuneet haasteet koodattiin, jotta näihin olisi helpompi viitata ratkaisuja läpikäydessä. Koodaamisella tarkoitetaan sitä, että jokaiselle havaintoyksikölle (tässä kohtaa haasteille) annetaan luokittelu tai merkkejä. Koodaukset ovat nähtävillä taulukossa 9.

Vastaajien aktiivisuus ratkaisujen ehdottamiseen ei ollut yhtä aktiivista kuin tutkitun roolin hyötyjen ja haasteiden kommentoinnissa kyselyssä. Kysymykseen ”Miten ratkaisisit mahdolliset haasteet?” vastasi yhteensä 49 vastaajaa (51% vastaajista). Vaikka vastauksia tähän kysymykseen tuli selkeästi vähemmän, oli määrä vielä varsin hyvä koko kyselyn vastaajien määrään verrattuna.

Taulukko 8. Vastaajien kommentteja full-stack -suunnittelijan roolin haasteista

Kommentteja full-stack -suunnittelijan roolin haasteista
Kuten fullstack-developerissakin, vaarana on että varsinainen asiantuntijuus kaikista osa-alueista on vaikea saavuttaa, ja vaikka teoriassa osaisikin pintapuolisesti toteuttaa kaikki määritellyt osa-alueet, jää syvälinen osaaminen vajavaiseksi joltain osin.
Kun tekee periaatteessa kolmen ihmisen työtehtäviä, mutta on käytössä vain yhden ihmisen aika. Välillä tuntuu, että on vain epämääräinen generalisti ilman suurempaa erityistaitoa. Joskus voisi haluta keskittyä tiettyihin asioihin enemmän.
Rooli vaatii niin montaa eri osaamista, että harvalla ihmisellä on aikaa oikeasti syventyä kaikkiin osa-alueisiin. Näin ollen lopputulos voi olla huonompi kuin silloin, jos projektille on riittävästi aikaa, hyvät yhteistyömahdollisuudet ja kaikkiin osaamisiin olisi tarjolla juuri jokaiseen osaamiseen erikoistunut henkilö.
Yksi suunnittelijan tehtävä on pitää käyttäjän puolta. Jos suunnittelija tekee myös toteutusta, syntyy eturistiriita: käyttäjän kannalta parempi ratkaisu on yleensä työläämpi toteuttaa. Toinen haaste on vaadittavan skillsetin laajuus: vaikeampi löytää yksittäistä ihmistä joka osaa näitä kaikkia hyvin.
Koodaustaidot. Vaikka osaa koodata käyttöliittymän toimivaksi, ei sitä tule välttämättä tehtyä parhaiden toteutustapojen mukaisesti tai se ei välttämättä tue jatkokehitystä tarpeeksi. Aikarajoitteet. Koodaus itsessään on jo hyvin aikaa vievää ja tutkimus + design + koodaus voi osoittautua projektista riippuen haasteelliseksi vetää kokonaisuutena läpi. Jokainen osa-alueista myös kehittyä ja vaatii opiskelua ollakseen tietoinen kunkin osa-alueen uusista menetelmistä.

Työn kannalta oleelliset vastaajien ehdottamat ratkaisut on esitetty taulukossa 10. Vastaajat korostivat ratkaisuisaan koulutuksen tarvetta erityisesti generalistiseen rooliin, ohjelmointiosaamisen puutteeseen, osaamisen ylläpitämiseen, ajankäyttöön sekä vaikeuteen löytää rooliin sopivia tekijöitä. Vastaajat kokivat, että full-stack -suunnittelijan tulisi keskittyä yhteen tehtävään (suunnittelun osa-alueeseen) sen sijaan, että full-stack -suunnittelija yrittäisi tehdä kaikkia rooliin kuuluvia tehtäviä. Tällä tavoin vastaajat ratkaisisivat haasteita liittyen jo mainittuihin generalistiseen rooliin, ohjelmointiosaamisen puutteeseen ja osaamisen ylläpitämiseen, mutta myös haasteita liittyen roolin laajaan osaamisvaatimukseen, muiden ihmisten näkökulmien puutteeseen sekä henkiseen kuormitukseen.

Lisäksi vastauksissa painotettiin työtehtävien delegoinnin tärkeyttä projektissa. Tätä ratkaisua ehdotettiin esimerkiksi henkisen kuormituksen vähentämiseksi, tuotteen laadun parantamiseksi sekä ohjelmistokehityksen nopeuttamiseksi. Ratkaisuksi ehdotettiin myös sitä, että projekteissa työskentelisi useampia henkilöitä full-stack -suunnittelijan roolissa. Sama määrä vastaajia (6 vastaajaa) myös koki, että rooli tulisi sijoittaa oikean kokoiseen projektiin eli mieluiten pienempään projektiin.

Myös perinteiset toimintatavat, kuten nykyisten suunnittelijan ja kehittäjän roolien säilyttäminen sekä suunnittelutaitojen opetus kehittäjille, saivat vastaajien keskuudessa kannatusta haasteiden ratkaisemiseksi. Muita otteita vastaajien kommentteista on esitetty taulukossa 11.

Taulukko 9. Haasteiden koodaukset

Koodaus	Haaste
A1	Generalisti
A2	Laaja osaamisvaatimus
A3	Ei tarpeeksi ohjelmointiosaamista
A4	Osaamisen ylläpitäminen
A5	Ajankäyttö
A6	Toteutuksen huono laatu
A7	Paljon vastuualueita
A8	Muiden ihmisten näkökulmien puute
A9	Vaikeus löytää rooliin sopivia tekijöitä
A10	Kehitys hidastuu
A11	Ristiriita
A12	Yksin jääminen
A13	Ei tarpeeksi ymmärrystä ohjelmistokehityksestä
A14	Henkinen kuormitus
A15	Ketterien menetelmien toteutumattomuus
A16	Kontekstin vaihtaminen vähentää tehokkuutta
A17	Rooliin kehittyminen vie kauan aikaa
A18	Suunnittelijoita ei kiinnosta ohjelmointi
A19	Ylityöllistyminen
A20	Asiakkaan tarpeiden ymmärtäminen rivien välistä
A21	Ei voi tehdä useampia projekteja päällekkäin
A22	Full-stack-suunnittelijoita tarvitaan lukumäärällisesti enemmän
A23	Itseä kiinnostamattomien asioiden tekeminen
A24	Kommunikaation painopiste siirtyy enemmän frontendin ja backendin rajapintaan
A25	Liian valmiin näköiset suunnitelmat antavat asiakkaalle väärän kuvan toteutuksen tilanteesta
A26	Sokeutuminen omalle työlle
A27	Tiimityö kärsii
A28	Vaikeus fokuoita olennaiseen
A29	Yhden projektin työstäminen voi syödä motivaatiota

6.6 Työntekijöiden kokemus roolin toimivuudesta yrityksessä

Suurin osa (87,6%) vastaajista koki, että full-stack -suunnittelija pystyisi toimimaan vastaajan työskentelemässä yrityksessä täysin roolin mukaisesti. Niissä yrityksissä, joissa tämä ei ollut mahdollista, syiksi kerrottiin seuraavia:

- Ei tarpeeksi resursseja UX-suunnittelun tutkimusvaiheeseen
- Pieni yritys, jossa kaikkien tulee toimia kehittäjän roolissa
- Suunnitteluun ja laadulliseen tutkimukseen menee paljon aikaa, joten ei voida olettaa, että suunnittelijat ehtisivät tekemään muuta
- Yritys ei pysty aina myymään sopivia projekteja
- Projektien diversiteetti laajaa (kuka tekee mitäkin: asiakas, joku muu, projektin tiimi jne.)

Taulukko 10. Ratkaisut haasteisiin ja niiden esiintymät vastauksissa

Ratkaisu	Esiintymä	Haasteet
Keskittyminen tiettyyn osa-alueeseen	7	A1, A2, A3, A4, A8, A14
Koulutus	7	A1, A3, A4, A5, A9, A13, A19
Tehtävien delegointi	6	A1, A4, A6, A10, A14
Useampia full-stack-suunnittelijoita projektissa	6	A2, A6, A7, A12, A15
Valitaan rooli oikean kokoiseen projektiin (pienempi projekti)	6	A2, A4, A5, A6, A16, A21
Erilliset suunnittelijan ja kehittäjän roolit	4	A3, A4, A11, A13
Suunnittelutaitojen opettaminen kehittäjille	4	A3, A4, A8, A13
Kokemuksen kerryttäminen	3	A1, A2, A3
Parikoodaus	3	A3, A4, A5
Rajataan osaamisvaatimuksia toteutuspuolella	3	A1, A2, A4
Aikaresurssointi	2	A1, A5
Kommunikaatio kehittäjien ja suunnittelijoiden välillä	2	A1, A8
Koodikatselmointi	2	A2, A3
Ohjelmointiosaamisen painottaminen rekrytointivaiheessa	2	A19, A22
Uusien teknologioiden oppiminen työajalla	2	A1, A4
Vaadittavien taitojen opettelu	2	A1, A2
Vähemmän ketteryyttä, enemmän inkrementaalisuutta	2	A1, A2
Allokoimalla järkevästi	1	A7
Markkinointi	1	A9
Ohjelmistokehityspäätöiden opetus asiakkaalle	1	A25
Ohjelmointi suunnittelun yhteydessä	1	A5
Oman mukavuusalueen ulkopuolelle meneminen	1	A1
Omien rajojensa hyväksyminen	1	A1
Erilaisten projektien tekeminen	1	A1
Suunnittelutehtävien jakaminen työntekijöiden mieltymysten mukaan	1	A18
Responsiivisuuden tiedostaminen	1	A3
Käyttöliittymäelementtikirjastojen käyttö toteutuksessa	1	A3
Suunnittelija mukana koko projektin ajan	1	A8
Tarpeeksi suunnittelijoita projektiin kokoon nähden	1	A 19
Tuki muilta samassa tai osittain samassa roolissa olevilta	1	A2
Yksi full-stack -suunnittelija projektissa	1	A7
Yksi suunnittelija ja eri osa-alueisiin keskittyneitä full-stack-suunnittelijoita	1	A15

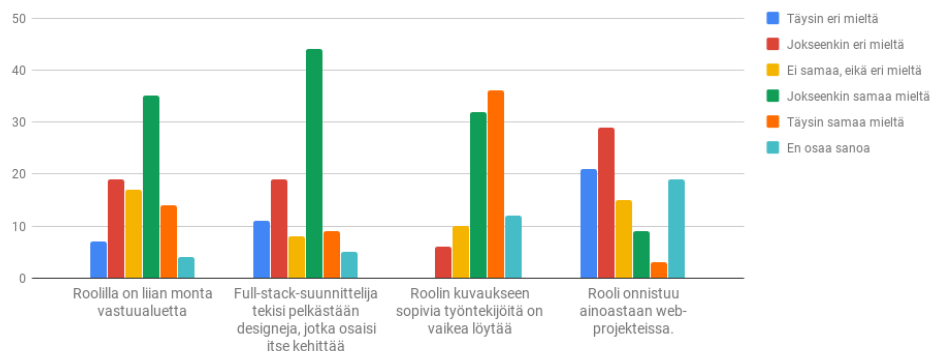
Taulukko 11. Vastaajien esittämiä ratkaisuja full-stack -suunnittelijan rooliin liittyviin haasteisiin

Ratkaisuja full-stack -suunnittelijan rooliin liittyviin haasteisiin
Jakaisin suunnittelutehtävät henkilöiden mieltymysten mukaan.
Hyvä kommunikointi on kaiken A ja O. Käyttökokemuksen miettiminen on keskeinen osa kaikkien palvelun kehittämisen parissa työskentelevien työnkuva. Sen langat ovat kuitenkin suunnittelijan käsissä, siinä missä esimerkiksi backend-kehityksen langat ovat backend-kehittäjien. Kaikkien on ymmärrettävä päällisin puolin, minkälaisia rajoitteita eri osa-alueet asettavat.
Erikoistuneet roolit, pitäisin designerin irti JS-teknologioista. HTML+CSS protot tuottavat lisäarvoa, sen jälkeen muotoilija tuottaa keskinkertaista koodia, jonka hyvä fronttikoodari tekisi itse nopeammin ja paremmin.
Itse käyttäisin projektissa erillistä suunnittelijaa, mutta toisaalta korostaisin devaajille perusteiden opettelun merkitystä tehokkaan työskentelyn edistämiseksi.
Toteuttaminen/ohjelmoiminen designin yhteydessä auttaisi todennäköisesti ajankäytössä.

- Suunnittelijat työskentelevät useissa projekteissa
- Suunnittelijat ja kehittäjät istuvat eri paikoissa
- Kehittäjien ennakkoluulot: ei haluta, että suunnittelija osallistuu kehitykseen

6.7 Työntekijöiden mielipiteet full-stack -suunnittelijasta

Kuvassa 23 on esitetty työntekijöiden vastaukset full-stack -suunnittelijan roolia koskevista väittämistä. Väite full-stack -suunnittelijan liian monesta vastuualueesta jakoi eniten mielipiteitä vastaajien kesken: yli puolet oli kuitenkin sitä mieltä, että roolilla on liian monta vastuualuetta. Suuri osa vastaajista oli myös yhtä mieltä väitteestä ”full-stack -suunnittelija tekisi pelkästään designeja, jotka osaisi itse kehittää”, mutta löytyi myös niitä vastaajia, jotka olivat väitteestä eri mieltä.



