



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MARIE-ELISE KONTRO
SULAUTETTU OHJELMISTOTUOTANTO JA
KÄYTTÄJÄKOKEMUSTYÖ – TEHOKKUUTTA KETTERISTÄ
MENETELMISTÄ

Diplomityö

Tarkastaja: Professori Kaisa Väänänen
Tarkastaja ja aihe hyväksytty
Tieto- ja sähkötekniikan
tiedekuntaneuvoston kokouksessa
4. toukokuuta 2016

TIIVISTELMÄ

MARIE-ELISE KONTRO

Tampereen teknillinen yliopisto

Diplomityö, 83 sivua, 4 liitesivua

Kesäkuu 2016

Tietotekniikan diplomi-insinöörin tutkinto-ohjelma

Pääaine: User Experience

Tarkastaja: Professori Kaisa Väänänen

Avainsanat: Sulautetut järjestelmät, sulautettu ohjelmistotuotanto, ketterä kehitys, agile, käyttäjäkokemustyö, käyttäjäkokemus, käytettävyys, UX

Sulautettujen järjestelmien käyttö on nykyään ihmisille arkipäivää. Vuonna 2014 69 prosentilla suomalaisista oli käytössään älypuhelin, eli eräänlainen sulautettu järjestelmä (Tilastokeskus 2014). Sulautetun järjestelmän tuottaminen vaatii monia erikoistuneita tekniikan aloja. Näiden yhdistäminen toimivaksi kokonaisuudeksi vaatii asiantuntevaa prosessinhallintaa. Laitteisto tehdään erityispiirteidensä vuoksi edelleen pääosin suunnitelmaohjautuvasti, mutta prosessiin kaivataan ohjelmistotuotannossa yleistynyttä ketteryyttä.

Käyttäjäkokemuksella tarkoitetaan käyttäjälle syntyvää kokemusta tuotteesta ja sen käytöstä, tai odotetusta käytöstä. Käyttäjäkokemustyön yhdistäminen ohjelmistoprojektiin on haasteellista kaikenlaisissa projekteissa, eli myös silloin kun tuotetaan sulautettuja järjestelmiä. Tutkimusta käyttäjäkokemustyön tekemisestä sulautettujen järjestelmien valmistusprosessissa on tehty vain vähän.

Tässä tutkimuksessa selvitettiin sulautettuja järjestelmiä valmistavien yritysten ohjelmistotuotannon käytäntöjä ja käyttäjäkokemustyön toteuttamista. Tutkimuksessa etsittiin tapoja kehittää yritysten olemassa olevia tuotantoprosesseja ja määritellä käyttäjäkokemustyölle paikka tuotannon prosessikaaviossa.

Tutkimus suoritettiin haastattelemalla kahden suuren työkoneita ja kuluttajatuotteita valmistavan yrityksen työntekijöitä. Tutkimuksessa oli mukana yhteensä 11 työntekijää monenlaisista rooleista. Haastattelujen jälkeen aineistosta kerättiin sekä yrityksen toiminnalle haasteita aiheuttavat toimintatavat että hyviksi koetut käytännöt.

Yritysten suurimmiksi ongelmiksi tutkimuksessa nousivat käyttäjäkokemustyöhön, tiedonkulkuun ja kommunikointiin, prosessimalliin, uudelleenkäytettävyyteen ja projektin johtamiseen liittyvät asiat. Ratkaisuksi ongelmiin ehdotettiin ketteristä menetelmistä tuttuja käytänteitä esimerkiksi itseorganisoituvia tiimejä ja loppukäyttäjien sitouttamista osaksi tuotekehitysprosessia. Lisäksi käyttäjäkokemustyöhön liittyviin ongelmiin etsittiin ratkaisuja ketterän käyttäjäkokemustyön malleista.

ABSTRACT

MARIE-ELISE KONTRO

Tampere University of Technology

Master of Science Thesis, 83 pages, 4 Appendix pages

June 2016

Master's Degree Programme in Information Technology

Major: User Experience

Examiner: Professor Kaisa Väänänen

Keywords: Embedded Systems, Embedded Software Engineering, Agile Software Engineering, Agile, User Experience Work, User Experience, Usability, UX

Modern people use embedded software every day. In 2014, 69 % of Finnish people had access to a smartphone, that is one example of an embedded system (Tilastokeskus 2014). Production of embedded systems requires many specialized fields of engineering: hardware, software, and mechanics at least. A good process management is needed to combine the work of different fields of technology. In the field of software engineering the use of agile methodologies has increased. Hardware production instead is still often plan-based due to its specific characteristics.

User experience refers to person's perception of the product and the use, or anticipated use of it. Combining User Experience work to software engineering is not simple, as many of the problems found in software projects can also be found with embedded systems. There is no good research coverage yet about user experience work in the field of embedded software development.

In this study, software engineering practices and user experience work of two companies were examined. The study aimed at finding ways to develop the companies' existing production process and to define place for user experience work in the process diagram for developing embedded software.

The study was conducted by interviewing employees of two big companies (over 250 employees). The companies operate in the field of industry equipment and small electronics. The study included a total of 11 employees working in different stages of projects. Based on the interviews, challenges and existing good practices in the current processes were highlighted and categorized. Findings were compared to the scientific literature and solutions were proposed.

Most considerable problems in the companies were related to user experience work, communication, process model, reusability, and process management. Solutions for the problems were proposed based on practices used in well-known agile methods. For instance, self-organized teams and engaging end-users to product development process were suggested. Solutions were also searched from agile user experience frameworks.

ALKUSANAT

Tämä diplomityö tehtiin Tampereen teknillisellä yliopistolla, Tietotekniikan laitoksella. Tutkimukseen osallistui kaksi yritystä. Kiitän molempia yrityksiä tasapuolisesti hyvästä yhteistyöstä. Haluan myös kiittää kaikkia tutkimukseen osallistuneita avoimesta suhtautumisesta tutkimukseen

Haluan kiittää Väänäsen sukuseura Ry:tä diplomityön tueksi annetusta stipendistä, jonka sain vuonna 2013.

Lisäksi haluan kiittää työn tarkastajaa Kaisa Väänästä nopearytmisestä tarkastusprosessista ja arvokkaista ehdotuksista työni parantamiseksi. Suuret kiitokset myös diplomityöni ohjaajalle Kati Kuusiselle hänen kärsivällisestä suhtautumisestaan pitkäaikaiseen, niin kirjoittamisen kuin elämänopettelun prosessiini. Kiitän häntä myös tarkasta ja asiantuntevasta palautteen antamisesta ja aidosta kiinnostuksesta työn loppuunsaattamista kohtaan.

Kiitokset valmistumisestani kuuluvat myös ystäväilleni. Laura: Ilman ystävyttäsi koulutieni TTY:llä olisi luultavasti päättynyt lyhyeen. Saila, kiitos matematiikan tukiopetuksesta viimeisenä vuonna. Kiitos myös Saanareetta, Hanna, Jenni ja Tiia. Olette kaikki olleet minulle tärkeä tuki yliopistovuosina.

Kiitokset perheelle tiedonjanoni herättämisestä, vapaudesta valita oma tieni, persoonani ehdottomasta tukemisesta ja ainaisesta turvasta puhelinsoiton päässä.

Ja viimeisimpänä Markus. Iloni, tukeni, rauhani, turvani. Kiitos.

Tampereella, 24.05.2016

Marie-Elise Kontro

SISÄLLYSLUETTELO

1.	JOHDANTO	1
1.1	Taustaa	1
1.2	Tutkimusongelma ja aiheen rajaus	2
1.3	Työn rakenne	2
2.	SULAUTETTUJEN JÄRJESTELMIEN OHJELMISTOTUOTANTO.....	3
2.1	Sulautettu ohjelmisto osana sulautettua järjestelmää	3
2.2	Sulautettujen järjestelmien erityispiirteitä	4
2.3	Suunnitelmaohjautuva sulautettujen järjestelmien kehitysmenetelmä	5
2.4	Suunnitelmaohjautuvien kehitysmenetelmien ongelmakohtia	7
2.5	Ketteryys ohjelmistotuotannossa	8
2.6	Ketteryys sulautetussa ohjelmistotuotannossa.....	9
2.6.1	Ketteryyden haasteet ja mahdollisuudet.....	11
3.	KÄYTTÄJÄKOKEMUSTYÖ OHJELMISTOPROJEKTISSA	13
3.1	Mitä on käyttäjäkokemus?	13
3.2	Käyttäjäkeskeinen suunnitteluprosessi	14
3.3	Ketteryys käyttäjäkeskeisessä ohjelmistoprojektissa	15
3.3.1	"Yksi pyrähdys edellä" -malli	15
3.3.2	BoB (Best of Both Worlds) -viitekehys	16
3.3.3	Lean UX	18
4.	TUTKIMUSMENETELMÄT.....	20
4.1	Tutkimuksen tavoitteet	20
4.2	Tutkimusmenetelmän valinta.....	20
4.2.1	Monitapaustutkimus	20
4.3	Tiedonkeruumenetelmä	21
4.3.1	Haastatteluympäristö, aineiston tallennus ja haastateltavat	22
4.4	Aineiston käsittely ja analyysimenetelmä	23
4.5	Raportointi	24
4.6	Tutkimuksen luotettavuuden arviointiin käytetyt mittarit	24
4.6.1	Luotettavuus kvalitatiivisen tutkimuksen piirissä?	24
4.6.2	Tutkimuksen sisäinen ja ulkoinen validiteetti	25
4.6.3	Tutkimuksen reliabiliteetti	25
4.6.4	Muita kvalitatiivisen tutkimuksen arviointimittareita	26
4.7	Tutkimuksen yleistettävyys	26
4.8	Tutkimusetiikka	27
5.	TULOKSET: YRITYS 1.....	29
5.1	Kuvaus	29
5.2	Tutkimuksen tulokset.....	30
5.2.1	Käyttäjäkokemustyöhön liittyvät haasteet	30
5.2.2	Projektin vaiheisiin ja synkronointiin liittyvät haasteet	33

5.2.3	Tiedonsiirtoon ja kommunikointiin liittyvät haasteet	37
5.2.4	Projektin johtamiseen ja työn organisointiin liittyvät haasteet	39
6.	TULOKSET: YRITYS 2.....	43
6.1	Kuvaus	43
6.2	Tutkimuksen tulokset.....	43
6.2.1	Käyttäjäkokemustyöhön liittyvät haasteet	44
6.2.2	Projektinhallintaan liittyvät haasteet	46
6.2.3	Esitutkimus- ja määrittelyvaiheeseen liittyvät haasteet	48
6.2.4	Testaukseen liittyvät haasteet.....	51
6.2.5	Alihankintaan liittyvät haasteet.....	53
7.	TUTKIMUSTULOSTEN KOONTI	56
7.1	Tulosten vertailu kirjallisuudessa esiintyneisiin ongelmiin.....	56
7.2	Yhteenveto haasteista ja ratkaisuehdotuksista.....	57
7.3	Ehdotettu malli hyvistä käytännöistä.....	61
7.3.1	Esitutkimus ja määrittelyvaihe	62
7.3.2	Toteutusvaihe	63
7.3.3	Järjestelmätestausvaihe	64
7.3.4	Käyttöönotto- ja ylläpitovaihe.....	64
7.4	Parhaat ketterät käytännöt.....	65
8.	JOHTOPÄÄTÖKSET	66
8.1	Pohdinta	66
8.2	Tutkimuksen luotettavuuden arviointi.....	70
8.2.1	Tutkimuksen validiteetti ja reliabiliteetti	71
8.2.2	Muut mittarit	73
8.3	Tutkimusetiikka	74
8.4	Jatkotutkimuskysymyksiä.....	75
8.5	Yhteenveto.....	76
	LÄHTEET	78

LIITE A: TUTKIMUSSUOSTUMUSLOMAKE

LIITE B: TAUSTATIETOKYSELY

LIITE C: HAASTATTELUKYSYMYKSET

LYHENTEET JA MERKINNÄT

HW	engl. Hardware, laitteisto
IoT	engl. Internet of Things, esineiden internet
MVP	engl. Minimum Viable Product, yksinkertaisin toimiva tuote
SW	engl. Software, ohjelmisto
UX	engl. User Experience, käyttäjäkokemus

1. JOHDANTO

Tässä luvussa esitellään työn aihe, tutkimuskysymykset ja aiheen rajausta. Lisäksi käydään läpi työn rakenne.

1.1 Taustaa

Tämän diplomityön aihe nousi kandidaatintyöni aineistosta. Kandidaatintyöni käsitteli Scrumin käyttöä ja käyttäjäkokemustyötä eräässä suomalaisessa ohjelmistoalan yrityksessä. Koska yritys tuotti perinteisen ohjelmiston lisäksi myös sulautettua ohjelmistoa, nousi tutkimusaineistosta kysymykseksi miten projekti, jossa tehdään sekä sulautettua ohjelmistoa että laitteistoa, saataisiin sujuvaksi. Ohjelmistoa ja laitteistoa sisältävän tuotteen kehittäminen on haaste itsessään. Käyttäjäkokemuksen lisääminen yhtälöön mutkistaa prosessia entisestään. Tämä aihe alkoi kiinnostaa minua, ja päätin jatkaa diplomityöni aiheesta. Aihe on kuitenkin hieman muokkautunut yritysten toiveiden mukaisesti.

Aiheen kiinnostavuuden lisäksi valintaan vaikutti lisäksi se, ettei aiheesta löydy juurikaan tutkimusta. Sulautettujen järjestelmien ketteryttä käsitellään joissakin artikkeleissa, mutta käyttäjäkokemustyön tekemisestä sulautettujen järjestelmien tuotantoprosessissa on hyvin vähän tutkimustietoa. Tutkimusaihe on lisäksi hyvin opettavainen, sillä aihetta ei juurikaan käsitellä ohjelmistotuotannon kurseilla. Tämä on sääli, sillä esimerkiksi mobiililaitteiden kehitys viime vuosina on ollut päätähuimaavaa, ja monelle laitevalmistajalle kynnyskysymyksiksi ovat nousseet niin kilpailun edellyttämä nopea julkaisusykli kuin hyvän käyttäjäkokemuksen tarjoamiseen liittyvät haasteet.

Tutkimuksessa oli mukana kaksi suurta työkoneita ja kuluttajaelektroniikkaa tuottavaa yritystä. Ensimmäinen tutkimus koski läpimenoajan lyhentämistä ja käyttäjäkokemustyön luonnetta projektissa, jossa tehdään sulautettua ohjelmistoa. Toisessa tutkimuksessa päivitettiin yrityksen sulautetun ohjelmiston tuotantoprosessikuvausta, tarkennettiin käyttäjäkokemustyön paikkaa kuvauksessa ja pyrittiin prosessin tehostamiseen. Näiden tutkimusten pohjalta diplomityöhön on valittu aineisto, joka käsittelee sulautetun ohjelmistoprosessin tehostamista ja käyttäjäkokemustyön paikan määrittelyä prosessissa. Ketterät menetelmät tarjoavat vastauksia moniin ohjelmistoprosesseissa esiintyviin ongelmiin, siksi ketterien menetelmien tarjoamat työtavat on otettu tässä työssä keskeiseen tarkasteluun.

Tutkimusmenetelmä, jota tässä työssä on käytetty, on monitapaustutkimus ja aineistonkeruumenetelmänä käytettiin haastattelua. Tapaustutkimus valittiin tutkimusmenetel-

mäksi, koska tutkimuksen tavoitteena on lisätä niin työntekijöiden kuin johdonkin tietämystä yrityksen omasta toiminnasta, ja löytää tapoja kehittää toimintaa.

1.2 Tutkimusongelma ja aiheen rajaus

Sulautettu ohjelmisto on osa sulautetun järjestelmän tuottavaa prosessia (Järvinen & Mikkonen 2012). Prosessiin liittyy ohjelmiston lisäksi monia muitakin erikoistuneita aloja, jotka toimivat muuttujina osana isompaa kokonaisprosessia. Tutkimuksen tavoitteena on selvittää miten sulautettua ohjelmistoa tehdään ja miten käyttäjäkokemustyön tekeminen näkyy ja mihin se sijoittuu tuotantoprosessissa.

Tutkimuskysymykset ovat:

1. Millaisia ongelmia yritykset kohtaavat sulautettujen järjestelmien tuotannossa?

2. Millaisia ongelmia käyttäjäkokemustyön tekemiseen liittyy sulautettujen järjestelmien ohjelmistotuotannossa?

3. Millaiset ketterät käytännöt voivat tehostaa sulautettujen järjestelmien ohjelmistotuotantoprosessia, jossa tehdään myös käyttäjäkokemustyötä?

Sulautetut järjestelmät pitävät sisällään laitteistoa, ohjelmistoa ja mekaniikkaa. Tämän lisäksi prosessissa tehdään käyttäjäkokemustyötä. Tässä työssä tarkastellaan tätä kokonaisuutta sulautettujen järjestelmien ohjelmistotuotannon kannalta. Haastateltavat on valittu laitteisto- ja ohjelmistokehittäjistä, projektipäälliköistä sekä käyttäjäkokemustyöntekijöistä. Tämä tarkoittaa sitä, että esimerkiksi mekaniikkaa käsittelevät asiat on jätetty työn ulkopuolelle, vaikka ne olennaisesti kuuluvat sulautettujen järjestelmien tuotantoprosessiin. Työssä on kuitenkin käsitelty sulautettujen järjestelmien teoriaa, koska se on kokonaisprosessin kautta olennaista.

1.3 Työn rakenne

Luvut 2-4 luovat teoriapohjan tämän työn tutkimusosuudelle. Luvussa 2 kerrotaan yleisesti ohjelmistotuotannosta ja lisäksi ohjelmistotuotannosta osana sulautetun järjestelmän tuotantoprosessia. Luvussa 3 pohditaan mitä on käyttäjäkokemus ja millaisia malleja käyttäjäkokemustyön tekemiseen ohjelmistoprosessissa on ehdotettu. Luvussa 4 kuvataan käytetty tutkimusmenetelmä.

Luvuissa 5 ja 6 on tutkimuksen pohjana olevien kahden tapaustutkimuksen kuvaukset ja tulokset. Luku 7 kokoaa kahden tapaustutkimuksen tulokset ja vastaa esitettyihin tutkimuskysymyksiin. Luvussa 8 kootaan johtopäätökset ja pohditaan tutkimuksen luotettavuutta.

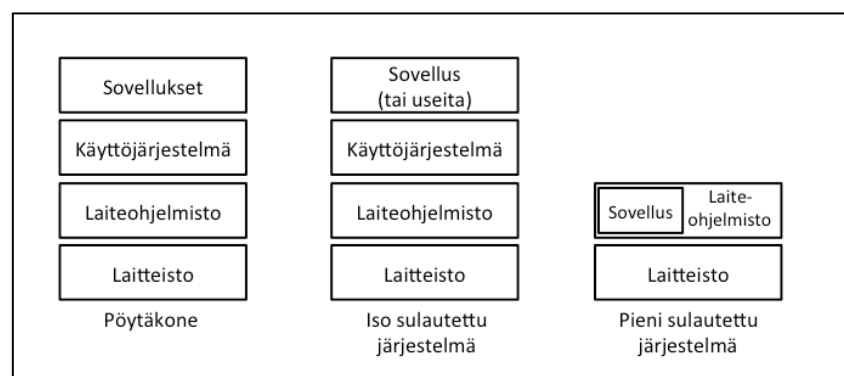
2. SULAUTETTUJEN JÄRJESTELMIEN OHJELMISTOTUOTANTO

Kohdassa 2.1 määritellään sulautetut ohjelmistot ja sulautetut järjestelmät. Kohdassa 2.2 kerrotaan miten sulautettuja ohjelmistoja on perinteisesti tuotettu ohjelmistoprojekteissa. Kohdassa 2.3 käydään läpi uusia tapoja sulautetun järjestelmän kehitykseen.

2.1 Sulautettu ohjelmisto osana sulautettua järjestelmää

Perinteisellä ohjelmistolla tarkoitetaan erilaisia tietokoneissa toimivia sovellusohjelmia, kuten laskin, tekstinkäsittelyohjelma, internetselain, jne. Tietokoneiden lisäksi ohjelmia on myös useissa laitteissa, kuten matkapuhelimissa ja kodinkoneissa. Tällaisia laitteita kutsutaan sulautetuiksi järjestelmiksi. Järvisen ja Mikkosen mukaan (2012, s. 11) sulautetulla järjestelmällä tarkoitetaan *"sellaista järjestelmää, jossa tietokoneen laitteisto ja ohjelmisto liittyvät toisiinsa saumattomasti"*. Yhdessä saumattomasti toimivia järjestelmän osia voi olla myös enemmän kuin kaksi. Turun yliopiston sulautettujen järjestelmien käsikirja (2014, s. 10) tarkentaakin sulautetun järjestelmän käsitteen tarkoittamaan *"jonkin tehtävän suorittamiseen suunnitelluksi laitteeksi, joka sisältää elektroniikkaa, mahdollisesti mekaniikka sekä laitteelle erityisesti suunniteltua ohjelmistoa"*.

Sulautetussa järjestelmässä toimivaa ohjelmistoa kutsutaan sulautetuksi ohjelmistoksi. Sanastokeskuksen termipankki (2014) määrittelee sulautetun ohjelmiston *"sulautetussa järjestelmässä olevaksi ohjelmistoksi, joka on asennettu kiinteästi tietokoneeseen ja jolla toteutetaan tietokoneen perustoimintoja"*. Tieteen termipankki (2014) määrittelee laitteiston *"tietokonejärjestelmään kuuluvista laitteista muodostuvaksi kokonaisuudeksi."*



Kuva 1: Pöytäkoneen sekä ison ja pienen sulautetun järjestelmän rakenteen vertailu. (Ihalainen, 2007).

Sulautettu järjestelmä muodostuu siis ainakin ohjelmistosta ja laitteistosta. Kuvassa 1 verrataan sulautetun järjestelmän rakennetta perinteiseen, esimerkiksi pöytätietokoneen rakenteeseen (Ihalainen 2007). Lähimpänä laitteistoa on laiteohjelmisto (engl. firmware), joka alustaa laitteiston käynnistyksen yhteydessä. Laiteohjelmisto mahdollistaa käyttöjärjestelmän toiminnan, jossa voidaan ajaa sovelluksia. Pienessä sulautetussa järjestelmässä ei välttämättä ole muuta ohjelmistoa kuin laiteohjelmisto.

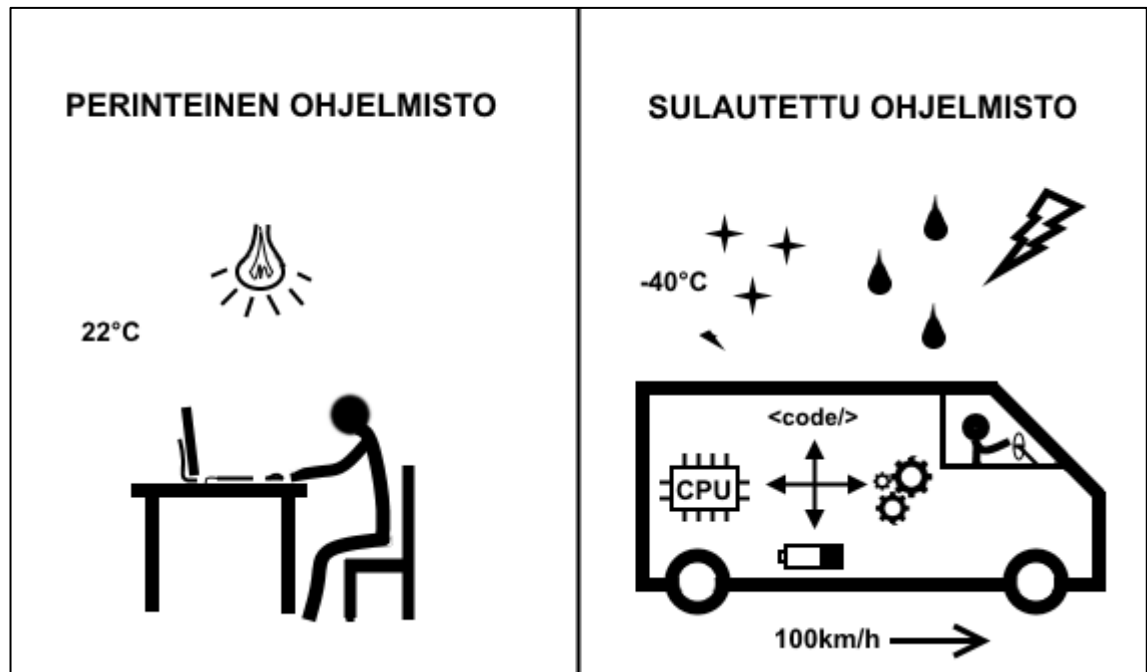
Hyvä esimerkki sulautetusta järjestelmästä on nykyaikainen mobiililaitte, jota ihmiset käyttävät päivittäin. Laite sisältää monenlaista tekniikkaa ja sen sisällä on monimutkainen tietokone, mutta käyttäjien ei tarvitse tietää tuon tietokoneen olemassaolosta. Nykyään oman lisänsä sulautettuihin järjestelmiin tuo laajasti saatavilla oleva internet-yhteys. Sulautetut järjestelmät, esimerkiksi televisiot ja työkoneet voivat olla liitettynä internetiin, jolloin puhutaan esineiden internetistä eli IoT:sta (engl. Internet of Things). IoT on sulautetuista järjestelmistä koostuva internet.

2.2 Sulautettujen järjestelmien erityispiirteitä

Sulautettujen järjestelmien koko vaihtelee suuresti. Pienimmät laitteet voivat olla esimerkiksi ihmiseen asennettavia biosensoreita (Ebert 2009) ja suurimmat lentokoneita (Järvinen & Mikkonen 2012). Erityisen tyypillistä sulautetuille järjestelmille on, ettei käyttäjän tarvitse tietää laitteen sisällä olevan tietokoneen olemassaolosta (ibid.). Esimerkiksi painamalla nappia tapahtuu toiminto, jonka syvemmästä logiikasta käyttäjän ei tarvitse tietää mitään. Tärkeintä on, että laite toimii käyttäjän kannalta odotetusti.

Perinteisesti sulautetut järjestelmät ovat olleet suljettuja (Järvinen & Mikkonen 2012), eli laitteen sisäistä logiikkaa voi muuttaa ainoastaan järjestelmän toimittaja. Toisin sanoen, niin sanottu kolmas osapuoli ei voi muuttaa tai lisätä siihen omia ohjelmiaan. Tässä on kuitenkin tapahtumassa muutos (ibid.), sillä osa uusista ohjelmointiympäristöistä tarjoaa myös asiakkaalle rajapinnan järjestelmän logiikan muuttamiseen ja omien sovellusten lisäämiseen. Esimerkiksi nykyaikaiset matkapuhelimien käyttöjärjestelmät tarjoavat tähän mahdollisuuden. Myös avoimen lähdekoodin käyttö on nopeasti yleistymässä sulautettujen järjestelmien kehityksessä. Esimerkiksi matkapuhelinten käyttöjärjestelmistä Symbian, Android ja Sailfish perustuvat ainakin osittain avoimeen lähdekoodiin.

Reaaliaikaisuus on myös yksi sulautettujen järjestelmien tunnuspiirteistä (Järvinen & Mikkonen 2012). Tämä tarkoittaa, että toiminnallisten vaatimusten lisäksi järjestelmällä on myös suoritusajaksi liittyviä vaatimuksia. Esimerkiksi käyttäjä saattaa toivoa, että laite vastaisi annettuihin käskyihin mahdollisimman nopeasti. Käyttäjän odottaessa vastausta saattaa koneen sisällä kuitenkin tapahtua useita käskyketjuja, joiden takia vasteaika pitenee. Voi olla jopa käyttäjän turvallisuuden takaamiseksi tarkoituksellista, ettei laite vastaa käyttäjän kommentoihin nopeasti. Laitevalmistajalle tällaiset ovat hankalia tapauksia, jos asiakkaalle ei voi antaa mitä asiakas haluaa.



Kuva 2: Sulautettujen järjestelmien käyttöympäristö voi olla hyvin erilainen kuin pöytäkoneen. Lisäksi sulautetut järjestelmät ovat monimutkaisempia kokonaisuuksia.

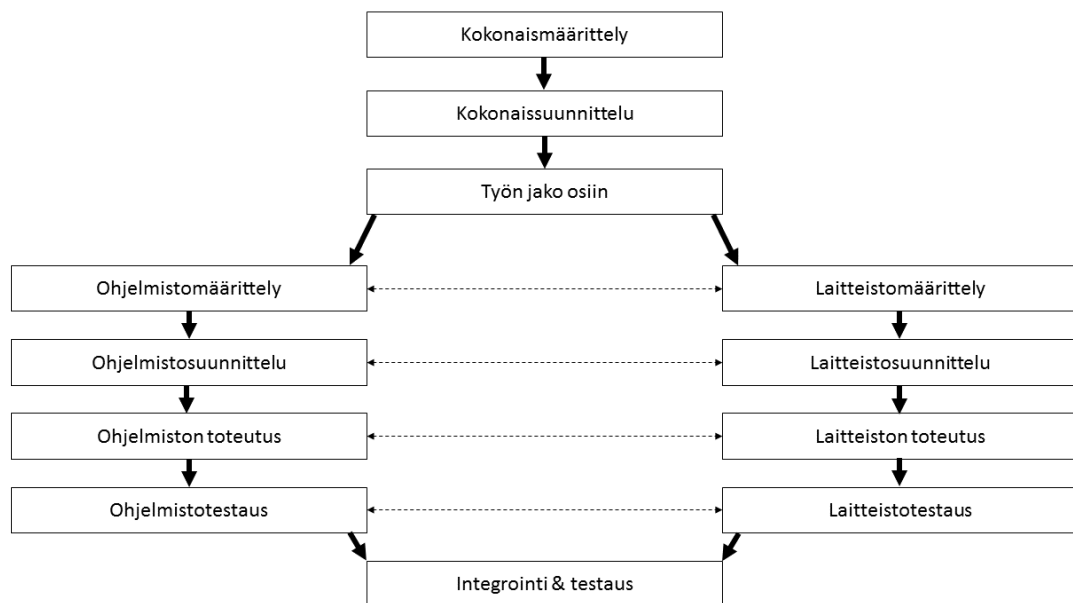
Sulautetuilta järjestelmiltä vaaditaan edellisten ominaispiirteiden lisäksi usein erityisen suurta turvallisuutta ja vikasietoisuutta (Järvinen & Mikkonen 2012). Myös käyttöympäristön olosuhteet voivat sulatetun järjestelmän tapauksessa olla hyvin poikkeukselliset perinteiseen järjestelmään verrattuna (Kuva 2). Sulautetun järjestelmän tulisi aina toimia niin, ettei se edes viallisena tai väärin ohjattuna aiheuta vaaraa käyttäjälle tai ympäristölle (ibid.). Lisäksi ohjelma pitäisi pystyä päivittämään joskus jopa niin, että käyttökätköt eivät ole sallittuja (ibid.). Esimerkiksi pesukone ei anna käyttäjän aukaista konetta, mikäli siellä on vettä sisällä. Lisäksi laitteen tulisi toipua virhetilanteista mahdollisimman automaattisesti, jopa niin, että käyttäjä ei huomaa virhettä tapahtuneen. Tästä esimerkkinä voisi antaa tilanteen, jossa sähköt menevät poikki pesukoneen ollessa päällä. Sähköjen palattua kone jatkaa siitä mihin jäätiin, eli edeltävä tilanne tallennetaan koneen muistiin ja siitä jatketaan tilanteen palautuessa. Sulautettuja järjestelmiä suunniteltaessa on siis tärkeää että ohjelmistokehittäjä tuntee sekä laitteen tulevan käyttöympäristön että sen tulevat käyttäjät saadakseen koneen reagoimaan kaikissa tilanteissa halutulla tavalla.

2.3 Suunnitelmaohjautuva sulautettujen järjestelmien kehitysmenetelmä

Perinteisessä ohjelmistotuotantoprosessissa tarvitsee yleensä ajatella vain ohjelmiston kehitystä, mutta sulautettujen järjestelmien tapauksessa tuotekehitystä vaaditaan monella erikoisosaamisalueella (Lehtonen et al. 2014). Kehitysprosessin suunnittelu voi olla hankalaa, koska kehitystyö eri osa-alueilla etenee eri tahdissa, eri osa-alueiden välillä

vaaditaan jatkuvaa kommunikointia ja kokonaisuutta pitäisi pystyä ohjaamaan ja hallitsemaan.

Kuvassa 3 on esitettyä Järvisen ja Mikkosen (2012) kirjassa esitetty yleinen suunnitelumalli sulautettujen järjestelmien kehitykseen. Malli näyttää perinteisessä ohjelmistotuotannossa käytetyltä vesiputousmallilta (Royce 1970), jossa vaiheet tehdään peräkkäin, mutta tuotevirtoja on vain yksi. Sulautetun järjestelmän tuottamisessa on otettava huomioon, että samassa prosessissa tuotetaan sekä ohjelmistoa että laitteistoa ja "vesiputous" on jaettu kahteen eri virtaan. Ohjelmiston ja laitteiston kehityspotkut ovat kuitenkin sulautettujen järjestelmien kehityksessä harvoin ainoat. Prosessiin liittyy usein myös esimerkiksi mekaniikan ja automaation suunnittelua, mikä ei näy oheisessa kuvassa.



Kuva 3. Yleinen prosessimalli sulautettujen järjestelmien kehitykseen (Järvinen & Mikkonen 2012).

Sulautettujen järjestelmien yleisessä kehitysmallissa (Kuva 3) järjestelmä suunnitellaan kokonaisuutena ja tämän jälkeen määrittely-, suunnittelu- ja testausvaihe tehdään ohjelmistolle ja laitteistolle erikseen omissa virroissaan. Lopussa virtojen tuotteet integroidaan ja testataan kokonaisuutena. Malli oheistaa myös katkonuolin laitteisto- ja ohjelmistopuolen jatkuvaan kommunikointiin.

Yleinen malli voi toimia edelleen hyvin tietynlaisissa projekteissa. Kettunen & Laanti (2005) antavat esimerkkinä "valtavat projektit", jotka vaativat hyvät suunnitelmat, ja paljon valvontaa. Lisäksi yleisen malli sopii pienempiinkin projekteihin, joissa viranomais- ja turvallisuusvaatimusten täyttymisestä ollaan erityisen tarkkoja.

2.4 Suunnitelmaohjautuvien kehitysmenetelmien ongelmakohtia

Sulautetut järjestelmät monimutkaistuvat jatkuvasti, siksi uudet näkemykset sulautettujen ohjelmistojen kehitykseen ovat tarpeen (Liggesmeyer & Trapp 2009). Aiemmin tilanne oli se, että laitteistokehittäjät saivat mallin (Kuva 3) mukaisesti toteutetussa prosessissa odottaa ohjelmiston valmistumista, jotta tuotantoprosessissa päästäisiin integrointivaiheeseen (Luciano et al. 1999). Nykyaikana tilanne on usein toinen. Laitteiston kehittäminen siihen vaiheeseen, että ohjelmisto voidaan lisätä siihen, vie usein kauemmin kuin ohjelmiston kehittäminen (Grenning 2002). Samaan aikaan kun laitteistoa kehitetään, ohjelmisto rakennetaan siis usein oletuksille siitä, miten se tulee laitteistossa toimimaan. Kun laitteisto tulee kokonaisuutena toimittajalta, ohjelmisto lisätään siihen ja aloitetaan testaus. Mahdolliset arkkitehtuuriset ongelmat, tai vaikkapa väärät valinnat laitteiston hankinnassa, ovat tässä vaiheessa lähes mahdottomia korjata (Kettunen & Laanti 2004). Testausvaiheessa huomattavat ongelmat ovat usein seurausta jo aiemmin prosessissa tapahtuneista laiminlyönneistä. Laitteiston muuttaminen pitkän suunnitelmallisen prosessin jälkeen tulee kalliiksi ja siksi ohjelmiston avulla usein korjataan laitteistossa vikoja eikä toisinpäin.

Liggesmeyer & Trapp (2009) lisäävät muutamia syitä sille, miksi perinteinen tapa tehdä ohjelmistoja ei sovi sulautettujen ohjelmistojen kehitykseen. Ensinnä, sulautetun ohjelmiston tuotannossa pitää huomioida, että sulautettu ohjelmisto on aina osa suurempaa kokonaisuutta (ibid.). Ohjelmiston ja laitteiston lisäksi sulautettu järjestelmä voi sisältää mekaniikkaa, elektroniikkaa ja sähkötekniikkaa (Lehtonen et al. 2014) ja jokaisella näistä erityisaloista on omat rajoitteensa, jotka kaikki vaikuttavat tuotteen kokonaisuuteen. Aiemmin esitetyssä prosessikuvauksessa (kuva 3) näkyy vain ohjelmiston ja laitteiston tuotevirrat. Tuotevirtoja voi prosessissa kuitenkin olla useita toisin kuin yksinkertaistettu kaavakuva antaa ymmärtää.

