

Laura Kivikangas & Markus Mulari

**SKAALAUTUVAT KETTERÄT MENETELMÄT: TA-
PAUSTUTKIMUS KIPUPISTEISTÄ**



JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2021

TIIVISTELMÄ

Kivikangas, Laura & Mulari, Markus

Skaalautuvat ketterät menetelmät: Tapaustutkimus kipupisteistä

Jyväskylä: Jyväskylän yliopisto, 2021, 109 s.

Tietojärjestelmätiede, pro gradu -tutkielma

Ohjaaja: Abrahamsson, Pekka

Tässä pro gradu -tutkielmassa selvitetään kirjallisuuskatsauksen avulla, millaisia menestystekijöitä ja haasteita skaalautuvien ketterien menetelmien käyttöönotossa ja käytössä esiintyy. Yleisimmin esiintyvistä menestystekijöistä ja haasteista tehdään kipupistemalli, jossa menestystekijät ja haasteet kuvataan kipupisteinä. Kipupistemalli sisältää myös ratkaisuehdotukset kipupisteiden selvittämiseksi. Kipupistemallissa on kahdeksan kipupistettä, jotka ovat muutostavastarinta, konseptien ja ketteryyden määrittely, tiimin koordinointi ja autonomia, johtaminen: tuen puute, uusien ja vanhojen toimintatapojen kohtaaminen, laadunvarmistus, ketterä osaaminen ja valmennus: osaamisen ja valmennuksen puute ja viestintä: huono viestintä.

Tutkielmassa toteutetaan tapaustutkimus, jonka avulla tutkimuksen kohteena olevan järjestelmäkehityshankkeen kehitysmenetelmää tutkitaan ja arvioidaan kipupistemallin pohjalta. Toteutimme hankkeessa seitsemän teemahaastattelua, joiden pohjalta kuvasimme ja vertasimme hankkeessa käytössä olevaa kehitysmenetelmää skaalautuviin ketteriin menetelmiin SAFe:en, LeSS:iin, DA:han ja Essenceen. Rooleista teimme vastaavan vertailukuvion. Löysimme hankkeen menetelmässä ja rooleissa vastaavuuksia kaikkiin vertailtuihin menetelmiin. Hankkeen menetelmässä oli myös kolme elementtiä: PO-aika, refaktointiviikko ja arkkitehtuurikokous ja yksi rooli: hankepällikkö, joita emme täysin vastaavana löytäneet vertailtavista menetelmistä.

Tarkastellessa haastatteluiden tuloksia kipupistemalliin saimme selville, että hankkeessa esiintyy kipupistemallin mukaisia kipupisteitä, joita on tunnistettu ja ainakin osittain ratkaistu ja kipupisteitä, joita ei ole ratkaistu. Hankkeessa esiintyvät kipupisteet, joita on tunnistettu ja ainakin osittain ratkaistu ovat tiimin koordinointi ja autonomia, ketterä osaaminen ja valmennus: osaamisen ja valmennuksen puute, viestintä: huono viestintä ja konseptien ja ketteryyden määrittely. Esiintyvät kipupisteet, joita ei ole ratkaistu ovat konseptien ja ketteryyden määrittely, uusien ja vanhojen toimintatapojen kohtaaminen ja laadunvarmistus. Kipupistemallissa esiintyviä muutostavastarintaa ja johtaminen: tuen puutteeseen viittaavia kipupisteitä ei havaittu.

Asiasanat: skaalautuva ketteruus, ketterät kehitysmenetelmät, haasteet, menestystekijät

ABSTRACT

Kivikangas, Laura & Mulari, Markus

Scaled agile frameworks: Case study on pain points

Jyväskylä: University of Jyväskylä, 2021, 109 pp.

Information Systems, Master's Thesis

Supervisor: Abrahamsson, Pekka

In this Master's Thesis we conduct a literature review on the factors affecting success and failure for the use and adoption of scaled agile frameworks. The most common success and failure factors are used to construct a pain point model in which success and failure factors are described as pain points. The pain point model also suggests solutions for overcoming pain points. The eight pain points in our model are change resistance, defining concepts and agility, team's coordination and autonomy, management: lack of support from management, clash of old and the new ways of working, quality management, agile skills and training: lack of skills and training and communication: bad communication.

We also conduct a case study in which a software project development method is evaluated and compared to our pain point model. We interviewed seven people working on the same development project. From these theme interviews we described and compared the methods used in the project to scaled agile methods SAFe, LeSS, DA, and Essence. We did an equivalent comparison to the roles used in the project. We discovered corresponding methods and roles to all examined methods. The project's development method also had three elements: PO time slot, refactoring week, architecture meeting, and role: project chief. These three elements and one role were not found as such from the compared methods.

Comparing the results from the interviews to the pain point model we discovered that the project holds identified and partly resolved pain points and pain points that are unresolved. Identified and partly resolved pain points in the project are team's coordination and autonomy, agile skills and training: lack of skills and training, communication: bad communication, and defining concepts and agility. Unresolved pain points are defining concepts and agility, clash of old and the new ways of working and quality management. Change resistance and management: lack of support from management pain points were not met.

Keywords: scaled agile, agile frameworks, challenges, success factors

KUVIOT

KUVIO 1 Scrum	14
KUVIO 2 Kanban.....	17
KUVIO 3 XP:n elinkaaren vaiheet.....	19
KUVIO 4 Metrokartta skaalautuvien ketterien menetelmien elementeistä.....	23
KUVIO 5 Metrokartta skaalautuvien ketterien menetelmien rooleista.....	24
KUVIO 6 Essencen alfat.....	37
KUVIO 7 Skaalautuvien ketterien menetelmien omaksuminen	44
KUVIO 8 Kipupistemalli	51
KUVIO 10 Tutkimusprosessi	58
KUVIO 11 Tutkimusmalli.....	58

TAULUKOT

Taulukko 1 Skaalautuvien menetelmien vertailua	25
Taulukko 2 Essencen elementtejä.....	33
Taulukko 3 Skaalautuvien ketterien kehitysmenetelmien vertailua	40
Taulukko 4 Haastatteluiden tiedot	61
Taulukko 5 Havaintojen suhde kirjallisuuteen	85
Taulukko 6 Käytännön ehdotukset.....	89

KUVAT

Kuva 1 Essencen alfakortti	34
Kuva 2 Essencen työntuloskortti.....	35
Kuva 3 Essencen aktiviteetikortti.....	35
Kuva 4 Essence kielen kompetenssikortti.....	36
Kuva 5 Essencen käytänteet.....	36

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KUVIOT JA TAULUKOT

1	JOHDANTO.....	7
1.1	Motivaatio.....	7
1.2	Tutkimuskysymykset.....	8
1.3	Työn rajaus.....	9
1.4	Työn rakenne.....	9
2	KETTERÄT KEHITYSMENETELMÄT.....	11
2.1	Perinteiset ketterät kehitysmenetelmät.....	11
2.1.1	Scrum.....	12
2.1.2	Lean.....	15
2.1.3	Kanban.....	17
2.1.4	Extreme Programming XP.....	18
2.2	Skaalautuvat ketterät kehitysmenetelmät.....	20
2.2.1	SAFe.....	26
2.2.2	LeSS ja LeSS Huge.....	28
2.2.3	DA.....	29
2.3	Essence.....	32
2.3.1	Essencen kieli ja ydin.....	32
2.3.2	Skaalautuva essence.....	38
2.3.3	Essencen nykytila.....	39
2.4	Skaalautuvien ketterien menetelmien haasteet ja menestystekijät	40
2.4.1	Menestystekijät.....	40
2.4.2	Haasteet.....	43
3	KIPUPISTEMALLI.....	49
3.1	Muutosvastarinta.....	51
3.2	Konseptien ja ketteryyden määrittely.....	52
3.3	Tiimin koordinointi ja autonomia.....	52
3.4	Johtaminen: tuen puute.....	53
3.5	Uusien ja vanhojen toimintatapojen kohtaaminen.....	54
3.6	Laadunvarmistus.....	54
3.7	Ketterä osaaminen ja valmennus: osaamisen ja valmennuksen puute	55
3.8	Viestintä: huono viestintä.....	55
4	TUTKIMUSMALLI- JA MENETELMÄ.....	57
4.1	Tutkimusmenetelmä ja datan keruu.....	59
4.2	Osallistujat.....	60
4.3	Datan analysointi.....	61
4.4	Tutkimuksen validiteetti ja luotettavuus.....	62

5	TULOKSET.....	63
5.1	Hankkeen ketterät kehitysmenetelmät.....	63
5.1.1	Hankkeessa esiintyviä ketterien kehitysmenetelmien piirteitä.....	64
5.1.2	Hankkeen erityispiirteitä	68
5.1.3	Hankkeen roolit.....	69
5.2	Kipupisteet.....	74
5.2.1	Muutosvastarinta	74
5.2.2	Konseptien ja ketteryyden määrittely	74
5.2.3	Tiimin koordinointi ja autonomia.....	76
5.2.4	Johtaminen: tuen puute	77
5.2.5	Uusien ja vanhojen toimintatapojen kohtaaminen.....	77
5.2.6	Laadunvarmistus.....	78
5.2.7	Ketterä osaaminen ja valmennus	79
5.2.8	Viestintä: huono viestintä	80
6	KESKUSTELU.....	84
6.1	Teoreettiset implikaatiot.....	84
6.2	Käytännön implikaatiot	88
7	SÄRKYLÄÄKETTÄ KIPUPISTEISIIN	92
7.1	Tutkimuskysymyksiin vastaaminen.....	92
7.2	Tutkimuksen luotettavuus ja rajoitukset.....	94
7.3	Jatkotutkimusaiheet.....	94
	LÄHTEET	95
	LIITE 1 ESSENTIAL SAFE.....	100
	LIITE 2 FULL SAFE VIITEKEHYS	101
	LIITE 3 LESS VIITEKEHYS	102
	LIITE 4 LESS HUGE VIITEKEHYS	103
	LIITE 5 DISCIPLINED AGILE (DA)	104
	LIITE 6 METROKARTTA KEHITYSMENETELMISTÄ.....	105
	LIITE 7 METROKARTTA ROOLEISTA	106
	LIITE 8 KIPUPISTEMALLI	107
	LIITE 9 METROKARTTA HANKKEEN MENETELMISTÄ	108
	LIITE 10 METROKARTTA HANKKEEN ROOLEISTA	109

1 JOHDANTO

Skaalautuvat ketterät kehitysmenetelmät, kuten SAFe, LeSS ja DA, lupaavat perinteisistä ketteristä menetelmistä, kuten scrum ja extreme programming, tunnettua tehokkuutta ja ketteryyttä suuremmalle organisaatiolle. Siksi ne ovat entistä houkuttelevampia vaihtoehtoja organisaatiolle, jotka harkitsevat ketteryyden hyödyntämistä koko organisaation laajuisesti. Kuten myös perinteisissä ketterissä kehitysmenetelmissä, myös skaalatuissa ketterissä kehitysmenetelmissä on omat ongelmansa. Ongelmana skaalatuissa ketterissä kehitysmenetelmissä on perinteisten ketterien kehitysmenetelmien lisäksi myös skaalautumisesta johtuvat ongelmat.

Tässä pro gradu -tutkielmassa selvitetään, mitkä ovat ketterät kehitysmenetelmät ja skaalautuvat ketterät kehitysmenetelmät, avataan skaalautuvan ketterien kehitysmenetelmien haasteita ja menestystekijöitä käsittelevää tieteellistä kirjallisuutta, luodaan löydösten avulla kipupistemalli, joka tulisi ottaa huomioon skaalautuvan ketterän kehitysmenetelmän käyttöönotossa ja käytössä. Lopuksi toteutetaan tapaustutkimus, jonka tuloksia arvioidaan kipupistemallin kautta.

1.1 Motivaatio

Ohjelmistokehityksessä suositaan tänä päivänä ketteriä kehitysmenetelmiä. Perinteiset ketterät menetelmät on suunniteltu pieniä kehitystiimejä varten. Tämän vuoksi niiden skaalaaminen suuriin projekteihin tuo haasteita, sillä suuremmat projektit tarvitsevat enemmän koordinaointia. (Dikert, Paasivaara & Lassenius, 2016.) Ketterän ohjelmistokehityksen piirteitä ovat inkrementaalisuus, yhteistoiminnallisuus, suorasukaisuus ja mukautuvuus. Ketterässä ohjelmistokehityksessä tehdään pieniä julkaisuja usein, asiakkaat ja kehittäjät tekevät tiiviisti yhteistyötä, menetelmä on helposti opittavissa, muokattavissa ja dokumentoitavissa ja viime hetken muutosten teko on mahdollista. (Abrahamsen, Salo, Ronkainen & Warsta, 2002.)

Skaalautuvat ketterät viitekehukset, kuten Scaled Agile Framework (SAFe), Large-Scale Scrum (LeSS) ja Disciplined Agile (DA) pyrkivät vastaamaan suurten projektien, globaalisti jakautuneiden tiimien ja kriittisten järjestelmien aiheuttamiin haasteisiin ohjelmistokehityksessä. Suurten organisaatioiden keskuudessa SAFe on suosituin skaalautuva ketterä menetelmä ja käytössä etenkin globaalisti hajautuneissa organisaatioissa (Putta, Paasivaara & Lassenius, 2018). Ketterien kehitysmenetelmien skaalaaminen suuriin projekteihin on vaikeaa eikä todistetusti toimivia malleja ole vielä esitetty. Pelkkien skaalautuvien ketterien viitekehysten käytäntöjen käyttöönotto ei tee organisaatiosta ketterää, vaan se vaatii kulttuurin ja ajattelutavan muutosta organisaation eri tasoilla. (Ebert & Paasivaara, 2017.) Skaalautuvien ketterien viitekehysten tutkimus on vielä vajavaista niiden omaksumisen, käytön, tehokkuuden ja haasteiden osalta (Conboy & Carroll, 2019).

Ketterät kehitysmenetelmät ovat olleet tutkimuksen kiinnostuksen kohteena ketterän manifestin (2001) julkaisusta lähtien. Ketterät kehitysmenetelmät tutkimuskohteena on keskittynyt kehitysmenetelmien käyttöönottoon ja omaksumiseen, menetelmien ihmisslähtöisiin ja sosiaalisiin näkökulmiin, kehitysmenetelmien käsitysten ymmärtämiseen sekä kehitysmenetelmien vertailuun (Dybå & Dingsøy, 2009). Ketterät kehitysmenetelmät tutkimuskohteena kaipaavat lisää tutkimusta ketteryyden määrittelystä: mitä ketteryydellä tarkoitetaan ja mikä lasketaan ketteryydeksi. Perinteiset ketterät menetelmät myös ovat usein soveltuvia pienille, paikkariippuvaisille tiimeille, joten ketterien menetelmien soveltuvuutta ja räätälöintiä laajempaan kontekstiin kaipaa lisää tutkimusta. Tutkimus keskittyy myös usein ketterien menetelmien käyttöönottoon, eikä käyttöönoton jälkeistä tutkimusta tehdä. (Abrahamsson, Conboy & Wang, 2009.)

Skaalautuvan ketterän kehitysmenetelmän kiinnostavina tutkimusaiheina ovat viime vuosina olleet mm. tiimien välinen koordinointi skaalautuvissa ketterissä kehitysmenetelmissä sekä skaalautuvan ketterän kehitysmenetelmän käyttöönotto ja organisaatiomuutos. Myös tiedonjakaminen skaalautuvissa kehitysmenetelmissä sekä liiketoiminnan ketteruus ovat olleet nousseita tutkimusaiheita. (Moe & Dingsøy, 2017.) Tässä tutkielmassa keskitymme ketteryyden skaalautumiseen, joka on myös yksi Moe ja Dingsøy (2017) julkaisussa esitetyistä tutkimusagendoista. Ketteryyden skaalaamisella etsitään vastauksia kysymyksiin, kuten mitä rooleja tarvitaan ketteryyden skaalaamiseen, mitkä ovat ketteryyden skaalaamisen syyt sekä mitä pitää tehdä ennen ketteryyden skaalaamista. Tässä tutkielmassa selvitetään skaalautumisen onnistumisia ja sudenkuoppia, joiden avulla luodaan erityinen kipupistemalli.

1.2 Tutkimuskysymykset

Tässä tutkielmassa selvitetään skaalautuvan ketterän kehittämisen kipupisteitä. Tutkimusongelma on skaalautuvien ketterien kehitysmenetelmien kipupisteiden ratkaisu. Tutkimuskysymykset ovat:

1. Millaisia rooleja ja menetelmiä skaalautuvaan ketterään kehittämiseen kuuluu?
2. Millaisia kipupisteitä tiimit kohtaavat skaalautuvassa ketterässä kehittämisessä?
3. Millaisilla keinoilla skaalautuvien ketterien kehitysmenetelmien kipupisteitä voidaan ratkaista?

Ensimmäiseen ja toiseen tutkimuskysymykseen vastataan kirjallisuuskatsauksen avulla. Kirjallisuuskatsauksessa tutustutaan yleisimpiin ja käytetyimpiin skaalautuviin ketteriin kehitysmenetelmiin. Toiseen tutkimuskysymykseen vastataan selvittämällä yleisimmät menestystekijät sekä haasteet skaalautuvan ketterän kehitysmenetelmän käyttöönotossa sekä käytössä. Kolmanteen tutkimuskysymykseen vastataan toteuttamalla tapaustutkimus skaalautuvaa ketterää kehitysmenetelmää käyttävässä organisaatiossa. Tapaustutkimuksen avulla selvitetään organisaatiossa esiintyvät kipupisteet, jotka ratkaistaan tutkielmassa myöhemmin esitetyn kipupistemallin avulla.

1.3 Työn raja

Tutkielman aihe tulee rajata oleellisesti tutkimukselle sopivalla tavalla. Tässä tutkielmassa tarkastelun ulkopuolelle jätetään perinteiset kehitysmenetelmät, joita ovat esimerkiksi vesiputousmalli ja siitä kehitetty spiraalimalli (Boehm, 1988). Tutkimuksessa toteutetaan kirjallisuuskatsaus ketteristä kehitysmenetelmistä, koska usein skaalautuvat ketterät kehitysmenetelmät ovat johdettuja tai käyttävät samanlaisia menetelmiä kuin perinteiset ketterät kehitysmenetelmät. Tässä tutkielmassa ei kuitenkaan avata perinteisten ketterien kehitysmenetelmien menestystekijöitä tai haasteita, koska tutkielmassa keskitytään skaalautuviin kehitysmenetelmiin.

Tässä tutkielmassa keskitymme ketteryyden skaalautumiseen, joka on yksi Moe ja Dingsøyr (2017) julkaisussa esitetyistä tutkimussuuntauksista. Tutkielmassa käsitellään käytetyimmät ja tunnetuimmat ketterät kehitysmenetelmät, joita ovat scrum, xp ja kanban. Skaalatuista ketteristä kehitysmenetelmistä käsitellään SAFe, LeSS ja DA. Myös lean-ajatusmalli huomioidaan tutkielmassa, koska useat ketterät kehitysmenetelmät hyödyntävät leanin periaatteita. Ketterien kehitysmenetelmien lisäksi tutkielmassa arvioidaan Ivar Jacobsonin Essencen sopivuutta erilaisten kehitysmenetelmien tarkasteluun, arviointiin ja mukauttamiseen.

1.4 Työn rakenne

Tutkielmassa toteutetaan ensimmäisenä kirjallisuuskatsaus perinteisiin ketteriin kehitysmenetelmiin, joka alkaa kappaleesta numero kaksi. Ketterien kehi-

tysmenetelmien käsitteleminen nähdään oleellisena tälle tutkielmalle, sillä skaalautuvat ketterät kehitysmenetelmät pohjautuvat osaltaan perinteisiin ketteriin kehitysmenetelmiin. Erityisesti scrumin erilaiset roolit, artefaktit ja menetelmät esiintyvät useassa skaalatussa ketterässä kehitysmenetelmässä, joten roolien, artefaktien ja menetelmien avaaminen kirjallisuuskatsauksessa on oleellista. Kirjallisuuskatsauksessa esitetään myös skaalautuvat ketterät kehitysmenetelmät SAFe, LeSS sekä DA. Kappaleessa tehdään myös vertailua skaalautuvien ketterien kehitysmenetelmien kesken ja vertaillaan kehitysmenetelmissä esiintyviä rooleja ja menetelmiä.

Skaalautuvien menetelmien jälkeen kappaleessa 2.3 esitetään Ivar Jacobsonin Essence. Essence yhdistää yleisesti hyväksytyjä ja käytettyjä parhaita käytänteitä yleiseen ohjelmistotuotannon työkalupakkiin, ja hyödyntää näiden esittämiseen Essencen omaa kieltä. Essencen filosofian taustalla on ajatus hyödyntää parhaita ketterän kehittämisen käytänteitä sitomatta organisaatiota tiettyyn ketterään kehitysmenetelmään. Tässä tutkielmassa avataan Essencen ydin sekä kieli, sekä vertaillaan Essencen rooleja ja menetelmiä skaalautuviin ketteriin kehitysmenetelmiin. Myöhemmin arvioimme Essencen kypsyttä ja sopivuutta, ja arvioimme myöhemmin, miten Essenceä olisi mahdollista hyödyntää kipupisteiden ratkaisemisessa.

Kirjallisuuskatsauksessa löytyneet skaalautuvien ketterien kehitysmenetelmien käyttöönoton ja käyttämisen menestystekijät ja haasteet esitetään kappaleessa 2.4. Kirjallisuuskatsaukseen valikoituneet artikkelit avataan ja niiden päälöydökset kerrotaan. Kappaleessa esitetään useimmiten esiintyneet haasteet sekä menestystekijät. Kolmannessa kappaleessa esitetään kipupistemalli, joka on luotu kirjallisuudessa esiintyneiden haasteiden ja menestystekijöiden avulla. Kipupistemallin avulla pyritään tunnistamaan ja ratkaisemaan organisaatiossa esiintyviä kipupisteitä. Tutkimusmalli, -prosessi ja -menetelmä esitetään neljännessä kappaleessa. Kappaleessa viisi avataan empirian tulokset, jonka jälkeen kappaleessa kuusi tehdään teorian ja käytännön johtopäätökset tulosten pohjalta. Viimeisessä kappaleessa arvioimme tutkielman tuloksia sekä pohdimme mahdollisia jatkotutkimusaiheita.

2 KETTERÄT KEHITYSMENETELMÄT

Tässä luvussa määritellään ketterä ja skaalautuva ketterä kehitys ja alaluvuissa esitellään perinteisiä ja skaalautuvia ketteriä kehitysmenetelmiä. Ketteriä kehitysmalleja on kehitetty useita ja ne eroavat toisistaan niiden prosessien, roolien ja velvollisuuksien, toteutuksen, omaksumisen ja kokemusten ja käyttöönoton ulottuvuuden puolesta. Tunnettuja perinteisiä ketteriä kehitysmenetelmiä, jotka tässä luvussa esitellään ovat muun muassa scrum, extreme programming eli XP sekä kanban. Edellä mainittujen menetelmien lisäksi tässä luvussa esitellään lean-ajatusmalli, jota myös usein hyödynnetään ketteryyden ohella. Skaalautuvista ketteristä menetelmistä esittelemme SAFe:n, LeSS:n ja DA:n. Lisäksi esittelemme ketterän ja skaalautuvan Essencen. Lopuksi tässä luvussa esittelemme skaalautuvien ketterien menetelmien haasteita ja menestystekijöitä.

2.1 Perinteiset ketterät kehitysmenetelmät

Ketterän ohjelmistokehityksen julistus on mullistanut ohjelmistoalan ja sen kaksitoista periaatetta edelleen kiteyttävät ketteryyden ytimen (Williams, 2012). Ketterillä kehitysmenetelmillä tarkoitetaan ohjelmistokehityksen lähestymistapaa, jossa ketteryys määrittyy kehitysmenetelmien työtapojen joustavuudella ja liikkuvuudella. Ketterä kehittäminen voidaan nähdä perinteisen suunnitelmalähtöisen ohjelmistokehityksen vesiputousmallin vastakohtana. Ketterät kehitysmenetelmät perustuvat vuonna 2001 esitettyyn ketterän ohjelmistokehityksen julistukseen (The Agile manifesto, 2001). Yhteensä 17 ohjelmistokehittäjän suunnittelemassa julistuksessa kerrotaan yhteiset arvot ketterälle ohjelmistokehittämiselle. Julistuksessa esitetään seuraavat arvot:

- yksilöitä ja kanssakäymistä tulee arvostaa enemmän kuin menetelmiä ja työkaluja,
- toimivaa ohjelmistoa tulee arvostaa enemmän kuin kattavaa dokumentaatiota,

- asiakasyhteistyötä tule arvostaa enemmän kuin sopimusneuvotteluja ja
- vastaamista muutokseen tulee arvostaa enemmän kuin suunnitelmassa pitäytymisessä.

Nykyiset erilaiset ketterät kehitysmenetelmät perustuvat ketterän ohjelmistokehityksen julistukseen. Julistuksessa esitetyt arvot esiintyvät myös edelleen ketterissä kehitysmenetelmissä. Ohjelmistokehityksessä suositaan tänä päivänä ketteriä kehitysmenetelmiä, joista Suomessa käytetyimpiä ovat scrum, XP, AM eli Agile Modelling, FDD eli Feature-Driven Development sekä kanban. Tosin viime vuosina tieteellinen kiinnostus XP:tä kohtaan on laskenut. Sen sijaan tieteellinen tutkimus on kohdistunut erityisesti scrumiin, leaniin ja kanbaniin sekä DevOpsiin. (Rodríguez, Mäntylä, Oivo, Lwakatere, Seppänen & Kuvaaja, 2019) Organisaatioiden muutosta ketteriksi on tapahtunut Suomessa vuosien ajan ja muutos tapahtuu eri teollisuuden aloilla (Laanti, Fagerholm, Mikkonen & Männistö, 2019). Näistä menetelmistä scrum on kaikkien suosituin, muiden menetelmien jäädessä pienemmälle huomiolle. Ketterien menetelmien lisäksi lean-ajatusmalli on säilyttänyt tieteellistä kiinnostusta. (Rodríguez, Markkula, Oivo & Turula, 2012; Rodríguez ym. 2019.)

Ketterien kehitysmenetelmien tarkoituksena on yksinkertaisuus ja nopeus. Kehitystyössä keskitytään ensisijaisesti tärkeimpien tuotosten toteuttamiseen, tuotosten nopeaan toimitukseen, palautteen saamiseen sekä palautteeseen reagointiin. Ketterille kehitysmenetelmille ovat tyypillistä inkrementaalisuus, yhteistoiminnallisuus, suorasukaisuus sekä joustavuus. (Abrahamsson, Salo, Ronkainen & Warsta, 2002.) Lisäksi ketterien menetelmien käytössä on tärkeää, että niitä käytetään yhtäaikaista ja menetelmiä toisiinsa sekoittaen aina tarpeen mukaan. Ketterän menetelmän käyttö ei myöskään ole koskaan valmis tai täydellinen, vaan ne ovat työkaluja, joita sovelletaan tarpeiden mukaan. (Kniberg & Skarin, 2010.) Ongelmana ketterissä kehitysmenetelmissä ovat epäselvyydet siitä mitä tarkoitetaan ketterällä ja ketteryydellä. Ihmisillä on erilaisia näkökulmia siitä mitä ketteryydellä tarkoitetaan ja miten sitä tulisi hyödyntää. Esimerkiksi joku voi kokea ketteryyden osana tiimien välistä työskentelyä, kun taas toinen näkee ketteryyden organisaation leikkaavana ajatusmallina. (Laanti, Similä & Abrahamsson, 2013.)

2.1.1 Scrum

Scrum on useista tieteenaloista muotoutunut projektijohtamiseen keskittyvä ketterä kehitysmenetelmä. Scrumin historia juurtuu vuoteen 1986, kun johtamisen ammattilaiset Hirotaka Takeuchi ja Ikujiro Nonaka viittasivat julkaisussaan "The New New Product Development Game" (1986) rugbyssä tunnettuun muodostelmaan scrum. Artikkelissa termillä painotetaan yhteistyön merkitykseen projektien onnistumisessa. Scrum sai ensimmäisen muotonsa kehitysmenetelmänä vuonna 1995, kun yhdysvaltalaiset ohjelmistokehittäjät Jeff Sutherland ja Ken Schwaber julkaisivat ensimmäisen artikkelin liittyen scrumiin. Myöhemmin vuonna 2002 Ken Schwaber julkaisi yhdessä Mike Beedlen kanssa ensimmäisen kirjansa "Agile Software Development with Scrum", jossa käsitel-

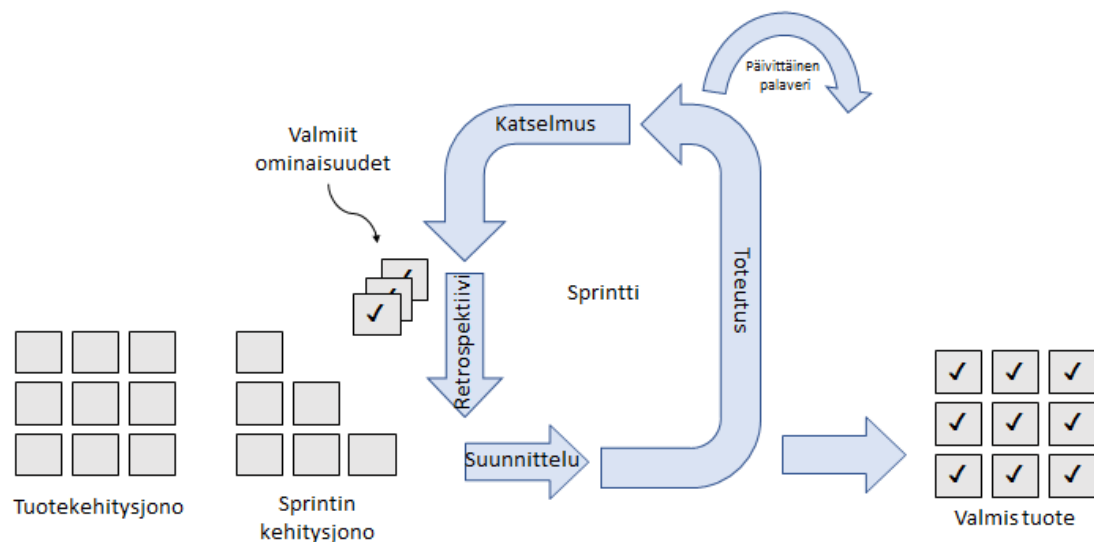
tiin tarkemmin scrumin käytänteet, roolit ja artefaktit. Edellisenä vuonna Sutherland sekä Schwaber olivat allekirjoittamassa ketterän ohjelmistokehityksen julistusta (Agile manifesto, 2001). Schwaber on julkaissut scrumiin liittyviä julkaisuja myöhemmin ja hän on Scrum-allianssin perustaja, joka mm. myöntää virallisia scrum master -sertifikaatteja. Myöhemmin Schwaber siirtyi scrum.org -verkkosivuston ylläpitäjäksi ja julkaisee tällä sivulla yhteisöä kiinnostavia artikkeleita sekä tarjoaa scrum-koulutusta.

Ketterä kehitysmalli scrum on yksi tunnetuimpia ja käytetyimpiä kehitysmalleja. Scrum jakaa ohjelmistoprojektit useampaan iteraatioon, jotka helpottavat projektin hallittavuutta. Projektin osittaminen ja jakaminen tapahtuu projektin päämäärän löytämisen jälkeen, joka on kehitettävä lopputuote. Lopputuotteen spesifikaation jälkeen toteutetaan projektin tuotekehitysjojo (eng. product backlog). Tuotekehitysjojosta valitaan tehtävät, jotka tullaan toteuttamaan seuraavan kuluvan sprintin aikana. Sprintin kehitysjojojen tehtävien pituus tulisi olla helposti jaettavissa tiimin kesken, ja yksittäisen tehtävän ihanepituus on 4–16 tuntia. Sprintin kehitysjojolista on näkyvillä jokaiselle tiimin jäsenelle. Jokainen tiimin jäsen saa näin yleiskuvan projektin tilasta. Nämä tuotekehitys sprintit ovat tyypillisesti joustavia ja ne eivät toimi lineaarisesti. Sprinttien lopussa kehitysjojosta valitut tehtävät tulisi olla valmiita toimitettaviksi ja käyttöön otettaviksi. (Schwaber, 2004.)

Scrum-kehitysmenettelmissä on rooleja, joiden avulla varmistetaan projektin koordinointi. Scrum kehitysmenettelmissä on kolme pääroolia, joita ovat tuoteomistaja, scrum master sekä kehittäjä. Tuoteomistaja vastaa asiakkaan tarpeiden ja vaatimusten ymmärtämisestä scrum-tiimin sisällä sekä ohjaa kehittämistä asiakkaan tarpeiden ja arvojen mukaiseen suuntaan. Tuoteomistajan tärkeimmät vastuut liittyvät tehtävien priorisointiin, kehitysjojojen ylläpitämiseen, julkaisusuunnitteluun, asiakkaan tarpeiden ymmärtämiseen ja scrum-tiimin ja asiakkaan välisenä linkkinä toimimiseen (Bass, Beecham, Razzak, Canna & Noll, 2018). Scrum masterin tehtäviin kuuluu scrum-tiimin johtaminen ja ohjaaminen oikeisiin tehtäviin ja arvoihin. Scrum master myös suorittaa perinteisiä työtehtäviä osana scrum-tiimiä. Scrum masterin tärkeimpänä tehtävänä on varmistaa, että tiimit toimivat scrumin mukaisesti. Scrum master varmistaa myös seremonioiden järjestämisestä ja toimii tiimin ja ulkopuolisten resurssien välisenä linkkinä. Sprinttien aikana järjestettävä sprinttisuunnittelu ja sprintin katselmointi ovat scrum masterin vastuulla. (Noll, Razzak, Bass & Beecham, 2017.) Kehittäjän rooli scrumissa voi olla hyvin monimuotoinen. Kehittäjän roolissa voi työkennellä esimerkiksi ohjelmistokehittäjät, suunnittelijat, kirjoittajat tai testaajat. Scrumissa kehittäjä nähdään monimuotoisena osana tiimiä, jolla on oikeat tiedot ja taidot yhteisen päämäärän saavuttamiseen. Scrumissa tiimiin kuuluvat tuoteomistaja, scrum master sekä muutama kehittäjä. Yleensä scrum-tiimien koko on muutamasta työntekijästä kymmeneen työntekijään. (Schwaber, 2004)

Scrumin sprintti on kehitysjakso, jonka aikana toteutetaan ohjelmiston osia ja ominaisuuksia sprintin kehitysjojosta. Sprintin kesto on yleensä yksi kuukausi. Sprintti aloitetaan yleensä suunnittelukokouksella, jossa tuoteomistaja ja tiimi keskusteleval yhdessä mitä tulevan sprintin aikana toteutetaan. Tuo-

teomistajan roolilla on kertoa tiimille mitä lopputuotteen ominaisuuksia tarvitaan. Tiimi vastaa tuotteenomistajalle siitä mitä tuotteen mahdollisia toimivia osia on mahdollista toimittaa tulevan sprintin aikana. Suunnittelupalaverin jälkeen alkaa itse työskentely ja valittujen kehitysjonon tehtävien suorittaminen, joita havainnollistetaan kuviossa 1. Scrumiin olennaisesti kuuluvat scrum masterien pitämät päiväpalaverit, joiden tarkoituksena on kartoittaa työskentelyn eteneminen. Niin kutsutuissa päiväpalaverissa, joita myös dailyiksi kutsutaan, jokainen tiimi kokoontuu 15 minuutin pituiseen kokoukseen, jossa on tarkoituksena selvittää mitä on toteutettu viimeisen palaverin jälkeen, mitä aikoo toteuttaa kyseisen palaverin jälkeen sekä mitä mahdollisia ongelmia näkee mahdollisten velvoitteiden suorittamisessa. (Schwaber, 2004) Päiväpalaverien avulla mahdollistetaan työn koordinointi ja seuranta, sekä mahdollistetaan työn sujuvuus. Sprintin lopussa on sprinttikatselmus, jossa tiimi esittää tuotteenomistajalle, osakkeenomistajille ja muille sidosryhmille mitä ovat saaneet aikaan kuluneen sprintin aikana. Katselmuksen aikana esitetään tuotteen toiminnallisuuksia. Jokaisen iteraation päätteeksi scrum master pitää tiimillensä retrospektiivin, jonka tarkoituksena on keskustella menneestä sprintistä ja kehittää prosesseja ja toimintatapoja. (Schwaber, 2004; Kniberg & Skarin, 2010.) Scrumiin olennaisesti kuuluu uudelleen oppiminen ja scrummasterilla on rooli ohjata tiimiä toimimaan tehokkaammin (Schwaber, 2004).



KUVIO 1 Scrum Schwaberin (2004) mukaan

Scrumin erikoisuus on, että se antaa päätöksentekovaltuuden operatiiviselle tasolle. Scrumissa ajatuksena on, että tiimit työskentelevät itseohjautuvasti. Tiimit ja tiimin jäsenet suunnittelevat, aikatauluttavat ja tekevät päätöksiä itsenäisesti. (Moe, Dingsøyr & Dybå, 2010) Scrumissa johtaminen ja työn koordi-

nointi tapahtuu asiakkaan tarpeiden pohjalta, jossa vastuussa ovat tuoteomistaja ja scrum master yhdessä tiimin jäsenten kanssa.

Scrum on ketteristä kehitysmenetelmistä suosituin menetelmä (Rodríguez ym., 2012; Rodríguez, Mäntylä, Oivo, Lwakatere, Seppänen & Kuvaja, 2019). Sen suosiota perustellaan scrumin yksinkertaisuudella ja sopivuudella monelle eri toimialalle. Scrumin tilaa nykyisessä ohjelmistokehityksen kentässä on tutkittu paljon, ja se on edelleen suosituin ketterä kehitysmenetelmä (Sharma & Hasteer, 2016). Scrumista on johdettu suuremmille organisaatiolle tarkoitettu suuresti skaalautuva scrum (eng. Large-scale Scrum, LeSS).

2.1.2 Lean

Lean kehittäminen on tuotekehityksen ajatusmalli, joka keskittyy arvonluontiin asiakkaalle, hukkan eliminointiin, arvovirtojen optimointiin, ihmisten voimaantumiseen ja jatkuvaan kehittymiseen (Ebert, Abrahamsson & Oza, 2012). Lean-ajattelutapa on peräisin Toyotan tuotantjärjestelmästä, joka kehitettiin ajoneuvojen tuotantoon ja logistiikan kontrolloimiseen. Lean termiä käytettiin ensimmäisen kerran vuonna 1986 Massachusettsin teknillisen korkeakoulun tutkijoiden toimesta. Tutkijat selvittivät maailmanlaajuisesti autotuotantojen eroja, ja havaitsivat että japanilainen autotuotanto oli huomattavasti tehokkaampaa kuin amerikkalainen. Tutkijat havaitsivat, että Lean-ajatusmallilla tarkoitus on tehdä enemmän vähemmällä, juuri oikeaan aikaan ja oikeassa paikassa. Tarkoituksena on arvon maksimoiminen ja turhan työn vähentäminen. Lean-ajattelutavan yleistyessä myöhemmin sitä on hyödynnetty usealla eri toimialalla, kuten ohjelmistotuotannossa.

Lean-ajattelu ja -tuotantotapa ovat tuotekehityksen ajatusmalli, joka keskittyy arvon tuottoon. Yleisesti arvontuoton luominen asiakkaalle koetaan siten, että asiakas saa arvoa tuotteella, joka vähentää asiakkaan menoja. Lean-ajattelussa asiakkaalle luotua arvoa tehostetaan poistamalla hukkaa, ja lisäämällä ominaisuuksia, jotka asiakas kokee arvokkaina. (Hines, Holweg & Rich, 2004) Ohjelmistokehityksessä lean-ajattelu perustuu seuraavaan seitsemään periaatteeseen:

- optimoi arvovirrat,
- eliminoi hukka,
- rakenna laadukkaasti
- opi jatkuvasti,
- toimita nopeasti,
- osallista jokainen ja
- kehity jatkuvasti

Lean-ajattelu ei sisällä menetelmiä tai tapoja, joilla päästäisiin haluttuihin arvoihin. Sen vuoksi se on mahdollista ottaa käyttöön lähes jokaisella toimialalla. Leanin ympärille on kuitenkin vuosien saatossa kehittynyt parhaita käytänteitä, joilla voidaan päästä periaatteen arvoihin. Tällaisia käytänteitä ovat esimerkiksi automaatio, kanban, arvokartat, work-in-progress ja lead time. (Rodríguez ym. 2019)

Lean periaatteista arvovirtojen optimoinnilla tarkoitetaan asiakkaan ymmärtämistä siitä mitä asiakas haluaa ja millä tavalla. Asiakkaan ymmärtäminen ja heidän arvostusten selvittäminen ei ole yksinkertaista ja asiakkaalle pelkkä ohjelmisto ei tuo arvoa (Poppendieck & Cusumano, 2012). Arvo luodaan kokonaisuudesta, jossa ohjelmisto on osa sitä. Hukalla tarkoitetaan lean-ajattelussa tuotteen ominaisuuksia tai toiminnallisuuksia, joista asiakas ei saa lisäarvoa. Turhat ominaisuudet, kadonnut tieto, osittain tehty työ, vanhojen koodivikojen korjaaminen ovat esimerkkejä hukasta (Poddendieck & Cusumano 2012). Laadukkaalla rakentamisella tarkoitetaan kehitys- ja toimitusprosesseja, jotka on todettu tehokkaaksi ja luotettavaksi. Esimerkiksi pienempien ohjelmistokokonaisuuksien toimittaminen ja integroiminen suurempiin järjestelmiin on pidetty parhaana käytänteenä jo 1970-luvulta saakka. (Poddendieck, 2012) Kehittäminen on uuden tiedon luomista ja tiedon sijoittamista lopputuotteeseen. Jatkuvalla oppimisella jokainen kriittinen päätös tulee jättää viimeiseen mahdolliseen hetkeen, jotta kaikki mahdolliset vaihtoehdot on tutkittu. Jatkuva oppiminen voi tapahtua myös asiakkaan palautteiden kautta, jonka kautta tehdään mahdollisia muutoksia asiakkaalle toimitettavaan tuotteeseen. Lopputuotteiden nopealla toimituksella vähennetään virheiden mahdollisuutta ja yleisen laadun koetaan kasvavan. Leanin ajatukseen kuuluu jatkuvan toimituksen sisällyttäminen kehityskaareen (Poddendieck, 2012). Jokaisen organisaation jäsenen osallistaminen kehityskaareen on osa lean-ajattelua, jolla pyritään yksittäisen IT-osaston muodostuminen. Tämän tarkoituksena on siirtää IT-osaaminen osaksi koko liiketoimintamallia ja lisätä yhteistyötä organisaation sisällä eri toimijoiden kesken. Jokaiseen leanin perusajatukseen kuuluu prosessien ja periaatteiden jatkuva kehittäminen.