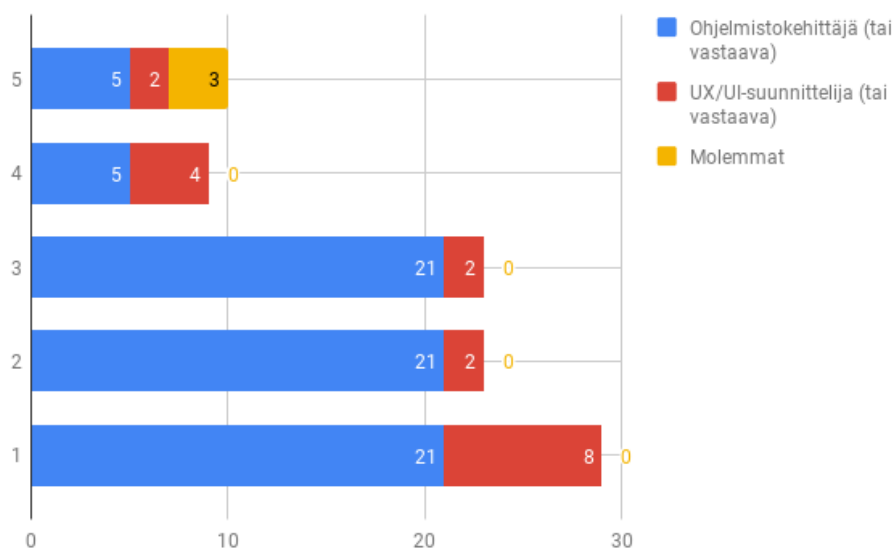
Kuva 23. Vastaajien mielipiteet full-stack -suunnittelijan rooliin liittyvistä väittämistä

Kuten oli odotettu, vastaajat olivat sitä mieltä, että full-stack -suunnittelijan roolin kuvaukseen sopivia ihmisiä on vaikea löytää. Vain muutama vastaajista oli tästä eri mieltä. Erityisesti vastaajien mielipide siitä, että rooli onnistuisi ainoastaan web-projekteissa,

erosi hyvin paljon siitä, mikä oli odotettu vastaus. Useammat muut verkkolähteet arvelivat roolin toimivan erityisesti web-projekteissa ja erittäin huonosti muissa projekteissa. Tässä selkeästi enemmistö oli sitä mieltä, että web-projektit eivät olisi ainut alue, jossa rooli voisi onnistua.

6.8 Työntekijöiden kiinnostus roolissa toimimiseen

Kuvassa 24 on esitetty suunnittelijoiden, kehittäjien ja molemmissa rooleissa toimivien vastaajien kiinnostus full-stack -suunnittelijan roolissa toimimiseen. Kuvassa kiinnostusasteikko on 1-5 siten, että vastaus numerolla 1 tarkoittaa vastaajan olevan ”ei ollenkaan kiinnostunut” ja vastaus numerolla 5 tarkoittaa vastaajan olevan ”erittäin kiinnostunut”. Kehittäjistä ainoastaan 13,7% oli erittäin kiinnostunut toimimaan full-stack -suunnittelijan roolissa. Suunnittelijoista kolmasosa, 33,3%, oli jonkin verran tai erittäin kiinnostunut toimimaan roolissa. Kaikki ne kolme vastaajaa, jotka määrittivät roolikseen sekä ohjelmistokehittäjän että suunnittelijan, vastasivat olevansa erittäin kiinnostuneita työskentelemään roolissa. Kaiken kaikkiaan vastaajien kesken kiinnostus roolissa työskentelyyn jakautui melko tasaisesti: 42,7% vastaajista olisi joksensikin tai erittäin kiinnostunut työskentelemään roolissa ja kolmasosa ei ollut juurikaan kiinnostunut.



Kuva 24. Suunnittelijoiden, kehittäjien sekä molemmissa rooleissa toimivien vastaajien kiinnostus full-stack -suunnittelijan rooliin

Kaiken kaikkiaan jopa 72 vastaajaa (75% vastaajista) perusteli, miksi he olisivat tai eivät olisi kiinnostuneita työskentelemään full-stack -suunnittelijan roolissa. Myös näissä perusteluissa oli havaittavissa selkeästi samankaltaisuuksia erityisesti roolista ja omasta työnkuvasta riippuen. Vastaajat, jotka olivat selkeästi kiinnostuneita (vastanneet numeron 4 tai 5 aiempaan kysymykseen kiinnostuksesta työskennellä roolissa) perustelivat kiinnostustaan roolissa toimimiseen kiinnostuksella ylipäätään suunnitteluun sekä ohjelmointiin (15,8 % kiinnostuneista ja 16,7% vastauksensa perustelleista kiinnostuneista vastaajista),

sopivalla osaamisella (10,5 % / 11,1%) sekä sillä, että haluaa hallita kokonaisuutta (21,1 % / 22,2%).

Niiden vastaajien, jotka eivät olleet kiinnostuneita roolissa toimimisesta, vastaukset riippuivat siitä, missä roolissa he työskentelivät. Niistä perustelleista, jotka eivät olleet kiinnostuneet roolista, suunnittelijoita oli 9 ja kehittäjiä 29. Suunnittelijat eivät olleet kiinnostuneita ohjelmoinnista (22,2% perustelleista, jotka eivät olleet kiinnostuneita), eivät halunneet ottaa niin suurta rooliin kuuluvaa tehtäväkokonaisuutta hallittavakseen (22,2%) ja kokivat, että full-stack -suunnittelijan roolissa on liian suuri vastuu yhdelle ihmiselle (33,3%). Kehittäjät sen sijaan eivät joko olleet kiinnostuneita suunnitelutehtävistä (34,5% kysymykseen vastanneista kehittäjistä, jotka eivät olleet kiinnostuneita) tai kokivat, että heillä ei ole tarpeeksi osaamista suunnittelun puolella (27,6%). Myös nykyinen rooli ohjelmistokehittäjänä nousi perusteluissa useasti esille (24,1%). Ne vastaajat, jotka kokivat olevansa jossain siltä väliltä (vastanneet numeron 3), kommentoivat saman tyyppisiä asioita kuin sekä kiinnostuneet että ne, jotka eivät olleet erityisen kiinnostuneita roolissa työskentelystä.

6.9 Muuta kommentoitavaa full-stack -suunnittelijan roolista

Vastaajat kommentoivat full-stack -suunnittelijan roolia esimerkiksi sen termistä, rooliin kohdistuvista ennakoasenteista sekä roolin vaikutuksesta projektissa kehitettävään tuotteeseen. Kommenttien joukossa oli sekä positiivisia että negatiivisävytteisiä kommentteja, mutta myös neutraaleja, pohdiskeluvia kommentteja.

Termi ”full-stack -suunnittelija” koettiin hölmöksi ja siihen koettiin liittyvän liikaa tehtäviä. Rooli koettiin aliarvostetuksi, koska roolin määritelmän mukaisia ominaisuuksia kaivattaisiin suunnittelijalta, mutta osa koki roolin myös turhaksi, sillä koettiin, että roolin tehtävät saataisiin hoidettua paremmin useamman työntekijän kesken. Eräs kommentoija halusi painottaa, että roolin kautta saataisiin laadukkaampia, muttei loppukäyttäjän kannalta arvokkaampia tuotteita.

6.10 Palaute kyselystä

Pääosin palaute kyselystä oli positiivista. Palaute kohdistui kyselyn rakenteeseen sekä aihepiiriin. Vastaajat kommentoivat aiheen olevan mielenkiintoinen sekä ajatuksia herättävä. Myös käsitteiden kuvaaminen ja kaaviot saivat positiivista kommenttia.

Kyselyyn vastaaminen koettiin pääosin helpoksi ja nopeaksi. Muutamat palautetta antaneet vastaajat kokivat joidenkin kysymysten olleen huonosti muotoiltuja. Eräs kommentoija koki suunnitteluun ja kehitykseen liittyvät väittämät siten, että haluttiin tiedustella asioiden tärkeyttä niiden oikeellisuuden sijaan. Tämä otettiin huomioon vastauksia analysoitaessa, sillä väittämät olivat tarkoitettu juuri oikeellisuuden mittaamiseen. Toinen kommentoija koki vastaavansa kyselyssä siten, jotta ei tarvitsisi perustella omaa vastaustaan. Perustelut olivat kyselyssä kuitenkin vapaaehtoisia, vaikka vastaajia yritettiinkin

kannustaa vastaamaan myös perusteluihin. Kolmas kommentoija olisi toivonut full-stack -suunnittelijan määritelmän heti kyselyn alkuun. Määritelmä esiteltiin vasta myöhemmin kyselyssä, jotta määritelmä ei ohjaisi liikaa vastaajaa kyselyn alussa esitetyissä nykyiseen suunnittelun ja ohjelmistokehityksen tilaan liittyvissä kysymyksissä.

7. TULOSTEN TARKASTELU

Tämän luvun tarkoituksena on verrata tämän työn yhteydessä tehtyä tutkimusta aiempiin tutkimuksiin, joita käsiteltiin tämän diplomityödokumentin alussa, sekä vastata tutkimuskysymyksiin. Ensin tässä luvussa vertaillaan tehtyä tutkimusta muihin tutkimuksiin ja pohditaan tarkemmin edellisessä luvussa 6 esitetyjä tutkimustuloksia. Sen jälkeen vastataan tutkimuskysymyksiin, joita oli työssä asetettu kolme:

1. Mitä hyötyjä ja haasteita full-stack -suunnittelijan roolista on yritykselle?
2. Mitkä ovat työntekijöiden näkemykset/kokemukset full-stack -suunnittelijan roolin hyödyistä ja haasteista?
3. Miten full-stack -suunnittelijan roolin haasteet voidaan ratkaista?

Tutkimuskysymyksiin vastaamisen jälkeen tarkastellaan vielä tutkimuksen luotettavuutta.

7.1 Tutkimustulosten vertailu aiempiin tutkimuksiin

Työssä nousi esiin useampia vastaavuuksia aiempiin tutkimuksiin, joissa oli tutkittu käyttäjäkokemussuunnittelijoiden roolia ohjelmistokehitysprojekteissa. Esimerkiksi Kuusisen [24] tekemä huomio siitä, että projektissa syntyy odotteluaikaa, jos yksi ihminen vastuussa suuresta työmäärästä, nousi esiin myös kyselyyn tulleissa vastauksissa. Tämä nostettiin vastauksissa esille erityisesti full-stack -suunnittelijan rooliin liittyvissä haasteissa. Kyselyssä tuli lisäksi useasti esille se, että kyselyyn vastanneiden mielestä rooli vastaa liikaa generalistia. Roolissa toimivalla ajateltiin siis olevan vähän ymmärrystä kaikista osa-alueista, mutta ei erittäin hyvää osaamista mistään, tai ainakaan useammasta osa-alueesta. Tämän koettiin myös vaikuttavan kehitettävien tuotteiden laatuun negatiivisesti. Myös Ibragimova et al. [21] nostivat tämän asian esille omassa tutkimuksessaan. Heidän mukaansa yhden ihmisen on vaikeaa suorittaa tehtäviä erittäin hyvin jokaisella osa-alueella.

Kuusisen [22] mukaan tilanteet, joissa kehittäjät tekevät muutoksia käyttöliittymäsuunnitelmaan ilmoittamatta siitä käyttäjäkokemussuunnittelijalle, aiheuttavat myös käyttäjäkokemussuunnittelijan tekemiin konsepteihin muutoksia. Kuusinen samassa tutkimuksessaan myös mainitsee näistä tilanteista sen, että tilanteet tällöin heikentävät lisäksi kehitettävän tuotteen laatua. Samaa asiaa kommentoivat myös jotkut kyselyyn vastanneista: he kokivat, että käyttöliittymäsuunnitelma ja visio säilyvät yhtenäisempinä, mikäli koko käyttäjäkokemussuunnitteluprosessi on full-stack -suunnittelijan käsissä. Ohjelmointiosaaminen ja oman käyttöliittymäsuunnitelmansa toteuttaminen koettiin tästä syystä positiivisena asiana. Tämän koettiin vaikuttavan positiivisesti kehitettävän tuotteen laatuun.

Tästä voidaan päätellä, että etenkin vision säilyttäminen tuotteen laadun kannalta on tärkeää.

Ibragimova et al. [21] tekemässä tutkimuksessa selvisi, että käyttäjäkokemussuunnittelijan monipuolisuus koettiin pienissä projekteissa hyödylliseksi, sillä se yksinkertaistaa prosessia ja mahdollistaa tiiviimmän kommunikaation kehittäjien kanssa. Kyselyn kautta saadut vastaukset ja erityisesti pienissä yrityksissä työskennelleiden työntekijöiden kokemukset tukevat tätä väitettä. Canziba [6] mainitsi kirjassaan, että yksi full-stack-suunnittelijan roolin eduista on mahdollisuus suunnitella toteutettavissa olevia käyttöliittymäsuunnitelmia. Tämä nousi myös tämän työn tutkimuksessa eniten esille full-stack-suunnittelijaan liittyvissä hyödyissä. Kyselyyn vastanneet kommentoivat roolissa työskentelevän henkilön pystyvän tekemään realistisia käyttöliittymäsuunnitelmia. Standardissa SFS-ISO 9241-210 [35] mainitaan, että realistisista suunnitteluratkaisuista saadaan merkittävää hyödytä, jos realistisuuden taso ja yksityiskohdat ovat tarkoituksenmukaisia. Tosin, jos näihin investoidaan liian paljon aikaa ja rahaa prototyypin tuottamiseksi, voi syntyä haluttomuutta suunnitteluratkaisujen muuttamiseen. Tämä sama asian käänttöpuoli kävi ilmi myös tämän diplomityön kyselyn vastauksissa. Työntekijät kommentoivat roolista syntyvän mahdollisen ristiriidan, jolloin käyttäjäkokemussuunnittelija ei toimi enää käyttäjien parhaaksi vaan ajattelee myös omaa etuaan. Esimerkiksi tilanteissa, jossa käyttäjäkokemussuunnittelija tietää, että muutokset aiheuttavat hänelle itselleen lisätehtäviä, erityisesti vaikeampien tehtävien kohdalla, saattaa käyttäjäkokemussuunnittelija yrittää olla tekemättä muutoksia.