Toisena ongelmakohtana Liggesmeyer & Trapp (2009) mainitsevat sulautettujen järjestelmien monimuotoisuuden. Sulautettu järjestelmä voi yhtä hyvin olla lentokone, matkapuhelin, tai sydämentahdistin. Tällaisia laitteita ei voida rakentaa samankaltaisen alustan päälle, vaan jokainen laite tulee suunnitella perusteista asti uudelleen. Sen sijaan nykyaikana monet perinteiset sovellusohjelmistot voidaan rakentaa samalle alustalle, mikä helpottaa huomattavasti kokonaistuoteprosessia.

Näiden haasteita prosessimalliin aiheuttavien tekijöiden lisäksi Liggesmeyer & Trapp (2009) mainitsevat, että sulautetut järjestelmät ovat hyvin usein massatuotantoon suunnattuja tuotteita, joiden kustannusrajoitteet voivat olla hyvinkin tiukat. Kustannusrajoitteiden lisäksi pitää ottaa huomioon vielä esimerkiksi resurssirajoitteet ja ympäristöolosuhteet. Yksi sulautettujen järjestelmien erityispiirre onkin, että kehitystyö tehdään eri ympäristössä missä ohjelmisto tullaan testaamaan ja lopulta suorittamaan (Järvinen &

Mikkonen 2012). Testaaminen ei siis ole niin yksioikoista, kuin perinteisessä ohjelmistotuotannossa. Tämä tulee ottaa huomioon myös prosessimallissa.

Tavallisten kotitietokoneiden hyvä puoli on se, että ohjelmistot ovat riippumattomia laitteistosta (Henzinger & Sifakis 2006). Sulautetussa järjestelmässä internet aiheuttaa lisähaasteen, sillä käyttäjä saattaa odottaa internet-yhteyden toimivan kaikissa olosuhteissa samoin kuin esimerkiksi pöytätietokoneella. Kotitietokoneen tapauksessa käyttäjä ymmärtää laitteen toimivan oikein vain tietyissä olosuhteissa (esimerkiksi vain kun verkkojohto on kytkettynä).

2.5 Ketteruus ohjelmistotuotannossa

Ketterä kehitys on yhteisnimitys ohjelmistotuotannon menetelmille, jotka perustuvat Ketterän kehityksen julistukseen:

*”Kokemuksemme perusteella arvostamme:
Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja
Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota
Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja
Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa”
(Ketterän ohjelmistokehityksen julistus, 2001)*

Useimmat ketteristä menetelmistä pyrkivät minimoimaan riskejä ja jatkuvaa epävarmuutta jakamalla kehityksen pieniin iteraatioihin. Jokainen iteraatio on kuin pieni ohjelmistoprojekti. Näihin asioihin huomiota kiinnittämällä pyritään siis poistamaan perinteisiä ohjelmistotuotannossa havaittuja ongelmia (Gothelf & Seidan 2013). Yleisesti käytössä olevia ketteriä kehitysmenetelmiä ovat mm. Scrum, XP ja Lean. Seuraavassa esitellään lyhyesti tutkimuksen kannalta olennaiset ketterien viitekehysten pääperiaatteet. Viitekehysten erilaisia käytäntöjä tarkennetaan luvuissa 5 ja 6 tutkimustulosten yhteydessä.

Scrum

Schwaber & Sutherland (2013) määrittelevät Scrumin *”monimutkaisten tuotteiden kehittämiseen ja ylläpitoon tarkoitettuksi viitekehykseksi.”* Scrum on ketteristä menetelmistä kirjallisuuden perusteella (Lehtonen et al. 2014) käytetyin. Scrum keskittyy tuotekehitykseen muutaman viikon mittaisissa pyrähdyksissä (engl. sprint). Olennaista Scrumille on iteratiivisuus ja inkrementaalisuus. Malli sisältää myös erilaisia työntekijöille osoitettuja rooleja sekä muodollisia tapahtumia. Rooleja ovat tuoteomistaja (engl. Product Owner), Scrum mestari (engl. Scrum Master) ja tuotekehitystiimin jäsen. Projektin aikana pidettäviä muodollisia tapahtumia ovat esimerkiksi päiväpalaverit (engl. Daily Scrum), katselmoinnit (engl. Sprint Review) ja edellisen pyrähdyn läpikäynnit (engl. Sprint Retrospective). Roolien avulla tuotantoprosessin vastuunjako pyritään selkiyt-

tämään, ja muodollisten tapahtumien tarkoituksena on jatkuvan kehityksen varmistaminen ja muutosten hallinta.

Lean

Womackin & Jonesin mukaan (2003) Lean on johtamisfilosofia, joka keskittyy seitsemän erilaisen "turhuuden" (engl. waste) poistamiseen. Tällä pyritään parantamaan niin asiakastytyvääisyyttä kuin laatua. Lisäksi tavoitteena on toiminnan kustannuksien vähentäminen ja tuotantoprosessin läpimenoaikojen lyhentäminen. Lean luettelee arvoa tuottamattomiksi eli turhiksi asioiksi kuljetukset, varastot, liikkeen, odotusajan, ylituotannon, yliprosessoinnin ja vialliset tuotteet. Näiden ongelmien poistamiseen Lean tarjoaa erilaisia työkaluja kuten jatkuvaa kehittämistä, imuohjausta (Kanban), ja virheiden prosessista poistamista.

Kanban

Kanban pohjautuu Lean-periaatteeseen (Kniberg & Skarin 2010). Knibergin ja Skarinin mukaan Kanban voidaan tiivistää kolmeen pääsääntöön. Näistä ensimmäinen on työnkulun visualisointi, esimerkiksi Kanban-taulun avulla. Työ jaetaan korteille ja kortit laitetaan tauluun havainnollistamaan työnkulkua. Toisena pääsääntönä on käynnissä olevien töiden määrän rajoittaminen vaihetta kohti. Kolmas pääsääntö on työn osien läpimenoajan mittaaminen. Läpimenoajalla tarkoitetaan aikaa, joka yhdellä työnosalla kestää kulkea työprosessin alusta loppuun (ibid.). Kanban on siis joukko periaatteita, joiden avulla Leanin mukaista ajattelua voidaan tuottaa organisaatiossa. Kanbanin käyttäjän oletetaan myös kokeilevan prosessia eri tavoin mukauttaen sitä oman yrityksen prosessiin sopivaksi. Lean ajattelussa tätä jatkuvaa kehitystä kutsutaan nimellä Kaizen (ibid.).

XP (engl. Extreme Programming)

XP perustuu viiteen arvoon ja niiden pohjalta luotuihin käytäntöihin (Wells 2009). Arvoja ovat kommunikaatio, yksinkertaisuus, palaute, rohkeus ja kunnioitus. Menetelmä pyrkii ohjelmiston laadun parantamiseen ja painottaa asiakastytyvääisyyttä Ohjelmistojulkaisuja pyritään tekemään lyhyissä sykleissä ja usein. Näin tiimillä on mahdollisuus vastata vaatimusten muutokseen mahdollisimman nopeasti iteraatioiden välillä. XP:n käytäntöjä ovat mm. Pariohjelmointi, tasainen tahti, jatkuva integrointi, pienet julkaisut ja testivetoinen kehitys.

2.6 Ketteryys sulautetussa ohjelmistotuotannossa

Siinä missä ohjelmistomaaailma on siis jo pitkään suunnannut kehityksen kohti ketteryyttä (Gothelf & Seidan 2013), sulautettujen järjestelmien kehityksessä ketteryyden suuntaan on liikuttu hitaammin (Lehtonen et al. 2014). Vaikka asiantuntijat ovat esitelleet monia erilaisia ketteriä menetelmiä, yhtäkään niistä ei ole kohdennettu sulautetun

ohjelmiston tuotantoon (Ronkainen & Abrahamsson 2003). Suuri osa ketteristä menetelmistä on kehitetty nimenomaan ohjelmistotuotannon prosessinhallintaan. Parhaiten ketterät menetelmät toimivatkin ympäristössä, jossa kehitystyötä voidaan tehdä yhdessä paikassa, vähäisellä arkkitehtuurisuunnittelulla ja olio-pohjaisesti. Myös asiakas on mukana prosessissa kiinteästi. Kun taas mietitään normaalia laitteistokehitysympäristöä, on se hyvin erilainen. Sulautettuja ohjelmistoja kehitettäessä koodin määrä ei toimi mittarina, vaan tärkeintä on tuotteen suorituskyky, luotettavuus ja jatkuvasti muuttuvien laitteistovaatimusten huomioiminen.

Laitteiston kehittämisestä ketterien menetelmien mukaan on olemassa hyvin vähän tutkimustuloksia, mutta sulautetun järjestelmän ohjelmistokehitystä voidaan tehdä ketterästi omassa prosessissaan. Tämä tarkoittaa siis sitä, että luvussa 2.3 (kuva 3) mukainen prosessimalli on monissa yrityksissä edelleen käytössä, mutta laitteisto tehdään vesiputous mallin mukaisesti ja ohjelmisto sen rinnalla iteratiivisesti.

Koska ketterät menetelmät on ensisijaisesti suunniteltu ohjelmistojen kehitykseen, ei niiden käyttäminen sellaisenaan sovi sulautettujen järjestelmien kehitykseen (Salo & Abrahamsson 2008). Joitakin menetelmien piirteitä voidaan kuitenkin soveltaen käyttää hyväksi (ibid.). Useissa lähteissä todetaan (Lehtonen et al. 2014; Kasila 2013), että ketteristä menetelmistä voi olla apua sulautettujen järjestelmien kehitysprosessissa. Kasila (2013) päätyi työssään päätelmään, että ketterien menetelmien käyttö sulautetun järjestelmän suunnittelussa ja toteutuksessa saattaa parantaa projektin mahdollisuutta onnistua täyttämään asiakkaan vaatimukset. Toimintamallien muuttaminen ketteriksi vaatii kuitenkin yrityksiltä paljon panostusta, mutta on palkitsevaa, kun prosessi saadaan pyörimään (Lehtonen et al. 2014). Lisäksi Abrahamssonin (2005) mukaan ketterien menetelmien soveltaminen voi lyhentää läpimenoaikaa 70% monilla teollisuuden aloilla. Kustannustehokkuus onkin tärkein tekijä kehitettäessä sulautettuja järjestelmiä (ibid.). Sulautetut järjestelmät ovat usein tuotteita massamarkkinoille, ja valmistuksen kustannukset muodostavat ison osan projektien budjetista.

Green (2004) kertoo julkaisussaan ketterien menetelmien soveltamisesta sulautetun järjestelmän tuotantoprosessiin. Tutkimuksen kohteena olleessa yrityksessä tunnistettiin samoja ongelmia kuin monissa muissa sulautetun järjestelmän projekteissa: mm. suuret suunnitelmat, joiden toteutuminen loppui ensimmäisen viikon jälkeen, liian erikoistuneet työntekijäroolit ja huono koodin ylläpidettävyys. Ongelmat tuntuivat juontuvan nimenomaan tehokkaan ohjelmistotuotannon menetelmän käytöstä, mikä on yleistä sulautettuja järjestelmiä tuottaessa. Green (ibid.) toteaa myös, että suurimman osan sulautetusta ohjelmistosta tekevät laitteistoinsinöörit, joiden ohjelmistupuolen osaaminen ei välttämättä ole parhaalla mahdollisella tasolla. Green valitsi tiimilleen uudeksi metodiksi Scrumin ja XP:n yhdistelmän. Green otti käyttöön esimerkiksi testitapausten suunnit-

telun, jolla sittemmin tuettiin kehitystyötä. Lisätehoa tiimityöskentelyyn saatiin XP:ssä esitellystä parikoodauksesta. Parikoodauksessa ohjelmisto- ja laitteistokehittäjät työskentelevät pareittain saman työaseman ääressä (Wells 2009). Yhdessä työskennellessä virheiden huomaaminen ja koodin luettavuus paranee.

2.6.1 Ketteryyden haasteet ja mahdollisuudet

Sulautettujen järjestelmien ketterä käsikirja (2014) mainitsee ketteryyden haasteiksi ja mahdollisuuksiksi sulautettujen järjestelmien kehityksessä kuusi kohtaa:

- *kokonais kuvan ymmärrys,*
- *projektin etenemisen näkyvyys,*
- *monialainen kehitysympäristö,*
- *suunnitelmallisuuden tarve,*
- *tavoittamattomat asiakkaat ja*
- *vaihtelua sietämättömät ominaisuudet.*

Kokonais kuvan ymmärryksen haasteella käsikirjan mukaan (Lehtonen et al. 2014) tarkoitetaan sitä, että pieneen erikoisosaamisalueeseen perehtyneet asiantuntijat saattavat unohtaa projektin kokonais kuvan muodostamisen ja sen ylläpitämisen projektin edetessä. Sulautetuille järjestelmille on tyypillistä, että monien erikoisosa-alueiden tiedot yhdistyvät prosessissa yhdeksi kokonaisuudeksi. Tällöin ymmärrys siitä, miten oma työ vaikuttaa ja linkittyy toisten työhön saattaa on erityisen tärkeää. Ketterät menetelmät ohjaavat yhteisvastuuseen projekteissa ja yhdessä tiimissä on mukana monen osa-alueen osaajia, jolloin näkyvyys läpi projektin paranee. Kokonaisuus pilkotaan osiin pikemminkin toiminnallisuuden, kuin työntekijöiden teknisen osaamisen perusteella (ibid.).

Edelliseen kohtaan tiukasti liitoksissa oleva *projektin etenemisen näkyvyys* on yksi sulautetun ohjelmistotuotannon haasteista (Lehtonen et al. 2014). Kehittäjältä ei perinteisessä toimintaympäristössä vaadita kovin tarkkaa tietoa projektin etenemisestä. Projektinhallintaan liittyvät toimet saatetaan kokea taloudelliseen näkökulmaan liittyviksi asioiksi, jotka eivät juurikaan kuulu työntekijöille. Ketterät menetelmät panostavat nimenomaan siihen, että kaikilla työntekijöillä olisi näkyvyys projektin kokonaistilaan. Tähän satsataan mm. päivittäisillä tilannepalavereilla, mikä helpottaa myös tarttumista ongelma-kohtiin mahdollisimman nopeasti. "*Hyvä näkyvyys projektin tekniseen etenemiseen helpottaa oman työn organisointia ja sujuvuutta sekä motivoi etenemään kohti yhteistä tavoitetta*" (Lehtonen et al. 2014, s. 13).

Monialaisella kehitysympäristöllä tarkoitetaan sitä, että kehitystiimi koostuu hyvin erikoistuneista, osaamiseltaan hyvin rajatuista työntekijöistä. Ketterät menetelmät taas kannustavat osaamisen jatkuvaan laajentamiseen niin, että kehitystiimi voisi sisältää kaiken projektissa tarvittavan osaamisen (Schwaber & Sutherland 2013). Tällainen tehtävien täydelliseen jakamiseen perustuva työnjako ei sulautettujen järjestelmien piirissä

ole tarkoituksenmukaista, mutta perehtyminen muiden tiiminjäsentien alueisiin saattaa helpottaa kokonaiskuvan ja riippuvuussuhteiden ymmärtämistä (Lehtonen et al. 2014). Laaja perehtyneisyys omaan osa-alueeseen ei myöskään estä toimimista osana tiimiä.

Suunnitelmallisuuden tarpeen haasteella tarkoitetaan sitä, että toisin kuin laitteistosuunnittelussa on totuttu, ketterässä kehityksessä suunnitelmat tarkentuvat projektin edetessä. Jotta laitteisto saataisiin testauskuntoon, saattaa se vaatia muutaman viikon kehitysjakson. Tämä perustavanlaatuinen ristiriita on hyvä huomioida ketteriä menetelmiä lisättäessä. Projekteissa on osa-alueita, joihin ketterät menetelmät eivät sovi. Lehtonen et al (2014) lisäävätkin, että projektille kannattaa suunnitella etapit, jotka ovat laitteistokehityksessä tärkeitä, ja rakentaa ketteryyttä näiden etappien välille.

Tavoittamattomien asiakkaiden haasteella tarkoitetaan sitä, että sulautettujen järjestelmien käyttäjiä ei prosessissa välttämättä tunneta. Lehtonen et al. (2014) väittävät, että koska sulautetut järjestelmät ovat usein tuotteita, jotka myydään massamarkkinoille, tai suunnitteluvaiheessa vielä tuntemattomille loppukäyttäjille, tiivistä asiakasyhteistyötä on usein hankala ylläpitää. Käyttäjäkokemuksen piirissä väitteen todenperäisyydestä ollaan hyvin eri mieltä. Juuri tämänkaltaisissa prosesseissa, joissa menetykset saattavat projektin epäonnistuessa olla suuria, loppukäyttäjän tarpeiden tunteminen olisi erityisen tärkeää. Usein loppukäyttäjän edustajana saattaa toimia yrityksen markkinointihenkilö, jonka tieto tuotteista ei välttämättä pohjautu todellisen käyttäjän tarpeisiin. Ketterät menetelmät suosittelevatkin tiivistä asiakasyhteistyötä menetysten välttämiseksi (Lehtonen et al 2014). Olisikin tärkeää, että sulautetun järjestelmän tuotantoprosessikuvauksessa olisi linkitys ja jatkuvaa kommunikointia niin kehittäjien, käyttäjäkokemushenkilöiden kuin myynti- ja markkinointiosaston välillä.

Vaihtelua sietämättömillä ominaisuuksilla tarkoitetaan sitä, että laitteiston ja ohjelmiston on sulautetussa järjestelmässä toimittava saumattomasti yhteen. Sana "sulautettu järjestelmä" itsessään kertoo kuinka tärkeää on rakentaa laitetta niin, että näiden kahden osa-alueen yhteentoimivuus varmistetaan. Tämä aiheuttaa haasteita tuotekehitystyöhön. Molemmilta osapuolilta vaaditaan ennakoivuutta suhteessa toiseen. Kehittäjien välinen kommunikointi saa tässä erityisen suuren roolin. Näin mahdolliset muutokset saadaan kaikkien tietoon mahdollisimman pian ja turhan työn tekeminen on mahdollista minimoida (ibid.).

3. KÄYTTÄJÄKOKEMUSTYÖ OHJELMISTOPROJEKTISSA

Tässä luvussa tarkastellaan käyttäjäkokemuksen käsitettä ja käyttäjäkokemustyön tekemistä ohjelmistotuotannon prosessissa. Luvussa esitellään myös ketterän kehityksen mahdollisuudet käyttäjäkeskeisessä tuotekehitysprosessissa erilaisten mallien avulla.

3.1 Mitä on käyttäjäkokemus?

Yksi ensimmäisistä käyttäjäkokemuksen määritelmistä on vuodelta 1996. Alben (1996, s.4) määrittelee käyttäjäkokemuksen seuraavasti:

”Käyttäjäkokemus kattaa kaikki näkökulmat siitä, kuinka ihminen käyttää vuorovaikutteista tuotetta – miltä se tuntuu kädessä, kuinka hyvin sen toimintaa ymmärretään, miltä käyttäjästä käyttöhetkellä tuntuu, kuinka hyvin tuote palvelee tarkoitustaan, kuinka se soveltuu käyttökontekstiin ja kuinka hyvin se sopii käyttäjän elämänlaatuun.”

Albenin määritelmä ei sisällä tuotteen käyttöä edeltävää aikaa, eli kokemusta tuotteesta ennen varsinaista käyttötilannetta. ISO-standardin mukaan (ISO 9241-210 2010, s.3.) käyttäjäkokemuksella (engl. User Experience, UX) tarkoitetaan: *”Yksilön havaintoja ja reaktioita, jotka ovat tulosta tuotteen, systeemin tai palvelun käytöstä ja/tai ennakoitusta käytöstä”*. Käyttäjäkokemus ei siis koostu ainoastaan tuotteen käytön jälkeisestä kokemuksesta, vaan se voi liittyä tuotteeseen monella eri tavalla. Esimerkiksi tuotemerkki voi vaikuttaa tuotteen käyttäjäkokemukseen, samoin kuin käyttäjän aiemmat oletukset, taidot, ja persoonallisuus.

Yksi viimeisimmistä käyttäjäkokemusmääritelmistä lisää määritelmään mielihyvän käsitteen. Sutcliffen (2010) mukaan käyttäjäkokemus on: *”Käyttäjän arvio tuotteen laadusta, joka perustuu vuorovaikutuskokemukseen, käytön tehokkuuteen sekä tuotteen synnyttämään mielihyvän tunteeseen.”* Jotta tuotteen käyttäjäkokemus olisi hyvä, sen käytöstä tulee seurata mielihyvää. Käyttäjäkokemuksen määritelmä on siis edennyt funktionaaliseen käytettävyyden- ja tehokkuusajattelusta hedonistiseen ihmisen tarpeita täyttävään ajatteluun.

Käyttäjäkokemuksen kohteena ovat käyttäjät. Eason (1987) jakaa tuotteen käyttäjät kolmeen ryhmään: ensisijaiset, toissijaiset ja muut. Ensisijaiset käyttäjät ovat niitä, jotka todella käyttävät tuotetta. Toissijaiset käyttäjät ovat niitä, jotka käyttävät tuotetta silloin tällöin tai ne jotka käyttävät sitä jonkun muun kautta. Kolmas käyttäjäryhmä sisältää henkilöt, jotka joutuvat välillisesti olemaan tekemisissä tuotteen kanssa. Näitä käyttäjä-

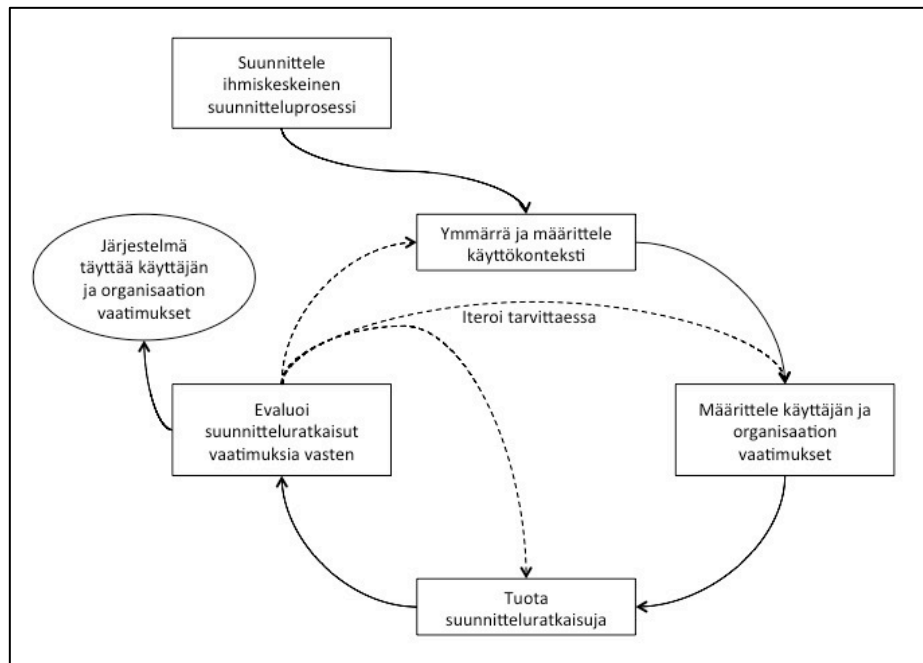
ryhmiä tarkastellessa huomaa, että käyttäjäkokemus voi ylettyä kauas perinteisestä käyttäjäkäsitteestä, jossa käyttäjäkokemus syntyy tuotteen käytön aikana. Bainbri (Ab-ras & Al, 2004) ehdottaakin, että onnistunut käyttäjäkokemussuunnittelu ottaa huomioon kaikki nämä ryhmät, eli kaikki mahdolliset tuotteen käyttäjäkokemuksen piirissä olevat käyttäjät.

3.2 Käyttäjakeskeinen suunnitteluprosessi

Käyttäjäkokemustyötä tehdään erilaisissa käyttäjakeskeisissä suunnitteluprosesseissa. Näille prosesseille, jossa tuotetta suunnitellaan pitäen silmällä nimenomaan lopputuloksen sopivuutta käyttäjälle on monia nimityksiä. Eri näkemyksissä korostuvat hieman eri näkökulmat.

Nykyään käyttäjakeskeisyys termin sijaan käytetään usein ihmiskeskeisyys-termiä. Tällä halutaan korostaa sitä, että tuote vaikuttaa myös muualle kuin käyttäjiinsä, eli sidosryhmiinsä. Ihmiskeskeisellä suunnittelulla tarkoitetaan ISO 9241-210 -standardin (2010, s. 2) mukaan ”*näkökulmaa systeemien suunnitteluun ja kehitykseen, jossa pyritään tekemään interaktiivisista systeemeistä paremmin käytettäviä keskittymällä systeemin käyttöön ja soveltamalla inhimillisiä tekijöitä ja käytettävyyden tietoutta ja tekniikoita.*”

ISO-standardi (ISO 9241-210, s.10) esittelee ihmiskeskeisen suunnitteluprosessin (engl. Human-centered design = HCD). Prosessin (kuva 4) tarkoituksena on oheistaa suunnittelijoita ottamaan työssään ihminen käyttäjänä ja käyttäjäkokemus huomioon kaikissa projektin vaiheissa.



Kuva 4. Ihmiskeskeinen suunnitteluprosessi (ISO-9241-210 mukaan).

Ihmiskeskeisen suunnittelun malli (ISO-9241-210 2010) perustuu neljän askeleeseen, joista ensimmäinen on tuotteen käyttökontekstin määrittely ja ymmärtäminen. Käyttökontekstit voivat vaihdella käyttäjäryhmien ja käyttötilanteiden välillä. Tämä pitää ottaa huomioon suunnittelussa. Toisena askeleena on käyttäjävaatimusten määrittely. Käyttäjävaatimukset voivat vaihdella käyttäjästä toiseen ja olla jopa toisilleen vastakkaisia (ISO-9241-210 2010).

Kun konteksti ja käyttäjävaatimukset on saatu määriteltyä, voidaan aloittaa suunnitteluratkaisujen tuottaminen perustuen määrittelyihin. Ensimmäiset ratkaisut ovat kuitenkin harvoin käyttäjiään tyydyttäviä, ja siksi suunnitteluratkaisujen arviointi on seuraava kohta suunnittelussa. Arvioinnin jälkeen voidaan taas palata edellisiin kohtiin ja iteroida suunnitteluratkaisuja kunnes päästään toivottuun lopputulokseen (ISO-9241-210 2010).

Tuotteen käyttäjiin pyritään siis saamaan kontakti mahdollisimman aikaisessa vaiheessa tuotekehitysprosessia, ja palautetta kerätään erilaisin käyttäjäkeskeisin menetelmin. Malli on iteratiivinen, eli tuotteen määrittelyä tarkennetaan jatkuvasti palautteen perusteella läpi prosessin.

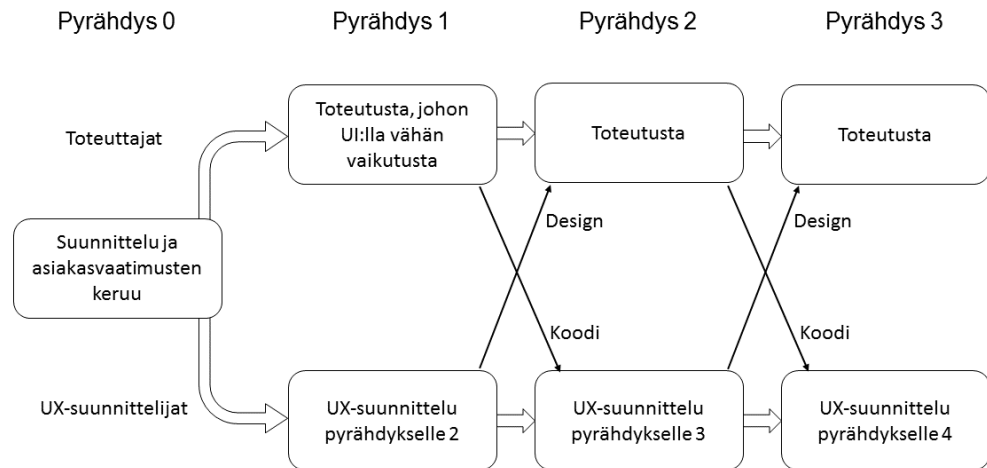
3.3 Ketteryys käyttäjäkeskeisessä ohjelmistoprojektissa

Kaikkialla ohjelmistotuotannon kentällä yleistynyt ketteryys (Gothelf & Seidan 2013, s. 6) on vaikuttanut myös käyttäjäkeskeisen suunnittelun kenttään. Ketterän ohjelmistokehityksen julistus (2001) ei ota selkeää kantaa siihen, missä vaiheessa ja miten käyttäjäkokemustyö pitäisi tuotekehitysprosessissa suorittaa.

Scrum on ehkä yksi käytetyimmistä ketteristä menetelmistä (West, 2011). Vaikkei Scrum käsikirja (Schwaber & Sutherland 2001) otakaan suoraa kantaa käyttäjäkokemustyön toteuttamiseen, löytyy Scrum-aiheisista julkaisuista kuitenkin epäsuoria viitteitä siihen, missä vaiheessa prosessia ja miten käyttäjäkokemustyötä pitäisi tehdä. Scrum käsikirjassa (ibid.) sanotaan, että "*ainoastaan kehitystiimin jäsenet osallistuvat tuoteversion kehitykseen.*" Tämä voidaan ymmärtää niin, että myös käyttäjäkokemustyön tulisi löytyä tuotekehitystiimin sisältä (Highsmith 2001). Koska Scrum ei kuitenkaan ole luotu nimenomaan käyttäjäkeskeiseen suunnitteluprosessin läpiviemiseen, esitetään seuraavissa luvuissa muutamia käyttäjäkeskeisiä ketteriä prosessimalleja. Mallit pyrkivät hyvin samankaltaiseen lopputulokseen iteratiivisin keinoin, mutta malleissa korostetaan erilaisia toimintatapoja.

3.3.1 "Yksi pyrähdys edellä" -malli

Sy (2007, s. 112–132) ehdottaa vuonna 2007 kirjoittamassa jornaalijulkaisussaan malliksi "yksi pyrähdys edellä" -ratkaisua (*engl. one sprint ahead*), jossa osa suunnittelusta tehdään erillisenä vaiheena ennen varsinaisen kehitysprosessin alkua (Kuva 5).

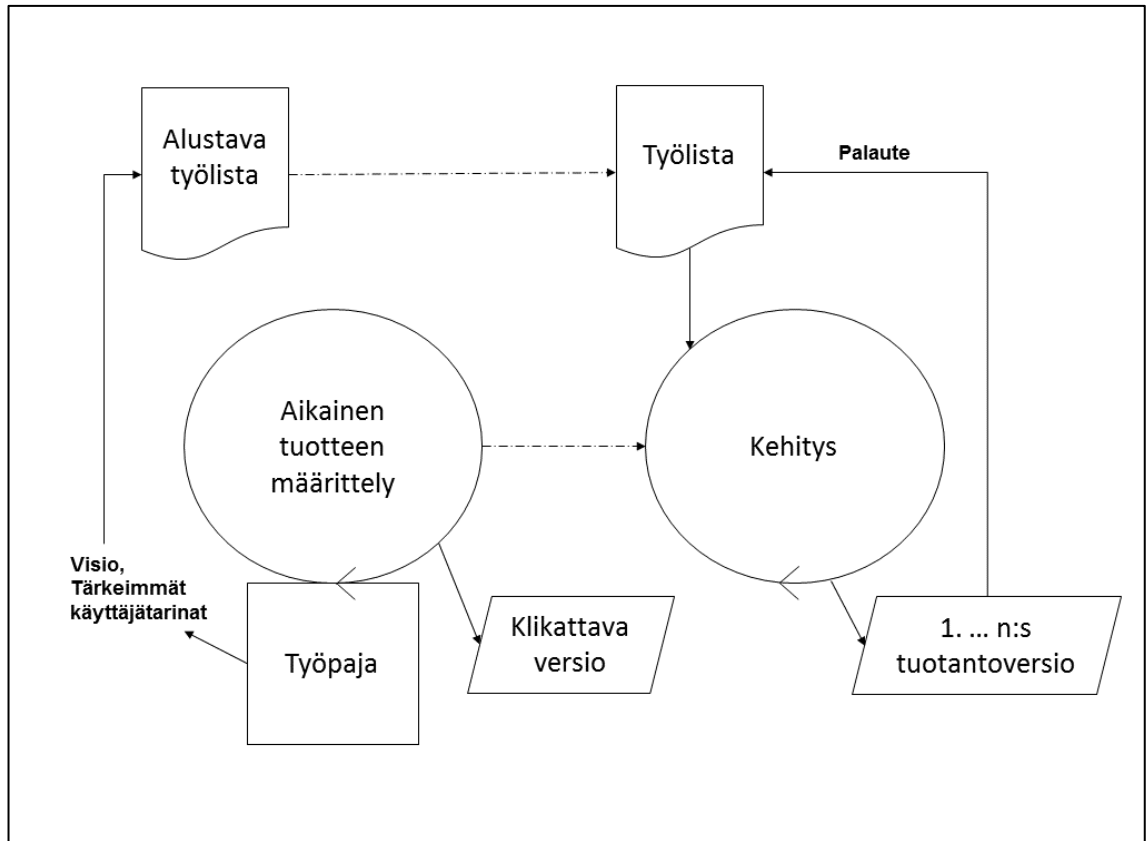


Kuva 5. Yksi pyrähdys edellä -malli (Sy 2007).

Siirtyminen tällaiseen iteraatioihin perustuvaan malliin on iso askel, mikäli tuoteprosessi on aiemmin ollut vesiputousmallinen ja suunnittelutyö on vaadittu melko valmiina pakettina ennen varsinaisen kehitystyön alkua. Näkökulma ei kuitenkaan ole ongelmaton (Kuusinen 2015). Kehittäjien tulee aloittaa suunnittelutyönsä yksi tai kaksi iteraatiota ennen varsinaisen kehitystyön alkua. Lisäksi käyttäjäkokemussuunnittelijoiden tulee testata tuotokset edelliseltä iteraatiolta, kun kehittäjät jo työstävä seuraavaa koodilisäystä. Myös Ferreira et al. (2007) ovat tutkineet mallin toimivuutta todeten, että ketterissä menetelmissä suosittu moniosaamisen korostaminen tiimiä suunnitella on tehokkaampi tapa toimia, kuin että käyttäjäkokemussosaamiseen perehtynyt käyttäjäkokemustiimi tekee töitä tiiminä "yksi pyrähdys edellä".

3.3.2 BoB (Best of Both Worlds) -viitekehys

Kuusinen (2016) esittelee julkaisussaan BoB-viitekehysten (kuva 6). BoB tulee sanoista Best of Both Worlds, mikä tarkoittaa parhaiden käytänteiden yhdistämistä käyttäjäkokemustyön ja ketterien menetelmien maailmoista (Kuusinen 2016). BoB-malli perustuu Kuusisen pitkäaikaiseen yritysten ketteriä käyttäjäkokemuskäytäntöjä arvioineeseen tutkimukseen.



Kuva 6. Best of Both Worlds (BoB) -malli (Kuusinen 2016).

Kuusinen (2016) suosittaa prosessin aloittamista muutamilla pienillä työpajoilla, joihin osallistuvat tiimin käyttäjäkokemussuunnittelija, kehittäjä ja käyttäjä. Projektin visio on usein alussa epäselvä, ja näin voidaan varmistaa tuotteen rakentaminen käyttäjän tarpeiden pohjalta ja että kaikilla on tuotteesta yhteinen visio. Samalla luodaan tärkeimmät käyttäjätarinat, joiden avulla voidaan aloittaa tuotteen rakentaminen.

Kuusinen (2015) korostaa prosessin alkuvaiheen suunnittelukeskeisyyttä, mutta jättää tarkasti määrittelemättä ensimmäisten iteraatioiden toimet. Iteratiivisuutta ja inkrementtaalisuutta korostetaan heti ensimmäisestä pyrähdyksestä. Tämän lisäksi pitkän analyysivaiheen olemassaoloa suositellaan välttämään.