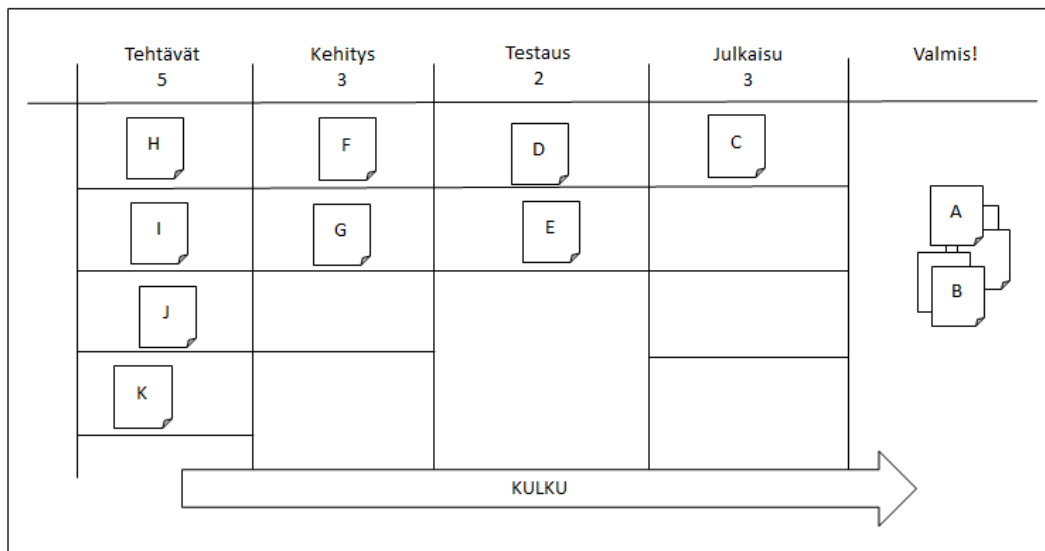
Ketterät kehitysmenetelmät sekä lean ajattelu ovat filosofialtaan hyvin samankaltaisia (Rodríguez ym., 2019). Ketteriä kehitysmenetelmät, kuten XP, Scrum ja Kanban, ovat kokonaisuus tapoja ja menetelmiä, jolla luodaan asiakkaalle hänen mieltymyksiensä mukainen lopullinen tuote. Kuten ketteryyden arvot, periaatteet ja käytänteet, myös lean sisältää nämä kolme elementtiä. Leanissa nämä elementit ovat lean konseptit, lean periaatteet ja lean käytänteet. (Wang, Conboy & Cawley, 2012.) Ketterät kehitysmenetelmät eivät yleensä ole, että kehitystiimi on vastuussa tuotteen tai projektin onnistumisesta, vaan vastuu on yleensä tuoteomistajalla tai asiakkaalla. (Poddendieck, 2012) Lean ohjelmistokehitys sen sijaan sisällyttää jokaisen organisaation jäsenen osaksi tuotteen arvon muodostumisen kehitystä. Lean ohjelmistokehityksessä ei ole erillisiä tuoteomistajia, vaan jokainen organisaation jäsen on vastuussa tuotteen onnistumisesta. Leanin koetaan olevan enemmän filosofinen ajattelutapa kuin valmis kehitysmenetelmä, jonka vuoksi lean-ajattelua käytetään vaihtelevasti (Wang, Conboy & Cawley, 2012). Lean-ajattelua käytetään vaihtelevasti organisaatorakenteen, toimialan ja liiketoimintamallien pohjalta. Lean on mahdollista yhdistää käytettäväksi ketterien kehitysmenetelmien kanssa tai sitä voidaan käyttää erillisesti ketterien menetelmien rinnalla. Leania voidaan hyödyntää ketterien kehityksen menetelmien kehittämisessä tai ketterien menetelmien käyttöönotossa. Mikäli ketteriä menetelmiä hyödynnetään yhdessä leanin kans-

sa, tulee ymmärtää mitkä ovat leanin ja ketterien menetelmien eroavaisuudet. (Rodríguez ym. 2019.)

2.1.3 Kanban

Kanban on ketterä menetelmä, jota myös lean-menetelmä hyödyntää. Kanbanin juuret ovat autojen valmistusprosessissa ja nykyään se on ohjelmistokehittämiseen levinnyt menetelmä. (Ahmad, Markkula & Oivo, 2013.) Ohjelmistokehityksessä kanban toimii keskeneräisen työn (WIP) seuraamisen ja rajoittamisen työkaluna sekä pohjana tehokkaalle ja flow-tyyppiselle ohjelmistokehitykselle.

Kanban on mielellään fyysisesti seinälle asetettava taulu, johon asetellaan kortteja. Kortit kuvastavat osaa työstä ja työn edetessä korttia siirretään taululla sarakkeelta toiselle. Kanbanin erikoisuus on se, että taulun jokaiselle sarakkeelle annetaan raja, kuinka paljon sarakkeella saa olla kortteja eli työtä. (Poppendieck & Cusumano, 2012.)



KUVIO 2 Kanban Knibergin ja Skarin (2010) mukaan

Kanban menetelmänä ei ole yhtä ohjaava kuin scrum, sillä se tarjoaa vähemmän sääntöjä. Kanbanin voi jakaa kolmeen tehtävään. Ensimmäinen on työnkulun visualisointi, jossa työ jaetaan osiin, kirjoitetaan ylös ja asetetaan seinälle. Jokainen osa asetetaan nimetyn sarakkeen alle. Toinen tehtävä on asettaa rajoitukset, kuinka paljon keskeneräistä työtä (WIP) voi kullakin sarakkeella olla yhtä aikaa. Vaiheiden 1 ja 2 lopputulos voi näkyä esimerkiksi kuviossa 2 esitetyllä tavalla. Kolmantena tulee mitata, kuinka kauan keskimääräisesti kestää yhden työn osan tekemiseen. Ajasta tulisi tehdä mahdollisimman pieni ja ennalta arvattava. (Kniberg & Skarin, 2010.)

Tärkeimmät kanbanin tuomat edut ohjelmistokehityksessä ovat työn visualisointi, WIP:n rajoittaminen, työnkulun mittaaminen ja hallinnointi helpottuu, se tekee prosessikäytännöistä tarkkoja ja se tukee jatkuvaa kehitystä. Kanbanin

avulla kaikki sidosryhmät ja etenkin kehittäjät saavat paremman kokonaisku-
van koko prosessista, mikä auttaa tiimin välistä viestintää ja nostaa asiakasty-
tyväisyyttä. Kanbanin käyttöönotto yhdessä ketterien menetelmien kanssa voi
olla haasteellista, kuten myös organisaatiokulttuurin muuttaminen leanin mu-
kaisesti ja kanbanin käyttöönotto hajautetussa ympäristössä. Scrumbanissa
kanbanin ja scrumin käyttö yhdistyvät. Siinä scrum-tiimi visualisoi työtään
kanban-aulun avulla ja tarpeen mukaan tiimi käyttää joko sprinttejä tai jatku-
van kehityksen työkulkua. Scrumbanissa kuitenkin ylläpidetään joitain scru-
min elementtejä, kuten päivittäiset tapaamiset ja säännölliset tarkastelut.
(Rodríguez, Mäntylä, Oivo, Lwakatere, Seppänen & Kuvaja, 2019.)

2.1.4 Extreme Programming XP

Extreme programming eli XP on ketterä kehitysmenetelmä pienille ja keskiko-
koisille kehitystiimeille, jotka vastaavat alati muuttuviin tai epämääräisiin vaa-
timusmäärittelyihin. Kuten scrumissa, tällekin menetelmälle on tyypillistä alle
kymmenen työntekijän kokoiset kehitystiimit, ja tiimin jäsenet työskentelevät
tyypillisesti samassa sijainnissa. XP:n noudattamat arvot ovat jatkuva keskuste-
lu, yksinkertaisuus, nopea palaute ja rohkeus toimia ennakoivasti ongelmien
edessä. (Paulk, 2001.)

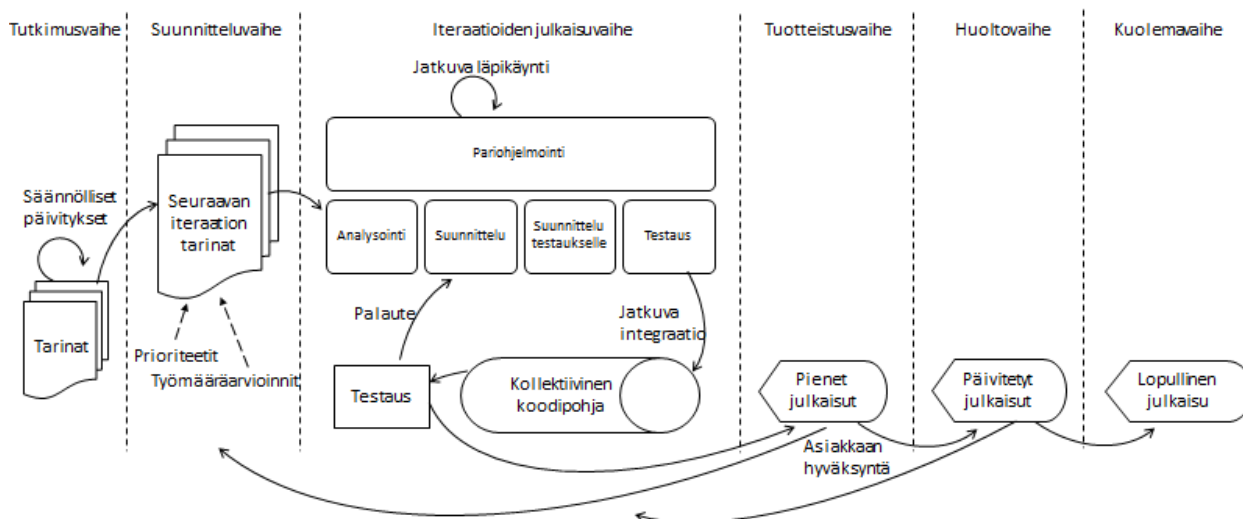
XP:ssä on seuraavat roolit: kehittäjä, asiakas, testaaja, jäljittäjä, valmentaja,
konsultti ja johtaja (Abrahamsson, Salo, Ronkainen & Warsta, 2002; Beck, 2000)
ja se koostuu ohjelmistokehityksen tavanomaisista osista, kuten suunnittelusta
ja analysoinnista. Perinteisestä kehityksestä poiketen XP tekee näitä pala kerral-
laan koko ohjelmiston kehittämisen ajan. Ohjelmiston kehityssyklissä asiakas
valitsee seuraavan julkaisun kehitettävät asiat valitsemalla ohjelmiston tär-
keimmät ominaisuudet eli tarinat. Tarinat ovat liiketoimintaorientoituneita, tes-
tattavia ja arvioitavia. Kehittäjät pilkkovat tarinat tehtäviksi, joilla jokaisella on
vastuukehittäjänsä. Kehittäjä tekee vastuutehtävästään testitapauksia, joilla
näytetään tehtävän olevan valmis. Yhden tehtävän tekeminen ja implementointi
saisi kestää yhdeltä kehittäjältä pari päivää.

Tehtävän vastuukehittäjän tulee ottaa työpari, jonka kanssa hän työskente-
lee yhdeltä koneelta. Mikäli parilla tulee kysyttävää, heidän tulee pitää lyhyt
kokous asiakkaan tai parhaiten osaavien toisten kehittäjien kanssa. Tarinoiden
julkaisujärjestys riippuu asiakkaan asettamasta tärkeysjärjestyksestä. Aluksi
julkaistaan pienin mahdollinen määrä tarinoita. Tämän julkaisun jälkeen voi-
daan julkaista loput. Jokaisen iteraation tavoitteena on saada julkaisuun testat-
tuja ja valmiita tarinoita. (Beck, 1999.)

Tutkiminen, suunnittelu, julkaisuiteraatiot, tuotteistus, huoltaminen ja
kuolema ovat XP:n elinkaaren vaiheet. Elinkaaren vaiheet on esitetty kuviossa 3.
Tutkimusvaihe kestää muutamasta viikosta muutamaaan kuukauteen. Tässä
ensimmäisessä vaiheessa asiakkaat kirjoittavat auki ohjelmistoon implementoi-
tavat ominaisuudet eli tarinat. Samalla kehittäjät tutustuvat käytössä oleviin
teknologioihin, työkaluihin ja työtapoihin. Suunnitteluvaiheessa asiakkaiden
tarinat priorisoidaan ja ensimmäisen julkaisun tarinat valitaan ja työmäärä- ja
aika-arvio annetaan. Tähän kuluu pari päivää.

Iteraatiovaihe sisältää monta 1-4 viikkoa kestävästä iteraatiosta ennen ensimmäistä julkaisua. Asiakkaalla on päätäntävalta, mitkä tarinat valitaan mihinkin iteraatioon. Julkaisuvaiheessa tehdään enemmän testausta ja järjestelmän suorituskyvyn testausta. Vaiheen aikana voi vielä löytyä muutoskohteita, joista on tehtävä päätös, sisällytetäänkö ne vielä samaan julkaisuun vai dokumentoidaan ne seuraavia implementointeja varten.

Huoltovaiheessa tiimin rakenteet saattavat muuttua ja uusia jäseniä tulla projektiin mukaan, sillä huoltovaiheen tehtävä on tukea asiakkaita, pitää järjestelmä pystyssä tuotannossa ja tuottaa uusia iteraatioita. Kun asiakkaalla ei enää ole uusia tarinoita implementoitavaksi, alkaa lähestyä kuolemavaihe. Tässä vaiheessa ei enää tehdä arkkitehtuuri-, muotoilu- tai koodimuutoksia. Vaihe voi koittaa myös, mikäli järjestelmä ei täytä tehtäväänsä tai siitä tulee liian kallis jatkokehitettäväksi. (Abrahamsson ym., 2002; Beck, 2000.)



KUVIO 3 XP:n elinkaaren vaiheet Abrahamssonin ym. (2002) mukaan

Paulk (2001) puolestaan kuvaa XP:n elinkaaren koostuvan neljästä olennaisesta toiminnasta, jotka ovat koodaus, testaus, kuuntelu ja suunnittelu. XP koostuu myös lukuisista eri toimintatavoista, joista yleisimpiä ovat kaksitoista perusosaa. Nämä ovat Abrahamsson ym. (2002) ja Paulk (2001) mukaan

- *suunnittelupeli*, joka käsittää asiakkaan ja kehittäjien tiiviin yhteistyön. Asiakas tarkastelee liiketoiminnan puolta ja tekee päätökset kehityskohteista sekä niiden prioriteeteista ja aikataulusta. Kehittäjät arvioivat teknistä toteutusta ja työmäärää.
- *pienet julkaisut*, tulisi pyrkiä julkaisemaan mahdollisimman nopeasti ja usein pieniä julkaisuja.
- *metafora*, metaforat selittävät, kuinka järjestelmän tulisi toimia. Kehittäjillä ja asiakkailla on yhteisymmärrys kielikuvista.

- *yksinkertainen suunnittelu*, suunnittelun tulee olla mahdollisimman yksinkertaista.
- *testaus*, yksikkötestit tehdään ennen koodausta ja testaus on jatkuvaa. Asiakkaat kirjoittavat toiminnalliset testit.
- *refaktorointi*, järjestelmää uudelleen järjestellään poistamalla toisintoja, parantamalla kommunikointia, yksinkertaistamalla ja lisäämällä joustavuutta.
- *pariohjelmointi*, kaksi ohjelmoijaa ohjelmoi samalta koneelta.
- *kollektiivinen omistajuus*, kuka tahansa voi muuttaa mitä tahansa osaa koodista milloin tahansa.
- *jatkuva integraatio*, uutta koodia integroidaan heti, kun se on valmista eli monta kertaa päivässä.
- *40-tuntiset työviikot*, tätä työaikaa ei saa ylittää kahta viikkoa putkeen.
- *läsänä oleva asiakas*, joka on täysipäiväisesti tiimin käytettävissä.
- *ohjelmointistandardit*, ohjelmoijat noudattavat yhteisiä sääntöjä, jotka painottavat kommunikointia läpi koodin.

Näiden lisäksi Abrahamsson ym. (2002) mainitsevat XP:n käytänteistä

- *avoin työtila*, joka on suuri ja pariohjelmoijat sijaitsevat huoneen keskellä.
- *vain sääntöjä*, tiimillä on omat sääntönsä, joita noudatetaan. Säännöt ovat kuitenkin muokattavissa tarpeen mukaan.

2.2 Skaalautuvat ketterät kehitysmenetelmät

Tässä luvussa tarkastellaan, kuinka laajasti skaalautuvat ketterät kehitysmenetelmät ovat saaneet alkunsa ja määritellään suuresti skaalautuva ketterä kehittäminen. Alaluvuissa esitellään yleisimpiä kehitysmenetelmiä, joita ovat Scaled Agile Framework (SAFe), Large-Scale Scrum (LeSS) ja Disciplined Agile (DA).

Agile manifeston (2001) sai ohjelmistokehittäjät omaksumaan ketterät kehitysmenetelmät hiljalleen, joista erityisesti scrum on noussut yhdeksi tunnetuimmista ja eniten käytetyistä kehitysmenetelmistä. Muut menetelmät, kuten XP ja lean-ajatusmalli ovat myös jääneet nykyaikaisen ohjelmistokehittäjän työkalupakkiin. (Dingsøyr & Moe, 2014.) Ketteryyden takana on idea pienten kehitystiimien hyödyntämisestä tekemällä korkealaatuista ohjelmistoa hyödyntämällä jatkuvan kehittymisen ja toimittamisen mallia. Ketterät menetelmät toimivat yleensä kaikista ketterämmin pienissä korkeintaan 50 henkilön organisaatioissa. Ongelmaksi onkin muodostunut ketterien kehitysmenetelmien omaksuminen suurempiin projekteihin. Kehitysmenetelmät, jotka skaalataan suurempaan projektiin tai organisaatioon, kutsutaan skaalautuvaksi ketteräksi kehitysmenetelmäksi. Niillä pyritään hakemaan samanlaisia hyötyjä kuin perinteisistä ketteristä menetelmistä, mutta useimmiten uusia ongelmia esiintyy. (Dingsøyr & Moe, 2014.)

Skaalautuva ketterä kehitysmenetelmä tarkoittaa ketterien kehitysmenetelmien skaalaamista koko organisaation tasolle hyödyntäen ketteriä arvoja. Skaalautuvan ketterän kehitysmenetelmän määritelmänä voidaan pitää esimerkiksi

- yli 50 kehittäjää TAI 0,5 miljoonaa riviä koodia TAI yli kolme sijaintia / kolme eri aikavyöhykettä.
- yli 50 henkilöä, yli 5 tiimiä, kehittävät samaa tuotetta käyttäen ketteriä kehitysmenetelmiä.
- ketteryyden hyödyntämistä koko organisaation tasolla.
- uusien seremonioiden ja tapahtumien hyödyntäminen koordinoinnissa, esimerkiksi Scrum of Scrums.
- useampi tiimi toimittamassa ohjelmiston osaa tai kokonaisuutta.

Skaalautuvan ketterän kehitysmenetelmän määritelmään on erilaisia vaihtoehtoja. Dingsøyr ja Moe (2014) konferenssijulkaisussaan määrittelevät neljä näkökulmaa suuresti skaalautuviin ketteriin kehitysmenetelmiin: arkkitehtuuri, tiimien välinen koordinointi, portfolion hallinta sekä skaalaus. Laanti (2014) vuorostaan esittää tutkimuksessaan kahdeksan erilaista näkökulmaa ketteryyden tarkasteluun. Yleisesti erilaisia ketteryyden määritelmiä yhdistää koko organisaatiota koskeva ketteryyden hyödyntäminen. Ketteryyttä halutaan yhdistää organisaation rakenteeseen, liiketoimintaan, työkuultuuriin, henkilöihin, viestintään ja työkaluihin. (Laanti, 2014.; Dingsøyr & Moe, 2014.)

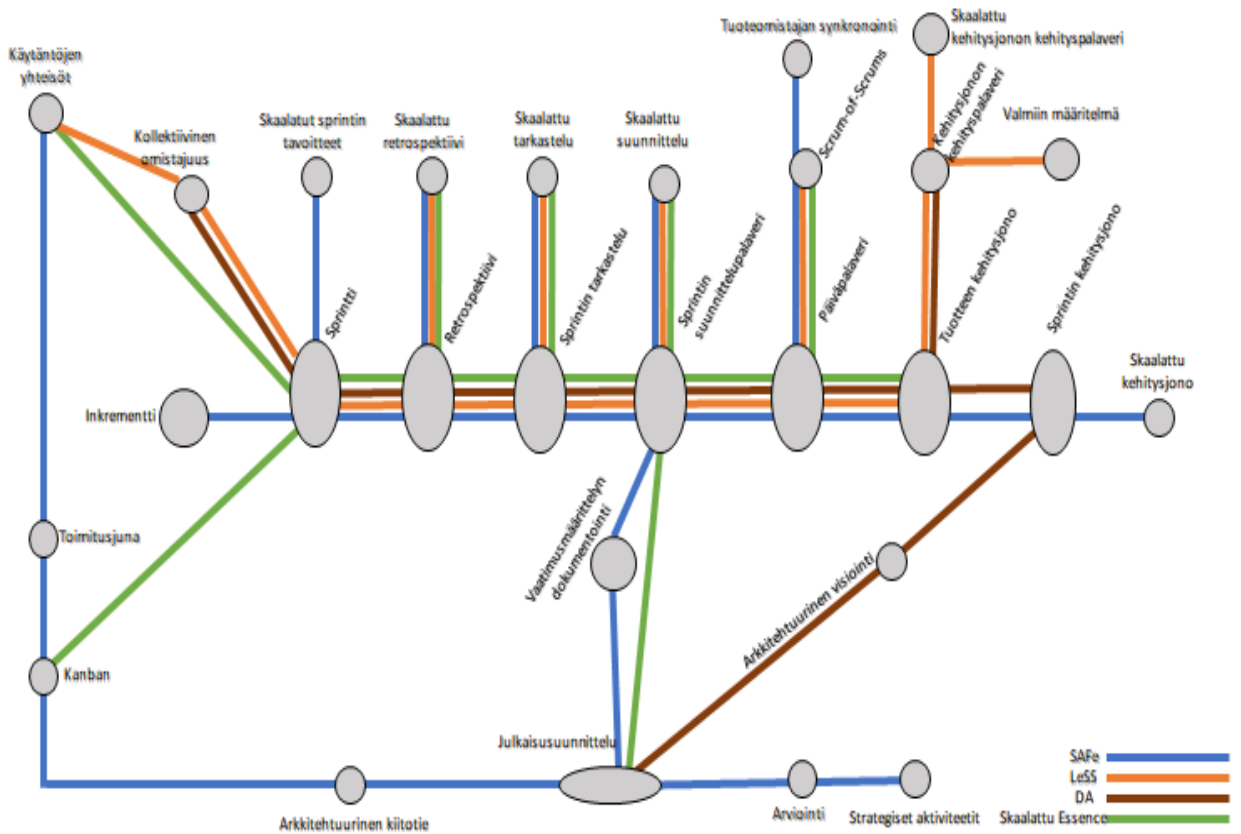
Skaalautuvat menetelmät, kuten SAFe, LeSS ja DA pyrkivät vastaamaan suurten projektien, globaalisti jakautuneiden tiimien ja kriittisten järjestelmien aiheuttamiin haasteisiin ohjelmistokehityksessä. Skaalautuvat ketterät kehitysmenetelmät eroavat perinteisistä ketteristä kehitysmenetelmistä niiden kehitystiimien henkilöstömäärän, projektin budjetin, koodirivien sekä projektin pituuden perusteella. Ketterien kehitysmenetelmien skaalaaminen suuriin projekteihin on vaikeaa eikä todistetusti toimivia malleja ole vielä esitetty. Pelkkien skaalautuvien ketterien menetelmien käytäntöjen käyttöönotto ei tee organisaatiosta ketterää, vaan se vaatii kulttuurin ja ajattelutavan muutosta organisaation eri tasoilla. Skaalautuvia ketteriä menetelmiä voidaan omaksua sellaisinaan tai ne voidaan räätälöidä organisaation kulttuurin, ajattelu- tai työskentelytavan mukaisesti. (Ebert & Paasivaara, 2017.)

Skaalautuvilla menetelmillä pyritään myös tavoittelemaan niin kutsuttujen perinteisten ketterien kehitysmenetelmille tyypillisiä ominaisuuksia. Suurien ketterien kehitysmenetelmien aikaiset omaksujat ovat kertoneet uusien menetelmien nostaneen merkittävästi tuottavuutta ja laatua. Organisaatioiden syytä lähteä käyttämään suuresti skaalattua ketterää kehitysmenetelmää on monisyinen. Taustalta yleensä löytyy halu kehittää tuottavuutta ja laatua, mutta myös maailmanlaajuiset trendit muuttavat organisaatioiden tapoja tehdä töitä. (Laanti, 2014.) Operatiivisen tuottavuuden ja laadun lisäksi ketteryyttä pyritään lisäämään vastaamalla asiakkaiden ja markkinoiden muutoksiin, työtyytyväisyydellä, jatkuvalla organisaatiotason oppimisella nopeasti muuttuvassa opera-

tiivisessä ympäristössä ja kilpailukykyisillä ja haluttavilla tuotteilla (Kettunen, Laanti, Fagerholm, Mikkonen & Männistö, 2019). Skaalautuvat ketterät kehitysmenetelmät käyttävät ketterien kehitysmenetelmien arvoja sekä lean-ajattelumallia. Organisaatiot voivat käyttää skaalautuvia kehitysmenetelmiä mm. liiketoimintamallin, johtamisen, prosessien, kompetenssin, ohjelmistokehityksen tai arkkitehtuurin muuttamiseen. (Ebert & Paasivaara, 2017.)

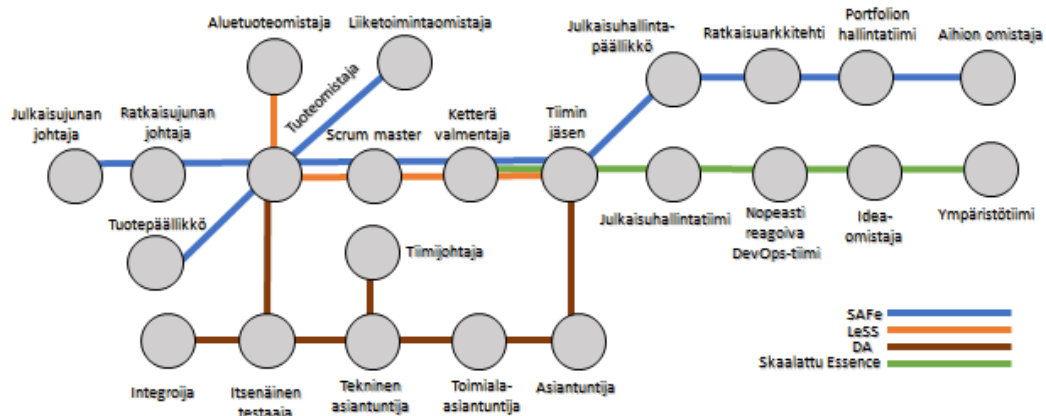
Skaalautuvien menetelmien käytöstä organisaatioissa on paljon empiiristä tutkimusta, joka on johtanut useiden menetelmien ja niiden muunnosten esiintymiseen kirjallisuudessa. Tutkimuksista selviää, että nykyinen kirjallisuus sisältää paljon teoreettista ja käytännön ongelmia. Esimerkiksi kirjallisuudessa korostuu kaupallisten skaalautuvien ketterien kehitysmenetelmien käytänteet niiden perustana olevien periaatteiden sijaan. SAFe, LeSS, Spotify, DA ja Scrum at Scale ovat useimmiten kirjallisuudessa esiintyvät suuresti skaalautuvat ketterät kehitysmenetelmät. Jokaisesta on esitetty kirjallisuuskatsauksen avulla niiden periaatteet, käytänteet, työkalut ja mittarit. SAFe:n periaatteita ovat sen ydinarvot, lean-ketterä ajattelu ja periaatteet. SAFe:n käytänteitä ovat ratkaisukomponenttien rakentaminen ketterien toimitusjunien avulla. Useimmissa suuresti skaalautuvien ketterien kehitysmenetelmien vertailututkimuksissa on keskitytty menetelmien käytäntöjen, työkalujen ja mittaamisen vertailuun. Harvat ovat keskittyneet menetelmien periaatteiden vertailuun. SAFe:n arvot ja periaatteet on esitetty mallin ydinarvoissa lean-ketterä-ajattelussa ja kymmenessä periaatteessa. LeSS:ssä on kymmenen periaatetta ja DA:ssa seitsemän. (Edison, Wang & Conboy, 2021.)

Lähes kaikki skaalautuvat ketterät kehitysmenetelmät toimivat tiettyjen ketteryyden menetelmien ja arvojen kautta. Scrumin menetelmät ovat yksi käytetyimmistä työkaluista myös skaalautuvissa ketterissä kehitysmenetelmissä (Diebold, Schmitt & Theobald, 2018). Kuviossa 4 ja liitteessä 6 esitetään SAFe, LeSS, DA sekä skaalatun Essencen elementit metrokartan muodossa. Kuvioista havaitaan, että scrumista yleensä eniten hyödynnettävät konseptit ovat dailyt ja sprintit, mutta myös muita scrumin osia hyödynnetään vaihtelevasti. Skaalautuvat ketterät kehitysmenetelmät eroavat toisistaan yleensä tietyille kehitysmenetelmälle tarkoitetun erityisen konvention, toimialan, joustavuuden tai roolien perusteella. (Theobald, Schmitt & Diebold, 2019; Diebold, Schmitt & Theobald, 2018.)



KUVIO 4 Metrokartta skaalautuvien ketterien menetelmien elementeistä Theobaldia, Schmittiä ja Dieboldia (2019) mukaillen

Skaalautuvissa ketterissä kehitysmenetelmissä käytettyjen menetelmät linjautuvat scrumin ympärille. Kuviossa 4 esitetyt scrumista tutut sprintit, sprintin suunnittelupalaverit, tuotekehitysjonot, päivittäiset palaverit sekä kuviossa 5 esitetyt roolit kuten scrum master ja tuoteomistaja ilmenevät kaikissa aikaisemmin mainituissa ja esitetyissä skaalautuvista menetelmissä. Scrumin roolien ja menetelmien hyödyntäminen skaalautuvissa ketterissä menetelmissä on enemmän sääntö kuin poikkeus. Myös muut skaalatut kehitysmenetelmät, joihin ei tässä tutkimuksessa tarkemmin tutustuta, kuten Nexus, Rage, Scrum At Scale sekä DSDM, hyödyntävät scrumin peruseriaatteita lähes identtisesti (Theobald, Schmitt & Diebold, 2019).



KUVIO 5 Metrokartta skaalautuvien ketterien menetelmien rooleista

Skaalautuvat ketterät kehitysmenetelmät poikkeavat myös toisistaan. SAFe viitekehyksessä suunnittelupalaverit on jaettu ohjelman inkrementteihin (PI), jossa on SAFe:lle tyypilliset innovointi ja suunnittelupalaverit. Inkrementtiajattelu on ainoastaan käytössä SAFe-viitekehyksessä. LeSS viitekehyksessä sprintin suunnittelu tapahtuu kahdessa eri palaverissa, joista ensimmäisessä fokus on kaikkien ketterien tiimien yleisestä ohjauksesta, ja toisessa fokus on tiimin sisäisten toimintatapojen selvittämisestä. Tuotekehitysjonoa hyödynnetään jatkaisessa kehitysmenetelmässä, mutta SAFe sekä DA sisältävät myös omat erityiset sprintin tuotekehitysjonot. SAFe:ssa kehitystiimejä kutsutaan ketteriksi toimitusjuniksi, joita ei konseptina esiinny muissa kehitysmenetelmissä (Diebold, Schmitt, Theobald, 2018). SAFe sisältää myös konseptit ohjelman kehitysjono, ohjelman kanban-taulu sekä ohjelman inkrementti. Käyttäjätarinat ja epicit ovat käytössä SAFessa sekä Scrum At Scale menetelmissä. Scrumin lisäksi SAFe hyödyntää omaa versiota XP:stä yhdessä kanbanin kanssa. LeSS ja DA hyödyntävät pääpiirteittäin scrumia. (Theobald, Schmitt & Diebold, 2019.) SAFe sekä DA hyödyntävät myös lean-ajatusmallin periaatteita (Ebert & Paasivara, 2017).

Kuviossa 5 ja liitteessä 7 esitetään skaalautuvissa ketterissä kehitysmenetelmissä esiintyneitä rooleja. Metrokartasta voi havaita, että scrumista tutut roolit, kuten scrum master ja tuoteomistaja ovat käytössä sellaisenaan SAFe:ssa ja LeSS:ssa. SAFe sisältää huomattavasti enemmän rooleja kuin LeSS. LeSS pitää sisällään vain viisi roolia. DA määrittelee hyvin tarkasti, millaisia rooleja on mahdollista hyödyntää. Kokonaisvaltainen SAFe pitää sisällään useita erilaisia rooleja. Roolien määrään vaikuttaa SAFen käyttöönnoton laajuus, mutta kuvassa

on huomioitu kokonaisvaltaisen SAFe:n roolit. Skaalautuva Essence pitää sisälleen scrumimaisia rooleja, mutta erottelee ne scrumista nimeämällä ne geneerisesti.

Skaalattujen kehitysmenetelmien vertailua on tehty taulukossa 1, pohjautuen Kalendan, Hynan ja Rossin (2018) julkaisemaan kirjallisuuskatsaukseen. SAFe on hyvin tarkasti dokumentoitu ja jäsenelty kokonaisuus, josta löytyy paljon informaatiota esimerkiksi Scaled Agile Inc. -verkkosivuilta sekä sertifioitujen konsulttien koulutuksista. SAFe:ssä määritellään hyvinkin tarkasti eri organisaatiotasojen vastuut sekä toimenpiteet, joka tekee SAFe:sta hieman muita menetelmiä vähemmän joustavamman ja monimutkaisemman. LeSS on SAFea vähemmän käytetty skaalautuva ketterä kehitysmenetelmä. LeSS on dokumentoitu The LeSS Company B.V toimesta LeSS:n omille verkkosivuille, joka tarjoaa dokumentaation lisäksi myös linkkejä tapahtumiin ja valmennuksiin. LeSS pohjautuu scrumiin, joten LeSS:n kokonaisuuden hahmottaminen on helpompaa kuin SAFe:n. LeSS soveltuu myös hieman pienemmille organisaatiotyypeille kuin SAFe. DA soveltuu taas usealle erilaiselle organisaatiotyypille, mutta sen ongelmana on sen monimutkaisuus, joka osaltaan voi johtaa sen matala käyttöasteeseen.

Taulukko 1 Skaalautuvien menetelmien vertailua mukailien Kalenda, Hyna ja Rossi (2018)

Näkökulma	SAFe	LeSS	DA
Tiimikoko	50–120 henkilöä julkaisujunissa	10 Scrum tiimiä, 7 jäsentä	200 työntekijää tai enemmän
Käytön laajuus	Korkea	Kohtalainen	Matala
Kypsyys	Korkea	Korkea	Kohtalainen
Monimutkaisuus	Korkea / kohtalainen	Kohtalainen / matala	Korkea
Organisaatiotyyppi	Perinteiset yritykset	Suuret yritykset	Monet organisaatiot ja yritykset

Suuresti skaalautuviin ketteriin kehitysmenetelmiin on tarjolla useita valmiita kokonaisuuksia. Menetelmiä ylläpitävät ja kehittävät yksityiset yritykset, kuten Scaled Agile Inc, Less Company tai IBM. Organisaatiot tarjoavat verkkosivuilla oppaita ja dokumentointia suuresti skaalautuvien menetelmien käyttöön, sekä maksullisia kursseja. Myös muut ketterät kehitysmenetelmien organisaatiot, kuten Scrum.org ja ScrumAlliance hyödyntävät maksullisia kursseja.

Organisaatioiden ottaessa käyttöön suuresti skaalautuvia ketteriä kehitysmenetelmiä valintaperusteena ovat yleensä kehitysmenetelmän tunnettuus tai konsulttien omat suositukset. Konsultit saattavat suosia tiettyä skaalautuvaa kehitysmenetelmää oman kokemuksen tai koulutuksen perusteella, mutta eivät osaa ottaa huomioon mahdollisesti paremmin sopivia kehitysmenetelmiä (Diebold, Schmitt & Theobald, 2018).

2.2.1 SAFe

Skaalattu ketterä kehitysmenetelmä Scaled Agile Framework, myöhemmin SAFe, on Dean Leffingwellin suunnittelema ketterä kehitysmenetelmä, joka on skaalattu ketterä kehitysmenetelmä suurempiin organisaatioihin. Se yhdistää parhaita piirteitä scrumista, XP:stä sekä kanbanista että lean-filosofiasta. (Ebert & Paasivaara, 2017.) SAFe on yksi tunnetuimmista ja käytetyimmistä skaalatuista ketteristä kehitysmenetelmistä (Laanti, 2014). SAFen viimeisin julkaistu versio on SAFe 5.0 ja sitä ylläpitää ja kehittää Scaled Agilen julkaisijat sekä yhteisö.

SAFesta on mahdollista hyödyntää pienempää kokonaisuutta (Liite 1) tai mahdollista ottaa koko SAFen kompetenssit käyttöön (Liite 2). Perusluonteinen SAFe sisältää tiimiin ja teknisen ketteryyden sekä ketterän tuotetoimituksen kompetenssit. Kokonaisvaltainen SAFe 5.0 rakentuu seitsemän kompetenssin ympärille, joita ovat

- Tiimi ja tekninen ketteruus
- Ketterä tuotetoimitus
- Yritysratkaisutoimitukset
- Lean portfolion hallinta
- Organisaationaalinen ketteruus
- Jatkuvan oppimisen kulttuuri
- Lean-ketterä johtajuus

SAFen pääasiallinen ydintarkoitus perustuu kolmen tärkeimmän kompetenssin ympärille; lean-ketterä johtajuuden, kehitystiimin ja teknisen ketteryyden ja ketterän toimituksen ympärille. Kehitystiimin ja teknisen ketteryyden kompetenssilla tarkoitetaan toimivaa, tiimien välisesti ja tehokkaasti työskentelevää ketterää joukkoa, joka rakentaa laadukkaita ratkaisuja asiakkailleen. SAFessa ketterään kehitystiimiin sisältyy 5–11 työntekijää, jotka toimittavat tuotteen palasia jokaisen tuotteen inkrementin avulla. Tuotteen inkrementti on 8–12 viikkoa kestävä kehitysjakso, jonka aikana toteutetaan tuotteen osa. Tiimien sisällä voidaan käyttää esimerkiksi Scrum, XP tai Kanbanista tuttuja työskentelytapoja (Kalenda, Hyna & Rossi, 2017). Esimerkiksi päivittäiset palaverit ovat osa SAFea. Iteraatioiden lopussa käydään retrospektiivit sekä tiimin että tuoteomistajan kanssa. Toimitusjunien tarkoituksena on toteuttaa mahdollisesti toimitettavia ohjelmiston osia ketterästi. SAFeen sisältyvät aloituspalaverit toimitusjunan päällikön sekä tuoteomistajan kanssa, jossa päätetään seuraavat toimitettavat tuotteen osat tai ominaisuudet. SAFe hyödyntää jatkuvan tutkimisen, toimituksen ja integraation mallia DevOpsia. (Scaled Agile, 2021.)

Julkaisujunan inkrementti on SAFen 8–12 viikkoa kestävä aikakokonaisuus, jonka aikana tuotetaan toimiva lopputuotteen osa. Tuoteosaan sisältyy yleensä neljä kehitysiteeraatiota sekä yksi innovointi- ja suunnitteluiteeraatio. Tuoteosa aloitetaan suunnittelupalaverilla, jossa suunnitellaan mitä seuraavan 8–12 viikon aikana toteutetaan ja mikä on kehitystiimien ja -junien riippuvaisuussuhde. Kehitysiteeraation aikana toteutetaan itse tuotetta, johon myös sisältyvät iteraatiosuunnittelu, iteraation retrospektiivit, tuotekehitysjonon kehitys-

palaveri, päivittäiset palaverit sekä iteraatiotarkastelu. Scrum of scrums, on iteraation scrum mastereiden, pitämä palaveri, jossa tiimien vastuuhenkilöt arvioivat tuoteosan toteutuksen etenemistä. Samantyyppinen palaveri on tuoteomistajan synkronointi, jossa tuoteomistajat, scrum masterit, julkaisujunan kehittäjät ja tuotehallinta tarkastavat tuotteen kehityssuunnat. (Scaled Agile, 2021.)

Lean-ketterä johtajuus on SAFen sisältämä organisaation johtamisen malli. Se sisältää ajatuksen lean-filosofiasta sekä ketteryyden manifestin perusteet. Organisaatioiden päätöksistä ja toteutuksesta vastaavat henkilöt ovat vastuussa ketterien menetelmien ja lean-filosofian hyödyntämisestä. Lean-ketterä johtajuuden tarkoituksena on vähentää hukkaa, organisoida työtä arvontuoton perusteella, poistaa motivaatiolle haitallisia esteitä, inspiroida ja motivoida työntekijöitä. Lean-ketterä johtajuus antaa ohjenuoran organisaatiolle, kuinka työskentelyyn ja töiden tekemiseen tulee suhtautua, ja kuinka organisaation tulee muuttaa tapojaan työskennellä. Ainoastaan organisaation lean-ketterä johtajilla on auktoriteettia muuttaa tapoja työskennellä. Lean-ketterä johtajuuden avulla työntekijät oletetaan toimivan paremmin organisaatiossa, jos heidän ajatuksiansa ohjataan ajattelemaan kasvamista ja jatkuvaa oppimista.

Yritysratkaisutoimitukset-kompetenssi määrittelee, kuinka lean-ketterä menetelmiä hyödynnetään ohjelmistokehityksen määrittelyssä, kehityksessä, käyttöönnotossa, käytössä sekä muutoksessa. Yritysratkaisutoimitukset sisältävät valmiit käytänteet, jotka auttavat linjaamaan organisaation käytännön ja ketteryyden yhteen, ja se on tarkoitettu todella suurien järjestelmien toteutuksiin. Tärkeimpinä elementteinä ovat Lean-ratkaisun rakentaminen, toimitusjunien ja alihankkijoiden koordinointi sekä julkaistujen järjestelmien jatkuva kehittäminen. Lean-ratkaisulla haetaan design thinking -ajattelumallia, jossa organisaatio on jatkuvasti etsimässä uusia mahdollisuuksia ja toteuttamassa näihin ratkaisuja. Työkaluja näihin ovat esimerkiksi markkinatutkimukset, asiakaskokemukskartat (eng. customer journey map) sekä muut liiketoiminnan kehittämisen mallit (eng. business capability models). Toimitusjunien ja alihankkijoiden koordinoinnilla ymmärretään organisaation ja sen muiden yhteistyökumppaneiden arvovirtoja. Jatkuvaan kehittämiseen sisältyy ajatus jatkuvasta toimittamisesta tuotantoputken avulla. (Scaled Agile Inc, 2021.)

Lean-portfoliohallinta kompetenssi linjaa strategian ja sen toteuttamisen lean-filosofian mukaiseksi. Se on johtamisen kompetenssi, joka auttaa tunnistamaan organisaation ja sen toimialan arvovirtoja. Kompetenssin tarkoitus on vähentää ylhäältä-alas johtamista, organisaation sisäisten funktionaalisten siilojen muodostumista, sekä muuttaa tapoja, kuinka projektien eri vaiheita tarkastellaan. (Scaled Agile Inc, 2021.)

Organisaationaalisen ketteryydellä tarkoitetaan organisaation kykyä ja mahdollisuuksia reagoida markkinoiden muuttuviin tarpeisiin. Kompetenssin tarkoitus on luoda havaittuihin tarpeisiin nopeasti arvoa, luoda uusia strategioita ja lähestymistapoja sekä tarvittaessa uudelleen organisoida tehtäviä. Organisaationaalinen ketteruus vaatii jokaiselta organisaation jäseneltä ymmärtämistä ketteryyden mallista. Organisaationaaliseen ketteryyteen sisältyy myös ajatus

omien ketterien prosessien ja toimien jatkuvasta kehittämisestä ja tehostamisesta. (Scaled Agile Inc, 2021.)

Jatkuvan oppimisen kulttuurin kompetenssilla sekä organisaatio että yksittäisen organisaation jäsenen pyrkii kehittämään ja kouluttamaan itseään. Tällä mahdollistetaan niin organisaation kuin organisaation jäsenen valmistautuminen mahdolliseen markkinan tarpeiden muutoksiin. Organisaation jäseniä kehoitetaan innovoimaan ja kokeilemaan uusia toimintatapoja ja -malleja, jotka voivat luoda tulevaisuudessa uusia arvoja. (Scaled Agile Inc, 2021.)