Sekä Ibragimova et al. [21] että Kuusinen [22, 24] esittävät, että käyttöliittymäsuunnitelman siirtyminen kehittäjille nopeutuu, mikäli käyttöliittymäsuunnitelma kommunikoidaan toimivana ohjelmistona. Suuri osa kyselyyn vastanneista oli tämän väitteen kanssa samaa mieltä. Myös muutama vastaaja oli eri mieltä. Tähän on saattanut vaikuttaa esimerkiksi vastaajan omat ennakkoluulot siihen, kuinka hyvin vastaaja kokee käyttäjäkokemussuunnittelijan pystyvän toteuttamaan käyttöliittymäsuunnitelmansa, jotta visio tulisi toteutuksesta selkeästi esille.

Työn tutkimustuloksissa oli myös eroja aiemmin toteutettuihin tutkimuksiin. Ibragimova et al. [21] huomasivat omassa tutkimuksessaan, että käyttäjäkokemussuunnittelija, joka vastasi sekä tutkimustyöstä, että käyttäjäkokemus- ja käyttöliittymäsuunnittelusta oli tyytyväisempi työhönsä kuin muut käyttäjäkokemussuunnittelijat. Tämä johtui siitä, että käyttäjäkokemussuunnittelija on sitä tyytyväisempi, mitä enemmän koko suunnitteluprosessi on käyttäjäkokemussuunnittelijan hallinnassa. Tässä tutkimuksessa tästä väitteestä oltiin keskimäärin enemmän eri mieltä. Iso osa vastaajista ei myöskään osannut sanoa omaa mielipidettään asiaan, joten on mahdollista, että vastaajilla ei myöskään ole ollut tarpeeksi kokemusta asiasta. Väitteen todistamiseen tarvittaisiin lisää empiiristä tutkimusta ja näyttöä todellisista tilanteista. Lisäksi asia on voitu ymmärtää eri tavalla tai kyselyyn vastaajat ovat kokeneet, ettei asia ole niin yksiselitteinen kuin mitä esimerkiksi Ibragimova et al. antavat ymmärtää. Sekä Canziba [6] ja Esteves et al. [12] esittävät, että

prosessista myös säästyy aikaa ja rahaa, mikäli käyttäjäkokemussuunnittelija toteuttaa itse omat käyttöliittymäsuunnitelmansa. He sanovat tämän johtuvan siitä, että kehitystiimi pystyy keskittymään back-end-ohjelmointiin ja näin ollen front-end-ohjelmoinnin puolella ei tule yhtä paljon päällekkäistä työtä. Tämän työn tutkimuksessa asiaa ei koettu yhtä suoraviivaisena. Sekä ajan ja rahan säästymisessä sekä back-end-ohjelmointiin keskittymisestä mielipiteet vastaajien välillä vaihtelivat. Väitteet ovat voitu kokea samalla tavalla kuin käyttöliittymäsuunnitelman kommunikointi toimivana ohjelmiston: vastaajan mielessä käyttäjäkokemussuunnittelijan ohjelmointitaidot eivät välttämättä ole sellaisella tasolla, joka mahdollistaisi käyttöliittymäsuunnitelman sujuvan toteuttamisen, jolloin kehittäjät joutuvat avustamaan käyttäjäkokemussuunnittelijaa front-endin ohjelmoimisessa.

Kuten verkossa käydyistä keskusteluista sekä tässä tutkimuksessa kävi ilmi, on full-stack-suunnittelijan rooli usealle ihmiselle yhä vieras käsite. Myöskään niille, jotka roolista olivat aiemmin kuulleet, roolin määritelmä ei ollut vakiintunut ja selkeä. Erityisesti se, kuinka paljon full-stack-suunnittelijan odotetaan osaavan ohjelmistokehityksen puolella, jakaa mielipiteitä. Termi ”full-stack-suunnittelija” on saattanut olla joillekin vastaajille hämäävä. Joistain kyselyn avoimiin kysymyksiin vastanneiden työntekijöiden vastauksista voidaan tulkita, että vastaaja kokee full-stack-suunnittelijan olevan yhtä perinteisen käyttäjäkokemussuunnittelijan kanssa. Full-stack-suunnittelijan koettiin joissain vastauksissa olevan jopa joku, joka siirtyy käyttäjäkokemussuunnittelijan roolista full-stack-suunnittelijaksi, mikä tällöin vaatii käyttäjäkokemussuunnittelijalta ylimääräistä opetusta ohjelmistokehityksen puolelta.

Kuitenkin, mikäli roolia tulkittaisiin tasavertaisena roolina muihinkin ohjelmistokehityksen rooleihin (käyttäjäkokemussuunnittelija, kehittäjä, full-stack-kehittäjä), voitaisiin olettaa, että tällaisessa roolissa toimiva henkilö jo osaisi molempia osa-alueita, suunnittelua sekä kehitystä, sekä olisi jo kiinnostunut roolissa toimimisesta. Tällöin rooliin opettelu tai osaaminen eivät olisi ongelmia. Eräs vastaajista toikin full-stack-suunnittelijan hyötyihin liittyvässä kommentissaan esille, että roolissa voisi hyvin toimia joku, jolla on kokemusta ohjelmistokehityksestä ja on myöhemmin saavuttanut kokemusta myös suunnittelun puolelta. Käyttäjäkokemussuunnittelija terminä ei välttämättä ole joillekin kehittäjille houkuttava. Tämä saattaa johtua nykyisestä tarkasta rajauksesta käyttäjäkokemussuunnittelijoiden ja kehittäjien tehtävien välillä projekteissa.

Yksi tutkimuksessa esiin tullut havainto oli rooliin liittyvissä ja projekteissa samaan aikaan toteutettaviin tehtäviin liittyvien tulosten välillä. Kyselyssä full-stack-suunnittelijaan liittyvissä haasteissa sekä väittämässä tuli esille, että vastaajat olivat sitä mieltä, että full-stack-suunnittelijan roolin kuvaukseen sopivia henkilöitä on vaikeaa löytää (edellisen luvun taulukko 7 ja kuva 23). Kuitenkin kyselyssä kuvassa 21 oli nähtävissä, että vastaajista moni itse toimii täysin tai osittain full-stack-suunnittelijan roolissa. Täysin roolin mukaisia tehtäviä teki neljäsosa vastaajista ja ylipäätään poikkitieteellisiä tehtäviä tekivät jopa lähes 60% vastaajista. Toinen tulos, jota tutkimuksessa ei osattu odottaa, oli se, että rooli koettiin olevan mahdollinen - mahdollisesti jopa helpompi - toteuttaa muissakin pro-

jekteissa kuin web-sovellus-projekteissa. Tällaisia olivat vastaajien mielestä esimerkiksi työpöytäsovellusprojektit.

7.2 Full-stack -suunnittelijan hyödyt ja haasteet yritykselle

Pressman 2001 [29] jakaa ohjelmistoprojektinhallinnan seuraaviin tekijöihin: ihmiset, tuote, prosessi ja projekti. Ihmisiin sisältyvät projektitiimi ja sidosryhmät, tuotteella viitataan laajuuteen ja ratkaistavan ongelman erittelyyn ja prosessiin liittyvät vaiheet sekä tehtävät. Chow et al. jakavat ohjelmistoprojektinhallintaa myös Pressmanin tapaa vastaavalla tavalla, mutta erittelevät erikseen epäonnistumistekijät ja onnistumistekijät. Chow et al. jakavat epäonnistumistekijät neljään kategoriaan ja onnistumistekijät viiteen kategoriaan. Epäonnistumistekijöitä ovat organisaatio, ihmiset, prosessi sekä tekninen, ja onnistumistekijöitä ovat edellä mainittujen tekijöiden lisäksi projekti (engl. project). Organisaation alle liittyvät johtaminen ja työyhteisön kulttuuri, ihmisten alle liittyvät tiimi ja sidosryhmät, prosessin alle liittyvät projektinhallinta ja ketterän kehityksen menetelmät. Lisäksi Chow et al. määrittelevät tuloksen onnistumisen mittareiksi laadun, soveltamisalan (engl. scope), ajan sekä kustannukset. [8]

Brhel et al. [5] tutkivat projektin onnistumista suunnittelun näkökulmasta. He jakavat onnistumistekijät myös edellä mainittuja vastaaviin kategorioihin: prosessi, menetelmät (engl. practices), ihmiset / sosiaalisuus sekä tekniikka. Prosessiin sisältyvät ohjelmistokehityksen organisointiin liittyviä asioita, kuten ketterä kehitys ja sprintit ja suunnittelun integrointi kehitykseen. Menetelmiin sisältyvät suunnittelun eri vaiheet ja ihmisiin / sosiaalisuuteen esimerkiksi kehittäjien ja käyttäjäkokemussuunnittelijoiden yhteistyö.

Taulukko 12. Yhteenveto tarkasteltavista projektin onnistumisen ulottuvuuksista

Ulottuvuus	Tekijät
Prosessi	Organisointi ja suunnittelun integrointi kehitykseen
Ihmiset / sosiaalisuus	Yhteistyö, kommunikaatio, roolit ja tiedonvaihto
Projekti	Projekti, tuote ja sen onnistumisen tekijät: laatu, soveltamisala, aika ja kustannukset

Tässä työssä tehdyssä tutkimuksessa tutkitut teemat kohdistuivat ihmisiin resursseina, projektiin sekä prosessiin sen osalta, miten suunnittelua integroidaan kehitykseen. Hyödyntäen edellä mainittuja tekijöitä sekä tässä työssä tehdyn tutkimuksen teemoja, tässä työssä tarkastellaan full-stack -suunnittelijan hyötyjä ja haasteita yritykselle seuraavien ulottuvuuksien kautta: prosessi, ihmiset/sosiaalisuus sekä projekti. Nämä kategoriat on esitetty vielä selkeyden vuoksi 12. Seuraavissa alaluvuissa tarkastellaan aiempia tutkimuksia sekä tässä työssä saatuja tutkimustuloksia yrityksen näkökulmasta hyödyntäen edellä mainittuja ulottuvuuksia.

7.2.1 Prosessiin, ihmisiin sekä projekteihin liittyvät hyödyt

Työntekijöiden kiinnostus rooliin liittyvien osa-alueiden tehtävistä

Tutkimuksessa selvisi, että työntekijöiden kiinnostus full-stack -suunnittelijan rooliin liittyvien eri osa-alueiden tehtäviin oli vaihtelevaa, mutta jokaiseen osa-alueeseen löytyi niistä kiinnostuneita työntekijöitä. Tuloksista oli nähtävissä, että työntekijöiden kokemus ja kiinnostus tiettyihin tehtäviin nousi samassa suhteessa. Tulos on vielä tulkinnanvarainen. Suuri kiinnostus ylipäättään tiettyyn tehtävään on voinut vaikuttaa siihen, että työntekijä on halunnut tehdä näitä mainittuja tehtäviä pidemmän aikaa ja kiinnostus niihin on säilynyt. On toki myös mahdollista, että työntekijän tehdessä näitä tehtäviä kiinnostus on kasvanut sitä mukaa kuin tekemisen kautta on saatu lisää osaamista ja varmuutta tehtävien suorittamiseen. Kehittäjien ohella myös käyttäjäkokemussuunnittelijat ovat kiinnostuneita toteuttamaan omia käyttöliittymäsuunnitelmiaan. Käyttöliittymien toteutus ei siis ole yksinomaan kehittäjien mielenkiinnon kohde.

Työntekijöiden kokemukset poikkitieteellisyydestä ja sen hyödyllisyydestä projekteissa olivat pääosin positiivisia. Full-stack -suunnittelijan poikkitieteellisyydestä saatavia konkreettisia hyötyjä ovat esimerkiksi kehittäjien ja käyttäjäkokemussuunnittelijoiden välisen kommunikaation paraneminen, kuten myös Ibragimova et al. [21] tutkimuksessaan mainitsee. Koska full-stack -suunnittelija osallistuisi käyttöliittymien kehitykseen, hänet koettaisiin myös enemmän kehitystiimin jäsenenä.

Työntekijöiden kokemus rooliin liittyvien osa-alueiden tehtävistä nykyisissä projekteissa

Moni kyselyyn vastanneista työntekijöistä on projekteissaan työskennellyt täysin full-stack -suunnittelijaa vastaavissa tehtävissä tai lähes vastaavissa tehtävissä. Full-stack -suunnittelijana toimiminen yrityksissä ei siis ole mahdottomuus, sillä roolia vastaavissa tehtävissä toimitaan jo. Niiden vastaajien kesken, jotka toimivat projekteissaan täysin roolin mukaisesti, oli huomattavissa, että roolin osa-alueiden tehtäviin löytyi myös kiinnostusta.

