

Emmi Björkman

**HAJAUTETUN KETTERÄN OHJELMISTOKEHITYK-  
SEN KRIITTISET MENESTYSTEKIJÄT JA HAASTEET**



JYVÄSKYLÄN YLIOPISTO  
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA  
2020

## TIIVISTELMÄ

Björkman, Emmi

Hajautetun ketterän ohjelmistokehityksen kriittiset menestystekijät ja haasteet

Jyväskylä: Jyväskylän yliopisto, 2020, 87 s.

Tietojärjestelmätiede, pro gradu -tutkielma

Ohjaaja: Halttunen, Veikko

Monet yritykset lähtivät 2000-luvun alussa hyödyntämään ketterää ohjelmistokehitystä, mikä on seurausta nopeasti muuttuvasta tekniikasta ja liiketoimintaympäristöstä. Ketterät menetelmät keskittyvät epämuodollisiin prosesseihin ja suoraan viestintään koordinoinnin helpottamiseksi. Vaikka ketterä ohjelmistokehitys on tuonut mukanaan monia hyötyjä, niin se on koitunut erityisen haasteelliseksi maantieteellisesti laajasti hajautetuille tiimeille. Hajautettuun työhön liittyy paljon tietojärjestelmien ulkoistamista, mikä on myöhemmin vaikuttanut myös ohjelmistokehitysprojektien ketteryyteen. Ulkoistamisen merkitys on kasvanut, koska se on mahdollistanut kustannussäästöjä ja pääsyn suurempiin työvoimavaroihin. Ketteryyden ja hajautetun ohjelmistokehityksen yhdistäminen koetaan haasteena paitsi maantieteellisestä myös kulttuurillisesta ja ajallisesta etäisyydestä johtuen. Tämän pro gradu -tutkielman tutkimuskysymys oli *”Mitkä ovat hajautetun ketterän ohjelmistokehityksen kriittiset menestystekijät ja miten ne suhteutuvat koettuihin haasteisiin?”* Tutkimuksessa keskityttiin kohdeorganisaatioon, joka toimii monikansallisten ohjelmistokehitysprojektien parissa. Tutkielma koostui kirjallisuuskatsauksesta ja empiirisestä tutkimuksesta. Empiirinen aineisto kerättiin laadullisten teemahaastatteluiden ja kyselyn avulla. Aiempi hajautetun ketterän ohjelmistokehityksen kirjallisuus keskittyi paljolti haasteiden tunnistamiseen. Tässä tutkimuksessa tunnistettiin haasteita, jotka ovat linjassa aiempien tutkimustuloksien kanssa. Haasteita olivat muun muassa kasvotusten työskentelyyn liittyvien hyötyjen menetykset, tekniset haasteet, kulttuurierot ja kielihaasteet. Tutkimuksessa selvitettiin myös kriittisiä menestystekijöitä, jotka ovat sellaisia, joihin hajautetun kehitystiimin tulisi keskittyä ja sisällyttää toimintaansa, jotta projekti onnistuisi. Kriittisiä menestystekijöitä olivat kommunikointi, tiimin sisäinen yhteistyö, yhteinen ymmärrys ja suunta tavoitteista, henkilökohtaiset ominaisuudet, asiakasyhteistyö, tiimin kyvykkyys, tiimin toimintaympäristö, seuranta, koulutus ja oppiminen sekä päätöksenteon nopeus. Merkittävä tutkimuslöydös oli kuitenkin se, että moni tunnistettu kriittinen menestystekijä koettiin myös haasteena kohdeorganisaatiossa. Tämä tutkimus toi oman näkemyksensä kriittisistä menestystekijöistä ja haastoi myös kohdeorganisaatiota pohtimaan omia resurssejaan. Ketteryyden varmistaminen hajautetussa ohjelmistokehityksessä lähtee liikkeelle asianmukaisten resurssien valitsemisesta ja järjestämisestä projektiin.

Asiasanat: ketteruus, ketterä ohjelmistokehitys, hajautettu ohjelmistokehitys, kriittiset menestystekijät

## ABSTRACT

Björkman, Emmi

Critical Success Factors and Challenges of Distributed Agile Software Development

Jyväskylä: University of Jyväskylä, 2020, 87 pp.

Information Systems, Master's Thesis

Supervisor: Halttunen, Veikko

Many companies set out in the early 2000s to take advantage of agile software development as a result of rapidly changing technology and business environment. Agile methods focus on informal processes and direct communication to facilitate coordination. While agile software development has brought many benefits, it has proved particularly challenging for geographically distributed teams. Distributed work involves a lot of outsourcing of information systems, which has later also affected the agility of software development projects. The importance of outsourcing has grown as it has enabled cost savings and access to skilled personnel. Combining agility and distributed software development is seen as a challenge not only due to geographical but also cultural and temporal distance. The research question for this master's thesis was *"What are the critical success factors of distributed agile software development and how do they relate to the experienced challenges?"* The research focused on a target organization working on multinational software development project. The study consisted of a literature review and empirical research. Empirical material was collected through qualitative theme interviews and a qualitative survey. The previous literature on distributed agile software development focuses much on identifying challenges. This study was identified challenges that are in line with previous research findings. The challenges found in the study were related to the loss of benefits associated with working face-to-face, technical challenges, cultural differences and language challenges. The study also identified critical success factors that are those that a distributed development team should focus on and include its activities for the project to succeed. The identified critical success factors were communication, collaboration within the team, common understanding and direction of goals, personal characteristics, customer collaboration, team capability, team environment, control, training and learning and decision time. A significant finding is that many of the identified critical success factors were also seen as a challenge in the target organization. This study provided its own insight into the critical success factors and also challenges the target organization to consider its own resources. Ensuring agility in distributed software development starts with selecting and organizing the appropriate resources for the project.

Keywords: agility, agile software development, distributed software development, critical success factors

## KUVIOT

KUVIO 1 Esimerkki Kanban-taulusta. ....	17
KUVIO 2 Vesiputousmallin kehitys Extreme Programming -menetelmäksi .....	18
KUVIO 3 Mukautuvan ohjelmistokehityksen vaiheet. ....	25
KUVIO 4 Ketteryyden kolme ulottuvuutta. ....	28
KUVIO 5 Chow'n ja Caon esittämät ketterän ohjelmistokehityksen hypoteettiset menestystekijät.....	33
KUVIO 6 Misran ym. esittämät ketterän ohjelmistokehityksen hypoteettiset menestystekijät.....	34
KUVIO 7 Kyselyn tulokset .....	68
KUVIO 8 Hajautetun ketterän ohjelmistokehityksen kriittiset menestystekijät	72

## TAULUKOT

TAULUKKO 1 Ketterän ohjelmistokehityksen määrittely .....	11
TAULUKKO 2 Offshoring vs. outsourcing .....	23
TAULUKKO 3 Tutkimuksen tarkoitus .....	38
TAULUKKO 4 Tutkimustuloksien vertailu aiempiin tutkimuksiin.....	73

# SISÄLLYS

TIIVISTELMÄ .....	2
ABSTRACT .....	3
KUVIOT .....	4
TAULUKOT .....	4
SISÄLLYS.....	5
1 JOHDANTO.....	7
2 KETTERÄ OHJELMISTOKEHITYS .....	10
2.1 Ketterän ohjelmistokehityksen määrittely .....	10
2.2 Ketterän ohjelmistokehityksen periaatteet .....	13
2.3 Käytetyimmät ketterät menetelmät .....	14
2.3.1 Scrum .....	14
2.3.2 Kanban .....	16
2.3.3 Extreme Programming (XP).....	17
2.4 Perinteisestä lähestymistavasta siirtyminen ketterään lähestymistapaan .....	20
3 HAJAUTETTU KETTERÄ OHJELMISTOKEHITYS.....	21
3.1 Hajautettu työ.....	21
3.2 Hajautetun ketterän ohjelmistokehityksen määrittely .....	23
3.3 Hajautetun ketterän ohjelmistokehityksen tuomat haasteet.....	26
3.4 Ketteryyden toteuttaminen käytännössä hajautetussa ohjelmistokehityksessä .....	27
3.4.1 Resurssin ketteryys .....	28
3.4.2 Prosessin ketteryys.....	29
3.4.3 Osapuolten väliseen vuorovaikutukseen liittyvä ketteryys .....	29
4 KETTERÄN OHJELMISTOKEHITYKSEN KRIITTISET MENESTYSTEKIJÄT AIEMMAN TUTKIMUKSEN VALOSSA .....	31
4.1 Ketterän ohjelmistokehitysprojektin onnistumisen määrittely .....	31
4.2 Kriittisten menestystekijöiden määrittely .....	32
4.3 Ketterän ohjelmistokehityksen menestystekijät.....	32
4.4 Havaintoja hajautetun ketterän ohjelmistokehitysprojektin onnistumisesta ja menestystekijöistä .....	35
5 EMPIIRINEN TUTKIMUS .....	37
5.1 Tutkimusprosessi.....	37

5.2	Empiirisen tutkimuksen tarkoitus.....	38
5.3	Tutkimuksen lähestymistavan valinta .....	38
5.4	Tiedonkeruumenetelmä.....	39
5.5	Haastattelurunko ja kysely.....	41
5.6	Haastateltavien valinta ja haastattelujen toteutus .....	42
5.7	Aineiston analysointi.....	43
6	TUTKIMUSTULOKSET .....	45
6.1	Haastateltavien taustatiedot.....	45
6.2	Hajautettu ketterä ohjelmistokehitys.....	46
6.2.1	Toteutus .....	46
6.2.2	Tiimin kommunikointi .....	51
6.2.3	Odotukset hajautetulle ketterälle tiimille .....	53
6.3	Haasteet.....	53
6.3.1	Kasvotusten tapahtuvaan työskentelyyn liittyvien hyötyjen menetys .....	54
6.3.2	Tekniset haasteet .....	55
6.3.3	Kulttuurierot .....	56
6.3.4	Kielihaasteet .....	57
6.3.5	Muita esille nousseita haasteita.....	57
6.4	Hajautetun ketterän ohjelmistokehitysprojektin onnistuminen ja sen mittaaminen.....	59
6.4.1	Hajautetun ketterän ohjelmistokehitysprojektin onnistumisen määrittely.....	59
6.4.2	Onnistumisen mittaaminen .....	61
6.5	Haastateltavien esittämät hajautetun ketterän ohjelmistokehityksen kriittiset menestystekijät.....	62
6.6	Kysely kirjallisuuden esittämistä menestystekijöistä.....	67
7	TULOSTEN TULKINTA JA POHDINTA .....	70
7.1	Tuloksien vertailu aiempiin tutkimuksiin - Haasteet.....	70
7.2	Tuloksien vertailu aiempiin tutkimuksiin - Kriittiset menestystekijät.....	71
7.3	Hajautetun ketterän ohjelmistokehityksen kriittiset menestystekijät suhteessa koettuihin haasteisiin .....	74
7.4	Tutkimuksen luotettavuus ja rajoitteet.....	76
7.5	Jatkotutkimusaiheita .....	78
8	YHTEENVETO .....	79
	LÄHTEET.....	81
	LIITE 1 HAASTATTELUKYSYMYKSET .....	86
	LIITE 2 KYSELY.....	87

# 1 JOHDANTO

Monet ohjelmistokehittäjät ovat pitäneet 1990-luvulta lähtien kehitysprojektien alussa päätettäviä vaatimuksia koskevaa dokumentaatiota turhauttavana ja jopa mahdottomana, koska tekniikka ja liiketoimintaympäristö muuttuvat nopeasti (Williams & Cockburn, 2003, s. 39). Tämä johti myöhemmin ketterien menetelmien soveltamiseen kehitysprojekteissa, joissa vaatimukset voivat muuttua useampaan otteeseen. Ketterästä ohjelmistokehityksestä julkaistiin tasaiseen tahtiin tieteellisiä artikkeleita ja pidettiin konferensseja vuoteen 2010 saakka. Sen jälkeen julkaisujen määrä on alkanut vähetä (Dingsøyr, Nerur, Balijepally & Moe, 2012).