Käyttäjäkokemuksen rakentamista viedään eteenpäin jatkuvalla vuorovaikutuksella tiimissä. Käyttäjäkokemussuunnittelija toimii siis tiimin sisällä ja myös käyttäjät otetaan mukaan iteraation aikaiseen työskentelyyn silloin kun siihen on tarve. Kuusinen (2016) korostaa, että käyttäjäkokemuksen suunnittelun pitäisi olla näkyvää / elinkelpoista koko prosessin ajan. Tämä tarkoittaa sitä, että tehdään jatkuvasti töitä sen eteen, että ominaisuuksia validoidaan ja niistä tulee osa toimivaa ohjelmistoa. Kuusinen (2016) ehdottaa myös tiukoista roolijaoista luopumista. Pyrähdykseen suunnitellut tehtävät ovat valittavissa kaikille tiimin jäsenille, joista kukin valitsee kompetensseihinsä sopivat. Tämä

tarkoittaa sitä, että joskus kehittäjä voi valita käyttäjäkokemussuunnittelijalle tyypillisesti allokoitun tehtävän tai toisinpäin.

Testaus tehdään BoB-mallin mukaan pyrähdysten aikana. Tällä pyritään siihen, että työntekijällä ei ole samanaikaisesti useita iteraatioita käynnissä ja että pyrähdykseen suunnitellut tehtävät tehdään loppuun asti. Tuotteen ensimmäinen versio pyritään myös saamaan nopeasti ulos, jotta todellista käyttäjäpalautetta aletaan saada mahdollisimman pian ja tuotetta voidaan rakentaa oikeaan suuntaan. (Kuusinen 2016)

Eri vaiheiden lisäksi Kuusinen (2016) korostaa, miten tärkeää on että kehittäjät ymmärtävät syyt suunnitteluratkaisuihin. Mikäli todellista ymmärrystä ei ole, saattaa se johtaa tuplatyöhön ja kokonaiskuvan hämärtymiseen. Jos käyttäjäkokemustyö tehdään tiimin sisällä ja tiimin sisäisiä rooleja pyritään aktiivisesti hämärtämään, tiimin sisällä tapahtuu oppimista, ja kehittäjät voivat parhaassa tapauksessa omaksua itse käyttäjäkokemustyön tekemisen. On kuitenkin tärkeää, että joko käyttäjäkokemussuunnittelijalla tai tuotemistajalla on selkeä kokonaiskuva tuotteesta hallussaan. Lisäksi olisi tärkeää, että kehittäjät ovat aktiivisesti yhteydessä käyttäjiin, ja näkevät työnsä tuloksen asiakkaiden käytössä. Tämä auttaa kehittäjiä kokonaiskuvan jatkuvassa tarkentamisessa.

3.3.3 Lean UX

Lean UX on kehitetty Lean filosofian ja Agile liikkeen pohjalle (Gothelf & Seiden 2013). Lean UX rakentuu kolmelle tukipilarille: design-ajattelulle, ketterälle ohjelmistotuotannolle ja Lean Startup -metodille. Design-ajattelulla tarkoitetaan innovointia joka saa voimansa sen tarkkailemisella mitä ihmiset haluavat ja tarvitsevat elämässään ja toisaalta mistä he eivät pidä olemassa olevissa tuotteissa (ibid.). Lean UX on oppisuuntaus, joka käyttää suunnittelijan herkkyyttä ja tutkimusmetodeja arvioimaan ihmisten tarpeita ja etsimään niistä liiketoimintamahdollisuuksia saatavilla olevan teknologian ja taloudellisen kannattavuuden rajoissa. Design-näkökulmassa otetaan näkökulma, jossa jokaista liikeajattelu näkökulmaa voidaan lähestyä design-metodilla. Tämä tarkoittaa sitä, että suunnittelijoilla on lupa työskennellä myös normaaleiden rajojensa ulkopuolella ja toisaalta rohkaista muita käyttämään design-metodeja ongelmanratkaisuun. Design-ajattelu siis rohkaisee tiimejä tekemään yhteistyötä ohi roolirajojen ja ajattelemaan tuotesuunnittelua kokonaisvaltaisesti.

Toinen Lean UX:n tukipilareista on ketterä kehitys (ibid.). Ketterästä kehityksestä on kerrottu enemmän tämän työn luvussa 2.5 (s. 8.).

Kolmas tukipilari, Lean Startup -Metodi (Klein 2013). Tässä metodissa tehdään iteraatioita, joita kutsutaan "rakenna-mittaa-opi" -iteraatioiksi. Tällä tavoitellaan projektien riskin minimoimista ja pyrkimyksenä on saada tiimi sekä rakentamaan että oppimaan nopeasti. Tiimit rakentavat MVP:itä (engl. Minimum Viable Product) ja toimittavat ne nopeasti eteenpäin, jotta palautteesta oppimisprosessi saadaan mahdollisimman nopeasti

käyntiin. MVP:llä tarkoitetaan pienintä mahdollista toteutusta tai prototyyppiä tuotteesta, jolla voidaan testata tuoteidea oikealla asiakkaalla (ibid.). Tällä pyritään vähentämään "jätettä", joka syntyy rakentamalla tuotteeseen ominaisuuksia, joita loppukäyttäjä ei halua.

Tämän ideologian lisäksi Gothelf & Seidan (2013) tarkentavat tärkeimmiksi noudatettaviksi periaatteikseen seuraavat kohdat:

- Moniosaajatiimit
- Pienet omistautuneet yhdessä paikassa sijaitsevat ratkaisukeskeiset tiimit
- Edistys = lopputulos
- ”Jätteen” (engl. Waste) poisto
- Pienet tuotantoerät
- Jatkuva löytöretkeily
- GOOP (engl. ”Get Out Of the Building”)
- Jaettu yhteinen ymmärrys asioista
- Ei guruja ja velhoja, vaan tiimin voimaa
- Työn ulkoistaminen
- Tuota mieluummin kuin analysoi
- Oppiminen kasvun sijaan
- Lupa epäonnistua

4. TUTKIMUSMENETELMÄT

Tässä luvussa käydään läpi tutkimuksen tavoitteet ja käytetyt tutkimusmenetelmät. Lisäksi esitellään tutkimuksen arviointiin käytetyt mittarit ja pohditaan tutkimusetiikkaa.

4.1 Tutkimuksen tavoitteet

Tutkimuksen tavoitteena on selvittää millaisia haasteita sulautettuja järjestelmiä toimittavat yritykset kohtaavat tuotantoprosesseissaan. Tämän lisäksi kartoitetaan käyttäjäkokemustyöhön liittyviä ongelmakohtia sulautettujen järjestelmien ohjelmistotuotantoprosessissa. Tuotantoprosessin tehostamiseen etsitään kehitysehdotuksia ketteristä menetelmistä kertovan tieteellisen kirjallisuuden avulla.

Tutkimuksen tulokset on esitetty yrityskohtaisesti, koska molemmilla tutkimuksessa mukana olevilla yrityksillä oli tutkimukselle hieman erilaiset tavoitteet. Tutkimuksessa mukana olevista yrityksistä ensimmäisessä tutkimuksen tulosten toivottiin selkeyttävän nykyistä prosessimallia. Toisessa yrityksessä tavoitteena oli selvittää miten tuotantoprosessin läpimenoaikaa saataisiin lyhennettyä. Näistä näkökulmista yrityksille tuotetaan erilliset raportit. Tutkimus toteutetaan vuosina 2014–2016.

4.2 Tutkimusmenetelmän valinta

Tutkimusmenetelmällä tarkoitetaan tutkimuksen aineiston hankinta- ja analyysimetodeja, jotka voidaan luokitella kvalitatiivisiin (laadulliset) ja kvantitatiivisiin (määrälliset) menetelmiin (Tilastokeskus 2016). Tutkimusmenetelmä tulee valita tutkimusongelman perusteella. Menetelmää valitessa tulee pohtia, millaisilla aineistoilla tai tekniikoilla tiedon saa parhaiten kerättyä ja tulkittua. Seuraavissa kohdissa on esitettynä valitut tutkimusmenetelmät ja perusteet valinnoille.

4.2.1 Monitapaustutkimus

Tutkimusmenetelmäksi valittiin monitapaustutkimus. Mikäli tutkimuksessa on mukana useampi kuin yksi tapaustutkimus, silloin puhutaan monitapaustutkimuksesta (Baxter & Jack 2008). Baxterin & Jackin mukaan monitapaustutkimus valitaan menetelmäksi usein, kun pelkän tapaustutkimuksen sijaan halutaan rakentaa suurempaa ymmärrystä ilmiöstä. Monitapaustutkimuksessa jokaisen tutkimukseen kuuluvan tapauksen konteksti on erilainen ja tutkijalla on mahdollisuus analysoida jokaista tapausta erikseen ja toisaalta ristiin muiden tutkimusten kanssa (ibid.). Monitapaustutkimus on hyvä menetelmä, kun halutaan selvittää tapausten välillä olevia yhtäläisyyksiä ja eroja (ibid.). Tässä

tutkimuksessa haluttiin selvittää kahden eri yrityksen tuotekehitysprosessissa kohtaamia ongelmia. Koska tutkimuskontekstit oli valittu jokseenkin samanlaisiksi (sulautettujen järjestelmien kehitys), mielenkiinnon kohteeksi nousivat juuri löydettyjen ongelmien yhtäläisyydet ja erot, ja miten nämä vaikuttavat tuotantoprosessin sujuvuuteen.

Monitapaustutkimuksessa on siis mukana aina monta tapaustutkimusta. Hirsjärven & Hurmeen (2004, s. 125–126) mukaan tapaustutkimus valitaan tutkimusmenetelmäksi usein, kun tutkimuskohteena on yksittäinen tapaus, tilanne, tapahtuma tai joukko tapauksia, joiden tarkastelussa kiinnostuksen kohteena ovat usein prosessit. Yin (1994, Saaränen-Kauppinen & Puusniekka 2006 mukaan) esittää, että tapaustutkimusta on paras käyttää, kun tutkimuksella pyritään kuvailemaan tai selittämään asioita. Esimerkkinä tutkimuskysymyksistä voisivat olla: "Mitä tapahtui, miten ja miksi" (ibid.). Juuri näihin kysymyksiin etsittiin haastattelukysymyksillä vastauksia yrityksen toiminnasta.

Hyvä suunnittelu on tärkeää tapaustutkimukselle (Hirsjärvi & Hurme 2004). Taulukossa 1 on määritelty tämän tutkimuksen suunnittelun kannalta olennaisimmat asiat.

Taulukko 1. Tämän tutkimuksen suunnittelun tärkeimmät asiat.

Tutkimussuunnitelman aihealue	Tässä tutkimuksessa
1. Mikä on tutkimuksen päämäärä?	Kuvaileva, parantava
2. Tapaus – mitä tutkitaan?	Sulautettu ohjelmistotuotantoprosessi
3. Teoria – mikä on viitekehys?	Viitekehys on esitetty kirjallisuuden avulla luvuissa 2 ja 3.
4. Tutkimuskysymykset – mitä halutaan tietää?	Tutkimuskysymykset on kuvattu kohdassa 1.3
5. Tutkimusmenetelmät – miten data kerätään?	Suora. Haastattelu. Kuvattu kohdassa 4.2
6. Valintastrategia - mistä dataa kerätään?	Saatavuuden perusteella, mukavuusotanta

4.3 Tiedonkeruumenetelmä

Haastattelu valitaan usein tiedonkeruumenetelmäksi (Hirsjärvi & Hurme 2004, s. 35) kun kyseessä on tuntematon alue ja/tai halutaan korostaa, että ihminen on subjekti joka voi tuoda esiin itseä koskevia asioita mahdollisimman vapaasti. Tässä tutkimuksessa haluttiin lisätietoa työntekijöiden näkemistä epäkohdista yrityksen toiminnassa. Haastattelun etuna on myös se, että haluttaessa haastattelutilanteessa voi saada lisätietoa ja varmistaa ymmärryksen lisäkysymyksillä. Diplomityössä ei voi tehdä laajoja tutkimuksia, siksi on hyvä saada halutuista asioista tieto saman haastattelun aikana. Tämä on hyvä asia myös haastateltavalle.

Haastattelu on hyvä tapa hahmottaa aineistoa paremmin, etenkin jos haastattelun aihepiiri on vähemmän tunnettu (Hirsjärvi & Hurme 2004). Haastattelemalla myös tosielämän esimerkkien saaminen on helpompaa. Tutkimukseen osallistuvia henkilöitä on helppompi motivoida henkilökohtaiseen haastatteluun kuin esimerkiksi vastaamaan pitkään kyselyyn. Hyvänä puolena haastattelussa on myös se, että validiutta voidaan lisätä tarkkailemalla haastateltavaa. Nämä ovat Hirsjärven ja Hurmeen (ibid.) mukaan syitä joiden vuoksi tiedonkeruumenetelmäksi kannattaa valita haastattelu.

Tutkimuksen aineisto on kerätty haastattelemalla kahden eri yrityksen työntekijöitä. Jottei aineisto kuitenkaan kasvaisi kooltaan liian suureksi, valittiin haastattelumenetelmistä puolistrukturoitu haastattelu. Puolistrukturoidulla tarkoitetaan menetelmää, jossa haastattelukysymykset suunnitellaan etukäteen, mutta lisäkysymyksiä asetettiin tarpeen vaatiessa ja haastateltavan roolista riippuen. Kysymykset tutkimuksessa eivät olleet yksityiskohtaisia, koska haastateltavien työtehtävät olivat hyvin erilaisia.

Ennen haastattelua haastateltavalle kerrottiin tutkimuksen tavoitteista ja haastattelun kulusta. Haastateltavalle annettiin tiedoksi, että kerätyistä tiedoista tullaan poistamaan tunnistetiedot, ja että tunnistetietojen poistamisen jälkeenkin haastateltava olisi mahdollisesti tunnistettavissa tutkimusaineistosta roolinsa perusteella (Saaranen-Kauppinen & Puusniekka 2006). Koska kyse ei kuitenkaan ole arkaluontoisesta aineistosta ei tutkimusetiikan mukaan (Lee, 1993, s. 3–4) tarvitse juurikaan välittää, vaikka vastaajien henkilöllisyys saattaa paljastua vielä tunnistetietojen poistamisen jälkeenkin.

Haastatteluiden alussa haastateltava täytti tutkimussuostumuslomakkeen (LIITE A) ja taustatietolomakkeen (LIITE B). Kaikki haastateltavat suostuivat haastattelujen äänitallennukseen.

4.3.1 Haastatteluympäristö, aineiston tallennus ja haastateltavat

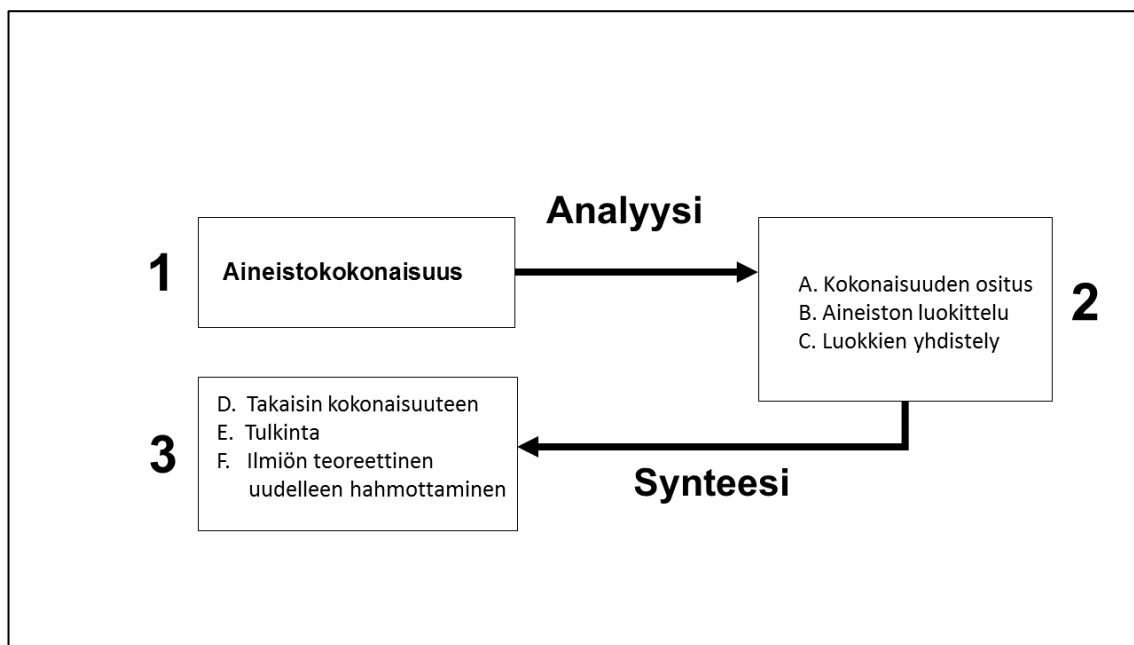
Haastateltavat valittiin tutkimukseen mukavuusotannalla eli kohdeyrityksen tutkimuksesta vastaava henkilö valitsi haastateltavat mahdollisimman kattavasti projektin ympäriltä (Hirsjärvi et al. 1997). Tutkijan toive yritykseen oli, että haastateltavissa henkilöissä olisi työntekijöitä mahdollisimman monesta projektin vaiheesta ja että haastateltavien joukossa olisi käyttäjäkokemuksesta vastaavia työntekijöitä. Haastateltavia tutkimuksessa oli yhteensä 11. Näistä 5 kuului ensimmäiseen tutkimukseen ja 6 toiseen tutkimukseen.

Molemmat haastattelukierrokset tehtiin yritysten omissa tiloissa. Haastattelussa oli mukana haastattelija ja haastateltava. Yksi haastattelu (yrityksessä 1) tehtiin parihaastatteluna, eli haastattelijan lisäksi paikalla oli kaksi haastateltavaa. Haastattelujen kesto vaihteli 40 ja 70 minuutin välillä. Haastattelut tallennettiin äänitallentimella ja haastattelija teki haastattelun aikana muistiinpanoja.

Lisätiedot haastateltavista on löydettävissä tutkimuksen tuloksista taulukosta 2 (s. 29) ja taulukosta 7 (s. 43).

4.4 Aineiston käsittely ja analyysimenetelmä

Haastattelujen jälkeen äänitallenteet litteroitiin, eli sanelunauhat kuunneltiin ja sisältö kirjoitettiin tekstiksi (Hirsjärvi & Hurme 2011, s. 138). Litteroidessa haastateltaville annettiin tunnistetiedot numeroin. Tämän jälkeen data yhdistettiin, josta saatiin kuvan 7 (kohta 1) mukainen analyysiketjun aloittava "aineistokokonaisuus". Tämän jälkeen aineisto purettiin teema-alueittain (kuva 7), eli kokonaisuudesta osiin. Purkaessa tehtiin oletus, ettei haastattelujen aikajärjestyksellä ole tämän tutkimuksen kannalta suurta merkitystä. Osiin jakamisen tarkoituksena oli siis löytää litteroidun haastattelumateriaalin tosiasiallinen sisältö ja jättää pois tutkimuksen kannalta turha aineisto (ibid.).



Kuva 7. Haastatteluaineiston käsittely analyysistä synteisiin (Hirsjärvi & Hurme 2001).

Aineiston purkamisen jälkeen se luettiin ja luokiteltiin (Kuva 7, kohta 2B). Hirsjärven & Hurmeen mukaan (2011, s. 147) luokittelulla tarkoitetaan "*tutkittavan ilmiön jäsentämistä, vertailemalla aineiston eri osia toisiinsa*". "Koska tutkimuskysymyksenä oli esimerkiksi haasteiden etsiminen, aineisto luokiteltiin melko mutkattomasti eri "haaste-luokkiin". Luokkia muodostui noin kaksinkertainen määrä verrattuna lopullisiin luokkayhdistelmiin.

Mikäli löytyneille kategorioille löytyi yhdistävä korkeamman tason otsikko, luokat yhdistettiin (Kuva 7, kohta 2C). Yhdistelyn tarkoituksena on säännönmukaisuuksien tai toisaalta poikkeavuuksien etsiminen (Hirsjärvi & Hurme 2011, s. 149). Erityisen tärkeää tässä vaiheessa onkin tutkijan ajattelutyön ja ymmärtämisen osuus: "*Löydetyt yhteydet*

olisi pystyttävä ymmärtämään ei vain empiirisinä ilmiöinä vaan myös teoreettisesti" (*ibid.*). Tässä tutkimuksessa yhdistelyn (Kuva 7, kohta 2c) tuloksena saadut otsikot on löydettyissä kohdista 5.2 (s.29-41) ja 6.2 (s.43-54).

Luokkien yhdistelyn jälkeen tarkasteltiin yleisellä tasolla, onko aineisto todella sitä, mitä nämä tyyppikuvaukset antavat ymmärtää, eli palattiin takaisin katsomaan kokonaisuutta (Kuva 7, kohta 3D) ja siirryttiin tulkintaosuuteen. (Kuva 7, kohta 3E) Hirsjärven & Hurmeen kirjassa (2011, s. 151) todetaan, että onnistuneen tulkinnan tunnistaa siitä, että tutkijan kanssa saman näkökulman omaksuva lukija voi löytää tekstistä samat asiat kuin tutkija, riippumatta lukijan mielipiteestä. Ilmiön teoreettista hahmottamista on tehty tämän työn pohdintaosuudessa eli kohdassa (8.1). Tässä yrityksille palautetuista raporteista on jätetty pois aineisto, joka ei kuulu yrityksen kiinnostuksen piiriin, samoin diplomityöstä on jätetty pois tutkimuskysymysten ulkopuolinen aineisto.

4.5 Raportointi

Runesonsin ja Höstin mukaan (2008) tapaustutkimus vaatii aina myös raportointia. Raportin tarkoituksena on kommunikoida tutkimusten tulokset ja arvioida tutkimuksen laatua. Riippuen yleisöstä, raportit voivat olla erilaisia (*ibid.*). Tämän tutkimuksen tärkein raportti on diplomityö. Tämän lisäksi molemmille yrityksille toimitetaan omat raportit, joissa korostetaan yritysten kanssa erikseen sovittuja asioita.

4.6 Tutkimuksen luotettavuuden arviointiin käytetyt mittarit

Seuraavissa kohdissa on esitelty tämän tutkimuksen luotettavuuden arviointiin valitut mittarit ja syyt valintoihin. Lisäksi on esitetty tutkimusetiikkaan liittyviä asioita. Tämän tutkimuksen luotettavuuden ja tutkimusetiikan noudattamisen arviointi tehdään luvussa kohdassa 8.2.

4.6.1 Luotettavuus kvalitatiivisen tutkimuksen piirissä?

Saaranen-Kauppinen ja Puusniekan (2006) mukaan luotettavuuden arviointi on tärkeä osa tieteellistä tutkimusta. Jotta tutkimusta voidaan pitää luotettavana, on sen arvioimiseen annettuja ohjeita, normeja ja arvoja pyrittävä noudattamaan (*ibid.*). Tunnetuimmat mittarit luotettavuuden arviointiin ovat kvantitatiivisen tutkimuksen piiristä validiteetti ja reliabiliteetti. Eskolan ja Suorannan mukaan (1998, s. 209) näiden kahden mittarin käyttö kvalitatiivisen tutkimuksen piirissä on kuitenkin herättänyt keskustelua. Kvalitatiivisen tutkimuksen luotettavuutta on ylipäättään kritisoitu. Verrattuna kvantitatiiviseen tutkimukseen, jossa tutkija peilaa tekemiään ratkaisuja jatkuvasti kattavuuteen ja luotettavuusmittareihin numeroin, kvalitatiivisessa tutkimuksessa tutkijan apuna saattaa olla ainoastaan tutkijan henkilökohtainen teorian oppineisuus, sekä omat ja muiden tutkijoiden ennakko-oletukset (*ibid.*). Eskolan ja Suorannan (1998, s. 209) mukaan onkin selvää, että laadullisen tutkimuksen luotettavuutta ja pätevyyttä on arvioitava eri tavalla:

kysymys luotettavuudesta kvalitatiivisen tutkimuksen piirissä rakentuu eri tavoin. Eskola ja Suoranta esittävät kirjassaan "Johdatus laadulliseen tutkimukseen" (1998) hieman erilaisia mittareita kvalitatiivisen tutkimuksen piiriin ja tässä tutkimuksessa tutkimuksen luotettavuutta on arvioitu näiden mittareiden avulla.

4.6.2 Tutkimuksen sisäinen ja ulkoinen validiteetti

Eskola ja Suoranta (1998, s. 214) esittävät perinteisen validiteetin käsitteen jakamista ulkoiseen ja sisäiseen validiteettiin. Sisäinen validiteetti (pätevyys), viittaa teoreettisten ja käsitteellisten määrittelyjen sopusointuun. Sisäisellä validiteetilla osoitetaan tutkijan tieteellisen otteen ja tieteenalan hallinnan voimakkuutta. Esimerkiksi voidaan kysyä, onko tutkija valinnut tutkimusstrategian kohteen mukaisesti.

Ulkoisella validiteetilla taas tarkoitetaan "*tehtyjen tulkintojen ja johtopäätösten sekä aineiston välisen suhteen pätevyyttä*" (Eskola & Suoranta, 1998). Tutkimushavainnon voidaan sanoa olevan on ulkoisesti validi kun se kuvaa tutkimuskohteen sellaisena kuin se on (ibid.). Ulkoinen validiteetti kiinnostaa tutkijaa erityisesti silloin, kun halutaan pohtia tutkimustulosten yleistettävyyttä: mitkä ovat niitä tilanteita, tai asetelmia missä tutkimustulokset voitaisiin yleistää?

4.6.3 Tutkimuksen reliabiliteetti

Eskolan ja Suorannan mukaan (1998, 214 s.) tutkimusaineisto on reliaabeli, kun se ei sisällä ristiriitaisuuksia. Reliabiliteetin tarkistamiseksi he ehdottavat kolmenlaisia toimia: indikaattorien vaihtoa, useampaa havainnointikertaa ja useamman havainnoitsijan käyttöä.

Indikaattoreiden vaihdolla tarkoitetaan ilmiön yhdenmukaisuuden osoittamista eri tavoin. Esimerkkinä tällaisesta tilanteesta olisi saman tilanteen tai kommentin toistuminen tutkimusaineistoissa eri henkilöiden toimesta. Tässä tutkimuksessa haastateltavien kommentteja tullaan käyttämään analyysiosiossa osoittamaan kuinka eri henkilöt puhuvat samoista asioista eri ilmaisuin.

Toinen reliabiliteetin tarkistamiseen liittyvä toimi oli useamman havaintokerran käyttö tutkimuksessa. Tällä varmistetaan aineistonkeruumenetelmän tarkkuus. Haastattelututkimuksessa tällä tarkoitetaan sitä, että sama asia kysytään eri muodoissa. Tämän tutkimuksen haastatteluissa tutkittavaa aihepiiriä tullaan lähestymään erilaisten kysymysten avulla useasta eri näkökulmasta. Tämän tutkimuksen rajauksen puitteissa ei ole mahdollista käyttää useampaa tiedonkeruumenetelmää, esimerkiksi haastatteluiden lisäksi havainnoida henkilöitä työtehtävissään.

Kolmas tapa tutkia reliabiliteettia on käyttää useampaa havainnoitsijaa. Tällä koetetaan varmistaa tutkimuksen objektiivisuus. Objektiivisuutta voidaan lisätä vaikka kahden toisiaan tukevan (Eskola & Suoranta, s. 215) menetelmän käytöllä. Toisaalta kahden

menetelmän käytöllä voi aineisto laajentua hallitsemattomaksi. Tässä opinnäytetyössä usean havainnoitsijan käyttö on jo lähtökohtaisesti poissuljettu vaihtoehto, koska työ tehdään yksilötyönä.

4.6.4 Muita kvalitatiivisen tutkimuksen arviointimittareita

Reliabiliteetin ja validiteetin lisäksi Eskola ja Suoranta (1998, s. 212-213) lisäävät kirjassaan uskottavuuden, siirrettävyyden, varmuuden ja vahvistuvuuden arviointivälineiksi kvalitatiivisen tutkimuksen arviointiin. Uskottavuudella tarkoitetaan tutkijan valintojen vastaavuutta tutkittavien käsityksiin (ibid.). eli tutkijan omia ennako-odotuksia ja tutkimuksessa tehtyjen johtopäätösten oikeellisuutta. Tutkija voi kysyä itseltään "olenko todella ymmärtänyt, mitä tutkittava on sanonut?" Tutkijan on siis tärkeä persoonastaan ja osaamistaustastaan huolimatta pyrkiä objektiivisuuteen.

Siirrettävyydellä tarkoitetaan tutkimustulosten siirtämistä toiseen kontekstiin (Eskola & Suoranta 1998). Siirrettävyyttä voidaan parantaa tutkimusmenetelmän, ympäristön ja tutkittavan kohteen tarkalla kuvaamisella. Laadullinen tutkimus on kuitenkin aina ainekertainen prosessi, eikä sen tutkimusmenetelmänä voi sanoa tukevan siirrettävyyttä. Esimerkiksi haastattelun tulokset ovat tuskin koskaan täysin samanlaisia, vaikka ne olisi tehty samalle tutkittavalle eri ajankohtana.

Varmuudella tarkoitetaan tutkimukseen vaikuttavien ennakkoehtojen huomioimista (Eskola & Suoranta 1998, s.213). Tutkijan tulee siis ottaa huomioon ennakoimattomat tekijät, joilla saattaa olla vaikutusta tutkimuksen tuloksiin. Tällaisia valintoja tehdään heti esimerkiksi tutkimusmenetelmää valitessa. Jotta tutkimuksesta voisi saada päteviä tuloksia, on valitun menetelmän oltava oikea.

Tutkimuksen vahvistuvuudella tarkoitetaan sitä, että "*tutkijan tekemät tulkinnat saavat tukea toisista vastaavaa ilmiötä tarkastelleista tutkimuksista*" (Eskola & Suoranta, 1998). Vahvistuvuutta voidaan siis arvioida muiden samaa ilmiötä tarkastelleiden tutkimusten avulla. Usein tutkimuksessa tehdäänkin esitutkimusta aiheesta erilaisten artikkelihakujen ja kirjallisuuskatsausten kautta. Tutkijan perehtyminen tutkittavaan aiheeseen on tärkeää. Näin tutkimuksessa tehtyjä tulkintoja voi peilata aiempiin teorioihin. Vahvistuvuuden miinuspuolena on tutkijan objektiivisuuden heikentyminen. Odotukset tutkimustuloksista ovat sitä suuremmat, mitä enemmän asiasta tiedät (ibid.).

4.7 Tutkimuksen yleistettävyys

Aiemmin mainitun ulkoisen validiteetin käsitteen lisäksi Sulkunen (1990, Saaranen-Kauppinen & Puusniekka 2006) esittävät lisähuomiona, että jotta tutkimustuloksia voitaisiin yleistää, on tutkimuksen aineisto pystyttävä kokoamaan tarkoituksenmukaisesti. Esimerkiksi haastateltavia valittaessa on tärkeää, että "haastateltavilla on suhteellisen samanlainen kokemusmaailma, kokemusta ja ensikäden tietoa tutkimuksen aihepiiristä,

ja kiinnostusta ja myönteinen suhtautuminen tutkimukseen (Eskola & Suoranta 1998). Tässä tutkimuksessa haastateltavat valittiin yhden projektin sisältä (per yritys), jotta näkemys prosessista olisi yhtenäinen.

Saaranen-Kauppinen ja Puusniekan mukaan (2006) tutkittaessa yksittäistä tapausta riittävän perusteellisesti, saadaan esille se, mikä ilmiössä on merkittävää ja mikä saattaisi toistua myös yleisemmän tason tarkastelussa. Yksittäisessäkin on jotain yleistä (ibid.). Yleistyksiä ei voida kuitenkaan tehdä suoraan aineistosta, vaan aineistosta tehdyistä tulkinnoista (ibid.).

Laadullisessa tutkimuksessa yleistämistä voidaan ajatella myös siirrettävyytenä (Saaranen-Kauppinen & Puusniekka 2006) (ks. luku 4.5.4). Ensinnäkin joitakin teoreettisia käsitteitä voidaan soveltaa toisenlaisissa yhteyksissä kuin missä niitä alun perin on käytetty (ibid.). Toisekseen siirrettävyydellä voidaan tarkoittaa tutkimustulosten soveltumista toiseen toimintaympäristöön (Eskola & Suoranta 1998, s. 66-68.). Tulosten siirrettävyyteen vaikuttaa se, miten samanlaisia ovat tutkittu ympäristö ja ympäristö, johon tuloksia on tarkoitus siirtää (Eskola & Suoranta 1998). Eskolan ja Suorannan (1998) mukaan arviointia edesauttaa mm. perusteellinen tutkimuksen kuvaaminen (luvut 3.3.3 ja 5.5).

Tapaustutkimuksen kohdalla on Saaranen-Kauppinen ja Puusniekan (2006) mukaan erityisesti pohdittu edustavuutta ja tulosten yleistämistä. Vaikka tapaustutkimuksessa voidaanakin käyttää niin kvantitatiivisia kuin kvalitatiivisiakin menetelmiä, tutkimuksen arvioinnissa painottuvat yleensä laadullisen tutkimuksen arviointiin liittyvät seikat, ellei kyse sitten ole pelkästään määrällisistä aineistoa sisältävästä tutkimuksesta (ibid.).

4.8 Tutkimusetiikka

Tutkimuksessa on pyritty noudattamaan Tutkimuseettisen neuvottelukunnan (2012) laatimia hyvän tieteellisen käytännön ohjeita tutkijoille. Hyvä tieteellinen käytäntö on osa tutkimusorganisaatioiden laatuja järjestelmää (ibid.). Tutkimusetiikkadokumentti (ibid.) on jaettu yhdeksään kohtaan, joista seuraavassa lyhyet esittelyt

- Rehellisyys, huolellisuus, tarkkuus
- Tarkastelua kestävät tutkimusmenetelmät, avoimuus tuloksista
- Muiden tutkijoiden tuotannon kunnioittaminen
- Tutkimuksen huolellinen suunnittelu, toteutus, raportointi ja tallennus
- Tutkimusluvut, ennakoarviointi
- Tutkijan, tutkittavien ja muiden osapuolien oikeuksista sopiminen
- Rahoitus ja sidonnaisuudet ilmoitetaan asianomaisille
- Esteellisyyksien tiedostaminen ja noudattaminen
- Tutkimusorganisaation hyvä henkilöstö- ja taloushallinto

Tutkimusetiikkadokumentissa korostetaan lisäksi, että käytännön noudattamisesta vastaa ensisijaisesti jokainen tutkija itse (Tutkimuseettinen neuvottelukunta 2012). Vastuu on kuitenkin myös jaettu ja kuuluu koko tiedeyhteisölle (ibid.). Tämän tutkimuksen eettisiä ulottuvuuksia arvioidaan työn luvussa 8.3.

5. TULOKSET: YRITYS 1

Tässä luvussa kuvataan ensimmäisen tutkimuksen tutkimusaineisto ja tulokset. Käytetty tutkimusmenetelmä on avattu luvussa 4.

5.1 Kuvaus

Tutkimus tehtiin eräälle suurelle Suomessa toimivalle kuluttajaelektroniikkaa ja työlaitteita valmistavalle sulautettujen järjestelmien toimittajalle. Tutkimuksen keskeisenä tavoitteena oli saada päivitettyä yrityksen käytössä oleva tuotantoprosessimalli, joka ei vastaa nykyistä toimintaa, lisätä puuttuvat käyttäjäkokemustyön osuudet prosessimalliin sekä löytää ratkaisuja prosessin parantamiseen. Tähän työhön tutkimustuloksista on koottu toimintaan liittyneet ongelmakohdat, niiden ratkaisuehdotukset, sekä käyttäjäkokemustyön teko prosessissa.

Haastattelut yritykseen tehtiin keväällä 2016. Raportti yritykseen palautetaan diplomityön valmistumisen jälkeen.

Taulukossa 2 on esitelty haastateltavien työnimikkeet ja työkokemus ohjelmistoalalta sekä nykyisessä yrityksessä että kokonaisvuosina. Työnimikkeet ovat haastateltavien itse ilmoittamia, rooli yrityksessä on tutkijan lisäämä tieto.

Taulukko 2. Yritys 1, haastateltavien työnimikkeet ja kokemus yrityksessä sekä alalla yhteensä.