Roolin onnistuminen yrityksessä

Lähes 90% kyselyyn vastanneista oli sitä mieltä, että rooli onnistuisi heidän työskentelemisensä yrityksissä. Mikäli yrityksellä on jo panostettu käyttäjäkokemussuunnitteluun ja -prosessiin, roolissa työskentelevällä on hyvät edellytykset toimia yrityksessä hyödyntäen täyden potentiaalinsa. Rooli toimii erityisesti pienissä yrityksissä sekä pienissä projekteissa [21], kuten myöhemmin kohdassa *Projektityöskentelyn tehostuminen* voidaan todeta. Tässä tutkimuksessa myös selvisi, että roolin yleisimmästä määritelmästä [6, 10] poiketen, roolin uskotaan onnistuvan myös muissa projekteissa kuin web-projekteissa. Osa vastaajista oli jopa sitä mieltä, että rooli onnistuisi paremmin muissa projekteissa (kuten työpöytäsovelluksissa ja mobiilisovelluksissa) kuin web-projekteissa, joissa käytetyt teknologiat päivittyvät vastaajien mukaan ahkerammin.

Lopputuotteen parempi laatu

Kuusisen [22] mukaan tilanteet, joissa kehittäjät tekevät muutoksia käyttöliittymäsuunnitelmaan ilmoittamatta siitä käyttäjäkokemussuunnittelijalle, aiheuttavat myös käyttäjäkokemussuunnittelijan tekemiin konsepteihin muutoksia. Kuusinen samassa tutkimuksessaan myös mainitsee näistä tilanteista sen, että tilanteet tällöin heikentävät lisäksi kehitettävän tuotteen laatua. Samaa asiaa kommentoivat myös jotkut kyselyyn vastanneista: he kokivat, että käyttöliittymäsuunnitelma ja visio säilyvät yhtenäisempinä, mikäli koko suunnitteluprosessi on full-stack -suunnittelijan käsissä. Ohjelmointiosaaminen ja oman käyttöliittymäsuunnitelmansa toteuttaminen koettiin tästä syystä positiivisena asiana. Tämän koettiin vaikuttavan positiivisesti kehitettävän tuotteen laatuun. Tästä voidaan päätellä, että etenkin vision säilyttäminen tuotteen laadun kannalta on tärkeää.

Projektityöskentelyn tehostuminen

Ibragimova et al. [21] tekemässä tutkimuksessa selvisi, että käyttäjäkokemussuunnittelijan monipuolisuus koettiin pienissä projekteissa hyödylliseksi, sillä se yksinkertaistaa prosessia ja mahdollistaa tiiviimmän kommunikaation kehittäjien kanssa. Tässä tutkimuksessa erityisesti pienissä yrityksissä työskennelleet full-stack -suunnittelijaan verrattavissa olevat vastaajat kommentoivat, että heidän osaamisestaan on ollut hyötyä pienissä projekteissa. Sekä Ibragimova et al. [21] että Kuusinen [22, 24] esittävät, että käyttöliittymäsuunnitelman siirtyminen kehittäjille nopeutuu, mikäli design kommunikoidaan toimivana ohjelmistona.

7.2.2 Prosessiin, ihmisiin sekä projekteihin liittyvät haasteet

Roolin onnistuminen saattaa vaatia enemmän resursseja yritykseltä

Roolin onnistuminen vaatii yritykseltä jo olemassa olevia suunnitteluprosesseja. Tämä voi erityisesti pieneille yrityksille olla haaste. Myös se, miten lähellä kehittäjät ja käyttäjäkokemussuunnittelijat työskentelevät voiva vaikuttaa. Esimerkiksi se, kumpaan tiimiin full-stack -suunnittelija laitetaan, ei välttämättä ole niin ilmiselvää yrityksissä. Full-stack -suunnittelijan roolissa työskentelevän ajankäyttö voi olla erilaisempi kuin käyttäjäkokemussuunnittelijan tai kehittäjän ajankäyttö. Roolissa työskentelevälle tulisi siis varata tarpeeksi aikaa projektissa työskentelyyn. Myös täysin sopivien projektien löytäminen roolille voi osoittautua yritykselle haastaviksi, joten full-stack -suunnittelija voi joutua joustamaan sen suhteen, mitä kaikkea pääsee ja pystyy projekteissa tekemään.

Roolin termi, määritelmä ja sen tunnettavuus

Full-stack -suunnittelija terminä oli edelleen usealle ihmiselle vieras. Myös full-stack -suunnittelijan määritelmässä on edelleen epäselvyyksiä. Tästä syystä voi olla yritykselle vaikeaa etsiä rooliin sopivia ihmisiä pelkästään termin perusteella, sillä ihmiset ymmärtävät roolin eri tavoin. Yrityksen tulee siis tarkoin kertoa, mitä roolilla tarkoitetaan ja min-

kälaista osaamista roolilta kaivataan. Vielä yleisimmin tunnettu on kuitenkin full-stack-kehittäjä -termi [6], mikä saattaa esimerkiksi rekrytoinnissa aiheuttaa sekaannusta.

Budjetin ja aikataulun toteutuminen

Budjetin ja aikataulun toteutuminen voi riippua paljon full-stack -suunnittelijan todellisesta osaamisesta. Vaikka Canziba [6] ja Esteves et al. [12] esittivät väitteen, että prosessista myös säästyy aikaa ja rahaa, mikäli käyttäjäkokemussuunnittelija toteuttaa itse omat käyttöliittymäsuunnitelmansa, tässä tutkimuksessa väite ei saanut paljon kannatusta kyselyyn vastaajien kesken. Kuten jo edellä mainittiin, on voitu ajatella, että käyttäjäkokemussuunnittelijan ohjelmointitaidot eivät ole sellaisella tasolla, joka mahdollistaisi käyttöliittymäsuunnitelman sujuvan toteuttamisen, jolloin kehittäjät joutuvat avustamaan käyttäjäkokemussuunnittelijaa front-endin ohjelmoimisessa. Tästä syystä full-stack -suunnittelijan osallistuminen kehitykseen ei välttämättä auta projektin pitämisessä budjetin sisällä saati nopeuta projektin valmistumista. Eli sillä, kuinka taitava full-stack -suunnittelija on kehityksen puolella, voi olla merkitystä. Kuusisen [24] tekemä huomio siitä, että projektissa syntyy odottelu-aikaa, jos yksi ihminen vastuussa suuresta työmäärästä, nousi esiin myös kyselyyn tulleissa vastauksissa. Tämä etenkin voi vaikuttaa negatiivisesti budjetin ja aikataulun toteutumiseen.

Vähäinen kiinnostus itse rooliin

Vaikka työntekijät jo tekivät rooliin liittyviä tehtäviä, oli kiinnostus roolissa toimimiseen hyvin vähäistä. Ainoastaan vain noin 20% kaikista vastaajista olisi kiinnostuneita roolissa toimimisesta. Tähän voi vaikuttaa seuraavassa kohdassa, Ennakkoluulot roolia kohtaan.

Ennakkoluulot roolia kohtaan

Yksi tämän tutkimuksen kautta esiin noussut asia oli ennakoasenteet full-stack -suunnittelijan roolia kohtaan. Erityisesti kehittäjät yleisesti ottaen kokivat full-stack -suunnittelijan roolin siten, että full-stack -suunnittelijan koettiin olevan käyttäjäkokemussuunnittelija, joka on opetellut ohjelmointia. Monen kehittäjän vastauksissa oli pääteltävissä, että heidän mielestään käyttäjäkokemussuunnittelija on kykenemätön ohjelmoimaan. Roolissa kuitenkin voisi toimia esimerkiksi sellainen henkilö, joka on aiemmin työskennellyt kehittäjänä, mutta on halunnut myöhemmin kasvattaa osaamistaan suunnittelun puolella. Yritykselle kuitenkin kehittäjien ennakoasenteiden muuttaminen voi olla aikaa vievä prosessi.

Toinen ennakkoluuloihin liittyvä havainto kohdistui rooliin sopivien henkilöiden olemassaoloon ja yleisyyteen. Useammat kyselyyn vastanneista vastasivat, että rooliin on vaikeaa löytää sopivia henkilöitä. Toisaalta kyselyssä selvisi, että suurin osa kyselyyn vastanneista työntekijöistä jo työskentelee projekteissaan roolia vastaavissa tai lähes vastaavissa tehtävissä.

7.3 Työntekijöiden näkemykset full-stack -suunnittelijan roolin hyödyistä ja haasteista

Tämän aliluvun tarkoituksena on tarkastella tähän työhön liittyneessä tutkimuksessa työntekijöiden vastauksista ilmenneitä hyötyjä ja haasteita, joita vastaajat esittivät kyselyn avoimiin kysymyksiin. Edellisessä luvussa esitettiin kyselyn kautta saadut työntekijöiden mielipiteet full-stack -suunnittelijan hyödyistä. Full-stack -suunnittelijan roolista löydettiin molempia, sekä hyötyjä että haasteita, useampia. Haasteita löydettiin kuitenkin lukumäärällisesti enemmän: rooliin liittyvistä haasteista löydettiin yhteensä 28 erilaista teemaa ja hyödyistä löydettiin erilaisia teemoja 22. Hyödyistä ja haasteista löydettyt teemat tyypiteltiin yleisemmän käsityksen saamiseksi. Nämä tyypit on esitelty seuraavissa aliluissa työntekijöiden kyselyyn antamien kommentteihin ja vastauksiin perustuen.

7.3.1 Hyödyt

Hyötyjen tyyppiä löytyi lopulta kuusi kappaletta. Hyödyistä löydettiin seuraavia tyyppiä:

Suunnittelun ja käyttöliittymäsuunnitelman tarkoituksenmukaisuus

Full-stack -suunnittelijan ymmärrys ohjelmistokehityksestä antaa full-stack -suunnittelijalle valmiudet suunnitella ja luoda uutta käyttöliittymäsuunnitelmaa ohjelmistokehityksen rajoitteet huomioiden. Tällä tavoin mahdollistetaan myös se, että toteutukseen päätyy sellaisia ratkaisuja, jotka ovat teknisesti toteutettavissa.

Vision säilyminen ja lopputuotteen laadun paraneminen

Full-stack -suunnittelija voi osallistua itse käyttöliittymäsuunnitelman toteutukseen missä tahansa projektin vaiheessa. Projektin alussa – suunnitteluvaiheiden jälkeen – full-stack -suunnittelijan on mahdollista itse toteuttaa omiat käyttöliittymäsuunnitelmansa, kuten oli suunnitellut. Full-stack -suunnittelija varmistaa tällöin, että kaikki suunnitellut tuotteen kannalta oleelliset asiat päätyvät toteutukseen, ja visio näkyy toteutuksessa. Full-stack -suunnittelijan on tällöin helpompi ja nopeampi siirtää implisiittistä informaatiota toteutukseen, mikä voi olla hankalaa tai aikaa vievää muulla tavoin. Full-stack -suunnittelija voi myös korjata huomaamiaan suunnitteluongelmia tai toteutuksessa olevia virheitä itse. Tällöin esiintyy vähemmän katkoksia suunnittelun ja toteutuksen välillä, ja visio säilyy ehyenä. Kun lopputuote säilyttää sen vision, joka full-stack -suunnittelijalla alussa oli, pirstaloitumatta matkalla, tämä tekee lopputuotteesta paremman.

Osaaminen

Monipuolinen ja poikkitieteellinen osaaminen mahdollistaa erityisesti koko suunnitteluprojektin viemisen loppuun tutkimuksesta käyttöliittymäsuunnitelman toteuttamiseen. Full-stack -suunnittelijan on mahdollista osallistua eri vaiheisiin projektissa ja tarjota

osaamistaan ja tietoaan muille. Erityisesti ymmärrys kehityskokoprosessista on full-stack -suunnittelijan etu. Full-stack -suunnittelijan on mahdollista ottaa useampi osa-alue suunnittelussa huomioon, jolloin käyttöliittymäsuunnitelma pysyy full-stack -suunnittelijan omistuksessa.

Prosessin ja projektin helpottuminen

Projekti voi edetä nopeammin, kun full-stack -suunnittelija voi suunnittelun lisäksi toteuttaa käyttöliittymäsuunnitelman. Tämä johtuu siitä, että tällöin yksi välivaihe, jossa käyttäjäkokemussuunnittelija yrittää kertoa visiotaan kehittäjille, poistuu. Ideaalitulanteessa, full-stack -suunnittelija pystyy samaan aikaan suunniteltaessa myös toteuttamaan käyttöliittymäsuunnitelmaansa, jolloin käyttöliittymäsuunnitelman toteutuksesta tulisi valmista nopeammin. Myös, kun full-stack -suunnittelija vastaa käyttöliittymäsuunnitelmansa toteuttamisesta, kehittäjät voivat keskittyä käyttöliittymän logiikan kehittämiseen. Tällöin projekteissa tulee selkeä tehtävänjako käyttäjäkokemussuunnittelijoiden ja kehittäjien välille: full-stack -suunnittelija hoitaa kaiken suunnitteluun ja käyttöliittymään liittyvät asiat ja kehittäjät syvemmin ohjelmistoteknisen puolen ja funktionaalisen toteutuksen. Full-stack -suunnittelija on myös henkilö, joka tietää ja ymmärtää projektin kokonaistilanteen.

Kehittäjien ja käyttäjäkokemussuunnittelijoiden välisen kommunikaation parantaminen

Kun full-stack -suunnittelijalla on kokemusta käyttöliittymien toteuttamisesta, vision välittäminen sanallisesti ohjelmistokehityksen termein on käyttäjäkokemussuunnittelijalle helpompaa. Kommunikaatio full-stack -suunnittelijan ja kehittäjien välillä on sujuvampaa, kun asioista voidaan puhua kehittäjien ymmärtämin termein. Ylipäätään osaaminen käyttöliittymien toteutuksesta on eduksi. Kun käyttöliittymäsuunnitelma toteutetaan projektin teknologioita käyttäen, käyttöliittymäsuunnitelman toteutuksen voi siirtää sellaiseen käyttäjille. Kehittäjien voi olla helpompi ymmärtää full-stack -suunnittelijan näkemyksiä ohjelmistokoodin ja toimivan käyttöliittymän kautta. Käyttäjäkokemussuunnittelijan ja kehittäjien välinen kommunikaatio paranee. Kommunikaatioon ei tarvitse käyttää yhtä paljon aikaa kuin siinä tilanteessa, kun käyttäjäkokemussuunnittelija välittää visiona sanoin ja kuvin kehittäjille. Full-stack -suunnittelija on lisäksi lähempänä kehittäjiä ja voidaan kokea enemmän osaksi tiimiä, koska full-stack -suunnittelija osallistuu toteutukseen.