Vaikka ketterä ohjelmistokehitys on tuonut mukanaan monia hyötyjä, siitä on tullut haasteellista hajautetuissa tiimeissä. Erityisen haasteelliseksi se on muodostunut ulkoistetuissa projekteissa, kun tiimi on maantieteellisesti laajalle hajautettu. IT-alan kasvavien tarpeiden tyydyttämiseksi kehitystoimintaa on ulkoistettu maihin, joissa on tarjolla osaavaa henkilöstöä. Todellinen kiinnostus tietojärjestelmien ulkoistamiseen syntyi 1980-luvun puolivälissä. Ulkoistamisen tausta-ajatuksena on ollut se, että yritykset ovat luopuneet monipuolistamisstrategioistaan keskittyäkseen ydinosaamiseensa.

Globalisaation myötä hajautettujen tiimien, erityisesti virtuaalitiimien, suosio yleisty. Kun ohjelmistokehitys on hajautettu maantieteellisesti eri aika-vyöhykkeisiin, niin suora viestintä on muuttunut todelliseksi haasteeksi. Sen vuoksi on tärkeää kysyä, kuinka hajautetussa ympäristössä pystytään jatkamaan ketterän periaatteen tehokkuutta ja menestystä, jolle on ominaista suora viestintä.

Tämä tutkielma käsittelee hajautettua ketterää ohjelmistokehitystä. Hajautetusta työstä ja siihen liittyvistä tiimeistä on tutkimusta 20 vuoden takaa. Myös ketterää ohjelmistokehitystä on toteutettu jo pidemmän aikaa. Ketterien käytäntöjen hyödyntäminen yritysten erilaisissa kehitysprojekteissa on nykypäivää. Ketterän ohjelmistokehityksen julistusta (engl. Manifesto for Agile Software Development) myötäillen ketterän menetelmän käyttö perustuu tavallisimmin suoraan viestintään, mikä tarkoittaa sitä, että tiimin jäsenet työskentelevät samassa tilassa ja kohtaavat kasvokkain viestiessään projektin kulkuun liittyvistä

asioista. Sen on myös todettu olevan tehokkaampi tapa kuin kirjallinen dokumentointi.

Tutkielmassa on kaksi lähtökohtaa: ketterä ohjelmistokehitys ja hajautettu työ, joiden yhdistämistä on pidetty haasteena. Kun ketterät menetelmät keskittyvät pääasiassa epävirallisiin prosesseihin koordinoinnin helpottamiseksi, hajautettu ohjelmistokehitys riippuu tavallisimmin muodollisesta toimintaperiaatteesta (Ramesh, Cao, Mohan & Xu, 2006, s. 41).

Aiemman kirjallisuuden tapaustutkimuksista selviää, että ketterien käytäntöjen soveltaminen maantieteellisesti hajautettuun työskentelytapaan, on projektin luonteesta riippuen haasteellista. Vaikka ketteriä käytäntöjä on tutkittu jo pitkään, siitä huolimatta monet kansainväliset organisaatiot yrittävät löytää jatkuvasti parempia ratkaisuja tarpeisiinsa viemään projektin onnistuneesti päätökseen.

Tutkielman kirjallisuuskatsauksessa käytettiin tutkimuksen osakysymyksiä seuraavia: "Mitä empiirisesti tiedetään hajautetusta ketterästä ohjelmistokehityksestä?", "Kuinka saavuttaa ketteryys hajautetussa ohjelmistokehityksessä?" ja "Miten ketterän ohjelmistokehitysprojektin onnistuminen määritellään?". Näiden kysymysten pohjalta tutkimusta pystyttiin jatkamaan empiiriseen osuuteen, jossa vastattiin tämän tutkimuksen tutkimuskysymykseen:

Mitkä ovat hajautetun ketterän ohjelmistokehityksen kriittiset menestystekijät ja miten ne suhteutuvat koettuihin haasteisiin?

Tutkimuskysymykseen ja sen osakysymyksiin vastaamiseksi menetelminä käytetään kirjallisuuskatsausta ja laadullista tutkimusta. Tutkimustulosten analysoinnissa käytettiin menetelmänä suunnattua sisällönanalyysia.

Lähdeaineisto koostuu tieteellisistä artikkeleista, kirjalähteistä sekä dokumentoiduista tapaustutkimuksista. Lähdeaineistoa on etsitty Google Scholar -hakukoneella, IEEE Xplore Digital Library -tietokannalla, ACM Digital Library -tietokannalla, Jyväskylän yliopiston verkkokirjastosta (JykDok), tutkimusartikkelien lähdeluetteloista sekä Helsingin yliopiston (Helka) kirjastosta. Hakuosana on käytetty muun muassa "agility", "agile", "software development", "distributed agile software development" ja "agility in distributed development".

Valittua lähdeaineistoa on arvioitu kriittisesti. Arvioinnissa on hyödynnetty julkaisukanavahakua, joka on tieteellisen julkaisutoiminnan laadunarviointia tukeva luokitusjärjestelmä. Julkaisuhaun kautta pystyttiin tarkistamaan muun muassa lehtien ja konferenssien julkaisufoorumi-luokat. Niiden jako on seuraava: 1 eli perustaso, 2 eli johtava taso ja 3 eli korkein taso. Kirjallisuuskatsauksen pyrittiin valitsemaan lähdeaineistoksi korkeimman tason artikkeleita, mutta kuitenkin vähintään perustason artikkeleita. Lisäksi valikoidun lähdeaineiston hyödyllisyyttä on arvioitu tutustumalla sen sisältöön, esimerkiksi lukemalla tieteellisen artikkelin tiivistelmä ja keskeiset käsitteet.

Tutkielma on jäsennelty kahdeksaan lukuun. Kirjallisuuskatsaus sisältää kolme sisältölukua, joista ensimmäisessä eli luvussa 2 määritellään, mitä on ketterä ohjelmistokehitys ja esitellään ketterän ohjelmistokehityksen periaatteet ja



käytetyimmät menetelmät sekä kuinka on siirrytty perinteisestä lähestymistavasta ketterään lähestymistapaan.

Luku 3 käsittelee hajautettua ketterää ohjelmistokehitystä. Luvussa määritellään hajautettu työ, hajautettu ketterä ohjelmistokehitys ja kerrotaan sen haasteet sekä, kuinka ketteryyttä toteutetaan käytännössä hajautetussa ohjelmistokehityksessä. Luku 4 pitää sisällään ketterän ohjelmistokehitysprojektin onnistumisen ja kriittisten menestystekijöiden määritelmän. Luvussa esitellään myös aiemmassa tutkimuksessa tunnistetut ketterän ohjelmistokehityksen menestystekijät.

Empiirinen osuus alkaa luvusta 5, joka käsittelee tutkimusprosessia, tutkimuksen tarkoitusta ja lähestymistavan valintaa sekä tiedonkeruumenetelmää. Luvussa kerrotaan tarkemmin haastattelurungosta, kyselystä, haastateltavien valinnasta, haastattelujen toteutuksesta ja aineiston analysoinnista. Luku päättyy tutkimuksen luotettavuuden määrittelyyn.

Luvussa 6 esitellään tutkimustulokset teemoittain. Luvussa 7 tulkitaan ja pohditaan näitä tutkimustuloksia, arvioidaan tutkimuksen luotettavuutta ja rajoitteita sekä esitetään jatkotutkimusaiheita. Tämä tutkielma päättyy lukuun 8, joka on yhteenveto. Siinä tiivistetään koko tutkielman sisältö. Tutkielman lopussa on lähdeluettelo sekä liitteet 1 ja 2, jotka sisältävät haastattelurungon ja kyselyn.

## 2 KETTERÄ OHJELMISTOKEHITYS

Tässä luvussa määritellään, mitä on ketterä ohjelmistokehitys sekä kuvataan ketterät periaatteet ja käytetyimmät menetelmät. Tämä osio johdattelee oheisen tutkimusraportin seuraavaan päälukuun, joka jatkuu hajautetun ketterän ohjelmistokehityksen esittelyllä.

### 2.1 Ketterän ohjelmistokehityksen määrittely

Aiemmassa kirjallisuudessa on noussut esille useita määrittelyjä ketterälle ohjelmistokehitykselle. Näissä määrittelyissä yhdistyy ketterän ohjelmistokehityksen julistus (engl. Manifesto for Agile Software Development), joka julkaistiin vuonna 2001 (Fitzgerald, Hartnett & Conboy, 2006). Sen jälkeen ketteriin menetelmiin liittyvä tuntemus ja niiden käyttö ohjelmistokehittämisessä ovat kasvaneet nopeasti tietojärjestelmien kehittäjien keskuudessa (Conboy, 2009, s. 329).

Ohjelmistokehitys on muuttunut vuosien aikana prosessien kehittyttyä ketterämmiksi. Estler, Nordio Furia, Meyer ja Schneider (2014) esittävät artikkelissaan ohjelmistokehityksen prosessina, jolla voidaan rakentaa ja hallita kehityksen eri näkökohtia, kuten vaatimusten kartoitusta, järjestelmän suunnittelua, toteutusta, todentamista ja ylläpitoa. Aikaisemmin strukturoidut prosessit, kuten vesiputousmalli, RUP (engl. Rational Unified Process) tai spiraalimalli, määrittelivät ohjelmistosuunnittelua. Tiukasti määritellyt käytännöt, laaja dokumentointi ja yksityiskohtainen suunnittelu ja hallinta ovat olleet osa strukturoitua prosessia. Tämä kaikki muuttui kuitenkin ketterien prosessien suuntaan, jossa korostuu kehittäjien välinen epävirallinen viestintä sekä pienet yhtenäiset kehitystiimit suurten jäseneltyjen kehitysyksiköiden sijaan (Estler, Nordio, Furia, Meyer & Schneider, 2014).