Haastateltavan koodi	Työnimike / rooli yrityksessä	Työkokemus vuosina yrityksessä / ohjelmistoalalla
H1	Senior HW-Designer / Projektinvetäjä	5/19
H2	Senior HW-Designer / elektroniikkasuunnittelija	3/17
H3	Chief Engineer / SW-esimies	8/14
H4	HW Chief Engineer	10/13
H5	Software Design Engineer	0,5/12
H6	UX Designer	1,5/16

5.2 Tutkimuksen tulokset

Seuraavissa kohdissa käydään läpi tutkimuksessa esiin tulleet haasteet tuotekehitysprosessissa. Löydökset on esitetty taulukoissa 3-6. Taulukkojen ensimmäisessä sarakkeessa on haasteen kuvaus ja sen esittäjä numerolla. Toinen sarake sisältää joko kirjallisuuden avulla etsityn tai haastatteluissa esiin tulleen haastateltavan esittämän kehitysehdotuksen. Kirjallisuus on merkitty taulukkoon merkinnällä (K), haastateltavalta tullut paranusehdotus haastateltavan tunnuksella (H-)

Löydetyt haasteet on jaettu neljään kategoriaan: käyttäjäkokemustyö (5.2.1), projektin vaiheet ja synkronointi (5.2.2), tiedonsiirto ja kommunikointi (5.2.3) sekä projektinjohdaminen ja työn organisointi (5.2.4). Jokaiselle näistä kategorioista on varattu alakohta, joissa ensin esitetään taulukko haasteista ja kehitysehdotuksista ja tämän jälkeen käydään taulukon sisältöä ja lähdekirjallisuus tarkemmin läpi tekstimuodossa.

5.2.1 Käyttäjäkokeemustyöhön liittyvät haasteet

Yritykseen palkattiin 18 kuukautta sitten käyttäjäkokemussuunnittelija. Käyttäjäkokeemustyön osoittaminen tietylle henkilölle on siis yrityksessä vielä uusi asia. Yksi käyttäjäkokemussuunnittelija osallistuu useaan projektiin yhtä aikaa. Kaikkia projekteja, joissa käyttäjäkokemussuunnittelija on ollut mukana, ei ole vielä saatu valmiiksi. Tämä tulee ottaa huomioon tutkimuksen tulosten tulkinnassa. Jokaisessa kehittäjiä koskevassa haastattelussa mainittiin, että suunnittelutyö on parantunut käyttäjäkokemustyöntekijän palkkaamisen myötä. Käyttäjäkokeemustyön tekemiseen kaivattiin myös lisäresursseja. Taulukossa 3 on lueteltu käyttäjäkokemustyöhön liittyviä haasteita ja kehitysehdotuksia.

Taulukko 3. Käyttäjäkokeemustyöhön liittyvät haasteet ja kehitysehdotukset.

Haaste	Kehitysehdotus (K: Kirjallisuudesta, H: Haastattelusta)
Loppukäyttäjiltä ei juurikaan saada palautetta. (H2, H3, H4, H5). Kritiikki saatetaan tulla läpi, mutta positiivinen palaute ei.	Sitoutetaan asiakas koko tuoteprosessiin. Tehdään parhaita arvauksia, kunnes saadaan käyttäjäpalautetta suunnittelun avuksi. (K)
Tuote tehdään asiakkaan edustajalle, ei loppukäyttäjälle. (H2, H6) Tieto loppukäyttäjistä tulee välikäsien kautta ja muuttuu matkalla. (H6)	Loppukäyttäjä mukaan alusta asti. Jatkuva kommunikointi prosessin ajan asiakkaan kanssa. (H2) (K)
Vaikka käyttäjäkokemustyöntekijällä saattaa olla tieto loppukäyttäjistä, ei tieto välttämättä saavu kehittäjille asti. (H5)	Loppukäyttäjä mukaan määrittelyyn. Persoonat aikaisessa vaiheessa kehittäjille. Jatkuva kommunikointi. (K)
Määrittelyvaihe: Yhteisen näkemyksen muodostaminen tuotteen käyttäjäkoke-	Otetaan loppukäyttäjät mukaan suunnitteluprosessiin. (H6) (K)

muksesta on hankalaa. (H1, H2, H3)	
Määrittelyt käyttäjäkokemukselle liian yleisiä esim: "Supervau", "Helppokäyttöinen". (H5)	Muutetaan määrittely jatkuvaksi prosessiksi, jossa määrittelyt tarkentuvat ajan myötä. Käyttötapaukset, systeemitapaukset käyttöön. (K)
Myydyn käyttäjäkokemustyön määrä ei vastaa tarvetta. (H6)	Otetaan käyttäjäkokemustyö mukaan myyntiprosessiin. (K)
Käyttäjäkokemustyötä tehdään liian vähän. Enemmän käytettävyyshenkilöitä firmaan. (H4)	Luodaan tarkistuslistat käyttäjäkokemusasioista erilaisiin tapauksiin kaikille työntekijöille. (K)

Loppukäyttäjät eivät ole mukana prosessissa

Sulautettujen järjestelmien laitteistokehityksessä käyttäjäkokemukseen liittyvät teemat saattavat olla sellaisia, joita ei välttämättä heti osaa mieltää käyttäjäkokemussuunnitteluun liittyviksi teemoiksi. Laitteistosuunnittelussa tulee ottaa huomioon esimerkiksi kuinka usein laitetta tullaan käyttämään, millaisissa olosuhteissa käyttö tapahtuu, kuinka kauan akun tulee kestää, tai kuinka verkkoyhteys saadaan toimimaan. Laitteiston tulee yleensä myös mahtua sisään johonkin tiettyyn koteloon tms.

Yrityksen työntekijöillä oli mielikuva, että he eivät niinkään tee laitteistoa ja sen ohjelmistoa käyttäjiä varten, vaan ennemminkin yrittävät piilottaa sen heiltä. ”*Mitä vähemmän siitä elektroniikasta näkyy käyttäjälle, niin sen parempi*” (H3). Useissa haastattelussa kysymys ”teetkö käyttäjäkokemustyötä?” sai pienen pohdinnan jälkeen haastateltavan itsensäkin hämmentymään kuinka käyttäjäkeskeistä työ lopulta onkaan.

Ongelmat käyttäjäkokemukseen liittyen tunnistettiin hyvin. Jokaisessa haastattelussa (5/6) kehittäjät kertoivat, ettei loppukäyttäjä näy tuotekehitysprosessissa. ”*Varsinainen loppukäyttäjä jää mysteeriksi*” (H5). Kehittäjällä ei ole välttämättä lainkaan tietoa siitä, millaiselle käyttäjälle hän tekee vaikkapa käyttöliittymää. Kehittäjät kertoivat myös (5/6), ettei tuotteesta saada lainkaan loppukäyttäjäpalautetta. Jos palautetta tulee, se on tuotereklamaatiota. Joissain tapauksissa asiakkaan edustajilla on toisen käden tietoa käyttäjistä, toisinaan ei lainkaan. Yhteisen näkemyksen muodostaminen on myös hankalaa, sillä loppukäyttäjän mielipidettä, jonka perusteella päätöksiä tuotteen ominaisuuksista voitaisiin tehdä, ei ole saatavilla. Mikäli tuotetussa käyttöliittymässä on jotain suurempaa vikaa, aloitetaan uusi projekti. Positiivisista ratkaisuista ei siis saada palautetta.

Gothelf & Seidanin (2013) mukaan loppukäyttäjän pitäisi olla mukana jo tuotteen määrittelyvaiheessa, parhaassa tapauksessa läpi koko prosessin. Monet yrityksen rakentamista tuotteista ovat uusittuja versioita vanhoista, ja erityisesti näiden laitteiden käyttäjät olisi tärkeää saada mukaan prosessiin. Mahdolliset virheet vanhassa versiossa on hyvä saada korjattua uuteen tuotteeseen.

Ketterässä prosessissa syntyy jatkuvasti uusia välituotteita. Kun jotain näkyvää saadaan kehitettyä nopealla rytmillä, voidaan käyttäjätestausta tehdä läpi prosessin. Käyttäjäpalaute saadaan näin muuallakin kun prosessin loppuvaiheessa. Mikäli prototyyppi ei laitteistopuolella onnistu, voidaan sitä kuitenkin onnistuneella suunnittelulla soveltaa ohjelmistopuolella.

Käyttäjäkokeilusuunnittelun tulokset eivät välity toteuttajille

Jo edellä mainittiin, että loppukäyttäjän tarpeet eivät välity kehittäjille riittävän aikaisessa vaiheessa projektia. Persoonien käyttäminen (Gothelf & Seidan 2013) on yksi tapa saada kehittäjät ymmärtämään enemmän loppukäyttäjää, eli henkilöä, jolle tuote on tuossa. Persoonat ovat tuotteiden käyttäjien kuvauksia (ibid.). Persoonia on haastattelujen mukaan käytetty jonkun verran, mutta lisätietoa käyttäjistä kaivataan. Gothelf & Seidan (2013) suosittavat kirjassaan Lean UX persoonien käyttämistä tiedon jakamisessa. Heidän mukaansa persoonien tekoprosessi on muokattu hieman aiempaa erilaiseksi. Prosessi aloitetaan oletuksilla, ja sen jälkeen tehdään tutkimusta näiden oletuksien validoimiseksi. Tällaisia persoonia kutsutaan "protopersooniksi". Mikäli ensikäden tietoa käyttäjistä ei heti prosessin alussa saada, käytetään hyväksi parhaista arvauksista ja prosessin edetessä arvaukset muuttuvat tiedoksi.

Sen lisäksi, että suora yhteys loppukäyttäjään puuttuu, käyttäjäkokeiluun liittyvät määrittelyt tuntuvat välillä turhan yleisiltä ja jättävät kehittäjille liikaa valinnanvapautta, mikä saattaa aiheuttaa ongelmia myöhemmässä vaiheessa. Jotta sitoutuminen tietyn tehtävän suorittamiseen voi tapahtua, on tavoitteen oltava niin selkeä, että siihen voidaan sitoutua. Asiakas ei useinkaan osaa määrittellä millaista käyttäjäkokeilusta tuotteelta toivoo. ”Asiakas tilaa jotain ’super vau:ta’ tai ”helppoa käytettävyyttä”(H5)”. Loppujen lopuksi asiakas ei kuitenkaan tiedä millaista työtä tällaiset vaatimukset aiheuttavat. Asiakkaan kiinnostus käyttäjäkokeilutyöhön satsaamiseen saattaa lopahtaa, kun saadaan tietää, että loppukäyttäjää pitäisi haalia ja haastatella. Onkin käyttäjäkokeiluhenkilön vastuulla tasapainotella asiakkaan vaatimusten kanssa ostetun käyttäjäkokeilumäärän ehdoilla.

Projekteissa käytetään käyttötapauksia selventämään vaatimuksia. Käyttötapaukset eivät ole täysin selkeitä laitteistopuolella ja (PwC Technology Institute 2013) ehdottavatkin laitteistovaatimusten määrittelyyn ”systeemitarinoita”. Tuotteen käyttämisen sijaan katsotaankin asiaa systeemin näkökulmasta, eli asetetaan ensiehdot, suoritusvaiheet ja jälkiehdot tietyille toiminnolle. Laitteistopuolella on tärkeämpää kuvata toiminnot systeemin kannalta, ottaen huomioon myös käyttäjän.

Yhteisen näkemyksen muodostaminen tuotteesta on hankalaa

Yhteisen näkemyksen muodostaminen tuotteesta koettiin haasteeksi. Asiakas, käyttäjäkokeiluhenkilö ja laitteistosuunnittelija katsovat kaikki kokonaisuutta hyvin erilaisista

näkökulmasta. Yhtenäisen näkemyksen kokoamiseen saattaisi auttaa alkuvaiheessa tapahtuva pitempiaikainen saman pyöreän pöydän äärelle kokoontuminen.

Esimerkkinä huonosta käyttäjäkokemussuunnittelusta on tapaus, jossa valmis asiakas on sanonut, että ei ole kuullut valitusta vanhasta laitteesta, mennään samantyyllisellä. Laitte on tehty lähes samanlaiseksi ja tuotteen ollessa valmis asentaja on kertonut, että kaikkien liittimien olisi tullut olla erilaisia, kaikki ovat surkeita. Näin voi käydä kun tieto vaatimuksista tulee asiakkaan markkinointi- tai myyntipuolen kautta, ja varsinainen käyttäjä jää taka-alalle. Yksi haastateltavista ehdotti laitepäivitysten tapauksessa hinnan nostoa, mikäli asiakkaalta ei saada palautetta edellisestä laitteesta käyttäjiltä tai asentajilta. Riskinä on siis se, että jos käyttäjiä ei saada mukaan projektin alkuun, asiakas saattaa haluta muuttaa vaatimuksia jatkuvasti projektin edetessä. Tämä on riski erityisesti laitteistoprosessille. Muuttuneet vaatimukset lisäävät projektin kestoja, ja ovat siis hinnannääräytymistekijöitä.

Käytettävyytyötä tehdään liian vähän, tarvitaan lisää resursseja

Tutkimuksessa kävi selville, että yrityksessä ymmärretään käyttäjäkokemustyön merkitys yritykselle. Eräs haastateltavista kokosi, että käyttäjäkokemussuunnittelu on valtti asiakkaan suuntaan ja siihen panostaminen nähdään hyvänä kehityssuuntana. Käyttäjäkokemussuunnittelu on yrityksessä kuitenkin vielä aika pienessä roolissa, eikä se välttämättä saa itseään tarpeellisella tavalla esille. Vaikka työtä arvostetaan, sitä ei oteta niin vakavasti kuin pitäisi. Yrityksessä myös tiedostetaan, että käyttäjäkokemustyö jää toissijaiseksi jos se jätetään kehittäjien vastuulle (H3). Käyttäjäkokemustyöhön kaivattiin siis lisää resursseja ja yksi haastateltavista koki (H5), että nimenomaan uusi vähemmän kokenut työntekijä voisi tuoda uutta näkemystä asioihin.

5.2.2 Projektin vaiheisiin ja synkronointiin liittyvät haasteet

Yrityksessä on käytössä prosessimalli, jonka toimivuudesta on työntekijöiden välillä näkemuseroja. Vaikka malli onkin vain taustalla toimiva oheistus projektin läpiviemiseen, hyvä prosessimalli ohjaa työntekijöitä tekemään eri vaiheisiin liittyvät toimet systemaattisesti. Tässä tutkimuksessa suurimmiksi haasteiksi nousivat ketteryyteen ja eri vaiheiden linkityksiin liittyvät kysymykset. Taulukossa 4 on lueteltu projektin vaiheisiin ja synkronointiin liittyviä haasteita ja kehitysehdotuksia.

Taulukko 4. Projektin vaiheisiin ja synkronointiin liittyvät haasteet ja kehitysehdotukset.

Haaste	Kehitysehdotus (K: Kirjallisuudesta, H: Haastattelusta)
Määrittelyvaihe ei ole tarpeeksi ketterä. (H2)	Asiakkaan sitouttaminen prosessiin, yhden suunnittelupäivän sijaan viikko yhteisen pöydän ääressä, dokumenttien

	muodon mietintä. (K)
Vesiputousmalli ei toimi (H5). Ketteryyttä kaivataan. (H6, H5, H4)	Ketterien käytänteiden lisääminen kaikilla osa-alueilla. (K)
Laitteisto ja ohjelmisto tehdään omissa prosesseissaan, vaikka kyseessä on sulautettu systeemi. Prosesseilla ei ole ainakaan kunnolla määriteltyjä yhtymäkohtia. (H2, H4)	Tarkistuspisteet laitteiston suunnasta ohjelmiston suuntaan ja takaisin. Myös käyttäjäkokemussuunnittelu ja mekaniikka mukaan katselmointeihin. (H2) Prosessimalliin oheistukset yhteydenpidosta. (K)
Linkitystä prosessin eri vaiheista myyntiprosessiin ei ole. Dataa hinnan muodostumisesta ei kerätä. (H2)	Linkitykset kuntoon. Suunnitelma historiadatan keruulle. (K)
Laitteiston saapuminen valmistuksesta testaukseen aiheuttaa pienimuotoisen kaaoksen. Toimintojen järkevä priorisointi puuttuu testausvaiheen alusta. (H2, H3)	Priorisointi- ja yhteistyösuunnitelma toimille prototyypin saapumisen jälkeen. Asioiden listaaminen kaikkien osapuolien kesken. (H3)

Määrittelyvaiheeseen liittyy monia ongelmia

Määrittelyvaihe koetaan yrityksessä hankalaksi, koska asiakas ei tiedä mitä haluaa ja yhteisymmärrys eri osapuolien välillä on hankala saavuttaa. Tämä on yleinen ongelma sulautettujen järjestelmien valmistuksessa (Orr 2004). Orr (2004) kuvaakin osuvasti vaatimusmäärittelypalaverien ongelmaa: Käyttäjä ei tiedä mikä on mahdollista, asiakas ei tiedä mitä käyttäjä haluaa, markkinointitiimi tietää mikä myy, mutta ei mikä on käytettävää, kehittäjä tietää mitä voi kehittää. Yhteistä näkemystä asiaan näyttäisi olevan mahdotonta saada. Yksi haastateltavista kuvaa kuitenkin osuvasti: ”*Jos ei tule pientä riitaa siinä alkumetreillä uuden asiakkaan kanssa, niin se ei voi onnistua kauhean hyvin se projekti.*” (H2) Ongelmakohtat on siis tärkeä tuoda esiin jo alkuvaiheessa, jotta asioista päästäisiin yhteisymmärrykseen mahdollisimman nopeasti. Prototyyppien rakentaminen alusta asti olisikin ehdottomasti paras saada käyttäjät sitoutettua haluttujen tulosten ja rajoitteiden määrittelyyn (Orr 2004).

Määrittelyt tulevat yrityksessä usein asiakkaalta ja yritys tulee työhön mukaan vasta vaiheessa, jossa työtä jaetaan jo sopiviin osakokonaisuuksiin. Vaikka määrittelylle on yrityksessä varattu oma vaiheensa prosessin alussa, vaatimusmuutokset ovat osa ohjelmisto- ja laitteistotuotannon luonnetta ja vaatimusmäärittelyä tehdään uudestaan useasti prosessin aikana. Tämän olisi hyvä näkyä myös prosessimallisissa ketteränä vaatimusmäärittelykäytäntönä (H2). Määrittelyt tehtäisiin yhdessä asiakkaan kanssa ja toisaalta määrittelyn tulisi olla koko projektin läpi jatkuva prosessi.

Toisin kuin usein ajatellaan, ketteryys ei tarkoita sitä, etteikö vaatimuksia tuotteelle kerättäisi. Jokainen tuote tarvitsee riittävän hyvin dokumentoidut vaatimukset, jotka mielellään johtuvat loppukäyttäjän tarpeista (Orr 2004). Vaatimuksia ei kuitenkaan tule mitata niiden pituuden mukaan. Vesiputousmallin laaja määrittelydokumentti prosessin alussa ei välttämättä sovi maailmaan, jossa vaatimukset muuttuvat koko prosessin ajan. Vaatimusten tulisi aina määritellä jotain mitä voi ja pitää rakentaa tuotteeseen (ibid.). Orr (2004) jakaa artikkelissaan vaatimukset kahteen luokkaan "tuotoksiin" (engl. output) ja "rajoitteisiin" (engl. constraints). Kaikki vaatimukset, jotka eivät ole tuotoksia (output) ovat rajoitteita. Tällaista näkemystä kutsutaan suorite- tai tavoiteorientoituneeksi tavaksi jakaa vaatimukset. Orrin (ibid.) mukaan halutut tuotokset täytyy määritellä ensin, vasta sitten rajoitteet. Orr lisää myös, että kaikkien edellä mainittujen vaatimusten tulee olla testattavia. Mikäli jotain ei voi testata, sitä ei tarvitse. Vaatimusten takana tulee aina olla loppukäyttäjän tarve ja käyttäjät eivät aina tiedä tarkalleen mitä tarvitsevat. Tässä yrityksessä olisikin tärkeää, että käyttäjäkokemussuunnittelija (ja toisaalta joku joka ymmärtää liiketoimintapuolen tarpeita) on mukana vaatimusmäärittelyssä prosessin alusta loppuun. On toki projekteja, joissa alussa todetaan, ettei tuotteeseen makseta käyttäjäkokemustyön tekemisestä tai kyseessä on vain laatikko, jossa ei ole minkäänlaista käyttöliittymää. Nämä ovat erikoistapauksia, jotka vahvistavat säännön.

Linkitykset prosessien väliltä puuttuvat

Yrityksen standardiprosessi määrittelee neljä erilaista polkua, eli ohjelmisto-, elektrooniikka- ja mekaniikkasuunnittelun polut sekä testauksen. Tämän lisäksi mukana on käyttäjäkokemustyö, joka ei näy prosessimallissa. Jokainen näistä kulkee omaa tahtiaan, eikä selkeää edustusta systeemitason suunnittelulle ole olemassa. Tämä saattaa näkyä jopa niin, että jokaiselle osa-alueelle on olemassa omat dokumentit, ja yhteen projektiin liittyvä tieto saattaa olla monessa eri paikassa, muttei silti tarvittaessa saatavilla. Tämä luonnollisesti vaikeuttaa myös kommunikointia, mikäli projektidokumentaatio on hankalasti saatavissa työntekijän sitä tarvitessa.

Yksi suurimmista ongelmista sulautettujen järjestelmien valmistamisessa on projektin useiden erilaisten osa-alueiden yhdistäminen. Yksi haastateltavista kommentoi osuvasti: *"Laitteisto tehdään omassa prosessissa ja ohjelmisto omassaan. Näillä ei ole yhtymäkohtia, vaikka lopputuote on sulautettu järjestelmä, jossa osa-alueiden sulava kommunikointi on 'tuotteen tärkein ominaisuus'."* (H2) Ongelma on suuri, mikäli järjestelmällistä kommunikointia eri osa-alueiden välillä ei ole määritelty. Linkitykset prosessien välillä on yrityksessä hoidettu työntekijöiden välisellä yhteydenpidolla. Tapoja yhteydenpitoon ei ole tarkasti määritelty, mutta se on päivittäistä, jopa tunneittain tapahtuvaa. Työntekijät ovat siis löytäneet tavan linkittää, mutta prosessimallissa tämä ei näy, ja saattaa aiheuttaa ongelmia uusille työntekijöille. Lisäksi linkitykset prosessimallin tasolla auttaisivat työntekijöitä. Haastatteluissa mainittiin myös, että linkityksistä huolehtiminen on loppukädessä projektipäällikön vastuulla. Työntekijät ovat kuitenkin päävas-

tuullisia omasta työstään, ja prosessimallissa olevat tarkastuspisteet voisivat sujuvoittaa työtä entisestään. Myös työntekijöiden kokoaminen tiimeiksi voisi auttaa pitämään selkeämmin yhdessä eri osa-alueiden välillä työskentelevät ihmiset. Nykyprosessissa varsinaisia tiimejä ei ole nimetty, vaan työntekijät tietävät vain henkilöt keiden kanssa heidän on oltava tekemisissä prosessin aikana.

Myös linkitys muihin sidosprosesseihin, on löyhä. Tuotteen hinta-arviot, jotka johdetaan tehdyn työn määrästä, ovat suunnittelijoiden arvausten varassa. Historiadataa hinnan muodostumisesta ei kerätä systemaattisesti.

Työntekijät toivovat lisää ketteryyttä prosessiin

Haastatteluissa toivottiin joko koko prosessin tai ainakin joidenkin sen vaiheiden ketteröitämistä. *”Hankala edes luoda yhtenäistä mallia prosessin läpiviemiseen. - mä en niinkun oo kokenu sitä prosessimallia soveltuvaks siihen oikeestaan millään tasolla.” (H1)* Scrum-pohjainen laitteistokehitys nähdään ideaalina, mutta jotta sen saisi toimimaan, pitäisi projektien olla isoja (H1). Yksi haastateltavista kertoi, että Scrumia käytetään ohjelmistopuoleen ja vesiputousta rautaläheiseen osuuteen. Prosessimalli on myös sinänsä iteratiivinen, että yleensä tuotteen prototyypistä tehdään kaksi versiota (H3), ja kierrokset on projektin alussa laskettu prosessin kokonaiskestoon.

Laitteistoprosessin muuttaminen ketterämmäksi on haaste. Ongelmana on esimerkiksi komponenttien pitkä tilausaika. Layout-suunnittelun ja piirikaavioiden pitää olla valmiina ennen komponenttien tilausta (H3). Tähän työhön voi hyvinkin mennä yhteensä muutama kuukausi. Tämä onkin laitteistoprosessin ketteröitämisen ongelman ydin. Joidenkin artikkeleiden mukaan (Lehtonen et al. 2014, Grenning 2002) laitteistoprosessia voidaan kuitenkin pikkuhiljaa muilta osin viedä ketterämpään suuntaan. Hyvillä käytännöillä voidaan saada joitakin ketteryyden hyviä puolia korostumaan, ilman että vahvasti sisäistetystä suunnitelmapohjaisuudesta tarvitsisi kokonaan luopua.

Eräs haastateltavista näki, että yrityksen työntekijät olisivat valmiita muutokseen kohti ketteryyttä. *”Jos katsoo agile manifeston sivuja, ja näyttää sieltä ne neljä kohtaa, mitä pitäis ruveta toteuttaa sieltä niin kaikki sanoo, et ilman muuta.” (H5)* Kaikki yrityksessä ymmärtävät, että muutosten tullessa niihin tulee reagoida. Näin toimitaan toki nytkin. Olisi kuitenkin tärkeää, että tekeminen olisi näiltä osin hallittua. Yksi haastateltavista (H5) näki pitkän palautesyklin tämän hetken toiminnassa suunnitelmapohjaisen lähestymisen suurimpana ongelmakohtana. Nykyprosessissa palautetta saadaan pahimmillaan vasta puolen vuoden kehitysjakson jälkeen. Vaikka muutoksia tulee koko ajan, niin muutoksiin ei reagoida, vaan niiden kanssa yritetään elää. Toimintatapaa muutosten hallitsemiseen ei ole olemassa.

Yksi haastateltavista (H6) kertoi näkevänsä Kanbanin sopivimpana ja selkeimpänä menetelmänä heidän prosessiinsa. Kanban-taulu auttaisi jäsentämään ja visualisoimaan mitä aiotaan tehdä. Työntekijä lisäsi myös, että Scrumista tuttu ajattelumalli, jossa tuote

pilkotaan kehitysjonoon ominaisuuksiksi, olisi hyvä saada rantautumaan koko yritykseen. Tällä tarkoitettiin sitä, että projektin toteutusta ositettaisiin enemmän vaikkapa käyttötapausten avulla, eikä niinkään esitutkimuksen (engl. feasibility study) avulla. Kommunikoinnin nähtiin parantuvan tätä kautta.

Myös esimerkiksi XP:stä tuttu (Grenning 2002) parikoodaus on koettu hyvänä uutena muutoksena. Lisäksi Scrumin roolijaosta (Schwaber & Sutherland 2013) nähtäisiin olevan hyötyä. Näin vastuunjakoon tiimin ja muiden työntekijöiden välille saataisiin selkeyttä.

Prototyyppien valmistumista odotetaan, ja kun ne saapuvat, syntyy kaaos

Prototyyppien valmistuminen kestää kauan (H1). Tämä näyttäisi olevan laitteistokehityksessä yleinen ongelma, ja asian ratkaisemiseen tarvittaisiin lisätutkimusta. Lisäksi mikäli prototyyppien valmistuksen tiedetään kestävän tietyn ajan, voidaan siihen varautua jo projektisuunnitelmassa. Yrityksessä kuitenkin kerrottiin ongelmaksi myös se, että kun ensin odotellaan prototyyppien saapumista pitkään, niiden saapuessa asioita ei ehkä osata tehdä oikeassa järjestyksessä. Syntyy pienimuotoinen kaaos, kun kaikki projektin osapuolet haluavat suorittaa protoilla testauksiaan. ”-- on kauhee hinku saada se rauta käteen ja sit kun se tulee käteen, niin -- unohdetaan että olikin kiire. Kaikki on ikään kun sen jälkeen et ”nyt se on, nyt rauhoitutaan” vaikka silloin -- leviää se työ sitten paljon laajemmaksi.” Eri osa-alueiden pitäisi siis sopia keskenään kuka tekee mitäkin prototyyppien saavuttua ja missä järjestyksessä asioita tehdään.

Tähän projektin vaiheeseen tarvitaan suunnittelua ja rivakkaa ohjien käsissä pitoa projektipäälliköltä. Tämä koettiin myös ongelmaksi: vastuu isojen osa-alueiden koossa pitämisestä kaatuu yhden henkilön hoidettavaksi. Mikäli toimittaisiin selkeissä tiimeissä, kommunikointi voisi olla helpompaa. Vastuuta prosessin aikataulussa pitämisestä ja osa-alueiden kommunikoinnista voisi antaa tiimin sisälle. Tässä vaiheessa prosessia pitää myös miettiä myös lopputuotteen kannalta tärkeimmät asiat, eli priorisoida virheiden korjaus. Tästä vastuu kuuluu sille, joka on vastuussa koko tuotteesta asiakkaalle, eli tämän yrityksen kohdalla projektipäällikkö. Kaikkia virheitä ei tarvitse korjata, jos eivät ne ole merkittäviä.

5.2.3 Tiedonsiirtoon ja kommunikointiin liittyvät haasteet

Edellisen kohdan haasteet ovat osaltaan kommunikointiin liittyviä ongelmia. Taulukossa 5 on lueteltu tiedonsiirtoon ja kommunikointiin liittyviä haasteita ja kehitysehdotuksia.

Taulukko 5. Tiedonsiirtoon ja kommunikointiin liittyvät haasteet ja kehitysehdotukset.

Haaste	Kehitysehdotus (K: Kirjallisuudesta, H: Haastattelusta)
Uusissa projekteissa valtava uuden tiedon määrä. Tuotoksia ei dokumentoida uudelleenkäytettävään muotoon. "Pyörä keksitään aina uudelleen ja uudelleen." (H1, H2)	Uudelleenkäytettävyyteen panostaminen. Työntekijöille vapaus valita omaa osaamista vastaavat työtehtävät. (H2)
Turhia palavereita. Turhia osallistujia palavereissa. (H3)	Palaverien tarkoituksen selvittäminen, palaverijohtamisen parantaminen, palaveriajan rajoittaminen. Tiedon talteen keruu. (K)
Kommunikointi neljän osa-alueen (UX, HW, SW, Mekaniikka) välillä vajavaista. (H4, H6)	Prosessikuvauksen tarkentaminen ja linkitysten lisääminen. (H4)
Yhteistyö toimipisteiden välillä ja koko firman kesken rajoittunutta (H2). Projektin henkilöstön työpisteet hajallaan, jopa eri toimipisteissä (H6). Ei ole aikaa auttaa toisia projekteja.	Henkilöitä, jotka eivät ole 100% mukana sidottuina projektiin (H2). Resursoitua aikaa myös muuhun, kuin oman projektin edistämiseen. (K)

Resursointi toteutetaan henkilöittäin

Kun prosessimalli ei selkeästi oheista työntekijöitä kommunikoimaan tietyissä vaiheissa eri osa-alueiden ja sidosryhmien kanssa, voi myöhemmässä vaiheessa prosessia syntyä ongelmia. Kommunikointiin liittyväksi haasteeksi koettiin erityisesti yleinen tiedonkulku. Sulautettuja järjestelmiä tuottaessa uuden tiedon määrä on usein valtava. Tätä lisää yrityksen prosessi- ja yritysmallin oheistus siihen, että työntekijät myydään projekteihin henkilöresursseina, mikä tarkoittaa sitä, ettei yritys käytä hyväksi niinkään työntekijöiden erityisosaamista vaan jos projektiin myydään vaikkapa kaksi ihmistä, tekevät he projektin alusta loppuun riippumatta projektin luonteesta ja työntekijöiden osaamisesta.

Kommunikointi projektin eri osa-alueiden välillä on tehotonta

Kolmen eri osa-alueen kommunikointi koettiin ongelmaksi. Kommunikoinnin eri osa-alueiden välillä kerrottiin kaatuvan projektin vetäjän vastuulle. Koska linkityksiä prosessimallissa ei ole, linkitys tapahtuu työntekijöiden kommunikoinnin kautta. Kommunikointi on itsenäistä ja perustuu ammattitaitoon siitä, mistä tietoa tarvitaan milloinkin. Prosessimalli ei siis oheista linkittämään projektin kolmea isoa osa-aluetta yhteen, ainaakaan tarpeen vaatimalla tavalla. Työntekijät ehdottivatkin ratkaisuksi katselmoiteja, joissa kaikki projektin osa-alueet olisivat mukana. Osa-alueiden yhteen linkittämistä

helpottaisi myös työntekijöiden tiimiyttäminen. Tiimissä työntekijät voivat valita tehtävänsä ja kommunikointi on päivittäistä.

Haastateltavat raportoivat myös, että palavereita pidetään liikaa, tai kaikki niiden osallistujat eivät hyödy palaverista. Myös työkaluihin liittyen kommunikointiin toivottiin monipuolisuutta ja etenkin sitä, että mikäli jokin työkalu on valittu käyttöön, kaikki perehtyisivät sen toimintaan ja käyttäisivät sitä asianmukaisesti. Jo aiemmin mainittu parikoodaus on yrityksessä todettu hyväksi tavaksi (H4,H5) ja se on myös yksi tapa jakaa tietoa työntekijöiden välillä.

Yhteistyö firman eri toimipisteiden välillä on vähäistä

Haastateltavat mainitsivat vähäiseksi kommunikaation firman eri toimipisteiden välillä. Tämä tarkoittaa myös uudelleenkäytettävien osien niukkaa jakamista firman eri osastojen kesken. On ketterän ajattelun mukaista, että tiimit voivat työskennellä ilman keskeytyksiä pyrhdyksen ajan (Schwaber & Sutherland 2013), ja siksi avun antaminen toisaalle voisi sekoittaa tiimien työtä pyrhdyksen aikana. Mikäli kuitenkin työntekijöille allokoidaan projekteihin aikaa myös firman sisäiseen kommunikointiin, voisi merkittävästi se nopeuttaa projektien läpimenoaikoja tulevaisuudessa. Eräs haastateltava ehdotti ongelman ratkaisuksi, että projekteihin lisättäisi henkilöitä, jotka eivät ole 100 % sidottuina projekteihin, vaan roolissa olisi enemmän vapautta tehdä asioita joihin toteuttajilla ei ole resursseja. Tämä olisi hyvä lisä moniosajatiimin sisälle

5.2.4 Projektin johtamiseen ja työn organisointiin liittyvät haasteet

Yrityksessä projektilla on projektipäällikkö, ja yksittäisiä työntekijöitä tekemässä projektia. Osa työntekijöistä tuntee kuuluvansa tiimiin, mutta varsinaisia määritettyjä tiimejä ei ole olemassa. Taulukossa 6 on lueteltu projektinjohtamiseen ja työn organisointiin liittyviä haasteita ja kehitysehdotuksia.

Taulukko 6. Työn organisointiin liittyvät haasteet ja kehitysehdotukset.

Haaste	Kehitysehdotus (K: Kirjallisuudesta, H: Haastattelusta)
Ihmisten resursointi johdon näkökulmasta on ongelma. (H4) Työntekijöiden osaaamista ei kohdenneta. Asioiden opettelu vie aikaa. (H4) (H5)	Kohdennetut tehtävät työntekijöille, ei laajoja kokonaisuuksia. (H4) Tiimiyttäminen, jolloin toistettavuus paranee. (K)
Tiimit pieniä tai niitä ei ole. Roolien vaihto hankalaa tai mahdotonta. Työntekijöiden valinnanvapaus minimaalista. Toistettavuus vähenee, opittavuus huononee. (H5)	Moniosajatiimien muodostaminen. (K)

Oikeanlaista dataa hintojen muodostumisesta ei ole kerätty. Hinnan arviointi hankalaa, koska toistettavuutta ei ole. (H2)	Linkitykset kehityksestä myyntiin. Luodaan suunnitelma datan keräämisestä. (K)
Projektipäällikön vastuulla, että kaikki projektin osa-alueet toimivat synkronoidusti. (H2)	Työntekijöille enemmän vastuuta omasta tekemisestä. Prosessimalliin selkeät stepit ja toimet synkronointiin. (K)
Projektien johtaminen ei ole ammattimaista. Iso osa projektipäälliköistä laitteistotaustaisia, joten ei osata katsoa projektia muista näkökulmista tasavertaisesti. (H4)	Tehdään katselmuksia sekä laitteiston suunnasta ohjelmiston suuntaan, että takaisin. (H4) Ammattimaisuutta johtamiseen. (K)

Työntekijöiden resursointi henkilöittäin koetaan ongelmaksi

Yrityksessä työntekijöitä ei ole jaettu tiimeihin, mutta osa työntekijöistä koki kuuluvansa pieneen tiimiin. Työntekijät kuvasivat monia projekteja "*yhden miehen projekteiksi*", jotka vedetään alusta loppuun itsenäisenä työnä. Näissä projekteissa monia asioita tehdään pintapuolisesti, niin että paketti saadaan kasaan ja työssä ei näin välttämättä ole juurikaan toistuvuutta. Jos tehty virhe unohdetaan kunnes joku muu tekee toisessa projektissa samat virheet, syvempää oppimista ei tapahdu. Projektien läpimenoaika saattaa olla vain muutama kuukausi, mikä ei kuulosta pitkältä ajalta, mutta tätäkin aikaa saattaisi pystyä nopeuttamaan jakamalla tehtäviä tiimin kesken ja osaamisen mukaan.