Roolin kustannustehokkuus

Pienissä tiimeissä ja projekteissa sopii full-stack -suunnittelijalla, koska näissä full-stack -suunnittelija voi toteuttaa projekteja kokonaisuudessaan. Kun projektiin tarvitaan yksi työntekijä hoitamaan sekä käyttäjäkokemussuunnittelun että osan front-end-toteutuksesta, voidaan säästää kustannuksissa, mikäli muuten projektiin oltaisiin tarvittu kolme henkilöä (yksi suunnittelija, yksi front-end kehittäjä ja yksi back-end-kehittäjä). Full-stack -suunnittelija voi myös säästää budjettia jo suunnitteluvaiheessa. Full-stack -suunnittelija

voi ottaa huomioon teknisten toteutusten kustannukset ja valita budjetin kannalta järkevämpiä suunnitteluratkaisuja. Hyvin pienissä yrityksissä ja projekteissa on tarvetta sellaisille työntekijöille, jotka pystyvät työskentelemään monella osa-alueella, sillä työntekijöiden määrää vähentämällä voidaan vähentää kustannuksissa.

7.3.2 Haasteet

Haasteiden teemoista muodostettiin kahdeksan erilaista tyyppiä. Haasteiden tyyppiä ovat seuraavat.

Tietojen ja taitojen riittämättömyys

Vaikka full-stack -suunnittelijan monipuolinen osaaminen koettiin hyödylliseksi, sen haasteita on eri osa-alueiden osaamisen epätasaisuus. Full-stack -suunnittelija on niin kutsuttu "generalisti", joka osaa vähän eri osa-alueiden tehtäviä, mutta ei osaa kaikkea välttämättä tarpeeksi hyvin. Myös taidot voivat jakaantua hyvin epätasaisesti eri osa-alueiden välillä. Epätasaisuus voi näkyä erityisesti ohjelmointiosaamisessa. Full-stack -suunnittelija ei välttämättä osaa yhtä hyvin toteuttaa kuin suunnittelulla. Toteutettu ohjelmistokoodi voi tällöin olla hyvien tapojen vastaista tai muuten huonosti tehty, vaikka olisikin toimivaa. Kokonaisvaltaista hyötyä roolista ei myöskään ole, mikäli full-stack -suunnittelija ei myös ymmärrä back-end-kehitystä front-end-kehityksen lisäksi.

Lopputuotteen huono laatu ja ristiriidat käyttäjien tarpeiden kanssa

Jos full-stack -suunnittelijan ohjelmointitaidot ovat vähäiset, kuten mainittiin edellisessä kohdassa, tämä voi näkyä myös lopputuotteessa. Huonosti toteutettu käyttöliittymä tekee lopputuotteesta huonomman. Lisäksi yksi käyttäjäkokemussuunnittelijoiden tehtävistä on suunnitella lopputuotteen käyttäjän kannalta hyviä käyttöliittymäsuunnitelmia. Full-stack -suunnittelijan suhteen voi syntyä ristiriitaa, sillä full-stack -suunnittelija saattaa suunnitella sellaisia käyttöliittymäsuunnitelmia, jotka hän pystyy itse kehittämään. Esimerkiksi teknisesti erittäin haastavat käyttöliittymäsuunnitelmat halutaan jättää pois suuren työ määrän vuoksi, vaikka ne voisivat olla loppukäyttäjän kannalta olennaisia. Tällöin käyttäjän näkökulma voi jäädä vähemmälle huomiolle ja vaikuttaa negatiivisesti tuotteen laatuun.

Roolin vaatimusten täyttäminen

Full-stack -suunnittelijan roolissa on paljon vastuuta, ja se vaatii paljon erilaista, poikkeatieteellistä tietoa ja taitoa. Tämä vaatii paljon yhdeltä ihmiseltä. Osaamisen ylläpitäminen on myös haastavaa: erityisesti teknologiat muuttuvat kokoajan, joten jo pelkästään ohjelmistokehitykseen liittyvissä kielissä ja teknologioissa ajantasalla pysyminen vaatii paljon työtä. Myös muissa osa-alueissa hyvänä pysyminen vaatii työntekijältä aikaa. Rooliin kehittyminen vie paljon aikaa ja vaivaa, joten se vaatii hyvin paljon kokemusta eri osa-alueilta.

Yksin toimiminen

Roolissa on vaarana se, että työntekijä jää työskentelemään yksin. Tällaisessa roolissa työskentelevää työntekijää on mahdollista laittaa työskentelemään yksin projekteihin, milloin työntekijältä puuttuu mahdollisuudet omien ideoidensa ja ajatustensa peilaamiseen ja muiden näkökulmien saamiseen. Roolissa toimiminen voi pidemmän päälle käydä yksinäiseksi.

Projektien organisointi vaikeutuu

Full-stack -suunnittelija joutuu usein vaihtamaan kontekstia suunnittelusta ja kehittämiseen. Tämä vähentää full-stack -suunnittelijan työskentelyn tehokkuutta. Lisäksi yhden ihmisen ollessa vastuussa monesta tehtävästä, on vaarana projektin hidastuminen etenkin yllättävissä tilanteissa, kuten työntekijän sairastuessa. Myös full-stack -suunnittelijan taidot toteuttaa käyttöliittymä hyvin valmiin näköiseksi, voi aiheuttaa asiakkaille sellaisen mielikuvan, että projekti on lähellä valmista, vaikkei näin olisi. Myös kommunikaation määrä voi lopulta huonontua, sillä kehittäjien ei tarvitse niin paljon keskustella full-stack -suunnittelijan kanssa.

Haasteet yritykselle

Yrityksen näkökulmasta full-stack -suunnittelija aiheuttaa haasteita ajankäytössä, sopivien ihmisen sekä sopivien projektien löytämisessä. Full-stack -suunnittelija tekee useamman ihmisen tehtäviä, jolloin resurssija voi olla vähemmän eri projekteihin. Full-stack -suunnittelijan tapauksessa vaarana on tässä tilanteessa myös ylityöllistyminen.

Työntekijän omat resurssit

Koska full-stack -suunnittelija työskentelee useammalla osa-alueella, se vaikeuttaa työntekijän omaa ajankäyttöä sekä lisää henkistä kuormitusta. Työntekijällä voi olla vaikeuksia keskittyä olennaiseen, koska tehtäviä on niin monta.

Kehittäjien ennakoasenteet

Kehittäjät suhtautuvat käyttäjäkokemussuunnittelijoiden ohjelmointiosaamiseen sekä käyttäjäkokemussuunnittelijoiden tuottamaan koodiin varautuvasti. Full-stack -suunnittelija koettiin kyselyn vastauksissa käyttäjäkokemussuunnittelijana, joka osaa hieman ohjelmoida.

7.4 Full-stack -suunnittelijan rooliin liittyvien haasteiden ratkaisu

Myös haasteisiin teemoitellut ratkaisut tyypiteltiin yleispätevämmiksi. Ratkaisutyyppejä tuli yhteensä kahdeksan. Seuraavassa on esitelty nämä ratkaisutyypit sekä niiden tiivistelmät työntekijöiden kyselyyn antamiin vastauksiin perustuen. Jotkin ratkaisuehdotukset

ovat toistensa vastakohtia, näin esimerkiksi projektissa toimivien roolien ja niiden määrien järkevässä valinnassa. Ratkaisut ovat erilaisille sekä eri kokoisille yrityksille eri tavalla toimivia. Ratkaisut ovat sovellettavissa harkiten. Nämä ratkaisut on esitetty seuraavissa aliluvuissa työntekijöiden kyselyyn antamien kommentteihin ja vastauksiin perustuen.

7.4.1 Opiskelu ja koulutus

Roolin kannalta oikeiden asioiden opiskelu ja koulutus auttavat sekä yksilö- että yritystasolla ratkaisemaan haasteita, jotka liittyvät erityisesti roolin osaamiseen, taitojen ylläpitämiseen ja työntekijän ajankäyttöön. Käyttäjäkokeiluun tulisi ymmärtää ohjelmistokehitystä ja kehittäjien suunnittelua. Kokemuksen kautta full-stack -suunnittelija voi kehittää osaamistaan usealla eri osa-alueella. Opiskelun ja koulutuksen kautta myös rooliin sopivia henkilöitä voi löytyä lisää.

7.4.2 Full-stack -suunnittelijan roolin termin ja määritelmän tekeminen tunnetuksi

Yritysten tulisi panostaa full-stack -suunnittelijan roolin vakiinnuttamiseen ja tehdä roolista tunnetumpi. Roolin määritelmää erityisesti ohjelmistokehityksen osalta tulisi selkeyttää. Tämä voidaan saada aikaan hyvällä ja sopivalla markkinoinnilla. Esimerkiksi rekrytoinnissa tulisi korostaa ohjelmoinninmerkitystä roolissa. Lisäksi asiakkaille tulisi kertoa roolista, jotta he ymmärtävät mitä rooli tekee. Tällöin voidaan välttyä siltä tilanteelta, että asiakkaat luulevat projektin olevan valmiimpi kuin se todellisuudessa on, silloin kun full-stack -suunnittelija toteuttaa omat käyttöliittymäsuunnitelmansa toimivaksi käyttöliittymäksi.

7.4.3 Full-stack -suunnittelijan resurssointi järkevästi

Ajankäytön sekä roolin laajaan vastuualueeseen ratkaisuna ehdotettiin, että full-stack -suunnittelijan tulisi valita oikean kokoihin projekteihin, eli mahdollisuuksien mukaan pienempiin projekteihin. Full-stack -suunnittelijan ei myöskään tulisi työskennellä monessa projektissa kerrallaan.

7.4.4 Projektin tehtävien tekemisen painottaminen osaamisen ja kiinnostuksen perusteella

Toteutuksen huonon laadun ratkaisemiseksi ja roolin vaatimusten laajuuden sekä henkisen kuormituksen vähentämiseksi projektin tehtäviä tulisi delegoida eri ihmisille. Tehtäviä tulisi jakaa työntekijöiden oman kiinnostuksen mukaan, milloin jokainen voisi keskittyä tiettyyn osa-alueeseen.

7.4.5 Projektissa toimivien roolien ja niiden määrien järkevä valinta

Full-stack -suunnittelijan roolin täyden potentiaalin hyödyntämiseksi projekteihin valittavat roolit sekä eri roolien määrä tulisi valita järkevästi. Tämä riippuu yrityksen koosta. Kyselyn vastausten perusteella useampi full-stack -suunnittelijan rooli toimii suuressa projektissa, mikä vähentää roolissa työskentelevän työtaakkaa. Tällöin roolissa työskentelevät voivat keskittyä omien vahvuuksiensa mukaan eri osa-alueisiin. Pienemmässä projektissa taas toimii yksi full-stack -suunnittelijan rooli, jolloin yksi henkilö voi vastata useammasta tuotesuunnittelun ja -kehityksen osa-alueesta. Käyttäjätarpeiden ja toteutuksen ristiriitaan sekä ohjelmointitaitojen puutteeseen ehdotettiin ratkaisuksi myös perinteistä roolijakoa eli sitä, että projektissa työskentelee sekä käyttäjäkokemussuunnittelija että kehittäjä.

7.4.6 Muiden työntekijöiden tuki

Roolin vaatimusten täyttämiseen sekä muiden näkökulmien puutteeseen tuki muilta vastaavassa roolissa toimivilta on hyvin tärkeää. Lisäksi full-stack -suunnittelijan ei tulisi unohtaa kommunikaatiota kehittäjien kanssa.

7.4.7 Ohjelmointia helpottavien menetelmien ja työkalujen käyttäminen

Ohjelmointityöhön liittyvä haasteita kuten ohjelmointitaitojen puutteellisuutta tai toteutukseen kuluvaa aikaa voidaan ratkaista erilaisin ohjelmistokehityksen menetelmin. Näitä ovat esimerkiksi koodikatselmointi, jolla voidaan varmistaa, että ohjelmointi on tehty hyviä tapoja noudattaen, sekä parikoodauksella, jossa full-stack -suunnittelija ohjelmoi yhdessä kehittäjän kanssa. Lisäksi full-stack -suunnittelija voi yrittää ohjelmoida suunnittelun yhteydessä. Tähän auttaa esimerkiksi, jos full-stack -suunnittelija hyödyntää olemassa olevia käyttöliittymäelementtikirjastoja.

7.4.8 Omien vahvuuksiensa tunteminen ja halu kehittyä

Monen osa-alueen hallinta vaatii työntekijältä sitä, että hän ymmärtää omat vahvuutensa ja on valmis kehittämään niitä osa-alueita, joissa osaaminen ei ole yhtä hyvällä tasolla. Omien taitojen kehittäminen vaatii työntekijältä työskentelyä oman mukavuusalueensa ulkopuolella. Oman mukavuusalueen ulkopuolelle meneminen voi tarkoittaa esimerkiksi työskentelyä erilaisissa projekteissa, jolloin näistä muodostuu lisää kokemusta.