Ketteryys määritellään yleisesti laadullisena tekijänä tai kykynä olla nopeasti liikkuva ja joustava (Lyytinen & Rose, 2006). Conboyn (2009) mukaan ketteryyden lopullinen määritelmä on tietojärjestelmien kehittämismenetelmän jatkuva valmius luoda ja toteuttaa muutoksia nopeasti ja luontaisesti. Menetelmä on muutoksen omaksumista ennakoivasti tai reaktiivisesti ja oppimista muu-

toksesta samalla, kun se myötävaikuttaa koettuun asiakasarvoon (taloudellisuus, laatu ja yksinkertaistuminen) kollektiivisten komponenttiensa ja suhteidensa kautta ympäristöön (Conboy, 2009, s. 340). Seuraavassa taulukossa (TAULUKKO 1) voi nähdä Lean ja Xian (2010) eri lähteistä keräämät määrittelyt ketterälle ohjelmistokehitykselle. Taulukosta voi huomata, ettei ketterälle ohjelmistokehitykselle ole tullut yhtä täysin vakiintunutta määritettä.

TAULUKKO 1 Ketterän ohjelmistokehityksen määrittely (Lea & Xia, 2010)

Käsite	Relevantit määritelmät / käsitteet / ideat
Ketterä ohjelmistokehitys	Ketteryydellä tarkoitetaan yksikön jatkuvaa valmiutta omaksua muutokset nopeasti tai luontaisesti, ennakoivasti tai reaktiivisesti, korkealaatuisten, yksinkertaistettujen taloudellisten komponenttien ja suhteiden avulla ympäristöönsä.
	Ketteryys on kykyä sekä luoda että reagoida muutoksiin voiton saavuttamiseksi myrskyisässä liiketoimintaympäristössä. Se on kykyä tasapainottaa joustavuutta ja vakautta.
	Ketteryys on nopea ja joustava vastaus muutoksiin.
	Ketteryys liittyy sellaisiin läheisiin käsitteisiin, kuten ketterä, joustavuus, nopeus, taipuisuus, eloisuus tai valppaus. Se tarkoittaa perinteisten ohjelmistokehitysmenetelmien raskaiden toimintatapojen poistamista nopean reagoinnin edistämiseksi muuttuviin ympäristöihin ja muutoksiin käyttäjän vaatimuksissa.
	Ketteryydellä tarkoitetaan toiminta- tai muutosvalmiutta. Sillä on kaksi ulottuvuutta: (1) kyky mukautua erilaisiin muutoksiin ja (2) kyky hienosäätää ja muuttaa ohjelmistokehitysprosesseja tarvittaessa.
	Ketteryydellä tarkoitetaan kykyä havaita ja reagoida nopeasti teknisiin muutoksiin ja uusiin liiketoimintamahdollisuuksiin. Se toteutetaan etsintä- ja hyödyntämispohjaisella oppimisella.
	Ketteryys on kevyt, lähes riittävä ja ohjattavissa.
Ketteryys on yksikön jatkuvaa käyttäytymistä tai kykyä, joka osoittaa joustavuutta mukautua odotettuihin tai odottamattomiin muutoksiin nopeasti, noudattaa lyhyintä ajanjaksoa ja käyttää taloudellisia, yksinkertaisia ja laadukkaita instrumentteja dynaamisessa ympäristössä. Ketteryyttä voidaan arvioida joustavuudella, nopeudella, uuden oppimisella ja reaktiivisuudella.	

Abrahamssonin, Salon, Ronkaisen ja Warstan (2017) artikkelissa käsite ”ketteryys” on selitetty laadullisena tekijänä, muutosvalmiutena, nopeutena, aktiivisuutena ja dynaamisuuutena. Ketterien ohjelmistokehitysmenetelmien tarkoituksena on pyrkiä tarjoamaan yritykselle mahdollisuus toimia pienemmän paineen alla ketterästi ja vastaamaan nopeasti muuttuviin vaatimuksiin (Abrahamsson, Salo, Ronkainen & Warsta, 2017, s. 11, 13-14).

Beckin ym. (2001) ketterän ohjelmistokehityksen julistuksen mukaan voimme löytää parempia tapoja tehdä ohjelmistokehitystä, kun toteutamme sen itse

ja autamme muita siinä. Ketterän ohjelmistokehityksen julistuksen mukaan perusarvoja ovat:

- Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja
- Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentointia
- Asiakasyhteistyötä enemmän kuin sopimusneuvotteluita
- Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa (Beck ym., 2001)

Abrahamsson ym. (2017) esittävät ensimmäiseksi painopistealueeksi sitä, että ketterä ohjelmistokehitys korostaa kehittäjien suhdetta ja yhteisöllisyyttä. Nykyään tämä näkyy tiiviissä tiimisuhteissa ja työympäristöjärjestelyissä sekä muissa ryhmähenkeä lisäävissä menettelyissä.

Toinen tärkeä asia on se, että ohjelmistotiimi tavoittelee jatkuvasti tuotettuja, testattuja ja toimivia ohjelmia. Uusia julkaisuja tuotetaan säännöllisin väliajoin, tavallisimmin kahden viikon tai kuukauden välein, joskus myös jopa tunnissa tai päivittäin. Sen vuoksi kehittäjiä kehoitetaan pitämään koodi yksinkertaisena, selkeänä ja teknisesti mahdollisimman kehittyneenä, mikä vähentää dokumentointirasitusta asianmukaiselle tasolle.

Kolmanneksi merkittävä asia on se, että kehittäjien ja asiakkaiden väliselle suhteelle ja yhteistyölle annetaan etusija joustamattomien sopimusten sijasta. Itse neuvotteluprosessi tulisi nähdä pikemminkin keinona saavuttaa ja ylläpitää kannattava suhde. Liiketoiminnan kannalta ketterän kehityksen tarkoituksena on keskittyä liiketoiminnan arvon tuottamiseen heti projektin alkaessa, mikä vähentää sopimukseen liittymättä jättämisen riskiä.

Neljänneksi on otettu huomioon kehitystiimi, joka koostuu sekä ohjelmistokehittäjistä että asiakkaiden edustajista. Heidän tulisi olla hyvin tietoisia, osaavia ja valtuutettuja harkitsemaan kehitysprosessin elinkaaren aikana ilmenviä mahdollisia sopeutumistarpeita. Tämä tarkoittaa sitä, että heidän täytyy olla valmiita tekemään muutoksia, ja että myös olemassa olevat sopimukset tehdään niillä työkaluilla, jotka tukevat ja mahdollistavat näiden parannusten tekemisen (Abrahamsson, Salo, Ronkainen & Warsta, 2017, s. 11, 13-14).

Ketteryyden on nähty liittyvän myös usein sellaisiin käsitteisiin, kuten joustavuus, kätevyys, elävyys ja valppaus (Erickson, Lyytinen & Keng, 2005). Yritys- ja teknologiaympäristöt muuttuvat ennennäkemätöntä vauhtia, minkä vuoksi ohjelmistokehityksen ketteryyden reagoinnista muuttuviin käyttäjän vaatimukseen on tullut entistä kriittisempi tekijä suorituskyvyn kannalta (Lee & Xia, 2010).

Ericksonin ym. (2005) mukaan ketterän ohjelmistokehityksen ytimen tehtävänä on karsia suuri osa "raskaasta" toiminnasta, joka liittyy yleensä perinteisiin ohjelmistokehitysmenetelmiin. Tällä tavoin voidaan edistää nopeaa reagoimista muuttuviin ympäristöihin, muutoksiin käyttäjän vaatimuksissa, nopeutettuja projektin määräaikoja ja vastaavia. Perinteiset menetelmät nähdään liian hitaina, jolloin ne eivät pysty reagoimaan riittävän nopeasti muuttuvaan ympäristöön. Silloin ne eivät ole myöskään riittävän joustavia kaikissa tapauksissa (Erickson, Lyytinen & Keng, 2005).

Leen ja Xian (2010) toteavat artikkelissaan, että ketterät ohjelmistokehitysmenetelmät ovat kehittyneet 1990-luvun puolivälistä lähtien uusina vaihtoehtoisina ratkaisuin perinteisten ”raskaiden” menetelmien kyvyttömyydelle puuttua sellaisiin kestäviin ongelmiin, kuten ajan ja kustannusten ylityksiin tai reagointiin muuttuviin vaatimuksiin. Ketterän kehityksen kannalta ohjelmistokehitysprosessi on dynaamista, kehittyvää ja orgaanista eikä staattista, ennalta määriteltyä ja mekaanista. Tunnetuimpia ketterän toiminnan kehitysmenetelmiä ovat XP (eXtreme Programming), Scrum, DSDM (Dynamic Systems Development Method) ja FDD (Feature-Drive Development) (Lee & Xia, 2010, s. 88-89).

Näitä menetelmiä ovat kehittäneet ja edistäneet ennen kaikkea ammatinharjoittajat ja konsultit. Tutkimusyhteisö on osallistunut niihin vain evoluution varhaisvaiheessa (Conboy, 2009, s. 329). Lean ja Xian (2010) mielestä ketterästä kehityksestä on puuttunut teoreettisia perusteita ja tieteellistä näyttöä. Tämä koskee erityisesti väitettyjä etuja ja keskeisiä periaatteita.

Monet organisaatiot ovat omaksuneet ketterät kehityssuunnitelmat ymmärtämättä selkeästi, kuinka ketteryys-käsite määritellään ja mitataan ja mitä tekijöitä ne voivat hallita vaikuttaakseen ketteryyteen. Miten, miksi ja kuinka ketterä kehitys toimii tai ei toimi - pysyy valitettavasti osittain sokeana pisteenä (Lee & Xia, 2010, s. 88-89).

## 2.2 Ketterän ohjelmistokehityksen periaatteet

Ketterän ohjelmistokehityksen julistuksen takana on Beckin ym. (2001) mukaan 12 periaatetta. Niistä tärkein liittyy asiakkaan saamiseksi tyytyväiseksi toimitamalla hänelle hänen tarpeensa täyttävät ohjelmistoversiot hyvissä ajoin ja säännöllisesti. Julistuksen mukaan on tärkeää vastaanottaa muuttuvat vaatimukset myös myöhemmissä vaiheissa kehityksen aikana. Tämä perustuu ketterien menetelmien hyödyntämiseen muutoksessa asiakkaan kilpailukyvyyn parantamiseksi.

Ketterän ohjelmistokehityksen tarkoituksena on tarjota versioita toimitettavasta ohjelmistosta säännöllisesti. Tavoiteaika on kaksi kertaa kuukaudessa tai kuukauden välein, kuitenkin suosien lyhyempää aikaväliä. Ohjelmistokehittäjien ja liiketoiminnan edustajien tulee tehdä yhteistyötä päivittäin koko kehitysprojektin ajan. Projektin onnistumiseksi on tärkeää löytää motivoituneet yksilöt, joille annetaan sellaiset puitteet ja tuki, jonka he tarvitsevat. On myös luotettava siihen, että kehittäjät saavat tehtyä työnsä.

Tehokkainta ja toimivinta on kasvokkain käytävä keskustelu, jossa korostuu suora viestintä tiedon välittämiseksi kehitystiimille tai tiimin jäsenten kesken. Ketterän ohjelmistokehityksen edistymistä voidaan mitata sillä, kuinka toimivaa ohjelmisto on. Ketterät menetelmät kannustavat kestävään toimintatapaan. Työtahdin ylläpitäminen on tärkeää myös tulevaisuudessa projektin omistajien, kehittäjien ja ohjelmiston käyttäjien osalta. Ketteryyttä edistää tekniikan laadun ja ohjelmiston hyvän rakenteen jatkuva huomiointi.