Perinteisissä menetelmissä tiimille/yksilöille annetaan ulkoa käsin työtehtäviä, ja jopa määritellään miten työ tulee tehdä. Tässä yrityksessä menetelmät työn tekemiseen ovat vapaat, mutta työ on ulkoa ohjautunutta. Työntekijä ei siis ole itsenäisesti sitoutunut työtehtävän tekemiseen, mistä voi johtua myös tehtävän suorittamisen vaikeus (uuden oppimisen määrä lisääntyy), ja projektin läpimenoajan pitkittyminen. Schwaber & Sutherland (2013) painottavat tiimin sitoutumista työhön ja tehtäviin. Tiimin jäsenille annetaan valta valita tehtäviä, jotka se pyrähdysten aikana sitoutuu tekemään. Tavoitteiden on oltava niin selkeitä, että tiimi voi sitoutua niihin. Selkeät tavoitteet myös mahdollistavat myöhemmässä vaiheessa tiimin onnistumisen arvioinnin. Selkeiden tavoitteiden asettamisen käytäntö ja tiimin sitouttaminen lisää tiimin kykyä arvioida tehtävien kokoa ja omaa osaamistaan jatkuvasti.

Työntekijöitä pitäisi siis antaa valtaa valita tehtävänsä ja osaamista tiimiyyttä. Eräs ohjelmistopuolella toimiva haastateltava ehdotti, että "*olisi hyvä, kun tiimit saisi toimimaan vähän Scrum-tyyppisesti*" (H4). Scrum-mallissa (Schwaber & Sutherland 2013) kehitystiimi koostuu itseohjautuvista monitaitoisista ammattilaisista, joiden tehtävänä on tuotteen kehittäminen lyhyissä pyrähdyksissä. Monitaitoisuutta ja isojen osa-alueiden välistä osaamista yrityksestä löytyykin. Ainoastaan tiimin käsite, ja miten se voisi toimia, on vielä haussa. Lisäksi pyrähdysten kesto ja niissä tapahtuvat toimet täytyisi so-

vittaa sopivaksi laitteistovaatimuksiin tai käyttää ketteryyttä ainoastaan ohjelmiston kehittämiseen. Ketterät menetelmät nojaavat lopulta toistettavuuden parantamiseen ja tehdyistä virheistä oppimiseen. Virheistä on helpompi oppia, mikäli palaute tulee nopeasti ja ympärillä on tiimi, jossa jaetaan tietoutta ongelmien osuessa kohdalle.

Tehdyn työn dokumentointi uudelleenkäytettäväksi on vähäistä

Uudelleenkäytettävyyden puute nähdään ongelmana. Koska firma myy resursseina ihmisiä projekteihin koko projektin ajalle, uutta opittavaa riittää. Myöskään valmiiksi dokumentoituja palasia ohjelmistosta ja laitteistosta ei ole olemassa, eli uudelleenkäyttämistä ei tehdä systemaattisesti. Eräs haastateltavista kuvasi myös, että syynä tähän on asenneongelma. Ajatellaan, että tällaista uudelleenkäytettävyyttä *"ei pystytä tekemään (salassapitovelvollisuudet, lakipykälät), ei uskalla, menee paljon rahaa hukkaan, ei onnistuta, iso juttu"* jne. Nähdään että *"pyörän keksiminen uudelleen ja uudelleen"* (H2) on pienempi paha, kuin ongelmaan tarttuminen, ja siihen resursointi. Myös salassapitovelvollisuus aiheuttaa sen, ettei tuotettua tietoa ja osatuotteita voida aina käyttää seuraavissa projekteissa. Ongelmaan tarttuminen vaatisi lisätutkimusta. Yritykseen tulisi löytää malli uudelleenkäytettävyyteen ohjaavan mallin löytämiseksi.

Historiadatan keruu ei ole systemaattista

Haastatteluissa mainittiin, että projektien hinta- ja aika-arviot ovat suunnittelijoiden vastuulla, eli kehitystyötä tekevien parhaita arvauksia projektin kestosta. Myyntiosasto ei siis kerää systemaattisesti informaatiota projektien keskimääräisistä kestoista ja niihin liittyvistä kustannuksista. Tieto ei kulje myynnin ja kehityksen välillä. Eräs haastateltavista kertoi, että eräänlainen palautekeskustelu on merkattu prosessimalliin, mutta palautekeskustelujen pitäminen on nimellistä (H2). Scrumissa (Schwaber & Sutherland 2013) jokaisen pyrähdysten lopussa pidetään katselmointitilaisuus, jossa tarkastetaan mitä pyrähdysten aikana on tehty ja miten tämä vaikuttaa seuraaviin pyrähdysiin. Tilaisuuteen osallistuvat Scrum-tiimin lisäksi sidosryhmien tärkeimmät edustajat. Palaverissa keskustellaan hyvin ja huonosti menneistä asioista sekä budjettiin ja markkinointiin liittyvät asioista. Riippuen mahdollisten pyrähdysten pituudesta, tällainen käytäntö olisi hyvä omaksua myös kohdeyritykseen. Tilaisuuteen voisi osallistua myynnin henkilöstön edustaja, joka keräisi dataa projektin keston ja hinnoitteluun liittyvistä asioista. Palaverissa voisi tehdä myös ennusteita tulevasta, jolloin arviointikykykin saattaisi vuosien mittaan parantua. Lisäksi tällainen datan keruu saattaisi lisätä edellisessä kohdassa käsiteltyä oppimista virheistä. Se, ettei virheistä opita, näytti olevan yrityksen yksi suurimmista ongelmista.

Projektipäälliköillä liian suuri vastuu osa-alueiden koordinoinnista

Projektipäälliköille kasaantuu yrityksessä iso vastuu projektin koordinoinnista. Projektipäällikön pitäisi hallinnoida asiat liittyen laitteiston ja ohjelmiston valmistukseen, suunnitteluun ja tuotantoon. Yhdenkin osa-alueen johtamisessa on iso vastuu. Projekti-

päälliköt ovat lisäksi usein laitteistotaustaisia ja ohjelmistosuunnittelun ymmärrys saattaa olla ymmärrettävistä syistä hyvin pintapuolista.

Näihinkin haasteisiin saattaisi auttaa työntekijöiden tiimiyttäminen ja mahdollisesti muut Scrumista tunnetut (Schwaber & Sutherland 2013) roolit. Tiimiyttäminen voisi lisätä ja tehostaa kommunikointia projektin eri osa-alueiden välillä. Tiimissä työskennellessä ollaan oman työpanoksen lisäksi vastuussa myös tiimin tuloksesta, ja tällä tavalla tiimi ottaisi enemmän vastuuta myös projektin eri osa-alueisiin liittyvistä linkityksistä. Tätähän tehtiin yrityksessä jo jatkuvalla kommunikoinnilla, mutta systemaattisia ohjeita ei tällaiseen linkitykseen ole olemassa.

Schwaber & Sutherland (2013) esittävät lisäksi projektipäällikön tilalle Scrum mestarin (engl. Scrum Master) ja tuoteomistajan (Engl. Product owner) rooleja. Vastuu erilaisista projektiin kuuluvista asioista jakautuisi näille kahdelle niin, että Scrum mestari toimii tiimin sisäisenä mahdollistajana ja valmentajana. Tämä tarkoittaa siis tuotekehitystiimin työn edistämistä kaikin mahdollisin keinoin. Tällaisesta roolista saattaisi tässä yrityksessä olla hyötyä myös uudelleenkäytettävyyteen ja muiden toimipaikkojen yhteydenpitoon liittyvissä ”ylimääräisten asioiden hoitamisessa”. Tuoteomistajan rooli sen sijaan on toimia tiimin ulkopuolisena rajapintana asiakkaaseen (ibid.). Tuoteomistaja olisi vastuussa myös tuotteen kehitysjonon ominaisuuksien selkeästä kirjallisesta ilmaisusta ja priorisoinnista. Näiden roolien avulla tuotekehitystiimillä olisi suurin vastuu tuotekehityksestä, eli myös linkityksistä huolehtimisesta.

6. TULOKSET: YRITYS 2

Tässä luvussa kuvataan toisen tutkimuksen tutkimusaineisto ja tulokset. Käytetty tutkimusmenetelmä on kuvattu luvussa 4.

6.1 Kuvaus

Tutkimuksen tilasi eräs suomalainen työkoneita valmistava suuryritys. Tutkimuksen tuloksena haluttiin saada vastaus kysymykseen: ”Miten sulautetun järjestelmän tuotantoprosessin läpimenoaika voitaisiin lyhentää?” Oma panokseni yritykselle tuotettavaan raporttiin oli prosessin pullonkaulojen tunnistaminen ja parannusehdotusten etsiminen. Koska käyttäjäkokemustyön sisällyttäminen diplomityöhän oli keskeistä, lisäsin tutkimuskysymykseksi myös käyttäjäkokemustyön osuuden prosessissa ja siihen liittyvät ongelmat.

Haastattelut yritykseen tehtiin vuonna 2014. Raportti yritykseen tutkimuksen tuloksista palautettiin syksyllä 2014.

Taulukossa 7 on esitelty haastateltavat työnkuvineen ja kokemusvuosineen. Poiketen muista haastatteluista, projektimanagerille ja pääsuunnittelijalle tehty haastattelu oli parihaastattelu.

Taulukko 7. Yritys 2, haastateltavien roolit yrityksessä ja työkokemus yrityksessä sekä alalla yhteensä.

Haastateltavan koodi	Työnimike / rooli yrityksessä	Työkokemus vuosina yrityksessä / ohjelmistoalalla
V2	Ohjelmistosuunnittelija	19/19
V3	Projektipäällikkö / Laitteisto-osaston esimies	1/15
V4	Laitteiston pääsuunnittelija	29/29
V5	Program manager / Esimies	19/31
V6	Testausinsinööri	ei tiedossa

6.2 Tutkimuksen tulokset

Seuraavissa kohdissa käydään läpi tutkimuksessa esiin tulleet haasteet tuotekehitysprosessissa. Löydökset on esitetty taulukoissa 8-12. Taulukkojen ensimmäisessä sarakkeessa on haasteen kuvaus ja sen esittäjä numerolla. Toinen sarake sisältää joko kirjallisuu-

den avulla etsityn tai haastatteluissa esiin tulleen kehitysehdotuksen. Kirjallisuus on merkitty taulukkoon merkinnällä (K), haastateltavalta tullut parannusehdotus haastateltavan tunnuksella (V-).

Löydetyt haasteet on jaettu viiteen haastekategoriaan: käyttäjäkokemustyö (6.2.1), projektinhallinta (6.2.2), esitutkimus- ja määrittelyvaihe (6.2.3), testaus ja alihankinta (6.2.4). Jokaiselle näistä kategorioista on varattu oma alakohta, joissa ensin esitetään taulukko haasteista ja kehitysehdotuksista ja tämän jälkeen käydään taulukon sisältöä ja lähdekirjallisuutta tarkemmin läpi tekstimuodossa.

6.2.1 Käyttäjäkokemustyöhön liittyvät haasteet

Yrityksessä tehdään käyttäjäkokemustyötä, mutta sen näkyvyys työntekijöille oli kyseenalaista. Projekteilla on tarvittaessa käytössä käyttäjäkokemustiimin resursseja. Haastateltavista lähes kaikki (4/5) kokivat kuitenkin tekevänsä ainakin joskus käyttäjäkokemustyötä, ja osasivat nimetä monia käyttäjäkokemukseen liittyviä teemoja. Taulukko 8 sisältää löytyneitä käyttäjäkokemustyöhön liittyviä haasteita ja kehitysehdotuksia.

Taulukko 8. Käyttäjäkokemustyöhön liittyvät haasteet ja kehitysehdotukset.

Haaste	Kehitysehdotus (K: Kirjallisuudesta, V: Haastattelusta)
Manuaalit ja käyttöohjeet saaneet huonoa palautetta kyselyissä käyttäjiltä (V4) ja testaajilta. (V6)	Käyttöohjeen tekemisen vastuu annetaan käyttäjäkokemustiimille. (K)
Testaukseen tulevaa tuotetta ei voi testata käytettävyysongelmien takia. (V6)	Testaaja yhteistyöhön käyttäjäkokemustiimin kanssa aikaisessa vaiheessa projektia. (K)
Määrittelyjä ei ole olemassa. (V2)	Muotti vähimmäismäärittelylle ja sen noudattaminen. Tarkistuslistat. (K)
Tuotteen käytettävyyttä testataan vasta projektin loppuvaiheessa, juuri ennen asiakkaalle menoa. (V2)	Tuotteen testaaminen jatkuvasti prosessin aikana, jotta virheet löytyisivät aiemmin. (K)
Laitteistopuolella käytettävyyttä ei pidetä kovin tärkeänä asiana. (V5)	Tietoisuuden lisääminen, ainakin tuotteen testattavuuteen, huoltoon ja asennukseen liittyvistä asioista. (K)

Käyttöohjeista tulee huonoa palautetta

Haastatteluissa mainittiin muutamaa otteeseen käyttöohjeisiin liittyvät ongelmat. Yrityksessä tuotekehitystiimin jäsenet tekevät käyttöohjeet. Kehittäjiltä ei vaadita erityisiä dokumentointitaitoja, eikä ajantasaisia dokumentointiohjeita ole. *"Onhan se aina vähän tekijöidensä näköinen."* (V3) Dokumentointiosasto ainoastaan viimeistelee ohjeet, muotoilee sivurakennetta jne. Pohjia käyttöohjedokumentin tekemiseen on, mutta ne ovat vanhentuneita. Saattaa olla, että käyttöohjeiden sisältöä kopioidaan tuotteesta toiseen.

Järjestelmätestaajat saattavat testata käyttöohjetta, mutta koska testaajat ymmärtävät tuotteen ilman ohjettakin, osa dokumentaation heikkouksista jää huomaamatta.

Tuotteen käytettävyydestä aloitetaan liian myöhään

Haastateltava kertoi, että tuotteen käytettävyydestä aloitetaan usein vasta kun tuote on lähes valmis ja menossa asiakkaalle pilottiin. Mikäli vikoja löydetään vasta tässä vaiheessa, virheiden korjaaminen saattaa olla hankalaa ja joissain tapauksissa jopa liian myöhäistä. Lisäksi virheistä korjataan usein vain kriittisimmät ja käyttäjäkokemukseen liittyviä asioita pidetään toissijaisina.

Testitapausten avulla on mahdollista myös määrittellä järjestelmää. Tämä saattaisi olla tapa tehdä määrittelytyötä sopivan kevyesti läpi prosessin. Testien suunnittelu pitäisi aloittaa mahdollisimman aikaisin (Vuori 2010) tai testaajien olisi ainakin hyvä olla mukana tuotetta suunniteltaessa, jotta tarvittavat testipenkit olisi mahdollista rakentaa valmiiksi. Testitapausten suunnittelu on hyvä tapa määrittellä ohjelmistoa ja koko tuotetta. Mikäli ohjelmistopuolella työtä on mahdollista tehdä ketterästi ja koko prosessia halutaan kehittää ketterään suuntaan, ohjelmiston, testitapausten ja käyttäjäkokemuksen tekeminen tulisi kulkea käsi kädessä. Etukäteissuunnittelua ei juuri tarvitse tehdä, vaan testaaminen tehdään tiimin sisällä tuotetta rakennettaessa.

Tuotteen testaaminen ei onnistu, koska käyttäjäkokemussuunnittelua ei ole tehty etukäteen

Tuote menee järjestelmätestaukseen ennen pilottiprojektiin toimitusta. Testauksen kestoaikaa ihmeteltiin monessa haastattelussa, mutta syitä tähän löytyi monia. Testaaja kuvasi tilannetta hyvin: *"Pääsääntöisesti mikään ei toimi ensimmäisellä kertaa."* Tuotteen käyttäjäkokemus saatetaan arvioida ensimmäistä kertaa testaajalle tullessa. Esimerkkinä tilanteesta testaaja kertoi purkaneensa 10 ruuvia saadakseen tuotteen auki, jonka jälkeen testaaminen oli mahdollista. Tuotteen testattavuus on yksi käyttäjäkokemuksen osaluokista, eikä tällaisessa tapauksessa testattavuutta ole selkeästi otettu huomioon. Toisaalta testaaja kertoo huonojen komponenttivalintojen vaikuttavan tuotteen testattavuuteen. Esimerkiksi ruuvien kireälle vääntäminen saattaa aiheuttaa jonkun muoviosan murtumisen. Tällaisten asioiden korjaaminen vie aikaa.

Määrittelydokumentit eivät ole kaikkien saatavilla

Yksi haastateltavista kertoi, ettei projektille yleensä löydy määrittelydokumentaatiota lainkaan. Mikäli määrittelydokumentaatio löytyy, se sisältää yleensä myös jotain käyttäjäkokemussuunnitteluun liittyvää asiaa. Usein tällaiset asiat kommunikoidaan kuitenkin vain sähköpostilla.

Ketterät menetelmät painottavat vuorovaikutusta dokumentoinnin sijaan. Tämä ei suinkaan tarkoita sitä, että minkäänlaisia määrittelyjä ei ole olemassa. Lisäksi yritys käyttää edelleen vesiputousmallia hyvin lähellä olevaa prosessikehystä. Koska tiimit yrityksessä

ovat todella erikoistuneita, olisi tärkeää, että halutut asiat kommunikoitaisiin selkeästi. Erityisesti käyttäjäkokemukseen liittyvät asiat olisi hyvä ainakin ohjelmistopuolella olla osittain toteuttajien käsissä heti projektin alusta alkaen. Mikäli ohjelmistopuolta haluttaisiin lisäksi kehittää ketterämmäksi, jatkuva yhteydenpito ja ominaisuuksien arviointi sekä työntekijöiden että käyttäjien kesken pitäisi tehdä mahdolliseksi.

Työntekijät kuitenkin mainitsivat, että määrittelyasiassa on tapahtunut kehitystä. Käyttöön on otettu uusi työkalu, jonka avulla esimerkiksi jäljitettävyyks on parantunut. Haastattelut antoivat hieman ristiriitaista tietoa haastateltavien tiedoista ja ymmärryksestä yrityksessä tehtävästä käyttäjäkokemustyöstä. Yksi haastateltavista kuitenkin totesi, että tietoisuus käyttäjän tarpeista on hyvä, eli tuotteiden tekeminen perustuu hyvään tietoon käyttäjistä.

6.2.2 Projektinhallintaan liittyvät haasteet

Projektinhallintaan ja siihen liittyviin aikatauluongelmiin liittyvät ongelmat ovat yksi yritysten yleisimmistä haasteista (Brook 1986). Tutkitussa yrityksessä pilottiprojektilla tarkoitetaan testauksen jälkeen alkavaa projektinvaihetta, jossa asiakas ottaa tuotteen testattavakseen. Taulukossa 9 on lueteltu projektinhallintaan liittyviä haasteita ja kehitysehdotuksia.

Taulukko 9. Projektinhallintaan liittyvät haasteet ja kehitysehdotukset.

Haaste	Kehitysehdotus (K: Kirjallisuudesta, V: Haastattelusta)
Pilottiprojektien hankkiminen haasteellista. Pilotteja ei ole tarpeeksi, tai niiden hankkimiseen ei panosteta. (V4)	Pilottien etsiminen ja organisointi etusijalle. Vastuhenkilö pilottien etsimiseen ja niistä tiedottamiseen koko projektille. (K)
Prosessimallin vaiheesta toiseen siirtymisen aiheuttaa odottelua. Erityisesti tuotteen tullessa kehityksestä testaukseen. (V4, V6)	Vastuhenkilöt siirtymien hallinnointiin. Jatkuva uudelleenmäärittely, erityisesti aikatauluasioissa. (K)
Pilotin jälkeen usein korjauskiirroksia, joita ei ole laskettu projektin kokonaiskeston. (V4)	Historiatiedon käyttö projektien suunnittelussa esimerkiksi tarkistuslistojen avulla. Listat mukaan suunnitteluun. (K)
Raportteja aiemmista projekteista on olemassa, mutta historiatietoa ei käytetä hyväksi uusissa projekteissa. Edellisten projektien ongelmat unohdetaan. (V5)	Tarkistuslistat edellisten projektien ongelmakohdista. Tarkistuslistat mukaan projektien suunnitteluun. Retrospektiivit. (K)
Projektit voisivat olla osittain ketterämpiä. (V3)	Ketterien käytäntöjen kokeilu. (K)

Pilottiprojekteja ei saada käynnistettyä ajallaan

Integrointitestauksen jälkeen tuote menee asiakkaalle testaukseen. Yksi haastateltavista (V4) kuvasi pilottiprojektien hankkimisen ja koordinoinnin vaikeutta näin: " *Ensin sopi-*

via pilotteja ei ollut saatavilla. Sitten aika kului, ja kun pilotti oli vihdoinkin saatu sovittua, ilmaantui tuotteeseen korjauskierroksia. Jotenkin puolen vuoden päivitystä ei julkistettu, vaikka piti. Sitten seuraava päivitys tuli, ja taaskaan pilotteja ei ollut saatavilla." Haastatteluja tehdessä (25.3.2014), projekti oli yhä kesken, vaikka sen alkamisesta oli aikaa monta vuotta. Mikäli pilottiprojektien saamisen vaikeus on yksi tärkeimmistä asioista, jotta tuote saataisiin asiakkaalle asti, ja toisaalta epäonnistuessaan johtaa projektin pitkittymiseen, tulisi tämä ongelma priorisoida. Piloteille tulee olla vastuuhenkilö, jolla on tieto pilottien keskimääräisestä hankinta-ajasta, tieto mahdollisimman aikaisin siitä, milloin tuote on täyttänyt testauksen menon hyväksymiskriteerit. Pilottivastuuhenkilöllä tulee lisäksi olla erittäin läheinen työskentelysuhde testauksen vastaavaan henkilöön ja toisaalta pilottiprojektin toiseen osapuoleen. Pilotin vastuuhenkilön tehtävä vaatii siis jatkuvaa kommunikointia tuotteen tilasta. "Pilotti hankittu" voisi olla yksi projektin etappi tai tehtävä, josta lähtee tieto heti kaikille projektissa työskenteleville ihmisille. Näin pilottivastuuhenkilöllä on mahdollisuus alkaa hoitaa tuotteen viivästyksestä aiheutuneita harmejä heti niiden ilmetessä.

Edellä siteeratusta kommentista kuultaa myös selkeiden hyväksymiskriteerien puute. On todennäköisempää, että projekti ei jää seisovaan tilaan, mikäli jokaiselle vaiheelle on nimetty vastuuhenkilö. Nämä vastuuhenkilöt olisivat vastuussa sekä tuotteen vastaanottamisesta uuteen vaiheeseen että tuotteen luovuttamisesta seuraavaan. Eri vaiheiden vastuuhenkilöiden on tärkeä pitää yllä jatkuvaa kommunikointia projektin ajan näistä siirtymistä ja siirtymien tapahtumiseen on oltava selkeitä kriteereitä.

Projektivaiheesta toiseen siirtyminen on haastavaa

Eri vaiheiden tahdistaminen toistensa kanssa on myös suunnitteluun liittyvä ongelma. Tämä on yleistä ohjelmistoprojekteissa. Wallsin mukaan (2014) yleensä ohjelmisto on valmiina ennen kuin laitteisto ja tämä aiheuttaa turhaa odottelua. Yksi haastateltavista järkeili, että tällaisissa tapauksissa on kyse kommunikaatio-ongelmista. Walls (2014) ehdottaakin ongelmaan ratkaisua, jossa ohjelmistoa testataan ja debugataan odotteluajalla. Lisäksi laitteiston emulointi tai virtualisointi on lisäarvoa tuottava tehtävä, joka vähentäisi odotteluaikaa. Näyttäisi siltä, että kyseessä on jälleen projektinhallintaan ja kommunikaatioon liittyviä ongelmia.

Edellisten projektien opit unohdetaan

Yhdessä haastattelussa (V3) mainittiin, että edellisten projektien oppeja ei oteta tarvittavalla tavalla käyttöön. Kokonaisdatan perusteella kyse näyttää olevan siitä, ettei nimenomaan projektien historiadatan keräämiseen ole suunnitelmaa ja jo kerättyä dataa ei osata käyttää hyväksi projekteja suunniteltaessa. Historiadata on merkittävä tiedonlähde etenkin kun kaikkien työntekijöiden kokemus on, että heille tulee työtä tehdessä keskeytyksiä ja oman työn hoitaminen viivästyy. Tarkemmalla suunnittelulla ja historiadatan käytöllä saatettaisiin huomata, että projektien aikataulut on tehty liian optimistisiksi työntekijöille "sattumalta" osuvaan lisätyömäärään nähden. Näiden

ongelmien selättämiseen Schwaber & Sutherland (2015) esittävät säännöllisen työmääräarvioiden tekemisen työntekijän toimesta. Työntekijä on paras oman työnsä keston arvioija. Toimintoihin kuluvan ajan arvionti paranee, kun sitä tekee säännöllisesti ja usein (ibid.). Lisäksi päivittäisten katsausten avulla voidaan huomata, mikäli aikaa menee muuhun, kuin varsinaisten työtehtävien tekemiseen. Schwaber & Sutherland (2015) esittävät lisäksi jokaiseen pyrähdykseen kuuluvat retrospektiivit, jotka kokoavat kaikki edellisen sprintin huonot ja hyvät käytännöt tiimien jäseniltä. Tätä tietoa tulisi käyttää tarkasti hyväksi seuraavien sprinttien suunnittelussa. Ilman Scrum-terminologiaa tämä tarkoittaa siis sitä, että kaikissa työtiimeissä tulisi olla vastuuhenkilö, joka on vastuussa historiadatan keräämisestä, säilyttämisestä ja sen arvioimisesta. Yksi projektin onnistumiskriteereistä voisi olla, että tällainen raportti on olemassa, kaikille nähtävillä ja että sen tuloksia voitaisiin käyttää seuraavan projektin suunnittelussa.

Prosessiin kaivataan ketteryyttä

Kaksi viidestä yrityksen työntekijästä kehui yrityksen omaa prosessimallia. Siinä on järkevät etapit, mutta ketteryyttä prosessimalliin kaivataan. Ohjelmistopuolella keuhuttiin beta-julkaisujen painotusta ja toivottiin, että prosessimallissa olisi puolen vuoden välein vähintään virstanpylväs, jossa pitäisi olla jotain konkreettista esillä. Ketterässä maailmassa puolen vuoden palautesykli kuulostaa todella pitkältä ajalta, mutta sulautettujen järjestelmien maailmassa puoli vuotta on useimmilla yrityksillä varsin kohtuullinen aika. Koska prototyyppejä voidaan rakentaa, niiden osuutta voitaisiin nostaa, tällöin projektivaiheesta toiseen siirtyminen ei toisi niin suurta painoa etenemiseen.

Osa työntekijöistä oli sitä mieltä, että prosessimallia voisi kehittää muuttamalla prosessimallissa olevien etappien 3 ja 4 väliä iteratiivisempaan suuntaan. "*Projekteihin voisi lisätä myös ketterän kehityksen muita toimintatapoja.*" Ohjelmistopuolella ketteryys on jo yrityksessä lisääntynyt, mutta laitteistopuoli menee hyvinkin vesiputousmallisesti. Työntekijät keuhivat yrityksen sisällä lämpimiä suhteita työntekijöiden välillä, työntekijöiden taitavuutta ja hyvää työskentely-ympäristöä (V5). Palaset ketteryyden suuntaan siirtymiseen ovat siis olemassa. Työntekijät luottavat toistensa tekemiseen ja arvostavat toistensa mielipiteitä. Yksilöt ja kanssakäyminen ovat yksi ketteryyden korostettavista arvoista (Ketterän ohjelmistokehityksen julistus 2001). Tämä onkin hyvä pohja ketterien menetelmien tuomiseen osaksi prosessia

6.2.3 Esitutkimus- ja määrittelyvaiheeseen liittyvät haasteet

Esitutkimus tuottaa listauksen asiakkaan tarpeista, mikä on ideaalisessa tapauksessa perusta määrittelylle (Järvinen & Mikkonen 2012). Koska esitutkimus ja määrittelyvaihe ovat hyvin toisistaan riippuvia vaiheita, haasteet näihin liittyen on koottu samaan taulukkoon. Ongelmat näissä vaiheissa aiheuttavat usein myöhästymisiä tuonnempana projektia, vaikei ongelmia loppuvaiheissa varsinaisesti esiintyisikään (Verner et al.

2008). Taulukossa 10 on lueteltu esitutkimus- ja määrittelyvaiheeseen liittyviä haasteita ja kehitysehdotuksia.

Taulukko 10. Esitutkimus- ja määrittelyvaiheeseen liittyvät haasteet ja kehitysehdotukset.

Haaste	Kehitysehdotus (K: Kirjallisuudesta, V: Haastattelusta)
Työmäärä arvioinnit ovat pienempiä, kuin tarvittun työn määrä. (V4)	Kaksi työmääräarviota jokaiselle projektille. Jatkuva aikataulun päivitys projektin aikana. Historiadatan käyttö projektin suunnittelussa. (K)
Määrittelyvaihe ei ole onnistunut. Asiakasvaatimuslista puutteellinen tai siellä on ominaisuuksia, joille ei varmistettua tarvetta. Asiakasta ei kuunnella. (V2, V3, V5)	Jatkuva vaatimusten uudelleenarviointi. Vaatimusten tarkentaminen projektin edetessä. Jatkuva kommunikaatio asiakkaan kanssa. (K)
Määrittelyt tehdään sähköpostilla tai niitä ei ole lainkaan. (V2)	Vastuuhenkilö määrittelyn tekoon, uudelleenarviointiin ja varmistamaan, että jaettu kaikille asianomaisille. (K)
Testaajat eivät mukana projektisuunnittelussa. Työmääräarviot tehdään niiden henkilöiden toimesta, jotka eivät ole oikeasti vastuussa työn tekemisestä. (V6)	Testaajat mukaan projektin suunnitteluun heti alkuvaiheessa. Työmääräarvioiden tekoon ne työntekijät, jotka työn suorittavat. (K)

Työmääräarviot ovat pienempiä, kuin tarvittun työn määrä

Työmääräarviointit ovat Brooks (1987) mukaan ohjelmistoprojekteissa säännöllisesti pienempiä, kuin varsinaisen työn määrä. Ongelman taustalla on ohjelmien kompleksisuus, mikä saattaa aiheuttaa lisätyötä (ibid.). Usein myös arvioita tehtäessä, palaset ovat liian suuria arvioitavaksi. Kanbanin kolmas sääntö perustuu siihen, että mahdollisimman pieniksi palasiksi pilkotut tehtävät ovat helpommin ennustettavissa, jolloin kokonaisprosessin suunnittelu helpottuu. Lisäksi yksittäisten työtehtävien läpimenoaikojen optimointi pienentää tehokkaasti ”pullonkaulojen” syntymisen riskiä. (Kniberg & Skarin 2010). Myös Glass (2002) kävi läpi useita epäonnistuneita projekteja ja yksi suurimmaksi osoittautuneista epäonnistumiseen johtaneista tekijöistä oli juurikin huono työmäärien ja tehtävien koon arviointikyky. Glass (2002) ehdottaakin, että arviot ovat enemmänkin kauniita toiveita siitä miten projekti voisi parhaassa tapauksessa mennä, kuin että ne perustuisivat oikeaan dataan ja realistiseen arviointiin.

Hyvä arviointi alkaa kaikkien odotusten dokumentoinnista, tuotteen ja ympäristön tilasta. Heti kun nämä muuttuvat, pitää arvioinnit tehdä uusiksi (Cubas 2010, PMBOK Guide 2000). Ohjelmistoprojekteissa suunnitelmat muuttuvat hyvinkin usein: asioita ei voidaan tehdä niin kuin on suunniteltu. Suunnitteluprosessin tulisi olla jatkuva aktiviteetti, joka laajenee koko projektin ajan (ibid.). Hyvä estimaatti tuottaa tarkkoja tuloksia, jos projektin olosuhteet pysyvät samoina koko projektin ajan. Mikäli yrityksessä halutaan

pitää kiinni hyväksi koetusta vesiputousmallista, tällainen jatkuva projektisuunnitelman ja projektityömääräarvioinnin päivittäminen olisi hyvä ottaa mukaan.

Konkreettinen parannus työmääräarviointien tekoon voisi olla tarkistuslistat. Cubas (2010, PMBOK Guide 2000) ehdottaa tarkistuslistaa niille tekijöille, jotka voivat vaikuttaa projektin lopputulokseen. *"Tarkistuslista pakottaa ajattelemaan jokaisen yksittäisen tekijän vaikutusta ja opettaa oppimaan aiemmista kokemuksista"* (ibid.). Tällaista tarkistuslistaa voisi käyttää mukana projektin suunnittelupalaverissa muistuttamassa tekijöistä, jotka vaikuttavat projektin läpimenoaikaan ja myös muihin asioihin, jotka ovat hidastaneet aiempien projektien valmistumista.

Yksi arviointiin liittyvä ongelma oli myös se, ettei testaushenkilöstö osallistunut projektin suunnittelupalaveriin. Haastatteluissa kävi ilmi, että testausvaihe on yksi projektin eniten aikaa vievistä vaiheista. Mikäli tämä on huomioitu projektisuunnittelussa, se ei liene ongelma (testauksen optimointi ei myöskään kuulu tämän tutkimuksen fokukseen). Mikäli testauksen aika-arvion tekevät ihmiset, jotka eivät suorita testausta, ei ole ihme, että ajat saattavat kuulostaa pitkiltä. Scrum-mallissa (Schwaber & Sutherland 2013) vastuu aika-arviointien tekemisestä annetaan niille, jotka suorittavat työn. Mitä useammin estimaatteja annetaan, sitä parempia arvioijista tulee.

Määrittelyvaihe koetaan puutteelliseksi

Määrittelyvaiheeseen liittyvät puutteet oli myös mainittuina ongelmiksi usean haastatellijan toimesta. Ebertin ja Jonesin (2009) mukaan ongelmat liittyen huonoon määrittelyyn eivät usein tule esiin ennen kuin testausvaiheessa. Ebert & Jones (2009) lisäävät, että kaikki voi näyttää projektin alkuvaiheessa hyvältä, mutta saavuttaessa testausvaiheeseen projekti alkaa näyttää ongelmalliselta. Syy on harvoin testauksessa (ibid.). Pitkemminkin huonosta määrittelystä johtuvat seuraukset näkyvät vasta myöhemmässä vaiheessa projektia (ibid.). Pahimmassa tapauksessa tuote joudutaan tekemään uudelleen, ja työmäärä vähintäänkin kaksinkertaistuu. Ebert & Jones (2009) lisäävät julkaisussaan, että *"40% kaikista ohjelmisto-ongelmista sulautetuissa järjestelmissä johtuvat epäselvistä vaatimuksista ja analyysitoimista"*. Haastatteluissa ei käynyt ilmi miten vaatimuksia kerätään, arkistoidaan ja katselmoidaan.

Noelin (2013) mukaan yritykset, joiden *"vaatimukset ovat mitattavia ja testitapaukset kehitetty ja ylläpidetty kaikille vaatimuksille, ovat markkinoilla niitä jotka johtavat. Ja yritykset, joiden vaatimukset ovat huonosti dokumentoituja ja niistä puuttuu testattavuus ja jäljitettävyyys, ovat häviäjiä"*. Se että vaatimukset ovat mittavia ei kuitenkaan tarkoita sitä, että vaatimusmäärittely olisi tehty projektin alussa, vaan määrittelyä voi kertyä paljon tehtynä pieninä palasina projektin matkan varrella. Yksi haastateltavista mainitsikin hyväksi kehitykseksi työkalun, jolla jäljitettävyyys on parantunut. Tähän on siis jo kiinnitetty huomiota, mikä on hyvä asia.