SAFe hyödyntää scrumin rooleja, joita ovat ketterät tiimit, scrum master ja tuoteomistajat. SAFe sisällyttää Scrumin perinteisten roolien lisäksi ketterän toimitusjunan päällikön, jotka johtavat toimitusjunia. Muita rooleja ovat myös järjestelmätiimit, julkaisujohtaja sekä portfolion hallintatiimi. Julkaisujohtaja varmistaa, että oikeat ominaisuudet julkaistaan ensin. Ratkaisuarkkitehti varmistaa yhdessä tuoteomistajan kanssa, että tuotejono sekä etenemissuunnitelma ovat ajan tasalla. Roolien määrä ja tarve vaihtelee myös SAFen käyttöönoton tasosta: roolien määrä luonnollisesti kasvaa, kun useampia SAFen kompetensseja otetaan käyttöön. (Scaled Agile Inc, 2021)

SAFe hyödyntää scrumin tapahtumia, mutta hyödyntää myös skaalattuja versioita näistä. Scrum tapahtumista hyödynnetään tiimitasolla päivittäisiä palaverieja, sprintin tarkastelua ja sprintin retrospektiivejä. SAFessa sprintin suunnittelupalaveri on korvattu SAFen omalla tuoteinkrementin suunnittelupalaverilla, joka on kahden päivän mittainen tapahtuma, jossa tiimit sopivat tulevista tuoteinkrementin tavoitteista, mahdollisista riippuvuuksista tiimien välillä, mahdollisista riskeistä ja kehitysjonoon tulevista tehtävistä. Tuoteinkrementin aikana on mahdollista pitää erilaisia synkronointipalavereita, joita ovat mm. toimitusjunan synkronointi, scrum of scrums sekä tuoteomistajan synkronointi. Näiden tapahtumien tarkoitus on pitää kehitystiimit, tuoteomistajat ja tuotehallinto ajan tasalla muutoksista. (Scaled Agile Inc, 2021.)

2.2.2 LeSS ja LeSS Huge

Suuresti skaalattu Scrum (eng. Large-Scale Scrum), lyhennettynä ja myöhemmin LeSS, on skaalattu ketterä kehitysmenetelmä, joka perustuu scrum ketterään kehitysmenetelmään. LeSS jakaa yhtenäisiä prosesseja ja toimia yhdessä scrumin kanssa, mutta ne kuitenkin eroavat toisistaan roolien sekä tuotosten perusteella. LeSS pyrkii lean-filosofiasta tyypilliseen asiakaskeskeisyyteen, ylimääräisten roolien vähentämiseen sekä kommunikoinnin tehostamiseen tiimien välillä. LeSSistä on olemassa kaksi erilaista versiota. LeSS (liite 3), joka on tarkoitettu enintään kahdeksan tiimin väliseen hallintaan, ja LeSS Huge (liite 4), joka on tarkoitettu tuhansien työntekijöiden ja tiimien organisointiin. (The LeSS Company B. V., 2021.)

Kuten scrumiin, myös LeSSiin sisältyvät sprintit. Jokainen sprintti alkaa ensimmäisellä sprintin suunnittelupalaverilla, jossa tuoteomistaja ja kehitystiimit perehtyvät lopputuotteen kehitysjonoon. Tuotekehitysjonosta valitaan tehtävät, jotka jaetaan tiimien kesken. Jaetut työtehtävät tulee toteuttaa sprintin aikana. Sprintin toisessa suunnittelupalaverissa kehitystiimit keskustelevat

strategiastaan, jolla aikovat suunnitellut työtehtävät toteuttavat. Kehitystiimien tulee keskustella keskenään, jotta yhteistyö tiimien välillä toimii. Jokainen tiimi pitää scrumin tapaan päivittäisen lyhyen palaverin dailyn, jossa tiimi organisoii päivän tehtävänsä. LeSSissä toisen tiimin jäsen voi vieraillla toisen tiimin dailyssä tiedonjaon tehostamiseksi.

Noin sprintin puolessa välissä pidetään tuotekehitysjonon kehityspalaveri yhdessä tiimien ja asiakkaan kanssa, jonka tarkoituksena on luoda yhteys tuotetta kehittävien työntekijöiden ja asiakkaan kanssa. Tärkeimmät aktiviteetit ovat suurten kokonaisuuksien osittaminen, osien selkiyttäminen ja työmäärän, arvon, riskein ynnä muun arviointi. Tuotekehitysjonon kehityspalaveri on virallinen tapahtuma, joka toteutetaan usean tiimin kesken, kokonaisvaltaisesti tai yksittäisen tiimin kesken. Usean tiimin kesken palaveri on näistä vaihtoehdoista suositeltavin. Siinä monta tiimiä ovat samassa huoneessa yhtä aikaa pitämässä tuotekehitysjonon kehityspalaveria. Tämä tehostaa oppimista ja tiimien välisiä koordinoitua.

Sprintin lopussa on katselmointi, jossa kehitystiimit ja asiakas tarkastelevat sprintin aikana tuotettua lopputuotteen osaa sekä päättävät seuraavat kehityssuunnat projektille. Jokainen tiimi pitää myös keskenään oman retrospektiivin, jossa tarkastellaan työskentelytapoja ja tarvittaessa muutetaan näitä. Kokonaisretrospektiivissä scrummaster, tuoteomistaja, hallinto ja kehitystiimit tutkivat onko tiimien toiminnalla esteitä tai ongelmia, jotka voivat haitata arvon tuottoa. (The LeSS Company B. V., 2021.)

Suuremmille organisaatioille skaalattu LeSS Huge sisältävät LeSS sprinttien ja palaverien lisäksi myös alueellisen tuotekehitysjonon sekä alueellisen tuoteomistajan. Jokaisen sprintin lopussa tulisi olla mahdollisesti toimitettava tuote tai tuotteen osa. LeSS kehitystiimejä kutsutaan ominaisuustiimiksi (eng. feature team). Jokaisella kehitystiimillä on oma roolinsa lopputuotteen ominaisuuden kehittämisessä ja tämän integroimista osaksi lopputuotetta. (The LeSS Company B. V., 2021.) Jokaisen ominaisuustiimin kehittäessä omaa osuuttaan ohjelmistosta, tiimit ovat vähemmän riippuvaisia toisistaan. Ne mahdollistavat joustavamman organisaation, koska jokainen kehitystiimi voi keskittyä oman ominaisuuden kehittämiseen. (Kalenda, Hyna & Rossi, 2018.)

2.2.3 DA

DA:ta eli Disciplined Agilea on kehitetty vuodesta 2009. Alkuun DAD:na eli Disciplined Agile Deliverynä tunnettu kehitysmenetelmä on kehitetty IBM:llä tiimin, Mark Linesin ja Scott Amblerin yhteistyössä. Scott ja Ambler julkaisivat DA:n version 1.x. kirjana 2012. Seuraavassa julkaisussa 2015 menetelmässä keskityttiin joustavaan ja kontekstiriippuvaiseen lähestymiseen DevOps:n ja IT prosessin suhteen. (Ambler & Lines, 2016.) 2017 julkaistu versio sisälsi ajatuksen laajentaa menetelmä ottamaan huomioon yritys kokonaisuutena (Disciplined Agile Enterprise, DAE). Viimeisin versio 5.x. on julkaistu keväällä 2020. (Project Management Institute, Inc., 2021.)

Siinä missä ketterät menetelmät, kuten scrum, kanban ja SAFe toimivat itsenäisinä rakennuspalikoina kehitykselle, DA on menetelmä, joka kertoo, kuin-

ka näitä lukuisia menetelmiä tulee käyttää yhdessä. Ensimmäisen strategian mukaan DA on hybridimenetelmä, joka käyttää muita menetelmiä ja ohjelmistoprosessi-viitekehyksiä perustana. DA:n perustana on tukea koko toimituksen elinkaaren ajan eli ensimmäisestä tuotteen konseptista toimitukseen, operaatioon ja tukeen ja lopulta poistumiseen. (Ambler & Lines, 2016.)

DA koostuu neljästä tasosta (liite 5), jotka ovat perustus, kurinalainen DevOps, arvovirrat ja kurinalainen ketterä yritys eli DAE. Ensimmäinen taso perustus pitää sisällään

- periaatteet, lupaukset ja ohjeet,
- peruseriaatteet ketterydestä ja Lean-ajattelusta,
- peruseriaatteet perinteisistä lähestymistavoista,
- roolit ja tiimin rakenteen sekä
- perustuksen työskentelytapojen (Way of Working, WoW) valitsemiselle.

Toinen taso kurinalainen DevOps käsittää ohjelmistokehityksen ja IT operaatioiden aktiviteettien ohjautumisen, joka tapahtuu DevOps periaatteita noudattaen. DA:n kurinalaisessa DevOpsissa otetaan huomioon myös enterprise-class -näkökulma, jossa turvallisuus ja datanhallinta pyritään integroimaan organisaatioon tehokkaasti. Kun tuotettavia järjestelmiä on satoja tai tuhansia, myös tuen ja julkaisuhallinnan tulee olla vahvoja. DA:n kolmas taso arvovirrat pohjautuvat FLEXiin. FLEX yhdistää organisaation strategiat ja näyttää, miltä tehokkaan arvovirran tulisi näyttää. Tämä auttaa kehittämään organisaation eri osa-alueita ja näkemään organisaation yhtenä kokonaisuutena. Viimeinen taso DAE pyrkii vastaamaan markkinoiden muutoksiin. Markkinoiden muutokset vaikuttavat organisaation muutostarpeeseen ja organisaation muutos toteutuu sen kulttuurin ja rakenteiden kautta. DAE-taso keskittyy muihin organisaation aktiviteetteihin, jotka tukevat organisaation arvovirtoja. (Project Management Institute, Inc., 2021.)

DA on prosessin päätöksentekomalli, joka määrittää joukon käytänteitä, toimenpiteitä ja tietotaitoa. Kurinalaisuus varmistaa, että eteneminen on tavoitteellista käyttämällä hybridistä lähestymistapaa IT-ratkaisun toimittamiseksi. DA koostuu useista viitekehystä kuvaavista piirteistä, jotka ovat:

- ihmiset ensin,
- skaalautumisen tuki,
- päämäärätietoisuus
- yritystietoisuus,
- riski- ja arvojohtoisuus,
- toimituskeskeisyys,
- IT ratkaisukeskeisyys,
- ketteryys,
- hybridi ja
- oppimiskeskeisyys.

Viitekehyksen ihmiset ensin -ajattelu tarkoittaa, että tiimin kesken ei pitäisi olla muodollista hierarkiaa. Tiimin jäseniä myös kannustetaan työskentelemään monialaisesti ja oman erikoisosaamisen ulkopuolella. Tällainen kehitys johtaa resurssien tehokkaampaan käyttöön ja vähentää dokumentaation tarvetta. Skaalaaminen puolestaan helpottuu helpomman viestinnän ja yhteistyön myötä. Skaalautumisen tuki pyrkii auttamaan tiimejä niiden erilaisissa ja yksilöllisissä tarpeissa. Skaalautumisen osa-alueita ovat tiimin koko, maantieteellinen hajautuminen, määräysten noudattaminen, alustan monimutkaisuus, tekninen monimutkaisuus, organisaation hajaantuneisuus, organisaation monimutkaisuus ja yrityskuri.

Viitekehyksen päämäärätietoisuus tarkoittaa, että jokaisella tavoitteella on omat haasteensa, jotka vaativat paneutumista. Haasteisiin puuttumiselle on useita eri ratkaisuja, joiden sopivuus on riippuvaista kontekstista. Viitekehys lähtekin siitä ajatuksesta, että on tunnettava ympäristönsä, mihin sitä ollaan soveltamassa. Yritystietoisuudella puolestaan tarkoitetaan viitekehyksessä tietoisuutta siitä, että tiimi on osa suurempaa kokonaisuutta. Organisaatiossa voi työskennellä päällekkäisiä tiimejä, jotka hyödyntävät toistensa tuotoksia ja organisaation yhteinen visio yleensä ohjaa lopulta tiimien tekemistä. Jotta tiimi olisi tehokas, sen tulee ottaa huomioon muu organisaatio.

Riski- ja arvojohtoisuus noudattavat riskin ja arvon elinkaarta, jossa tiimit pyrkivät huomioimaan projektin riskejä, kuten sidosryhmien yhteisymmärryksen saavuttaminen visiosta ja arkkitehtuurin todistaminen elinkaaren alkuvaiheessa. Projektin elinvoimaisuuden kannalta on tärkeä määritellä tarkistuspisteet, jolloin myös arvioidaan, onko tarvittavat toiminnallisuudet toteutuneet ja onko ratkaisu julkaisukelpoinen. Viitekehyksen yksi piirre on toimituskeskeisyys. Elinkaari ulottuu alullelaitosta rakennukseen ja tuotantoon julkaisuun. Siihen sisältyy myös ennen alkuunlaiton portfoliojohtamista ja julkaisun jälkeisiä toimintoja. Tämä laajentaa elinkaaren rakentamisvaiheeseen keskittyviä ensimmäisen sukupolven ketteriä menetelmiä. Viitekehys myös tukee lean-ajattelua tai jalostettua elinkaarta, joihin monet ketterät tiimit kehittyvät kehittäessään prosessejaan ajan saatossa.

IT ratkaisukeskeisyys tarkoittaa viitekehyksessä sitä, että IT-alan ammattilaiset eivät vain kehitä ohjelmistoa. He myös tarjoavat uudet laitteet ja muuttavat prosesseja, joita sidosryhmät noudattavat. He myös auttavat organisaation rakenteiden muuttamisessa. Viitekehys noudattaa ketterän julistuksen arvoja ja periaatteita, sillä ketterällä on paljon etuja verrattuna perinteiseen vesiputousmalliin tai ei-määritellyihin prosesseihin. Ketteryys on esimerkiksi tehokkaampi menetelmä ja sen sidosryhmät ovat tyytyväisempiä.

DA on hybridimalli, sillä se ottaa käyttöön ja muokkaa käytänteitä muista menetelmistä, kuten scrum, XP ja kanban. Se onkin toisen sukupolven ketterä prosessimenetelmä, sillä se on saanut hyödyntää ennen sitä esitettyjä malleja. Viitekehyksen oppimiskeskeisyys näkyy siinä, että viitekehys pyrkii luomaan tehokkaan oppimisympäristön kolmen tärkeän osa-alueen avulla. Ensimmäinen pureutuu siihen, miten tiimiä autetaan ymmärtämään sidosryhmien tarpeita. Toisena prosesseja pyritään kehittämään paremmaksi yksilöiden, tiimin ja yri-

tyksen tasolla. Kolmas oppimisen näkökulma on tekninen oppiminen ja ymmärtäminen, kuinka käyttää tehokkaasti työkaluja ja teknologiaa ratkaisun rakentamiseksi. (Brown, Ambler & Royce, 2013.)

2.3 Essence

Essence on Ivar Jacobsonin ja SEMAT (Software Engineering Methods and Theory's) -yhteisön kehittämä viitekehys, jonka tarkoituksena on avartaa erilaisia näkökulmia kehitysmenetelmien tarkasteluun (Jacobson, Lawson, Ng, McMahon & Goedicke, 2017). SEMAT toimintakehotus julistaa, että jotkin ohjelmistotuotannon osa-alueet kärsivät epäkypsistä käytänteistä, kuten

- muotivillitysten yleisyys kuuluu tyypillisemmin muotialalle kuin ohjelmistotuotantotalle.
- vakaata, yleisesti hyväksyttyä teoreettista perustaa puuttuu.
- käytetään suuria määriä menetelmiä ja menetelmien muunnoksia, joiden eroja ei tunneta hyvin ja joita on keinotekoisesti suurennettu.
- uskottavaa kokeellista arviointia ja validointia ei ole.
- jako teollisuuden käytänteiden ja akateemisen tutkimuksen välillä.

SEMAT yhteisönä pyrkii tekemään ohjelmistotuotannon uudelleenmäärittelyä, jonka on tarkoitus pohjautua vakaaseen teoriaan, todistetusti hyödynnettäviin periaatteisiin ja parhaisiin käytänteisiin. Näiden tulee sisältää laajasti hyväksytyjen elementtien ytimen. Näiden tulee käsitellä sekä teknologian että ihmisten haasteita. SEMAT:n kaksi suurta tavoitetta ovat laajasti hyväksytyjen elementtien ytimen löytäminen ja vakaan teoreettisen perustan määrittely näiden pohjalta. (Jacobson, Ng, McMahon, Spence & Lidman, 2012.)

SEMAT:n tavoitteena on kaventaa teollisuuden, koulutuksen ja tutkimuksen välisiä kuiluja ja Essence ehdotetaan työkaluksi ohjelmistotuotannon ja koulutuksen välisen kuilun kaventamiseen. Essencen ydin ja kieli luovat pohjan ymmärtää ja kuvailla ohjelmistokehityksen käytäntöjen sekä monimuotoisuutta että yhdenmukaisuutta. Essence tarjoaa mahdollisuuden liittää eri kehitysmenetelmien käytänteitä yhteen. (Ng & Huang, 2013.) Maailmassa on useita tuhansia eri menetelmiä lähestyä ongelmia ja etsiä ratkaisua. Essence-viitekehityksen tarkoituksena on löytää kehitystiimeille sopivimmat tavat työskennellä, löytää millaisia riippuvuuksia kehitystiimeillä on sekä ymmärtää kuinka valitut työskentelytavat heijastuvat käytäntöön. Viitekehityksen avulla tiimin jäsenet pystyvät Jacobsonin ym. (2017) mukaan arvioimaan kehityskaaren tilaa ja toimivuutta. (Jacobson ym., 2017.)

2.3.1 Essencen kieli ja ydin





Kehitysmenetelmät, kuten RUP, SAFe, DAD, Nexus ja LeSS, ovat Jacobsonin mukaan suosittuja enemmän niiden markkinoinnin kuin niiden oppien perus-

teella. Essence-menetelmässä näiden menetelmien opit on käännetty osaksi Essencen kieltä. Niiden hyödyntäminen ei ole sidottu tiettyyn kehitysmenetelmään. Essence-viitekehityksessä ajatus on, että kehitysmenetelmän osia ja rooleja valitaan sen mukaan, joita kehitystiimi uskoo tarvitsevänsä saadakseen tuotteen tai tuotteen osan valmiiksi.

Essence koostuu kahdesta komponentista: Essence-kielestä sekä Essencen ytimeistä. Essencen kielellä kuvataan ohjelmistotuotannossa esiintyviä ilmiöitä, tapahtumia ja menetelmiä mahdollisimman yleisellä tavalla - menetelmän alkuperästä riippumattomana. Essencen kielellä esitetään kehitysmenetelmät ja sen vaiheet mahdollisimman selkeällä ja toistettavalla tavalla. Essencen kieli hyödyntää erilaisia muotoja ja ikoneita, joilla tunnistetaan kehitysmenetelmien tyyppiä. Muodot ja ikonit jaetaan työkaluihin, työtehtäviin ja kompetensseihin. Essence-kielessä käytetään opetus- sekä pelillistämisen tarkoituksessa pelikortin kokoisia kortteja, joita on mahdollista hyödyntää hahmottamaan kokonaisuuksia. Kortit sisältävät sen nimen lisäksi myös pienen kuvauksen kortista sekä alfan, toteuttavan asian, eri tilat tai vaiheet.

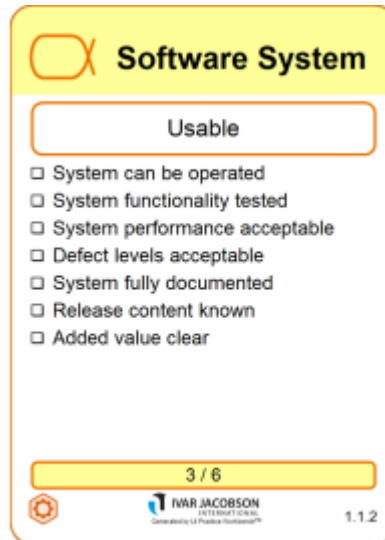
Seuraavassa kappaleessa avaamme Essencen kielen merkityksiä enemmän.

Taulukko 2 Essencen elementtejä

Elementin tyyppi	Syntaksi	Kuvaus
Alfa		Kehitystyön ydinelementti, josta on mahdollista seurata kehitystyön edistymistä ja kehitystyön pyrkimysten tilaa.
Työn tulos		Konkreettinen ammatinharjoittajan toteuttama työn tulos
Aktiviteetti		Työ tai tehtävä, jota ammatinharjoittaja suorittaa.
Kompetenssi		Kyvyt, taidot tai kyvykkyudet, joita tarvitaan työn suorittamista varten.

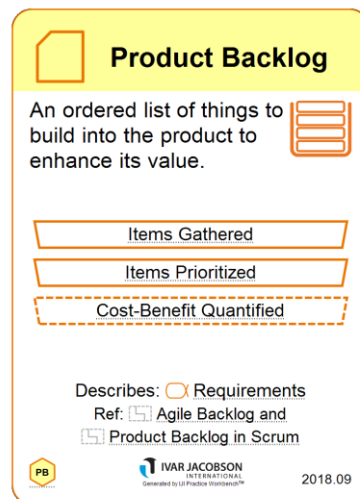
Ohjelmoinnin työkaluihin kuuluvat vaatimukset, ohjelmistojärjestelmät sekä koodipohja. Essencen kielessä nämä on jaettu alfaksi ja työn tuloksiksi. Alfamat ovat tärkeitä elementtejä työn johtamisen kannalta, sillä näiden perusteella on mahdollista arvioida aktiviteetteja ja ohjelmoinnin elinkaaren tilaa. Alfa esitetään essencen kielessä kreikkalaisen alpha-kirjaimen muodossa. Alfamat ovat niitä abstrakteja kokonaisuuksia, joiden etenemistä seurataan, kontrolloidaan ja lopuksi toteutetaan. Essencen alfalla voi olla myös erilaisia tiloja, jotka määrittyvät työn tulosten perusteella. Esimerkkinä tietojärjestelmää voidaan pitää Es-

sencen alfana. Tietojärjestelmällä voi olla useita tiloja. Tämä tila voi olla esimerkiksi, että tietojärjestelmä on käytettävä. Jotta tietojärjestelmä on käytettävä, on sen täytettävä tietyt kriteerit. Tällaisia kriteerejä ovat esimerkiksi ”järjestelmää voi hyödyntää” tai ”järjestelmän toiminnallisuus on testattu” tai ”virhetaso on hyväksyttävä”. Kuvassa 1 esitetyssä Essencen alfakortissa esitetään ohjelmistojärjestelmän tila ”käytettävä”, ja se sisältää useita eri kriteerejä.



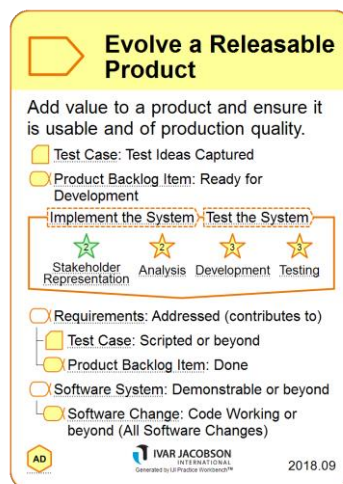
Kuva 1 Essencen alfakortti (Ivar Jacobson International, 2021)

Dokumentaation työkaluja kutsutaan työn tulokseksi (kuva 2), joita voivat olla esimerkiksi vaatimusmäärittelyn, suunnittelumallit tai itse ohjelmiston koodi. Työn tulokset ovat konkreettisia asioita, joilla on erilaisia yksityiskohtaisia tasoja. Tasot voivat vaihdella sen mukaan, kuinka tarkasti esimerkiksi asiakas tai laki vaatii dokumentaation toteutettavan. Työn tulokset voivat myös määrittellä missä tilassa mikäkin alfa on. Työntulokset voivat olla dokumentteja, sopimuksia tai käytänteitä, joista on olemassa jonkinlainen dokumentaatio. Esimerkiksi kanbanin kaltainen progressiotaulu esitetään Essencen kielellä työntuloksena.



Kuva 2 Essencen työntuloskortti, joka on tässä tapauksessa tuotteen kehitysjono. (Ivar Jacobson International, 2021)

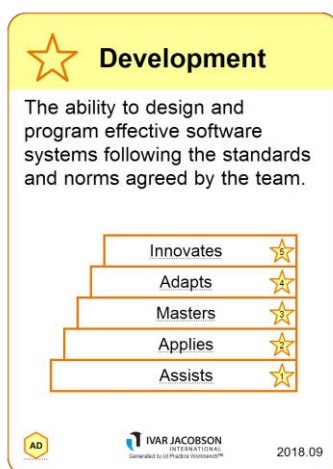
Essencen kielessä aktiviteetit ovat niitä päivittäisiä työtehtäviä, joiden avulla toteutetaan lopullisen tuotteen osaa tai kokonaisuutta. Aktiviteetit voivat olla esimerkiksi kokouksen pitämistä, vaatimusmäärittelyn toteuttamista sekä ohjelmointia. Aktiviteetit voivat olla osa aktiviteettitilaa, jossa on useita yhdistettyjä aktiviteetteja. Aktiviteetti-kortit (kuva 3) sisältävät kuvauksen työstä, jota toteutetaan sekä viittaukset kompetensseihin ja alfoihin. Aktiviteetit ilmaistaan viiren muotoisella kuviolla (taulukko 2)



Kuva 3 Essencen aktiviteettikortti. Sisältää alfoja, kompetensseja ja työntuloksia. (Ivar Jacobson International, 2021)

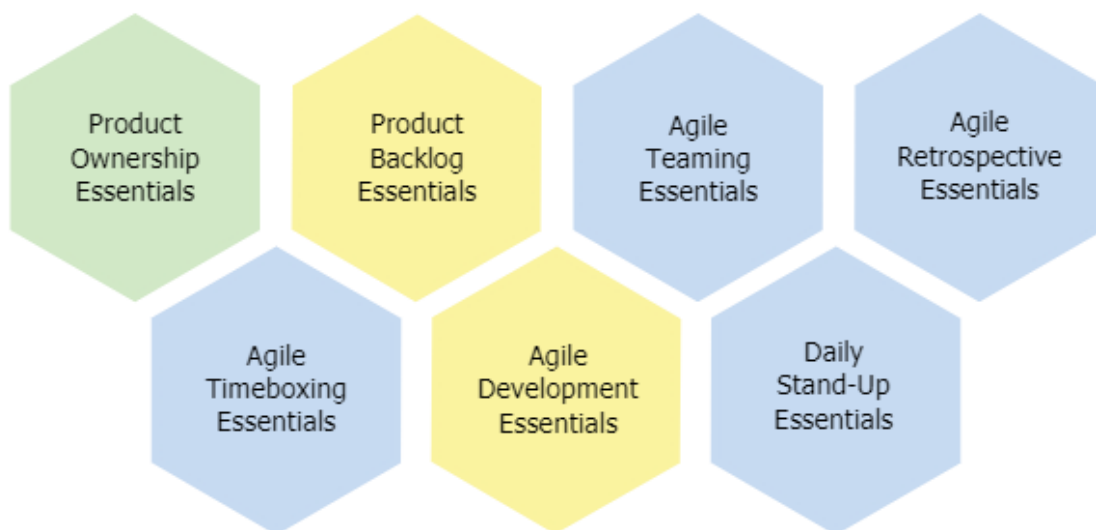
Essencen kielessä kompetensseilla tarkoitetaan kyvykkyyttä toteuttaa työtehtävät. Kompetenssia tarvitaan jokaiseen ohjelmiston osan tai työtehtävän toteuttamiseen. Kompetenssi-kortin avulla tiimit pystyvät tutkimaan löytyykö heidän joukostaan tarvittava osaaminen tehtävän suorittamiseen. Kompetensseja on myös eritasoisia. Esimerkiksi kuvan 4 ohjelmistokehittämiseen liittyvässä kompetenssikortissa tasoja on viisi. Tiimin jäsenet voivat arvioida omaa kompe-

tenssiaan kortin avulla, sekä tehdä tarvittavia muutoksia tiimin jäsenten kesken, mikäli osaamista puuttuu tiimien kesken. Kompetenssit merkitään Essencen kielessä tähden muotoisella kuviolla.



Kuva 4 Essence kielen kompetenssikortti. Kuvaa ohjelmistokehityksen eri osaamistasoja. (Ivar Jacobson International, 2021)

Essencen kielellä on mahdollista muodostaa ja kuvata kokonaisia kehitysmenetelmiä ja sen käytänteitä. Essencen kielen avulla kehityksen tilaa on mahdollista seurata sekä ymmärtää aktiviteettien, kompetenssien ja työn tulosten välisiä suhteita. Ivar Jacobson International on luonut valmiiksi useita erilaisia alfojen, työntulosten, aktiviteettien ja komponenttien kokonaisuuksia, joita kutsutaan käytänteiksi. Käytänteitä on Essencessä tällä hetkellä seitsemän kappaletta. Ne koostuvat parhaista käytänteistä, ja ne esitetään Essencen kielellä. Essencessä on käytänteitä mm. tuoteomistajuudelle, ketteryuden aikalaatikoinnille, kehitysjonolle, ketterän tiimin muodostamiselle, päiväpalaverien pitämiselle sekä retrospektiiville. Kuvassa viisi esitetään kaikki värikoodatut käytänteet.

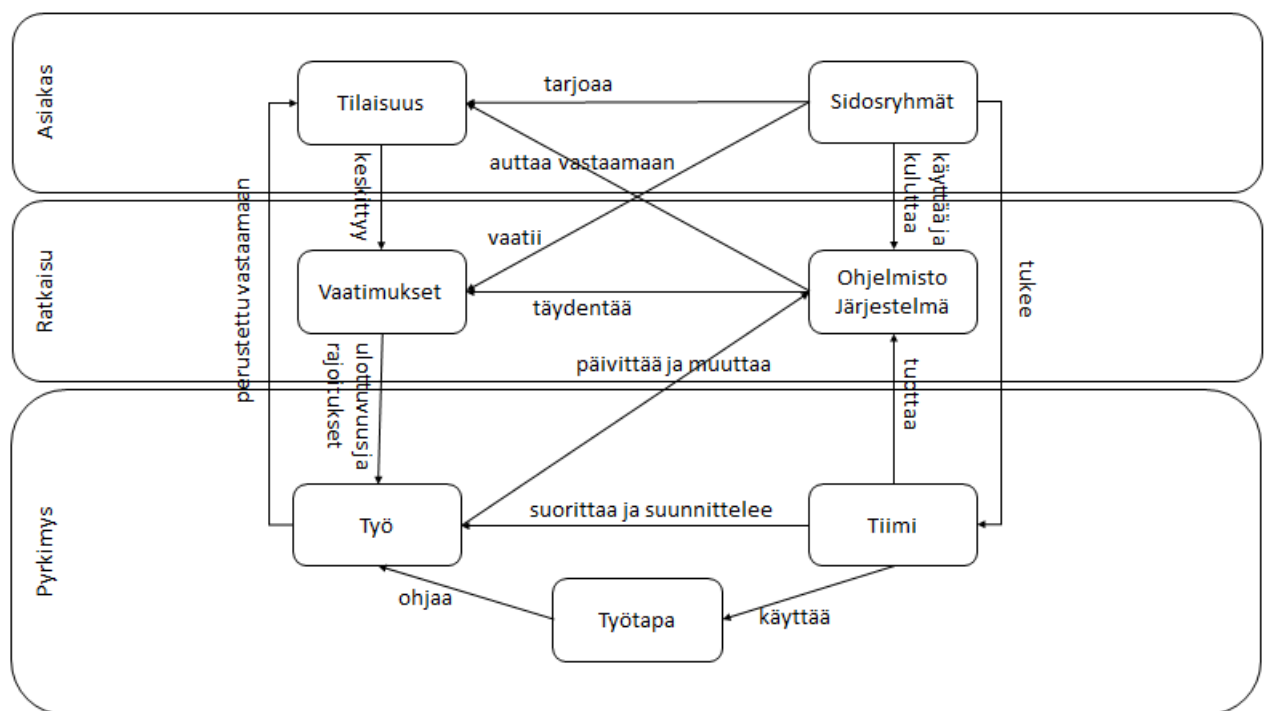


Kuva 5 Essencen käytänteet. Essence sisältää valmiita käytänteitä usealle eri ketterälle tahtumalle ja työkalulle. (Ivar Jacobson International, 2021)

Essencen ydin koostuu kolmesta huomioon otettavasta elementistä.

- Asiakas - sisältää kaiken liittyen itse tuotettavan järjestelmän käyttöön ja hyödyntämiseen. Kuvassa 5 merkitty vihreällä.
- Ratkaisu - sisältää kaiken liittyen tuotettavan järjestelmän spesifikaatioon ja kehitykseen. Kuvassa 5 merkitty keltaisella.
- Pyrkimykset - sisältää kaiken liittyen kehitystiimeihin ja tapoihin työskennellä. Kuvassa 5 merkitty sinisellä.

Ytimen elementit muodostuvat itse Essencen kielen alfoista, aktiviteettiiloista, kompetensseista sekä näiden erilaisista yhdistelmistä eli kaavoista. Kaavat ovat yleismaailmallisia ratkaisuja ja ne voidaan sovittaa tosielämän tilanteisiin. Kaavioita voidaan hyödyntää esimerkiksi esittämällä tietyn ohjelmistoprojektin jäsenen roolin vastualueet ja työtehtävät, tai kaavioilla voidaan esittää projektin elinkaaren tarkistuspisteet, joihin mennessä tietyt alfat tulee toteuttaa. Kuviossa 6 esitetään essencen alfat, jotka ovat jaettuna kolmelle eri elementille.



KUVIO 6 Essencen alfat jaettuna kolmeen elementtiin ja alfojen keskinäiset suhteet

Essencen alfat luovat kivijalan ohjelmistotuotannon ymmärtämiselle ja se ymmärtää suhteen näiden kolmen elementin välille. Asiakas-elementissä ovat alfat tilaisuus/ mahdollisuus sekä sidosryhmä. Tilaisuudella / mahdollisuudella tarkoitetaan liiketoiminnallista mahdollisuutta tai ratkaisua tiettyyn ongelmaan. Sidosryhmät, yleensä sijoittajat ja osakkeenomistajat, mahdollistavat tämän luomisen esimerkiksi rahoittamalla pyrkimyksiä toteuttaa haluttu ratkaisu. Ratkaisu-elementin alla ovat vaatimukset- sekä ohjelmisto/ järjestelmä -alfat.

Ohjelmisto/järjestelmä vastaa havaittuun ongelmaan, mutta sen tulee täydentää tietyt vaatimukset. Vaatimukset asettavat työlle tietyt rajoitukset sekä ulottuvuudet, joita ratkaistaan. Pyrkimys-elementissä tapahtuu itse työ, jossa tiimit tuottavat tarvittavaa ohjelmistoa ja työ päivittää ja muuttaa. Tiimit voivat itse valita työtapansa, jolla toteuttavat työtänsä. Essencen alfat ovat toisistaan riippuvaisia ja niiden riippuvuudet ovat toisiaan täydentäviä.

2.3.2 Skaalautuva essence

Essenceä voidaan käyttää niin pieniin kuin suuriinkin tiimeihin. Skaalatussa ketterässä kehityksessä toisen kehitystiimin tarvitsemat työtavat ovat erilaiset toiselle kehitystiimille, vaikka tiimeillä on yhteinen pyrkimys. Erot johtuvat tiimien kohtaamien asioiden erilaisuudesta. Työskenteli tiimi sitten tiimi- tai ohjelmatasolla tai mitä käytäntöjä sitten käyttäkään, kaikki käytännöt on kuvattu ytimen yläpuolella. Lisäksi ohjelmien ja tiimien eteneminen ja terveys voidaan perustaa ytimen alfoihin ja valittujen käytäntöjen alfoihin. Essencen käyttö tarjoaa sekä pienille että isoille tiimeille

- linssin arvioida työtapoja (the way of working), mitä käytäntöjä käyttää, korostaa riskejä ja niin edelleen.
- tavan tehdä käytänteistä täsmällisiä.
- tavan arvioida jokaisen ponnistuksen etenemistä ja terveyttä. (Jacobson, Ng, McMahon & Goedicke, 2019.)

Skaalautuvan ketterän tärkeät käytännöt antavat aloittelijan työkalupakin, joka vastaa tavanomaisimpiin ja kriittisiin osa-alueisiin skaalautuvassa ketterässä kehityksessä. Käytännöt ovat hyvä haaste pyrkimyksille, joissa työskentelee useita tiimejä ja näiden tiimien halutaan työskentelevän yhdenmukaisella ketterällä tavalla. Oppimisenäkökulma on pelillinen oppiminen, jossa käytetään pokerimaisia pelikortteja ja/tai ketterää simulaatiota. Elementit on kuvattu riittöisästi, jotta harjoittajat voivat jatkuvasti ottaa niitä käyttöön. Pääsääntöisesti elementit otetaan käyttöön harjoituskorttien avulla. Skaalautuvan ketterän tärkeitä käytänteitä on kymmenen:

- Mukautuva rahoitus (Adaptive Funding Essentials): hanki ja ohjaa rahoitusta tukemaan jatkuvaa arvovirtausta ja maksimoi investointi.
- Ketterä arkkitehtuuri (Agile Architecture Essentials): ohjaa ratkaisulähtöisyyden kehitystä, joka sopeutuu muuttuviin tarpeisiin ja haasteisiin.
- Ketterä hallintotapa (Agile Governance Essentials): varmista rahoituksen ohjaus ja seuranta. Varmista myös lain ja muiden säännösten, prosessien ja menettelytapojen noudattaminen.
- Kehitystuki (Development Support Essentials): automatisoi ja operoi ympäristöjä kehitystiimien tuen avulla optimoidaksesen kehitysprosessin.

- DevOps (DevOps Essentials): Dev ja Ops työskentelevät yhdessä parantaakseen vastaavuutta asiakkaan tarpeisiin ja muuntaakseen julkaisuja liveksi usein, sujuvasti ja turvallisesti.
- Jaksollinen sovittaminen (Periodic Alignment Essentials): sovita monen tiimin työn priorisoinnit ja suunnitelmat kokonaisuuden tavoitteisiin käyttämällä synkronoitua suunnittelurytmiä.
- Tuotteen johtaminen (Product Management Essentials): johda tuoteideoiden kehitystä arvon maksimoimiseksi annetulla rajoitetulla resurssilla.
- Julkaisun johtaminen (Release Management Essentials): kehitä ja säädä strategia, joka mahdollistaa säännölliset, nopeat, turvalliset ja oikea-aikaiset julkaisut.
- Jaettu kehitysjono (Shared Backlog Essentials): priorisoi ja johda tuotteen arvon toimitusta vaativien monen tiimin yhteistyötä.
- Tiimien tiimi (Team Of Teams Essentials): saavuta kollektiivisesti korkeatasoinen tiimi yhteistyötä tekevästä tiimeistä. (Ivar Jacobson International SA, 2021b.)

2.3.3 Essencen nykytila

Essence on vielä varsin vähän käytetty ketterien kehitysmenetelmien työkalupakki. Essencen tutkimus on vielä hyvin vähäistä verrattuna muihin kehitysmenetelmiin. Julkaistut tutkimukset käsittelevät pääasiassa ketteryyden ymmärtämistä ja kehittämistä Essencen avulla. Essencen

Park, McMahon ja Myburgh (2016) julkaisema tutkimus Essencen vahvistamasta scrumista, jonka havaintona on Essencen monimuotoisuus ja sen mahdollistamat erilaiset käyttötavat. Tutkimuksessa he mainitsevat tiimien mahdollisuuden tarkastella ja mukauttaa toimintatapojaan Essencen alfojen eri tilojen ja listojen avulla. Myös organisaatioiden kokoa ei ole rajattu, ja Essencessä nähdään potentiaalia myös suurten organisaatioiden tiimien työskentelytapojen hallintaan. Essencen hyödyntämistä on tutkittu opiskelijaympäristössä Ng ja Huang (2013), jota käsitellään heidän julkaisemassa artikkelissa "Essence: A Framework to Help Bridge the Gap between Software Engineering Education and Industry Needs". Tutkimuksessa kerättiin palautetta seitsemästä kiinalaisesta yliopistosta Essencen käytöstä. Ng ja Huang havaitsivat, että Essencen kaltainen yhteinen kielen konsensus on hyödyllistä ja sitä voidaan hyödyntää usean ketterän kehitysmenetelmän tavoin. Huang ja Ng (2014) ovat myöhemmin tutkineet Essencen hyödyntämistä tapaustutkimusten arviointiin, mutta tutkimus on ollut tästä vähäistä. Essencen hyödyntämistä opiskelijaympäristössä on myös tutkinut Jan-Phillipp Steghöfer vuonna 2019 julkaisemassa artikkelissaan. Essenceä hyödynnettiin kehitysmenetelmien kurssilla. Steghöfer kertoo Essencen olevan hyödyllinen opetustyökaluna. Essencen hyödyntämisestä käytännössä on heikosti tutkimusta, joista myös suurin osa ovat harmaata kirjallisuutta. Myös monet Essenceä käsittelevät julkaisut ovat itse Essencen kehittäjän, Ivar Jacobsonin, käsialaa.

Taulukko 3 Skaalautuvien ketterien kehitysmenetelmien vertailua, mukailien Kalenda, Hyna & Rossi (2018)

Näkökulma	SAFe	LeSS	DA	Skaalattu Essence
Tiimikoko	50–120 henkilöä julkaisujutissa	10 Scrum tiimiä, 7 jäsentä	200 työntekijää tai enemmän	Pienistä suuriin
Käytön laajuus	Korkea	Kohtalainen	Matala	Matala
Kypsyys	Korkea	Korkea	Kohtalainen	Matala
Monimutkaisuus	Korkea / kohtalainen	Kohtalainen / matala	Korkea	Kohtalainen / matala
Organisaatiotyyppi	Perinteiset yritykset	Suuret yritykset	Monet organisaatiot ja yritykset	Pienistä suuriin

Taulukossa kolme esitetään skaalattu Essence ja muita skaalautuvia ketteriä kehitysmenetelmiä. Essencen voidaan nähdä olevan kypsyydeltään matala. Toisaalta Essencen avulla on mahdollista muodostaa lähes mitä tahansa ketteriä kehitysmenetelmiä, eli kyse on enemmän Essencen tietämyksestä ja tuntemisesta. Essenceä voidaan hyödyntää pienistä yrityksistä suurempiin organisaatioihin, suurille tai pienille tiimeille, koska sitä on mahdollista skaalata käytänteiden avulla. Essencen monimutkaisuus nähdään muodostuvan Essencen siitä, mitä käytänteitä Essenceen valitaan.

2.4 Skaalautuvien ketterien menetelmien haasteet ja menestystekijät

Tässä luvussa tarkastellaan skaalautuvien ketterien menetelmien yleisimpiä menestystekijöitä ja haasteita, joita kirjallisuudessa esiintyy. Useimmiten esiintyvät menestystekijät ja haasteet esitellään myöhemmin luvun 6 kipupistemallissa.