Yksinkertaisuus on oleellista. Itseorganisoituva projektitiimi tukee parhaiden arkkitehtuurien, vaatimusten ja suunnitelmien syntymistä. Tiimin tehokkuuden aktiivinen seuraaminen on tärkeää, jotta sitä voidaan mukauttaa uudeen toimintaan (Beck, ym., 2001).

Lee ja Xia (2010) osoittavat, että aikaisempi kirjallisuus on pitänyt tiimin autonomiaa ja monimuotoisuutta tärkeinä periaatteina ohjelmistokehityksen ketteryyden parantamiseksi. Ketterä kehitys on pohjimmiltaan ihmiskeskeistä ja tiimin jäsenten osaamisen arvon tunnistamista. Ketterän kehityksen menestykselle on ratkaisevan tärkeää oikeiden ihmisten hankkiminen, sellaisten, joilla on asianmukaiset taidot ja valtuutus päätöksentekoon (Lee & Xia, 2010, s. 90).

## 2.3 Käytetyimmät ketterät menetelmät

Ketterään ohjelmistokehitykseen lukeutuu useita menetelmiä, kuten Scrum, Extreme Programming (XP), Lean, Feature-Driven Development (FDD), Crystal, Dynamic Systems Development Method (DSDM), Agile Modeling ja erilaiset muunnokset (Lee & Xia, 2010; Dingsøyr ym., 2012; Abrahamsson ym., 2017). Abrahamssonin ym. (2017) mukaan osa menetelmistä on keskittynyt enemmän käytänteisiin, kuten XP ja toiset enemmän ohjelmistoprojektien hallintaan, kuten Scrum. VersionOnen (2019) tutkimuksen mukaan Scrum (54%) on edelleen selkeästi yleisimmin käytetty ketterä menetelmä. Tämän lisäksi suosittuja ovat Scrum/XP -hybridi (10%), Scrumban (8%) ja Kanban (5%) (VersionOne, 2019). Tässä osiossa kerrotaan lyhyesti seuraavista menetelmistä: Scrum, Kanban ja XP.

### 2.3.1 Scrum

Scrum on ketterä ohjelmistokehitysprosessi, joka on suunniteltu lisäämään keskittymistä, selkeyttä ja läpinäkyvyyttä ohjelmistoa kehittäville projektitiimeille (Sutherland & Schwaber, 2007, s. 93). Scrum tarjoaa räätälöidyn tavan työskennellä erilaisissa projekteissa, joilla voi olla vaihtuvat vaatimukset (Srivastava, Bhardwaj & Saraswat, 2017).

Scrum-menetelmä sopii pienempiin tiimeihin, mikä tarkoittaa mieluiten alle kymmentä ohjelmistosuunnittelijaa (Abrahamsson ym., 2017, s. 102). Scrumin luoja Ken Schwaberin ja Jeff Sutherlandin mukaan menetelmä koostuu rooleista (engl. The Scrum Team), tapahtumista (engl. Scrum Events) ja tuotoksista (engl. Scrum Artifacts). Scrum on prosessikehitys, jota on käytetty 1990-luvun alusta lähtien monimutkaisten tuotteiden työn hallintaan. Kehitys sisältää erilaisia prosesseja ja tekniikoita (Schwaber & Sutherland, 2017, s. 3).

Scrum-kehitys koostuu tiimeistä ja niihin liittyvistä rooleista, tapahtumista, tuotoksista ja säännöistä. Scrum-säännöistä puhuttaessa tarkoitetaan sitä, että ne sitovat yhteen roolit, tapahtumat ja tuotokset sekä ylläpitävät niiden välisiä suhteita ja vuorovaikutusta. Scrumille olennaisia arvoja ovat sitoutuminen, rohkeus, keskittyminen, avoimuus ja kunnioitus. Scrum-tiimin jäsenet oppivat

tutkimaan näitä arvoja sellaisenaan, kun he työskentelevät Scrum-roolien, tapahtumien ja tuotoksien parissa (Schwaber & Sutherland, 2017, s. 3, 5).

Scrum-tiimiin kuuluu *tuotteen omistaja, kehitystiimi ja Scrum Master*. Scrum-tiimin tarkoitus on olla itseorganisoituvaa, eli he valitsevat itse väylän, joka heille sopii parhaiten työn suorittamiseksi. Tiimin toiminta perustuu jatkuvien tuotteiden toimittamiseen (Schwaber & Sutherland, 2017, s. 6).

Myös tutkijoiden Moen, Dingsøyryn ja Dybån (2010) artikkelissa korostetaan sitä, että itsensä johtaminen on Scrumissa tärkeä ominaisuus. Scrum-tiimille annetaan merkittävä vastuu työnsä monista näkökohdista, kuten suunnittelusta, aikataulusta, tehtävien jakamisesta jäsenille ja päätöksenteosta. Toisin sanoen, tiimille annetaan täydet valtuudet tehdä kaikki, mitä se katsoo tavoitteen saavuttamiseksi välttämättömäksi (Moe, Dingsøyr & Dybå, 2010).

Kehitystiimissä on oleellista, että jokainen jäsen on ymmärtänyt oman roolinsa ja tehtävänsä jokaisessa tuoteversiossa, mitä tuotetaan. Oleellista on, että koko tiimi menee kohti samaa, yhteistä tavoitetta (Rising & Janoff, 2000, s. 30).

Scrum-tapahtumat ovat seremonioita, jotka toteutetaan säännöllisesti ja aikataulutetusti. *Sprintti* on Scrumin sydän. Sprintin kesto on korkeintaan yksi kuukausi. Sen aikana on pyrkimyksenä luoda "valmis" ja käyttökelpoinen tuoteversio kehitettävästä tuotteesta. Sprintti kestää koko kehitystyön ajan, ja uusi sprintti käynnistyy heti edellisen päätyttyä. Sprintit koostuvat sprintin suunnittelusta (engl. Sprint Planning), päiväpalaverista (engl. Daily Scrum), kehitystyöstä (engl. the development work), sprintin katselmoinnista (engl. the Sprint Review) ja sprintin retrospektiivistä (engl. the Sprint Retrospective).

*Sprintin suunnittelun* tulee vastata seuraaviin kysymyksiin: "Mitä tulevasa sprintissä tehdään?" ja "Kuinka tarvittava työ voidaan saavuttaa?". Sprintin suunnittelu tarkoittaa sen aikana suoritettavien töiden suunnittelua. Suunnittelu toteutetaan koko Scrum-tiimin yhteistyöllä (Schwaber & Sutherland, 2017, s. 9-10).

*Päiväpalaveri* on kehitystiimille aikataulutettu tapahtuma, joka kestää 15 minuuttia. Se pidetään jokaisen sprintin päivänä. Päiväpalaverin tarkoitus on se, että kehitystiimi suunnittelee työskentelyään seuraavalle 24 tunnille sekä tarkastelee edistymistä kohti sprintin tavoitetta. Kehitystiimin kommunikointi voi olla kysymyksen esittämistä tai keskustelua. Esitettäviä kysymyksiä voivat olla esimerkiksi "Mitä tein eilen, mikä auttaa kehitystiimiä saavuttamaan sprintin tavoitteen?" tai "Mitä teen tänään kehitystiimin tueksi sprintin tavoitteen saavuttamiseksi?". Scrum Masterin tehtävä on varmistaa, että kehitystiimille on sovittu palaveri, mutta kehitystiimi vastaa itse päivittäisen palaverin suorittamisesta. Päivittäisen palaverin tarkoituksena on lisätä kommunikointia ja parantaa kehitystiimin tietotasoa (Schwaber & Sutherland, 2017, s. 12).

*Sprintin katselmointi* pidetään sprintin lopussa, jolloin tarkistetaan kehitettyä tuoteversiota ja tarvittaessa muokataan tuotteen kehitysjonoa. Katselmoinnin aikana Scrum-tiimi ja sidosryhmät tekevät yhteistyötä sprintin aikana tuotetun kanssa. Sprintin katselmointia voidaan pitää epämuodollisena kokouksena, ei tilannekokouksena, jossa on tarkoitus saada palautetta ja edistää yhteistyötä. Sprintin katselmoinnin kesto on korkeintaan neljä tuntia yhden kuukauden sprinttejä varten. Lyhyemmillä sprinteillä tapahtuma on yleensä lyhyempi. Scrum Masterin tehtävä on tässäkin varmistaa, että kokous toteutetaan, ja osal-

listujat ymmärtävät sen tarkoituksen. Katselmoinnin tuloksena on päivitetty tuotteen kehitysjohto, joka määrittelee seuraavan sprintin todennäköiset kehitysjohtoon osiot (Schwaber & Sutherland, 2017, s. 13).

*Sprintin retrospektiivi* on tilaisuus, jossa Scrum-tiimi voi tarkastella ja arvioida itseään ja luoda suunnitelman parannettavista asioista, jotka otetaan käyttöön seuraavan sprintin aikana. Maksimissaan kolme tuntia kestävä sprintin retrospektiivi toteutetaan katselmoinnin jälkeen ja ennen seuraavaa sprintin suunnittelutapahtumaa. Tässäkin Scrum Masterin tehtävä on varmistaa, että tilaisuus pidetään, ja osallistujat ovat ymmärtäneet sen tarkoituksen (Schwaber & Sutherland, 2017, s. 14).

Scrum-tiimin ja -tapahtumien lisäksi on hyvä ymmärtää, mitä tarkoitetaan Scrum-tuotoksilla. Tuotoksiin kuuluvat tuotteen kehitysjohto (engl. Product Backlog), sprintin kehitysjohto (engl. Sprint Backlog) ja tuoteversio (engl. Increment). Nämä eri tuotokset edustavat työtä tai arvoa, joka tarjoaa läpinäkyvyyttä ja mahdollisuuksia tarkastamiseen ja mukauttamiseen. *Tuotteen kehitysjohto* on luettelo kaikista vaatimuksista, mitä tiedetään tarvittavaan tuotteeseen. Se ei ole kuitenkaan koskaan täydellinen vaan kehittyvä vaatimusten muuttuessa. Kehitysjohtossa luetaan kaikki ominaisuudet, toiminnot, vaatimukset, parannukset ja korjaukset, jotka tulisi toteuttaa tulevissa julkaisuissa. Kehitysjohto on tuotteen omistajan vastuulla (Schwaber & Sutherland, 2017, s. 14-15).