Prosessissa tehdään arvoa tuottamatonta työtä

Yksi haastateltavista mainitsi ongelmaksi ylituotannon, mitä voidaan pitää todella suurena haasteena. Womackin ja Jonesin (2003) mukaan ylituotanto on yksi Lean-ideologian "jätteistä". Jätteellä tarkoitetaan sellaisen tuottamista, mitä asiakas ei tarvitse, halua tai käytä. Mikäli läpimenoaika on venynyt, tuotantoketjussa tehdään jotain, mikä ei tuota asiakkaalle arvoa. Tuotteessa on jotain mitä ei tarvita, ja siihen on kulunut työ-aikaa ja rahaa. Tämänkaltaisen jätteen poistaminen ketjusta alkaa siitä, että asiakasvaatimukset katselmoidaan asiakkaan kanssa, ja kun vaatimukset muuttuvat projektin aikana, sitä tehdään uudestaan ja uudestaan. Tämä tarkoittaa samalla kommunikation lisäämistä asiakkaan kanssa. Niin kauan kun työntekijöitä palkitaan ylituotannosta, se jatkuu.

6.2.4 Testaukseen liittyvät haasteet

Yksi kaikkein eniten keskusteluun nousseista ongelmista haastatteluissa oli testausvaihe. Ongelmia liittyen testaukseen nousi niin määrittelyvaiheesta kuin itse testausvaiheesta. Testausvaiheen pituus määriteltiin ongelmaksi, mutta mikäli vaiheen pituus on kaikille tiedossa, sen ei "sinänsä" pitäisi olla ongelma. Ongelma on vaiheen pitkittymisen arvioiden yli, mikä johtuu yleensä aiemmissa vaiheissa tapahtuneista ongelmista (Ebert & Jones 2009). Taulukossa 11 on lueteltu testaukseen liittyviä haasteita ja kehitysehdotuksia.

Taulukko 11. Testaukseen liittyviä haasteita ja kehitysehdotuksia.

Haaste	Kehitysehdotus (K: Kirjallisuudesta, V: Haastattelusta)
Testauksen sirpaloituminen. Muita töitä silloin, kun pitäisi rakentaa testiympäristöä. (V6)	Vastuuhenkilö tarviketarvaston ylläpitoon. Testaajat mukaan projektinsuunnitteluun. (K)
Asiantuntemuksen puute. Saattaa johtua henkilöstön vaihtuvuudesta. NykYTEKNOLOGIAA EI TUNNETA, jolloin testaaminen vie aikaa. (V6)	Koulutus, työnjaon suunnittelu, hiljaisen tiedon tallennus työvoiman vaihtuessa. (K)
Prioriteetit testauksessa vaihtuvat kesken kaiken. (V6)	Työnjohtaja määrittää prioriteetit ja toimitaan sen mukaan. Prioriteettien uudelleenarviointia muutosten mukaan. (K)
Tuote, joka tulee testaukseen ei ole valmis testattavaksi. (V6)	Testaushenkilö mukaan suunnittelemaan projektia alusta asti. Vastuuhenkilö testauksesta ja implementoinnista määrittelemään tuotteen testausvalmius. (K)

Testausvaihe kestää liian kauan

Tuoteprosessin testausvaihe kestää kauan. Testausvaihe saattaa onnistuessaankin olla aikaa vievä. Testausvaiheen kestoa lisää se, ettei testaushenkilöstö tiedä milloin tuote

tulee testaukseen. Tämä taas aiheuttaa sen, ettei testipenkki ole valmiina. Saattaa käydä myös niin, ettei tuote olekaan testaukseen tullessaan valmis testattavaksi tai ettei suunniteltuja testejä voidakaan suorittaa, koska tuote on erilainen kuin on määritelty, ja suunniteltuja testejä ei voida ajaa. On siis erityisen tärkeää, että ainakin testauksen vastuhenkilö saadaan mukaan suunnittelemaan projektia ja sen aikataulutusta. On yleisesti tiedettyä (Uusitalo et al. 2008) että testaushenkilökunnan tulisi olla mukana vaatimus- katselmuksissa. Testaajien mukanaolo projektin alkuvaiheessa varmistaa sen, että toteutettavat ominaisuudet voidaan myöhemmässä vaiheessa testata ja validoida (ibid.). Mikäli määrittelyä muutetaan projektin edetessä, myös testaustiimin pitää olla jatkuvasti tietoinen muutoksista. Tällä varmistetaan, että testilaitteisto on rakennettu sopivaksi testaukseen tulevalle tuotteelle. Lisäksi jo aiemmin mainittu määrittelyn jatkuva katsel- mointi on hyvä tapaa pitää kaikki projektin jäsenet tietoisina sen hetkisen määrittelyn sisällöstä ja mahdollisista muutoksista ja viivästyksistä. Se, että koko projektihenkilöstö on tietoinen projektin nykytilasta ja seuraavista askeleista prosessissa vaikuttaa merkittävästi projektin läpimenoaikaan.

Edellä mainittuun liittyen yksi haastatelluista (V6) kertoi, että testauksen alkaessa osia ja työkaluja testipenkin rakentamiseen ei välttämättä ole. Mikäli työntekijöillä ei ole tarkkaa tietoa siitä, milloin tuote tulee testaukseen, tarvittavia osia ei ehkä ehditä hankkia, ja tuote seisoo niin kauan kunnes testilaitteisto on valmis. Testaustyöntekijöiden on myös tärkeää tietää tarkalleen mitä ollaan testaamassa (etenkin laitteistoa testatessa) ja mitä voidaan ylipäätään testata (Ebert & Jones 2009). Tämä on yksi lisäsyys siihen, että on tärkeää että testaajat ovat mukana tuotesuunnittelussa. Tarvittavien osien puute voi aiheuttaa paljon hukka-aikaa ja isot kustannukset, kun osia ja testauslaitteita yritetään rakentaa itse. Näin voi käydä, mikäli testaukseen tulee tuote, jonka testaamiseen osia ei ole, tai niiden tilaamiseen menisi kauan aikaa. Tämä olisi voitu välttää hyvällä tuotesuunnittelulla ja testaajien ammattitaitoa hyödyntäen. Monessa haastattelussa ongelmaksi mainittu testausvaihe pitenee siis entisestään, kun testausta ei voida aloittaa, vaikka laitteisto tai ohjelmisto olisi jo valmista testattavaksi.

Työntekijöitä on vähän, työtä paljon

Kuten muissakin yrityksen toiminnoissa, myös testauksessa on puutetta työntekijöistä. Työntekijät ovat vähentyneet ja jääneiden henkilöiden taidot eivät välttämättä sovi heille annettuun positioon. Tämä hidastaa testausta mikäli teknisesti osaavimmat henkilöt tekevät töitä, jotka vievät aikaa heidän ammattitaitoaan vaativilta tehtäviltä. Scrum-käsikirja (Schwaber & Sutherland 2013) opastaa siihen, että ammattitaitoisten työntekijöiden tulee saada tehdä rauhassa työtä, joka on heille sprinttipalaverissa määritelty. Mikäli tämän työn lisäksi on muita töitä, ne pitää olla merkittynä aikatauluun, ettei työkuorma kasva ja työteho alene. Tällaisesta työn sirpaloitumisesta puhuttiin myös muiden roolien kohdalla. Työaika menee muiden opettamiseen ja omat työt jäävät tekemättä. Tällaisessa tilanteessa ei ole ihme, että projektin kalenteriaika kasvaa, ellei tuota ylimääräistä työtä ole määritelty projektin alkaessa.

6.2.5 Alihankintaan liittyvät haasteet

Haastattelussa tuli ilmi monia alihankintaan liittyviä haasteita. Moni näistä haasteista liittyi suoraan projektinhallintaan, mutta koska alihankintaprosessin haasteellisuutta erityisesti korostettiin haastattelussa, on ne koottu omaan taulukkoonsa. Taulukossa 12 on lueteltu alihankintaan liittyviä haasteita.

Taulukko 12. Alihankintaan liittyviä haasteita ja kehitysehdotuksia.

Haaste	Kehitysehdotus (K: Kirjallisuudesta, V: Haastattelusta)
Vastuut epäselviä, kun toimitaan kahdessa maassa. Molemmissa maissa omat managerit. (V3)	Vastuiden selvittäminen. Yksi manageri kahden tilalle. (K)
Prototyypin valmistus vie aikaa, kun tilataan ulkomailta. (V3)	Prototyyppien tilaaminen paikallisilta valmistajilta. (V3)
Komponenttien puute alihankkijalla. Komponentit ovat saatavilla varmuudella vain jos tilaus on varmistettu. Jos tilausta ei ole varmistettu, komponentit ovat muiden käytettävissä, joten projekti voi myöhästyä. (V4)	On jo kokeiltu, että toinen alihankkija tilaa komponentit ja toimittaa ne. Tämä on toiminut. (V4) Silti vastuut asioiden loppuunsaattamisesta tulisi antaa jollekin nimetyille henkilölle. (K)
Alihankkijoilla erilaiset prosessimallit. Tämä saattaa aiheuttaa ongelmia. Datan liikkuminen sovitaan erikseen jokaisessa projektissa. (V3)	Selkeät vaatimukset tilatulle työlle. Jatkuva kommunikointi, aikataulu ja hyväksymiskriteerit työlle. (K)
Alihankkijan mielestä on normaalia, että sama työ tehdään kahteen kertaan. (V2, V6)	Jatkuvaa kommunikointia alihankkijoiden ja työntekijöiden välille. Hyväksymiskriteerien selvennys palavereissa. (K)
Alihankkijan lähdekoodit eivät ole toisen osapuolen nähtävissä. (V2)	Avoimuuteen pyrkiminen. Sopimusrikkereiden seuraukset tulisi tehdä selväksi kaikille osapuolille. (K)
Työn pirstaloituminen. Alihankkijan työntekijöiden neuvominen vie aikaa itselle suunnitellusta työstä. (V6)	Allokoitua aikaa ylimääräisille tehtäville. (K)
Kaikkea tarvittavaa ei saada alihankkijalta. Intian henkilöstöllä erilaiset taidot kuin suomalaisilla. (V3)	Määritellään tarkasti mitä on saatavilla ja mitä ei. Käytetään paikallisia toimijoita. (V3)
Resurssien uudelleenjärjestelyä ei tehty, kun Intian alihankkija julkistettiin. Muutokset organisaatorakenteessa ovat hämärtäneet työntekijöiden vastualueet. (V5)(V6)	Muutokset organisaatio- ja työntekijärakenteessa johtavat aina töiden uudelleenjärjestelyyn. Vastuut on oltava selvillä. (K)

Kulttuurierot aiheuttavat ongelmia kommunikoinnissa

Haastattelujen perusteella kommunikointi on yrityksen haasteista suurin. Kommunikaatioon liittyvät ongelmat ja tarvittavan tiedon puute ovat yleisesti (Charvat 2012) tyypillisin projektin epäonnistumiseen liittyvä tekijä. Tässä yrityksessä ongelmia kommunikointiin aiheuttavat erityisesti kulttuurierot. Kieltä ei ymmärretä riittävästi, jostain syystä asioita tehdään eri tavalla kuin on sovittu ja asioita joita tarvittaisiin toisella puolella maapalloa, ei saada käytettäväksi. Kommunikaatio-ongelmat vaikuttavat suoraan työlaatuun ja aiheuttavat myöhemmin projektissa pieniä myöhästymisiä, koska työtä pitää tehdä uudelleen. Jotta näiden pienistä myöhästymisistä projektin loppuun heijastuva suuri myöhästymisen saataisiin poistettua, kommunikaation on parannettava.

Yksi haastatelluista mainitsi, että vaikka ajatellaan että kommunikointi on ollut riittävän hyvää ja alihankkijalla pitäisi olla kaikki tilauksen yksityiskohdat tiedossa, jokin menee silti pieleen ja aikatauluja ei saada pidettyä.

Kommunikaatioon liittyvien ongelmien ratkaisu voisi alkaa siitä, että päivittäin tarkastetaan, onko tiedot ymmärretty juuri niin kuin on sanottu. Scrum-käsikirjassa (Schwaber & Sutherland 2013) otetaan kantaa tällaisiin ongelmiin päivittäisillä Scrum-palavereilla. Jos kommunikointia tarkastellaan päivittäin, isoja ongelmia ja informaatiokatkoja ei tulisi syntyä lainkaan. Kommunikointi ja yhteiseen tavoitteeseen pyrkiminen voi parantaa ensin päivittäistä kommunikointia ja myöhemmin monitoroinnilla osatuloksia ja etappeja. Mikäli ensimmäistä etappia ei saavuteta ajoissa, tarvittavat toimenpiteet projektin saamiseksi oikeille raiteille pitää ottaa heti käyttöön. Managerin tehtävä on aikatauluttaa projektille selkeät etapit ja varmistaa, että jokainen etappi tuottaa sille tarkoin määrätyt asiat.

Coplienin ja Harrisonin mukaan (2005) kehittäjien palkitseminen aikataulun onnistuneesta suunnittelusta voisi olla yksi keino tarttua ongelmaan. He ehdottavat myös, että projektissa pitäisi olla kaksi aikataulua. Yksi markkinoinnille ja yksi kehittäjille. Ulkoisen aikataulun kesto pitäisi sopia pidemmäksi, kuin sisäisen ja jos näitä aikatauluja ei saada soviteltua, aikataulu tulee uudelleen sopia. *"Jos aikatauluista tekee liian anteliaita, kehittäjistä tulee laiskoja, ja jos aikataulu on liian haastava, kehittäjät palavat loppuun."* (ibid.). Aikataulujen teon voidaan sanoa siis olevan taidetta.

Organisaatorakenne on muuttunut, mutta uusia vastuualueita ei ole selvitetty

Useassa haastattelussa tuli esiin alihankintaprosessin aloitus ja siitä aiheutuneet haitat niin työntekijöille kuin prosesseille. Haastatelluista sai käsityksen, että vaikka muutos oli yritykselle suuri, ei suuria muutoksia organisaatorakenteessa tai tiimeissä tehty juurikaan. Mikäli yrityksessä tehdään suuria muutoksia tai työtapoja muutetaan merkittävästi, tarvitaan pätevää muutosjohtamista (Luomala 2008). Jotkut resurssi- ja asiantuntijajamenetykset voidaan kompensoida hyvällä projektin johtamisella, työn allokoinnilla ja kommunikoinnilla. Kuitenkin, kun yritykseen luodaan uusia alueita, kommunikoinnin

määrä moninkertaistuu ja enemmän kommunikointia tarkoittaa usein myös projektien pitkittymistä. Siirtymäaika tällaiselle on hyvä huomioida projektien kestoja miettiessä muutoksen ollessa päällä.

Muutokset organisaatorakenteessa heijastuvat projekteihin, uuden haaran rooli organisaatiossa näytti epämääräiseltä. Mikäli kommunikointi on ongelma jo ennen muutosta, työntekijöillä ei ole mahdollisuutta uusissa rooleissaan tehokkaaseen kommunikointiin. Mikäli jopa vastuuhenkilöt ovat epävarmoja rooliensa työtehtävistä, on itsestään selvää, että isoja aikatauluongelmia syntyy. Tilanteen parantamiseksi työntekijöiden rooleille tulee luoda selkeät vastuut ja tehdä selväksi mitkä ovat kriteerit sille, miten vastuut voidaan täyttää.

Erilaiset prosessimallit alihankkijan kanssa on hankala sovittaa yhteen

Koska alihankkijalla on lisäksi erilainen prosessimalli kuin yrityksellä, se aiheuttaa ongelmia, koska esimerkiksi kommunikointiin tarkoitetut työkalut saattavat olla hyvinkin erilaisia ja ne määritellään jokaiselle projektille erikseen. Mikäli lisäksi vaatimukset alihankitulle työlle ovat selkeitä ja ymmärrettäviä, mikäli aikataulut on sovittu yhdessä ja niihin päästään ja mikäli hyväksymiskriteerit sille, että työ on tehty, ovat samanlaiset, ei tarvitse keskustella siitä, miten asiat eri yrityksissä tehdään. Tämän ei siis pitäisi olla ongelma, mikäli kaikki taustalla olevat, edellä jo moneen kertaan mainitut asiat ovat kunnossa.

7. TUTKIMUSTULOSTEN KOONTI

Tässä luvussa verrataan löydettyjä ongelmia alan kirjallisuudessa esitettyihin ongelmiin ja tehdään yhteenveto kahdessa tapaustutkimuksessa löytyneistä haasteista ja kehitysehdotuksista. Lisäksi luvussa esitetään prosessimalli ongelmien selättämiseksi ja esitetään suositukset parhaista ketteristä käytännöistä kahden tutkimuksessa mukana olleen yrityksen toimintaan.

7.1 Tulosten vertailu kirjallisuudessa esiintyneisiin ongelmiin

Kohdassa 2.3 käytiin läpi sulautettujen järjestelmien erityispiirteitä ja kohdassa 2.4 kehitysmenetelmien ongelmakehoita. Taulukossa 13 on esitetty sekä sulautettujen järjestelmien erityispiirteet, että niihin liittyvien ongelmien esiintyvyys yrityksessä.

Taulukko 13. Sulautettujen järjestelmien tuotantoon haasteita tuovat erityispiirteet.

Haasteita tuotantoon tuovat erityispiirteet	Yritys 1	Yritys 2
Ohjelmistokehitys joutuu odottamaan fyysisen laitteiston valmistumista.	x	x
Sulautetun järjestelmän tuottaminen vaatii aina useiden erikoisalojen yhteistyötä.	x	x
Tuotetaan useita eri tyyppisiä järjestelmiä.	x	
Kehitystyö tehdään eri ympäristössä kuin missä ohjelmistotullaan testaamaan.	x	x
Kehitettävillä järjestelmillä on tiukat turvallisuusvaatimukset.	x	x
Kehitettävillä järjestelmillä on tarkat vikasietoisuusvaatimukset.	x	x

Ensimmäinen kohdassa 2.3 läpikäydyistä ongelmista oli Grenningin (2002) esittämä: Laitteiston kehittäminen siihen vaiheeseen, että ohjelmisto voidaan lisätä siihen vie kauemmin aikaa kuin ohjelmiston kehittäminen. Molemmissa yrityksissä odoteltiin prototyypin saapumista, jotta sitä päästäisiin testaamaan. Lisäksi laitteiston saapumiseen ei oltu varauduttu tarpeen vaatimalla tavalla.

Liggemeyer & Trapp (2009) esittivät, että sulautettu ohjelmisto on aina osa suurempaa kokonaisuutta ja se aiheuttaa ongelmia myös tuotantoprosessiin. Molemmissa yrityksissä projektin monien eri erikoisosa-alueiden yhdistäminen ja linkittäminen koettiin hankalaksi.

Myös sulautettujen järjestelmien monimuotoisuus oli yksi ongelmia prosessimalliin aiheuttavista tekijöistä (Liggesmeyer & Trapp 2009). Ensimmäisessä yrityksessä tämä koettiin ongelmaksi, sillä toimivaa prosessimallia on vaikea muodostaa, koska projektit ovat keskenään niin erilaisia. On vaikea hallita kokonaisuutta, joka muuttaa muotoaan täysin projektista toiseen. Toisessa yrityksessä sulautettuja järjestelmiä tehtiin omaan käyttöön, eli malli ja toimiala pysyivät projektista toiseen samanlaisena. Tätä ei tutkimuksessa kuvailtu suureksi ongelmaksi.

Järvisen ja Mikkosen mukaan (2012) se, että kehitystyö tehdään eri ympäristössä kuin missä ohjelmisto tullaan testaamaan aiheuttaa usein ongelmia sulautettujen järjestelmien kehityksessä. Tämä ongelma voitiin havaita molemmissa yrityksissä. Ensimmäisessä yrityksessä ongelmia raportoitiin asentamiseen liittyen. Asennus paikkaan, jossa tuotetta alettiin käyttää, ei onnistunutkaan. Tätä ei oltu testattu oikeassa ympäristössä. Toisessa yrityksessä testaaminen oli ongelmallista, vaikka yrityksessä oli oma testilaboratorio tuotteiden testaamiseen.

Lisäksi sekä turvallisuuteen että vikasietoisuuteen liittyvät erityispiirteet koettiin prosessia hankaloittavina, sillä molemmissa yrityksissä prosessin määrittelyvaihe saattoi venähtää viranomaisvaatimusten takia. Lisäksi suunnitelmapohjaisuus myös määrittelyvaiheessa tuntui olevan ongelma prosessin ketteröittämiselle.

7.2 Yhteenveto haasteista ja ratkaisuehdotuksista

Taulukkoon 14 on koottu tutkittujen yritysten suurimmat ongelmakohdat. Ongelmat on valittu suurimmiksi niiden esiintymistaajuuden perusteella sekä arvioimalla haastateltujen henkilöiden asioita kohtaan ilmaisemaa tärkeyttä. Haasteen perään on merkitty raportoitujen ongelmien lukumäärä / kaikki haastateltavat. Taulukon jälkeen kuvaillaan haasteluokkien sisältö ja ratkaisuehdotukset.

Taulukko 14. Suurimmat haasteet yrityksessä 1 ja yrityksessä 2.

Haasteluokka	Yritys 1	Yritys 2
Käyttäjäkokemustyö	Loppukäyttäjiltä ei saada palautetta. (4/6)	Käyttöohjeista tulee huonoa palautetta. (2/5)
Tiedonkulku ja kommunikointi	Tuote tehdään asiakkaalle, ei loppukäyttäjälle. Tieto loppukäyttäjistä tulee välikäsien kautta ja muuttuu matkalla. (3/6)	Kommunikaatio-ongelmat alihankkijan kanssa aiheuttavat odottelua. (4/5)
Prosessimalli	Laitteisto ja ohjelmisto tehdään omissa prosesseissaan, niissä ei ole yhtymäkohtia. (2/6)	Kehitysvaiheesta testaukseen siirtyminen aiheuttaa odottelua, testausvaihe kestää. (2/5)
Kertaalleen tehdyn	Osatuotteiden uudelleenkäytet-	Edellisten projektien on-

työn hyödyntäminen	tävyys vähäistä. (2/6)	gelmat toistuvat uusissa projekteissa. (3/5)
Määrittelyvaihe	Määrittelyvaihe monella tapaa ongelmallinen: vaatimukset muuttuvat eikä asiakas tiedä mitä haluaa. (3/6)	Epäonnistunut määrittelyvaihe aiheuttaa ongelmia projektin muihin vaiheisiin. (3/5)
Projektin johtaminen	Ihmisten resursointi tehdään johdon näkökulmasta, ei osaamisen perusteella. Oppimista ei tapahdu. (2/6)	Työntekijöiden resursointi johdon toimesta. Isot muutokset yrityksessä eivät ole aiheuttaneet uudelleenresursointia. (2/5)

Käyttäjäkokeumustyö

Ensimmäisessä yrityksessä 4/6 haastateltavaa koki ongelmaksi sen, ettei loppukäyttäjiltä saada palautetta. Palautetta ei saada prosessin aikana, eikä sen jälkeen, kun tuote on markkinoilla, eli loppukäyttäjien käytössä. Ihanteellisessa tapauksessa loppukäyttäjä sitoutettaisiin tuotekehitysprosessiin tuotteen suunnitteluvaiheesta lähtien. Loppukäyttäjä olisi mukana testaamassa ja arvioimassa tuotetta aina kun jotain toimivaa saataisiin ulos. Tällä voitaisiin välttää kalliita korjauskierroksia tuoteprosessin loppuvaiheessa.

Sulautettua ohjelmistoa kehitettäessä asentajat ja testaajat on otettava käyttäjäryhmänä huomioon aina. Myös suorituskykyyn ja näin suoraan tuotteen käytettävyyteen liittyvät kysymykset laitteistopuolella on mietittävä lähes kaikissa sulautetuissa projekteissa. Tuotteen asentamiseen kuluva aika vaikuttaa suoraan projektin kustannuksiin ja on siis sekä merkittävä kustannus- että käytettävyystekijä. Asennettavuuden lisäksi tärkeää on huomioida tuotteen testattavuus. Tuote tulee suunnitella niin, että se on helppo testata. Tämä voidaan varmistaa testaajan läsnäololla aikaisissa vaiheissa projektia. Käyttäjäkokeumustyön tekemisen merkitys aikaisessa vaiheessa kehitysprosessia korostuu, kun huomataan että myöhemmin tehty työ tai työn tekemättömyys saattaa aiheuttaa myöhästymisiä niin testauksessa, kuin myöhemmässä vaiheessa asiakkaalle asennettaessa. Tästä syystä loppukäyttäjä palautetta tulisi saada jo projektin alussa, ja läpi projektin. Näin projektin loppuvaiheessa suuria muutuskustannuksia aiheuttavat ongelmat voidaan välttää.

Toisessa yrityksessä käyttäjäkokeumukseen liittyvistä ongelmista suurimmaksi (2/5) koettiin huono palaute käyttöohjeista. Käyttöohjeiden teko jää usein kehittäjien vastuulle ja dokumentointiin erikoistunut henkilö saattaa korjailta käyttöohjeesta kirjoitusvirheitä. Käyttöohjeet on usein tarkoitettu loppukäyttäjien lisäksi asentajille ja testaajille. Tuotekehitysprosessissa toimivat työntekijät osaavat ehkä kokemuksensa puolesta lukea ohjetta ilman suurempia ongelmia, mutta loppukäyttäjän tapauksessa tilanne voi olla toinen. Käyttäjän turvallisuuden kannalta on olennaista, että käyttöohje on luettava ja sel-

keä, ja sen tekeminen kuuluisi käyttäjäkokemustyöntekijöille tai henkilöille joilla on kokemusta käyttöohjeen tekemisestä.

Tiedonkulku ja kommunikointi

Tiedonkulku ja kommunikointi osoittautuivat yrityksissä monella tapaa ongelmalliseksi. Ensimmäisessä tapauksessa raportoitiin ongelmia esimerkiksi siinä, että tuote tehdään asiakkaalle, ei loppukäyttäjälle (3/6). Tieto loppukäyttäjistä saattaa muuttua välikäsien kautta toiseksi, kuin mitä se on alun perin ollut. Lopulta tuotetta ei tehdä loppukäyttäjälle sopivaksi, vaan suunnittelu perustuu arvauksiin. Tähän ratkaisuksi voi perustella samaa kuin edellä. Loppukäyttäjä tulee sitouttaa tuotekehitysprosessiin projektin alusta alkaen. Tällöin suunnitteluratkaisut eivät perustu arvauksiin, vaan tietoon käyttäjän tarpeista ja toiveista.

Toisessa yrityksessä kommunikaatio-ongelmat alihankkijan kanssa aiheuttavat odotte-lua. (4/5) Laitteisto tehdään yrityksessä ulkomailla ja kulttuurierot Suomeen verrattuna ovat isot. Alihankinnan työntekijöitä on hankala saada ymmärtämään mitä heidän halutaan valmistavan. Lisäksi kaikki työ vie enemmän aikaa kuin Suomessa: koetaan, että on hyväksyttävää tehdä asiat pariin kertaan uusiksi. Lisäksi alihankinnan työntekijöitä oheistavat palaverit koetaan turhina: palavereista ei tunnu olevan hyötyä ja lisäksi oma työ myöhästyy, mikäli aika menee muiden opettamiseen. Ratkaisuehdotus ongelmaan on moniosajista koostuvien etätiimien muodostaminen ja päivittäin tapahtuvat video-palaverit, kunnes työtavat saadaan sovitettua molemmille yrityksille sopiviksi.

Prosessimalli

Kummassakin yrityksissä ongelmalliseksi koettiin kehitysvaiheesta testausvaiheeseen siirtyminen. Yrityksessä 2 kerrottiin, ettei testaushenkilöstö ole aina valmiina, kun laitteisto tulee testaukseen. Tähän syinä olivat mm. se, ettei testaushenkilöstö tiedä laitteiston saapumisesta ja testipenkkejä ei ollut ehditty rakentaa. Aina tarvittavia osia testipenkin rakentamiseen ei ollut käsillä, ja ne piti hankkia, vaikka prototyyppi oli jo valmis testaukseen. Näiden ongelmien lisäksi saattoi ilmetä, ettei rakennettu testipenkki sopinutkaan laitteistolle, tai tuotteen käyttäjäkokemusta testaajan kannalta ei oltu mietitty, ja laitetta oli mahdoton testata. Yrityksessä 1 kuvailtiin samantapaista ongelmaa, mutta enemmänkin niin, että projektin eri osa-alueiden (laitteisto, ohjelmisto, mekaniikka ja käyttäjäkokemus) välillä oli epäselvyyttä mikä on prioriteettijärjestys toimille prototyypin saapuessa, eli kenen kannattaa ottaa tuote testattavaksi ensin, keiden kannattaa ensin tehdä yhteistyötä jne. Molemmissa yrityksissä olisi tärkeää, että testaushenkilöstö olisi mukana määrittelyvaiheessa. Tällä voitaisiin ensinnäkin varmistaa se, että tuotetta on mahdollista testata. Toisekseen testaushenkilö osaisi valmistautua tulevaan testaukseen ajoissa, koska olisi alusta asti tiedossa mitä tullaan testaamaan. Koska määrittelyt muuttuvat jatkuvasti, olisi testaushenkilöstön tärkeä pysyä mukana myös tuotekehitysprosessin ajan, ja päivittää tietojaan vaikkapa viikkopalavereissa, tällä pystyttäisiin varmistamaan testauksen valmius silloin kun prototyyppi on valmis ja saapuu testaukseen. Li-

säksi kaikkien projektin osapuolien olisi hyvä olla tietoisia siitä, missä kehityksessä mennään, milloin prototyyppi lähtee valmistettavaksi ja milloin se saapuu. Prototyypin valmistusaikana olisi hyvä sopia tarkasti siitä, mitä tapahtuu prototyypin saapuessa.

Aiemmin tehdyn työn uudelleenhyödyntäminen

Ensimmäisessä yrityksessä 2/6 haastateltavista mainitsi ongelmaksi kertaalleen tehdyn työn vähäisen uudelleenkäytettävyyden. Tällä tarkoitettiin sitä, että moni samanlainen suunnittelutyö tehdään projektista toiseen uudelleen ja uudelleen. Suunnittelumoduuleita ei siis dokumentoida uudelleenkäytettävään muotoon. Oppimista ei tapahdu, koska työtehtävät vaihtuvat projektista toiseen ja jokaisessa projektissa työntekijät opettelevat aina paljon uusia asioita. Ratkaisuksi ongelmaan voisi esittää ensinnäkin tiimien muodostamista. Tällöin työtehtävät olisivat työntekijöille kerrasta toiseen tutumpia. Uuden oppimisen määrä projektissa olisi näin pienempi ja aikaa jäisi itselle jo tuttujen suunnitteluosuuksien dokumentointiin. Dokumentoidut osuudet voisi vielä yrityksen yhteiseen tietokantaan, josta niitä voisi tarvittaessa etsiä. Uudelleenkäytettävyyden lisääminen on kuitenkin iso prosessi, joka tarvitsisi oman selkeän suunnitelman.

Toisessa yrityksessä haastateltavat (3/5) raportoivat samojen ongelmien toistuvan projektista toiseen. Historiadataa ei kerätä esimerkiksi retrospektiiveillä. Tällaisen voisi aloittaa yrityksessä, eli viikoittain tai kuukausittain kerättäisiin tietoa hyvin ja huonosti menneistä asioista. Tätä historiadataa käytettäisiin sitten hyväksi seuraavaa projektia suunniteltaessa. Pienillä muutoksilla ja jatkuvalla parantamisella (Womack & Jones 2003) voidaan saada isoja muutoksia ja oppimista aikaan pidemmän ajan kuluessa.

Määrittelyvaihe

Molemmissa yrityksissä raportoitiin ongelmia tuotteen määrittelyyn liittyen. Yritykset käyttivät suunnittelulähtöistä prosessimallia, jossa ennen tuotannon aloittamista on selkeä erillinen määrittelyvaihe. Tämä on hyvin tyypillistä sulautettujen järjestelmien tuotannossa (Järvinen & Mikkonen 2012). Projektin alussa tapahtuva massiivinen määrittely aiheuttaa kuitenkin ongelmia myöhemmässä vaiheessa projektia, kun suunnitelmat muuttuvat ja tuotetta on tehty pitkälle määrittelyn mukaan.

Näissä yrityksissä määrittelyvaiheessa tehdään päätökset siitä, millainen tuotteesta tulee. Määrittelyvaiheessa saman pöydän ääreen pitäisi saada mukaan testaustyöntekijä, asiakas, käyttäjäkokemussuunnittelija ja tuotearkkitehti. Tieto loppukäyttäjistä on äärimmäisen tärkeää tuotesuunnitteluprosessin kannalta, koska loppukäyttäjä on usein se, joka tekee päätöksen siitä käyttääkö tuotetta vai ei. Tämä ei kuitenkaan aina päde esimerkiksi työkoneita tuottaessa, mutta se on silti hyvä huomioida kaikissa tuotantoprosesseissa.

Projektin johtaminen

Molemmissa yrityksissä koettiin ongelmalliseksi se (4/11), että työntekijät resursoidaan projekteihin johdon toimesta. Tämän koettiin hidastavan oppimista. Lisäksi toisessa yrityksessä koettiin, että vaikka yrityksessä on tapahtunut suuria muutoksia, silti työntekijät ovat pysyneet samoina. Työntekijät eivät siis itse pysty vaikuttamaan toimenkuvaansa juurikaan. Sen sijaan he saavat kyllä päättää miten tekevät työnsä.

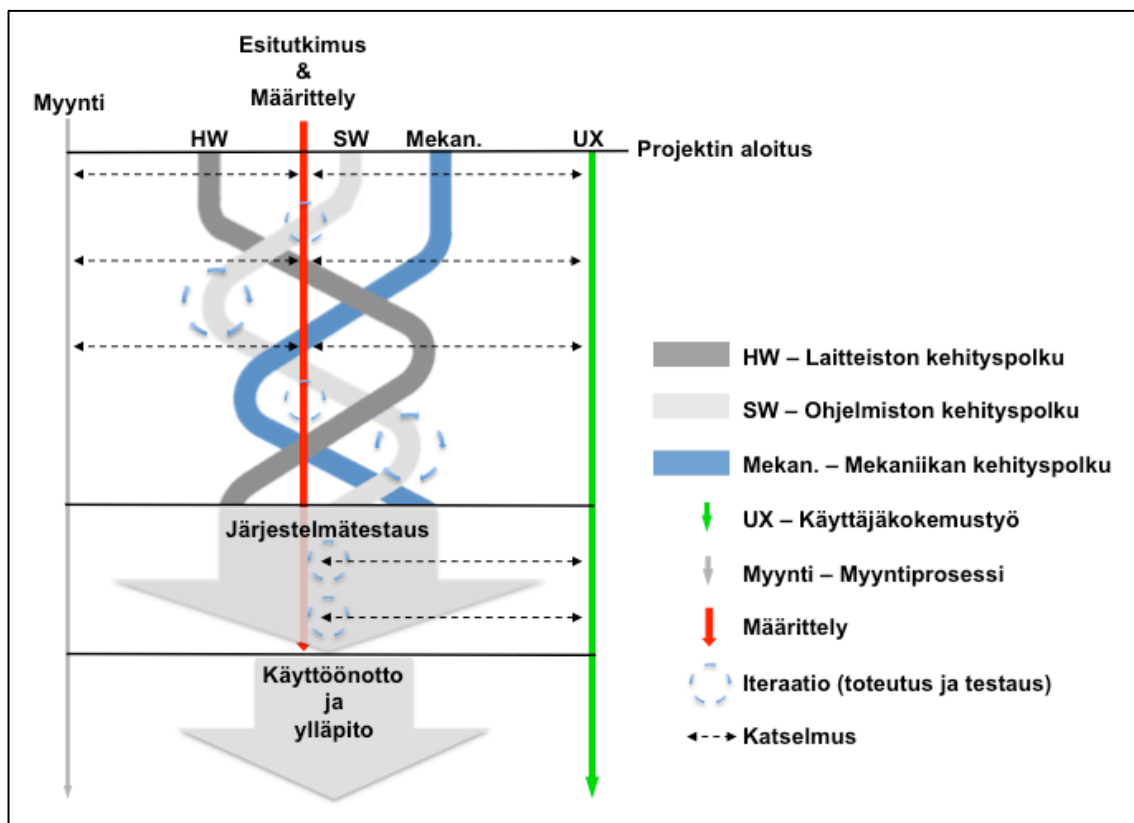
Ratkaisuksi ongelmaan voitaisiin koettaa itseorganisoituvia tiimejä, joissa jokainen työntekijä saa tiimin sisällä valita omat tehtävänsä. Vaikka ensimmäisessä yrityksessä yritysideoana on henkilöresurssien myyminen asiakkaalle projektiin, tiimityöskentely voisi tuoda yrityksen toimintaan tehokkuutta. Tiimissä on enemmän osaamista, kuin yksittäisellä työntekijällä, ja tämä voisi nopeuttaa projekteja. Nopeammat projektit voisivat tuoda myös lisää asiakkaita. Itseorganisoituvassa tiimissä työntekijät ovat motivoituneempia, kun saavat tehdä itse valitsemissaan työtehtäviä, jotka itse parhaiten osaavat.