7.5 Tulosten luotettavuus

Kyselyyn vastasi useampia alalla toimivia ammattilaisia ja kyselyn otokseen saatiin kattavasti vastauksia erilaisista taustoista. Vastaajien iät, sukupuoli, koulutustausta, työko-

kemus - sekä vuosina että eri aloilla - erosivat toisistaan. Yrityksen koossa ja rooleissa oli yhtä ryhmää selkeästi enemmän, mutta kyselyssä oli silti näissäkin erilaisista ryhmistä. Lähes puolella vastaajista oli ennestään kokemusta full-stack -suunnittelijan roolista, joko niin, että olivat itse toimineet joskus roolissa tai joku muu oli toiminut samassa projektissa full-stack-suunnittelijan roolissa. Yli puolella vastaajista ei kuitenkaan ollut omia kokemuksia roolista, joten osa vastauksista perustuu vastaajan oletukseen siitä, miten full-stack -suunnittelijan rooli toimii. Tämä saattaa aiheuttaa jonkin verran epätarkkuutta vastauksissa.

Lisäksi roolin määritelmä ei ollut kaikille roolin termin ennestään kuulleillekaan täysin samanlainen, ne vastaajat, jotka kokivat roolin erilaisena kuin määritelmässä, ovat voineet vastata kyselyssä mainitun roolin määritelmän mukaan. Tutkimuksessa kävi ilmi, että useat vastaajat ovat jo työskennelleet roolia vastaavissa tehtävissä. Se, mitä tässä tutkimuksessa ei vielä selvitetty tarkemmin, on muiden tehtävähdistelmien ja kiinnostuksen korrelaatio, esimerkiksi se, tekeekö joku tehtäviä vastoin omia kiinnostuksenkohteitaan.

Tuloksiin voi vaikuttaa tutkimusmenetelmän valinta, sillä tässä tutkimuksessa tehtiin pelkkä kysely, jolloin esimerkiksi tutkittavilta ei voitu kysyä täydennyksiä vastauksiin [19, 40]. Tutkijan päätelmät vastauksista ovat voineet erota siitä, mitä vastaaja niillä alunperin tarkoitti. Lisäksi aiheesta löytyi hyvin vähän teoriaa ennestään, mikä vaikuttaa siihen, kuinka paikkaansa pitävinä tuloksia voidaan pitää.

Kyselyn pituudeksi oli määritelty noin 15 minuuttia. Tämä on saattanut joillekin potentiaalisille vastaajille kuulostaa liian pitkältä ajalta kyselyyn vastaamiseen, minkä vuoksi vastauksia ei tullut vielä enemmän. Se, että kysely on yritetty käydä läpi mahdollisimman nopeasti, on saattanut myös vääristää tuloksia.

Kuitenkin kyselyn kysymykset sekä vastausvaihtoehdot oli harkitut ja huolellisesti mietityt. Tähän oltiin käytetty paljon aikaa. Myös vastaajat kommentoivat palautteissaan, että kysely oli sopivan mittainen, ja kysely ja sen kysymykset olivat hyvin muotoiltuja.

8. JOHTOPÄÄTÖKSET

Tässä luvussa tiivistetään tutkimuksen keskeiset tulokset yhteen aiemmista luvuista. Sen lisäksi arvioidaan tutkimuksen onnistumista. Lopuksi vielä esitetään työn jatkotutkimusmahdollisuudet.

8.1 Tutkimuksen keskeiset tulokset

Tutkimuksessa selvitettiin sekä kirjallisuuskatsauksen että empiirisen tutkimuksen avulla, minkälaisia hyötyjä ja haasteita full-stack -suunnittelijan roolista on. Tutkimus toteutettiin kyselynä. Kysely lähetettiin 13:lle eri ohjelmistoyritykselle, jotka sijaitsivat Pääkaupunkiseudulla, Pirkanmaalla ja Varsinais-Suomessa. Kyselyyn vastasi kaiken kaikkiaan 96 ihmistä erilaisista taustoista ja eri kokoisista yrityksistä.

Full-stack -suunnittelijan roolista on sekä hyötyjä että haasteita ohjelmistoyrityksille ketterässä ohjelmistokehityksessä. Yrityksen kannalta hyötyjä ja haasteita tutkittiin kirjallisuusselvitysten sekä kyselyssä ennalta määriteltyjen teemojen kautta. Tämän tutkimuksen kautta full-stack -suunnittelijan roolin tärkeimpiä löydettyjä hyötyjä yrityksen näkökulmasta ovat se, että roolissa toimimiseen löytyy jo ennestään osaamista yrityksistä, työntekijöiltä löytyy kiinnostusta rooliin liittyvien tehtävien tekemiseen ja se, että rooli onnistuisi jo nykyisellään yrityksissä. Lisäksi full-stack -suunnittelijan roolilla voi olla vaikutusta projekteissa kehitettävän lopputuotteen laadun paranemiseen ja projektiin liittyvien prosessien nopeutumiseen ja yksinkertaistumiseen. Yrityksen näkökulmasta full-stack -suunnittelijan suurimpia haasteita ovat se, että roolin termin ja määritelmän tunnettavuus on vielä vähäinen ja se, että rooliin kohdistuu paljon negatiivisia ennakkoluuloja. Roolissa työskentelevällä on myös yhdelle ihmiselle liikaa vastuualueita eikä välttämättä tarpeeksi osaamista tämän roolinkaan kannalta tärkeistä osa-alueista. Lisäksi niitä, jotka haluaisivat työskennellä roolissa, on tällä hetkellä vähän, ja roolin onnistuminen voi vaatia enemmän erilaisia resursseja yritykseltä, erityisesti jos kyseessä on hyvin pieni yritys.

Tutkimuksessa lisäksi selvitettiin työntekijöiden mielipiteitä ja kokemuksia full-stack -suunnittelijan roolista. Tutkimuksessa saaduista tuloksista voidaan todeta, että full-stack -suunnittelijan roolin hyödyt ja haasteet joiltain osin jakavat vielä mielipiteitä. Kuitenkin hyödyistä ja haasteista oli mahdollista löytää kyselyyn vastanneiden kesken pääteemoja. Työntekijöiden mainitsemia hyötyjä olivat suunnittelun ja käyttöliittymäsuunnitelman taroituksenmukaisuus, vision säilyminen ja lopputuotteen laadun paraneminen, full-stack -suunnittelijan monipuolinen osaaminen, ohjelmistokehitysprosessin ja projektin helppotuminen, kehittäjien ja käyttäjäkokemussuunnittelijoiden välisen kommunikaation paraneminen sekä roolin kustannustehokkuus. Työntekijöiden mainitsemia haasteita olivat

full-stack -suunnittelijan tietojen ja taitojen riittämättömyys, toteutukseen liittyvä ristiriita käyttäjätarpeiden ja toteutuksen haasteellisuuden välillä, roolin vaatimusten täyttäminen, yksin toimiminen, projektien organisoimisen vaikeutuminen, työntekijän omat resurssit sekä kehittäjien ennakoasenteet.

Tutkimuksessa myös selvitettiin työntekijöiden näkemyksiä löydettyjen haasteiden ratkaisemiseksi. Työntekijöiden esittämiä ratkaisuja olivat rooliin liittyvien asioiden opiskelu ja koulutus, panostaminen termin ja määritelmän tunnettavuuteen, full-stack -suunnittelijan järkevä resurssointi yrityksissä, tehtävien jakaminen tekijöille kiinnostuksen ja osaamisen perusteella, roolien valitseminen projekteihin järkevästi, muiden työntekijöiden tuki, ohjelmointia helpottavien menetelmien ja työkalujen käyttäminen sekä omien vahvuksiensa tunteminen.

Tutkimuksessa selvisi vastoin tutkijan sekä useamman kyselyyn vastanneen odotuksia rooliin sopivien henkilöiden löytymisestä, että yrityksissä tällaisessa roolissa toimivia työntekijöitä löytyy jo ennestään useampia. Osaamista ja kiinnostusta rooliin löytyy, mutta yrityksen tulisi osata löytää keinot hyödyntää työntekijöiden kiinnostusta ja osaamista. Tiukkojen roolien sijaan projekteissa ja tiimeissä voisi painottaa ihmisten erilaisia taitoja ja kiinnostuksen kohteita ja antaa työntekijöiden tehdä tehtäviä heidän omien mieltymystensä mukaan.

Full-stack -suunnittelijan roolia on kritisoitu siitä, että roolissa toimiva henkilö on niin kutsuttu generalisti. Roolissa toimivalla henkilöllä voi olla tietoa useammalta eri ohjelmistokehityksen osa-alueelta, mutta osaamista ei löydy näiltä osa-alueilta tarpeeksi. Vaikka ohjelmistokehitykseen voi olla saatavilla kehittäjien työtä helpottavia viitekehyksiä, ei tämä vähennä ohjelmistokehitysosaamisen tarvetta. Ohjelmistotarpeet ovat tekniikan kehittymisen ja tarjolla olevien mahdollisuuksien myötä monimutkaistuneet, jolloin ohjelmistokehitysprosesseissa on edelleen tarvetta tiettyjen osa-alueiden asiantuntijoille.

Jos rooli valitaan projektiin tarkoituksenmukaisesti, rooli voisi yksinkertaistaa ja helpottaa ohjelmistokehitysprosessia sekä ratkaista kehittäjien ja käyttäjäkokeussuunnittelijoiden kommunikaatioon liittyviä haasteita. Parhaimmassa tapauksessa rooli voi poistaa joitain välivaiheita ohjelmistokehityksestä. Koska full-stack -suunnittelijan on mahdollista toteuttaa itse käyttöliittymäsuunnitelmansa, full-stack -suunnittelijan ei tarvitse käyttää niin paljon aikaa siihen, että kaikki käyttäjäkokeussuunnittelun implisiittinen tieto siirtyy full-stack -suunnittelijalta ohjelmistokehittäjälle. Tämä voi säästää projektilta aikaa ja resursseja. Kuitenkin, mikäli full-stack -suunnittelijalla ei ole tarvittavaa ohjelmistokehitysosaamista, voi lopputuotteen laatu kärsiä.

8.2 Full-stack -suunnittelija terminä

Tässä työssä käytettiin tutkitusta käyttäjäkeskeiseen suunnitteluun liittyvästä roolista termiä **full-stack -suunnittelija**. Kuten johdannossa 1.1 ja luvussa 4 selvennettiin, ei tässä

työssä käytetty termi ole vielä vakiintunut ohjelmistoalalla. Termi oli tässä työssä muodostettu mukaillen ohjelmistokehityksen puolelta tuttua termiä ”full-stack -kehittäjä”, jotta termi ”full-stack -suunnittelija” olisi johdonmukainen myös ”full-stack -kehittäjä” -termin kanssa.

Sitä, kuinka hyvin termi ”full-stack -suunnittelija” vastaa roolin kuvausta, on pohdittava kriittisesti. Full-stack -suunnittelijalla tarkoitettiin tässä työssä sellaista henkilöä, jonka on mahdollista työskennellä usealla käyttäjäkeskeisen suunnittelun osa-alueella sekä myös ohjelmistokehityksen puolella käyttöliittymän toteuttajana. Termi full-stack -suunnittelija voi olla harhaanjohtava, sillä se ei sellaisenaan kerro paljon roolista tai siihen liittyvistä tehtävistä. Erityisesti sen sisältämä englanninkielinen full-stack -etuliite ei ole suomen kielen mukainen.

Roolin määritelmä voisi tarjota erilaisia näkökulmia roolin nimeämiseen suomen kielessä. Roolin määritelmän mukaan roolissa työskentelevä henkilö on käyttäjäkeskeisen suunnittelijan sekä käyttöliittymän toteuttajan yhdistelmä. Tällaisesta henkilöstä puhuttaessa voitaisiin sanoa esimerkiksi ”käyttöliittymiä toteuttava käyttäjäkeskeinen suunnittelija”, jossa viitataan koko käyttäjäkeskeiseen suunnitteluprosessiin sekä käyttöliittymän toteuttamiseen. Tämän lyhenteenä voitaisiin käyttää termiä ”toteuttava käyttäjäkeskeinen suunnittelija” tai jopa ”toteuttava suunnittelija”, kun konteksti on tiedossa.

Roolia voitaisiin kuvata myös muulla termillä. Roolin poikkitieteellisyys on asia, mikä tulisi kuitenkin termissä huomioida. Tässä työssä käytetty termi ”full-stack -suunnittelija” ei tällä hetkellä huomioi eksplisiittisesti roolin poikkitieteellisyyttä. Mikäli henkilö osallistuisi käyttäjäkeskeiseen suunnitteluun sekä ohjelmistokehityksen puolella käyttöliittymän toteutuksen lisäksi myös esimerkiksi palvelinpuolen toteutukseen, voitaisiin roolia kuvata termillä ”käyttäjäkeskeinen suunnittelija-kehittäjä”. Tämä termi voisi ilmaista henkilön toimivan poikkitieteellisesti käyttäjäkokeussuunnittelu- sekä ohjelmistokehitysprosessissa.