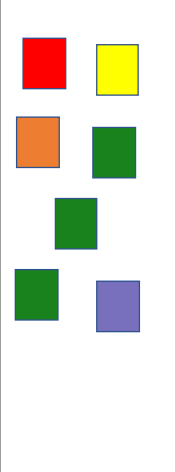
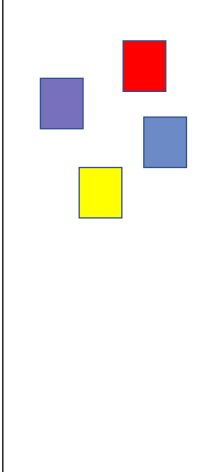
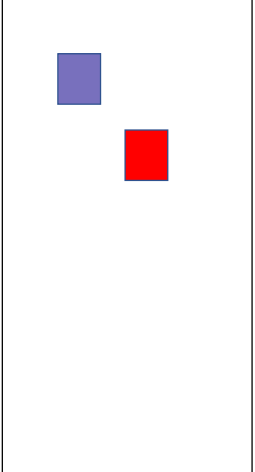
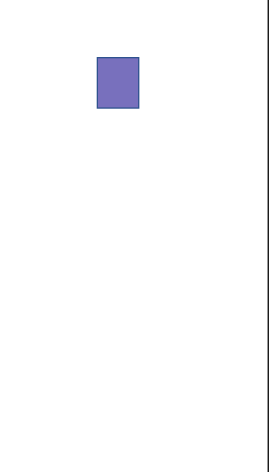
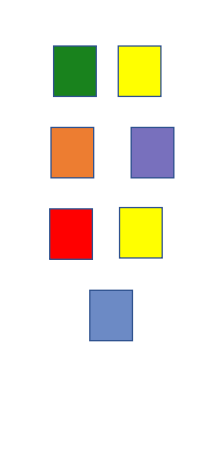
*Sprintin kehitysjohto* muodostuu sprintille valitun tuotteen kehitysjohtoon kohdistuvalle suunnitelman tuotteen tuoteversion toimittamisesta ja sprintin tavoitteen saavuttamisesta. Sprintin kehitysjohto on kehitystiimille ennuste siitä, mitä toiminnallisuuksia tulee olemaan seuraavassa tuoteversiossa ja työstä, joka tarvitaan toimituksen toimittamiseksi (Schwaber & Sutherland, 2017, s. 16).

*Tuoteversio* on kaikkien sprintin aikana suoritettujen tuotteen kehitysjohtoon kohtien summa ja kaikkien aikaisempien sprinttien tuoteversioiden arvo. Tuoteversion tulee olla käyttökelpoinen ja täytettävä Scrum-tiimin määrittelemä ”valmis”. Tuoteversion on oltava käyttökelpoisessa kunnossa riippumatta siitä, päättääkö tuotteen omistaja sen vapauttaa (Schwaber & Sutherland, 2017, s. 17).

### 2.3.2 Kanban

Myös Kanban on yksi käytetyimmistä ketteristä menetelmistä VersionOne (2019) tutkimuksen mukaan. Ikonen, Pirinen, Fagerholmin, Kettusen ja Abrahamssonin artikkelissa todetaan, että Kanban-menetelmä voidaan yhdistää Lean-ajatteluun, mikä on myös sen suosion taustalla. Japanilainen sana Kanban viittaa kylttiin. Se tarkoittaa aikataulutussysteemiä, joka kertoo mitä, milloin ja kuinka paljon tuotetaan. Kanbanin myötä projektiryhmä visualisoi työnkulun, rajoittaa keskenäisiä töitä kussakin työnkulun vaiheessa ja mittaa keskimääräisen ajan yhden tehtävän suorittamiseen (Ikonen, Pirinen, Fagerholm, Kettunen & Abrahamsson, 2011).



TEHTÄVÄT	KESKENERÄINEN (6)	KOODIN TARKISTUS (2)	HYVÄKSYTTY TESTAUS (2)	VALMIS
				

KUVIO 1 Esimerkki Kanban-taulusta (mukaillen Ikonen ym., 2011).

Marathu, Mishra, Singh ja Upadhyay (2015) kertovat artikkelissaan, että monet organisaatiot ympäri maailmaa ottavat Kanbanin käyttöön ja hyödyntävät sitä nykyisessä ohjelmistokehityksessä liiketoiminnan ketteryyden parantamiseksi. Kanban-taulu on visualisointityökalu, joka mahdollistaa työn optimoinnin ja ohjaa työnkulun jakamalla työn luokkiin (tehtävä-, keskeneräiset ja tehdyt työt). (Marathu ym., 2015, s. 3).

Kuten Ikonen ym. (2011) kuvaavat artikkelissaan, Kanban-taulu sisältää projektiin sisältyviä tehtäviä, jotka ovat KUVIOSSA 1 nähtävät värikkäät kortit. Kortit kulkevat tilasta toiseen (vasemmalta oikealle). Tehtäväkorttien dynaaminen siirtäminen osoittaa projektin etenemisen ajan kuluessa. Täten Kanban-taulun avulla nähdään koko projektin tilanne yhdellä silmäyksellä.

Esimerkiksi Kanban-taulusta näkyvät numerot, kuten kohdassa "koodin tarkistus" tarkoittaa sitä, että kyseisessä sarakkeessa saa olla korkeintaan kaksi korttia samanaikaisesti. Muut eivät voi laittaa uutta korttia sarakkeeseen, koska se on jo täynnä. He voivat siinä kohtaan auttaa ongelmallisten korttien vapauttamista. Tällä tavalla pyritään pullonkaulojen vähentämiseen.

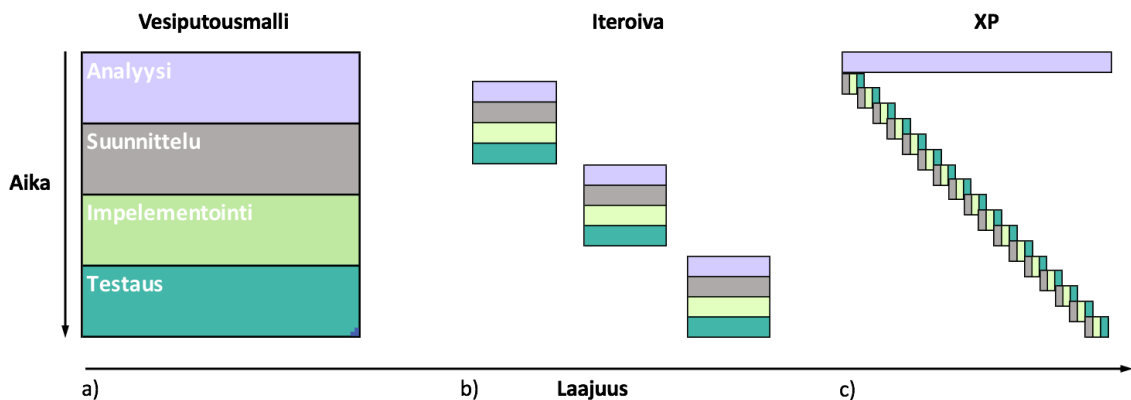
Kanban-menetelmän käyttöön liittyy joukko etuja, kuten parantunut työnkulku, parempi reagointi kysynnän muutoksiin, prosessin visualisoitu aikataulu ja hallinta. Nämä edut edesauttavat tuotantokustannuksien alentamisessa, laadun parantamisessa ja syklin nopeuttamisessa (Ikonen ym., 2011).

### 2.3.3 Extreme Programming (XP)

Extreme Programming (XP) luetaan ketteriin tekniikoihin. Tämä menetelmä perustuu ohjelmistokehittäjien yleiseen käytäntöön rakentaa yhteinen ymmärrys projektista ja jakaa tietoa tiimin jäsenten kesken. (Sadath, Karim & Gill, 2018). Extreme Programming perustuu ohjelmistokehityksen käyttäytymistä ohjaaviin sääntöihin, jotka taas perustuvat yksinkertaisuuden, viestinnän, palautteen ja rohkeuden arvoihin (Lindstrom & Jeffries, 2004, s. 43).

XP -menetelmä on ollut pidemmän aikaa yksi käytetyimmistä menetelmistä ketterässä ohjelmistokehityksessä. Menetelmä on kehittynyt ongelmista, jotka johtuvat perinteisten kehitysmallien pitkistä kehityssykleistä (Abrahamsson ym., 2017).

Beckin (1999) artikkelissa kerrotaan, että Extreme Programming -menetelmä on lähtöisin vesiputousmallista. Sen sijaan, että suunnittelisit ja analysoisit kaukaista tulevaisuutta, XP-menetelmä hyödyntää ohjelmistojen vaihtamiseen liittyen kustannusten alennukset tehdäkseen kaikki nämä toiminnot vähän kerrallaan koko ohjelmistokehityksen ajan. KUVIOSTA 2 voi nähdä, kuinka tapahtui vesiputousmallin ja sen pitkien kehityssyklien (analyysi, suunnittelu, implementointi ja testaus) evoluutio lyhyempiin, iteratiivisiin kehityssykleihin. Lopputulema oli XP, jossa nämä kaikki toiminnot toistuvat lyhyillä sykleillä, ja eteneminen tapahtuu vähän kerrallaan koko ohjelmistokehityksen ajan. Oleellinen ero XP:n ja vesiputousmallin välillä ovat lyhyemmät syklit, vaikka ne sisältävät kaikki samat piirteet (Beck, 1999, s. 70).



KUVIO 2 Vesiputousmallin kehitys Extreme Programming -menetelmäksi (muokailen Beck, 1999, s. 70)

XP on suunniteltu käytettäväksi 10-12 hengen kokoisessa tiimissä (Layman, Williams, Damian & Bures, 2006). Abrahamssonin ym. (2017) artikkelissa todetaan, että XP:ssä on erilaisia rooleja erilaisille tehtäville ja tarkoituksille prosessin ja sen käytäntöjen aikana: ohjelmoija, asiakas, testaaja, mittaaaja, valmentaja, konsultti ja johtaja (engl. Big Boss).

*Ohjelmoijat* kirjoittavat testit ja pitävät ohjelmakoodin mahdollisimman yksinkertaisena. Onnistumisen kannalta tärkeää on kommunikointi ja koordinaatio ohjelmoijien ja tiimin jäsenten kesken.

*Asiakkaan* rooliin kuuluu tarinoiden ja toiminnallisten testien kirjoittaminen. Asiakas päättää, milloin vaatimukset ovat täyttyneet. *Testaajat* ovat asiakkaan tukena toiminnallisten testien kirjoittamisessa. Heidän tehtävänsä on suorittaa toiminnallisia testejä säännöllisesti, lähettää testitulokset ja ylläpitää testaustyökaluja (Abrahamsson ym., 2017, s. 23).

*Mittaaajan* (engl. *tracker*) rooli on seurata tiimin iteraation etenemistä ja arvioida sitä, onko tekeminen pysynyt resurssien ja aikataulun puitteissa. Mittaajan tehtävä on myös antaa palautetta XP:ssä (Abrahamsson ym., 2017, s. 23-24).

*Valmentaja* on henkilö, joka vastaa koko prosessista. Tälle roolille on tärkeää tuntea XP perusteellisesti, jotta tämä voi ohjata muita tiimin jäseniä prosessin etenemisessä. *Konsultti* on ulkopuolinen jäsen, jolla on tarvittava tekninen tietämys. Hänen tehtävänsä on ohjata tiimiä erityisten ongelmien ratkaisemisessa (Abrahamsson ym., 2017, s. 24).

*Johtaja* tekee päätökset. Hänen toiminnalleen tärkeää on kommunikointi projektitiimin kanssa, jotta selviää, mikä on nykytilanne ja mitkä ovat senhetkiset mahdolliset vaikeudet tai puutteet (Abrahamsson ym., 2017, s. 24).