Erottava tekijä yritysten välillä

Suurin ero yritysten välillä oli se, että ensimmäisessä yrityksessä iso osa yrityksen projekteista valmistui ajoissa, ja mikäli pieniä myöhästymisiä tapahtui, niin projektit eivät kuitenkaan koskaan jääneet keskeneräisinä seisomaan: *”Kyl me aina päästään maaliin. Sitä kautta tulee onnistumisia kyllä, et saadaan teknisesti onnistumaan projektit ja nipin ja napin aikataulussa, niin sekin on hyvä, ja totta kai me tehdään kuitenkin ihan hyvää tulosta.”* (H2) Toisessa yrityksessä tutkimuksesta haettiin apua nimenomaan projektien läpimenoajan lyhentämiseen. Ongelma oli merkittävä; monet projekteista pitkittyivät, joskus projektit jopa pysähtyivät eikä niitä enää koskaan jatkettu.

7.3 Ehdotettu malli hyvistä käytännöistä

Ketterät hybridimallit ovat todellisuutta monissa yrityksissä, joissa käytetään ketteriä menetelmiä (West 2011). Hybridimallilla tarkoitetaan mallia, jossa on sekä ketteryyttä, että suunnitelmaohjautuvan kehityksen piirteitä. Kuvassa 8 (seuraava sivu) on esitetty hybridimalli, joka on koostettu hyvistä ohjelmisto- ja laitteistokehitykseen liittyvistä käytännöistä.



Kuva 8: Ehdotettu prosessimalli sulautetun ohjelmiston tuotantoon.

Malli tähtää ketteryyden lisäämiseen sulautettua ohjelmistoa kehittävässä yrityksissä. Seuraavissa kohdissa on kuvattu tarkemmin, mitä mallin eri osat pitävät sisällään. Malli olettaa, että yritykset pitävät muilta osin toimintansa ennallaan, mikä ei tietenkään ole täysin mahdollista, sillä muutokset saattavat aiheuttaa ongelmia toisaalle.

7.3.1 Esitutkimus ja määrittelyvaihe

Esitutkimus- ja määrittelyvaihe tehdään ketterästi. Ketterässä vaatimusmäärittelyssä yleisesti käytettyinä vaatimusten dokumentointitapoina ovat käyttäjätarinat, testitapaukset, tehtävät ja työlista (Ruuska 2012). Myös ”systeemitapausten” käsitettä on ehdotettu sulautettujen järjestelmien ketterässä määrittelyssä (Pwc 2013). Systeemitapaukset ovat käyttötapausten kaltaisia, mutta tehdään systeemin näkökulmasta. Dokumentaation tulee olla koko projektin osalta saatavilla kaikille ja jatkuvasti ajantasaista. Zhang (2010, Ruuskan 2012 mukaan) esittää, että vaatimusten dokumentaatio on jatkuva prosessi, jossa vaatimusten yksityiskohdat tarkennetaan juuri ajoissa (engl. just-in-time), ennen kuin niiden toteuttaminen alkaa. Etenkin ketterässä vaatimusmäärittelyssä vaatimuksia on vaikea tietää ja dokumentoida etukäteen, koska vaatimukset muuttuvat usein sidosryhmien ymmärryksen lisääntyessä ja kohdealueen muutoksien myötä.

Esitutkimus- ja määrittelyvaiheen palaverissa mukana tulee asiakkaan lisäksi olla ohjelmisto-/laitteistoarkkitehti, käyttäjäkokenemussuunnittelija, testauksesta vastaava henki-

lö ja myynnin työntekijä. Määrittelyvaiheessa asiakas pyritään saamaan saman pöydän ympärille riittävän pitkäksi aikaa (esim. viikoksi). Asiakkaalta voidaan pyytää jo projektin alkuvaiheessa kontakteja loppukäyttäjiin. Mikäli asiakkaalta ei saa kontakteja, voidaan ne hankkia myös itse. Näin varmistetaan, että käyttäjäkokemussuunnittelija pääsee tekemään haastatteluja loppukäyttäjille määrittelyä varten heti projektin alkuvaiheessa. Parhaassa tapauksessa loppukäyttäjä saadaan mukaan tuotesuunnittelupalaveriin.

Käyttäjäkokemussuunnittelija tuottaa persoonat mahdollisimman aikaisessa vaiheessa projektia koko tuotekehitysväelle. Persoonien avulla varmistetaan, että kaikki tuotekehittäjät tietävät kenelle tuotetta ollaan tekemässä. Käyttäjäkokemussuunnittelijalla on lisäksi aktiivinen rooli kommunikoida käyttäjien projektin edessä muuttuvia tarpeita kehittäjille ja testaajille.

Esitutkimus- ja määrittelyvaiheessa aletaan etsiä tuotteelle pilottiprojektia, mikäli tällainen on tarpeen. Pilottiprojektin on oltava valmiina heti testausvaiheen jälkeen, jottei tähän vaiheesta toiseen siirtymiseen synny turhaa odottelua. Pilottiprojektien hankinnalle voisikin laittaa tuotteen työlistaan tehtävän, jonka hyväksymiskriteerinä ennen testausvaiheen loppua on ”pilotti hankittu.”

7.3.2 Toteutusvaihe

Toteutusvaiheessa ohjelmisto-, laitteisto-, mekaniikka-, testaus-, ja käyttäjäkokemustyöntekijät tekevät tiukkaa yhteistyötä. Mahdollisuuksien mukaan eri osa-alueiden työntekijöistä muodostetaan tiimi, joiden kommunikointi on päivittäistä. Työtehtävien visualisointi seinälle esimerkiksi Kanban-taulun avulla, niin että yhdellä silmäyksellä voi nähdä projektin tilannekatsauksen, voisi olla hyvä lisä projekteihin. Lisäksi määrittelyä päivitetään läpi projektin. Koska laitteisto ei yleensä ole jaettavissa kokonaisuuksiin samalla tavalla kuin ohjelmisto, laitteistokehitys jatkaa työtään suunnitelmaohjautuvasti. Ohjelmistopuoli voi tehdä työtään ketterästi, mutta jatkuvassa vuorovaikutuksessa (tiimissä) laitteistokehityksen, mekaniikan, testauksen ja käyttäjäkokemussuunnittelun kanssa. Se, että yksi tai kaksi osapuolta työskentelee suunnittelupohjaisesti, ei estä työskentelyä tiimissä. Tiimien olisi lisäksi hyvä olla itseorganisoituvia.

Vaatimusten priorisointi tapahtuu projektipäällikön toimesta. Projektipäällikkö toimii läpi projektin linkkinä asiakkaaseen. Tiimit koostuvat moniosajista ja ovat itseorganisoituvia, eli projektipäällikön valta vähenee ja tiimi ottaa suurimman vastuun eri osa-alueiden linkityksistä.

Laitteistopuolen tiukat aikavaatimukset otetaan projektin aikataulun pohjaksi, ja aikatauluvaatimusten perusteella luodaan koko projektin yhteinen aikataulu. Aikatauluun merkitään katselmointeja (engl. check point) laitteistopuolelta ohjelmiston suuntaan ja takaisin. Samat katselmoinnit tehdään käyttäjäkokemustyön ja mekaniikan suuntaan.

Katselmoinneissa arvioidaan projektia ajallisesti niin taakse kuin eteenpäin. Taaksepäin mietitään mitä on tehty, mikä meni hyvin ja osuivatko aiemmin tehdyt arviot kohdalleen. Eteenpäin arvioidaan tulevia aikatauluja ja mitä on vielä tekemättä. Katselmoinnissa sovitaan seuraavan katselmoinnin ajankohta, ja hyväksymiskriteerit katselmoinnin läpäisemiseen.

Katselmoinnissa olisi hyvä olla mukana työntekijöitä kaikista projektin osa-alueista. Erityisen tärkeää olisi tuotekehityspuolen ja testauksen työntekijöiden lisäksi myyntipuolen henkilön läsnäolo. Katselmoinnissa kerättäisiin historiadataa projektiin kuluneista tunneista, tehtäisiin aika-arvioita tulevalle ja verrattaisiin arvioita toteutuneisiin arviointeihin. Näin saataisiin ajan myötä kerättyä tietoa projektin hinnanmuodostuksesta, ja hinta-arvioiden teko voitaisiin luovuttaa tuotekehityksen parhaiden arvioiden sijaan myynnille.

Viimeisessä katselmoinnissa ennen testausvaiheeseen siirtymistä sovitaan tarkasti toimet prototyypin saapumisen jälkeiselle ajalle. Tässä katselmoinnissa on myös erityisen tärkeää päivittää määrittely vastaamaan tuotteen nykytilaan, jotta testauksella on mahdollisuus valmistaa testit olemassa olevalle laitteelle sopivaksi. Lisäksi määritetään selkeä järjestys, jossa tuote annetaan testattavaksi projektin eri osa-alueille. Tämä järjestys on täysin riippuvainen tuotteesta.

7.3.3 Järjestelmätestausvaihe

Järjestelmätestausvaihe suoritetaan jälleen ketterästi. Prototyypikierroksia voi laitteistoprojekteissa olla yhdestä kahteen. Tämä on laskettu projektin keston jo projektin alkuvaiheessa. Järjestelmätestausvaiheessa tärkeintä on, että kaikilla tuotetta testaavilla on selkeä suunnitelma testattaville ominaisuuksille ja niiden prioriteeteille. Tuoteomistaja tai projektipäällikkö määrittelee prioriteetin ja tarpeellisuuden virheiden korjaamiseen.

Mikäli tuote menee pilotointiin asiakkaalle, tulee tuotteella olla pilottivastuuhenkilö, joka on jatkuvasti tietoinen siitä, milloin testaus loppuu, ja tuote voidaan luovuttaa asiakkaalle. Pilotin vastuuhenkilön tehtävä vaatii siis jatkuvaa kommunikointia tuotteen tilasta sekä testauksen että asiakkaan pilottivastaavan kanssa. "Tuote valmis pilottiin" voisi olla yksi projektin tehtävä Kanban taulussa tai tuotteen työlistassa. Tämän tehtävän valmistumisesta lähtee tieto kaikille projektin osapuolille. Näin pilottivastuuhenkilöllä on mahdollisuus alkaa hoitaa tuotteen mahdollisesta viivästyksestä aiheutuneita harmeja heti niiden ilmetessä.

7.3.4 Käyttöönotto- ja ylläpitovaihe

Käyttöönottovaihe voi olla joko sitä, että valmis tuote luovutetaan asiakkaan käyttöön, tai tuotteella aloitetaan pilotointiprojekti asiakasyrityksessä. Heti käyttöönottovaiheen

alussa huolehditaan siitä, että palaute käyttäjiltä siirtyy kehittäjille. Ylläpitovastuussa oleville kehittäjille varataan aikaa tuotteen ylläpitotehtäviin. Ajan varaaminen tarkoittaa joko Kanban-seinään lapun laittamista tai vaihtoehtoisesti kehitysjonoon tehtävän lisäämistä.

7.4 Parhaat ketterät käytännöt

Taulukkoon 15 on kerätty molemmille yritykselle sopivia hyviä menetelmiä toiminnan muuttamiseksi ketterämmäksi.

Taulukko 15. Parhaat ketterät käytännöt yritysten toiminnan tehostamiseksi.

Käyttäjäkokenemustyö	<ul style="list-style-type: none"> • Persoonat kaikille kehittäjille käyttäjistä • Testaushenkilöstön rekrytointi firman sisältä tuotteen jatkuvaan käyttäjäkokenemusraviointiin
Tiedonkulku ja kommunikointi	<ul style="list-style-type: none"> • Monitaitoiset tiimit samassa tilassa • Päiväpalaverit • Projektin kulun visualisointi Kanban-seinän avulla
Prosessi	<ul style="list-style-type: none"> • Jatkuva parantaminen • Testaus mukaan projektin alusta lähtien • Linkitykset osa-alueiden välille katselmoineilla
Jo tehdyn työn hyödyntäminen /virheistä oppiminen	<ul style="list-style-type: none"> • Monitaitoiset itseohjautuvat tiimit • Historiadatan keruu katselmoineissa
Määrittelyvaihe	<ul style="list-style-type: none"> • Jatkuva kommunikointi asiakkaan kanssa • Loppukäyttäjät ja testaus mukaan määrittelyyn • Määrittelydokumentit samaan paikkaan, systeemitarinat
Projektin johtaminen	<ul style="list-style-type: none"> • Lisävastuuta aikataulusta tiimeille • Jatkuva arviointi, historiadatan keruu katselmoineissa • Vaatimusten selkeä priorisointi

Parhaat ketterät käytännöt on valittu niin, että niistä olisi hyötyä yrityksille, jotka joutuvat toiminnassaan ottamaan huomioon sulautettujen järjestelmien tuottamiseen liittyvät erityispiirteet ja rajoitteet. Käytännöt on pyritty selittämään mahdollisimman konkreettisesti, jotta niiden käyttämisen aloittaminen yrityksessä olisi mahdollisimman helppoa.

8. JOHTOPÄÄTÖKSET

Tässä luvussa pohditaan tutkimuksen tuloksia tutkimuskysymyksiin vastaamisen kautta. Lisäksi arvioidaan tutkimuksen luotettavuutta ja tutkimusetiikkaa. Luvun lopussa on vielä yhteenveto.

8.1 Pohdinta

Seuraavassa tarkastellaan tutkimuksen tuloksia tutkimuskysymyksiin vastaamalla. Tutkimuskysymykset on esitetty myös luvussa 1.2 (s. 2)

1. Millaisia ongelmia yritykset kohtaavat sulautettujen järjestelmien tuotannossa?

Yritykset raportoivat ongelmia erityisesti käyttäjäkokemukseen, tiedonkulkuun, prosessimalliin, jo tehdyn työn uudelleenkäytettävyyteen ja projektinjohtamiseen liittyen. Sulautettujen järjestelmien luonteesta johtuen projektissa on mukana monia erikoisosa-alueita, joiden keskinäinen kommunikointi on haasteellista. Lisänsä kommunikointiin tuovat asiakas ja toisaalta loppukäyttäjäpalautteen puuttuminen. Haasteeksi koettiin erityisesti kommunikointi laitteistoa valmistavaan alihankkijaan. Tiivis etätyöskentely koettiin hankalaksi ja kuormittavaksi työkuulttuurin erilaisuuden vuoksi.

Myös siirtymä kehitysvaiheesta testausvaiheeseen koettiin hankalaksi. Ongelma ilmeni esimerkiksi niin, että prototyypin saapumiseen valmistuksesta ei oltu valmistauduttu ja toimia ei oltu tarkasti määriteltä, jotta työ olisi sujuvaa testauksen alkaessa. Toisaalta saattoi olla niin, että testaushenkilöstö ei ollut selvillä testattavista ominaisuuksista ja päivitetystä määrittelystä eikä pystynyt valmistautumaan testaukseen. Tämä aiheutti testausvaiheen pitkittymisen.

Testausvaiheeseen liittyvät ongelmat johtuvat usein aiemmin projektissa tehdyistä virheistä. Ongelmia raportoitiin myös määrittelyvaiheeseen liittyen. Asiakas ei tiedä mitä haluaa, yhteistä näkemystä tuotteesta ei saada sovittua ja projektin alussa tehdyt vaatimukset muuttuvat jatkuvasti. Nämä ongelmat ovat yleisiä ohjelmistokehityksen ongelmia ja osittain selätettävissä ketterillä menetelmillä. Laitteistokehityksessä ratkaisu ei kuitenkaan ole niin helppo: suunnitelmaohjautuvasta prosessimallista on hankala, ehkä myös tarpeetonta yrittää päästä eroon.

Tehdun työn uudelleenkäytettävyyden puute koettiin myös ongelmaksi. Kertaalleen tehtyjä ratkaisuja ei osattu käyttää hyväksi seuraavissa projekteissa. Toisaalta tätä estivät myös tuotteiden immateriaalioikeudet, jotka kuuluvat asiakkaalle. Ongelmaksi nos-

tettiin myös se, ettei virheistä opita. Tämä on tiukasti kytköksissä uudelleenkäytettävyyden puutteeseen.

Lisäksi haasteita koettiin projektijohtamisen tasolla. Työntekijöiden resursointi saatettiin tehdä johdon tasolta, mikä johti mahdollisesti yhden henkilön projekteihin. Uuden oppimista yhdelle henkilölle yhdessä projektissa on liikaa ja projektit kestävät kauemmin, verrattuna tilanteeseen, jossa jokainen tekisi parhaiten osaamistaan vastaavaa työtä tiimi ympärillään. Haasteeksi raportoitiin lisäksi, että projektipäälliköltä vaaditaan melkoista osaamista, kun isot projektiosa-alueet pitäisi pitää käsissä yhtä aikaa. Projekti-päälliköllä on todennäköisesti taustaa vain yhdestä osa-alueesta, kun projekteissa suuria osa-alueita on yleensä kolme. Ammattimaisuutta johtamiseen kaivattiin.

2. Millaisia ongelmia käyttäjäkokemustyön tekemiseen liittyy sulautettujen järjestelmien ohjelmistotuotannossa?

Suurin ongelma liittyen käyttäjäkokemustyön tekemiseen sulautetussa ohjelmistotuotannossa ovat vanhakantaiset asenteet käyttäjäkokemustyötä kohtaan sekä asiakkaan, että tuottajan johtajatasolla. Käyttäjäkokemustyön tarve sulautettujen järjestelmien piirissä tiedostetaan, mutta sitä ei silti pidetä tarpeeksi tärkeänä asiana. Ongelma liittyy myös asiakkaisiin, joilla asenne on sama: haluamme tuotteeseemme helppokäyttöisyyttä, mutta emme ole valmiita maksamaan siitä. Käyttäjäkokemus ei sulautetun ohjelmistotuotannon maailmassa edelleenkään myy, vaan tuotteita suunniteltaessa nopea aikataulu vaikuttaisi edelleen olevan selkeästi tärkein tuottajan valintakriteeri. Oikein tehtynä käyttäjäkokemustyö voi kuitenkin jopa nopeuttaa aikataulua. Testausaika lyhenee, mikäli tuote on suunniteltu testaajan tarpeiden mukaan. Samoin tuotteen asennusaika.

Toinen suuri ongelma on työntekijöiden oma tietotaso käyttäjistä ja käyttäjäkokemusasioista. Yritykset tuottavat myös laitteita, joissa ei ole käyttöliittymää, joiden kohdalla nähdään ettei tarvetta käyttäjäkokemustyölle ole. Testaus saattaa kuitenkin raportoida käytettävyysongelmista omaan työhönsä liittyen tällaisten tuotteiden kanssa. Tuote saattoi olla mahdoton testata eli tuotteen käytettävyyttä testaajan kannalta ei ollut mietitty etukäteen. Testaaja saattaa joutua purkamaan laitteen pystyäkseen testaamaan sitä. Asentajat ja jatkokehittäjät ovat myös sulautettujen järjestelmien käyttäjäryhmiä, jotka on helppo unohtaa. Suuri osa työntekijöistä tunnistaa kyllä tilanteita ja suunnitteluongelmia, jotka liittyvät tuotteen käyttöön. Käyttäjäkokemuksen käsite siis ymmärretään, mutta sitä ei tarvittavalla tavalla osata yhdistää omaan työhön.

Toisessa yrityksessä projekteissa oli mukana käyttäjäkokemussuunnittelija, mikäli asiakas näki tarpeen käyttäjäkokemustyön tekemiselle. Tämä saattoi aiheuttaa ongelmia, sillä näkemykset siitä, kuinka paljon käyttäjäkokemustyötä tuotteeseen tulisi tehdä, saattoivat vaihdella käyttäjäkokemussuunnittelijan ja asiakkaan välillä paljonkin. Käyttäjäkokemussuunnittelija joutuu tällöin taiteilemaan sen välillä, tehdäkö hyvä tuote ja ylitöitä, vaiko sen verran töitä kuin asiakas maksaa, välittämättä tuotteen laadusta niin paljon kuin itse haluaisi.

Sulautetun järjestelmän tuotantoprosessiin ei myöskään oltu määritelty paikkaa käyttäjäkokenemustyölle. Käyttäjäkokenemussuunnittelija tekee työtä monissa eri projekteissa silloin kuin tarve vaatii. Selkeä edustus käyttäjäkokenemustyölle prosessimallissa auttaisi myös kehittäjiä suuntaamaan kommunikointia käyttäjäkokenemussuunnittelijan puoleen säännöllisin väliajoin.

3. Millaiset ketterät käytännöt voivat tehostaa sulautettujen järjestelmien ohjelmistotuotantoprosessia, jossa tehdään myös käyttäjäkokenemustyötä?

Ketterän kehityksen julistuksessa (2001) korostetaan yksilöitä ja kanssakäymistä, toimivaa ohjelmistoa, asiakasyhteistyötä ja vastaamista muutokseen. Kaikkien näiden periaatteiden läpivieminen sulautettuun ohjelmistotuotantoprosessiin voisi tuoda lisäarvoa.

Yksilöiden ja kanssakäymisen korostamien menetelmien ja työkalujen sijaan voisi tarkoittaa tiimien muodostamista. Tiimissä mukana olisi työntekijöitä kaikilta projektin osa-alueilta. Ihanteellisessa tapauksessa tiimi työskentelisi yhteisessä tilassa, jolloin jatkuva kanssakäyminen olisi mahdollista. Parhaat kehitysratkaisut syntyvät itseorganisoituvissa tiimeissä. Käyttäjäkokenemustyöntekijöiden määrä yrityksessä oli kuitenkin pieni, eli käyttäjäkokenemushenkilöiden määrää tulisi joko lisätä, tarvittavat käyttäjäkokenemustyöhön liittyvät toimet voisi opettaa tiimin jäsenille, tai käyttäjäkokenemushenkilö voisi toimia osana useampaa tiimiä.

Toimivan ohjelmiston korostaminen kattavan dokumentaation sijaan voisi tarkoittaa sulautetun ohjelmistotuotantoprosessin ketteröitämistä niin, että ohjelmiston emulointi esimerkiksi asiakkaalle tai loppukäyttäjille onnistuisi jo varhaisessa vaiheessa projektia, ennen kuin laitteisto on valmis. Käyttäjäkokenemussuunnittelulle tuotteen emuloinnista on erityisen suurta hyötyä. Lisäksi dokumenttien tarkoituksesta ja muodosta voisi keskustella. Olisivatko esimerkiksi käyttäjä- tai systeemitarinat ja testitapaukset riittävä tapa kommunikoida erilaisten lakidokumentaatioiden lisäksi.

Asiakasyhteistyö massiivisten sopimusneuvottelujen sijaan voisi tarkoittaa asiakkaan ottamista mukaan tuotekehitysprosessiin. Liiketoiminnan edustajien ja kehittäjien olisi hyvä työskennellä yhdessä vähintään viikoittain koko projektin ajan. Näin välttyttäisiin isoilta muutoksilta projektin loppuvaiheessa. Erityisen tärkeää on tarkentaa laitteiston tilaa suunnitteluvaiheessa. Myös käyttäjät tulisi ottaa mukaan tuotekehitysprosessiin.

Muutokseen vastaaminen suunnitelmissa pitäytymisen sijaan voisi toimia ainakin niin, että ohjelmistoprosessi muutettaisiin ketteräksi. Lisäksi vaatimusmäärittelyä voisi ketteröittää. Prosessin alussa tapahtuva vaatimusmäärittely tulisi tehdä tiiviissä yhteistyössä asiakkaan, testauksen ja loppukäyttäjän kanssa. Loppukäyttäjän läsnäolo määrittelyn alkuvaiheessa voi ennaltaehkäistä laitteistoon pitkän suunnitteluprosessin aikana syntyviä vikoja.

Pohdintaa tutkimuskysymysten ulkopuolelta

Sulautetuille järjestelmille spesifisiä ongelmia on osittain hankala erottaa perinteiselle ohjelmistotuotannolle tyypillisistä ongelmista. Ongelmat ovat samoja, vain näkökulmat ja painotukset muuttuvat. Suurin selkeästi perinteisestä ohjelmistotuotannosta eroava ongelmakohta sulautettujen järjestelmien ohjelmistotuotannossa on suunnitelmaohjautuvan laitteistoprosessin ja usein nykyään ketteryyttä tavoittelevan ohjelmistoprosessin yhdistäminen. Kommunikointi näiden kahden tiiviisti toisissaan liitoksissa olevan eri prosessin välillä on ongelmallista, koska prosessit ovat hyvin erityyppiset ja prosessin tuotteiden yhdistäminen loppumetreillä voi olla hankalaa. Juuri tähän sulautetuille järjestelmille spesifiseen ongelmaan on tutkimuksen avulla pyritty tarttumaan mm. lisäämällä katselmoiteja ja jatkuvaa kommunikointia prosessien työntekijöiden välillä.

Toinen ongelmakohta on se, että sulautetut järjestelmät joudutaan aina rakentamaan alusta asti uudeksi tuotteeksi. Perinteisessä ohjelmistotuotannossa tuote voidaan usein rakentaa jo olemassa olevan alustan päälle, mikä saattaa helpottaa prosessia huomattavasti. Tutkimuksessa myös tähän sulautetuille järjestelmille spesifiseen ongelmaan on pyritty löytämään ratkaisuja ketteristä menetelmistä, joissa työntekijät voivat valita työtehtävistä itselleen parhaiten sopivat. Tällä voidaan vähentää merkittävästi uuden oppimisen määrää prosessissa, ja näin vähentää tuotekehitysprojektiin kuluva aikaa.

Yhdestätoista haastateltavasta kolmella oli selkeästi vähemmän kokemusta työskentelestä kohdeyrityksessä kuin kahdeksalla muulla. Näillä kolmella myös toimialakokemus oli vähäisempää muihin haastateltaviin nähden. Näiden "nuorempien" työntekijöiden haastatteluissa ajatus ketteryyden hyvistä puolista ja ketterän ajatusmaailman viemisestä koko prosessiin oli tarmokkaampaa kuin muissa haastatteluissa. Tämä saattaa johtua aiemman työpaikan vaikutuksesta, nykymenetelmissä käytettävän ajatusmaailman omaksumisesta, tai ehkä vähäisemmästä kokemuksesta sulautettujen järjestelmien valmistuksessa. Lisäksi lähes poikkeuksetta haastatteluissa pohdittiin ketterien menetelmien tuomia etuja ja haittoja, mutta huomautettiin, että yritykselle selkeä muutos ketteryyttä kohti voi olla liian suuri. Vaikutti siltä, että haastateltavia houkuttaisi kokeilla ketteryyttä monellakin tasolla, mutta koska ajatukset ja asenteet muutosta kohtaan ovat joko neutraaleja tai hieman vastahakoisia, johtuen ajan ja resurssien vähyydestä, ei asiaa ole vielä lähdetty kokeilemaan. Pienet muutokset toimintatavoissa, joilla ketteryyttä voitaisiin edistää otettaisiin kuitenkin mieluusti vastaan, ja haastateltavat olivatkin kiinnostuneita kuulemaan tutkimuksen tuloksia.

Suurin osa (10/11) molempien yritysten haastatelluista tunsi tekevänsä käyttäjäkokemustyötä. Tämä oli merkittävän positiivinen tutkimustulos, sillä suurin osa haastatelluista oli laitteistopuolen työntekijöitä, joilla ei ole suoraa yhteyttä laitteen käyttämiseen saati loppukäyttäjiin. Käyttäjäkokemuksen käsite on siis kaikille tuttu ja työntekijät ymmärtävät yhteyden työnsä ja käyttäjien välillä. Molempien yritysten työntekijät (11/11) näkivät myös käyttäjäkokemustyön tekemisen tärkeänä. Silti käyttäjäkokemus-

työhön liittyviä perustoimia ei ollut täysin sisäistetty. Loppukäyttäjän sitouttaminen prosessiin alusta asti ja kiinteä työ kehitystiimin kanssa oli haastateltavien keskuudessa vielä melko kaukainen ajatus.

Tutkimuksessa kävi ilmi myös se, että vaikka haastateltavat olivat hyvin selvillä käyttäjäkokemuksen käsitteestä, ja sanoivat tekevänsä käyttäjäkokemustyötä, silti joitain käyttäjäkokemustyöhön oleellisesti liittyviä asioita ei pidetty itsestäänselvyyksinä. Esimerkiksi haastatteluissa kävi ilmi asiakkaan kanssa tehdyn yhteisen suunnittelutyön vähyys. Myös loppukäyttäjien sitouttamista koko prosessiin ei nähty mahdollisena, mikä johtuu ehkä tottumuksesta. Osa tuotteista nähtiin myös sellaisina, ettei niiden käyttäjäkokemukseen tarvitse lainkaan panostaa. Luvussa kolme luokiteltiin käyttäjiksi myös kolmannet osapuolet, jotka eivät välttämättä koskaan käytä laitetta, mutta ovat syystä tai toisesta tekemisessä laitteen kanssa. Tällaisia käyttäjiä myös näissä projekteissa on paljon. Kaikki tuotteen kanssa tekemisissä olevat henkilöt ovat käyttäjiä, ja prosessissa luodaan käyttäjäkokemusta myös näille henkilöille.

Vaikka kohdeyrityksissä oli kiinnostusta käyttäjäkokemukseen ja käyttäjäkokemustyöhön liittyviin asioihin, ketterä sulautetun ohjelmiston kehitys oli lähempänä haastateltavien omaa työtä. Tämä on itsessään niin monimutkainen ja melko tuore asia, että haastateltavat keskittyivät vastauksissaan paljon nimenomaan siihen. Ketteryys kiinnostaa, koska se on trendikästä, ja monet yritykset ovat saaneet ketteryyden kautta merkittäviä etuja toimintaan. Näen kuitenkin tutkijana, että käyttäjäkokemussuunnittelu on osana kaikkea. Prosessit eivät ole erotettavissa toisistaan. Kaikki tuotteeseen liitoksissa olevat ihmiset ovat muodostavat tuotteesta jonkinlaisen käyttäjäkokemuksen, ja koko prosessi vaikuttaa siihen, miten käyttäjä kokee tuotteen. Käyttäjäkokemustyön tekeminen vaikuttaa kuitenkin sulautettujen järjestelmien ohjelmistotuotannossa olevan vielä lapsenkengissä niissä projekteissa, joissa loppukäyttäjälle ei tehdä varsinaista käyttöliittymää. Tämä on yksi syy siihen, miksei tutkimuksen käyttäjäkokemustyöosuudessa päästä kovin syvälle.

Eräs haastateltavista (H3) kertoi, että mikäli prosessia aiotaan kehittää, niin olisi tärkeää että kehittäminen olisi järjestelmällistä. Oli kyseessä sitten työkaluasiasia tai prosessikehitys. On järkevää, että tietyt työntekijät valmistelevat asian ja sitten muut sitoutetaan mukaan muutokseen. Ei niin, että kaikki kokeilevat uutta asiaa itsenäisesti. ”*Selkeästi valmisteltu, enemmän valmisteltu pala ja otetaan se nyt käyttöön ja sitten siirrytään seuraavaan*”. (H3)

8.2 Tutkimuksen luotettavuuden arviointi

Aiheen laajuus vaikutti työn selkeyteen. Rajatumpi aihe olisi ehdottomasti ollut tutkimukseen selkeämpi valinta, mutta tätä nimenomaista ilmiötä tarkastellessa on pakko katsoa sen kaikkia puolia. Lisäksi, jos ketterästä sulautetusta ohjelmistotuotannosta ja käyttäjäkokemuksesta olisi aiempaa tutkimusta, tutkimuksen tekeminen olisi ollut huo-

mattavasti helpompaa ja mielekkäämpää. Tutkimusta tehdessä kuitenkin huomasi miten tärkeä aihe on, sillä sulautettujen järjestelmien maailmassa käyttäjäkokemustyö osana prosessia on vielä lapsenkengissään. Parannuksia prosessiin kaivataan myös muilla projektin osa-alueilla.

8.2.1 Tutkimuksen validiteetti ja reliabiliteetti

Seuraavissa kohdissa pohditaan tutkimuksen validiteettia ja reliabiliteettia. Validiteetin ja reliabiliteetin tutkimiseen liittyvä teoria on esitetty luvussa 4.

Sisäinen validiteetti

Sisäisellä validiteetilla tarkoitetaan (4.4.2) tutkijan tieteellisen otteen ja tieteenalan hallinnan voimakkuutta. Tässä tutkimuksessa sisäistä validiteettia pyrittiin parantamaan tutkimuksen teoriaosuudessa, eli tutustumalla aiemmin tehtyyn tutkimukseen ja alan käsitteistöön, sekä tutkimukseen sopivan tutkimusmetodin valinnalla että tehden tutkimusta tutkimusmetodia seuraamalla. Validiteettia heikensi tutkimuksen aiheen laajuus. Tutkimuksessa käsitellään neljää erikoisosaamisaluetta (HW, SW, UX, Agile), joista yhteenkin perehtyminen vaatii paljon asiantuntemusta. Aihe on kuitenkin tieteellisesti kiinnostava, sillä tutkimusta siitä ei ole tehty ja hyvistä käytänteistä on teollisuudessa huutava pula. Sisäiseen validiteettiin on vaikuttanut heikentävästi myös aika, joka on kulunut kahden tutkimuksen välissä. IT-maailmassa asenteet ja käsitykset työkaluista ja menetelmistä muuttuvat nopeasti.

Ulkoinen validiteetti

Ulkoisella validiteetilla pyritään varmistamaan, että tutkija kuvaa tutkimuskohteen sellaisena kuin se on, ja päätelmät tehdään sen perusteella (Eskola & Suoranta 1998). Tässä tutkimuksessa ulkoiseen validiteettiin on pyritty vaikuttamaan mm. litteroimalla haastattelut sanatarkasti muuttamatta haastateltavien lausumaa sisältöä, tekemällä lisäkysymyksiä tarkentamaan haastateltavien näkemyksiä ja siteeraamalla tutkimushenkilöitä tuloksissa suoraan. Myös yrityksiin toimitetut raportit ja niistä saatu palaute auttaa varmistamaan ulkoista validiteettia. Lisäksi tutkimuksen alussa muodostettuihin tutkimuskysymyksiin on löydetty vastauksia. Ulkoista validiteettia heikensi se, ettei tutkimusta aiheesta ole olemassa. Tutkimukselle ei ole olemassa vertailukohteita.

Reliabiliteetti

Hirsjärvi & Hurme (2011) kehottavat tutkijaa luotettavuutta tutkiessaan kysymään itseltään lisäkysymyksiä, esimerkiksi: "Miten tutkija tekee tutkimusprosessin?" Saaranen-Kauppinen & Puusniekka (2006) lisäävät, että "*Tulosten merkitystä ja oikeellisuutta voidaan vahvistaa esittämällä perusteellinen kuvaus aineistosta ja sen analyysistä*". Tässä tutkimuksessa on seurattu tarkasti Hirsjärvi & Hurmeen (2011) "Tutkimushaastattelu"-kirjassa esitettyä tutkimusprosessia (luku 4). Tutkimuksen vaiheet on kuvailtu

mahdollisimman tarkasti, jotta tutkimus olisi toistettavissa. Haastattelut litteroitiin sana sanalta, vaikka kevyemmänkin menetelmän käyttö olisi diplomityössä ollut sallittua. Tutkija on kiinnittänyt tutkimusaineistoa karsiessaan erityistä huomiota siihen, ettei tutkimusaineistosta mitään olennaista jäisi tutkijan omien mieltymysten takia esittämättä. Itse asiassa objektiivisuuteen pyrkiminen on aiheuttanut haasteita tutkimusmateriaalin karsinnassa. Kaikkia haastatteluissa raportoituja haasteita ei ole sisällytetty diplomityöhön.

Eskolan ja Suorannan (1998) mukaan tutkimuksen reliabiliteettia voidaan arvioida kolmesta näkökulmasta: indikaattoreita vaihtamalla, useampaa havainnointikertaa käyttämällä ja useampaa havainnoitsijaa käyttämällä. Tässä tutkimuksessa indikaattorit vaihtuivat esimerkiksi niin, että "työn sirpaloitumisesta" puhuttiin eri haastateltavien (roolien) toimesta ja eri prosessin vaiheissa. Yksi haastateltavista mainitsi tutkimuksessa termin "työn sirpaloituminen", toinen sen, että ylläpito vie paljon hänen aikaansa, vaikka on kehittäjä". Lisäkysymyksen jälkeen selvisi, että haastateltavat puhuvat samasta asiasta, mutta eri näkökulmista.