2.4.1 Menestystekijät

Dikert, Paasivaara ja Lassenius (2016) toteuttamassa skaalautuvien ketterien kehitysmenetelmien kirjallisuuskatsauksessa ovat tutkineet organisaation muutoksen menestystekijöitä ja haasteita. Tutkimuksessa on tutustuttu useampaan tieteelliseen julkaisuun, joiden perusteella on tunnistettu menestystekijät ja haasteet. Tutkimuksessa eniten mainintoja saaneet menestyskategoriat ovat ketterän menetelmän valinta ja kustomointi, johdon tuki, asenne ja yhdenmukaisuus ja harjoitus ja valmennus. *Ketterän menetelmän valinnan ja kustomoinnin menestystekijät* ovat menetelmän huolellinen kustomointi, yhteen lähestymistapaan mukautuminen, vanhojen tapojen kartoittaminen ja yksinkertaisuus. *Joh-*

don tuessa menestystekijöitä ovat johdon tuen varmistaminen ja näkyminen ja johdon kouluttaminen ketteryyteen. *Asenne ja yhdenmukaisuus* -menestystekijät ovat ketteriin arvoihin keskittyminen, sosiaalisten tapahtumien järjestäminen, ketterien yhteisöjen vaaliminen ja organisaation yhdenmukaistaminen. *Harjoitus ja valmennus* -kategorian menestystekijät ovat ketterien menetelmien koulutuksen mahdollistaminen ja tiimien valmennus matkan varrella. Loput seitsemän menestyskategoriaa ovat muutokseen sitoutuminen, johtaminen, pilotointi, ihmisten osallistaminen, viestintä ja läpinäkyvyys, tiimin autonomia ja vaatimusten hallinta. *Muutokseen sitoutumisen* menestystekijöitä ovat viestiminen muutoksen olevan ei neuvoteltavissa ja vahvan sitoutumisen osoittaminen. *Johdaminen* menestystekijänä pitää sisällään muutosjohtajien tärkeyden tunnistaminen ja heidän valjastamisensa työhön puhtaalta pöydältä. *Pilotointi* menestyskategoriana käsittää menestystekijöinä muutoksen aloittamisen pilotilla, jonka tarkoitus on kerätä hyväksyntää. Pilotista on tarkoitus myös oppia. *Ihmisten osallistaminen* tulee aloittaa ketteryyden kannattajista ja muutokseen kannattaa osallistaa henkilöitä, joilla on ketterästä aikaisempaa kokemusta. Koko organisaatio tulee osallistaa muutokseen. *Viestintä ja läpinäkyvyys* menestystekijänä käsittää intensiivisen muutoksesta viestimisen, muutoksen pitämisenä läpinäkyvänä ja muutoksen alussa positiivisten kokemusten luomisen ja niistä kertomisen. *Tiimin autonomia* -menestyskategorian menestystekijät ovat tiimien itsenäisen organisoimisen ja ruohonjuuritason voimaannuttamisen mahdollistaminen. *Vaatimusten hallinta* on viimeinen menestyskategoria, jonka menestystekijöitä ovat tuoteomistajan roolin tärkeyden tunnistaminen ja investointi vaatimusten uudelleen määrittelyyn oppimiseen. (Dikert, Paasivaara & Lassenius, 2016.)

Kalenda, Hyna ja Rossin (2018) toteuttamassa tutkimuksessa selvitettiin suuresti skaalautuvien ketterien kehitysmenetelmien käytäntöjä, haasteita sekä menestystekijöitä. Tutkimus on kaksiosainen tutkimus, jossa ensimmäisenä selvitetään suuresti skaalautuvien ketterien kehitysmenetelmien käyttöönoton onnistumista tutustumalla tieteellisiin julkaisuihin ja seuraavaksi hyödyntämällä löydettyjä tieteellisten julkaisujen löytyneitä tuloksia tutkimuksen kohteena olevassa yrityksessä. Kalenda, Hyna ja Rossi (2018) tunnistivat tutkimuksessaan neljä menestystekijää. Neljä suurinta menestystekijää ovat

- yrityskulttuuri,
- aikaisempi ketterä ja lean kokemus,
- johdon tuki ja
- näkemysten ja arvojen yhdenmukaistaminen.

Aikaisempi kokemus ketteristä kehitysmenetelmistä ja lean-filosofiasta tukevat suuresti skaalautuvien ketterien kehitysmenetelmien omaksumista. Kulttuurinen muutos organisaatiossa onnistuu ketterien kehitysmenetelmien valmentajilta ja kouluttajilta, jotka vastaavat organisaation jäseniä mietittyttäviin kysymyksiin. Organisaation jäsenten tulee kokea ketterän kehitysmenetelmien läpinäkyvyys positiivisena asiana. Johdon tuella on merkitystä ketterien kehi-

tysmenetelmien käyttöönotossa. Johdon osallistuminen kehitysmenetelmän toteuttamiseen auttaa organisaation jäseniä motivoitumaan ottamaan uudet menetelmät käyttöön. Yleisesti jaetut arvot ja näkemykset edesauttavat suuresti skaalautuvien kehitysmenetelmien käyttöönottoa.

Ketterien kehitysmenetelmien skaalautuvuuden menestystekijöitä ovat myös tutkineet Shameem, Kumar, Chandra ja Khan vuonna 2017 julkaistussa tutkimuksessaan. Shameem sekä edellämainitut tutkijat selvittivät ketterien kehitysmenetelmien menestystekijöitä sekä kuinka näkemykset eroavat asiakkaan kuin toimittajan puolelta. Tutkimuskysymykset pyrkivät saamaan vastauksen siihen mitkä ovat tärkeimpiä menestystekijöitä maailmanlaajuisessa ketterän kehitysmenetelmän käytössä ja käyttöönotossa. Tutkimuksen tarkoituksena on myös luoda viitekehys, joka auttaa kategorisoimaan menestystekijöitä. Tutkimus on toteutettu systemaattisena kirjallisuuskatsauksena, johon valikoitui 20 tieteellistä artikkelia.

Kirjallisuuskatsauksessa avattiin valituissa tutkimuksissa esiintyneitä menestystekijöitä ja selvitettiin mitkä menestystekijät esiintyvät useimmin. Mikäli menestystekijä esiintyi kirjallisuudessa enemmän kuin 50 % koko aineistosta, menestystekijää kutsuttiin kriittiseksi menestystekijäksi. Tutkimuksessa löydetty kriittiset menestystekijät ovat

- asiakkaan sisällyttäminen projektiin,
- kommunikointi, koordinointi sekä kontrolli,
- johdon tuki ja
- projektin läpinäkyvyyden edistäminen.

Paasivaara on vuonna 2017 julkaistussa konferenssipaperissa käsitellyt SAFen käyttöönottoa maailmanlaajuisesti hajautetussa organisaatiossa. Tutkimuksen tyyppi on case-tutkimus, jossa kahta suomalaismalesialais organisaation osastoa ja heidän jäseniään on haastateltu. Tutkimuksessa haastateltiin yhteensä 12 organisaation jäsentä, joiden roolit vaihtelivat tuoteomistajan, johtajan, toimitusjunainsinöörin ja tiimijäsenen välillä. Haastattelut olivat löyhästi strukturoituja. Haastateltavilta kysyttiin heidän kokemuksiaan suuresti skaalautuvista ketteristä kehitysmenetelmistä. Haastateltuja tapauksia vertailtiin keskenään ja tulokseksi havaittiin, että suuresti skaalautuvien ketterin kehitysmenetelmien käyttöönotto ja muutoksessa on eroja. Tutkimuksessa mm. toinen organisaation linja oli jäänyt vähälle huomiolle SAFE-koulutuksessa sekä jättäneet linjan tulevan muutoksen vähemmälle huomiolle. Tästä syntyi kyseisen linjaston jäsenten keskuudessa muutosvastarintaa.

Tutkimuksessa Paasivaara (2017) esittää seitsemän menestystekijää suuresti skaalatun ketterän kehitysmenetelmän käyttöönottoon, joista neljää ensimmäistä on kriittisiä menestystekijöitä. Tutkimuksessa tunnistettiin seuraavat menestystekijät:

- henkilöstön koulutus tarpeeksi ajoissa,
- ihmisten informoiminen ja osallistaminen,

- muutosagenttien sisällyttäminen,
- ulkopuolisen konsultin hyödyntäminen,
- ensimmäisen PI-palaverin hyvä suunnittelu
- kokoaikaisen toimitusjunainsinöörin sisällyttäminen tiimiin sekä
- toteuttaa paranneltuja prosesseja antamalla velvollisuuksia ja seuraamalla käyttöönottoa.

Vaikka tutkimus käsittelee SAFea, siitä löydetyt menestystekijät ovat yleistettävissä myös muihin suuresti skaalautuviin ketteriin kehitysmenetelmiin (Paasivaara, 2017).

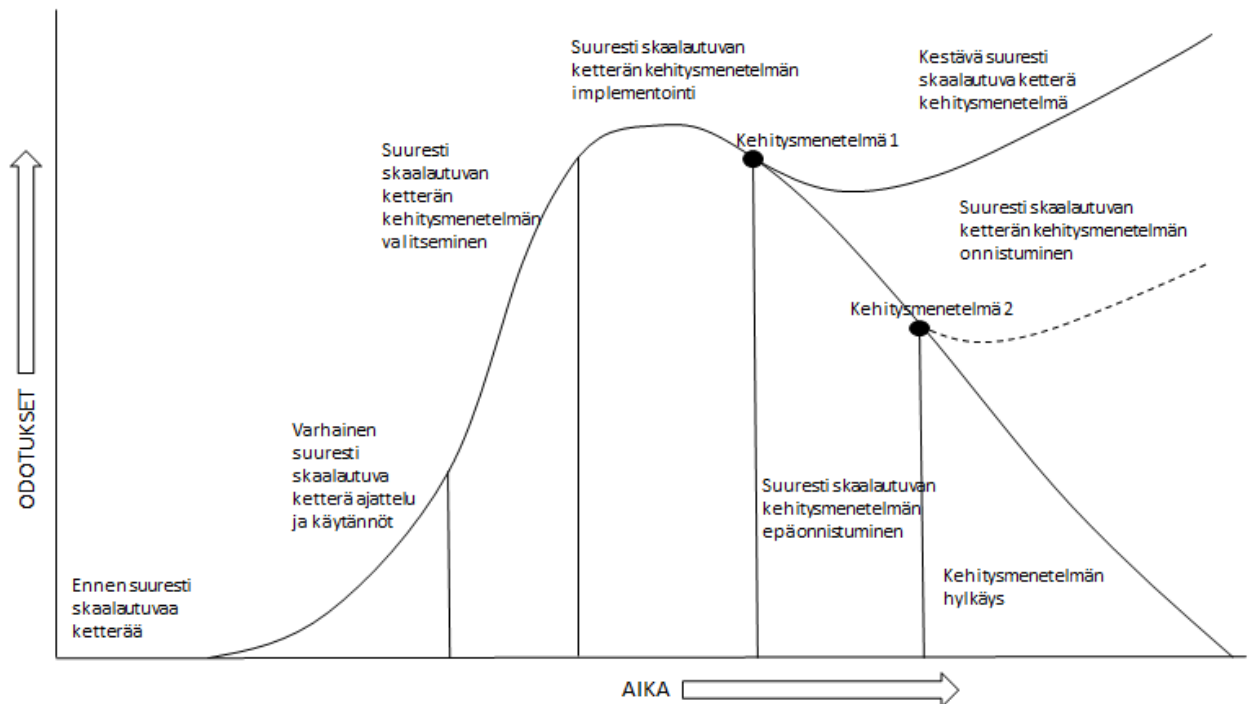
Putta, Paasivaara ja Lassenius (2018) tutkivat hyötyjä ja haasteita SAFE-menetelmän käyttöönotossa kirjallisuuskatsauksen avulla. Tuloksena esitetään SAFE-menetelmän käyttöönoton hyötyjä ja haasteita. Tutkimuksessa tunnistettiin tärkeimmät hyödyt, jotka ovat

- läpinäkyvyys,
- yhdenmukaisuus,
- tuottavuus,
- ennustettavuus ja
- tuotteen toimitusnopeus.

2.4.2 Haasteet

Skaalautuvien ketterien viitekehysten tutkimus on vielä vajavaista niiden omaksumisen, käytön, tehokkuuden ja haasteiden osalta. Conboy ja Carroll (2019) tutkivat viidentoista vuoden aikana kolmentoista globaalien yrityksen muuttumista ketteräksi organisaatioksi. He tutkivat organisaatioita ketteryyden omaksumisen alusta asti ja kun organisaatiot valitsivat, implementoivat ja ylläpitivät suuresti skaalautuvia ketteriä kehitysmenetelmiä, niiden yhdistelmiä tai organisaatiolle kustomoituja menetelmiä. Suuresti skaalautuvan ketterän kehitysmenetelmän omaksumisen vaiheet kuvataan kuviossa 7. Osa organisaatioista omaksui skaalautuvan ketterän menetelmän onnistuneesti ja osa ei niin menestyksekkäästi. Kuviossa 7 onnistunutta implementointia kuvaa Kehitysmenetelmä 1. Ei niin menestyksekkäästi menetelmän omaksuneet organisaatiot joko hylkäsivät menetelmän kokonaan tai ottivat käyttöön vaihtoehtoisen menetelmän, jota kuvaa Kehitysmenetelmä 2 kuviossa 7.

Syvällisen ja pitkäaikaisen tarkastelun, haastatteluiden ja jatkuvalla pääsillä dataan, dokumentointiin ja työkaluihin Conboy ja Carroll löysivät haasteita skaalautuvien ketterien kehitysmenetelmien implementoinnissa. Esitettävät haasteet ovat esiintyneet vähintään kahdessa tapauksessa tutkimukseen osallistuneista organisaatioista ja niiden tukena on selvä näyttö niiden osallisuudesta. (Conboy & Carroll, 2019.)



KUVIO 7 Skaalautuvien ketterien menetelmien omaksuminen Conboyn ja Carrollin (2019) mukaan

Vaikuttavimpia haasteita, joita organisaatiot kohtaavat suuresti skaalautuvien ketterien kehitysmenetelmien omaksumisessa on Conboy ja Carrollin (2019) mukaan yhdeksän. Ensimmäinen haaste on suuresti skaalautuvan ketterän viitekehityksen *käsitteiden ja termien määrittely*. Viitekehysten perusteet ovat usein selkeitä, mutta kun viitekehystä sovelletaan organisaation kontekstiin, viitekehystä on vaikea löytää ohjeistusta. Jotta haaste ei koituisi ongelmaksi, on suositeltavaa määrittellä, mitä eri termit tarkoittavat omassa organisaatiossa ennen implementointia. Muutoksen on vastattava liiketoiminnan tarpeita ja muutoksen visioista ja arvoista on puhuttava kaikille yhteisellä sanastolla. Saaston on oltava sidosryhmän ulottuvilla muutoksen alkuvaiheessa.

Toinen haaste on suuresti skaalautuvien ketterien viitekehysten *vertailu ja vastakkainasettelu*. Viitekehystä vertailevia menetelmiä ei ole, mikä vaikeuttaa kriittistä päätöksentekoa jonkin viitekehityksen omaksumisen puolesta. Suositus on, että organisaatiossa käytetään sen arvon ytimessä olevia mittareita, jolloin voidaan arvioida, miten viitekehyses vaikuttavat organisaation tuottavuuteen ja suoritukseen. Esimerkkejä mittareista ovat työntekijöiden sitoutuvuus, asiakastytyväisyys ja tuottavuus.

Seuraava haaste on *valmius ja halu* ottaa suuresti skaalautuva viitekehyses vastaan. Organisaation muutos vaatii sekä työntekijöiden että rakenteiden muutosvalmiutta ja -halukkuutta. Haluttomuutta ja valmiutta muutokseen lisää viitekehyses vaihtaminen useaan kertaan. Haasteellista on se, että viitekehyses eivät anna ohjeita, kuinka määrittää organisaation valmius ja halu aloittaa muutosprosessi. Suositeltavaa on tehdä valmiusarviointi ja tunnistaa ja tehdä

suunnitelmat ongelmien ratkaisemiseksi. Heikoilla alueilla on tärkeä tehdä muutos inkrementaalisesti.

Neljäs haaste on *tasapainon löytäminen* organisaation rakenteiden ja skaalautuvan ketterän viitekehysten välillä. Haasteellista on, että viitekehukset sisältävät paljon rakenteiden, rutiinien ja työkalujen määrittelyjä. Lisäksi organisaatiot ovat jo valmiiksi jatkuvassa muutostilassa ulkopuolelta tulevan kilpailun ja säännösten luoman paineen myötä. Suositeltavaa on tunnistaa muutoksen aiheuttamia uusia rakenteellisia muutoksia ja arvioida niiden vaikutuksia organisaation suorituskykyyn, standardien noudattamiseen ja joustavuuteen. Muutoksen edut ja haita tulee arvioida ja miten ne saattavat muuttaa liiketoimintaa.

Viides haaste on suuresti skaalautuvan viitekehysten *implementointipäätös ylhäältä alas tai alhaalta ylös*. Alhaalta ylös -implementointi on tehokkain tapa pienissä scrum-implementoinneissa. Skaalautuvan ketterän kehitysmenetelmän implementoinnissa alhaalta ylös tapahtuvan implementoinnin teho ei ole niin selvää, sillä usein ylemmän johdon tuki ja osallistuminen ovat merkittävässä asemassa implementoinnin onnistumisessa. Alhaalta ylös tapahtuvassa implementoinnissa johdon tuki jää monesti puuttumaan. Suositeltavaa on ensinnäkin selvittää, millaista implementointitapaa valittu viitekehys suosittelee. Näiden tapojen väliltä on löydettävä selkeä tasapaino ja kaikille organisaation tasoille on mahdollistettava koulutusta. Sekä ylhäältä että alhaalta implementoinnin tukeminen tulee olla jatkuvaa.

Viitekehysten tarkka noudattaminen arvon kustannuksella on kuudes haaste. Ketterää muutosta usein mitataan virheellisesti viitekehysten tarkalla noudattamisella viitekehysten implementoinnista syntyvän arvon sijaan. Arvon syntymisen mittaaminen viitekehysten implementoinnin vaikutuksesta on vaikeaa, sillä sen mittaaminen välittömästi ja konkreettisesti on vaikeaa ja viitekehysten vaikutusta on hankala erottaa muista arvoa nostavista tekijöistä. Suositeltavaa on päättää, suosiiko muutos viitekehykseen sulautumista vai hyötyykö liiketoiminta enemmän viitekehysten kokonaisvaltaisesta menestymisestä. Muutostekijät, jotka vaikuttavat enemmän sulautumiseen kuin arvoon, tulee tunnistaa. Muutostarve organisaation tavoitteiden näkökulmasta on hyvä kartoittaa.

Seitsemäs haaste on *puute todisteisiin perustuvasta näytöstä* suuresti skaalautuvien ketterien viitekehysten käytöstä. Vaikka tutkimusta on tehty suuresti skaalautuvista viitekehyksistä, vielä puuttuu tapaustutkimuksia, joissa viitekehukset asetetaan todelliseen ympäristöön. On suositeltavaa kerätä todisteita mittareiden avulla muutoksesta. Inkrementaalinen implementointi mahdollistaa oppimisen muutosprosessin aikana. On tärkeää tunnistaa ongelmia ja kohdistaa ne johonkin tapaukseen, jolloin ohjaus ja parhaiden käytäntöjen käyttöönotto muutosprosessissa on mahdollista.

Kehittäjän autonomian säilyttäminen suuresti skaalautuvassa ketterässä viitekehyksessä on kahdeksas haaste. Kehittäjät haluavat työskennellä mahdollisimman autonomisesti päättäen itse muun muassa työskentelyajasta, -paikasta, -kaluista ja -laitteista. Skaalautuessa autonomian säilyttäminen on vaikeaa ja suuresti skaalautuvat ketterät viitekehukset kohtaavat tämän ongelman kärjissä.

tetysti. Suosituksia ovat työntekijöiden tyytyväisyyden selvittäminen autonomiaan, jatkuva auditointi muutoksen tietoisuudesta ja käyttöön otosta ja selvittää, voivatko uudet käytännöt parantaa autonomiaa ja helpottaa muutosta.

Viimeinen haaste on *asiakasprosessien* ja suuresti skaalautuvan ketterän viitekehysten väärin *kohdistuminen*. Asiakkaat ovat tänä päivänä läheisesti mukana organisaatioiden prosessisuunnittelussa. Viitekehysten käsitteet ovat kuitenkin asiakkaille haastavia ja niiden tavat ja rakenteet on haastava piilottaa ja uudelleennimetä. Suositeltavaa on harkita sidosryhmien mukaan ottamista jo menetelmän valintavaiheessa. Menetelmistä on hyvä arvioida, kuinka ne tukevat organisaation joustavuutta vastata kasvaviin dynaamisiin asiakastarpeisiin. (Conboy & Carroll, 2019.)

Dikert, Paasivaara ja Lassenius (2016) tutkivat systemaattisessa kirjallisuuskatsauksessa suuresti skaalautuvan ketterän muutoksen haasteita ja menestystekijöitä. Haasteita tunnistettiin 35, jotka jaettiin yhdeksään kategoriaan. Eniten mainittuja haastekategorioita, joita organisaatiot kohtaavat muutoksessa suuresti skaalautuvaksi ketteräksi organisaatioksi ovat

- ketteryuden implementoinnin vaikeus,
- kehittämisen ulkopuolisten toimintojen integrointi,
- muutosvastarinta ja
- vaatimusmäärittelyyn liittyvät ongelmat.

Ketteryyden implementoinnin vaikeuteen liittyy haasteita, kuten konseptien väärinymmärtäminen, ohjaavan kirjallisuuden puuttuminen ja ketteryuden huono kustomointi. *Kehittämisen ulkopuolisten toimintojen integroinnin* haasteet puolestaan ovat muun muassa muiden osastojen haluttomuus muuttua ja vaikeus sopeutua ketteriin käytäntöihin. *Muutosvastarintaan* kuuluu yleinen haluttomuus muutokselle, skeptisyys uusia työtapoja kohtaan, ylhäältä alas ohjaus herättää vastustusta ja johdon haluttomuus muuttua. *Vaatimusmäärittelyyn liittyvät haasteita* ovat esimerkiksi vaatimusten uudelleen määrittelyn haasteet, käyttäjätarinoiden luonnin ja arvioinnin haasteet ja lyhyen ja pitkän aikavälin suunnittelun välinen kuilu.

Loput viisi haastekategoriaa ovat

- investoinnin puute,
- usean tiimin koordinointi,
- erilaiset lähestymistavat usean tiimin yhteistyössä,
- hierarkkinen johtaminen ja
- organisaation rajoitukset ja laadunvarmistuksen haasteet.

Investoinnin puutteet tarkoittavat muun muassa puutetta valmennuksesta, harjoittamisesta ja työvoimasta. *Usean tiimin koordinointi* -haasteita ovat tiimien yhteensovittaminen, autonominen tiimimalli, globaalisti hajautetut tiimit ja teknisen jatkuvuuden saavuttaminen. *Erilaiset lähestymistavat usean tiimin yhteistyössä* tarkoittaa haasteita ketteryuden erilaisessa tulkitsemisessä ja uusien ja

vanhojen lähestymistapojen rinnakkaisuus. *Hierarkkinen johtaminen ja organisaation rajoitukset* haasteita ovat esimerkiksi keskijohdon roolin epämääräisyys, johto toimii vesiputousmallin mukaan, vanhan byrokratian ja siilojen ylläpito. Viimeinen haastekategoria on *organisaation rajoitukset ja laadunvarmistuksen haasteet*, jossa haasteita ovat ei-toiminnallisten vaatimusten testaus, automaatiotestauksen puute ja vaatimusten monitulkintaisuus vaikuttaa laadunvarmistukseen. (Dikert, Paasivaara & Lassenius, 2016.)

Kalenda, Hyna ja Rossin (2018) toteuttamassa tutkimuksessa selvitettiin suuresti skaalautuvien ketterien kehitysmenetelmien käytäntöjä, haasteita sekä menestystekijöitä. Tutkimus on kaksiosainen tutkimus, jossa ensimmäisenä selvitetään suuresti skaalautuvien ketterien kehitysmenetelmien käyttöönoton onnistumista tutustumalla tieteellisiin julkaisuihin ja seuraavaksi hyödyntämällä löydettyjä tieteellisten julkaisujen löytyneitä tuloksia tutkimuksen kohteena olevassa yrityksessä. Heidän havaitsemansa haasteet skaalautuvien ketterien kehitysmenetelmien implementoinnissa ovat

- muutosvastarinta,
- liian nopea uuden prosessin omaksuminen,
- laadunvarmistus ja
- aikaisempien ja ei-ketterien osien integrointi organisaatioon.

Muutosvastarinta haasteena esiintyi tieteellisessä kirjallisuudessa kaikista useimmin. Muutosvastarintaa esiintyi jokaisella organisaation hierarkian tasolla. Vanhojen työskentelytapojen ja työkalujen unohtaminen ja uusien ketterien kehitysmenetelmien käyttöönotto voi aiheuttaa tiimien sisällä muutosvastarintaa. Myös liian nopeasti käyttöön otetut uudet toimintatavat ja kehitysmenetelmät eivät tukeneet skaalatun ketterän kehitysmenetelmän käyttöönottoa. Laadunvarmistuksen haasteilla tarkoitetaan ongelmaa ketterän kehitysmenetelmän luonteesta, jossa tiimit saavat valita oman tavan työskennellä. Kirjallisuuskatsauksessa he löysivät ketterien kehitysmenetelmien nostaneen tiimien työskentelymotivaatiota, mutta havaitsivat että tiimit eivät hallinneet kehitysprosesseja asiallisesti. Kehitysympäristöä ei hyödynnetty sen suomilla tavoilla ja tekninen velka kasvoi. Viimeisenä haasteena on vanhojen työskentelytapojen säilyttäminen osana ketterää kehitysmenetelmää, jonka vuoksi skaalautuvien ketterien kehitysmenetelmien joustavuus ja ketteryys eivät toimineet. Työntekijät käyttivät vanhoja menetelmiä ketterien kehitysmenetelmien lisäksi. (Kalenda, Hyna & Rossi, 2018).

Putta, Paasivaara ja Lassenius (2018) tutkivat hyötyjä ja haasteita SAFe-menetelmän käyttöönotossa. Tutkimus tehtiin kirjallisuuskatsauksena, jossa tarkasteltiin 52 artikkelia. Artikkeleista vain viisi olivat tieteellisiä julkaisuja ja 47 olivat muita julkaisuja. Artikkelit käsittivät vertaisarvioituja ja ei-vertaisarvioituja tapaustutkimuksia ja kokemusraportteja organisaatioista, jotka olivat käyttöönottaneet SAFe-kehitysmenetelmän. Kirjallisuuskatsauksen tuloksena esitetään SAFe-menetelmän käyttöönoton hyötyjä ja haasteita. Suurin osa haasteista löytyivät niin vertaisarvioiduista tutkimuksista kuin harmaasta kir-

jallisuudesta. Suurin ero vertaisarvioitujen ja harmaan kirjallisuuden löytyikin hyötyjen raportoinnissa. Kehitysmenetelmän käyttöönoton hyödyistä raportoi-tiin yksinomaan harmaassa kirjallisuudessa.

Haasteet jaettiin neljään kategoriaan: organisaatio ja kulttuuri, roolit, käy-tänteet ja skaalautuminen ja hajautuminen. Useimmiten kohdatut haasteet SA-Fe-menetelmän käyttöönotossa ovat

- muutosvastarinta
- haasteet ensimmäisessä hankkeen inkrementin suunnittelussa
- siirtyminen pois ketterästä.

Muutosvastarinnalla tarkoitetaan muun muassa muutoksen hyväksymisen vastustamista, muutos nähdään negatiivisena ja yhteisten työtapojen vastusta-mista. *Haasteet ensimmäisen inkrementin suunnittelussa* sisältää muun muassa tie-tämättömyyden PI:n tärkeydestä, tilaisuus koetaan epämukavana, kaoottisena ja vaikeana keskittyä. *Siirtymisellä pois ketterästä* -haasteella tarkoitetaan muun muassa sitä, että SAFe tuntuu perinteiseltä kehitysmenetelmältä ja menetelmä ei todella ole inkrementaalinen ja iteratiivinen.

SAFe-menetelmän käyttöönotossa on haasteita yhteensä 15, joista loput 12 haastetta ovat

- asenne
- autonomia
- suunnitelmakeskeinen tai perinteinen kulttuuri
- uusien roolien vastustaminen
- roolien täyttö
- arvovirtojen määrittely
- kehitysjonon hallinta ja toimintojen muodostaminen
- testiautomaatio
- kiista menetelmän kanssa
- ketterä toimitusjuna (ART)
- suuri ja hajautettu asetelma
- hajautettu organisaatio. (Putta, Paasivaara & Lassenius, 2018.)

3 KIPUPISTEMALLI

Tässä kipupistemallissa on tunnistettu erilaisia kipupisteitä, joihin esitetään ohjeita ja ratkaisuja tieteellisen kirjallisuuden pohjalta. Kipupistemalli muodostuu aiemmin luvussa viisi useimmiten kirjallisuudessa esiintyneistä skaalautuvien ketterien menetelmien menestystekijöistä ja haasteista sekä näihin kirjallisuudessa esitetyistä ohjeista ja suosituksista. Kipupistemalliin valitut ratkaisut ovat kirjallisuuskatsauksessa esiintyneiden tutkimusten tekijöiden antamia suosituksia.

Seuraavaksi esiteltävien skaalautuvien ketterien kehitysmenetelmien kipupisteiden avulla luodaan kipupistemalli. Tässä tutkielmassa käytämme menestystekijöistä ja haasteista yhteistä nimitystä kipupiste. Kipupisteen muodostamisessa esimerkiksi *johdon tuki* on menestystekijä, josta on johdettu kipupiste *johdon tuen puute*.

- *muutosvastarinta* (Dikert, Paasivaara & Lassenius, 2016; Kalenda, Hyna & Rossi, 2018; Conboy & Carroll, 2019; Putta, Paasivaara & Lassenius, 2018),
- *konseptien ja ketteryuden määrittelyyn* liittyvät haasteet (Dikert, Paasivaara & Lassenius, 2016; Conboy & Carroll, 2019; Putta, Paasivaara & Lassenius, 2018),
- *tiimin koordinointiin ja autonomiaan* liittyvät haasteet (Dikert, Paasivaara & Lassenius, 2016; Conboy & Carroll, 2019; Putta, Paasivaara & Lassenius, 2018),
- *uusien ja vanhojen toimintatapojen kohtaamiseen* liittyvät haasteet (Dikert, Paasivaara & Lassenius, 2016; Kalenda, Hyna ja Rossi, 2018; Putta, Paasivaara & Lassenius, 2018)
- *laadunvarmistukseen* liittyvät haasteet (Dikert, Paasivaara & Lassenius, 2016; Kalenda, Hyna ja Rossi, 2018; Putta, Paasivaara & Lassenius, 2018)

Tässä kirjallisuuskatsauksessa havaittuja useimmiten esiintyviä menestystekijöitä ovat puolestaan:

- johdon tuki (Dikert ym., 2016; Kalenda ym., 2018; Shameem ym., 2017; Paasivaara, 2017),
- viestintä (Dikert ym., 2016; Shameem ym., 2017; Paasivaara, 2017)
- ketterä koulutus ja valmennus (Dikert ym., 2016; Kalenda ym., 2018; Shameem ym., 2017; Paasivaara, 2017).

Haasteet ja menestystekijät vaikuttavat skaalautuvan ketterän kehitysmenetelmän käytön tai käyttöönoton onnistumiseen. Haasteet ja menestystekijät on tässä tutkimuksessa käännetty kipupisteiksi. Esimerkiksi menestystekijä johdon tuki on käännetty kipupisteeksi johdon tuen puute, viestintä on käännetty kipupisteeksi huono viestintä sekä ketterä koulutus ja valmennus on käännetty kipupisteeksi ketterän koulutuksen ja valmennuksen puute.

Kipupistemalliin valittiin kirjallisuuskatsauksessa useimmiten, 3–4 tutkimuksessa, esiintyneet menestystekijät ja haasteet. Harvemmin, alle 3 kertaa, esiintyneet jätettiin mallin ulkopuolelle. Vaikka osa haasteista ja menestystekijöistä jätettiin mallin ulkopuolelle, voi niiden huomioiminen olla oleellista tietyissä tilanteissa. Muita kirjallisuuskatsauksessa esiintyneitä menestystekijöitä tai haasteita, joita ei otettu mukaan kipupistemalliin ovat esimerkiksi

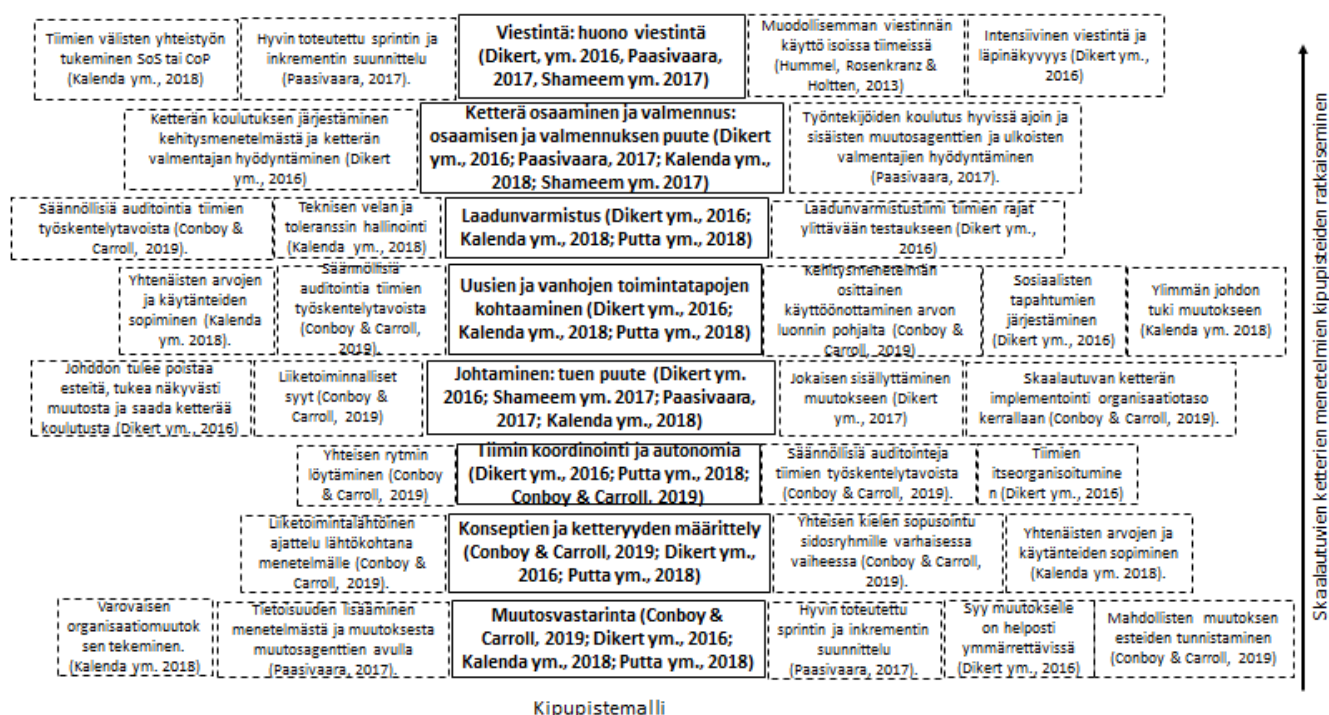
- *vaatimusten hallinta* (Dikert, Paasivaara & Lassenius, 2016; Paasivaara 2017)
- *kehittämisen ulkopuolisten toimintojen integrointi* (Dikert, Paasivaara & Lassenius, 2016; Putta, Paasivaara & Lassenius, 2018)
- *sosiaalisten tapahtumien järjestäminen* (Dikert, Paasivaara & Lassenius, 2016; Shameem, Chandra & Khan, 2017)

Tuoteomistajan roolin tärkeyden tunnistaminen vaatimusten hallinnassa on ensiarvoisen tärkeää. On tärkeää, että tuoteomistaja on riittävästi saatavilla ja hänen tulee olla omistautunut roolille. Tuoteomistajien tulisi saada valmennusta ketteristä periaatteista, kehitysjonon hallinnasta, käyttäjätarinoiden erittelystä ja ketterästä suunnittelusta. Tuoteomistaja tarvitsee tukea myös työkalujen käyttämisessä kehitysjonon hallintaan ja tiimin kanssa käytyyn kaksisuuntaiseen vuoropuheluun. (Dikert, Paasivaara & Lassenius, 2016.) Muista muutoksen onnistumisen kannalta tärkeitä tekijöitä ovat hyvä valmistautuminen ensimmäiseen PI-suunnitteluun, täysipäiväisen toimitusjunan päällikön (RTE) mukana oleminen ja asettamalla tunnistetuille kehitysehdotuksille vastuullisia henkilöitä ja seuraamalla niiden käyttöönottoa (Paasivaara, 2017).

Muiden osastojen haluttomuus muuttua ja vaikeus sopeutua ketteriin käytäntöihin on haasteellista *kehittämisen ulkopuolisten toimintojen integroinnissa* (Dikert, Paasivaara & Lassenius, 2016). Haasteena on täysin skaalata ketteryys koko organisaatioon ja globaaliin organisaatioon ja integroida muut kuin kehitysyksiköt, kuten esimerkiksi IT, HR ja myynti ja markkinointi (Paasivaara, 2017).

Sosiaalisten tapahtumien järjestämisellä pyritään mahdollistamaan skaalautuvan ketterän kehitysmenetelmän käyttöönoton onnistuminen. Sosiaalisten tapahtumien ideana on saada ketteryyden ajattelutapa iskostettua työntekijöi-

den työtapoihin (Dikert ym., 2019). Sosiaalisten tapahtumien järjestämisellä on mahdollista vähentää muutoksessa esiintyvää muutosvastarintaa indoktrinoidulla ketteryuden ajatuksen muutoksessa oleville henkilöille. Shameemin, Chandra ja Khanin (2017) toteuttamassa skaalautuvien ketterien kehitysmenetelmien menestystekijöiden kirjallisuuskatsauksessa sosiaalisten tapahtumien järjestäminen menestystekijänä ei nähty asiakkaan ja toimittajan puolelta yhteisenä, joten tämän vuoksi se on jätetty kipupistemallista pois.



KUVIO 8 Kipupistemalli

Kipupistemalli esitetään kuviossa 8. Keskellä pystysuunnassa esitetään edellä mainitut useimmiten kirjallisuuskatsauksessa esiintyneet kipupisteet. Vaakatasossa esitetään kullekin kipupisteelle niin ikään kirjallisuuskatsauksessa esiintyneet suositukset kipupisteiden ratkaisemiseksi. Seuraavissa alaluvuissa esitetään kipupiste ja suositukset niiden ratkaisemiseksi tarkemmin.

3.1 Muutosvastarinta

Vanhojen työskentelytapojen ja työkalujen unohtaminen ja uusien ketterien kehitysmenetelmien käyttöönotto voi aiheuttaa tiimien sisällä muutosvastarintaa. Myös liian nopeasti käyttöön otetut uudet toimintatavat ja kehitysmenetelmät eivät tukeneet skaalautuvan ketterän kehitysmenetelmän käyttöönottoa. (Kalenda, Hyna & Rossi, 2018). Muutosvastarintaan kuuluu yleinen haluttomuus muutokselle, skeptisyys uusia työtapoja kohtaan, ylhäältä alas ohjaus herättää vastustusta ja johdon haluttomuus muuttua (Dikert, Paasivaara & Lassenius, 2016).

Muutosvastarinnalla tarkoitetaan muun muassa muutoksen hyväksymisen vastustamista, muutos nähdään negatiivisena ja yhteisten työtapojen vastustamista (Putta, Paasivaara & Lassenius, 2018).

Organisaation muuttaessaan työtapojaan muutosvastarinnan muodostumista on mahdollista estää hitaalla ja huolellisella muutoksella. Skaalautuva ketterän kehitysmenetelmän käyttöönotossa huomioon tulee ottaa käytössä olevat resurssit muutoksen mahdollistamiseksi. Hitaassa muutoksessa työntekijöiden ajattelutapaa pyritään muuttamaan uusien toimintatapojen mukaiseksi. Liian nopea muutos voi aiheuttaa muutosvastarintaa. (Kalenda, Hyna & Rossi, 2018.) Organisaation tulee myös olla valmis muutokseen ja tunnistaa mahdolliset esteet skaalautuvan kehitysmenetelmän käyttöönotossa (Conboy & Carroll, 2019). Ihmiset ottavat muutoksen vastaan mieluummin, kun muutokselle on hyvä ja ymmärrettävä syy, joka on helppo oivaltaa (Dikert, Paasivaara & Lassenius, 2016). Organisaation ylimmän johdon mukana oleminen muutoksessa auttaa resurssien jakamisessa muutoksen mahdollistamiseen. Muutosvastarinnan voidaan olettaa olevan vähäisempää, mitä enemmän resursseja muutoksen mahdollistamiseen hyödynnetään. Tällaisia resursseja ovat esimerkiksi aika sekä koulutus. (Kalenda, Hyna & Rossi, 2018.)

3.2 Konseptien ja ketteryyden määrittely

Ketterä ajattelutapa on vaikea implementoida (Putta, Paasivaara & Lassenius, 2018). Konseptien ja ketteryyden määrittelyä hankaloittavat se, että konsepteja ei ymmärretä oikein, ohjaava kirjallisuus puuttuu ja ketteryys on huonosti kustomoitua (Dikert, Paasivaara & Lassenius, 2016). Myös Conyboyn ja Carrollin (2019) mukaan menetelmän soveltamisessa organisaation kontekstiin, viitekehksestä on vaikea löytää ohjeistusta.

Yhteisten näkemysten ja toimintatapojen sopimisesta yhdessä organisaation jäsenten kanssa konseptien ja ketteryyden määrittelyyn liittyviä ongelmat ovat vältettävissä (Kalenda, Hyna & Rossi, 2018). Organisaation tulee ymmärtää mitä ketteryyden skaalaamisella tavoitetaan, ja mitkä ovat motivaatiot sen takana. Ennen implementointia on myös suositeltavaa määritellä, mitä eri termit tarkoittavat omassa organisaatiossa ennen implementointia. (Conboy & Carroll, 2019.)

3.3 Tiimin koordinointi ja autonomia

Tiimin koordinoinnilla ja autonomialla viitataan tiimien yhteensovittamiseen, autonomiseen tiimimalliin, globaalisti hajautettuihin tiimeihin ja teknisen jatkuvuuden saavuttamiseen. Ketterän kehittämisen skaalautuessa suureneva organisaatio ja useamman tiimin yhteistyö herättää ongelmia. Vaikka tiimitasolla ketteryyden tuominen lisää ketteryyttä, sitä ympäröivä organisaatio ei välttä-

mättä ole tarpeeksi reagoiva. Ketteryys ei poista tiimien välisiä riippuvuuksia, mikä tekee kehityksen hallinnoimisesta vaikeaa. Autonomian haaste nousee esiin, jos tiimit keskittyvät vain omiin tavoitteisiinsa eivätkä huomioi laajempia organisaatiotason tavoitteita. Koordinointiongelmia tuovat myös tilanteet, joissa tiimit työskentelevät itsenäisesti laajemmasta kuvasta piittaamatta tai eri asiakkaille. Koordinointiongelmat lisääntyvät globaalin hajaantuneisuuden kasvaessa. Teknisen jatkuvuuden saavuttaminen on tärkeää tiimien yhteensovittamisen kannalta. Jos jatkuvuutta ei saavuteta, ongelmia voi tulla esimerkiksi tiimien tuotosten yhteensovittamisessa. (Dikert, Paasivaara & Lassenius, 2016.) Conboyn ja Carrollin (2019) mukaan puolestaan autonomian haasteena on kehittäjien autonomian säilyttäminen skaalatuissa ketterissä menetelmissä, kun autonomian säilyttäminen skaalatussa ketterässä on muutenkin vaikeaa ja skaalatut menetelmät kärjistävät ongelmaa entisestään tarjoamalla enemmän rajoituksia ja kankeutta. Myös Puttan, Paasivaaran ja Lasseniuksen (2018) mukaan tiimien autonomian puute on ongelma skaalautuvan ketterän menetelmän käytönotossa.