8.3 Tutkimuksen onnistumisen arviointi

Tutkimus onnistui odotuksiin nähden hyvin. Tutkimuksessa toteutettu kysely keräsi riittävästi ja odotettua enemmän vastauksia. Vastauksia tuli eri kokoisista yrityksistä sekä eri rooleissa toimivilta työntekijöiltä. Myös vastaajien taustat olivat monipuolisia. Vastaajien joukkoon olisi toivottu kuitenkin lisää suunnittelijoita, sillä suuri osa kyselyyn vastanneista työskenteli kehittäjänä. Käyttäjäkokeussuunnittelijoiden antamat vastaukset olisivat voineet tuoda vielä enemmän diversiteettiä vastauksiin sekä lisää uusia ajatuksia aiheeseen käyttäjäkokeussuunnittelijan työn näkökulmasta. Osallistuminen tutkimukseen oli täysin vapaaehtoista. Sitä, mikä vaikutti siihen, ketkä kyselyyn vastasivat ja ketkä eivät, voidaan pohtia kriittisesti. Tuloksista pystyi päättelemään, että aihe herätti vastaajissa paljon ajatuksia. Oletettavasti osalla vastaajista on ollut jo jonkinlaisia ennakoasenteita tutkittavaa aihetta kohtaan, mikä on saattanut ylikorostaa tiettyjä mielipiteitä.

Tutkimuksen kannalta olisi ollut toivotumpaa, että aiheesta olisi löytynyt aiempia tutkimuksia. Tämä olisi helpottanut tutkimuksen toteuttamista. Aiemmat tutkimustulokset olisivat voineet lisätä tämän tutkimuksen luotettavuutta. Työn kannalta olennaisia aiempia tutkimuksia kuitenkin löytyi sen verran, jotta tämän työn tuloksia pystyttiin vertaamaan aiempiin tutkimustuloksiin.

Eri ohjelmistoyritysten näkökulmasta tämän tutkimuksen tulokset ovat melko vaatimattomat. Vaikka tutkimusotanta oli melko kattava, yli puolella vastaajista ei ollut omakohtaista kokemusta aiheesta, mistä syystä tulokset voivat olla joltain osin enemmän arvailuja kuin todellista tietoa tai kokemusta oikeista tilanteista. Kattavamman tutkimuksen toteuttaminen olisi vaatinut tämän työn puitteissa lisää aikaa ja resursseja, joita tutkijan ei ollut mahdollista järjestää.

Tutkittava aihe oli tutkijan mielenkiinnonkohde sekä lähellä tutkijan omaa työnkuvaa. Tämä on voinut vaikuttaa mielikuviin sekä päätelmiin tutkimustuloksista. Tutkijalla oli myös ennalta epäformaalia empiiristä tietoa, kuten esimerkiksi omia kokemuksia aiheesta. Nämä eivät kuitenkaan tuota merkittäviä eroja itse tutkimuksen toteuttamiseen tai sen lopputuloksiin.

Työhön käytettävissä olleeseen aikaan ja resursseihin verrattuna työ onnistui hyvin. Kyselyn aihe herätti vastaajien mielenkiinnon. Useampi vastaaja kommentoi aiheen olleen mielenkiintoinen ja ajatuksia herättävä. Myös useampi yrityksen puolesta kyselyn vastannut ja eteenpäin välittänyt henkilö vastasi aihetta kommentoiden. Eräältä yritykseltä jopa tuli ehdotus siitä, mille muille yrityksille kysely kannattaisi laittaa eteenpäin.

8.4 Jatkotutkimusmahdollisuudet

Full-stack -suunnittelijan rooli on hyvin vähän tutkittu. Tämän työn aikana aiempia tutkimuksia ei löytynyt yhtään. Tästä syystä jatkotutkimusmahdollisuuksia on useampia.

Tässä tutkimuksessa keskityttiin ylipäättään keräämään full-stack -suunnittelijaan liittyviä ajatuksia. Jatkotutkimuksena voitaisiin tutkia full-stack -suunnittelijan roolia jossain todellisessa viitekehyksessä, esimerkiksi toteuttamalla projektiseurantaa ja tekemällä sen kautta päätelmiä roolin soveltuvuudesta esimerkiksi eri kokosiin yrityksiin, tiimeihin tai projekteihin. Lisäksi voitaisiin kerätä enemmän näkökulmia jo roolissa työskenteleviltä työntekijöiltä esimerkiksi haastatteluin.

Myös termiä ”full-stack -suunnittelija” voitaisiin tutkia tarkemmin esimerkiksi tutkimalla siihen kohdistuvia ennakkoluuloja. Termi kaipaisi myös tarkemman määritelmän muodostamista, jotta full-stack -suunnittelijan rooli voitaisiin yleistää samalle tasolle full-stack -kehittäjän termin kanssa.

Erityisen mielenkiintoinen jatkotutkimus olisi verrata todellisissa tilanteissa, minkälaista konkreettista hyötyä projektille tässä työssä määrittelystä full-stack -suunnittelijan roo-

lista (käyttäjäkemussuunnittelija osallistuu enemmän ohjelmistokehitykseen) on verrattuna perinteiseen käyttäjäkemussuunnittelijoiden ja kehittäjien roolijakoon. Tämän kautta voitaisiin selvittää, minkälaisia muutoksia rooli aiheuttaisi ohjelmistoprojekteissa. Esimerkiksi se, kuinka useampi full-stack -suunnittelija toimii samassa projektissa, voi olla mielenkiintoinen lähestymistapa aiheeseen.

LÄHTEET

- [1] The Definition of a Full Stack Designer, 1stWebDesigner, verkkosivu, 2017. Saatavissa (viitattu 5.10.2018): <https://1stwebdesigner.com/full-stack-designer/>
- [2] Agile 101, Agile Alliance, verkkosivu. Saatavissa (viitattu 10.10.2018): <https://www.agilealliance.org/agile101/>
- [3] Manifesto for Agile Software Development, UX Collective, verkkosivu, 2001. Saatavissa (viitattu 10.10.2018): <http://agilemanifesto.org/iso/en/manifesto.html>
- [4] I. Boivie, J. Gulliksen, B. Göransson, The Lonesome Cowboy: A Study of the Usability Designer Role in Systems Development, Interacting with Computers, Vol. 18, Iss. 4, jul, 2006, pp. 601–634.
- [5] M. Brhel, H. Meth, A. Maedche, K. Werder, Exploring Principles of User-centered Agile Software Development, Inf. Softw. Technol., Vol. 61, Iss. C, May 2015, pp. 163–181.
- [6] E. Canziba, Hands-On UX Design for Developers: Design, prototype, and implement compelling user experiences from scratch, Packt Publishing Ltd, Birmingham, UK, 2018, 350 p. Saatavissa: <https://books.google.fi/books?id=DAInDwAAQBAJ>
- [7] D. Cartwright, What is full-stack designer, and should you be one?, WebdesignerDepot, verkkosivu, 2015. Saatavissa (viitattu 5.10.2018): <https://www.webdesignerdepot.com/2015/06/what-is-a-full-stack-designer-and-should-you-be-one/>
- [8] T. Chow, D.B. Cao, A Survey Study of Critical Success Factors in Agile Software Projects, J. Syst. Softw., Vol. 81, Iss. 6, June 2008, pp. 961–971.
- [9] J.V. Dalen, Why You Shouldn't Hire A Designer/Developer Hybrid, Medium, verkkosivu, 2016. Saatavissa (viitattu 5.10.2018): <https://moduscreate.com/blog/why-you-shouldnt-hire-a-designer-developer-hybrid/>
- [10] S. Dash, The DesOps Enterprise: (Volume 1) The Overview & Culture: Re-invent Your Organization, DesOps.io, The DesOps Enterprise, 2018, 220 p. Saatavissa: <https://books.google.fi/books?id=7tdfDwAAQBAJ>
- [11] T. DeGroat, Becoming a Full-Stack Designer: How You Can Stand Out From the Crowd, Springboard, verkkosivu, 2018. Saatavissa (viitattu 5.10.2018): <https://www.springboard.com/blog/becoming-a-full-stack-designer/>

- [12] M. Esteves, V. Andrade, Designing Interaction Concepts, Managing Customer Expectation and Mastering Agile Development in Rich Application Product Development, in: Jacko, J.A. (ed.), Human-Computer Interaction. Design and Development Approaches, Berlin, Heidelberg, 2011, LNCS 6761 - Part I, Springer Berlin Heidelberg, pp. 54–62.
- [13] O. Filipova, R. Vilão, Software Development From A to Z: A Deep Dive into all the Roles Involved in the Creation of Software, Apress, 2018, 291 p. Saatavissa: <https://books.google.fi/books?id=VpujuQEACAAJ>
- [14] D. Fox, J. Sillito, F. Maurer, Agile Methods and User-Centered Design: How These Two Methodologies are Being Successfully Integrated in Industry, in: Agile 2008 Conference, Toronto, ON, Canada, Aug, 2008, pp. 63–72.
- [15] T. Gerring, What is a Full Stack Designer? How Do I Become One?, Topcoder, verkkosivu, 2017. Saatavissa (viitattu 5.10.2018): <https://www.topcoder.com/blog/full-stack-designer-become-one/>
- [16] Google Forms – Luo kauniita lomakkeita, Google, verkkosivu. Saatavissa (viitattu 13.11.2018): <https://www.google.com/forms/about>
- [17] J. Gulliksen, B. Göransson, M. Lif, A User-Centered Approach to Object-Oriented User Interface Design, in: van Harmelen (21), 2001.
- [18] I. Haikala, J. Märijärvi, Ohjelmistotuotanto, 11. painos, Talentum, Helsinki, 2006, XX, 440 s.
- [19] S. Hirsjärvi, P. Remes, P. Sajavaara, Tutki ja kirjoita, 15. uud. p. p., Tammi, Helsinki, 2009, 464 s.
- [20] J. Humble, 5 Reasons You Should Hire a Full Stack Designer, Jeffrey Humble, verkkosivu, 2015. Saatavissa (viitattu 5.10.2018): <http://www.jeffreyhumble.com/blog/2015/5/7/the-full-stack-designer>
- [21] E. Ibragimova, L. Verboom, N. Mueller, Building a Team to Champion User-Centered Design Within an Agile Process, in: Marcus, A., Wang, W. (eds.), Design, User Experience, and Usability: Theory, Methodology, and Management, Cham, 2017, Springer International Publishing, pp. 584–596.
- [22] K. Kuusinen, Integrating UX Work in Agile Enterprise Software Development, dissertation, Tampere University of Technology, Publication 1339, 2015, 296 p. Saatavissa: [https://tutcris.tut.fi/portal/en/publications/integrating-ux-work-in-agile-enterprise-software-development\(a5ac6841-95ad-4f3f-ba1b-12\).html](https://tutcris.tut.fi/portal/en/publications/integrating-ux-work-in-agile-enterprise-software-development(a5ac6841-95ad-4f3f-ba1b-12).html)

- [23] K. Kuusinen, Task Allocation Between UX Specialists and Developers in Agile Software Development Projects, in: Abascal, J., Barbosa, S., Fetter, M., Gross, T., Palanque, P., Winckler, M. (eds.), *Human-Computer Interaction – INTERACT 2015*, Cham, September 14–18, 2015, LNCS 9298 - Part III, Springer International Publishing, pp. 27–44.
- [24] K. Kuusinen, BoB: A Framework for Organizing Within-Iteration UX Work in Agile Development, in: Gilbert, C., Lárusdóttir, M., Gregory, P., Cajander Åsa (eds.), *Integrating User-Centred Design in Agile Development*, Springer International Publishing, Switzerland, 2016, pp. 205–224.
- [25] T. Øvad, L.B. Larsen, Templates: A Key to Success When Training Developers to Perform UX Tasks, in: Gilbert, C., Lárusdóttir, M., Gregory, P., Cajander Åsa (eds.), *Integrating User-Centred Design in Agile Development*, Springer International Publishing, Switzerland, 2016, pp. 77–96.
- [26] What is a Full Stack Designer? Will You Be One?, Mockplus, verkkosivu, 2017. Saatavissa (viitattu 5.10.2018): <https://www.mockplus.com/blog/post/full-stack-designer>
- [27] J. Moffett, *Bridging UX and Web Development*, Morgan Kaufmann, Boston, Massachusetts, USA, 2014, 224 p. Saatavissa: <https://www.sciencedirect.com/book/9780124202450/bridging-ux-and-web-development>
- [28] Onix-Systems, A Unicorn: The Tale of a Full-Stack Designer, Medium, verkkosivu, 2018. Saatavissa (viitattu 5.10.2018): https://medium.com/@onix_systems/a-unicorn-the-tale-of-a-full-stack-designer-eba30b3b565d
- [29] R. Pressman, *Software Engineering: A Practitioner’s Approach*, 7th ed., McGraw-Hill, Inc., New York, NY, USA, 2009, 888 p.
- [30] L. Ratcliffe, M. McNeill, *Agile Experience Design: A Digital Designer’s Guide to Agile, Lean, and Continuous*, 1st ed., New Riders Publishing, Thousand Oaks, CA, USA, 2011, 320 p.
- [31] G. Ruhe, C. Wohlin (eds.), *Software Project Management in a Changing World*, Springer-Verlag Berlin Heidelberg, Berliini, Saksa, 2014, XX, 477 p. Saatavissa: <https://link.springer.com/book/10.1007/978-3-642-55035-5>
- [32] D. Salah, R.F. Paige, P. Cairns, A Systematic Literature Review for Agile Development Processes and User Centred Design Integration, in: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, New York, NY, USA, 2014, ACM, EASE ’14, London, England, United Kingdom, pp. 5:1–5:10.