Toinen reliabiliteetin tarkistamiseen liittyvä toimi oli useamman havaintokerran käyttö tutkimuksessa. Haastattelututkimuksessa tällä tarkoitetaan sitä, että sama asia kysytään eri muodoissa ja eri henkilöiltä. Tämän tutkimuksen aineistossa haasteita nousi esiin sekä kysyttäessä prosessista yleisesti että ongelmista kysyttäessä. Lisäksi vastaukset toistuivat eri henkilöitä haastateltaessa. Pysyväksi ilmiöksi yrityksessä esiintyviä haasteita ei pystytty varmistamaan, sillä ensimmäisen haastattelukierroksen jälkeen lisähaastatteluja ei ole tehty ja toisaalta raportin palauttamisen jälkeen on toive, että yritys pyrkii parantamaan prosessiaan oheistuksen mukaan. Diplomityö on laajuudeltaan rajattu, joten lisää erilaisia menetelmiä ei tämän tutkimuksen piirissä ollut mahdollista käyttää. Silti tutkimuksen reliabiliteetin kannalta olisi mielenkiintoista tehdä yritykseen esimerkiksi havainnointitutkimusta tai laajempaa kyselytutkimusta mahdollisten muutostöiden jälkeen.

Kolmas tapa tukea tutkimuksen reliabiliteettia on käyttää useampaa henkilöä tekemässä tutkimusta (Eskola & Suoranta 1998). Diplomityö tehdään usein yksin, siksi tässä tutkimuksessa monen tutkijan käytölle ei ole ollut mahdollisuutta.

Aineisto on kerätty mukavuusotannalla ja haastatteluihin on valikoitunut tietynlaisia työntekijöitä, mikä vähentää tutkimuksen luotettavuutta. Haastatteluiden alussa varmistettiin, että kaikki työntekijät ovat haastattelussa omasta tahdostaan. On helpompaa kerätä luotettavaa tietoa työntekijöiltä, jotka ovat kiinnostuneita asiasta, ja motivoituneita vastaamaan, kuin niiltä, joita haastattelun sisältö ei kiinnosta. Toisaalta, tämä aiheuttaa sen, että kaikenlaiset mielipiteet asioista eivät pääse esiin.

8.2.2 Muut mittarit

Muita mittareita kvalitatiivisen tutkimuksen luotettavuuden arvioimiseksi on esitetty kohdassa 4.4.4. Eskola & Suoranta (1998) esittivät uskottavuuden, siirrettävyyden, varmuuden ja vahvistuvuuden käsitteet luotettavuuden pohtimiseen. Ohessa tutkimuksen luotettavuutta on arvioitu näiden mittareiden avulla.

Uskottavuus

Tässä tutkimuksessa uskottavuutta on pyritty varmistamaan jo haastattelutilanteessa lisäkysymyksiä kysymällä ja varmistamalla tutkijan ymmärrys haastateltavan vastauksen sisällöstä. Lisäksi sekä yritykselle toimitettava raportti että diplomityö on annettu luettavaksi yritykseen ja pyydetty kommentoimaan sen sisältöä ja oikeellisuutta. Mikäli muutostarpeita on ilmennyt, on niitä harkittu ja pyritty muuttamaan haastateltavien tarkoitusta vastaaviksi.

Siirrettävyys

Siirrettävyyttä, eli tilanteesta toiseen vastaavaan tilanteeseen siirtymistä on pyritty parantamaan valitsemalla tutkimukseen kaksi yritystä, joiden toimiala on sama ja työntekijöille on esitetty samat kysymykset. Molemmista yrityksistä on tutkimuksessa saatu samansuuntaisia tuloksia. Yleistyksiä on kuitenkin hankala tehdä, sillä toimialat ja tuotteen sulautettujen järjestelmien tuotannossa vaihtelevat suuresti. Mielenkiintoista olisi-kin käyttää tutkimustuloksia tutkimuksen ulkopuolella olevan yrityksen toiminnan parantamiseen. Siirrettävyyttä on pohdittu yleistettävyyden näkökulmasta tekstissä vielä myöhemmin.

Varmuus

Varmuudella tarkoitetaan sen pohtimista, kuinka paljon tutkijan ennakko-oletukset tutkimuksen tuloksista vaikuttavat lopputuloksiin. Hyvä puoli siinä, että tutkimusta tekee ensimmäistä kertaa, on ennakko-oletusten vähyys. Koska kokemusta alalta ei juurikaan ole, on helpompi ehkä olla objektiivinen, kuin alaa kymmeniä vuosia nähnyt. Vastapuolena on tietenkin sisäisen validiteetin heikkeneminen. Tässä tutkimuksessa varmuutta on vahvistettu esimerkiksi tutkimusmenetelmän valinnalla, teoriaosuuteen paneutumalla ja varmistamalla haastattelijoilta henkilökohtainen halukkuus haastatteluun.

Vahvistuvuus

Vahvistuvuudella tarkoitetaan sitä, että tutkimuksessa tehdyt tulkinnat saavat tukea aiemmista tutkimuksista. Tutkimuksen teoriaosuudessa käsitellyt tutkimukset viittaavat saman suuntaisiin tutkimustuloksiin sulautettujen järjestelmien kehityksessä esiintyvistä ongelmista. Koska aiempaa tutkimusta käyttäjäkokemustyöstä sulautettujen järjestelmien tuotantoprosessissa ei ole olemassa, suoraa tukea johtopäätöksille on mahdotonta

saada. Kuitenkin tutkimuksen teoriaosuudessa esitettyihin asioihin voidaan tutkimustuloksista löytää selkeitä yhteneväisyyksiä.

Yleistettävyyys

Tapaustutkimuksella pyritään lisäämään ymmärrystä tutkitusta ilmiöstä (Hirsjärvi & Hurme 2011). Eskolan ja Suorannan (1998, s. 66) mukaan tapaustutkimuksesta saatua tietoa ei voida yleistää tilastollisessa merkityksessä. Tässä tutkimuksessa ensimmäisen tutkimuskysymyksen vastauksia voidaan verrata aiempiin tutkimustuloksiin, sillä tutkimustuloksia sulautettujen järjestelmien tuotantoprosessin ongelmista on olemassa. Tutkimuskysymyksistä kaksi ja kolme ei kuitenkaan ole täysin samanlaista tietoa saatavilla. Tutkimuksen tulokset ovat jossain määrin samansuuntaisia kuin aiemmissa tutkimuksissa ilmenneet tulokset. Esimerkiksi Kuusisen (2015) tutkimuksessa tutkittavissa yrityksissä oli myös sulautettuja järjestelmiä tuottavia yrityksiä ja samankaltaisia ongelmia havaittiin kuin tämän tutkimuksen yrityksissä. Tämä voi antaa suuntaa myös tämän tutkimuksen yleistettävyyteen.

Yleistettävyyttä parantaessa Eskola & Suoranta (1998) korostavat, että haastateltavia valittaessa on tärkeää, että haastateltavilla on keskenään samanlainen kokemusmaailma ja kokemusta tutkimuksen aihepiiristä. Myös kiinnostus ja myönteinen suhtautuminen tutkimukseen mainittiin tärkeäksi yleistettävyyden kriteeriksi (ibid.). Tässä tutkimuksessa haastateltavat valittiin yrityksissä yhden projektin sisältä, jotta näkemys projektista saataisiin melko yhtenäiseksi. Lisäksi ennen haastattelua varmistettiin sekä tutkimus-suostumuksella (LIITE A) että suullisesti, että haastateltavat olivat halukkaita olemaan mukana tutkimuksessa. Näillä toimilla pyrittiin siis lisäämään niin tutkimuksen luotettavuutta, kuin yleistettävyyttä.

Tämän diplomityön tuloksia voidaan hyvin soveltaa tutkimuksessa mukana olleiden yritysten tasolla. Lisäksi tutkimuksen tuloksista hyötyy varmasti koko toimiala. Tutkimuksen tavoitteena oli etsiä haasteita yritysten toiminnasta, ja etsiä kehitysehdotuksia löydettyihin haasteisiin. Huolellisen tutkimuskirjallisuuden läpikäynnin tuloksena yrityksille tarjottiin ratkaisuehdotuksia ongelmiin, ja osa näistä otettiin myönteisinä ehdotuksina vastaan. Ehdotusten sopivuutta voitaisiin kuitenkin arvioida tarkemmin vasta myöhemmässä lisätutkimuksessa.

8.3 Tutkimusetiikka

Tämän tutkimuksen etiikkaa on arvioitu Suomen tutkimuseettisen neuvottelukunnan tutkimusetiikkadokumentin pohjalta. Dokumentin kohdat on lueteltu luvussa 4.7.

Tutkimusetiikkadokumentin ensimmäisessä kohdassa kehoitetaan tutkijaa rehellisyyteen tarkkuuteen ja huolellisuuteen. Tässä tutkimuksessa tämä on pyritty osoittamaan mm. tuottamalla tutkimuksesta dokumentti, diplomityö, joka menee arviointiin yliopistolle. Diplomityössä on huolellisesti selostettu tutkimuksen kulku ja sen tulokset (avoimuus,

huolellisuus, tarkkuus). Lisäksi tutkimuksen luotettavuutta on arvioitu monin eri mittarein.

Tutkimuksen tulee myös kestää tarkastelua tutkimusmenetelmän ja sen noudattamisen Tutkimukseen valittu tutkimusmenetelmä on valittu tutkimuskirjallisuuden avulla siihen sopivaksi. Tutkimuksessa on seurattu tapaustutkimuksen tutkimusmetodia ja tutkimuksen kulku on raportoitu tähän dokumenttiin.

Läpinäkyvyyttä on vahvistettu tutkimustulosten jakamisella kohdeyrityksiin ja pyytämällä palautetta raportista. Lisäksi tutkimus menee arvioitavaksi diplomityön tarkastajalle. Osa tutkimuksen tuloksista on salattu yritysten pyynnöstä.

Muiden tutkijoiden tuotantoa tässä tutkimuksessa on pyritty noudattamaan viittauskäytännöllä, eli lähteet ja niihin liittyvät lisätiedot on merkitty tutkimuksen loppuun. Viittauskäytäntö noudattaa TTY:n diplomityöohjeen viittausstandardia.

Tutkimus on suunniteltu huolellisesti käyttäen apuna lähdekirjallisuutta. Se on pyritty toteuttamaan suunnitelmien mukaisesti ja tutkimuksen tulokset on raportoitu ja tallennettu varmuuskopioin. Myös tutkimukseen liittyvä aineisto on tallennettu varmuuskopioin. Aineisto on salattu, joten sitä ei ole luovutettu kuin yrityksen ja työn tarkastajien käyttöön.

Luvista ja sopimuksista huolehtiminen on osa eettistä tutkimusta. Tämän diplomityön aihe ja kieli on hyväksytty Tampereen teknillisen yliopiston Tieto- ja sähkötekniikan tiedekunnan 4.5.2016 pitämässä kokouksessa. Tutkittavia on pyydetty täyttämään tutkimukseen liittyvä haastattelusuostumus, jossa pyydettiin mm. lupaa tallentaa haastattelut. Nämä tallenteet ovat...

Esteellisyyteen liittyviä kysymyksiä on mietitty mm. diplomityön tarkastajaa valittaessa. Koska tutkijan pääaine on ”käyttäjäkokemus” (engl. User Experience), on työn tarkastajaksi valittu pääaineen professori. Esteellisyyteen liittyviä asioita mietittiin myös tutkijan vaihtaessa työpaikkaa erääseen suuren ohjelmistoalan yritykseen. Tämän ei kuitenkaan nähty vaikuttavan tutkimuksen luotettavuuteen.

Diplomityön rahoitukseen on osallistunut toinen yrityksistä. Lisäksi työtä on tehty noin kuukausi TTY:N rahoituksella. Tutkimukseen on saatu myös stipendi Väänästen seura Ry:ltä vuonna 2013. Muut kulut ovat tutkijan itse kustantamia.

8.4 Jatkotutkimuskysymyksiä

Haastatteluista ja muista tapaamisista yrityksen henkilökunnan kanssa nousi säännöllisesti kysymykseksi koko prosessin muuttaminen ketteräksi. Olisikin erittäin mielenkiintoista selvittää onko ketteryyden lisääminen laitteistokehitykseen esimerkiksi Scrummenetelmän avulla mielekäs vaihtoehto. Laitteistopuoli nähdään luonteeltaan suunnit-

telmaohjautuvana, ja vain joissain isoissa prosesseissa on onnistuttu jakamaan laitteistokehitys lyhyiden iteraatioiden kokoisiksi siivuiksi. Tutkimuksessa olisi hyvä olla mukana sekä ohjelmisto- että laitteistopuolen osaamista. Mikäli yrityksissä riittäisi kiinnostusta laittaa resursseja laitteistopuolesta parhaiten ymmärtäviä tutkimukseen mukaan, tähän kysymykseen perehtyminen, mahdollinen ketterän menetelmän valinta tai parhaiden käytäntöjen lisäsoveltaminen olisi erittäin kiinnostavaa.

Myös muutosjohtamisen tutkiminen ja muutostoimenpiteiden realisointi tähän nimenomaiseen tutkimukseen liittyen olisi kiinnostavaa. Yrityksessä asenteet ketteriä periaatteita kohtaan ovat myönteisiä, mutta laitteistopuolen edustajat ovat ehkä syystäkin skeptisiä asian suhteen.

Helsingin sanomien mukaan (HS 17.2.2015) on tutkittu, että 80% kaikista palavereista tuntuu työntekijöiden mielestä turhilta. Tämä on merkittävää, sillä todella monet tietotekniikan alan yritykset käyttävät paljon aikaa erilaisten palaverien pittoon. Koska tämän työn toisessa tutkimuksessa esiin nousi kysymys palaverien turhuudesta, tästä aiheesta olisi kiinnostavaa tehdä pitkäaikaistutkimusta firman sisältä havainnoimalla ja keräämällä tietoa palaverien osallistujien ajatuksista pidettyjen palaverien tehokkuudesta ja kuinka sekä palaverikäytäntöjä, että niissä tapahtuvaa kommunikointia voisi kehittää.

8.5 Yhteenveto

Tässä tutkimuksessa kartoitettiin kahden yrityksen ongelmia sulautettujen järjestelmien tuotannossa ja käyttäjäkokemustyön tekemisessä sulautetussa ohjelmistotuotannossa. Näihin ongelmiin pyrittiin löytämään ratkaisuja ketterien menetelmien avulla. Ratkaisujen tarkoituksena oli tuotekehitysprosessin sujuvoittaminen ja tehostaminen.

Suurimmiksi ongelmiksi tutkimuksessa nousivat käyttäjäkokemustyöhön, tiedonkulkuun ja kommunikointiin, prosessimalliin, uudelleenkäytettävyyteen ja projektin johtamiseen liittyvät asiat. Käyttäjäkokemustyön tekeminen sulautettujen järjestelmien tuotantoprosesseissa on vähäistä ja loppukäyttäjiltä ei saada palautetta tuotteista. Prosessimalli ei tue työntekijöiden toimintaa riittävästi, esimerkiksi kommunikointi ei ole systemaattista ja määriteltyä, vaan työntekijöiden omaan osaamiseen perustuvaa. Myös jo kertaalleen tehdyn työn uudelleenkäyttö on vähäistä. Projektinjohtamista kritisoitiin liittyen työntekijöiden resursointiin.

Tutkimuksessa havaittiin samoja ongelmia sulautettujen järjestelmien kehityksessä kuin tutkimuskirjallisuudessa. Ongelmiksi koettiin mm. kahden luonteeltaan hyvin erilaisen prosessin yhdistäminen ja näiden prosessien välinen kommunikointi, tehdyn työn uudelleenkäytettävyyden puute ja ihmisten resursointi johdon tasolta. Näitä ongelmia tavaan myös perinteisessä ohjelmistotuotannossa, mutta erityisen tyypillisiä ne ovat sulautetuille järjestelmille.

Ratkaisuiksi ongelmiin ehdotettiin Scrumista, Leanista, Kanbanista ja XP:stä tuttuja käytänteitä. Esimerkiksi prosessien yhdistämiseen ja niiden väliseen kommunikointiin esitettiin ratkaisuksi katselmointeja prosessien välille ja tiimien muodostamista projekti-kohtaisesti. Työntekijöiden tehottomaan resursointiin ratkaisuksi ehdotettiin itseorganisoituvia tiimejä, joissa työntekijät voivat valita työtehtävänsä parhaan osaamisensa mukaan.

Tutkimuksen tulosten pohjalta kehitettiin malli, joka sopisi molempien tutkimuksessa mukana olleiden yritysten käyttöön. Malli on hybridimalli, jossa laitteistoa tehdään suunnitelmaohjautuvasti, ja ohjelmistotuotannossa pyritään ketteryuteen. Mallissa on otettu huomioon molempien yritysten erityispiirteet.

Tutkimuksen luotettavuutta on tarkasteltu reliabiliteetin, validiteetin ja yleistettävyyden käsitteiden kautta. Lisäksi tutkimuksen etiikkaa on arvioitu eettisen oheistuksen avulla.

LÄHTEET

Alben, L. 1996. Quality of Experience: Defining the Criteria for Effective Interaction-Design. *Interactions* 3, 3, pp. 11–15. Saatavissa: <http://albenfaris.com/downloads/pdf/quality.pdf>.

Abrahamsson, P. 2007. Speeding up embedded software development. VTT, Technical Research centre of finland. ITEA Information technology for european advancement Innovation report. Saatavissa: <https://itea3.org/project/result/download/5583>.

Abras, C., Maloney-Krichmar, D. & Preece, J. 2004. User-Centered Design. In: Bainbridge, W. *Encyclopedia of Human-Computer Interaction*. Thousand Oaks: Sage Publications.

A Guide to the Project Management Body of Knowledge. 2000. Newton Square Pennsylvania USA, Project Management Institute. 216 p.

Baxter, P. & Jack, S. 2008. Qualitative Case Study Methodology: Study Design and Implementation for Novice Researchers. *The Qualitative Report* 13, 4, pp. 544–559.

Brook, F. 1986. No Silver Bullet. In: Bowen, J., Hinchey, M. *High-Integrity System Specification and Design*. UK, Springer. pp. 11–27.

Catsoulis, J. 2005. *Designing Embedded Hardware*. California, USA, O'Reilly Media. 400 p. Saatavissa: <http://www.maerivoet.org/website/software/arduino/manuals/electronics/designing-embedded-hardware.pdf>.

Charvat, J. 2002. Project communications: A plan for getting your message across. Internetlähde. Viitattu: 09.05.2016. Saatavissa: <http://www.techrepublic.com/article/project-communications-a-plan-for-getting-your-message-across/1061894>.

Coplien, J. & Harrison, N. 2005. *Organizational Patterns of Agile Software Development*. 3rd edition. Pearson. Prentice Hall. 401 p.

Cubas, G. 2010. Effort Estimation for Embedded Software Development. Viitattu: 27.04.2016. Saatavissa: <http://www.avionyx.com/publications/e-newsletter/issue-3/123-effort-estimation-for-embedded-software-development.html>.

Ebert, C. & Jones, C. 2009. Embedded software: facts, figures and future. *Computer* 42, 4, 42–52.

Ebert, C. & Salecker, J. 2009. Embedded Software – Technologies and Trends. *IEEE Software* 26, 3, pp. 14–18.

Eskola, J. & Suoranta, J. 1998. Johdatus laadulliseen tutkimukseen. Jyväskylä, Suomi, Gummerus Kirjapaino OY. 268 s.

Ferreira, J., Noble, J. & Biddle, R. 2007. Up-front Interaction Design in Agile Development. In: Concas, G., Damiani, E., Scotto, M., Succi, G. LNCS vol. 4536. Springer, Heidelberg. pp. 9–16.

Gothelf, J. & Seiden, J. 2013. Lean UX - Applying Lean Principles to Improve User Experience. First edition. USA, O'reilly Media. 130 p.

Green, B, 2004. Agile methods applied to embedded firmware development. IEEE, Agile Development Conference, Salt Lake City, UT, USA, June 22–26, 2004. Los Alamitos, CA, USA. IEEE Computer Society. pp. 71–77.

Grenning, J. 2002. Extreme Programming and Embedded Software Development. Embedded Software Conference, Chigaco, March 2002.

Helsingin Sanomat. 2015. Verkkouutinen 27.3.2015. Viitattu: 30.04.2016. Saatavissa: <http://www.hs.fi/ura/a1305928921054>.

Henzinger, T. & Sifakis, J. 2006. The Embedded Systems Design Challenge. Lecture Notes in Computer Science 4085, 2006. pp. 1–15.

Hirsjärvi, S. & Hurme, H. 2011. Tutkimushaastattelu. Teemahaastattelun teoria ja käytäntö. Helsinki, Helsinki University Press. 213 s.

Hirsjärvi, S., Remes, P. & Sajavaara, P. 1997. Tutki ja kirjoita. Helsinki, Kirjayhtymä. 432 s.

Ihalainen, J. 2007 (arvio). Sulautettujen järjestelmien arkkitehtuurit. Luentomateriaali. Kokkolan yliopistokeskus Chydenius, Tietoliikennelaboratorio. Viitattu: 8.5.2016. Saatavissa: <http://slideplayer.biz/slide/5208659>.

IHTE. 2008. Käyttäjakeskeinen suunnittelu. Luentokalvot. Tekijä ei tiedossa. Viitattu: 09.05.2016. Saatavissa: <http://www.cs.tut.fi/~kaper/syksy08/luennot/S08L2.pdf>.

Is Involving Testers Early in the Project a Workable System. Viitattu 3.3.2016. Saatavissa: <http://www.testuff.com/is-involving-software-testers-early-in-the-project-a-workable-system>.

ISO. 1998. ISO 9241-11:1998. Ergonomic requirements for office work with visual display terminals (VDTs) - part 11: Guidance on usability. Genève, CH:International Organization for Standardisation.

Järvinen, H-M. Mikkonen, T. 2012. Sulautettu ohjelmointi. Tampere, Suomi, Kopio Niini Oy. 206 s. Saatavissa: <http://www.cs.tut.fi/~sulo/pruju/sulo-pruju-1.pdf>.

Kasila, E. 2013. Ketterien menetelmien hyödyntäminen sulautettujen järjestelmien kehityksessä. Diplomityö. Lappeenrannan teknillinen yliopisto. Saatavissa: http://www.doria.fi/bitstream/handle/10024/91461/Ketterien_menetelmien_k%E4ytt%E4rjestelmien_kehityksess%E4.pdf?sequence=2.

Ketterän ohjelmistokehityksen julistus. 2001. Viitattu: 20.02.2016. Saatavissa: <http://www.agilemanifesto.org/>.

Kettunen, P. & Laanti, M. 2005. How to steer an embedded software project: tactics for selecting the software process model. *Information and Software Technology* 47, 9, pp. 587–608.

Klein L. 2013. *UX for Lean Startups*. 1st edition. CA, USA, O'Reilly Media, Inc. 206 p.

Kniberg, H. & Skarin, M. 2010. *Kanban and Scrum – making the most of both*. USA, C4Media Inc, Publisher of InfoQ.com. 120 p. Saatavissa: http://www.agileinnovation.eu/wordpress/wp-content/uploads/2010/09/KanbanAndScrum_MakingTheMostOfBoth.pdf.

Kuusinen, K. 2015. *Integrating UX Work in Agile Enterprise Software Development*. Dissertation. Tampere, Tampereen Teknillinen Yliopisto, Department of Pervasive Computing. Juvenes Print. Publication XXX. XXX p.

Kuusinen, K. 2016. *BoB - A Framework for Organizing Within-Iteration UX Work in Agile Development*. In: Cockton, G., Larusdottir, M.K., Gregory, P. & Cajander, A. *Integrating User Centred Design in Agile Development*. London, Springer International Publishing. In press.

Lavagno, L., Sangiovanni-Vincentelli, A. & Sentovich, E. 1999. Models of Computation for Embedded System Design. In: Jerraya, A. & Mermet, J. *System-Level Synthesis*. NATO Science Series, vol 357. Springer Netherlands. pp. 45–102.

Lean Manufacturing Junction. 2014. Benefits of lean manufacturing. Web sivu. Viitattu: 25.04.2016. Saatavissa: <http://www.lean-manufacturing-junction.com/benefits-of-lean.html>.

Lee, R. 1993. *Doing research on sensitive topics*. London, SAGE Publications Ltd. 240 p.

Lehtonen, T. Tuomivaara, S. Rantala, V. Käsälä, M. Mäkilä, T. Jokela, T. Könnölä, T. Kaisti, M. Suomi, S. Ylitolva, M. & Isomäki, M. 2014. *Sulautettujen järjestelmien ketterä käsikirja*. Turku, Painosalama OY. 98 s. Saatavissa: <http://urn.fi/URN:ISBN:978-951-29-5838-2>.

Liggesmeyer, P. & Trapp, M. 2009. Trends in Embedded Software Engineering. IEEE Software 26, 3, pp. 19–25.

Luomala, A. 2008. Muutosjohtamisen ABC – Ajatuksia muutoksen johtamisesta ja ihmisten johtamisesta muutoksessa. Raportti. Ihmisten ja työhyvinvoinnin johtamisen tutkimus- ja kehittämisryhmä. Tampereen yliopiston kauppakorkeakoulu. Saatavissa: <http://www.uta.fi/jkk/synergos/tyohyvinvointi/oppaat/muutoskirja.pdf>.

Mäkelä, K. 1987. Yhteiskuntatieteellisen tiedonhankinnan eettiset säännöt ja tietosuojat. Teoksessa Mäkelä, K. (toim.) Tieteen vapaus ja tutkimuksen etiikka. Helsinki, Tammi, s. 180–195.

Orr, K. 2004. Agile Requirements: Opportunity or Oxymoron? IEEE Software 21, 3, pp. 71–73.

Pauwels, P. & Matthyssens, P. The Architecture of Multiple Case Study Research in International Business. In: Marschan-Piekkari, R. & Welch, C. Handbook of Qualitative Research Methods for International Business. Saatavissa: https://www.researchgate.net/publication/234021803_The_Architecture_of_Multiple_Case_Study_Research_in_International_Business.

PwC Technology Institute. 2013. Accelerating embedded software development via agile techniques. Raportti. Viitattu: 30.04.2016. Saatavissa: <http://www.pwc.com/us/en/technology/publications/assets/pwc-agile-embedded-software-development.pdf>.

Royce, W. 1970. Managing the development of large software systems. Proceeding of the IEEE WESCON, August 1970, pp. 1–9. Saatavissa: http://leadinganswers.typepad.com/leading_answers/files/original_waterfall_paper_winston_royce.pdf.

Runeson, P. & Höst, M. 2008. Guidelines for conducting and reporting case study research in software engineering. Empirical software engineering 14, 2, pp. 131–164.

Ruuska, T. 2012. Vaatimusmäärittely ketterässä ohjelmistokehityksessä. Pro gradu-tutkielma. Jyväskylän yliopisto, Tietojärjestelmätiede, 78 s.

Saaranen-Kauppinen, A. & Puusniekka, A. 2006. KvaliMOTV - Menetelmäopetuksen tietovaranto. Yhteiskuntatieteellinen tietoaarkisto. Verkkojulkaisu. Viitattu: 27.04.2016. Saatavissa: <http://www.fsd.uta.fi/menetelmaopetus>.

Salo, O. & Abrahamsson, P. 2008. Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of XP and Scrum. IET Software 2, 1, pp. 58–64.

Sanastokeskuksen termipankki. 2014. "Ohjelmisto". Alkuperäinen lähde: Tietotekniikan termitalkoot, 05.12.2014. Viitattu 21.4.2016. Saatavissa: <http://www.tsk.fi/tepa/netmot.exe?Opt=256&dic=1&SearchWord=@40@40ID@3D@22IDgjpa2PjqokLMrf@2DwdNtzxg@22&UI=file>.

Schwaber, K. & Sutherland, J. 2013. The Scrum Guide – The Definitive Guide to Scrum: The Rules of the Game. Ohje. Viitattu: 30.04.2016. Saatavissa: <http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf>.

Sutcliffe, A. 2009. Designing for User Engagement: Aesthetic and Attractive User Interfaces Synthesis. Lectures on Human-Centered Informatics 2, 1, pp. 1–55.

Sy, D. 2007. Adapting Usability Investigations for Agile User-Centered Design. JUS, Journal of Usability Studies 2, 3, pp. 112–132.

Tieteen termipankki. 2014. Ympäristötieteet: laitteisto (tietok.) Viitattu: 15.2.2016. Saatavissa: [http://tieteentermipankki.fi/wiki/Ymp%C3%A4rist%C3%B6tieteet:laitteisto_\(tietok.\)](http://tieteentermipankki.fi/wiki/Ymp%C3%A4rist%C3%B6tieteet:laitteisto_(tietok.)).

Tilastokeskus. 2016. Laadullisen ja määrällisen tutkimuksen erot. Verkkosivu. Viitattu: 22.4.2016. Saatavissa: <https://www.stat.fi/virsta/tkeruu/01/07>.

Tutkimuseettinen neuvottelukunta. 2012. Hyvä tieteellinen käytäntö. Verkkosivu. Viitattu: 30.04.2016. Saatavissa: <http://www.tenk.fi/fi/htk-ohje/hyva-tieteellinen-kaytanto>.

Uusitalo, E., Komssi, M., Kauppinen, M. & Davis, A. 2008. Linking requirements and testing in practice. 2008 16th IEEE International Requirements Engineering Conference, Barcelona, Catalunya, Spain, 8–12 September, 2008. pp. 265–270.

Verner, J., Sampson, J. & Cerpa, N. 2008. What factors lead to software project failure? 2008 Second International Conference on Research Challenges in Information Science, Marrakech, Morocco, 3–6 June 2008. IEEE. pp. 71–80.

Vuori, M. 2010. Testauksen kehityskaaria. Luentokalvot. Viitattu: 21.4.2016. Saatavissa: http://www.mattivuori.net/julkaisuluettelo/liitteet/testauksen_kehityskaaria.pdf.

Walls, C. Unified system design: Software and hardware on the same page. Verkkosivu. Viitattu: 1.3.2016. Saatavissa: <http://embedded-computing.com/articles/unified-system-design-software-hardware-the-page>.

Wells, D. 2009. Extreme Programming: A gentle introduction. Internetsivusto. Viitattu: 02.05.2016. Saatavissa: <http://www.extremeprogramming.org>.

West, D. 2011. Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today. Forrester Research Inc. Raportti. Viitattu: 14.05.2016. Saatavissa: <http://www.storycology.com/uploads/1/1/4/9/11495720/water-scrum-fall.pdf>.

Womack, J. & Jones, D. 2003. *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. First edition. Free press NY. 396 p.

Zhang, Z., Arvela, M., Berki, E., Muhonen, M., Nummenmaa, J. & Poranen, T. 2010. Towards Lightweight Requirements Documentation. *Journal of Software Engineering & Applications* 3, 9, pp. 882–889.

LIITE A: TUTKIMUSSUOSTUMUS

Tutkimussuostumus

Osallistun Tampereen teknillisen yliopiston Ihmiskeskeisen teknologian yksikön järjestämään haastatteluun ____ / ____ / 2016.

Haastattelu on osa diplomityötutkimusta, jonka tavoitteena on selvittää sulautetun ohjelmistotuotannon käytäntöjä ja käyttäjäkokemustyötä.

- Annan suostumukseni siihen, että haastattelu tallennetaan äänitallentimelle.
- Haastattelua ei saa tallentaa äänitallentimelle.

Aineisto tulee ainoastaan tutkimushenkilökunnan käyttöön, eikä sitä luovuteta ulkopuolisille.

Kaikki antamasi tiedot ovat luottamuksellisia. Kaikkia tietoja tullaan käsittelemään nimettömästi.

Tutkimukseen osallistuja:

Tutkimushenkilökunta:

Osallistujan allekirjoitus

Edustajan allekirjoitus

Nimen selvennys

Nimen selvennys

Tutkimukseen osallistujan tunniste anonymisointia varten: _____

LIITE B: TAUSTATIETOKYSELY

TAUSTATIETOKYSELY

Taustatietokyselyn tiedot menevät anonymisoituna Tampereen teknillisellä yliopistolla tehtävään tutkimukseen sulautettujen järjestelmien kehityksestä. Haastattelun tuloksia ei raportoida suoraan yrityksille eikä kolmansille osapuolille.

Mikäli sinulla on kysyttävää haastatteluun liittyen, ota yhteyttä kontro.m.e@gmail.com

IKÄ _____

MIKÄ ON TYÖ- /TEHTÄVÄNIMIKKEESI

LYHYT SELOSTUS NYKYISESTÄ TYÖNKUVASTASI

KOULUTUSTAUSTASI

TYÖKOKEMUS TÄMÄN HETKISESTÄ YRITYKSESTÄ _____ VUOTTA

ALAN KOKONAISTYÖKOKEMUS _____ VUOTTA

KOODI HAASTATELTAVAN ANONYMISOINTIA VARTEN _____

LIITE C: HAASTATTELUKYSYMYKSET

HAASTATTELYKYSYMYKSET 30.1.2014

(Laitteistokehitysprojektin läpimenoajan lyhentäminen)

Mitä tarkoittaa laitteisto-projekti?

Kerro tämänhetkisen/tietyn projektin nykytilanne.

Miten projekti lähtee liikkeelle? Kerro lyhyesti projektin vaiheista.

Käytetäänkö ohjelmistojen valmistuksessa samaa prosessikehystä kuin laitteiston valmistuksessa?

Miten nämä prosessit linkittyvät toisiinsa?

Mikä vie aikaa projektin aikana? Missä odotellaan jotakin? Mikä syö aikaa?

Millaisia kehityskohtia näet toiminnassanne?

Mitä hyvää näet toiminnassanne?

Jos voisit parantaa jotain tämänhetkisiä käytäntöjämme, niin miten sen tekisit?

Onko käyttäjäkokemuksen tekeminen mukana projekteissanne? Jos on, niin miten se näkyy sinun työssäsi?

Missä vaiheessa UX otetaan mukaan?

Näetkö tarpeen käyttäjäkokemustyön tekemiselle?

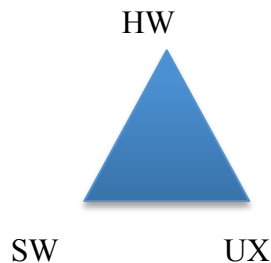
Miten itse osallistut käyttäjäkokemuksen tekemiseen?

Millaisia käyttäjäkokemukseen liittyviä teemoja osaisit nimetä? Anna esimerkki käytettävyyssyysmyksestä, joka on noussut esiin projekteissa.

HAASTATTELUKYSYMYKSET 22.03.2016

(Prosessimallin päivittäminen nykyistä vastaavaksi)

1. Kerro lyhyesti projektin vaiheista. Miten projekti lähtee liikkeelle jne.?
2. Millainen ohjelmistokehitys/projektimalli teillä on käytössä? Kuvaile mallia.
3. Käytetäänkö ohjelmistojen valmistuksessa samaa prosessikehystä kuin laitteiston valmistuksessa? Kerro eroavaisuuksista/yhteneväsyyksistä.
4. Miten laitteisto- ja ohjelmistoprosessit linkittyvät toisiinsa?
5. Kerro oheisen kuvan avulla omasta paikastasi projektissa, sekä kommunikoinnista ja työskentelystä muiden projekti-alueiden kanssa. Miten näet muiden osa-alueiden toiminnan?



6. Mikä vie liikaa aikaa projektin edetessä? Missä vaiheissa projektia odotellaan jotakin?
 7. Millaisia kehityskohtia näet toiminnassanne?
 8. Mitä hyvää näet toiminnassanne?
 9. Jos voisit parantaa jotain tämänhetkisiä käytäntöjämme, niin miten sen tekisit?
- ***
10. Miten käyttäjäkokemustyötä tehdään prosessissanne?
 11. Missä vaiheessa prosessia käyttäjäkokemustyö otetaan mukaan?
 12. Miten käyttäjäkokemustyö näkyy sinun työssäsi?
 13. Miten itse osallistut käyttäjäkokemuksen tekemiseen?
 14. Miksi käyttäjäkokemustyötä pitäisi/ei pitäisi tehdä?
 15. Millaisia käyttäjäkokemukseen liittyviä teemoja osaisit nimetä? Anna esimerkki käytettävyyks/käyttäjäkokemus-kysymyksestä, joka on noussut esiin projekteissa.
 16. Millaisia ongelmia käyttäjäkokemustyön tekemiseen liittyy?
 17. Mitä hyvää näet käyttäjäkokemustyöhön liittyvissä asioissa yrityksessänne?