XP käsittää neljä arvoa – *kommunikointi, yksinkertaisuus, palaute ja rohkeus* ja neljä perustoimintoa – *ohjelmointi, testaus, kuuntelu ja virheiden korjaaminen* (Erickson, Lyytinen & Siau, 2005). Nämä arvot ja toiminnot johtavat Beckin (1999) mukaan tärkeisiin käytäntöihin: suunnittelupeli, pienet julkaisut, metafora, yksinkertainen suunnittelu, testaus, refaktorointi, pariohjelmointi, jatkuva integrointi, yhteinen omistajuus, 40-tuntinen työviikko, läsnä olevat asiakkaat, koodausstandardeja ja avoin työtila.

*Suunnittelupelillä* tarkoitetaan sitä, että asiakkaat päättävät julkaisujen laajuuden ja ajoituksen ohjelmoijien toimittamien arvioiden perusteella. *Pienillä julkaisuilla* tarkoitetaan sitä, että uusia julkaisuja julkaistaan usein. Järjestelmä viedään tuotantoon muutamassa kuukaudessa ennen kuin koko ongelma on ratkaistu täysin.

Järjestelmän toiminnan kuvauksessa käytetään asiakkaan ja ohjelmoijien välistä *metaforaa eli vertauskuvaa* tai joukkoa vertauskuvia (Beck, 1999, s. 71; Abrahamsson ym., 2017, s. 26). *Yksinkertainen suunnittelu* voidaan tiivistää seuraavaan sääntöön: ”Sano kaikki kerran ja vain kerran”. Tarkoituksena on suunnitella yksinkertainen mahdollinen ratkaisu, joka on tällä hetkellä toteutettavissa sekä poistaa tarpeeton monimutkaisuus ja ylimääräinen koodi heti (Abrahamsson ym., 2017, s. 26).

*Testauksen* määritelmä voidaan tiivistää seuraavasti: ohjelmistokehitys on testivetoista (Abrahamsson ym., 2017, s. 26). Yksikkötestaukset kirjoitetaan ennen koodia ja niitä suoritetaan jatkuvasti. Asiakkaat kirjoittavat toiminnalliset testit (Beck, 1999, s. 71).

*Refaktoroinnin* tarkoitus on se, että järjestelmän suunnittelua kehitetään poistamalla päällekkäisyydet, parantamalla viestintää, yksinkertaistamalla ja lisäämällä joustavuutta (Abrahamsson ym., 2017, s. 26). *Pariohjelmoinnissa* kaksi ihmistä kirjoittaa kaikki tuotekoodit yhdellä näytöllä, näppäimistöllä ja hiirellä (Beck, 1999, s. 71).

*Jatkuva integrointi* tarkoittaa sitä, että uusi koodi integroidaan nykyiseen järjestelmään heti, kun se on valmis (Abrahamsson ym., 2017, s. 26). Integroinnin aikana järjestelmä rakennetaan tyhjästä, ja kaikkien testien on läpäistävä tai muutokset hylätään (Beck, 1999, s. 71).

*Yhteisen omistajuuden* määritelmä tulee siitä, että jokainen ohjelmoija voi muuttaa koodia missä tahansa vaiheessa, jos he näkevät siihen mahdollisuuden (Beck, 1999, s. 71). *Työviikko on rajattu 40 tuntiin*. Kaksi ylityöviikkoa peräkkäin ei ole sallittua, ja jos näin tapahtuu, siihen on puututtava (Beck, 1999, s. 71).

*Asiakkaan on oltava läsnä* ja käytettävissä kokopäiväisesti tiimille (Beck, 1999, s. 71). Ohjelmoijien on noudatettava *koodausstandardeja* sekä kommunikointia tulisi korostaa koodin kautta (Abrahamsson ym., 2017, s. 27).

Tiimi työskentelee *avoimessa työtilassa*. Huone on suuri, ja siellä on pieniä kaappeja. Pariohjelmoijat työskentelevät tilan keskellä (Beck, 1999, s. 71).

## 2.4 Perinteisestä lähestymistavasta siirtyminen ketterään lähestymistapaan

Estlerin, Nordionin, Furianin, Meuerin ja Schneiderin (2014) mukaan ohjelmistokehitys on prosessi, jolla rakennetaan ja hallitaan kehityksen eri näkökohtia: vaatimusten määrittelyä, suunnittelua, toteutusta, testausta ja ylläpitoa. Ennen ketteriä kehitysprosesseja ohjelmistokehitykseen ovat kohdistuneet strukturoidut prosessit, kuten RUP (engl. Rational Unified Process), vesiputousmalli tai spiraalimalli. Näille strukturoiduille prosesseille on ominaista keskittyminen tiukasti määriteltyihin käytäntöihin, laaja dokumentointi ja yksityiskohtainen suunnittelu ja hallinta (Estler ym., 2014, s. 1199).

Ketterät kehitysprosessit on otettu käyttöön, jotta voitaisiin poistaa joitain strukturoitujen prosessien rajoituksia. Ketterät prosessit, joihin SCRUM tai XP kuuluvat, korostavat kehittäjien välisen tehokkaan epävirallisen viestinnän ja käyttötapausskenaarioiden johtaman toteutuksen iteratiivisen parantamisen tärkeyttä. Tässä lähestymistavassa arvostetaan pieniä yhtenäisiä kehitystiimejä suurten jäseneltyjen yksiköiden sijaan (Estler ym., 2014, s. 1199).

Perinteisen tai ketterien prosessien suhteelliset edut ja soveltuvuus paikallisessa ohjelmistokehityksessä ymmärretään melko hyvin. Tästä esimerkkinä sovellukset, joiden vaatimukset tunnetaan tarkasti ja joihin ei kohdistu radikaaleja muutoksia. Perinteinen kehitys voi tarjota paremman hallittavuuden ja skaalautuvuuden esimerkkitapaukseen. Toisaalta ketterät prosessit toimivat silloin, kun vaatimuksia muutetaan usein, ja muodollisen ja jäseneltyyn viestinnän saavuttaminen sidosryhmien kanssa on vaikeaa tai epärealistista (Estler ym., 2014, s. 1199).

Ketterän prosessin kaikki vaiheet sisältävät kasvokkain tapahtuvan viestinnän asiakkaan kanssa, mikä on todettu tehokkaimmaksi viestintämenetelmäksi (Beck ym., 2001; Estler ym., 2014, s. 1199). Tämä herättää kuitenkin tärkeän kysymyksen siitä, mitkä ovat perinteisten ja ketterien prosessien suhteelliset hyödyt globaalisti hajautetulle ohjelmistokehitykselle, ja onko toinen niistä todennäköisempi vaihtoehto tehokkuuden kannalta. Estlerin ym. (2014) tekemän tutkimuksen mukaan ketterän tai perinteisen prosessin valitseminen ei vaikuta olevan ratkaiseva päätös globaalisti hajautetuissa hankkeissa (Estler ym., 2014, s. 1220-1221).

### 3 HAJAUTETTU KETTERÄ OHJELMISTOKEHITYS

Tässä osiossa kerrotaan, mitä ovat hajautettu työ ja ulkoistaminen sekä määritellään hajautettu ketterä ohjelmistokehitys. Osiossa tuodaan esille myös hajautetun ketterän ohjelmistokehityksen tuomat haasteet sekä, kuinka saavuttaa ketteryyttä hajautetussa ohjelmistokehityksessä.

#### 3.1 Hajautettu työ

Noin 20 vuotta sitten oli jo nähtävillä monien teollisuudenalojen globalisaatiota. Kotlarsky ja Oshri (2005) toteavat artikkelissaan, että sen seurauksena maailmanlaajuisesti jakautuneet yhteistyökuviot ja virtuaalitiimit yleistyivät monilla alueilla, esimerkiksi uusien tietojärjestelmien kehittämisessä. Hajautettujen kehityshankkeiden johtaminen on koettu paljon haastavammaksi kuin hankkeet, jotka tuotetaan yhdessä paikassa. Tieto- ja viestintätekniikka (ICT) on kuitenkin mahdollistanut yhteistyön hajautetussa mallissa.

Globaalisti hajautetut tietojärjestelmäkehitysprojektit ovat hankkeita, jotka koostuvat kahdesta tai useammasta tiimistä. Ne työskentelevät yhdessä hankkeen tavoitteiden saavuttamiseksi eri maantieteellisillä alueilla. Globaalisti hajautetut tiimit kohtaavat aikavyöhykkeitä ja kulttuurieroja, joihin voi sisältyä eri kieliä, kansallisia perinteitä, arvoja ja käyttäytymisnormeja (Kotlarsky & Oshri, 2005).

Hajautettuun työhön liittyy paljon tietojärjestelmien ulkoistamista, mikä on myöhemmin vaikuttanut myös ketterien kehitysprojektien laatuun. Dibbern, Goles, Hirschheim ja Jayatilaka (2004) esittävät artikkelissaan, että alun perin tietojärjestelmäpalvelujen ulkoistaminen koostui ulkoisesta toimittajasta, joka tarjosi asiakkaalle yhden perustoiminnon. Tietojärjestelmien ulkoistaminen alkoi kehittyä 1969. Todellinen kiinnostus ulkoistamiseen syntyi kuitenkin vasta 1980-luvun puolivälissä.

Tietotekniikka ohjaa nykyaikaista organisaatiota. Viimeisten vuosikymmenten aikana yksi laajimmista kehityssuunnista organisaation IT-tarpeiden täyttämässä on ulkoistamisen käytön kasvu (engl. outsourcing). Kun Eastman

Kodak ilmoitti ulkoistavansa tietojärjestelmänsä toiminnan vuonna 1989 IBM:lle, DEC:lle ja Businesslandille, se aiheutti melkoisen myllerryksen tietotekniikka-teollisuudessa. Monissa maissa tunnettujen yritysten johtajat ovat seuranneet Kodakin esimerkkiä.

Ulkoistamisen taustalla on ollut ajatuksena se, että monet yritykset ovat luopuneet monipuolistamisstrategioistaan keskittyäkseen ydinosaamiseensa. Monet ylimmän tason johtajat näkevät tietojärjestelmätieteen toiminnan samantyyppisenä kuin ydinliiketoiminnan, jolloin he uskovat, että tietotekniikan henkilöillä on teknistä asiantuntemusta tarjotakseen tietojärjestelmäpalveluja tehokkaammin kuin sisäiset tietojärjestelmäosastot (Dibbern, Goles, Hirschheim & Jayatilaka, 2004, s. 6-8).

Vielä noin 20 vuotta sitten Carmelin ja Agarwalin (2001) mukaan globaaliin ohjelmistokehitykseen osallistuvien lukumäärä oli pieni, mutta se muuttui nopeasti. Offshore-ohjelmistokehitykseen siirtymiseen on kaksi kriittistä strategista syytä: kustannusetu ja mittava työvoima. (Carmel & Agarwal, 2001, s. 22-23).