Ratkaisuehdotuksena skaalautuvan ketterän muutoksen läpiviemiseksi tiimeille tulee antaa täysi autonomia, jolloin tiimi sitoutuu muutokseen ja motivoituu jatkamaan menetelmän käyttöä. Se ohjaa tiimejä ottamaan omistajuuden kehitysprosessista ja jatkokehittämään sitä vapaaehtoisesti. (Dikert, Paasivaara & Lassenius, 2016.) Tiimien väleillä on myös hyvä toteuttaa sisäisiä katselmointeja, joilla varmistetaan tietoisuutta skaalautuvan ketterän menetelmän käytönotosta (Conboy & Carroll, 2019).

3.4 Johtaminen: tuen puute

Johdon tulee tukea muutosta poistamalla esteitä, olemalla näkyvästi mukana muutoksessa ja saada ketterää koulutusta (Dikert, Paasivaara & Lassenius, 2016). Organisaation ylimmän johdon mukana oleminen muutoksessa auttaa resurssien jakamisessa muutoksen mahdollistamiseen. Ylemmän johdon tuen saaminen mukaan alhaalta ylöspäin on hankalampaa (Kalenda ym., 2018), joten organisaatiossa tulee harkita, toteutetaanko muutos organisaation ylätasolta alaspäin vai alhaalta ylöspäin. Muutos tulee aloittaa taso kerrallaan, ja jokaiselle tasolle johdon tulee antaa tukea esimerkiksi koulutuksen muodossa. (Conboy & Carroll, 2019)

Johdon tuki kehitysmenetelmän käyttöön tai käyttöönottoon auttaa motiivoimaan myös työntekijöitä käyttämään ketterää kehitysmenetelmää (Kalenda ym., 2018). Mikäli motivointia ketterän kehitysmenetelmän käyttöön ei tule ylhäältä päin, niin se voi aiheuttaa ongelmia käyttöönotossa ja aiheuttaa esimerkiksi muutosvastarintaa.

3.5 Uusien ja vanhojen toimintatapojen kohtaaminen

Uusien ja vanhojen toimintatapojen kohtaamisella tarkoitetaan niitä ongelmia, joita muodostuu, kun uusia työ- ja toimintatapoja otetaan käyttöön organisaatiossa. Uusien ja vanhojen toimintatapojen kohtaaminen voi hankaloittaa uusien toimintatapojen käyttöönottoa, mikäli toimintatavat ovat päällekkäisiä tai ne jatkavat toimintaa uusien työtapojen rinnalla (Dikert, Paasivaara & Lassenius, 2018). Organisaatiossa tulisi löytää yhtenäiset näkemykset työtavoista, joilla töitä tehdään. Erilaisten roolien erot ja niiden vastualueiden selvittäminen auttaa uusien menetelmien käyttöönotossa. (Kalenda, Hyna & Rossi, 2018.) Muutos vesiputousmallista ketteräksi kehitysmenetelmäksi on haastavaa (Putta, Paasivaara & Lassenius, 2018).

Uusien toimintatapojen tulee myös tuottaa vähintään samanlaista arvoa kuin vanhojen menetelmien, jolloin myös johdon tuella on vaikutus tähän kippusteeseen. Erilaisten työskentelytapojen auditointien pitäminen voi auttaa havaitsemaan vanhoja toiminta- ja työskentelytapoja. Työskentelytapojen lisäksi valitun ketterän kehitysmenetelmän tulee olla organisaatioon sopiva, sekä prosessien tulee tukea liiketoiminnan tarpeita. (Conboy & Carroll, 2019.)

3.6 Laadunvarmistus

Laadunvarmistuksen ongelmat voivat liittyä testaukseen (Dikert, Paasivaara & Lassenius, 2016; Putta, Paasivaara & Lassenius, 2018) tai vaatimusten ja tarinoiden monitulkintaisuuteen (Dikert ym., 2016) tai tiimien kehitysprosessien ongelmiin (Kalenda, Hyna & Rossi, 2018). Laadunvarmistuksen ongelmat voivat liittyä esimerkiksi vaatimusmäärittelyn laajuuteen tai ne voivat olla monitulkintaisia. Testauksen ongelmat tai testaamattomuus voivat johtaa työn uudelleen tekemiseen. Mikäli testaustarve ylittää tiimirajat, on järkevää luoda testaukselle oma tiiminsä. Tämän laadunvarmistustiimin tulee määritellä työn koordinointi kehitystiimien kanssa. (Dikert, Paasivaara & Lassenius, 2016.)

Testauksen lisäksi myös hyvä teknisen velan hallinta ja toleranssitasojen asettaminen auttaa laadunvarmistuksessa (Kalenda, Hyna & Rossi, 2018). Hyvän teknisen velan hallinnan avulla ylimääräisen ja toistettavan työn syntyminen estetään ja huonolaatuisen koodin syntyminen estetään. Jokainen organisaatio ja työympäristö asettaa itselleen sopivat teknisen velan toleranssit, joista tulee pitää kiinni. Visuaalisten toleranssimittareiden hyödyntäminen voi auttaa toleranssitasojen ylläpitämisessä (Kalenda ym., 2018).

3.7 Ketterä osaaminen ja valmennus: osaamisen ja valmennuksen puute

Valmennuksen järjestäminen on skaalautuvan ketterän menetelmän menestystekijä (Shameem, Kumar, Chandra & Khan, 2017). Osaamisen ja valmennuksen puutteella tarkoitetaan muutoksen valmennuksen investoimisen puutteena, joka johtaa ketterän muutoksen vaikeutumiseen. Puute johtaa tiimien huonoon valmiuteen tehdä muutos, mikä puolestaan voi johtaa motivaation puutteeseen ja pahimmassa tapauksessa ketterien menetelmien hylkäämiseen. Lisäksi valmennus tulisi tuoda muutoksessa todelliseen työympäristöön eikä vain erillisille valmennussessioille. (Dikert, Paasivaara & Lassenius, 2016.) Myös Paasivaaran (2017) mukaan skaalautuvan ketterän kehitysmenetelmän käyttöönotossa työntekijöiden valmennus ennakkoon, tiedottaminen ja osallistaminen, muutosagenttien hyödyntäminen ja ulkoisten konsulttien palkkaaminen valmentamaan, neuvomaan ja tukemaan ovat tärkeitä seikkoja. Usein ketteryyden muutoksen vaikeuden aliarvioiminen, taloudelliset rajoitteet, johdon tuen puute tai muutoksen kiirehtiminen ovat syitä, jotka johtavat valmennuksen ja tiedon puutteeseen (Kalenda, Hyna & Rossi, 2018).

Valmentaminen edesauttaa ketterän muutoksen onnistumisessa ja se tulee tehdä hyvissä ajoin (Paasivaara, 2017). Tiimien valmentaminen ketterän muutoksen aikana on ensiarvoisen tärkeää, sillä ketteryys tavoitetaan parhaiten kokeilemalla. Sekä ulkoisten että sisäisten valmentajien hyödyntäminen on arvokasta (Paasivaara, 2017), sillä ulkoapäin näkee organisaation objektiivisesti ja sisäinen valmentaja puolestaan on paremmin saatavilla ja ymmärtää organisaation toiminnan paremmin. (Dikert, Paasivaara & Lassenius, 2016.) Ulkopuolisen valmentajan hyödyntämisen avulla on mahdollista löytää käytettävän kehitysmenetelmän ongelmia ja mahdollisesti ratkaista niitä muuttamalla toimintatapoja.

3.8 Viestintä: huono viestintä

Huonolla viestinnällä tarkoitetaan sellaista viestintää, joka ei tavoita tarvittavia sidosryhmiä. Viestinnän tulisi olla mahdollisimman läpinäkyvää ja jatkuvaa (Dikert, Paasivaara & Lassenius, 2016).

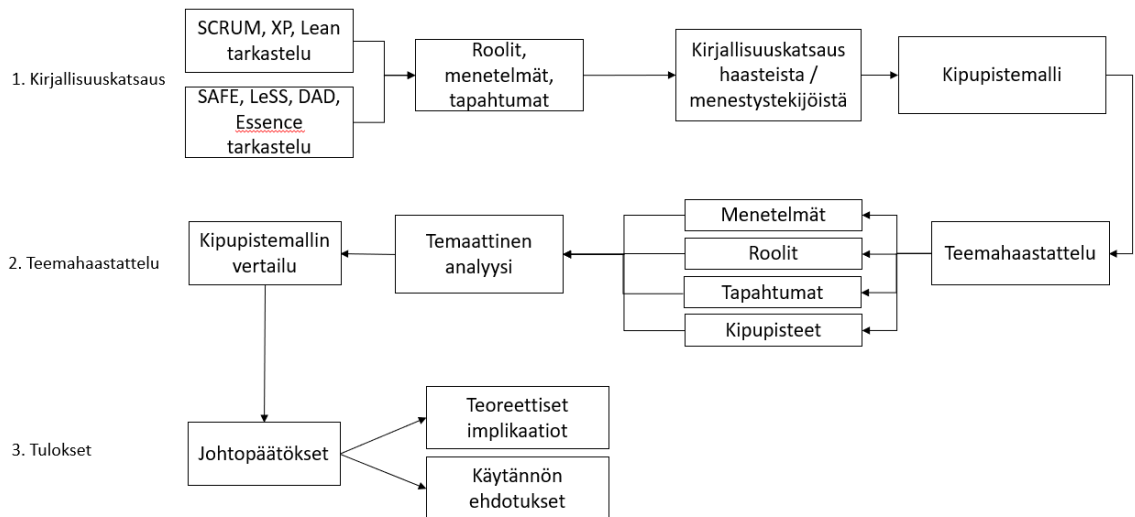
Viestinnän tapahtuvasta organisaatiomuutoksesta ja skaalautuvan ketterän kehitysmenetelmän käyttöönotosta ja muutoksesta tulee olla intensiivistä ja läpinäkyvää. Viestintää tulee tehdä useista eri kanavista, jonka tavoitteena on saavuttaa mahdollisimman monet sidosryhmät koko organisaatiosta ja sen ulkopuolelta. Jopa liiallinen viestintä tapahtuvasta organisaatiomuutoksesta on suositeltavaa. Myös muutoksen onnistumisien luominen ja niistä tiedottaminen on suositeltavaa. (Dikert ym., 2016.) Tiimien välisen yhteistyön tukeminen ja kommunikoinnin mahdollistaminen esimerkiksi scrum of scrumsilla tai käytän-

töjen yhteisöillä edistävät viestinnän onnistumista ja sen läpinäkyvyyttä (Kalenda, Hyna & Rossi, 2018).

4 TUTKIMUSMALLI- JA MENETELMÄ

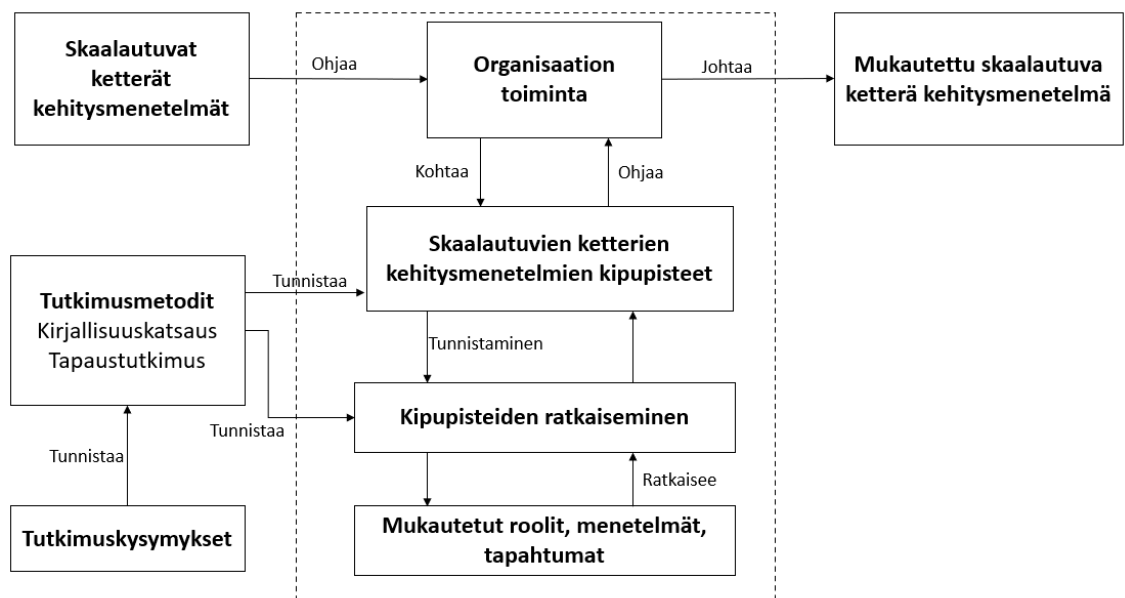
Tässä kappaleessa esitetään tämän pro gradun -tutkielman tutkimusmalli sekä tutkimusmenetelmä. Tämän tutkielman kappaleen tarkoitus on esittää, kuinka tutkimuksessa kerätään, käsitellään ja analysoidaan tietoa. Tässä kappaleessa esitetään erilaisia tutkielmalle sopivia tutkimusmenetelmiä, tutkimuksen datan keräämisen tapoja ja datan analysointiin liittyviä menetelmiä. Tutkielmaan valitun tutkimusmenetelmän jälkeen arvioimme tutkielman luotettavuutta ja validiteettia.

Tutkimusprosessi, joka on kuvattu kuviossa yhdeksän, on aloitettu tutustumalla tieteellisen kirjallisuuden ketteristä kehitysmenetelmistä. Tutkimukseen on valittu suurimmat ja yleisimmät tieteellisessä kirjallisuudessa mainitut kehitysmenetelmät, joita ovat perinteisistä ketteristä kehitysmenetelmistä scrum, kanban, XP sekä lean. Skaalatuista ketteristä kehitysmenetelmistä valittiin SAFe, LeSS ja DA. Essence on neljäs tutkielmaan valittu erityinen menetelmä, jota voidaan hyödyntää sekä pienessä että suuremmissa organisaatioissa. Kirjallisuuskatsauksen perusteella olemme löytäneet kehitysmenetelmien yleiset käytänteet, joita ovat erilaiset roolit ja tapahtumat. Ketterien menetelmien esittelyn jälkeen on toteutettu kirjallisuuskatsaus skaalautuvien kehitysmenetelmien käytön ja käyttöönoton menestystekijöistä ja haasteista. Eniten tieteellisessä kirjallisuudessa esiintyneet menestystekijät ja haasteet voidaan nähdä huomionarvoisimpina menestystekijöinä ja haasteina. Nämä menestystekijät ja haasteet ovat niitä tekijöitä, joiden vaikutus skaalautuvan ketterän kehitysmenetelmän omaksumiseen, käyttöönottoon ja käyttöön ovat kriittisiä. Valituista menestystekijöistä ja haasteista on muodostettu erityinen kipupistemalli. Menestystekijät on käännetty haasteiksi, jotka yhdessä haasteiden kanssa muodostavat kipupistemallin. Kipupistemalli sisältää myös tieteellisestä kirjallisuudesta löytyneet ratkaisuehdotukset.



KUVIO 9 Tutkimusprosessi

Kirjallisuuskatsauksen jälkeen on toteutettu tutkimuksen empiirinen osio. Teemahaastattelun avulla selvitetään tutkittavan kohdeorganisaation hankkeen menetelmät, roolit, tapahtumat ja kipupisteet. Haastattelujen temaattisen analyysin avulla ymmärretään hankkeessa esiintyneitä teemoja, jonka jälkeen teemoja vertaillaan luotuun kipupistemalliin. Lopuksi luodaan tulosten perusteella tutkielman teoreettiset implikaatiot ja arvioidaan miten tutkielman tulokset ovat linjassa tieteellisen kirjallisuuden kanssa sekä annetaan hankkeelle käytännön ehdotuksia.



KUVIO 10 Tutkimusmalli

Tutkimusmalli, kuvio 10, on luotu tutkimuksen kirjallisuuskatsauksen pohjalta. Tutkimusmalli pohjautuu ajatukseen, jossa skaalautuvat ketterät kehitysmenetelmät ohjaavat organisaation toimintaa. Organisaation toiminta voi

kohdata erilaisia tieteellisessä kirjallisuudessa tunnistettuja kipupisteitä, jotka estävät ketterien kehitysmenetelmien käyttöä tai käyttöönottoa, tai ohjaavat organisaation toimintaa kipupisteiden kautta. Kipupisteet ovat monisyisiä, eikä niiden ratkaisu ole aina yksiselitteistä. Kipupisteiden tunnistaminen ohjaa organisaation toiminnan muutosta ratkaisemalla kipupisteet muuttamalla rooleja, menetelmiä tai tapahtumia. Ratkaisut löytyvät siitä mistä toiminta koostuu: tapahtumista, menetelmistä sekä rooleista, sekä kirjallisuudesta esiintyneistä ehdotuksista. Kun organisaation toimintaa muutetaan kipupisteiden pohjalta, niin käytettävä skaalautuvan ketterän kehitysmenetelmä myös muovautuu tämän mukana. Ongelmaksi tulee kipupisteiden ratkaiseminen, eli mitä roolia, menetelmää tai tapahtumaa organisaatiossa muutetaan kipupisteen ratkaisemiseksi.

4.1 Tutkimusmenetelmä ja datan keruu

Tutkimusmenetelmät jaotellaan tyypillisesti kvalitatiivisiin eli laadullisiin ja kvantitatiivisiin eli määrällisiin menetelmiin. Kvantitatiivisia tutkimusmenetelmiä ovat muun muassa kyselytutkimukset, laboratoriokokeet, ekonometria ja numeeriset menetelmät, kuten matemaattinen mallinnus. Kvalitatiiviset tutkimusmenetelmät ovat kehittyneet sosiaalisten ja kulttuuristen ilmiöiden tutkimuksesta. Kvalitatiivisia tutkimusmenetelmiä ovat toimintatutkimus, tapaustutkimus, etnografia ja ankkuroitu teoria. Kvalitatiivisessa tutkimuksessa dataa kerätään tarkkailemalla ja osallistuvalla tarkkailulla, haastatteluilla ja kyselyillä, dokumenteilla ja tekstillä ja tutkijan vaikutelmalla ja reaktioilla. Kvalitatiiviset menetelmät mahdollistavat ihmisten ja heidän sosiaalisen ja kulttuurisen ympäristönsä ymmärtämisen. (Myers & Avison, 2002.)

Kvalitatiivinen haastattelu on käytetyin ja tärkein datan keräysmenetelmä laadullisessa tutkimuksessa. Kvalitatiivinen haastattelu voidaan jakaa strukturoituun haastatteluun, strukturoimattomaan tai puolistrukturoituun haastatteluun ja ryhmähaastatteluun. Kyselyissä, jotka eivät vaadi tutkijan läsnäoloa käytetään usein strukturoitua haastattelua. Ne ovat valmiiksi valmistettuja ja niissä ei ole tilaa improvisoinnille. Strukturoimattomassa tai puolistrukturoidussa haastattelussa on osittain voitu valmistella valmiita kysymyksiä, mutta haastattelussa on paljon tilaa improvisoinnille. Haastattelija on tutkija tai yksi tutkimustiimistä. Ryhmähaastattelussa voidaan käyttää sekä strukturoitua että strukturoimatonta haastattelua. Ryhmähaastattelussa yksi tai useampi haastattelija haastattelee kahta tai useampaa henkilöä. Tietojärjestelmätieteen kvalitatiivisessa tutkimuksessa näistä kolmesta useimmiten käytetään strukturoimatonta tai puolistrukturoitua haastattelua. Kvalitatiivisen haastattelun haasteita ovat esimerkiksi haastattelun keinotekoisuus, luottamuksen, ajan tai pääsyn puute, eliitin vinouma, Hawthorne-ilmiö, tiedon luominen, kielen monitulkintaisuus ja haastattelun pilalle meneminen. (Myers & Newman, 2007.)

Määrällisen eli kvantitatiivisen tutkimuksen avulla halutaan ymmärtää tutkittavan ilmiön syy- ja seuraussuhteita numeerisin tuloksin. Määrällisen tutkimuksen tarkoituksena on vastata lukumääriin liittyviin kysymyksiin, kuten

mikä, missä, kuinka usein tai kuinka paljon. Kvantitatiivisen tutkimuksen tärkein instrumentti on mittari. Ennen mittausta tutkittavat käsitteet tulee määrittellä. Käsitteiden määrittelyn avulla mitattavat ilmiöt selvennetään mitä ne tarkoittavat ja kuinka ne liittyvät teorian rakentamiseen. Muuttujien määrittelyn jälkeen seuraa operationalisointi, jossa muuttujille annetaan numeeriset arvot. (Singleton & Straits, 2018.) Arvottamisen avulla muuttujat tai muuttujaryhmät saadaan vertailtavaan muotoon. Määrällisiä aineistonkeruumenetelmiä ovat erilaiset lomakekyselyt, internet-kyselyt, puhelinhaastattelut, tarkasti strukturoidut haastattelut sekä systemaattinen havainnointi. Kvantitatiivinen tutkimus eroaa kvalitatiivisesta tutkimuksesta muun muassa haastateltavien lukumäärällä. Määrällisen tutkimuksen toteuttamistapana voi olla esimerkiksi kyselytutkimus, jossa tutkittavalle esitetään kysymyksiä kyselylomakkeen avulla. Kyselytutkimuksen tarkoituksena on antaa tilastollisesti merkittävää tietoa tutkittavasta ilmiöstä. Tutkimuksen onnistuminen riippuu kysymysten asettelusta, niiden ymmärrettävyydestä ja selkeydestä. Hyvässä kyselytutkimuksessa keskitytään sekä sisällöllisiin että tilastollisiin näkökohtiin. (Vehkalahti, 2008.)

Tässä tutkielmassa toteutetaan tapaustutkimus yhden organisaation ohjelmistokehityshankkeen menetelmistä, rooleista ja kipupisteistä. Tapaustutkimuksessa tutkimuksen tarkastelun kohteena on yksi tai useampi tapaus (Eriksson & Koistinen, 2005). Tutkimuksen keskeinen tavoite on tapauksen määrittely, analysointi ja ratkaisu. Tutkimus toteutetaan vapaamuotoisina teemahaastatteluiluina, jossa haastattelija tai haastattelijat esittävät keskustelun aiheen ilman ennalta asetettuja kysymyksiä. Teemahaastattelussa perehdytään haastateltavien käsityksiin ja näkemyksiin tutkittavasta ilmiöstä. Kvalitatiivisen tutkimuksen mukaisesti tavoitteena on ymmärtää tutkimuskohdetta ja tutkittavien näkemyksiä ilmiöstä. Haastatteluiden tarkoituksena on antaa vastaamisen vapaus haastateltavalle. Laadullinen tutkimusmenetelmä on perusteltavissa tutkittavan ilmiön luonteen vuoksi. Ilmiön tutkiminen kvantitatiivisin menetelmin on myös mahdollista esimerkiksi kyselyn muodossa, mutta kvalitatiiviset tutkimusmenetelmät mahdollistavat ihmisten ja heidän sosiaalisen ja kulttuurisen ympäristön ymmärtämisen (Myers & Avison, 2002).

4.2 Osallistujat

Kvalitatiivisen tutkimuksen tavoitteena on ymmärtää tutkimuskohteen ilmiöitä esimerkiksi haastattelun keinoin. Tutkimuksen tavoitteena ei ole haastatella jokaista organisaation jäsentä, vaan tarkoituksena on löytää tutkittavalle aineistolle merkityksellinen saturaatio, jolloin uutta tietoa ei synny (Hirsjärvi, Remes & Sajavaara, 2007). Tässä tutkielmassa haastateltiin seitsemää samassa ohjelmistokehityshankkeessa työskentelevää henkilöä. Haastatteluiden tiedot on esitelty taulukossa kolme.

Taulukko 4 Haastatteluiden tiedot

Tunniste	Haastateltavan rooli hankkeessa	Kesto
H1	johto	48 min
H2	johto	55 min
H3	kehittäjä/asiantuntija	45 min
H4	johto	57 min
H5	johto	27 min
H6	kehittäjä/asiantuntija	49 min
H7	johto	58 min

Tässä tutkimuksessa haastateltavat työskentelevät erään organisaation ohjelmistokehityshankkeessa erilaisissa rooleissa. Tällaisia rooleja olivat muun muassa erilaiset kehittäjät, asiantuntijat tai hankkeen johto. Ohjelmistokehityshankkeeseen kuuluu yhteensä kahdeksan ketterää tiimiä, mikä soveltuu skaalautuvien ketterien kehitysmenetelmien tutkimukseen. Kutsut haastatteluun lähetettiin hankepäällikön tarjoamalle listalle, josta halukkaat ilmoittautuivat haastatteluun. Osallistujille lähetettiin lomake, josta he valitsivat itselleen sopivimman haastatteluajan. Osallistujia haastatteluihin oli yhteensä seitsemän ja kullekin haastattelulle varattiin tunti aikaa. Haastatteluiden kestot vaihtelivat 27–58 minuutin välillä. Haastattelut järjestettiin etänä Microsoft Teams sekä Google Meetsin kautta. Ennen haastattelua haastateltavalle kerrottiin sekä haastattelun että tutkimuksen tarkoitus sekä yleisesti teemat, joista halutaan keskustella. Haastattelut nauhoitettiin Open Broadcaster Softwaren avulla myöhemmin tapahtuvaa litterointia varten ja tästä kerrottiin myös haastateltavalle. Haastattelijaille annettiin mahdollisuus tarkistaa haastattelun litterointi hyvän tutkimustavan mukaisesti.

4.3 Datan analysointi

Tutkimuksen johtopäätösten tekemistä varten kerätty aineisto litteroitiin. Litterointi toteutettiin nauhoitteiden avulla. Litteroinnissa haastattelijan kysymykset ja haastateltavan vastaukset kirjoitettiin sanatarkasti, vaikka tutkimuksen tarkoituksena ei ole eritellä kielellisiä tekijöitä. Litteroinnin avulla tutkimuksen dataa on mahdollista koodata, jonka avulla analyysia voidaan toteuttaa. Analysoimme lopullisen datan temaattisen analyysin avulla, sillä se soveltuu kvalitatiivisen datasta löytyvien teemojen tutkimiseen.

Temaattinen analyysi tunnistaa, analysoi ja raportoi datasta löytyviä kaavoja eli teemoja. Analyysi vähintäänkin organisoi ja kuvaa dataa yksityiskohtaisesti. Temaattista analyysia on usein käytetty menetelmä kvalitatiivisessa tutkimuksessa. Temaattinen synteesi puolestaan käyttää temaattisen analyysin periaatteita ja tunnistaa toistuvia teemoja tai asioita useista tutkimuksista, vetää johtopäätöksiä ja selittää nämä teemat systemaattisessa kirjallisuuskatsauksessa. Temaattisen synteessin prosessi alkaa datan tai tekstin lukemisella, josta sen jäl-

keen tunnistetaan tiettyjä segmenttejä. Sen jälkeen segmenteille annetaan nimi-tyksiä, jonka jälkeen päällekkäisyydet poistetaan ja koodaus käännetään teemoiksi. Tekstistä ja segmenteistä syntyy 30–40 koodia, joista puolestaan syntyy 15–20 teemaa. Viimeisenä luodaan malli korkeamman asteen teemoista. Teemoja syntyy 5–7. Temaattinen synteesi sisältää viisi vaihetta. Ensimmäisenä data erotetaan ensisijaisesta tutkimuksesta. Toisena mielenkiintoiset konseptit, kategoriat, löydökset ja tulokset tunnistetaan ja koodataan systemaattisesti koko datasta. Koodi käännetään teemoiksi, alateemoiksi ja korkeamman tason teemoiksi. Seuraavaksi korkeamman tason teemoille rakennetaan malli tutkimalla teemojen välisiä suhteita. Viimeisenä arvioidaan synteessin luotettavuus. (Cruzes & Dybå, 2011.)

Tutkielmassa litteroinnit koodattiin Googlen Sheets -työkalua hyödyntäen. Haastatteluissa käsiteltävät teemat erotettiin omille välilehdilleen. Tällaisia teemoja olivat roolit, menetelmät, kipupisteet ja etätyö. Yhtenäiset havainnot listattiin samalle riville, ja haastateltavat eroteltiin sarakkeittain. Ensimmäisen koodauksen jälkeen tehtiin ristiinkoodaus mahdollisten virheiden varalta ja tutkijoiden erilaisten tulkintojen välttämiseksi.

4.4 Tutkimuksen validiteetti ja luotettavuus

Tieteellisissä tutkimuksissa pyritään luomaan mahdollisimman luotettavaa ja toistettavia tieteellisiä löydöksiä, jotka ovat luotettavia ja valideja. Yleensä termit validiteetti ja reliabiliteetti liitetään kvantitatiiviseen tutkimukseen, mutta nämä ovat myös kvalitatiivisen tutkimukselle tärkeitä piirteitä. (Hirsjärvi, Remes & Sajavaara, 2007.) Kvalitatiivisen tutkimuksen luotettavuutta lisää tutkimuksessa kerrottu tutkimusmenetelmä ja sen eri vaiheet. Haastattelutilanteiden avaaminen, mahdolliset häiriötekijät, virhetulkinnat haastatteluissa ja tutkimusdataan liittyvät ongelmat tulee kertoa tutkimuksessa sen luotettavuuden korostamiseksi. (Hirsjärvi, Remes & Sajavaara, 2007.) Hyvän tutkimustavan mukaisesti käsittelemme myös tämän tutkimuksen validiteettia ja luotettavuutta kriittisellä näkökulmalla. Tutkielman luotettavuutta on pyritty lisäämään piirtämällä tutkielman tutkimusprosessi ja tutkimusmalli aiemmin tässä luvussa esitetyissä kuvioissa yhdeksän ja kymmenen. Tutkielman vaiheita ja etenkin datan keräystä ja analysointia on pyritty kuvaamaan mahdollisimman tarkasti.

Tutkimuksen toteutustapa on kvalitatiivinen tapaustutkimus. Tapaustutkimuksen luonteen vuoksi sen yleistettävyyden kyseenalaista, sillä tutkittavat organisaatiot voivat vaihdella useamman muuttujan perusteella. Tällaisia ovat esimerkiksi organisaation toimiala, organisaation koko, haastateltavien demografia sekä myös tutkijoiden omat asenteet ja näkemykset voivat vaihdella. Kvalitatiivisen tutkimuksen yleistettävyyden on yleensä mahdollista, mikäli tutkittavat ja vertailtavat organisaatiot ovat samankaltaisia. (Hirsjärvi, Remes & Sajavaara, 2007.)

5 TULOKSET

Tässä luvussa esitellään haastatteluiden tulokset. Luku on jaettu kahteen alalukuun, joista ensimmäisessä käsitellään hankkeen ketterää kehitysmenetelmää ja toisessa kipupisteitä. Haastatteluiden tuloksena esitetään kaksitoista primääriä empiiristä havaintoa.

5.1 Hankkeen ketterät kehitysmenetelmät

Haastatteluissa käytössä olevan kehitysmenetelmän kuvaukset sopivat soveltaen scrum, SAFe ja kanban -menetelmiin, jotka ainoina menetelminä myös mainittiin haastatteluissa nimeltä. Varsinaisesti hankkeessa käytetty kehitysmenetelmä on scrum, mutta hankkeessa on piirteitä myös kanban ja SAFe -menetelmistä. Kanban-taulun käyttö mainittiin työn edistymistä kuvaavana menetelmänä. Kaikkia SAFe:ssä esiintyviä menetelmiä ei kuitenkaan ole hankkeen käytössä. Siksi kehitystä tehdään hankkeessa soveltaen näitä menetelmiä. Usean menetelmän päällekkäin käyttäminen on järkeenkäypää, sillä SAFe hyödyntää scrumia, kuten myös kanbania ja omaa versiota XP:stä (Theobald, Schmitt & Diebold, 2019). Esiin nousi myös piirteitä, joita tämän tutkielman kirjallisuuskatsauksessa ei havaittu.

Scrum tunnistettiin varsinaisena kehitysmenetelmänä, jolla hankkeessa kehitystä tehdään.

“Scrum on kehitysmenetelmä. Tehdään määrämittäisinä - tässä hankkeessa kahden viikon pätkissä - töitä palasissa ja sovitaan ennalta, mitä kahden viikon aikana tavoitellaan ja speksataan sitä ennakkoon ja pyritään saamaan tehtyä ne siinä kahdessa viikossa. - - SAFehan skaalaa tätä tavallaan sitä scrumia useamman tiimin yli ja tehdään pitkäkestoisempaa suunnitelmaa.” (H1)

“scrum työmenetelmällä teen tiimien kaa töitä, niin se osa pyörii sen teorian kautta” (H4)

“No se [scrum] on tää viitekehys kautta malli, millä [hankkeen] kehitystä tehdään ja se määrittelee aika paljon arkea. Tietyllä tapaa kehys tekemiselle.” (H5)

EC1: Haastateltavat kokevat tiimien työskentelevän scrum-menetelmän mukaisesti.

Scrumista hankkeen kehitysmenetelmänä oli enemmän yksimielisyyttä haastateltavien kesken ja se tunnistettiin hankkeen kehitysmenetelmäksi. SAFe:n käytöstä oli myös yksimielisyyttä siinä mielessä, että täysin SAFe:a ei oppikirjan mukaan käytetä. SAFe:sta käytetään soveltaen yhteistä rytmiä ja julkaisusuunnittelua.

”SAFestahan me ollaan puhuttu, että on iso hanke kaiken kaikkiaan niin meillä on SAFe:maisia piirteitä, mutta ei tää niinku millään tavalla SAFe mallia tässä vielä ole lähelläkään.” (H2)

”Ei puhuta SAFe:sta. Meillä on oman tyyppinen tämä kehityskaari, mutta rakenteeltaan hyvin paljon muistuttaa SAFe:a.” (H3)

”No ei se SAFe:a oo oikeesti. Se julkaisusuunnittelu on tavallaan sellanen yks osa-alue, mikä on vähä siihen suuntaan, mutta se on se scrumi” (H6)

EC2: Hankkeen kehitysmenetelmässä koetaan olevan SAFe:maisia piirteitä. Nämä piirteet ovat tiimien yhteinen rytmi ja julkaisusuunnittelu.

Työjonon työvaiheiden seurantaan käytetään kanbania. Tehtävät työt siirtyvät kanbanin mukaisesti taululla käynnissä olevan työvaiheen mukaisesti vasemmalta oikealle. Haastatteluissa ei ollut mainintaa siitä, onko taulussa käytössä myös työmäärän rajoitus kullekin työvaiheelle eli sarakkeelle.

” Elikäs sarakkeittain, mitä on työjonossa. Onko jotain blokattuna, että on esteitä tämän kyseisen työtehtävän toteuttamiseen. Sitten työn alla ja validoinnit ja kun tiketti on liikkunut boardin vasemmasta reunasta oikeaan reunaan, niin siinä vaiheessa tämä taski on valmis.” (H3)

”Se oli siinä Jirassa sen niminen pohja. En minä tiedä, miten virallinen kanban se on. Se on vain esittelytapa siellä Jirassa. Sitä me ylläpidetään siksi, että testajat pystyisi sen julkaisun aikana nähdä, että mikä asia on toteutuksessa.” (H4)

EC3: Kanbania käytetään soveltaen tekeillä olevien töiden työvaiheen seurantaan.

PEC1: Hankkeen haastateltavat tunnistavat käytetyksi kehitysmenetelmäksi scrumin. He kertovat käytössä olevan myös SAFe:n ja kanbanin piirteitä.

5.1.1 Hankkeessa esiintyviä ketterien kehitysmenetelmien piirteitä

Seuraavaksi esitellään haastatteluissa esiin nousseita hankkeen kehitysmenetelmän ketterien menetelmien piirteitä. Haastatteluissa mainittiin kahden viikon sprintit. Sprinttien aikana tehdään kehitystyötä, suunnitellaan sprintti ja käydään tiimin kanssa läpi kuluneen sprintin aikaansaannokset ja retrospektiivi.

“Sprintille valittuja asioita tiimi toteuttaa kehitystyötä ja toki sprintti sisältää sitten myös näitä vakiintuneita tapaamisia, mitkä on dailyt joka aamu esimerkiksi ja mid sprintti planningit ja sitte lopussa sprint review:t, missä hyväksytään käytännössä näitä tiimin suorittamia tehtäviä tai toteuttamia kokonaisuuksia ja sitte on sprint planningit ja refinementit.” (H5)

“Sprinttiin meillä on demot, retrot, sitten plänäys, sitten synkronoidaan plänit, eli jokainen tiimi käy oman pläninsä läpi siinä.” (H1)

EC4: Hankkeessa tiimit suunnittelevat työnsä ja työskentelevät kahden viikon mittaisilla aikajaksoilla.

Sprintin aikana tehtävät työt nostetaan sprintille tuotteen kehitysjonosta eli backlogilta.

“[backlog] Lista tehtäviä asioita, joita halutaan tehdä, jotka on priorisoidussa järjestyksessä.” (H3)

EC5: Tiimien työjonoa hallinnoidaan tuotteen kehitysjonolla, josta tiimit ottavat työn alle priorisoidussa järjestyksessä.

Kehitysjonoa työstetään kehitysjonon kehityspalavereissa, jotka ovat luonteeltaan teknisiä eivätkä vaadi tuoteomistajan läsnäoloa. Kehityspalaverit ovat joustavia muutenkin osallistujien suhteen, sillä kehityspalaverihin voi osallistua muitakin tarpeen mukaan.

“refinement sessio, missä käydään backlogilla seuraavia asioita niin, että työstetään niitä eteenpäin, et on valmiina seuraavalle sprintin plänäykseen aiheita ja itemeitä, mitä siellä on – – pääasiassa ne refinementit on teknisestä näkökulmasta ja katotaan tiimin kanssa, mut case by case et ketä muita kutsutaan vierailemaan.” (H2)

EC6: Kehitysjonon kehityspalaverissa tehdään teknistä suunnittelua seuraavaksi työstettäville kehitysjonon asioille.

Hankkeen kehityksessä on päivittäin käytössä dailyt eli päiväpalaveri.

“me tavataan dailyssa joka aamu 15 minuuttia. Kaikki kertoo, mitä eilen teki ja mitä tänään tekee.” (H4)

“Se on joka aamu semmonen vartti viiva kolkyt minuuttia palaveri, missä kaikki tiimin jäsenet vuorotellen kertoo, mitä on tehnyt eilen, mitä tekee tänään ja onko jotain blokkereita tai selvitettäviä asioita. No siihenhän osallistuu kaikki tiimi jäsenet ja projektipäällikkö mahdollisuuksien mukaan.” (H5)

EC7: Tiimit kohtaavat päivittäin päiväpalavereissa, joissa tiimin jäsenet kertovat lyhyesti, mitä tekivät eilen, mitä tekevät tänään ja onko esteitä.

Sprintteihin kuuluu suunnittelupalaverit, joissa valmistaudutaan seuraavaan sprinttiin. Suunnittelupalaverit saattavat toteutua tiimeissä eri tavalla. Jotkut tiimit pitävät pidempiä suunnittelupalavereita kerralla ja jotkut tiimit pitävät lyhyempiä palavereita, mutta useammin. Suunnittelupalaverin jälkeen hankkeessa pidetään skaalattu suunnittelupalaveri, jossa tarkastellaan tiimien välisiä riippuvuuksia, sovitetaan kunkin tiimin suunnittelut yhteen ja tehdään tarpeen mukaan muutoksia suunnitelmaan. Koko tiimin sijaan skaalatussa sprintin

suunnittelupalaverissa tiimejä edustaa scrum masterit, projektipäälliköt ja tuoteomistajat.

“Sprinttisuunnittelu on seuraavan sprintin valmistelua ja työmääräarvioiden tekemistä niin ku, mitä taskeja otetaan, mitä storyja otetaan sprintille ja arvioidaan niiden työmääriä ja tarkennetaan sitä speksausta siinä suunnittelun yhteydessä. Sitä tehdään meillä nyt niinkun ainakin mun projektissa tiimit tekee sitä tämmösissä niinkun tavallaan ei ole yksi sessio, vaan on monta lyhyempää sessiota.” (H7)

“Perusrakenne on niinku sama, samat tehtävät tekee niissä, mutta toiset osa käyttää plänningiä pieneissä paloissa pienillä tiimeillä hyvinkin jatkuvasti kun osa taas tekee yksittäisiä pidempiä istuntoja, jossa on useampi paikalla.” (H1)

“synkkauksessa näytetään alkavan sprintin tulevat taskit muille näille tiimeille niin, että osaavat varautua ja jos tarvitaan, niin sitten järjestetään taskeja uudestaan. – – vaihtoehtoisesti siirretään seuraavaan sprinttiin tavaraa.” (H3)

“Eli tuossa synkrossa käydään läpi, kun jokainen tiimi on saanut sen sprinttisuunnittelun valmiiks, niin sitten jokainen esittelee omat pläninsä, koitetaan tunnistaa riippuvuudet. – – Scrumit ja PO:t ja projektipäälliköt [osallistuu].” (H1)

EC8: Sprintin suunnittelupalaverissa suunnitellaan seuraavalle sprintille työstettävät asiat.

EC9: Skaalatussa sprintin suunnittelupalaverissa tunnistetaan tiimien välisiä riippuvuuksia, sovitetaan suunnitelmat yhteen ja tehdään muutoksia tarpeen mukaan.

Haastateltavien kuvaukset retrospektiiveistä olivat yhteneväiset. Retrospektiiveissä tiimit arvioivat yhdessä kulunutta sprinttiä ja pohtivat kehitysehdotuksia.

“Vähä mietitään et mikä meni hyvin ja mikä sitten ei niin hyvin. Mietitään et missä voidaan kehittää.” (H7)

“...retro vielä kaikkein tärkeimpänä varmasti, koska retrospektiivissä kuitenkin katsotaan, että miten tämä edelline sprintti meni, mitä on parannettavaa, mitä nousi esille, mihin asioihin tarvii puuttua ja nostetaan action pointit elikkä tehtävät asiat, mitä halutaan parantaa.” (H3)

EC10: Tiimit käyvät läpi kuluneen sprintin onnistumisia ja kehitysehdotuksia retrospektiivissä.

Seuraavaksi esitellään haastatteluissa esiin nousseita hankkeen kehitysmenetelmän skaalautuvien ketterien menetelmien piirteitä. Hankkeen käytössä on julkaisusuunnittelu, jota hyödyntävät ja tunnistaa SAFen lisäksi DA ja skaalattu Essence (Ivar Jacobson International SA., 2021b; Theobald, Schmitt & Diebold, 2019). Hankkeessa julkaisusuunnittelu tapahtuu SAFE:a kevyempänä. Vaikka haastateltavat käyttävät eri termejä “kvartaalisuunnitelma” ja “release planning”, näillä viitataan julkaisun suunnitteluun eli samaan asiaan. Ero termien käytössä voi johtua siitä, että hankkeessa käytetään molempia termejä.

“PI-planning nimellähän se oikeasti menee, meillä puhutaan kvartaalisuunnitelmas-ta.” (H1)

“release planning on se termi, mitä täällä – – käytetään ja siinä valmisteli agile coach – – tuoteomistajien kanssa, että mikä se julkaisukokonaisuus on. Sitte otettiin mukaa

design ja arkkitehti ja scrum master ja sitte sen jälkeen suunniteltiin kaikki tiimit yhdessä se seuraava viis sprinttiä ja samallaahan big room planning henkisesti, ku esimerkiksi, jos on scaled agile frameworkki tuttu nii siinä SAFe:ssa on tällanen iha formaali prosessi vähän raskaampi vielä diiteilimpi, mutta se on vähä ottanu siitä heijastuksia” (H6)

Julkaisusuunnittelun lisäksi julkaisun puolessa välissä tarkastellaan, miten julkaisu on sujunut.