- [33] K. Schwaber, J. Sutherland, *The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game*, Scrum.Org and ScrumInc, 2017. Saatavissa: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>
- [34] T.S. da Silva, M.S. Silveira, C. de O. Melo, L.C. Parzianello, *Understanding the UX Designer's Role within Agile Teams*, in: Marcus, A. (ed.), *Design, User Experience, and Usability: Design Philosophy, Methods, and Tools*, Berlin, Heidelberg, 2013, LNCS 8012 - Part I, Springer Berlin Heidelberg, pp. 599–609.
- [35] S. standardisoimisliitto, *Ihmisen ja järjestelmän vuorovaikutuksen ergonomia. Osa 210: Vuorovaikutteisten järjestelmien käyttäjäkeskeinen suunnittelu*, Suomen standardisoimisliitto, SFS-ISO 9241-210, Helsinki, 2010, 66 s.
- [36] J. Sutherland, *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process*, Oct. 2015. Saatavissa (viitattu 10.10.2018): https://www.researchgate.net/publication/242437392_The_Scrum_Papers_Nuts_Bolts_and_Origins_of_an_Agile_Process
- [37] D. Sy, *Adapting Usability Investigations for Agile User-centered Design*, *J. Usability Studies*, Vol. 2, Iss. 3, May 2007, pp. 112–132.
- [38] Tilastokeskus – Tietoa tilastoista, *Mikroyritys*, Tilastokeskus, verkkosivu. Saatavissa (viitattu 13.11.2018): <https://www.stat.fi/meta/kas/mikroyritys.html>
- [39] Tilastokeskus – Tietoa tilastoista, *Käsitteet, PK-yritykset*, Tilastokeskus, verkkosivu. Saatavissa (viitattu 13.11.2018): https://www.stat.fi/meta/kas/pienet_ja_keski.html
- [40] J. Tuomi, A. Sarajärvi, *Laadullinen tutkimus ja sisällönanalyysi*, 5. uud. laitos p., Tammi, Helsinki, 2009, 175 s.
- [41] E. Vidal, *The rise of the “Full Stack” Designer and the tools they use*, Medium, verkkosivu, 2015. Saatavissa (viitattu 5.10.2018): <https://medium.com/@edenvidal/the-rise-of-the-full-stack-designer-and-the-tools-he-uses-3daf015eb3fc>

5. Rooli, jossa työskentelet pääasiassa? *

Merkitse vain yksi soikio.

- UX/UI-suunnittelija (tai vastaava)
- Ohjelmistokehittäjä (tai vastaava)
- Muu: _____

Kiinnostus ja osaaminen

Tehtävien määritelmät

UX-suunnittelu: Käyttäjätutkimus, rautalankamallit, tietoarkkitehtuuri.

UI-suunnittelu: Tarkat näkymät, käyttöliittymäelementit.

Vuorovaikutussuunnittelu: Prototyypit, animaatio, interaktiot.

Käyttöliittymän toteutus: Graafisen käyttöliittymän toteutus käyttäen projektissa käytössä olevia teknologioita esim. front-end-teknologioita kuten HTML, CSS, JavaScript.

6. Kuinka kiinnostunut olet työskentelemään seuraavien tehtävien parissa? *

Merkitse vain yksi soikio riviä kohden.

	1 - En ollenkaan kiinnostunut	2 - Vain vähän kiinnostunut	3 - Jonkin verran kiinnostunut	4 - Erittäin kiinnostunut
UX-suunnittelu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
UI-suunnittelu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Vuorovaikutussuunnittelu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Käyttöliittymän toteutus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7. Kuinka monen vuoden kokemus sinulla on seuraavista tehtävistä? *

Työkokemus tai sitä vastaava kokemus.

Merkitse vain yksi soikio riviä kohden.

	Ei kokemusta	alle 1 vuotta	1 - 3 vuotta	3 - 5	5 - 10 vuotta	Yli 10 vuotta
UX-suunnittelu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
UI-suunnittelu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Vuorovaikutussuunnittelu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Käyttöliittymän toteutus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Väittämiä suunnittelusta ja ohjelmistokehityksestä

Alla on esitetty suunnitteluun ja ohjelmistokehitykseen liittyviä väittämiä.

10. Mitä mieltä olet seuraavista väittämistä? *

Merkitse vain yksi soikio riviä kohden.

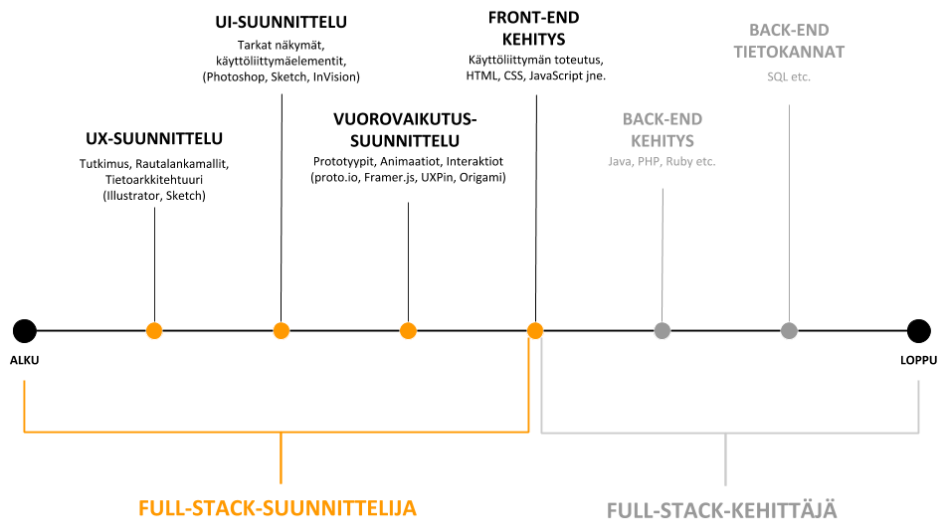
	Täysin eri mieltä	Jokseenkin eri mieltä	Ei samaa, eikä eri mieltä	Jokseenkin samaa mieltä	Täysin samaa mieltä	En osaa sanoa
Suunnittelijat ovat enemmän kehitystiimin jäseniä, jos suunnittelijat osallistuvat ohjelmistokehitykseen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Suunnittelija on tyytyväisempi, jos suunnittelija vastaa koko suunnitteluprosessista tutkimuksesta toteutukseen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Suunnittelijan ymmärrys ohjelmistokehityksestä parantaa designin laatua.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Full-stack-suunnittelijan rooli

Full-stack-suunnittelijan määritelmä

Full-stack-suunnittelijalla tarkoitetaan sellaista henkilöä, jolla on osaamista kaikilta tuotesuunnittelun osa-alueilta. Full-stack-suunnittelija on siis suunnittelija, joka osaa tarvittaessa viedä itsenäisesti koko tuotesuunnitteluprosessin alusta loppuun, käyttäjätutkimuksesta designin toteutukseen. Roolin tehtävät/osaaminen on kuvattu tiivistetysti alla olevassa kuvassa.

Full-stack-suunnittelijan tehtäviä tuotekehityksessä web-projektin mukaan esitettynä



Ruudunlukijoille:

Full-stack-suunnittelija

UX-suunnittelu: Käyttäjätutkimus, rautalankamallit, tietoarkkitehtuuri.

UI-suunnittelu: Tarkat näkymät, käyttöliittymäelementit.

Vuorovaikutussuunnittelu: Prototyypit, animaatio, interaktiot.

Front-end-kehitys: Käyttöliittymän toteutus, esim. HTML, CSS, JavaScript.

(Roolien osittainen päällekkäisyys)

Full-stack-kehittäjä

Front-end-kehitys: Käyttöliittymän toteutus, esim. HTML, CSS, JavaScript.

Back-end-kehitys: Java, PHP, Ruby jne.

Back-end-tietokannat: SQL jne.

Roolin tunnettavuus

11. Olitko aiemmin kuullut termiä "full-stack-suunnittelija" (engl. full-stack designer)? *

Merkitse vain yksi soikio.

Kyllä *Siirry kysymykseen 12.*

En *Siirry kysymykseen 14.*

Mielipide määritelmästä

Full-stack-suunnittelijalla tarkoitetaan sellaista henkilöä, jolla on osaamista kaikilta tuotesuunnittelun osa-alueilta. Full-stack-suunnittelija on siis suunnittelija, joka osaa tarvittaessa viedä itsenäisesti koko tuotesuunnitteluprosessin alusta loppuun, käyttäjätutkimuksesta designin toteutukseen.

12. Oletko samaa mieltä roolin määritelmästä?

Merkitse vain yksi soikio.

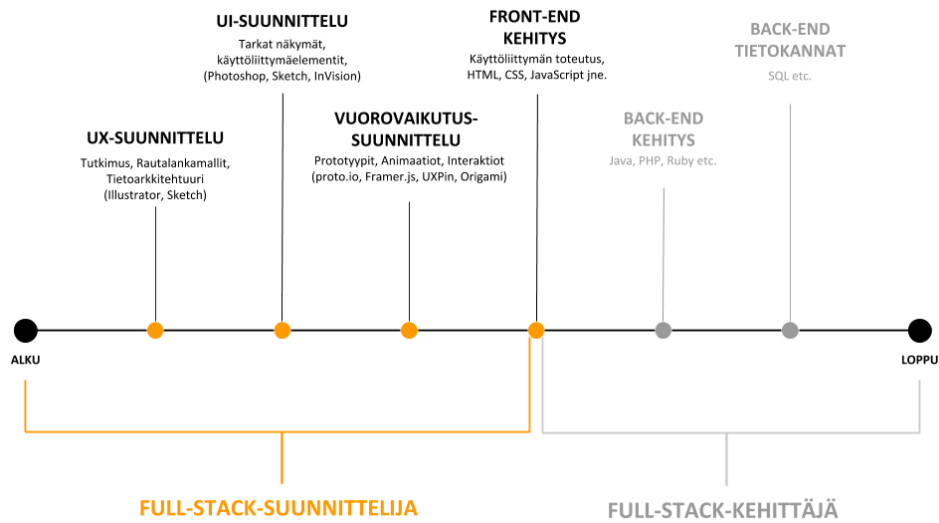
Kyllä

En

En osaa sanoa

13. Jos et ole samaa mieltä roolin määritelmästä, kuinka muuttaisit roolin määritelmää?

Hyödyt ja haasteet full-stack-suunnittelijan roolista



Full-stack-suunnittelija:

UX-suunnittelu: Käyttäjätutkimus, rautalankamallit, tietoarkkitehtuuri.

UI-suunnittelu: Tarkat näkymät, käyttöliittymäelementit.

Vuorovaikutussuunnittelu: Prototyypit, animaatio, interaktiit.

Front-end-kehitys: Käyttöliittymän toteutus, esim. HTML, CSS, JavaScript.

14. Mitä seuraavista tehtävistä olet tehnyt saman projektin aikana?

Jos et ole tehnyt näitä tehtäviä projekteissa, voit jättää vastaamatta kysymyksen.

Valitse kaikki sopivat vaihtoehdot.

- UX-suunnittelu
- UI-suunnittelu
- Vuorovaikutussuunnittelu
- Käyttöliittymän toteutus

15. Oletko joskus työskennellyt projektissa, jossa joku tiimin jäsen on toiminut full-stack-suunnittelijan roolissa tai tehnyt roolia vastaavia tehtäviä? *

Merkitse vain yksi soikio.

- Kyllä
- En

Vastaa seuraaviin kysymyksiin joko perustaen omiin henkilökohtaisiin kokemuksiisi tai siihen, miten kuvittelisit asioiden olevan projektissa, jossa joku tiimin jäsen työskentelee full-stack-suunnittelijan roolissa.

16. Mitä hyötyjä roolista on omasta mielestäsi?

17. Mitä haasteita roolista on omasta mielestäsi?

18. Miten ratkaisisit mahdolliset haasteet?

19. Olisiko full-stack-suunnittelijan mahdollista työskennellä täysin roolinsa mukaisesti yrityksessänne / organisaatiossanne hyödyntäen täyden potentiaalinsa? *

Merkitse vain yksi soikio.

Kyllä

Ei

20. Jos vastasit ei, perustele, miksi tämä ei olisi yrityksessänne / organisaatiossanne mahdollista.

21. Mitä mieltä olet seuraavista full-stack-suunnittelijan roolia koskevista väittämistä? *

Merkitse vain yksi soikio riviä kohden.

	Täysin eri mieltä	Jokseenkin eri mieltä	Ei samaa, eikä eri mieltä	Jokseenkin samaa mieltä	Täysin samaa mieltä	En osaa sanoa
Roolilla on liian monta vastuualuetta	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Full-stack-suunnittelija tekisi pelkästään designeja, jotka osaisi itse kehittää.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Roolin kuvaukseen sopivia työntekijöitä on vaikea löytää	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Rooli onnistuu ainoastaan web-projekteissa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

22. Kuinka kiinnostunut olisit itse työskentelemään tässä roolissa? *

Merkitse vain yksi soikio.

	1	2	3	4	5	
En ollenkaan kiinnostunut	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Erittäin kiinnostunut

23. Perustelut, miksi olisit tai et olisi kiinnostunut

24. Muuta kommentoitavaa full-stack-suunnittelijan rooliin liittyvistä asioista?

Omat tiedot

25. Syntymävuosi *

26. Sukupuoli *

Merkitse vain yksi soikio.

- Nainen
- Mies
- X
- En halua vastata

27. Kuinka monta vuotta olet työskennellyt IT-alalla? Anna vastaus numeroilla. *

28. Korkein suorittamasi tutkinto? *

Merkitse vain yksi soikio.

- Peruskoulu
- Ammattikoulututkinto tai muu ammatillinen koulutus
- Lukio, ylioppilastutkinto tai vastaava
- Alempi ammattikorkeakoulututkinto (AMK, Kandidaatti)
- Ylempi korkeakoulututkinto (YAMK, Maisteri)
- Lisensiaatti tai tohtori
- Ei tutkintoa
- Muu: _____

29. Palautetta kyselystä

Palvelun tarjoaa