Olssonin, Conchúirin, Ågerfalkin, ja Fitzgeraldin (2008) mukaan "offshoring" on saanut jonkin verran uuden merkityksen. Se ymmärretään tehtävien siirtämisenä kehittyviin maihin. Ohjelmistokehitykseen liittyviä tehtäviä, kuten ohjelmointi, ohjelmistojen testaus ja niiden ylläpito, lähetetään ulkoistettaviksi ulkomaille. Ulkoistamisen kohteena ovat usein halvemmän talouden maat, kuten Intia.

Offshore-sijainti voi olla mikä tahansa muu sijainti kotimaan ulkopuolella. Globaalia tietojärjestelmien ulkoistamista kuvataan usein IT-palveluiden siirtämisellä organisaation ulkopuolisille toimittajille. Nykyään monet yritykset ovat kuitenkin globalisoituneet yritysostojen kautta, mikä tarkoittaa muun muassa pienempien ohjelmistoyrityksien ostamista ja niiden integroimista globaaliin toimintaan. Osa yrityksistä on globalisoitunut perustamalla tytäryhtiöitä ja ohjelmistokeskuksia, jolloin asiakasyritys omistaa offshore-keskuksen. Tällaista järjestelyä ei pidetä ulkoistamisena, koska se toteutetaan yrityksen sisällä eikä kolmannen osapuolen suorittamana.

Termejä "outsourcing" ja "offshoring" käytetään usein melkein synonyymeina. Ne tulee pitää kuitenkin erillään toisistaan. (Olsson, Conchúir, Ågerfalk & Fitzgerald, 2008, s. 258-259). Seuraavassa taulukossa (TAULUKKO 2) esitellään käsitteiden erot ja suhde.

TAULUKKO 2 Offshoring vs. outsourcing (Olsson ym., 2008, s. 260)

	<b>Sisäinen (in-house)</b>	<b>Ulkoistettu (outsourced)</b>
<b>Samassa maassa (onshore)</b>	Yrityksen sisäinen tuotanto (perinteinen malli)	Ulkoistaminen alihankkijalle (kolmas osapuoli) samassa maassa
<b>Ulkomaille ulkoistettu (offshore)</b>	Tuotanto on sijoitettu saman yrityksen ulkomaalaiselle osastolle (captive center)	Ulkoistaminen alihankkijalle (kolmas osapuoli) ulkomaille

### 3.2 Hajautetun ketterän ohjelmistokehityksen määrittely

Monien organisaatioiden ja toimialojen globalisaatio on ollut jo nähtävissä pitkään (Holmstrom, Conchúir, Agerfalk, & Fitzgerald, 2006). Yrityksillä on kasvava tarve tutkia globaalia hankintaa, mikä on aiheuttanut sen, että se on johtanut hajautettuihin ohjelmistoprojekteihin (Stankovic, Nikolic, Djordjevic & Cao, 2013).

Paasivaaran ja Lasseniuksen (2014) mukaan alun perin perustettiin iteraatiivisen ja lisääntyvän ohjelmistokehityksen pohjalta ketterät menetelmät tukemaan yhdessä huoneessa työskenteleviä kokoneiden kehittäjien pienryhmiä. Ketterä kehitys on kuitenkin muuttunut vuosien varrella siten, että sitä toteutetaan yhä enemmän suurissa ohjelmistoprojekteissa, joissa on mukana useita maantieteellisesti hajautettuja tiimejä (Paasivaara & Lassenius, 2014).

Globaalisti hajautetun työn määrä on kasvanut, mikä on johtunut osittain kilpailukykyisten resurssivarantojen saatavuudesta kaikkialla maailmassa ja edistyneiden yhteistyöteknologioiden jatkuvasta käyttöönotosta. Esimerkkinä hajautetusta työjärjestelystä ovat juuri globaalit ohjelmistokehitystiimit (Vlaar, van Fenema & Tiwari, 2008, s. 228). Myös yksi tärkeimmistä syistä hajautetun ketterän ohjelmistokehityksen järjestämiseksi on löytää ja hyötyä syrjäisissäkin paikoissa sijaitsevasta korkeatasoisesta tiedosta (Dorairaj, Noble & Malik, 2012).

Alzoubi, Gill ja Al-Ani (2016) esittävät artikkelissaan, että maantieteellisesti hajautettu ketterä kehitys on yhdistelmä hajautetusta kehityksestä ja ketteristä käytännöistä (engl. Geograpy Distributed Agile Development, GDAD). Sen opiskelu ja soveltaminen ovat herättäneet entistä enemmän kiinnostusta. Aiemmasta kirjallisuudesta löytyy julkaisuja maantieteellisesti hajautetun ketterän kehityksen viestinnästä (Alzoubi, Gill & Al-Ani, 2016).

Korkalan ja Abrahamssonin (2007) mukaan hajautetusta ohjelmistokehityksestä (engl. Distributed Software Development, DSD) on tullut vuosien aikana yleinen käytäntö nykyajan ohjelmistoteollisuudessa. Sen merkitys on kiihtynyt eri tekijöiden vuoksi. Maantieteellisesti hajautettu ketterä ohjelmistokehitys mahdollistaa alhaiset tuotantokustannukset, tuotteen saamisen markkinoille nopeammin jatkuvan kehityksen ja markkinoinnin ansiosta, kehittäjien kokoamisen ympäri maailman sekä liiketoimintamarkkinaetujen tuomat hyödyt (Korkala & Abrahamsson, 2007; Alzoubi, Gill & Al-Ani, 2016).

Alzoubin ym. (2016) mukaan hajautettu ketterä ohjelmistokehitys edellyttää tiimin jäsenten työskentelemistä yhdessä projektin tavoitteiden saavuttamiseksi riippumatta maantieteellisestä sijainnista. Hajautetun ohjelmistokehityksen aikana tiimit eivät ole fyysisesti yhdessä (Layman, Williams, Damian & Bures, 2006). Korkalan ja Abrahamssonin (2007) mukaan tiimin jäsenet voivat olla sijoitettuina saman maan eri fyysisiin paikoihin tai globaalisti hajautettuina ympäri maailmaa eri aikavyöhykkeisiin tai maihin.

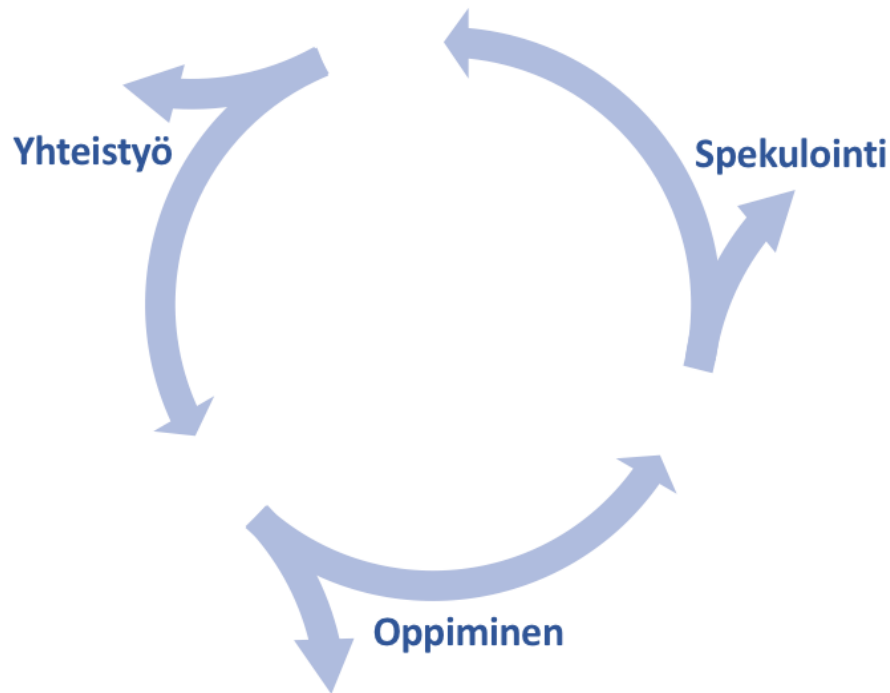
Nämä tarpeet ovat ajaneet ohjelmistokehitystyön kohti monisivustoista globaalisti hajautettua ympäristöä. Ketterät menetelmät perustuvat vaihteleviin vaatimuksiin, joita hallitaan tehokkaalla suullisella viestinnällä. Nämä menetelmät asettavat omat haasteensa hajautetun ohjelmistokehityksen alalle (Korkala & Abrahamsson, 2007).

Abrahamsson ym. (2017) toteavat, että suurin osa hajautetun kehityksen vaikeuksista liittyy sosiaalisiin, kulttuurisiin ja ryhmätaitoihin. Sen vuoksi mukautuva ohjelmistokehitys (engl. Adaptive Software Development) tarjoaa tekniikoita tiimin välisen yhteistyön tehostamiseksi ehdottamalla tiedon jakamisstrategioita, viestintävälineiden käyttöä ja sellaisia tapoja, joita voi ottaa vähitellen käyttöön projektityössä hajautetun kehityksen tukemiseksi (Abrahamsson ym., 2017, s. 77).

Mukautuva ohjelmistokehitys keskittyy pääosin monimutkaisten ja suurten järjestelmien kehittämisessä esiin tuleviin ongelmiin. Tämän menetelmän tarkoitus on kannustaa voimakkaasti asteittaiseen, iteratiiviseen kehitykseen jatkuvien prototyyppien avulla. Mukautuva ohjelmistokehitys on "tasapainoteltua kaaoksen reunalla". Sen tavoitteena on tarjota viitekehys, jonka avulla pyritään estämään hankkeiden joutumista kaaokseen. Samalla pidetään huoli siitä, että tiimit voivat toimia riittävän luovasti ja joustavasti (Abrahamsson ym., 2017, s. 72).

Mukautuva ohjelmistokehitysprojekti toteutetaan kolmivaiheisena jaksona. Jakson vaiheet ovat spekulointi, yhteistyö ja oppiminen. Seuraavassa kuviossa (KUVIO 3) on tuotu ilmi nämä vaiheet.





KUVIO 3 Mukautuvan ohjelmistokehityksen vaiheet (Abrahamsson ym., 2017, s. 73).

Globalisaation seurauksena maailmanlaajuisesti jakautunut yhteistyökuviot ja virtuaalitiimit ovat yleistyneet monilla aloilla (Holmstrom, Conchúir, Agerfalk, & Fitzgerald, 2006). Cadlen ja Yeatesin (2008) mukaan työskentely kansainvälisessä tai virtuaalisessa tiimissä on osin samanlaista kuin työskentely ”tavallisessa” tiimissä, mutta osin myös hyvin erilaista. Taustalla on ollut monien yritysten tarve laajentua globaalisti. Ulkoistaminen ulkomaille on tyypillistä erityisesti IT-alalla. Tämän suuntauksen uskotaan myös jatkuvan tulevaisuudessa (Cadle & Yeates, 2008, s. 383).