“releasin puolivälissä on semmone statustsekki, että missä siinä julkaussa mennää. Onko menny niiku on suunniteltu ja siinä voi vielä tarkentaa ja sit sinne ruvetaa suunnittelee seuraavaa” (H6)

EC11: Hankkeen käytössä kuvataan olevan käytössä julkaisusuunnittelu, jossa suunnitellaan seuraavien sprinttien muodostama julkaisu.

Sprintin aikana tehdyt asiat esitellään muille jokaisen sprintin ja julkaisun lopussa. Tilaisuuksiin on koko hankkeen laajuinen, mutta myös hankkeen rajat ylittävä jakelu. Sprinttidemot on jaettu kahteen osaan: tekninen ja toiminnallinen demo.

“sprinttidemo kahden viikon välein. Sinne on siis kaikki kehittäjät tai ketä tässä hankkeessa henkilöitä on suoraan kiinni ja sitten on kutsuttu toiminnasta referenssiryhmät eli sieltä tiettyjä henkilöitä, jotka on kiinnostunu [hankkeen nimi] kehityksestä eli heidät. Sitten on IT:stä ja yleensäki, jos on kiinnostusta niin henkilöitä on kutsuttu eli voivat osallistua sprinttidemoihin. Nää sprinttidemot on kahdessa osassa, eli on tekninen demo ja sitten on varsinainen toiminnallinen osa jossa on laajempi osallistumisjoukko. Eli siellä esitellään sitä toiminnallisuuksia, mitä näkyy käyttäjille ja sitten tekninen demo käydään teknisempiä kehitysasioita läpi, mitä tiimit on tehny ja kokeillut jotain teknologioita niin tän tyyppisiä asioita. Ja sitten systeemi demo on kvartaalin päätteeks eli käydään noin kolmen kuukauden kehitys ja esitellään kaikki, mitä siitä on tehty ja tähän viimeks oli kutsuttu koko toimiala eli oli laitettu intraan linkki eli saa osallistua jos haluaa tulla kuuntelemaan.” (H1)

“Näytetään mitä on tehty ihan yksinkertaisuudessaan. - - Demoihin osallistuu iso laaja jakelu.” (H3)

EC12: Hankkeessa on käytössä skaalattu sprintin tarkastelu, joka on jaettu tekniseen ja toiminnalliseen osaan. Hankkeessa ei ole tiimien omia sprintin tarkasteluita.

Retrospektiivien tuloksia tarkastellaan kaikkien tiimien edustajien ollessa paikalla.

“...käydään sitte vielä hankkeen kanssa yhdessä lävitte, että mitkä oli retron tulokset ja nostetaan nää isoimmat asiat siellä esille, mistä tiimi halus keskustella ja mistä tuli action pointeja niin, että muutkin saavat siitä ideaa ja meille itselle voidaan kehittää meidän tiimin toimintaa eteenpäin.” (H3)

EC13: Tiimien retroissa esiin nousseita action pointeja tarkastellaan skaalatussa retrospektiivitapahtumassa, jossa kaikki tiimit ovat edustettuina.

Haastateltavaa kertoivat, että hankkeessa on dailyjen lisäksi päivittäin scrum of scrums -palaveri. Tässä 15 minuutin palaverissa on päivittäin mukana eri tiimien jäseniä, kuten scrum masterit ja tuoteomistajat.

“Meillä on myös dailyjen jälkeen myös SOS eli scrum of scrums. Eli siellä me sitten tavataan ja jos siellä tulee jotain ongelmia tai jotain ristiriitoja, mitä pitää katsoa siis yhteen, tai että on tulossa jotain, niin siellä sitten scrumien kanssa yhdessä pohditaan ettei koko tiimin asiaa tarvii pohtia, vaan nää pohtii sitä asiaa siellä ja miettii jatko-toimenpiteitä.” (H4)

“Meillähän on sitten scrum of scrumit, jossa kerrotaan sitten täsmällisemmin, mitä tiimi on tekemässä.” (H1)

EC14: Hankkeessa on käytössä päivittäin scrum of scrums -tapahtuma, jossa keskustellaan tiimien tekemisistä ja ratkaistaan tiimien välisiä riippuvuuksia.

PEC2: Käytettävää ketterää kehitysmenetelmää on skaalattu hankkeen tarpeiden mukaisesti. Havaitsemme hankkeessa olevan skaalautuvien ketterien kehitysmenetelmien, kuten SAFE ja LeSS, piirteitä.

5.1.2 Hankkeen erityispiirteitä

Seuraavaksi esiteltävät tapahtumat esiintyvät hankkeessa, mutta eivät kirjallisuuskatsauksessa. Hankkeessa toteutetaan julkaisun viimeisellä viikolla refaktorointiviikko. Se toimii hankkeessa pääasiassa laadunvarmistuksena tulevaan julkaisuun. Kuten sitä ei skaalautuvissa ketterissä menetelmissä esiinny, myös sen olemassaoloa hankkeessa kyseenalaistetaan ja tunnistetaan haasteet. Toisaalta toinen haastateltava pitää refaktorointiviikkoa toimivana ratkaisuna, kunhan viikko pysyy refaktoroinnille tarkoitettuna. Toisaalta toisen näkemys on se, että refaktorointi tulisi sisällyttää koko julkaisun mitalle sprintteihin eikä viimeiselle julkaisun viikolle.

“kvartaalin viimeinen sprintti on viikon mittainen ja se on refaktorointisprintti nimellä mennyt, mut se on enemmän vaan laadunvarmistussprintti oikeesti niin siinä oikeesti viimeistellään mitä on tehty.” (H1)

“[refaktorointiviikko] on mun mielestä ihan hyvä ja toimiva. Tietenki siinä on ne haasteet, että jos on aikataulupaineita, niin helposti refaktorointiviikolle joudutaan ottamaan myös toiminnallisuuksien kehittämistä, mikä ei lähtökohtasesti ehkä kuulu sinne, mutta välillä näin joutuu tekemään.” (H5)

“En henk. kohtaisesti näe, että se on mikään paras ratkaisu – – refaktorointia pitäisi sisällyttää normaaliin sprinttityöhön. Pitäisi plänätä niin, että siinä on jonkinlainen kaista jätetty sille refaktorointityölle myös. – – Sinne [refaktorointiviikko] saattaa kertyä liikaa asioita, vaikka on jätetty se aika refaktorointiin niin ei pysty sitten keskittymään itse refaktorimaan, vaan he tekevät itse sprinttityötä. Eli se menettää itse merkityksensä silloin.” (H2)

EC15: Hankkeessa on käytössä laadunvarmistuksena refaktorointiviikko, joka toteutetaan jokaisen julkaisun viimeisellä viikolla. Refaktorointiviikolle otetaan sen idean vastaisesti myös uusien toiminnallisuuksien kehittämistä.

Etätyöskentelyajan tuoman haasteen vuoksi yhteisen ajan löytäminen tuoteomistajan ja tiimin jäsenten kanssa on hankalaa, kun ei enää työskennellä samassa tilassa. Tämän vuoksi tuoteomistaja on ottanut käyttöön PO ajaksi kutsutun tapahtuman, jossa tuoteomistaja vastaa tiimiläisten kysymyksiin ja puolin ja toisin esitellään uusia asioita.

”nyt kun me ollaan etänä, niin sehän on taas tuonut uusia haasteita. Niin me ollaan tehty niin, että mä tapaan tiimiä tällaisessa PO aika. Eli kolme kertaa viikossa tunnin verran on yhdelle tiimille ja toiselle aikaa. Eli mulla menee kuus tuntia viikossa, jotka on dedikoitu heille. Ja siellä he näyttää mulle, mitä he ovat tehneet, onko heillä kysymyksiä mulle tai sitten minä esittelen jotain tulevaa heille tai jotain asiaa, mitä on selvitetty niin käydään se läpi tai jos tuotannossa on bugeja.” (H4)

EC16: Etätyön takia, kun tiimi ei työskentele samassa tilassa, hankkeessa on otettu käyttöön PO ajaksi kutsuttu tapahtuma, joka on säännöllinen tiimin ja tuoteomistajan välinen aika.

Hankkeessa on käytössä säännöllinen arkkitehtuurikokous, jota käydään sovellusarkkitehdin johdolla tiimien scrum mastereiden ja arkkitehtien kanssa läpi. Kokouksessa suunnitellaan hankkeen arkkitehtuuria ja sovitetaan tiimien yhteiskäyttöisiä asioita yhteen.

”Tiimien välillä on ne arkkitehtuurikokoukset kerran viikossa, missä sitten ne keskustelee teknisistä asioista, jos tulee jokin iso asia.” (H4)

”säännöllisesti pyörii arkkitehtuurimiitit, missä käydään sovellusarkkitehdin vetämänä kaikkien tiimien scrumit tai sitten, jos tiimissä on erillinen arkkitehti, niin niiden kanssa sitten kahden viikon välein taitaa olla tällä hetkellä kokoontuminen. – – jos on yhteiskäyttöisiä alueita, niin siellä on sitten sorvattava niitä yhteisesti ja mietittävä yhteisesti, miten se sopii kaikkien tiimien tekemiseen.” (H2)

EC17: Arkkitehtuurikokouksissa käydään yhteiskäyttöisiä asioita läpi säännöllisesti.

PEC3: Käytettävää ketterää kehitysmenetelmää on muokattu hankkeen tarpeiden mukaisesti.

5.1.3 Hankkeen roolit

Teemahaastattelussa haastateltavat kertoivat omin sanoin hankkeessa esiintyvistä rooleista ja heidän vastuistaan hankkeessa. Teemahaastattelun pohjalta tunnistettiin useimmin esiintyneet roolit. Tällaisia rooleja olivat mm. tuoteomistaja, tekninen tuoteomistaja, scrum master, projektipäällikkö, tiimin jäsen sekä hankepäällikkö. Tutkimuksen kannalta nämä ovat oleelliset havainnot. Osassa haastatteluissa esiintyi myös rooleja, joita ei ilmennyt muissa haastatteluissa.

Haastatteluissa eniten esille nousut rooli on tuoteomistajan rooli. Tuoteomistajan roolin mainitsivat jokainen haastateltava. Tuoteomistajan vastuu sekä

työtehtävät olivat jokaisen haastateltavan osalta yhdenmukaiset, mutta osaltaan ne ovat myös päällekkäiset projektipäällikön tehtävien kanssa. Tuoteomistajan tehtävään kuuluu vahvasti ymmärrys asiakkaan puolella tapahtuvasta toiminnasta ja heidän tarpeiden ja toimintaprosessien määrittelystä.

“Tuoteomistaja on käytännössä miten mää miellän sen niin se on linkki tuonne asiakkaan suuntaan täältä tiimistä, joka hoitaa niitä tehtäviä asioita mitä meille tulee työn alle ja priorisoi niitä meille, että mitkä on asiakkaan kannalta tärkeitä” (H3)

“Scrum työmenetelmällä teen tiimien kaa töitä, niin se osa pyörii sen teorian kautta. – – Siis käydään keskustelua [asiakkaan] kanssa siitä, että miten ne haluaa, että nämä prosessit toteutetaan sähköisesti.” (H2)

“Tuoteomistaja tarkoittaa roolia, jossa määritellään asiakastarpeita ja ohjataan tiimien työskentelyä, että mennään oikeeseen suuntaan määrittelyssä. Iso osa [tuoteomistajan] työstä menee asiakastarpeiden määrittelyssä.” (H7)

“[featuren määrittely] pääasiallisesti se on tuoteomistajan vastuulla, mutta tavallaan se on hyvä suunta, että siihen osallistuu enemmän myös se tiimi.” (H6)

Tuoteomistajan vastuulla nähdään olevan tiimien työskentelyn ohjaaminen sekä asioiden priorisointi. Tuoteomistajat määrittelevät ja priorisoivat mitä sprintin aikana tulee saada valmiiksi, ja mikäli asiakkaan suunnalta tulee muutoksia, niin tuoteomistaja ohjaa tiimit työskentelemään haluttuun suuntaan.

“Tietysti [tuoteomistajana] priorisoin asioita, että missä järjestyksessä, millä prioriteetilla asioita tehdään. Sitten tietysti sen myötä oon mukana siinä sprintin suunnittelussa, että mitä sprintille otetaan. Ja tietysti mun kautta menee kaikki tikettien hyväksynät.” (H7)

EC18: Hankkeessa on rooli tuoteomistaja, jonka haastateltavat kokevat vastaavan asiakastarpeiden määrittelystä.

Scrum ei ketteränä kehitysmenetelmänä sisällä projektipäällikön roolia. Scrumissa projektipäällikön roolin on korvanut scrum master, joka hallinnoi scrumin toteutusta. (Schwaber, 2004.) Tutkittavassa hankkeessa sisällytetään projektipäällikö osaksi scrumin kehitysmenetelmää osaltaan hankkeen luonteen vuoksi. Tuoteomistajan ja projektipäällikön erosta haastateltavat kertoivat seuraavaa:

“Tuoteomistaja on lähempänä siinä tiimiä ja määrittelyä. Tehtävänä on luoda niitä määrittelyitä, tarpeita ja luoda sitten speksiä siltä kantilta. Projektipäällikön rooli ehkä pitäs yleisesti olla hieman ylempänä, eli vastata talouspuolesta esimerkiksi ja tämmöstä siinä sivussa seurata ja toki tarvittaessa tehdä niitä priorisointeja, mutta jonkin verran tehdään ristii näitä asioita, riippuu onko projektipäällikköä, onko yksi tuoteomistaja vai onko yksi, joka hoitaa molemmat [tehtävät]” (H5)

“Projektipäällikkö tarkastelee ylätasolla niitä tunteja, ja hoitaa niitä byrokraattisia puolita niiden hankintojen puoleen – – ja hän hoitaa sitä puolta hankepäällikön kanssa ja katsoo miten tehokkaita ne [tiimit] on. – – Hän ottaa myös kantaa näihin isoihin linjauksiin, isoihin määritelmiin, huolehtii siitä, että aikataulut pitää isosti.” (H4)

“Hänellä on sitä teknistä osaamista mitä mulla ei [tuoteomistajana] oo.” (H4)

EC19: Hankkeen projektipäällikön työtehtävänä nähdään olevan kannan ottaminen hankkeen linjauksiin ja aikatauluihin.

Hankkeeseen sisältyy myös tekninen tuoteomistaja. Teknisellä tuoteomistajalla kerrotaan hankkeessa olevan samanlainen rooli kuin tuoteomistajalla, mutta teknisellä tuoteomistajalla on enemmän teknistä osaamista, jonka voidaan nähdä auttavan määrittelyiden toteutuksessa. Teknisestä tuoteomistajasta kerrottiin haastatteluissa seuraavaa.

”minä tuoteomistajana tunnen toiminnan, palvelun ja substanssin hyvin, niin hän sitten auttaa siinä teknisessä speksauksessa sitten vaatimusten kanssa. Ja tukee sitten tietysti tiimejä vielä vahvemmin. – – Aika pitkälti työparityöskentelyä nyt on, hänellä on se tekninen näkökulma ehkä mitä mulla ei oo. Hän osaa ehkä esittää siitä näkökulmasta mulle kysymyksiä vielä tarkemmin, jolloin ne feature-vaatimukset ja speksaukset on tarkemmalla tasolla valmiimpia tiimeille.” (H7)

”Tekninen tuoteomistajuus vastaa arkkitehtuurista pitkälti. Teemme kestäviä ratkaisuja ja sitähan se paljon on. Paljon on uudelleen hyödynnettäviä osia.” (H1)

EC20: Tekninen tuoteomistajan vastuuna koetaan tekninen ja arkkitehtuurillinen määrittely.

Tekniset tuoteomistajat ja projektipäälliköt nähdään hankkeessa teknisinä osajina. Projektipäällikön ja teknisten tuoteomistajan eroavaisuuksista ja yhte-neväisyyksistä kerrottiin seuraavaa.

”Projektipäällikköhän on niinku me tuoteomistajat, et kun me ollaan loppupeleissä substanssiosaajia, niin hän on tekninen päällikkö, joka tekee teknistä määrittelyä ja teknistä suunnittelua – – omassa projektissa ei ole projektipäällikköä, mutta on tekninen tuoteomistaja, niin mikä on heidän työnkuvansa ero niin siihen en osaa oikein vastata.” (H7)

”[asiakkaan] osa-alueelle tullut oma tekninen tuoteomistaja, joka auttaa siinä määrittelytyössä. Ja sitten tälle [asiakkaalle] on tullut uusi projektipäällikkö helpottamaan tuoteomistajien työkuormaa, siinä on nyt enemmän ihmisiä tekemässä sitä asiaa. Sen lisäksi tiimit on otettu mukaan siihen feature-tason määrittelytyöhön.” (H4)

Projektipäällikön lisäksi hankkeeseen sisältyy hankepäällikkö, joka vastaa koko hankkeen laajuisesti sen etenemisestä. Hankepäällikkö vastaa hankkeen lopullisesta toteutumisesta.

”Hankepäällikkö vastaa koko hankkeen kaikkien projektien etenemisestä ja resursoinnista ja toimittajan kanssa sopimusten kanssa. Kaikkee tän tyyppistä. Kokonaisuus menee eteenpäin ja siihen liittyvät sitten myös niinku johdon kommunikointi on tärkeä osa sitä.” (H1)

”Tarkastelee isosi näiden tiimien työskentelyä, eli jos mun projektipäällikkö tarkastelee mun kahta tiimiä, niin hänen täytyy seurata koko hankkeen tiimiä vielä ylemmältä tasolta – – Hän on vastuussa kaikesta. Jos me mokataan, niin hän ottaa vastuun siitä. Luotamme siihen.” (H4)

EC21: Hankepäällikön koetaan vastaavan hankkeen suunnasta ja lopullisesta toteutumisesta.

Haastatteluissa keskusteltiin ketterästä tiimityöskentelystä ja työnohjauksesta. Sprintin aikana kehitettävistä asioista päättää tuoteomistaja yhdessä projektipäällikön kanssa, mutta tiimien koordinoinnissa ja ohjauksessa auttaa scrum master, jonka mainitsi jokainen haastateltava.

“Projektipäällikkö vastaa laajemmasta kokonaisuudesta yhdessä tuoteomistajan kanssa, priorisoinnista ja tosiaan laajemmasta teknisestä sopivuudesta. Scrumilla on enemmän se tiimin ohjausvastuu.” (H1)

“Tiimiin kuuluu aina projektipäällikkö, joka kattoo heidän peräänsä. Tiimissä on siten yksi scrumi, joka sitten hoitaa sitä Jira puolen jumppaa ja ohjaa sitä tiimiä agile scrumimaisesti.” (H2)

“[scrum master] pitää huolen siitä, että ihmisillä on töitä, pitää huolta siitä että backlogilla asiat on pisteytetty mikäli prioriteeteista on kysymys. Sitten pitää olla yhteydessä tuoteomistajaan ja pitää dailyt suunnitella, että minkälaisia työkokonaisuuksia me voidaan ottaa seuraavalle kahdelle viikolle ja myöskin eteenpäin siitä. Pidetään tän tiimin työkuorma hallittavana” (H3)

“[scrum master] johtaa sitä dailya ja katsoo, että oma tiimi tekee niitä tehtäviä mitä on luvannut tehdä. Auttaa myös koordinoinnissa, et jos siellä joku tarvitsee apua niin katsoo, että myös apua saadaan.” (H4)

“Scrumit on kehitystiimien vetäjiä, eli he ovat enemmän nimenomaan tässä suoraan mun kanssa suunnittelemassa niitä tulevia sprinttejä ja tietysti release suunnitteluissa. -- Pitkästi heidän aikansa menee myös suunnittelupalaverissa. -- Mut scrumeilla on vähemmän itse koodaamisaikaa, niin heillä menee enemmän aikaa siihen speksaamiseen ja suunnitteluun myös.” (H7)

EC22: Scrum master nähdään vastaavan tiimin päivittäisestä ohjauksesta ja ottaa kantaa arkkitehtuuriin linjauksiin.

Hankkeessa on yhteensä kahdeksan tiimiä, jotka sisältävät 8-12 henkilöä. Tiimit jäsenet toimivat eri rooleissa, joista fullstack-kehittäjiä on tiimeissä eniten. Tiimeihin sisältyy myös yksi scrummaster, arkkitehti, tuoteomistaja, designeri, projektipäällikkö ja testaaja. Testaajat ja käyttöliittymäsuunnittelijat ja designerit muodostavat omat tiiminsä, joista käsin he auttavat muita kehitystiimejä.

“Tässä hankkeessa oli siis tuoteomistaja, ja sitten konsultti-scrum tiimi. Scrummaster, designereita ja devaajia se tietty määrä aina.” (H1)

“Tiimi on tietylle osa-alueelle tehty kokonaisuus, missä on tietty määrä kehittäjiä. Tiimikohtaisesti vähän vaihtelee, mutta siellä on fullstack-kehittäjiä niinku 4-8 kappaletta saattaa olla tiimissä. Tiimiin kuuluu aina projektipäällikkö, joka kattoo heidän peräänsä ja tiimissä on sitten yksi scrumi. -- Ja sitten tiimejä meillä on tukemassa UX-poolista olevia kavereita. Uxilaiset ei oo tiimeissä, vaan on erillinen UX-tiimi, jossa on jaetut resurssit kaikkien tiimien kesken tarpeiden mukaan. Yhtä lailla meillä on jaettu testauspooli sillain, että testaajat ovat sitten yhteiskäyttöisesti kaikkien tiimien käytettävissä.” (H2)

“Tiimien koosta mä oon ollu aika tyytyväinen, et meillä ei oo ollu liikaa ihmisiä, koska liikaa ihmisiä niin sitten korostuu kommunikaation vaikeus. Jos on liian vähän niin sitten ei oo työnalla tarpeeksi asioita.” (H3)

EC23: Tiimi sisältyy 4-8 kappaletta fullstack kehittäjiä, scrummaster, tuoteomistaja sekä designer.

EC24: Testaajat muodostavat oman tiiminsä.

EC25: Käyttöliittymäsuunnittelijat ja designerit muodostavat oman tiiminsä.

Hankkeessa on mukana myös ketterä valmentaja, jonka tarkoitus on auttaa tiimejä työskentelemään ketterämmin. Ketterän valmentajan tarkoituksena on kehittää hankkeen ketteriä toimintatapoja ja hän avustaa niiden läpiviennissä. Ketterä valmentaja myös tekee julkaisusuunnittelua.

”Meillä on yks agile coachikin mukana tässä hankkeessa, hän toimii myös – – teknisenä tuoteomistajana. – – Kerran sprintissä silloin kun käydään toimintatapojen kehitystä läpi, scrumit, projektipäälliköt ja tuoteomistajat osallistuu näihin projektiryhmiin, etsitään sellaisia asioita mitä tulisi parantaa, ja tässä agile coach on vahvasti mukana. Eli mitä asioita sieltä lähdetään hakemaan ja mitä tulisi parantaa, hän avustaa niitten läpiviennissä.” (H1)

” [ketterä valmentaja] pyörittää tota julkaisusuunnitteluprosessia, että miten se menee. Ja sitten näitä toiminnan kehitysasioita, sen lisäksi toimii tuoteomistajien tukena näissä eri featureitten määrittelytyössä. – – Ehkä se miksi [ketterä valmentaja] roolissa on jotain järkeä, on se että on saatu sitä designia ja julkaisusuunnittelua parannettua, ja sit on saanu autettua tuoteomistajia niitten työkuormassa” (H6)

EC26: Ketterä valmentajan nähdään olevan mukana ketterien toimintatapojen kehittämisessä.

Hankkeessa tuoteomistajalla on kaksi tiimiä johdettavanaan. Testaajilla sekä käyttöliittymäsuunnittelijoilla on oma yksittäinen tiiminsä, joista heitä lainataan tiimeihin tarpeen mukaan. Designerit, joita osa haastateltavista kutsui palvelumuotoilijaksi, liikkuvat myös tiimien välillä. Designerien rooliin kuuluvat palveluiden suunnittelu ja prosessien määrittely. Käyttöliittymäsuunnittelusta vastaavat henkilöt toteuttavat kehittäjille suunnitelmat ohjelmiston ulkonäöstä, jota fullstack-kehittäjät toteuttavat. Kehitys- ja testausympäristöä ylläpitää erillinen infratiimi.

”Meillä on erillinen infratiimi, jossa on ylläpitäjä, mutta kaikki tiimit osallistuu sen suunnitteluun ja sen tarvittavaan ylläpitoon. Mutta infratiimi vastaa siitä CI/CD-putkesta.” (H2)

”Meillä menee periaatteessa suoraan tuotantoon mitä kehitämme tässä, eli automaatiotestaus on osa DevOpsia. Tietysti koodikatselmointi ja koodin laadun varmistaminen, ja sitten manuaalitestaukset myös automaattitestien jälkeen. Meillä on useampi ympäristö, jossa koodi testillaan eri työkaluilla, ja manuaalisesti, ja jos otamme koodin hyväksi niin se menee tuotantoon.” (H1)

EC27: Erillinen DevOps- /infrastruktuuri tiimi vastaa hankkeen jatkuvan integroinnin ja julkaisun putkesta.

Hankkeen arkkitehtuurisista ratkaisuksista vastaa sovellusalueen arkkitehti, joita hankkeessa on yksi. Hänen päätehtävänä on katsoa hankkeen arkkitehtuurin kokonaisuutta. Arkkitehtuurista osaltaan vastaavat myös scrum masterit,

jotka ovat arkkitehtitasoisia kehittäjiä. He katsovat yhdessä tiiminsä kanssa alueen arkkitehtuurillista kokonaisuutta.

”Meillä on periaatteessa hankkeessa yksi sovellusarkkitehti, joka katsoo koko arkkitehtuurin, sitä isoa arkkitehtuurin näkökulmaa. Jokaisessa tiimissä on oma scrumi otettuna sen mukaan, että he ovat arkkitehtuurillisesti päteviä kavereita. He katsovat oman tiimin ja työstöalueen arkkitehtuuria kokonaisuuteen liittyen.” (H2)

EC28: Sovellusarkkitehti nähdään vastaavan yhdessä scrum mastereiden kanssa koko hankkeen leikkaavista arkkitehtuurillisista linjavedoista.

PEC4: Hankkeessa hyödynnetään skaalautuvissa ketterissä kehitysmenetelmissä esiintyviä rooleja.

5.2 Kipupisteet

Tässä alaluvussa haastatteluiden tuloksia tarkastellaan tämän tutkielman kirjallisuuskatsauksen pohjalta. Aiemmin esitetyn kipupistemallin kahdeksan kipupistettä ovat huono viestintä, ketterän osaamisen ja valmennuksen puute, laadunvarmistus, uusien ja vanhojen toimintatapojen kohtaaminen, johdon tuen puute, tiimin koordinointi ja autonomia, konseptien ja ketteryyden määrittely ja muutosvastarinta.

5.2.1 Muutosvastarinta

Haastatelluissa ei noussut esiin muutosvastarintaan viittaavia havaintoja.

EC29: Haastatelluissa ei noussut esiin muutosvastarintaan viittaavia havaintoja.

5.2.2 Konseptien ja ketteryyden määrittely

Konseptien ja ketteryyden liittyvät haasteet liittyvät ketteryyden konseptien ymmärtämiseen ja ketterän ajattelutavan omaksumiseen (Dikert, Paasivaara & Lassenius, 2016; Conboy & Carroll, 2019; Putta, Paasivaara & Lassenius, 2018). Tutkittavassa hankkeessa ketteryyden ja ketterät menetelmät nousivat esille useammalla haastateltavalla.

”Pyritään tekemään ketterästi asioita. Ei niinkun näihin vanhoihin kankeisiin vesiputousmalleihin jäädä kiinni. Pyritään tuunistamaan tarpeita mahdollisimman nopeasti – – Ei jumiuuduta kaikessa tekemisessä semmoseen vanhanmalliseen, et tehdään kaksi vuotta jotain pelkkää speksausta ennen ku lähdetään tekemään jotain kehitystyötä, vaan pyritään siinä, että speksataan sen mukaan niitä asioita mitä tulee eteen. – – Koko ajan työstetään jotain uutta, mitä voidaan viedä tuotantoon ja sitä korjataan siinä juoksussa. Koko ajan valmista joka päivä ja joka sprintin aikana.” (H2)

“Tehdään pala kerrallaan. Ei yritetä suunnitella kaikkea valmiiksi. Tehdään se palikka kerrallaan ja selvitetään niihin liittyviä asioita – toisaalta se tarkoittaa myös sitä, että suunnitelmat ja tavoitteet voi muuttua hyvinkin nopealla aikavälillä.” (H5)

Ketteryyden ja ketterän julistuksen (2001) mukaiset konseptit ovat osalla organisaation jäsenen tiedossa. Jokainen tunnistaa ketteristä menetelmistä scrumin ja tunnistaa sieltä tulevia tapahtumia, rooleja ja artefakteja.

Ketterien menetelmien koetaan luovan myös haasteita. Haasteena koetaan nopeat muutokset asiakkaan tarpeissa ja näiden priorisoinnissa. Haasteena on myös hankkeen muu organisaatio ei ole ketterä, eikä ymmärrä ketterän kehitysmenetelmistä.

“Joskus on huono asia, että suunnitelmat muuttuu lennossa, kun voi olla, että on käytetty resursseja ” (H5)

“Priorisointihan on aina tietyllä tapaa haaste ketterässä kehittämisessä. – – Välillä voi muodostua haasteeksi, jos on kaksi yhtä tärkeitä asiaa ja vaan toinen keretään tekemään.” (H5)

“koko [organisaatio] kohta tajuaa, että mitä se ketterä kehitys on.” (H4)

EC30: Hankkeen työntekijöillä nähdään olevan yhteisymmärrys ketteryydestä, mutta muuta organisaatiota ei koeta ketteräksi.

Haastateltavat kertoivat käytössä olevista rooleista, artefakteista ja tapahtumisista hyvin, joihin keskitytään seuraavissa alaluvuissa. Ketteristä kehitysmenetelmistä scrum nousi useimmiten esiin. Sen sijaan skaalautuva ketterä kehitysmenetelmän SAFe:n mainitsi vain muutama haastateltava. SAFe:n käyttäminen osana organisaation menetelmiä jakoi myös haastateltavien näkemyksiä.

“Joo eli vähä niinku SAFe-kehikolla eli on myös kvartaalisuunnitelma – – Täällä mennään ihan SAFe:n ja Scrumin menetelmin ” (H1)

“Safestahan me ollaan puhuttu, että on iso hanke kaiken kaikkiaan niin meillä on safemaisia piirteitä, mutta ei tää niinku millään tavalla SAFe-mallia tässä vielä ole lähelläkään. – – Ei se [SAFe] tällä hetkellä tarkoita mitään. Enemmän ollaan mietitty, että miten tämmöstä isompaa kokonaisuutta voitaisiin hallita safemaisesti” (H2)

“Ei meillä kyllä SAFe:a käytetä, vaan ihan vanhalta kokemukselta vastasin, että tämä on mahdollista, että vähän siihen suuntaan ollaan siirtymässä, mutta ei me vielä SAFe:a oikeasti tehdä.” (H3)

“No ei se SAFe oo oikeesti, vaan yks osa-alue, mikä on vähä siihen suuntaan. – – Periaatteessa rytmitys sprinteille ja ne on samaan aikaan, on yhteiset demot, se on SAFe:ssa samaa,” (H6)

Jokaiselta haastateltavalle esitettiin kysymys käytettävissä olevista ketteristä kehitysmenetelmistä. Jokainen heistä esitti scrumin tai ainakin osan sen tapahtumista, rooleista tai artefakteista. Osa haastateltavista ei maininnut SAFea lainkaan. Haastattelujen pohjalta voidaan tehdä havainto, jonka mukaan organisaation jäsenillä on erilaisia näkemyksiä käytetyistä ketteristä menetelmistä ja skaalatuista ketteristä kehitysmenetelmistä.

“PI-planning nimellä se oikeasti menee, meillä puhutaan kvartaalisuunnitelmas-
ta.” (H1)

“release planning on se termi, mitä täällä – – käytetään ja siinä valmisteli agile coach
– – tuoteomistajien kanssa, että mikä se julkaisukokonaisuus on. Sitte otettiin mukaa
design ja arkkitehti ja scrum master ja sitte sen jälkeen suunniteltiin kaikki tiimit yh-
dessä se seuraava viis sprinttiä ja samallahan big room planning henkisesti, ku esi-
merkiks, jos on scaled agile frameworkki tuttu nii siinä SAFessa on tällanen iha for-
maali prosessi vähän raskaampi vielä diiteilimpi, mutta se on vähä ottanu siitä hei-
jastuksia” (H6)

Tutkittavassa hankkeessa työroolien näkemykset erosivat osalla haastatel-
tavista. Tällaisia rooleja olivat tekninen tuoteomistaja sekä projektipäällikkö.

“Omassa projektissa ei oo projektipäällikköä, mutta on tekninen tuoteomistaja. Niin
mikä on varsinaisesti heidän työnkuvansa ero niin siihen en oikein osaa vastata.” (H4)

EC31: Jotkin hankkeessa käytetyt ketterien kehitysmenetelmien konseptit ja näke-
mykset vaikuttavat eroavan työntekijöiden kesken.

PEC5: Hankkeessa työskentelevillä henkilöillä on samankaltainen
ymmärrys ketteristä toimintatavoista. Työntekijöillä on erilaisia nä-
kemyksiä skaalautuvan ketterän kehitysmenetelmän käyttöasteesta.

5.2.3 Tiimin koordinointi ja autonomia

Skaalatessa tiimin koordinointi ja autonomian säilyttäminen ovat skaalautumi-
sen yksi kipupiste (Conboy & Carroll, 2019; Dikert, Paasivaara & Lassenius,
2016; Putta, Paasivaara & Lassenius, 2018). Haastatteluissa ketteristä kehitys-
menetelmistä vain scrum ja SAFe mainittiin nimeltä. Tämä antaa olettaa, että
tiimit eivät itse päättä, miten koordinoivat työskentelynsä tai millaisia menetel-
miä käyttävät.

“No se [scrum] on tää viitekehys kautta malli, millä – – hankkeessa kehitystä tehään
ja se määrittelee aika paljon arkea. Tietyllä tapaa kehys tekemiselle.” (H5)

“scrum työmenetelmällä teen tiimien kaa töitä, niin se osa pyörii sen teorian kaut-
ta.” (H4)

“Scrumistahan puhutaan. Se on se arki, mikä tiimeille näkyy. Nyt julkasusuunnittelu
on tuonu tiimeille näkyville sen, et mitä muut tekee ja luonu tän viitekehysten.” (H6)

“Täällä mennään ihan SAFen ja Scrumin menetelmin.” (H1)

Yksi haastateltava nosti erikseen haasteena sen esille, että tiimit ovat pako-
tettuja toimimaan scrum-menetelmän mukaisesti. Toisaalta ymmärrystä antaa
usean tiimin työn yhteensovittamisen haaste, jota pyritään ratkaisemaan yhte-
näisellä rytmillä.

“tiimit on vähä niiku pakotettu ei vähänkää, vaa pakotettu käyttää scrumia. Jotkut tiimit on vähä meille sopis enemmän kanbantyylinen, mutta tavallaa on haluttu sitte, että kaikilla tiimeillä on yhdenmukainen sprintti tai tekemisen tapa ja myös se kaudenssi, että se tapahtuu samaan aikaan.” (H6)

EC32: Haastateltavat kertovat tiimien työskentelevän scrum-menetelmällä tiimien välisen yhteisen rytmin ylläpitämiseksi.

Toisaalta toisessa haastattelussa tuli esille, että tiimejä kuunnellaan ja hankkeen kehitysmenetelmää kehitetään jatkuvasti. Toisaalta taas tästä saa käsityksen, että pyrkimyksenä on kuitenkin kehittää kaikille tiimeille yhdenmukaista mallia.

“No siis täällä on tietty malli, mitä ollaan tätä hommaan kehitetty ja sitä tarkkaillaan koko ajan. Kerätään palautetta tiimeiltä ja [organisaation nimi] omilta tekijöiltä ja pyritään kehittämään niitä näiden tunnistettujen kipupisteiden kautta. Ei oo mitään semmosta puhdasta mallia mikään mukaisesti. On otettu agile scrumista parhaimpia puolia ja sovitettu ne meidän ympäristöön ja tekemiseen.” (H2)

EC33: Tiimit soveltavat scrumia tarpeen mukaan.

PEC6: Tiimit työskentelevät autonomisesti scrum-mallin mukaisesti, jota kehitetään ja sovelletaan tarpeiden mukaan.

5.2.4 Johtaminen: tuen puute

Haastatteluissa ei noussut esiin johtamiseen viittaavia havaintoja kipupisteinä.

EC34: Haastatteluissa ei noussut esiin johtamiseen viittaavia havaintoja kipupisteinä.

PEC7: Hankkeessa ei esiintynyt muutosvastarintaan tai johtamisen tuen puutteeseen viittaavia kipupisteitä.

5.2.5 Uusien ja vanhojen toimintatapojen kohtaaminen

Tieteellisessä kirjallisuudessa esiintyneet johdon tuen puutetta tai muutosvastarintaa ei esiintynyt. Muutosvastarinta kipupisteinä esiintyy luonteensa vuoksi organisaation muutoksessa esimerkiksi uusien työtapojen käyttöönotossa (Dikert, Paasivaara & Lassenius, 2016). Haastattelujen perusteella muutosvastarintaa ei havaittu. Muutosvastarinta on toisaalta yksi eniten esiintyneistä kipupisteistä tieteellisessä kirjallisuudessa. Myös uusien ja vanhojen toimintatapojen kohtaamiseen liittyviä haasteita ei havaittu suoraan, osaksi samasta syystä kuin muutosvastarinta. Uusien ja vanhojen toimintatapojen haasteeksi tosin voidaan

nimetä organisaatiossa mukana oleva projektipäällikön rooli, jota skaalautuviin ketteriin kehitysmenetelmiin tavallisesti ei sisälly.

“Omassa projektissa ei oo projektipäällikköä, mutta on tekninen tuoteomistaja. Niin mikä on varsinaisesti heidän työnkuvansa ero niin siihen en oikein osaa vastata.” (H4)

EC35: Hankkeeseen sisältyy ketterien kehitysmenetelmien roolien lisäksi perinteisiä projektinhallintarooleja, joita ei esiinny ketterissä kehitysmenetelmissä.

PEC8: Hankkeessa esiintyy organisaatorakenteen vuoksi perinteisiä rooleja, joita ei esiinny useimmissa skaalautuvissa ketterissä kehitysmenetelmissä.

5.2.6 Laadunvarmistus

Haastatteluissa nousi esiin erilaisia laadunvarmistukseen liittyviä haasteita. Laadunvarmistukseen liittyvät haasteet liittyvät ei-toiminnallisten vaatimusten testaukseen tai vaatimusten monitulkintaisuuteen (Dikert, Paasivaara & Lassenius, 2016), tiimien kehitysprosessien ongelmiin (Kalenda, Hyna & Rossi, 2018) tai testauksen ongelmiin (Dikert, Paasivaara & Lassenius, 2016; Putta, Paasivaara & Lassenius, 2018). Haastatteluissa nousi esiin dokumentaation ja määrittelyiden ongelmat.

“Nythän on myös ongelmana, että siellä [dokumentaatioissa] on hirveästi vanhaa, että kuka kerkee sitä siivoamaan. Pitäisi olla lyhyesti kirjoitettuna, ei kirjoitella diiba-daabaa, että tää on se MVP. – – mitä se [dokumentaatio] näyttää, niin siihen meillä ei oo mitenkään järkevää järjestelmää järjestelyyn siellä.” (H4)

“Aluksi oli vaikeuksia hahmottaa kokonaisuuksia, että Powerpoint minne itse kirjasin asioita, tiimi tykkäsi siitä esitystavasta, mut muut tiimit eivät nähneet tätä tai pysyneet kärryillä mitä ollaan tekemässä. Piti keksiä uusi työkalu, ja nyt ollaan otettu Miro käyttöön” (H4)

“Yks semmonen isompi kokonaisuusalue, mikä on haastavaa, että saadaan liiketoiminnalta ne tarpeet, vaatimukset ja määritykset kasaan ajoissa niin, että päästään tekemään sitä varsinaista toteutustyötä. Just tämmöstä vaatimusmäärittelyvoimaa tarvittaisiin, et saataisiin työstettyä speksausta seuraavalle tasolle – – sit on kipupisteenä ehdottomasti loppukäyttäjät yksiköt, jotka tulee palvelua käyttämään, niin he eivät – – välttämättä voi osallistua näihin määritysasioihin.” (H2)

“Ei liity pelkästään tähän hankkeeseen, vaan ihan kaikkeen tämän tyylliseen tekemiseen. Osoittautunut joka paikassa, riippumatta siitä missä on töissä, eli haluttavat featuret on väistämättä liian suuria. Määrittely on mietitty yleensä vaan substanssin kannalta ja harvoin teknisesti.” (H3)

Haastatteluissa nousseet laadunvarmistukseen liittyvät haasteet käsittelevät pääasiassa dokumentoinnin ja määrittelyyn liittyviä ongelmakohtia. Dokumentointi ei ole ajan tasalla tai määrittelyssä ongelmat koskevat niiden heikkoa laatua tai tarkkuuden puutetta. Ongelma tunnustetaan organisaatiossa, ja ratkaisu-

na tähän on esitetty teknisen tuoteomistajan ottamista mukaan määrittelytyöhön.

EC36: Haastateltavat kokevat, että vaatimusmäärittelyyn ei ole tarpeeksi tekijöitä ja sitä ei tehdä tarpeeksi teknisesti.

EC37: Haastateltavien kokemusten perusteella dokumentaatio ei pysy ajan tasalla, se on useassa paikassa ja tieto vanhentuu.

Laadunvarmistukseen yleisesti liittyvä menetelmä on ohjelmiston testaaminen. Testaamisella varmistetaan, että ohjelmisto toimii kuten pitääkin. Testauksen ongelmia ei havaittu haastatteluissa, mutta DevOps ja CI/CD-putkea, johon testaus myös sisältyy, kehitettiin.

“Kyllä tää on yks parhaimmista CI/CD-putkesta mitä oon nähny tän 20 vuoden aikana. – Se on luotettava, varma ja tekee sen mitä sen pitää tehdä. Kehittäjän ei itse tarvitse siitä hirveästi huolehtia” (H2)

Testaukseen liittyviä ongelmia ei tullut esiin. Osa haastateltavista kertoi erillisestä testaustiimistä, osa taas ymmärsi ja sisällytti testaajan osaksi jokaista tiimiä. Laadunvarmistukseen liittyvät ongelmat ovat osaltaan myös viestintään liittyviä ongelmia. Esimerkiksi dokumentointiin liittyvät haasteet voidaan nähdä sekä laadunvarmistuksen että viestinnän haasteina.

Hankkeessa on käytössä erityinen laadunvarmistusviikko, jonka tarkoituksena on korjata teknistä velkaa. Laadunvarmistusviikko pidetään jokaisen julkaisun viimeisellä viikolla. Laadunvarmistusviikosta kerrotaan seuraavaa.

“kvartaalin viimeinen sprintti on viikon mittainen ja se on refaktorointisprintti nimellä mennyt, mut se on enemmän vaan laadunvarmistussprintti oikeesti niin siinä oikeesti viimeistellään mitä on tehty” (H1)

EC38: Haastatteluissa laadunvarmistusta kerrotaan toteutettavan joka julkaisuun kuuluvalla laadunvarmistusviikolla.

PEC9: Hankkeessa on laadunvarmistukseen liittyviä haasteita. Haasteena ovat vaatimusmäärittelyn ongelmat, jotka johtuvat teknisen osaamisen tai resurssien puutteesta.

5.2.7 Ketterä osaaminen ja valmennus

Ketterän kehittämisen kipupisteenä voi olla ketterän koulutuksen puuttuminen (Dikert, Paasivaara & Lassenius, 2016; Kalenda ym., 2018; Paasivaara, 2017; Shameem ym., 2017). Organisaatiotasolta puuttuu ketterä osaaminen, mikä on ollut hankkeen lähtökohta myös joidenkin työntekijöiden osalta. Koulutusta ketterästä kehittämisestä on saatu, mutta sitä ei ole koettu yhtä hyödylliseksi kuin tiimin jakama tieto ketteryydestä.

“Juu ei ne [organisaatio] oo vielä tajunnu sitä, kyllä ne kohta tajuaa koko [organisaatio], että mikä se ketterä kehitys on.” (H4)

“Ei lainkaan [aiempaa kokemusta ketterästä kehittämisestä]. – – koulutuksia olen saanut, mut se ei auttanut. Ihmisten kautta opittua.” (H4)

Ketterien kehitysmenetelmien tuntemus on kuitenkin hankkeessa vahvaa. Mikä nousi esille, on se, että tuoteomistajilta puuttuu tekninen osaaminen, joka vaikuttaa tekniseen määrittelyosaamiseen.

“No agilehan on sitä, että pyritään tekemään ketterästi asioita. Ei niinkun näihin vanhoihin kankeisiin vesiputousmallin menetelmiin jäädä kiinni, vaan pyritään tunnistamaan tarpeita mahdollisimman nopeasti, mitä siinä on ja tekemään siinä mielessä ketterämmin asioita. Se on se perus ydin.” (H2)

“...uupuu sitä sellasta teknistä määrittelyosaamista. Sitä on paikattu nyt, et on otettu näitä konsultteja tekninen tuoteomistaja roolille tekee teknisen näkökulman siihe tuotemistajan rooliin. Tuoteomistajalla on tosi hyvä osaaminen siihe domainiin, mut sit, ku se tekninen osaaminen ei se ei oo tarpeeks korkealla.” (H6)

“[tekninen tuoteomistaja] auttaa niin ku siinä tavallaan teknisessä speksauksessa siten vaatimusten kanssa. Ja tukee sitten tietysti tiimejä vielä vahvemmin siinä suunnittelussa.” (H7)

EC39: Haastateltavat kokevat, että organisaatiotasolla ketteryyttä ei tunneta. Hanketasolla koetaan, että ketteryys on tunnettua.

EC40: Hankkeen tasolla työntekijöiden ketterään osaamiseen ja valmennukseen voidaan nähdä panostettavan hankkeessa.

EC41: Tuoteomistajien tekninen osaaminen koetaan riittämättömäksi tekniseen määrittelyyn.

PEC10: Koko organisaatiotasolla ketteryyttä ei tunneta. Hanketasolla on kuitenkin panostettu työntekijöiden ketterän osaamiseen ja valmennukseen, mutta tuoteomistajien osaaminen koetaan hankkeessa vielä riittämättömäksi tekniseen vaatimusmäärittelyyn.

5.2.8 Viestintä: huono viestintä

Haastatteluissa viestintä nousi esille haasteena läpinäkyvyyden puutteena päätöksentekotasolla, mutta sen onnistumista tiimitasolla korostettiin. Huono viestintä on yksi kirjallisuuskatsauksessa esitetyistä kipupisteistä (Dikert, Paasivaara & Lassenius, 2016; Paasivaara, 2017; Shameem ym., 2017).

Viestinnän merkitys korostuu suuressa kehitystyössä, jossa on paljon työntekijöitä ja kun tarkastellaan, mistä ongelmat ovat johtuneet.

“Aika laidasta laitaan, mut semmonen tietty yleisteema on niin ku isoo palettia pyörittää nii se on kommunikaatio, mihin kiteytyy aika paljon asioita. Sit kun mennään juurisyihin niin yleensä se ongelma on kommunikaation puute tai ongelma.” (H2)

EC42: Ongelmien juurisyiden koetaan liittyvän viestintään tai sen puutteeseen.

Haastatteluissa pohdittiin myös koronapandemian myötä lisääntyneen etätyön merkitystä hankkeen viestintässä heikentävänä seikkana. Etätyön aiheuttama haaste viestinnässä on katsottu johtaneen tehokkuuden laskuun ja määrittelyn vaikeutumiseen. Haastatteluiden perusteella etätyö on vaikeuttanut viestintää hiljaisen tiedon siirtämisen osalta ja niin sanottujen käytäväkeskustelujen poistuessa arjesta.

“Eliikkäs se on tämä hiljaisen tiedon välittyminen, että kaikki tekis samansuuntaista asioita ja eri suuntiin niin se on vaikeutunut. Sitten meillä näitä mittareita, että pieniä tehostomuutta on havaittavissa. Tikettejä ja stoorien määrittelyt ei oo täysin samaa, mitä läsnä tekemisessä.” (H1)

“Ei se [viestintä] oikeastaan oo [tehokasta etänä]. Johtuu just siitä, että jää tämmöne kahvipöytäkeskustelu kautta kaiken muun ohella keskustelu, vaikka on hyvät viestintäkanavat tämmöset, mutta ei se silti oo niin aktiivista ehkä ku face to face.” (H5)

“semmonen spontaani kommunikointi jää käytännössä kokonaan pois.” (H7)

“varsinkin isoa kokonaisuutta hoidetaan eteenpäin niin siinä on haasteita, että saadaan välitettyä kaikille tarvittaville ihmisille se tarvittava tieto. Ei kulje niin hyvin, kuin oltaisiin saman pöydän ääressä asioita. [...] Vaikka kuinka on hyvät videoyhteydet ja muuten työvälineet, niin se ei oo sama asia kun ollaan ihan läsnä.” (H2)

Viestinnän kannalta etätyö nähdään heikentävänä seikkana, mutta toisaalta etätyön katsotaan estävän jatkuvat työn keskeytykset. Lisäksi etätyön aiheuttaman etäviestinnän katsotaan kehittyneen ajan kuluessa ja yhteydenoton kynnystä on pyritty madaltamaan.

“Siinä on tietyt hyvät puolet etätyössä. Se, että pystyy keskittymään, jos on selkeitä alueita, mitä tehdään niin pystyy tekemään selkeemmin siinä sitten. Ei oo niitä häiriötekijöitä, joku koputtamassa olkapäälle, mutta tehdään yhteistä isoa asiaa niin siihen tulee välttämättä sitä sitten, että pitää olla muihin yhteyksissä.” (H2)

“Mut sitä [etänä viestintää] ollaan nyt pyritty ja harjoteltu nyt vuosi, et kynnys on niin matala, kun vaan voi olla. Kyl se varmaan on eteenpäin menny.” (H7)

EC43: Etätyön koetaan vaikeuttavan viestintää niin sanottujen käytäväkeskustelujen poistuessa arjesta.

EC44: Etätyön hyvänä ominaisuutena koetaan keskeytyksetön työajan mahdollisuus.

Vaikka kommunikaation katsotaan olevan haaste hankkeessa, tiimin sisäistä viestintää kehitetään. Tiimien välisen ja hankkeen sisäisen läpinäkyvyyden katsotaan olevan hyvää. Tiimien sisäisen kommunikaation nähdään toimivan, sillä apua saa aina, kun sitä tarvitaan. Myös tiimien välinen ja tiimien ulospäin antama läpinäkyvyys on hyvää. Läpinäkyvyys liitetään hankkeessa tiedon ja dokumentaation tavoitettavuuteen, mitkä nähdään hankkeessa hyvin tavoitettavina ja läpinäkyvinä.

“[kommunikaatio tiimissä] Toimii oikea hyvi. Keskustelua on jatkuvasti ja apua saa aina, ku tarvii ja kaikesta pystytään käytännössä keskustelemaan jatkuvasti. Et ei oo mitään mun näkemyksen mukaan ongelmaa.” (H3)

“On niistä [dailyista ja scrum of scrumsista], jos meillä ei olis näitä, kun ollaan etänä tapaamisia niin tosi vaikee tietää kehenkin pitäisi olla yhteyttä. Muuten ei yhtään tietäis, mitä ne muut tekee. Tässä ollaan nyt saatu keyasioita, mitä ei muuten saatais. Ei ne oo joka päivä yhtä arvokkaita, mutta ehdottomasti niin arvokkaita että niitä pitää pitää joka päivä.” (H4)

“mut mitä tiimit tekee nii sehän alkaa se läpinäkyvyys olee iha ok ja mun mielestä sitä jaetaan sitä tietoa noissa scrum of scrumeissa ja muissa ja tuolla slackissa sun muuta confluencessa se on iha ok tekemisen läpinäkyvyys tiimeiltä.” (H6)

“[kokee hankkeen] hyvin läpinäkyväksi. Ei ole mitään tähän hankkeen tekemiseen liittyvää dokumentaatioo tai materiaalia mihinkään piilotettu, vaan kaikki on julkisessa hankkeen sisäisessä nähtävillä kaikilla.” (H2)

Läpinäkyvyyttä on myös parannettu hankkeen sisäisesti.

“on periaateessa se tiedon jakaminen ihan ok ja se on yks, mikä on tullu tuon julkasusuunnittelun parantanu sitä, ku aikasemmin ne tavallaa suunnitelmat oli tuoteomistajien jossai exceleissä nii se toi ne sit sinne miroo näkyviin hei täällä nää näkee muutki, ku se tuoteomistaja ei se tavallaa tahallaa yrittäny piilottaa, vaa ne vaa on olu hänellä ei vaa oo ollu missää.” (H6)

EC45: Etenkin tiimien sisäinen viestintä nähdään toimivana ja läpinäkyvänä. Myös hankkeen sisäinen viestintä nähdään hyvänä ja läpinäkyvänä ja sitä pyritään kehittämään.

PEC11: Etätyö on vähentänyt kasvokkain tapahtuvaa viestintää, joka koetaan haasteelliseksi kommunikoinnin kannalta. Muuten tiimien välinen viestintä koetaan toimivana ja läpinäkyvänä.

Tiimin keskeistä viestintää ja hankkeen sisäistä läpinäkyvyyttä kehitettäessä läpinäkyvyys päätöksentekotasolla on haastatteluiden perusteella huono. Päätöksenteon läpinäkyvyyden lisäksi hankkeessa nähdään viestinnän haasteena vaatimusmäärittelyyn liittyvät ongelmat.

“No lähtökohat ehkä ei oo ollu kauhee läpinäkyvät ja edellee tietenki jotkut päätöksentekoprosessit on piilossa olevia – – mut se ehkä läpinäkyvyys prosesseihin, missä niitä määritellään ja kuka niistä päättää ja ehkä missä päätetään, mitä featureita tulee ja missä aikataulussa ja miksi. Se ei oo kauhee läpinäkyvää.” (H6)

“Meidän näkökulmasta se haaste on siinä, että millä tasolla niitä prosesseja määritellään. Me tarvitaan aika tarkalla sitä prosessimäärittystä, että mitä tapahtuu missäkin kohtaa, ja sitten kun asiakkaan kanssa käydään niin se on hyvin korkeamman tason määrittelyä. Siinä kohtaa on se meidän haaste. – – Niin siinä on ehkä meidän kommunikointi tavallaan, millä tasolla niitä prosesseja määritellään.” (H7)

“Yks ongelma tällaisissa isoissa ongelmissa aina, että miten varmistetaan ja valvotaan, että saadaan tarvittava tieto tarvittaville ihmisille. Yks semmonen isompi kokonaisuusalue, mikä on haastavaa, että saadaan liiketoiminnalta ne tarpeet, vaatimukset ja määrittymiset kasaan ajoissa niin, että päästään tekemään sitä itse varsinaista toteutustyötä. Just tämmöstä vaatimusmäärittämisvoimaa tarvittaisiin lisää jonkun verran tähän koneistoon, että saataisiin työstettyä speksausta tarvittavalle tasolle.” (H2)

Toisaalta viestinnän lisäksi haasteena vaatimusmäärittelyssä nähdään loppukäyttäjien ajanpuute.

“Sit on totta kai kipupisteenä ehdottomasti loppukäyttäjäyksiköt, mitä tulee palvelua käyttämään niin heillä on kova kiire oman operatiivisen työn kanssa niin he eivät välttämättä voi osallistua niin hyvin näihin määrittelyasioihin, mitä pitäisi olla mukana siinä sitten. Saattaa olla jarruna tekemiselle.” (H2)

EC46: Viestinnän kipupisteet nähdään piilevän päätöksenteon huonossa läpinäkyvyydessä ja vaatimusmäärittelyn tasossa.

PEC12: Hanketasolla konseptien ja ketteryden määrittely on pitkälti yhdenmukaista. Organisaation ja muiden sidosryhmien osalta konseptien ja ketteryden määrittelyä ei tunneta samalla tasolla, mikä aiheuttaa läpinäkymättömyyttä ja ketteryden puuttumista päätöksentekotasolla.

6 KESKUSTELU

Tässä luvussa tarkastellaan tämän pro gradu -tutkielman teoreettisia ja käytännön johtopäätöksiä. Tutkimuksen tuloksista tehdään teoreettiset johtopäätökset, ja niiden peilautumista arvioidaan toteuttamaamme kirjallisuuskatsaukseen. Toisena käsittelemme käytännön johtopäätökset, jossa esitämme käytännön ehdotuksia empiiristen havaintojen ratkaisemiseksi.

6.1 Teoreettiset implikaatiot

Tutkielman primääriset empiiriset havainnot esitetään taulukossa 5. Tutkielman kannalta näitä voidaan pitää tärkeimpinä havaintoina. Primäärihavainnot koostuvat useasta empiirisestä havainnosta. Seuraavaksi avaamme löytämiämme empiirisiä havaintoja.

Ensimmäisenä primäärihavaintoina (PEC1) havaitsimme haastateltavien kertovan yksimielisesti käytetyksi kehitysmenetelmäksi scrumin, johon on otettu mukaan muiden kehitysmenetelmien, kuten SAFen ja kanbanin piirteitä. Primäärihavaintona kaksi (PEC2) havaitsimme, että tällaisia piirteitä ovat esimerkiksi scrum-of-scrums, skaalatun sprintin tarkastelu sekä skaalattu sprinttisuunnittelun hyödyntäminen ja käyttäminen. Kolmantena primäärihavaintona (PEC3) havaitsimme, että hankkeessa hyödynnetään tapahtumia, joita ei esiinny sellaisenaan muissa ketterissä kehitysmenetelmissä. Nämä havaitsemamme tapahtumat ovat refaktorointiviikko, PO-aika ja arkkitehtuurikokous. Ensimmäiset kolme primäärihavaintoa ovat linjassa tieteellisen kirjallisuuden kanssa, sillä skaalautuvia ketteriä kehitysmenetelmiä voidaan omaksua sellaisenaan tai niitä voidaan räätälöidä organisaation kulttuurin, ajattelu- tai työskentelytavan mukaisesti (Ebert & Paasivaara, 2017).

Neljäntenä päähavaintona (PEC4) havaitsimme, että organisaatiossa hyödynnetään rooleja, joita ei esiinny muissa kirjallisuuskatsauksessa esitetyissä skaalautuvissa ketterissä kehitysmenetelmissä. Tällaisia rooleja olivat mm. hankepäällikkö ja projektipäällikkö. Kehitysmenetelmän rooleja oli kustomoitu or-

ganisaatioon sopivaksi lisäämällä erilaisia rooleja. Empirian perusteella hankkeessa esiintyy uusia sekä vanhoja toimintatapoja, joka on kahdeksas primäärihavainto (PEC8). On mahdollista, että organisaatiorakenteen vuoksi ketterät kehitysmenetelmät on otettu käyttöön perinteisen organisaatiorakenteen päälle, ilman suurempia muutoksia organisaation rooleihin.

Taulukko 5 Havaintojen suhde kirjallisuuteen

Tunniste	Empiirinen havainto	Suhde kirjallisuuteen
PEC1	Hankkeen haastateltavat tunnistavat käytetyksi kehitysmenetelmäksi scrumin. He kertovat käytössä olevan myös SAFE:n ja kanbanin piirteitä.	Vastaa, skaalautuvia ketteriä menetelmiä voidaan omaksua sellaisinaan tai ne voidaan räätälöidä organisaation kulttuurin, ajattelu- tai työskentelytytävän mukaisesti (Ebert & Paasivaara, 2017).
PEC2	Käytettävää ketterää kehitysmenetelmää on skaalattu hankkeen tarpeiden mukaisesti. Havaitsemme hankkeessa olevan skaalautuvien ketterien kehitysmenetelmien, kuten SAFE ja LeSS, piirteitä.	Vastaa, skaalautuvia ketteriä menetelmiä voidaan omaksua sellaisinaan tai ne voidaan räätälöidä organisaation kulttuurin, ajattelu- tai työskentelytytävän mukaisesti (Ebert & Paasivaara, 2017).
PEC3	Käytettävää ketterää kehitysmenetelmää on muokattu hankkeen tarpeiden mukaisesti.	Vastaa, skaalautuvia ketteriä menetelmiä voidaan omaksua sellaisinaan tai ne voidaan räätälöidä organisaation kulttuurin, ajattelu- tai työskentelytytävän mukaisesti (Ebert & Paasivaara, 2017).
PEC4	Hankkeessa havaitaan hyödynnettävän scrum-roolien ohella skaalautuvissa ketterissä kehitysmenetelmissä esiintyviä rooleja.	Vastaa, skaalautuvia ketteriä menetelmiä voidaan omaksua sellaisinaan tai ne voidaan räätälöidä organisaation kulttuurin, ajattelu- tai työskentelytytävän mukaisesti (Ebert & Paasivaara, 2017).
PEC5	Hankkeessa haastateltavilla henkilöillä on samankaltainen ymmärrys ketteristä toimintatavoista. Havaitsimme haastateltavilla olevan erilaisia näkemyksiä skaalautuvan ketterän kehitysmenetelmän käyttöasteesta.	Vastaa, konseptien ja ketteryyden määrittely voidaan nähdä kipupisteenä hankkeessa vaihtelevien näkemysten perusteella. (Dikert ym., 2016; Conboy & Carroll, 2019; Putta ym., 2018).
PEC6	Tiimit työskentelevät autonomisesti scrum-mallin mukaisesti, jota kehitetään ja sovelletaan tarpeiden mukaan.	Vastaa, tiimit työskentelevät yhteisen kehitysmenetelmän mukaisesti, sitä soveltaen. (Ebert & Paasivaara, 2017.) Kipupistettä ei esiinny.
PEC7	Hankkeessa ei esiintynyt muutosvastarintaan tai johtamisen tuen puutteeseen viittaavia kipupisteitä.	Ei vastaa, muutosvastarinta ja johdon tuen puute ovat yleisimpiä kipupisteitä. (Dikert, ym. 2016; Shameem ym., 2017; Paasivaara, 2017; Kalenda ym., 2018; Conboy & Carroll, 2019; Putta ym., 2018).
PEC8	Hankkeessa esiintyy organisaatiora-	Vastaa, uusien ja vanhojen toimintata-

- kenteen vuoksi perinteisiä rooleja, joita ei esiinny useimmissa skaalautuvissa ketterissä kehitysmenetelmissä.
- PEC9 Hankkeessa on laadunvarmistukseen liittyviä haasteita. Haasteena ovat vaatimusmäärittelyn ongelmat, jotka johtuvat teknisen osaamisen tai resurssien puutteesta. Vastaa, laadunvarmistukseen liittyvät ongelmat voidaan nähdä kipupisteinä (Dikert ym., 2016; Kalenda ym., 2018; Putta ym., 2018).
- PEC10 Koko organisaatiotasolla ketteryttä ei tunneta. Hanketasolla on kuitenkin panostettu työntekijöiden ketterän osaamiseen ja valmennukseen, mutta tuoteomistajien osaaminen koetaan hankkeessa vielä riittämättömäksi tekniseen vaatimusmäärittelyyn. Vastaa, ketterä osaaminen ja valmennuksen puute on huomioitu hankkeessa kipupisteinä (Dikert ym., 2016; Paasivaara, 2017; Kalenda ym., 2018; Shameem ym., 2017).
- PEC11 Etätö on vähentänyt kasvokkain tapahtuvaa viestintää, joka koetaan haasteelliseksi kommunikoinnin kannalta. Muuten tiimien välinen viestintä koetaan toimivana ja läpinäkyvänä Vastaa, organisaation saavuttaessa läpinäkyvän ja avoimen ympäristön, huonon viestinnän kipupiste ei pääse toteutumaan organisaatiossa (Kalenda ym., 2018). Etätöön aiheuttama viestinnän puute voidaan nähdä kipupisteinä.
- PEC12 Hanketasolla konseptien ja ketteryden määrittely on pitkälti yhdenmukaista. Organisaation ja muiden sidosryhmien osalta konseptien ja ketteryden määrittelyä ei tunneta samalla tasolla, mikä aiheuttaa läpinäkymättömyyttä ja ketteryden puuttumista päätöksentekotasolla. Vastaa, konseptien ja ketteryden määrittely on tunnistettu kipupiste (Conboy & Carroll, 2019; Dikert ym., 2016; Putta ym., 2018).

Viidentenä primäärihavaintona (PEC5) havaitsimme, että haastateltavilla on samankaltaiset näkemykset ketterydestä. Sen sijaan skaalautuvan ketterän kehitysmenetelmän käytössä olevasta menetelmästä oli eriäviä näkemyksiä. Osa haastateltavista kertoi, että organisaatiossa hyödynnetään SAFea, kun osa haastateltavista näki, että toimitaan SAFen kaltaisesti, mutta ei kuitenkaan hyödynnetä sitä kokonaisuudessaan. Osa haastateltavista ei maininnut SAFea lainkaan. Konseptien ja ketteryden määrittelyä voidaan pitää kipupisteinä hankkeessa (Conboy & Carroll, 2019; Dikert ym., 2016; Putta ym., 2018). Kipupistemallimme mukaan ratkaisuehdotuksia tälle kipupisteelle on liiketoimintalähtöinen ajattelu menetelmän lähtökohtana, yhteisen kielen sopusointu sidosryhmille varhaisessa vaiheessa (Conboy & Carroll, 2019) ja yhtenäisistä arvoista ja käytänteistä sopiminen (Kalenda ym., 2018). Kuudentena havaintona (PEC6) havaitsimme tiimien hyödyntävän scrum-kehitysmenetelmää hallitukseen kadenssia. Tiimit saavat itse päättää, kuinka pitävät dailyt, sprintin suunnittelut sekä retrospektiivit. Tiimit toimivat autonomisesti, mutta heidän tulee käyttää scrumia.

Muutosvastarintaan ja johtamiseen liittyviä empiirisiä havaintoja ei esiinny hankkeessa (PEC7), mikä ei ole linjassa tieteellisen kirjallisuuden kanssa, sillä muutosvastarinta ja johdontuen puute ovat yksi eniten mainittuja haasteita skaalattujen ketterien kehitysmenetelmien käyttöönotossa ja käytössä (Conboy & Carroll, 2019; Dikert ym., 2016; Kalenda ym., 2018; Putta ym., 2018; Shameem ym., 2017; Paasivaara, 2017). Haastateltavat eivät ilmaisseet tyytymättömyyttä käytettävästä ketterästä kehitysmenetelmästä, vaan siihen oltiin tyytyväisiä tai suhtauduttiin neutraalisti. Yksi haastateltava mainitsi, että osa kehitystiimeistä voisi hyötyä toisenlaisen kehitysmenetelmän käytöstä enemmän kuin scrumista. Muutosvastarinta yleensä ilmenee organisaation toimintatapojen tai käytänteiden muutoksessa. Johtamiseen viittaavia ongelmia ei ilmennyt haastatteluisissa. Muutosvastarintaa tai johdon tuen puutetta ei havaintojen perusteella pidetä kipupisteinä hankkeessa.

Yhdeksäntenä primäärihavaintona (PEC9) havaitsimme, että tutkittavassa organisaatiossa määrittelyn ongelmat johtuivat teknisen osaamisen tai resursien puutteesta. Vaatimusmäärittelyyn ei ole tarpeeksi resursseja, dokumentaatio ei pysy ajan tasalla ja tekniseen määrittelyyn ei ole tarvittavaa osaamista. Empiirinen havainto on linjassa tieteellisen kirjallisuuden kanssa, sillä se liittyy laadunvarmistuksen ongelmiin. (Dikert, Paasivaara & Lassenius, 2016; Kalenda, Hyna & Rossi, 2018; Putta, Paasivaara & Lassenius, 2018) Laadunvarmistuksen ongelmiin ratkaisuna ehdotetaan kipupistemallissamme säännöllisiä auditointeja työskentelytavoista (Conboy & Carroll, 2019), teknisen velan ja toleranssin hallintaa (Kalenda ym., 2018) ja tiimien rajat ylittävää laadunvarmistustiimiä testaukseen (Dikert ym., 2016). Laadunvarmistus voidaan nähdä kipupisteinä hankkeessa.

Kymmenes primäärihavainto (PEC10) koskee ketterää osaamista ja valmennusta. Hanketasolla ketteryyden ja sen valmennukseen on huomioitu ottamalla ketterä valmentaja osaksi hanketta, ja haastateltavat näkevät hankkeen ketteränä. Haastateltavat kokevat, että ketteryyttä ei tunneta ja ymmärretä organisaation tasolla, mikä voi heikentää ketterää työskentelyä. Osaksi tuoteomistajien tekninen osaaminen koetaan riittämättömäksi, joka voidaan nähdä myös heikkona ketteränä osaamisena. Toisaalta hanketasolla kipupiste on huomioitu lisäämällä hankkeeseen rooli tekninen tuoteomistaja. Ketterää osaamista ja valmennusta ei nähdä kipupisteinä hanketasolla.

Yhdenentoista havaintona (PEC11) useat haastateltavat kertoivat etätöön vähentäneen kasvokkain tapahtuvaa viestintää, mutta he kertoivat tiimien välisen sekä hankkeen sisäisen viestinnän toimivana. Etätö on vähentänyt kahvipöytäkeskusteluja, jonka vuoksi osa viestinnästä ei tavoita kaikkia työntekijöitä. Etätöön aiheuttama viestinnän puute voidaan nähdä kipupisteinä. Toisaalta viestintä oli haastateltavien mukaan hankkeen sisäisesti läpinäkyvää ja toimivaa. Silloin kun organisaatiossa saavutetaan läpinäkyvä ympäristö, niin huono viestinnän kipupiste ei pääse toteutumaan organisaatiossa (Kalenda ym., 2018). Huono viestinnän kipupiste ratkaisuehdotuksia ovat intensiivinen viestintä ja läpinäkyvyys (Dikert ym., 2016) ja muodollisemman tiimin käyttö isoissa tiimeissä (Hummel, Rosenkranz & Holten, 2013). Myös hyvin toteutettu sprintin

suunnittelu ja inkrementin suunnittelu (Paasivaara, 2017) auttavat kipupisteen ratkaisemisessa. Tiimien välisten yhteistyön tukeminen skaalautuilla tapahtumilla, esimerkiksi scrum of scrums tai käytäntöjen yhteisöt, voivat auttaa viestinnän kipupisteen ratkaisussa (Kalenda ym., 2018).

Viimeinen päähavainto (PEC12) on, että hanketasolla konseptien ja ketteryuden määrittely on yhdenmukaista. Sen sijaan organisaation ja muiden sidosryhmien osalta ketteryuden konseptit ja määrittelyjä ei tunneta samalla tavalla, mikä aiheuttaa läpinäkymättömyyttä ja ketteryuden puuttumista päätöksentekotasolla. Tämä voi johtaa ketteryuden tehottomuuteen hanketasolla. Ratkaisuna ehdotetaan liiketoimintalähtöistä ajattelua menetelmälle, yhteisen kielen sopusointi sidosryhmille varhaisessa vaiheessa (Conboy & Carroll, 2019) ja yhteinäisten ja arvojen sopiminen (Kalenda ym., 2018).

6.2 Käytännön implikaatiot

Tutkimuksen havaintojen pohjalta esitämme taulukossa viisi esitettyjä käytännön ehdotuksia. Käytännön ehdotukset pohjautuvat kipupistemaliin ja siinä esitettyihin ratkaisuehdotuksiin. Käytännön ehdotukset pohjautuvat empiiristen havaintojen perusteella tehtyihin päätelmiin hankkeen nykytilasta.

Ensimmäinen primäärihavainto (PEC1) kertoo, että hankkeen haastateltavat ymmärtävät miten he tekevät töitä ja millä menetelmillä. Hankkeessa haastateltavat kertovat scrumista hyvin, joka on merkki siitä, että he ymmärtävät scrumin merkityksen työlle. Osa haastateltavista mainitsi scrumin lisäksi myös SAFen tai sen kompetensseja. Haastateltavista osa kertoi, että he hyödyntävät skaalautuvista ketteristä kehitysmenetelmistä tunnettuja piirteitä. Osa ei maininnut SAFEa lainkaan, mutta kertoi skaalautuvista ketterien kehitysmenetelmien tapahtumista, kuten scrum-of-scrumsista. Toisena primäärihavaintona (PEC2) löysimme, että hankkeessa on piirteitä myös muistakin skaalatuista ketteristä kehitysmenetelmistä, mutta näitä haastateltavat eivät maininneet. Hankkeessa käytetty kehitysmenetelmässä sisältää samanlaisia piirteitä kuin SAFE ja LeSS, sekä sisältää Essencen. Hankkeessa hyödynnettävään ketterään kehitysmenetelmään on tehty räätälöintejä, joka tekee siitä uniikin, joka on kolmas primäärihavainto (PEC3). Yhteisesti näille kolmelle primäärihavainnolle ehdotamme, että hankkeen sisäisesti tulisi lisätä tietoisuutta mitä ketteriä kehitysmenetelmiä on käytössä ja selvittää miten niitä käytetään. Ketteriä kehitysmenetelmiä on muokattu organisaatioon sopivaksi, mikä on hyvä asia. Hankkeessa tulisi kuitenkin ymmärtää miksi kehitysmenetelmää on muokattu juuri sellaiseksi kuin se on. Ketterän kehitysmenetelmän prosessien ja eri vaiheiden ymmärtämiseksi suosittelemme Essence-korttien hyödyntämistä, vaikka Essenceä ei otettaisi käyttöön kokonaisuudessaan. Korttien avulla on mahdollista selvittää kehitysprosessien eri vaiheita ja riippuvuussuhteita. Kortit mahdollistavat myös toiminnan kriittistä tarkastelua ja prosessien leanausta.

Neljäntenä primäärihavaintona (PEC4) hankkeessa hyödynnetään rooleja, joita ei esiinny muissa skaalautuvissa ketterissä kehitysmenetelmissä sellaise-

naan. Tällaisia rooleja ovat mm. hankepäällikkö sekä projektipäällikkö. Perinteisesti scrum ei ketteränä kehitysmenetelmänä sisällä kuin kolme roolia: tiimin jäsen, scrum master sekä tuoteomistaja. Skaalautuvissa ketterissä kehitysmenetelmissä hyödynnetään rooleja monipuolisemmin, mutta usein scrumiin perustuen. Havaitsimme, että hankkeessa esiintyy perinteisiä projektinhallinta rooleja, jotka voidaan nähdä uusien ja vanhojen toimintatapojen kohtaamisen kipupisteinä (PEC8). Ketterät kehitysmenetelmät on oletettavasti otettu käyttöön organisaatiossa vanhoja rooleja kunnioittaen. Hankkeessa olisi hyvä pohtia myös eri roolien suhteita ja riippuvuuksia kehitysmenetelmän prosesseihin ja näin arvioida mitä rooleja tarvitaan ja onko rooleissa päällekkäisyyksiä. Hankkeessa voitaisiin myös arvioida vastaavatko nykyiset roolit jonkin skaalautuvan ketterän kehitysmenetelmän rooleja, jolloin skaalautuvan ketterän kehitysmenetelmän käyttöönotto helpottuisi. Roolien ja prosessien välistä suhdetta on mahdollista havainnollistaa Essence-korttien avulla.

Primäärihavainto kuusi (PEC6) on havainto hankkeen tiimien autonomisesta työskentelystä. Kehitystiimit saavat itse päättää kuinka ja miten pitävät esimerkiksi päivittäiset palaverit ja sprintin suunnittelut. Tiimit hyödyntävät retrospektiivejä kehittääkseen ja arvioidakseen omaa toimintaa, jota myös kehitetään myös hankkeen laajuisessa skaalatussa retrospektiivissä. Tiimien autonomiaa ei nähdä kipupisteinä hankkeessa, ja suosittelemme sitä myös esitetyn tieteellisen kirjallisuuden valossa. Tiimien itseorganisoituminen ja autonomia edistää ketteryyden käyttöönottoa. Osa tiimeistä voisi hyödyntää myös muita ketteriä kehitysmenetelmiä, kuten kanbania. Tiimit voivat toimia niin autonomisesti kuin haluavat, kunhan tiimien välinen kadenssi / rytmi on yhtenäinen.

Taulukko 6 Käytännön ehdotukset

Tunniste	Empiirinen havainto	Käytännön ehdotus
PEC1, PEC2, PEC3, PEC5	Käytettävää ketterää kehitysmenetelmää on skaalattu hankkeen tarpeiden mukaisesti. Havaitsemme hankkeessa olevan skaalautuvien ketterien kehitysmenetelmien, kuten SAFE ja LeSS, piirteitä.	Ketterän kehitysmenetelmän prosessien ja eri vaiheiden tunnistamiseksi ja ymmärtämiseksi suosittelemme Essence-korttien hyödyntämistä.
PEC4, PEC8	Hankkeessa havaitaan hyödynnettävän scrum-roolien ohella skaalautuvissa ketterissä kehitysmenetelmissä esiintyviä rooleja.	Suositlemme nykyisten ja skaalautuvissa ketterissä menetelmissä esiintyvien roolien tarkempaa vertailua ja niiden peilaamista kehitysmenetelmän prosesseihin.
PEC6	Tiimit työskentelevät autonomisesti scrum-mallin mukaisesti, jota kehitetään ja sovelletaan tarpeiden mukaan.	Osa tiimeistä voisi hyödyntää myös muita ketteriä kehitysmenetelmiä, mikäli hankkeen yhteinen rytmi säilyy.
PEC9	Hankkeessa on laadunvarmistukseen liittyviä haasteita. Haasteena ovat vaatimusmäärittelyn ongelmat, jotka johtuvat teknisen osaamisen tai resurssien puutteesta.	Hankkeessa tulisi sopia dokumentoinnin käytänteistä ja vastuista.
PEC10,	Koko organisaatiossa ketteryyttä	Hankkeen työntekijöiden lisäksi myös

- PEC12 ei tunneta. Hanketasolla on kuitenkin panostettu työntekijöiden ketterään osaamiseen ja valmennukseen, mutta tuoteomistajien osaaminen koetaan hankkeessa vielä riittämättömäksi tekniseen vaatimusmäärittelyyn.
- PEC11 Etätyö on vähentänyt kasvokkain tapahtuvaa viestintää, joka koetaan haasteelliseksi kommunikoinnin kannalta. Muuten tiimien välinen viestintä koetaan toimivana ja läpinäkyvänä.
- hankkeen sidosryhmien ketterää osamista on hyvä edistää ketterän kehittämisen tueksi.
- Viestinnän käyttöönotto useissa eri kanavissa muodollisemman ja intensiivisen viestinnän lisäämiseksi.

Hankkeessa esiintyi laadunvarmistukseen liittäviä haasteita, joka on tutkielman havainto yhdeksän (PEC9). Laadunvarmistukseen liittyvät haasteet koskivat vaatimusmäärittelyn ongelmia, jotka johtuvat teknisen osaamisen tai resurssien puutteesta. Vaatimusmäärittelyä ja tarinoita tehdään liian laajasti, jolloin halutuista ominaisuuksista tulee liian suuria. Myös tekninen määrittely on heikommassa asemassa. Tämä on kuitenkin hankkeen tasolla huomioitu lisäämällä tiimiin tekninen tuoteomistaja, jonka tarkoituksena on perinteisen vaatimusmäärittelyn lisäksi myös teknisen näkökulman huomioiminen. Laadunvarmistuksen liittyvinä kipupisteinä nähdään myös dokumentoinnin ongelmat. Dokumentointi jää haastattelujen perusteella tekemättä ja usein dokumentoinnin sisältämä tieto on myös vanhaa. Hankkeessa tulisi selvittää yhteisten dokumentointikäytänteiden säännöt ja roolit: kuka vastaa minkäkin dokumentin päivittämisestä ja ylläpitämisestä. Myös resurssien lisääminen tekniseen määrittelyyn auttavat luomaan tarkempaa vaatimusmäärittelyä.

Kymmenes (PEC10) ja kahdestoista havainto (PEC12) liittyvät siihen, että organisaatiotasolla ketteryyttä ei tunneta. Hankkeessa ymmärretään ja toimitaan ketterien periaatteiden pohjalta ja siihen myös panostetaan. Ketteryyden kehittämistä varten hankkeessa on mukana ketterä valmentaja, joka tarkastelee nykyisiä toimintatapoja yhdessä tiimien kanssa. Hankkeen sidosryhmät eivät myöskään välttämättä ymmärrä mitä ketteryyden tarkoitus on, joka voi haitata ketteryyden toimintaa ja sen tehokkuutta. Ketteryyttä olisi mielekäästä skaalata ylöspäin koko organisaatiotasolle, mutta tämä ei välttämättä ole aina mahdollista organisaation moninaisuuden ja laajuuden vuoksi. Sidoryhmien ymmärryksen edistäminen ketterästä kehittämisestä voisi auttaa ketteryyden skaalautumista myös muualle kuin hanketasolle.

Yhdestoista primäärihavainto (PEC11) liittyy kasvaneen etätyön aiheuttamaan informaatiovajeeseen. Etätyön vaikutuksena työntekijöiden sosiaalisen kanssakäymisen siirtyminen verkkopalveluihin, kuten Teamsiin ja Slackiin, on vähentänyt niin sanottujen ”kahvipöytäkeskustelujen” pitämistä. Tämän nähdään vähentäneen spontaanin viestinnän tavoitettavuutta ja määrää. Etätyöhön siirtyminen on myös aiheuttanut osalle haastateltavista ongelmia ajankäytön suhteen. Monet palavereista voidaan pitää peräkkäin, mikä vaikeuttaa työajan tauottamista. Haastateltavat kuitenkin näkivät tiimien välisen kommunikaation

toimivana ja läpinäkyvänä. Viestinnän tavoitettavuuden vuoksi suosittelemme muodollisen ja intensiivisen viestinnän käyttöä useassa eri kanavassa, jotta viesti tavoittaa mahdollisimman monet työntekijöistä.

7 SÄRKYLÄÄKETTÄ KIPUPISTEISIIN

Tämän luvun tarkoitus on koota tutkielman kirjallisuuskatsaus sekä empiirinen osio. Tässä luvussa esitellään tutkielman tulokset vastaamalla tutkimuskysymyksiin. Lisäksi luvussa pohditaan tutkielman luotettavuutta, rajoituksia ja mahdollisia jatkotutkimusaiheita.

7.1 Tutkimuskysymyksiin vastaaminen

Tutkielman tarkoituksena on ollut tutustua skaalautuvia ketteriä kehitysmenetelmiä käsittelevään tieteelliseen kirjallisuuteen ja selvittää millaisia haasteita ja menestystekijöitä skaalautuvat ketterät kehitysmenetelmät kohtaavat. Kirjallisuuskatsauksen pohjalta toteutettiin tapaustutkimus.

Ensimmäinen tutkimuskysymys on:

1. Millaisia rooleja ja menetelmiä skaalautuvaan ketterään kehittämiseen kuuluu?

Tutkielmassa avattiin skaalautuvien ketterien kehitysmenetelmien, SAFe, LeSS ja DA, erityspiirteitä. Näiden lisäksi tutkielman kohteena oli myös skaalattu Essence. Skaalautuvien ketterien kehitysmenetelmien havaittiin sisältävän piirteitä scrumista niin menetelmien kuin roolien perusteella. Myös skaalattu Essence sisältää scrumimaisia piirteitä, vaikka tässä menetelmässä ei puhutakaan scrumin termein. Skaalatut ketterät kehitysmenetelmät eroavat perinteisistä ketteristä kehitysmenetelmistä niiden skaalattujen menetelmien ja tapahtumien avulla. Tällaisia ovat esimerkiksi skaalatut retrospektiivit, skaalatut sprintin tarkastelut sekä skaalattu päiväpalaveri scrum of scrums. Skaalautuneilla tapahtumilla haetaan samanlaisia hyötyjä kuin perinteisistä tapahtumista. Skaalaamalla esimerkiksi päivittäiset palaverit myös tiimien väliseksi palavereiksi viestintä tiimien ja organisaation eri osa-alueiden välillä tehostuu. Tutkittavassa hankkeessa hyödynnetään skaalautuvien ketterien kehitysmenetelmien piirteitä

ja rooleja. Hankkeessa käytettävään ketterään kehitysmenetelmään on lisätty myös muita rooleja ja menetelmiä, joita ei esiinny sellaisenaan muissa menetelmissä. Tällaisia menetelmiä olivat tuoteomistajan aika (PO-aika), arkkitehtuurikokous sekä refaktorointiviikko. Rooleja puolestaan olivat hanke- ja projektipäällikkö. Skaalattujen ketterien kehitysmenetelmien sekä hankkeen roolit ja menetelmät on esitetty visuaalisessa muodossa kappaleessa 6.1, josta selviää millaisia yhtenäisiä tapahtumia ja rooleja ketterillä kehitysmenetelmillä on, sekä kuinka ne eroavat toisistaan.

Toinen tutkimuskysymys on:

2. Millaisia kipupisteitä tiimit kohtaavat skaalautuvassa ketterässä kehittämisessä?

Tutkielmassa selvitettiin, millaisia menestystekijöitä ja haasteita skaalatut ketterät kehitysmenetelmät kohtaavat. Lähdekirjallisuuteen tutustumalla selvitettiin useimmiten esiintyneet menestystekijät sekä haasteet, joiden avulla luotiin erityinen kipupistemalli. Lähdekirjallisuudesta löytyy myös paljon muita menestystekijöitä ja haasteita, jotka eivät mahtuneet kipupistemalliin. Eniten tieteellisessä kirjallisuudessa esiintyneitä kipupisteitä voidaan pitää huomionarvoisimpina muuttujina. Tällaisia kipupisteitä olivat muutosvastarinta, konseptien ja ketteryyden määrittely, tiimin koordinointi ja autonomia, johdon tuen puute, uusien ja vanhojen toimintatapojen kohtaaminen, laadunvarmistus, ketterän koulutuksen ja valmennuksen puute sekä huono viestintä. Kipupistemalliin valikoitiin myös kirjallisuudesta löytyneitä ehdotuksia kipupisteiden ratkaisemista varten. Hankkeessa esiintyi kipupistemallissa esiintyneitä kipupisteitä. Huomionarvoisena seikkana muutosvastarintaa ja johdon tuen puutetta ei havaittu, vaikka nämä ovat tieteellisessä kirjallisuudessa eniten esiintyviä kipupisteitä.

Kolmas tutkimuskysymys on:

3. Millaisilla keinoilla skaalautuvien ketterien kehitysmenetelmien kipupisteitä voidaan ratkaista?

Skaalautuvien ketterien kehitysmenetelmien ongelmia voidaan ratkaista tutkimalla tieteellisestä kirjallisuudesta löytyneitä esityksiä ja arvioimalla erilaisten skaalautuvien menetelmien, roolien ja tapahtumien hyödyntämistä organisaatiossa. Kipupisteet ovat usein monisyisiä, joten yksittäisen kipupisteen ratkaiseminen ei tosin välttämättä ratkaise kaikkia ongelmia, sillä osa kipupisteistä esiintyvät yhtäaikaaisesti. Tällaisia kipupisteitä ovat esimerkiksi muutosvastarinta, johdon tuen puute sekä uusien ja vanhojen toimintatapojen kohtaaminen. Kipupisteiden ratkaisemista varten loimme kirjallisuuskatsauksen pohjalta erityisen kipupistemallin, jonka avulla on mahdollista havaita ja ratkaista kipupisteitä.

7.2 Tutkimuksen luotettavuus ja rajoitukset

Tutkielma on toteutettu tapaustutkimuksena laadullisin menetelmin. Tutkimuksessa rajoituksena voidaan pitää kerätyn aineiston sijoittumista yhteen organisaation ja tämän yhteen hankkeeseen, jolloin tutkimuksen tulokset ovat yleistettävissä kapeaan segmenttiin. Tutkimuksen tulokset ovat kuitenkin siirrettävissä samankaltaisiin organisaatioihin. Tutkimuksessa haastateltiin yhteensä seitsemän haastateltavaa. Laadulliseen tutkimukseen haastateltavien määrä on tutkijoiden päätettävissä. Ohjenuorana pidetään haastateltavien määrän olevan silloin sopiva, kun uutta tietoa ei synny lisää. (Hirsjärvi, Remes & Sajavaara, 2007.)

Jokainen haastattelu on kestänyt noin yhden tunnin yhtä haastattelua lukuun ottamatta. Haastatteluissa nousi samankaltaisia teemoja esille. Rajoituksena voidaan pitää tutkittavien henkilöiden työtehtävien ja roolien homogeenisyyttä. Suurin osa haastateltavista toimii organisaatiossa johto- tai hallinnollisissa tehtävissä, jonka vuoksi tutkimuksen tulosten yleistettävyys voi rajoittua. Tutkimuksessa käytetty aineisto ja sen tulkinta ovat rajoittuneita pro gradu tutkielman mielekkääseen rajaamiseen.

7.3 Jatkotutkimusaiheet

Skaalatut ketterät kehitysmenetelmät ovat vielä suhteellisen uusi ilmiö ja tutkiminen on vielä vähäistä. Tutkimusteemoina ovat yleensä tiimien välinen koordinointi sekä skaalautuvan ketteryyden muutos. Myös liiketoiminnan ketteryys, tiedon jakaminen skaalautuvissa ketterissä kehitysmenetelmissä sekä ketteryyden skaalaaminen ovat olleet viime vuosien tutkimusaiheita. (Moe & Dingsøyr, 2017.)

Skaalautuvien ketterien viitekehysten tutkimus on vielä vajavaista niiden omaksumisen, käytön, tehokkuuden ja haasteiden osalta (Conboy & Carroll, 2019). Kiinnostavia jatkotutkimusaiheita voisivat olla skaalautuvien ketterien kehitysmenetelmien omaksumisen, käytön ja tehokkuuden näkökulmasta. Essenceä käsittelevä tutkimus on hyvin suppeaa, eikä tieteellisiä julkaisuja sen käytöstä tai hyödyistä juurikaan ole. Haasteena voi olla Essencen vähäinen tunnettuus ja tieteellisen kirjallisuuden vähäinen esiintyvyys. Jatkotutkimuksena Essencen tutkiminen toimintatutkimuksen avulla voisi lisätä tietoisuutta sen todellisista hyödyistä tai ongelmista.

LÄHTEET

- Abrahamsson, P., Conboy, K. & Wang, X. (2009). 'Lots done, more to do': the current state of agile systems development research. *European Journal of Information Systems*, 18(4), 281-284
- Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. (2002) Agile software development methods: Review and analysis, VTT publication 478, Espoo, Finland, 107p.
- Agile Manifesto (2001). Agile Manifesto. Haettu 25.10.2020 osoitteesta <http://agilemanifesto.org>
- Ahmad, M. O., Markkula, J., & Oivo, M. (2013, September). Kanban in software development: A systematic literature review. In 2013 39th Euromicro conference on software engineering and advanced applications (pp. 9-16). IEEE.
- Ambler, S. W., & Lines, M. (2016, January). The disciplined agile process decision framework. In *International Conference on Software Quality* (pp. 3-14). Springer, Cham.
- Bass, J., Beecham, S., Razzak, M., Canna, C. & Noll, J. (2018). An empirical study of the product owner role in scrum. *ISCE '18: Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings*. 123-124
- Beck, K. (1999). Embracing change with extreme programming. *Computer*, 32(10), 70-77.
- Beck, K. (2000). *Extreme Programming Explained: Embrace Change*.
- Boehm, B. (1988). *IEEE Computer*. A spiral model of software development and enhancement.
- Brown, A. W., Ambler, S., & Royce, W. (2013, May). Agility at scale: economic governance, measured improvement, and disciplined delivery. In 2013 35th International Conference on Software Engineering (ICSE) (pp. 873-881). IEEE.
- Conboy, K., & Carroll, N. (2019). Implementing large-scale agile frameworks: challenges and recommendations. *IEEE Software*, 36(2), 44-50.
- Cruzes, D. S., & Dybå, T. (2011, September). Recommended steps for thematic synthesis in software engineering. In 2011 international symposium on empirical software engineering and measurement (pp. 275-284). IEEE.
- Diebold, P., Schmitt, A. & Theobald, S. (2018). Scaling agile: how to select the most appropriate framework. *XP '18 Proceedings of the 19th International Conference on Agile Software Development: Companion*, 7, 1-4

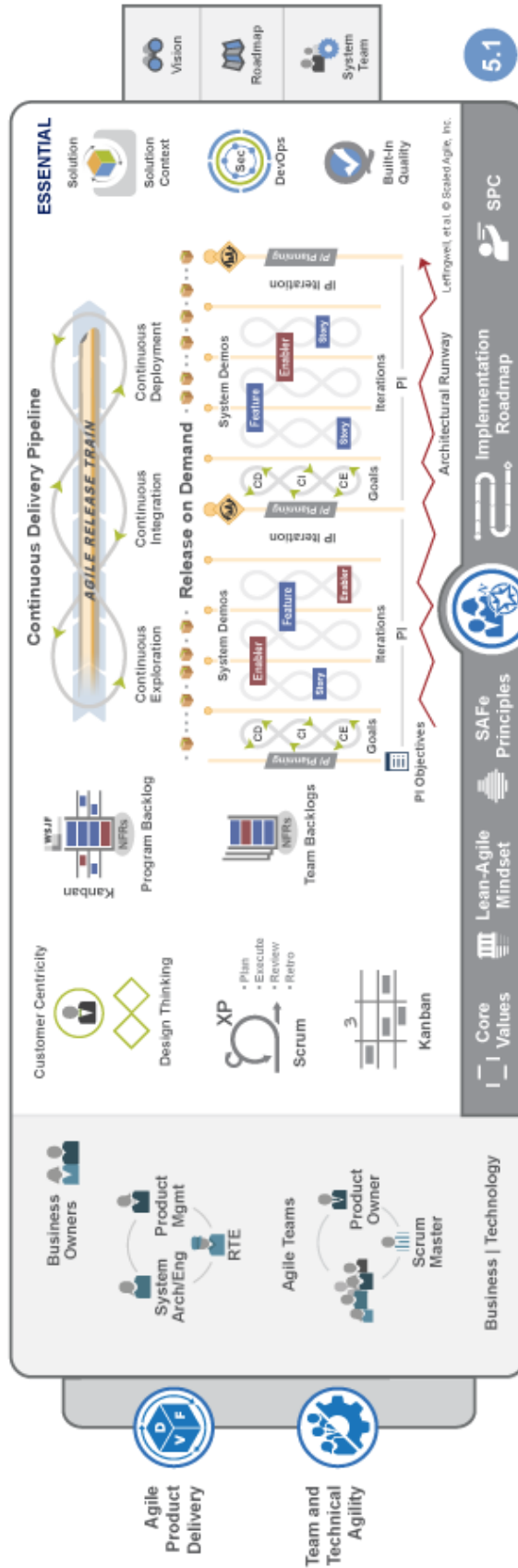
- Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, 119, 87-108.
- Dybå, T. & Dingsøy, T. (2009). What do we know about agile software development? *IEEE Software*, 26(5). 6-9
- Ebert, C., Abrahamsson, P., & Oza, N. (2012). Lean software development. *IEEE Software*, (5), 22-25.
- Ebert, C., & Paasivaara, M. (2017). Scaling agile. *Ieee Software*, 34(6), 98-103.
- Edison, H., Wang, X. & Conboy, K. (2021). Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review. *IEEE Transactions on Software Engineering*.
- Eriksson, P. & Koistinen, K. (2005). Monenlainen tapaustutkimus. Kuluttajatutkimuskeskus.
- Hines, P., Holweg, M. & Rich, N. (2004). Learning to evolve. A review of contemporary lean thinking. *International Journal of Operations & Production management*, 24(10), 994-1011
- Hirsjärvi, S., Remes, P. & Sajavaara, P. (2007). Tutki ja kirjoita. (13., osin uudistettu painos.) Kustannusosakeyhtiö Tammi. Helsinki.
- Huang, S. & Ng, P.W. (2014). Essence as Framework for Conducting Empirical Studies. *Software Engineering: Methods, Modeling, and Teaching*, 3(2), 99-18.
- Hummel, M., Rosenkranz, C., & Holten, R. (2013). The role of communication in agile systems development. *Business & Information Systems Engineering*, 5(5), 343-355.
- Ivar Jacobson International SA. (2021a). Agile Essentials. Haettu 1.4.2021 osoitteesta <https://practicelibrary.ivarjacobson.com>
- Ivar Jacobson International SA. (2021b). Agile at Scale Essentials. Haettu 1.4.2021 osoitteesta <https://practicelibrary.ivarjacobson.com/content/agile-scale-essentials>
- Jacobson, I., Ng, P. W., McMahon, P. E., & Goedicke, M. (2019). The essentials of modern software engineering: free the practices from the method prisons!. Morgan & Claypool.
- Jacobson, I., Ng, P. W., McMahon, P. E., Spence, I., & Lidman, S. (2012). The essence of software engineering: the SEMAT kernel. *Communications of the ACM*, 55(12), 42-49.
- Kalenda, M., Hyna, P., & Rossi, B. (2018). Scaling agile in large organizations: Practices, challenges, and success factors. *Journal of Software: Evolution and Process*, 30(10), e1954.
- Kettunen, P., Laanti, M., Fagerholm, F., Mikkonen, T., & Männistö, T. (2019). Finnish enterprise agile transformations: a survey study. In *International Conference on Agile Software Development* (pp. 97-104). Springer, Cham.

- Kniberg, H., & Skarin, M. (2010). Kanban and Scrum-making the most of both. Lulu. com.
- Laanti, M. (2014). Characteristics and Principles of Scaled Agile. *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation*, 9-20
- Laanti, M., Similä, J., & Abrahamsson, P. (2013). Definitions of agile software development and agility. In European Conference on Software Process Improvement (pp. 247-258). Springer, Berlin, Heidelberg.
- The LeSS Company B. V. (2021). LeSS Framework. Haettu 15.2.2021 osoitteesta <https://less.works/less/framework/index> .
- Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software*, 82(11), 1869-1890.
- Moe, N. B., Dingsøy, T., & Dybå, T. (2010). A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology*, 52(5), 480-491.
- Moe, N.B. & Dingsøy, T. (2017). Emerging Research Themes and updated Research Agenda for Large-Scale Agile Development: A summary of the 5h International Workshop at XP2017. 1-4
- Myers, M. D., & Avison, D. (Eds.). (2002). *Qualitative research in information systems: a reader*. Sage.
- Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and organization*, 17(1), 2-26.
- Noll, J., Razzak, M., Bass, J. & Beecham, S. (2017). A Study of the Scrum Master's Role. *International Conference on Product-Focused Software Process Improvement. Lecture Notes in Computer Science*, 307-323.
- Ng, P. W., & Huang, S. (2013). Essence: A framework to help bridge the gap between software engineering education and industry needs. *26th International Conference on Software Engineering Education and Training*, 304-308 IEEE.
- Paasivaara, M. (2017). Adopting SAFe to scale agile in a globally distributed organization. *12th International Conference on Global Software Engineering*, 36-40. IEEE.
- Park, J., McMachon, P. & Myburgh, B. (2016). Scrum Powered by Essence. *ACM SIGSOFT Software Engineering Notes*, 41(1), 1-8
- Paulk, M. C. (2001). Extreme programming from a CMM perspective. *IEEE software*, 18(6), 19-26.
- Poppendieck, M. & Cusumano, M. (2012). Lean Software Development: A Tutorial. *IEEE Software*, 29(5), 26-32.

- Project Management Institute, Inc. (2021). Introduction to Disciplined Agile (DA). Haettu 3.4.2021 osoitteesta <https://www.pmi.org/disciplined-agile/introduction-to-disciplined-agile>.
- Putta, A., Paasivaara, M., & Lassenius, C. (2018). Adopting scaled agile framework (SAFe) a multivocal literature review. *Proceedings of the 19th International Conference on Agile Software Development: Companion*, 1-4.
- Rodríguez, P., Markkula, J., Oivo, M. & Turula, K. (2012). Survey on agile and lean usage in Finnish software industry. *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement (ESEM '12)*. ACM.
- Rodríguez, P., Mäntylä, M., Oivo, M., Lwakatare, L., Seppänen, P. & Kuvaja, P. (2019). Advances in Using Agile and Lean Process for Software Development. *Advances in Computers*, 113, 135-224
- Scaled Agile Inc. (2021). Achieving Business Agility with SAFe® 5. A Scaled Agile, Inc. White Paper February 2021. Haettu 15.2.2021 osoitteesta <https://www.scaledagile.com/resources/safe-whitepaper>.
- Schwaber, K. (1995). Scrum Development Process. *OOPSLA'95 Workshop on Business Object Design and Implementation*, Springer-Verlag.
- Schwaber, K. (1997). Scrum development process. *Business object design an implementation*, 117-134
- Schwaber, K. and M. Beedle (2002). *Agile Software Development With Scrum*. Upper Saddle River, NJ, Prentice-Hall.
- Schwaber, K. (2004). *Agile Project Management with Scrum*. Microsoft Press, Redmond
- Shameem, M., Kumar, C., Chandra, B., & Khan, A. A. (2017). Systematic review of success factors for scaling agile methods in global software development environment: A client-vendor perspective. *24th Asia-Pacific Software Engineering Conference Workshops*, 17-24
- Sharma, S. & Hasteer, N. (2016). A comprehensive study on state of Scrum development. *2017 International Conference on Computing, Communication and Automation*. 867-872
- Singleton, R. & Straits, B. (2018). *Approaches to Social Research*. Oxford University Press, USA.
- Steghöfer, J-P. (2019). Using Essence in a Software Development Methodologies Course: An Experience Report. *ISEE 2019, 2nd Workshop on Innovative Software Engineering Education*, 7-10
- Takeuchi, H., & Nonaka, I. (1986). The new new product development game. *Harvard business review*, 64(1), 137-146.

- Theobald, S., Schmitt, A. & Diebold, P. (2019). Comparing Scaling Agile Frameworks Based on Underlying Practises. XP 2019 Workshops, LNBIP 364, 88-96
- Vehkalahti, K. (2008). Kyselytutkimuksen mittarit ja menetelmät. Kustannusosakeyhtiö Tammi.
- Wang, X., Conboy, K., & Cawley, O. (2012). "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems and Software*, 85(6), 1287-1299.
- Williams, L. (2012). What agile teams think of agile principles. *Communications of the ACM*, 55(4), 71-76.

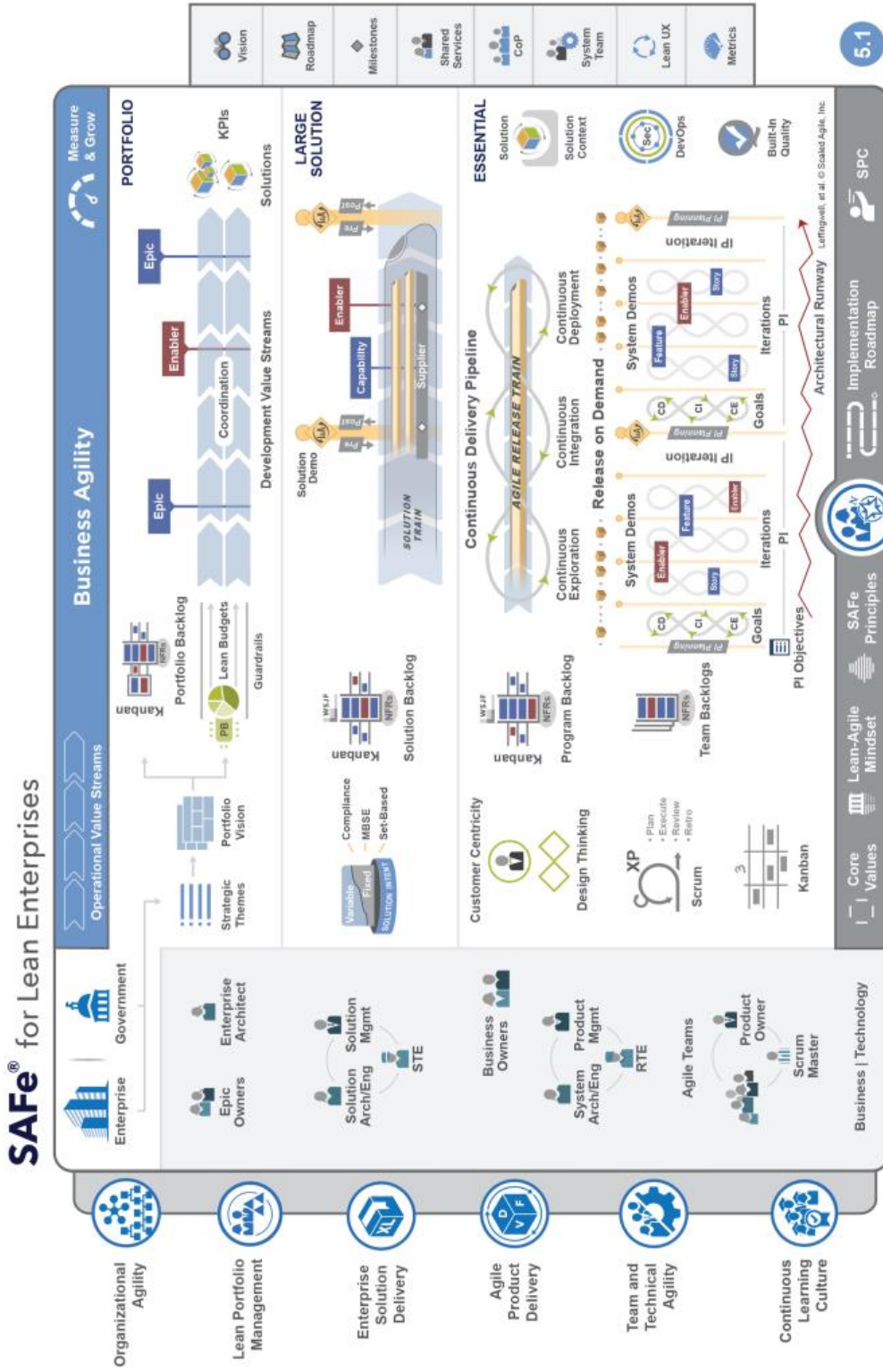
LIITE 1 ESSENTIAL SAFE



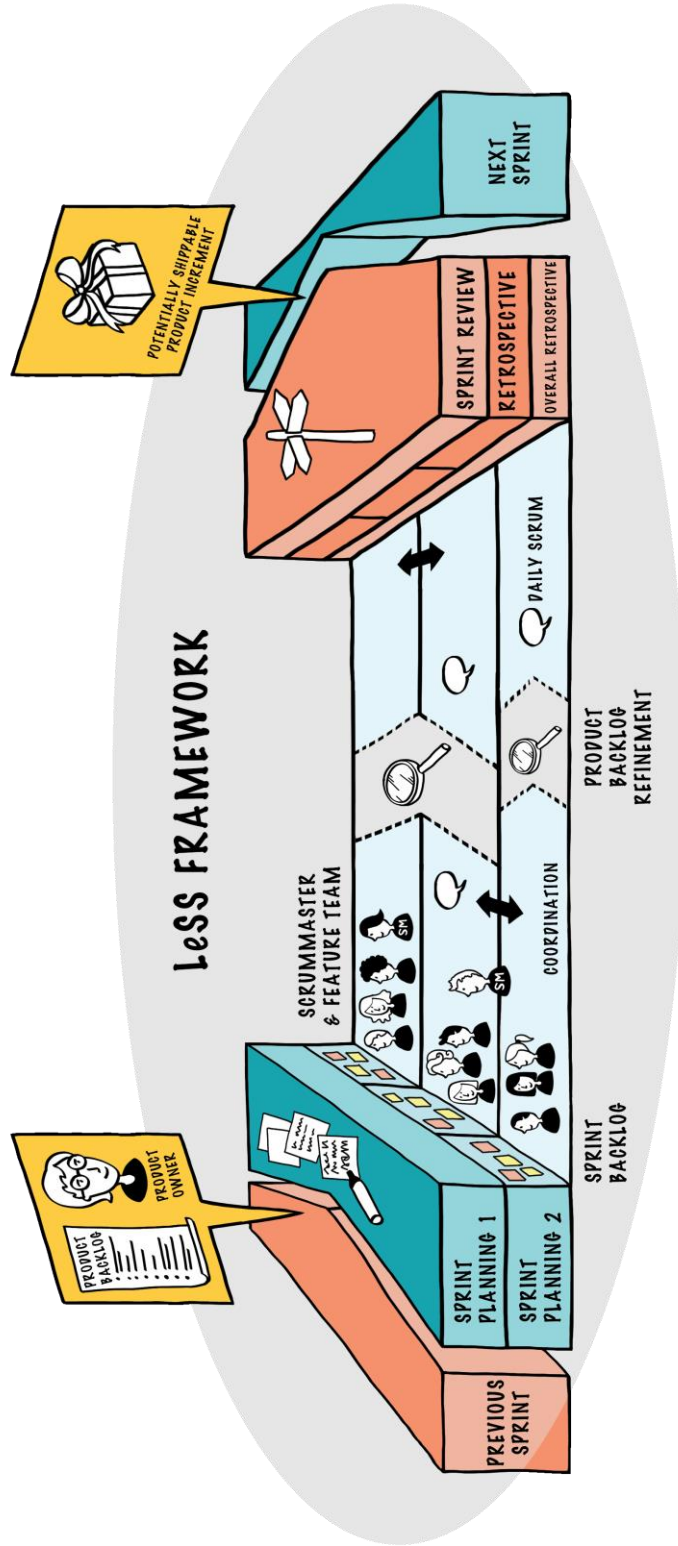
© Scaled Agile, Inc.

5.1

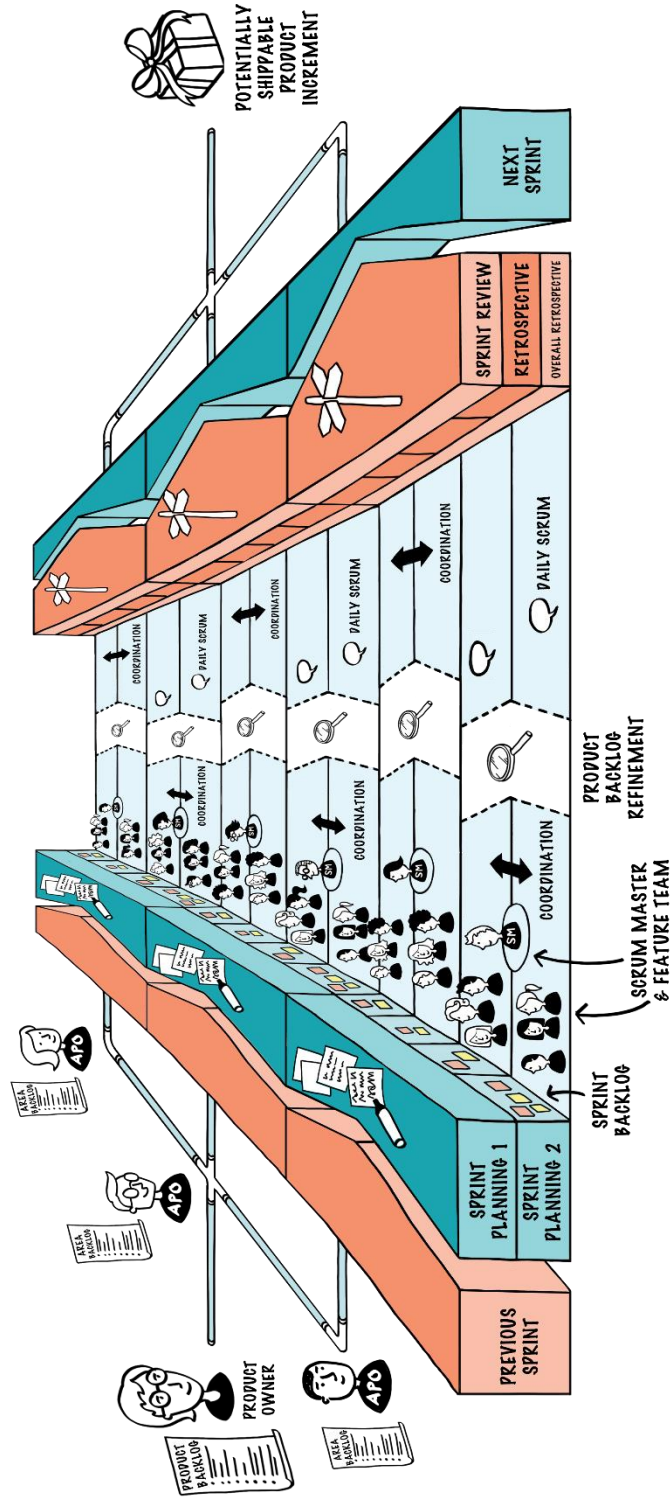
LIITE 2 FULL SAFE VIITEKEHYS



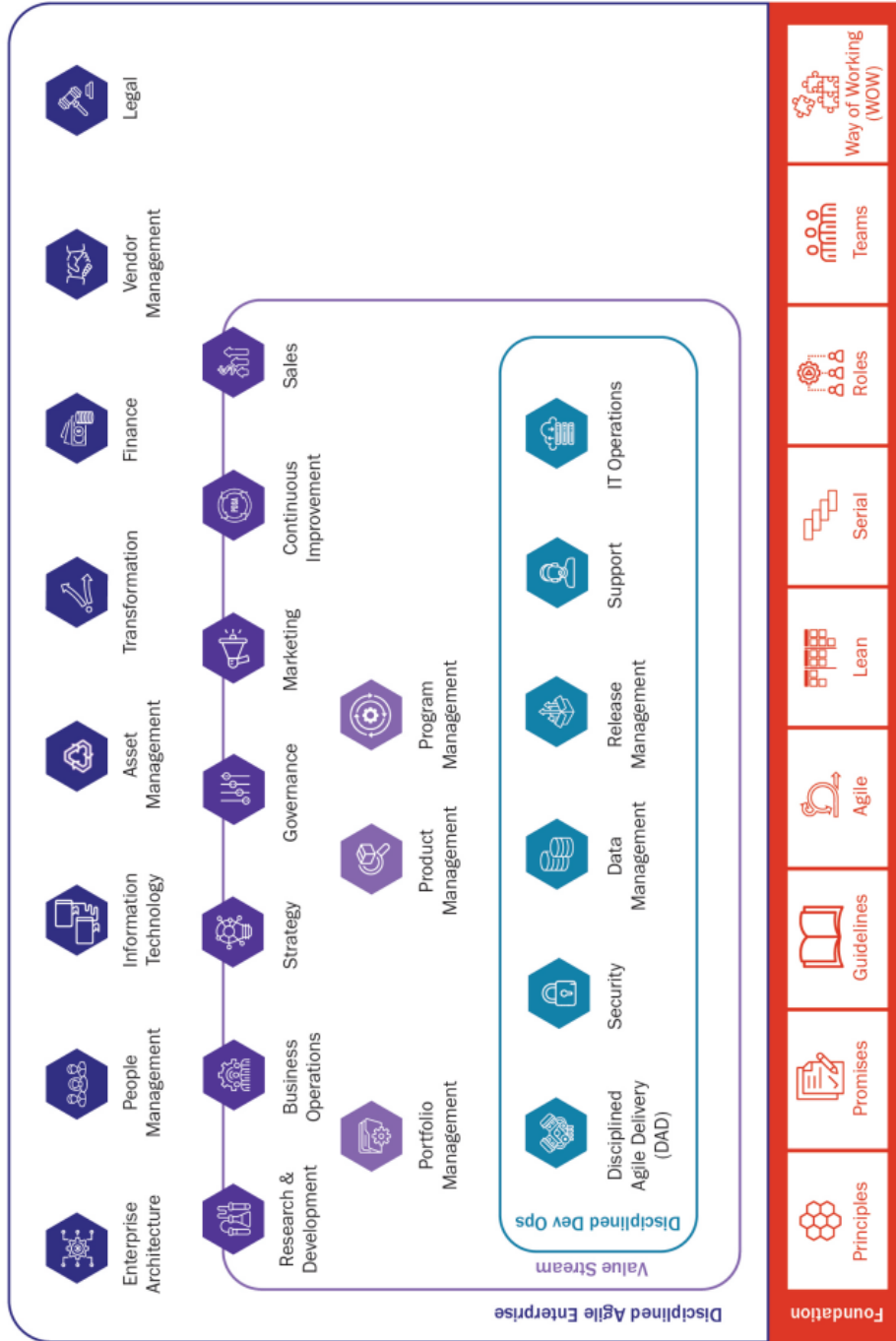
LIITE 3 LESS VIITEKEHYS



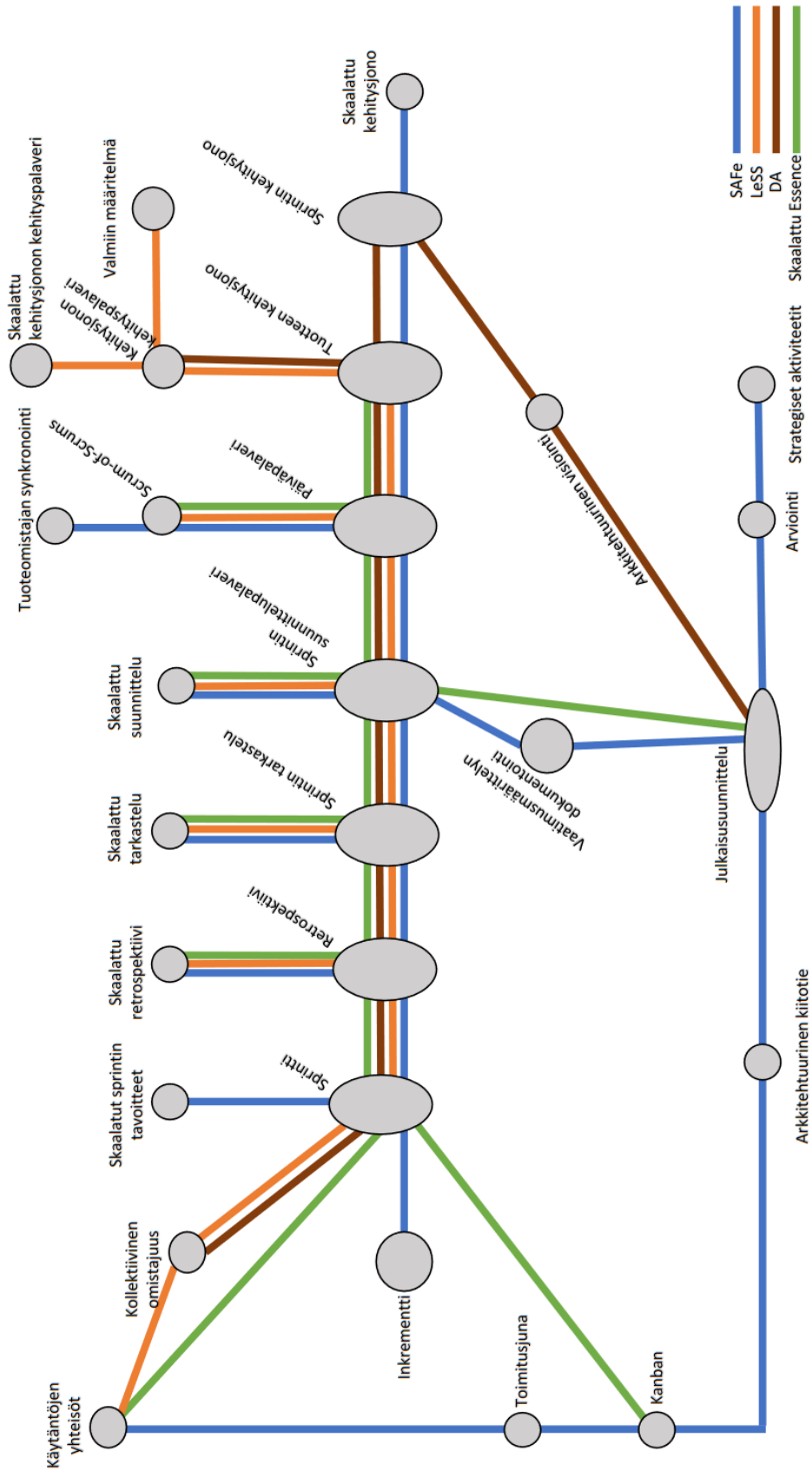
LIITE 4 LESS HUGE VIITEKEHYS



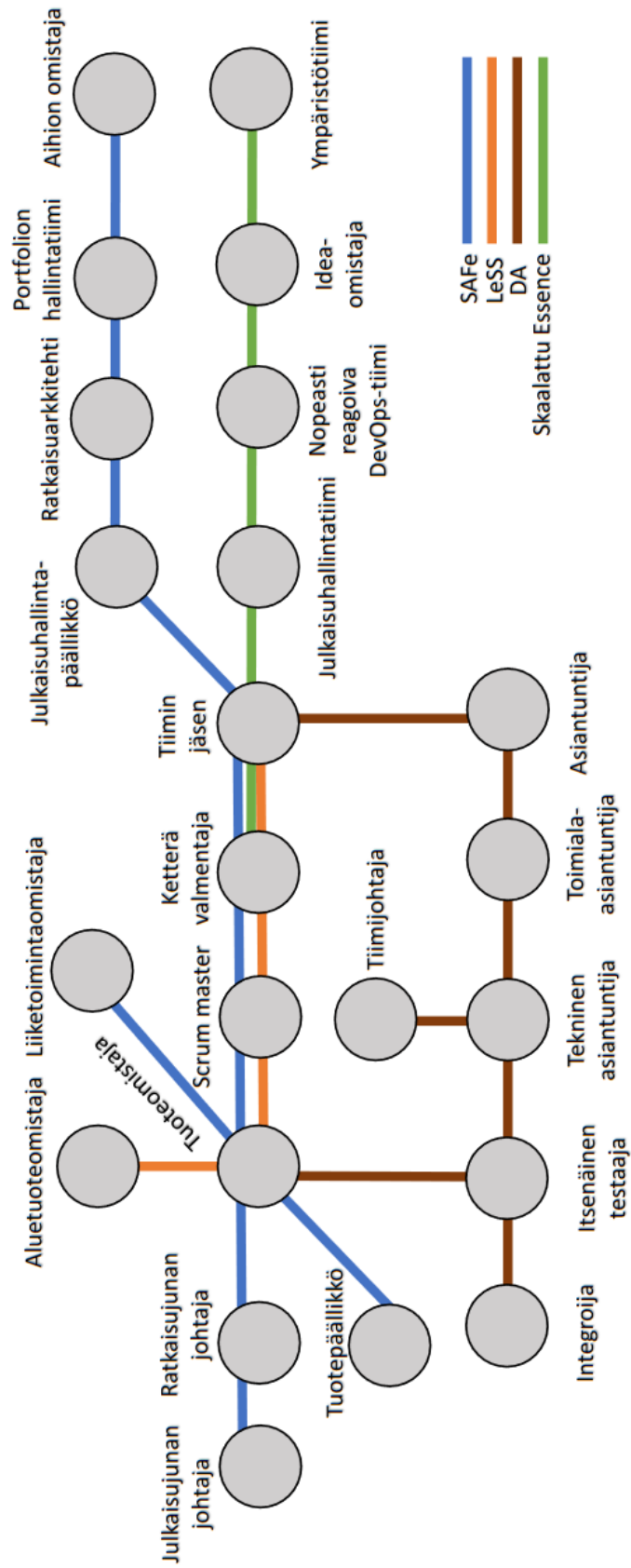
LIITE 5 DISCIPLINED AGILE (DA)



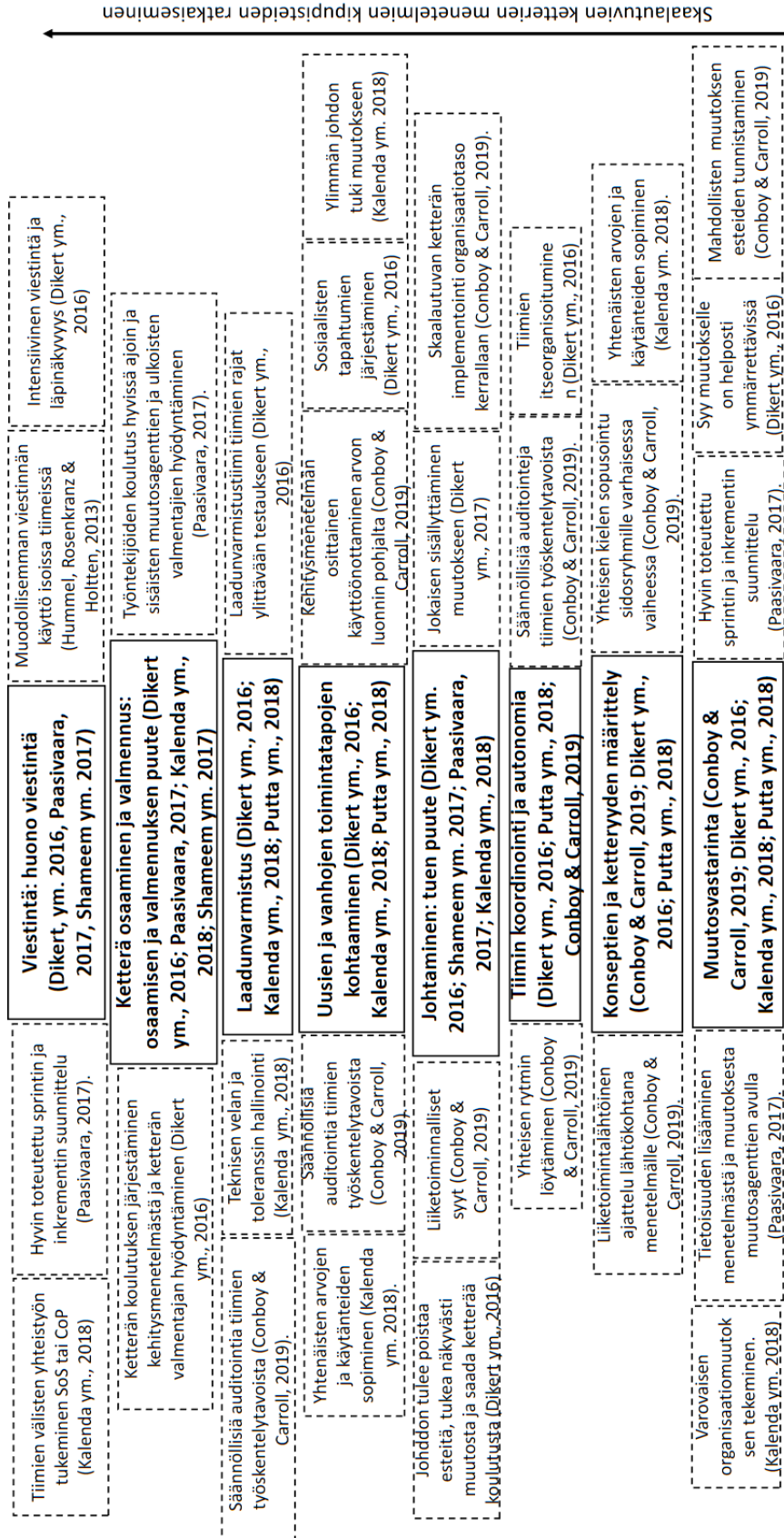
LIITE 6 METROKARTTA KEHITYSMENETELMISTÄ



LIITE 7 METROKARTTA ROOLEISTA

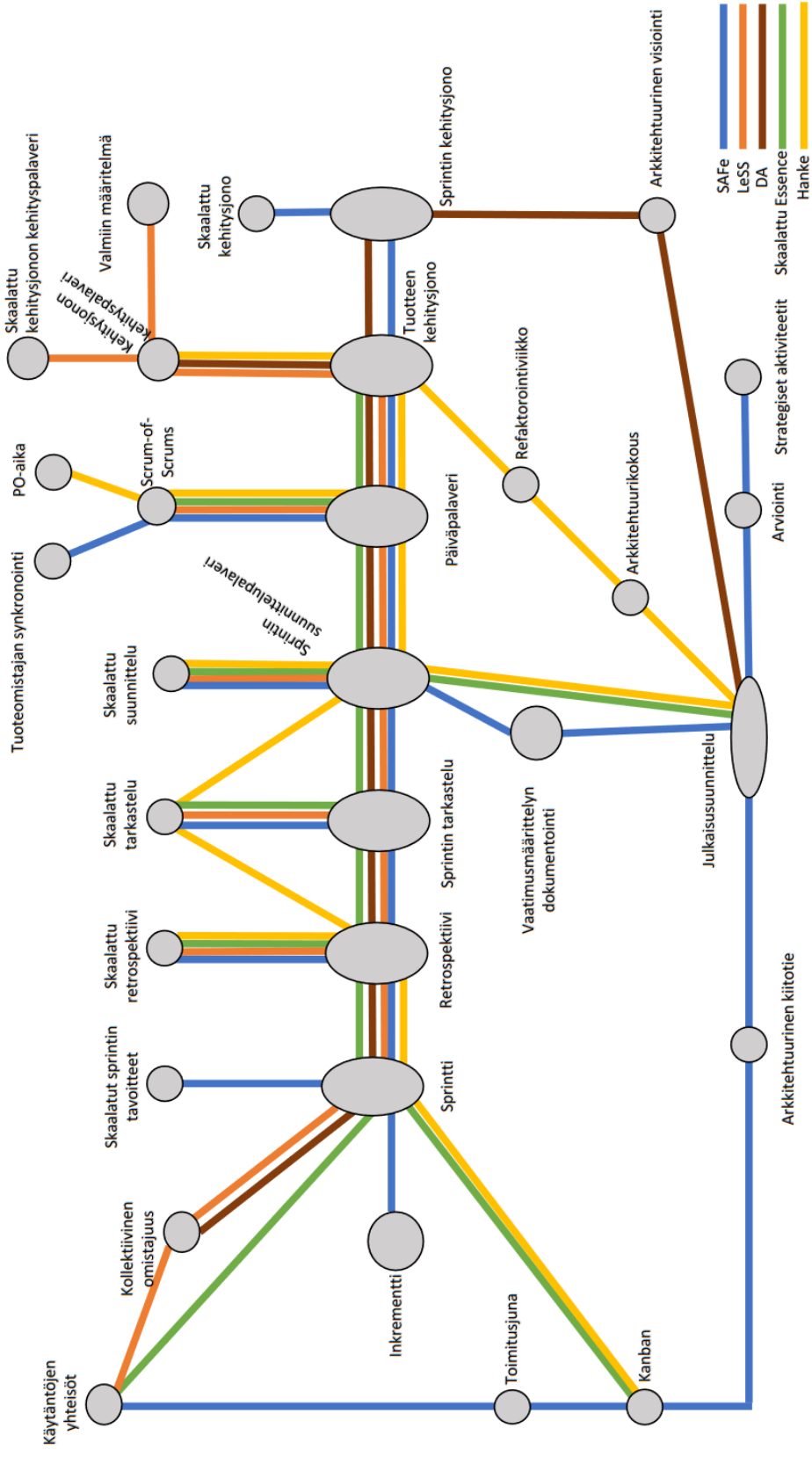


LIITE 8 KIPUPISTEMALLI



Kipupistemalli

LIITE 9 METROKARTTA HANKKEEN MENETELMISTÄ



LIITE 10 METROKARTTA HANKKEEN ROOLEISTA