Hajautetusti työskentely on tullut laajalti mahdolliseksi ennen kaikkea tietotekniikan ja viestintäteknikan kehityksen myötä. Kun tällaiseen työskentelyyn liittyy kehitystyö, se synnyttää lisähaasteita tiimille. Yhteistyöhön liittyvien ongelmien monimutkaisuuteen on löydetty apua siitä, että yrityksissä pyritään tunnistamaan eri kulttuuritaustoista lähteneiden työntekijöiden potentiaalinen kyvykkyys ja sen tuoma rikkaus. Tietotekniikka mahdollistaa näille hajautetuille ketterille kehitystiimeille kyberavaruuden, jossa eri aikavyöhykkeillä on enemmän merkitystä kuin fyysisellä sijainneilla. Yritystoimintaan liittyviä ongelmia ratkaistaan maailmanlaajuisesti käyttämällä muun muassa jaettuja työtiloja, verkkoympäristöjä, opetusohjelmia ja sähköpostia (Cadle & Yeates, 2008, s. 383-384).

Hajautetussa ketterässä ohjelmistokehityksessä on tärkeää, että tiimit noudattavat käytännöllistä ja yleistä lähestymistapaa. Haasteena on se, että kulttuurierot voivat synnyttää erilaisia työskentelytapoja. Työkulttuuriin liittyvät erot saatetaan nähdä helposti ongelmana, mutta erilaisuus voi tuoda myös

uusia oivalluksia tiimille ja sen toimintatavoille (Cadle & Yeates, 2008, s. 383-384).

### 3.3 Hajautetun ketterän ohjelmistokehityksen tuomat haasteet

Ohjelmistojen kehittäminen ei ole itsessään täydellinen prosessi, vaikka ohjelmistot nähdään tärkeänä yrityksille joka puolella maailmaa (Chow & Cao, 2008). Ketteriä menetelmiä ja erityisesti ketteriä käytäntöjä ei voida suoraan soveltaa hajautettuun ohjelmistokehitykseen, koska ne on suunniteltu lähinnä samassa työtilassa tapahtuvaan ohjelmistokehitykseen (Paasivaara, Durasiewicz, & Lassenius, 2009).

Hajautettu ohjelmistokehitys on tuonut esille omat haasteensa: kulttuuriset ja kielelliset erot, luottamuksen ja sitoutumisen, asynkronisen viestinnän sekä tiedonhallintaan liittyvät kysymykset (Layman ym. 2006). Lisäksi isojen ohjelmistokehitysprojektien tiimien koordinointi, tehokas tiedon jakaminen niiden välillä, suunnittelu ilman tietynlaista arkkitehtuuria tai oikein määriteltyjä vaatimuksia on koettu haasteiksi (Paasivaara & Lassenius, 2014).

Alzoubi ym. (2016) tuovat myös esille artikkelissaan, että tuottavista eduista huolimatta maantieteellisesti hajautettu ketterä ohjelmistokehitys sisältää erilaisia haasteita, kuten viestintä hajautettujen tiimien ja asiakkaan välillä. Se aiheuttaa suuren riskin hajautetulle ketterälle ohjelmistokehitykselle.

Ketteryys eli ketterän kehityksen ydin määrittelee, kuinka tiimin tulisi kommunikoida ja vastata tarpeiden muutoksiin. Aiemmin esitetystä ketteryyden määrittelystä käy ilmi, että ketterien tiimin jäsenten on kommunikoitava tehokkaasti. Jotta ketterän tiimin keskinäinen viestintä olisi suoraa ja tehokasta, ketterät lähestymistavat riippuvat pitkälti kasvokkain tapahtuvasta viestinnästä ja koordinaatiosta samanaikaisesti toimivien tiimin jäsenten ja asiakkaiden välillä. Sitä on kuitenkin vaikeaa toteuttaa maailmanlaajuisesti hajautetussa ketterässä ohjelmistokehityksessä erilaisten haasteiden, kuten viestintärajoitteiden takia. Tällä haasteella viitataan kunkin välineen ominaisuuksiin, jotka voivat vähentää viestinnän tehokkuutta ja vaikuttavuutta (Alzoubi, Gill & Al-Ani, 2016).

Korkalan ja Abrahamssonin (2007) tutkimukset osoittavat, että tehoton ja epäsäännöllinen viestintä vaihtuvien vaatimusten yhteydessä voi aiheuttaa vakaviakin ongelmia, jopa hyvin pienissäkin ketterissä projekteissa (Korkala & Abrahamsson, 2007). Viestinnällisiä haasteita voidaan katsoa Korkalan ja Maurerin (2014) artikkelissa eri lähteistä koottujen näkökulmien kautta. Ovatko maantieteelliset, kulttuuriset ja ajalliset etäisyydet viestinnän ja yhteistyön keskeisinä esteinä? Näiden etäisyyksien yhdistelmä tekee maailmanlaajuisesti hajautuneesta ohjelmistokehityksestä monimutkaista. Ajallinen etäisyys esteenä tarkoittaa synkronisen viestinnän mahdollisuuksien vähenemistä ja epätavanomaisina aikoina tapahtuvaa viestintää. Asynkroniset viestintävälineet puolestaan pidentävät vasteaikoja etätyöskentelyssä, ja vuorovaikutukseen liittyvä median käyttö tehokkaaseen viestintään voi olla erittäin vaikeaa.

Maantieteellinen etäisyys johtaa siihen, että kasvokkain tehtävien tapoimisten järjestäminen on haasteellista, ja puuttuva epävirallinen viestintä estää ideoiden jakamista. Kulttuurinen etäisyys esteenä tuo lisää haasteita, kuten kulttuurieroista johtuvat viestinnän väärinkäsitykset ja tilanteet, joissa erimielisyyksiä tai kielteisiä kysymyksiä ei ilmaista vapaaehtoisesti. Viestintää voivat estää myös kielelliset rajoitukset ja kulttuurillisista työtavoista johtuvat viestintäongelmat (Korkala & Maurer, 2014).

### **3.4 Ketteryyden toteuttaminen käytännössä hajautetussa ohjelmistokehityksessä**

Stankovicin ym. (2013) artikkelissa on tuotu esille aiemmin tehty tutkimus, jonka mukaan ketterien periaatteiden soveltaminen on onnistunut hajautetussa ympäristössä, vaikka näitä kahta lähtökohtaa voidaan pitää ääripäinä. Akateeminen tutkimus on ollut pääosin yritystä ymmärtää tapahtuvaa asiaa. Sitä on tutkittu tieteellisellä tavalla muun muassa tekniikoilla ja menettelytavoilla, jotka ohjelmistokehittäjien ja ketterien harjoittajien yhteisö on jo perustanut ja käyttänyt (Stankovic ym., 2013).

Hajautetun ketterän ohjelmistokehityksen erilaisten haasteiden ja ongelmien ratkaisemiseksi on ehdotettu useita erilaisia tekniikoita. Korkala ja Abrahamsson (2007) tuovat artikkelissaan esille tekniikoita, jotka vaihtelevat erilaisen työkalujen, kuten pikaviestinnän, videoneuvotteluiden ja taulutietokoneohjelmien käytöstä joukkoon yleisimpiä suosituksia. Pelkkä tehokas viestintä ei ole kuitenkaan avain hajautettuun ketterään ohjelmistokehitykseen. Selkeästi määritelty asiakas on tässä kontekstissa keskeinen suositus, mikä vaikuttaa asynkronisten viestintäkanavien käyttömahdollisuuksiin (Korkala & Abrahamsson, 2007).

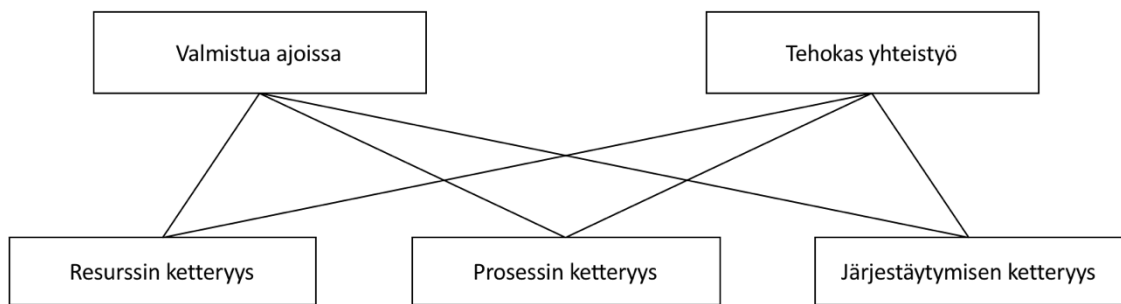
Kuten Ågerfalk, Fitzgerald ja Slaughter (2009) artikkelissaan toteavat, kehittäjille löytyy sopivia ketteriä kehittämismenetelmiä. Nämä kuitenkin joutuvat käytännössä ponnistelemaan menetelmien valinnassa ja siksi niitä räätälöidään omiin tarpeisiin. Ketterät menetelmät täytyy mukauttaa joustavasti käsillä olevaan kehitysympäristöön. Mielenkiintoista tässä on se, että monet vaikeudet, joita hajautetussa ketterässä ohjelmistokehityksessä kohdataan, ovat kuitenkin osittain samoja, joita tulee vastaan ketterissä käytännöissä ja niiden joustavuudessa (Ågerfalk, Fitzgerald & Slaughter, 2009, s.317).

Sarkerin, Munsonin, Sarkerin ja Chakrabortyn (2009) mielestä olisi tarkoituksenmukaista tunnistaa hajautettujen ohjelmistokehitysryhmien keskeiset ketteryyden näkökohdat ottamalla huomioon hallinto- ja tekniset näkökulmat yhdessä. Nämä voivat auttaa organisaatiota jakamaan resursseja tasapainoisella tavalla tiimin ketteryyden parantamiseksi. Ketteryyden saavuttamiseksi on tärkeää, että johtajat huolehtivat ensisijaisesti projektille annettavista asianmukaisista resursseista. Tehdyn tutkimuksen perusteella johtajat pitävät ihmisiin ja teknologiaan perustuvaa ketteryyttä merkittävänä tekijänä. He kokevat myös, että heidän pitää tarjota tiimilleen pääsy sopiviin teknisiin resursseihin, kuten viestintä- ja yhteistyöinfrastruktuuriin.

Ihmisten ja tekniikan tehokas ja joustava hyödyntäminen on johtanut asianmukaisten menetelmien perustamisesta noudattamiseen. Lisäksi myös pätevän henkilöstön saatavuus nopeaan käyttöönottoon missä tahansa tietyssä paikassa, on oleellinen osa ketterää toimintaa niin teknisestä kuin johdon näkökulmasta (Sarker, Munson, Sarker & Chakraborty, 2009).

Sarker ja Sarker (2009) pitävät ketteryyttä yhä enemmän olennaisena tekijänä globaalisti hajautettujen tietojärjestelmien kehittämistiimien tehokkuudessa. On kuitenkin useita syitä, miksi tällaiset tiimit eivät kykene aina kehittämään ja luomaan ketteryyttä muuttuvien tilanteiden käsittelemisessä. Tutkimusten tarkoituksena on antaa syvällisempi tuntemus ketteryudesta, kun tutkitaan intensiivisesti hajautettuja tietojärjestelmien kehittämiskokemuksia organisaatioissa.

Tutkijat osoittivat, että ketteryys pitäisi nähdä monitahoisena käsitteenä, jolla on kolme ulottuvuutta: resurssi, prosessi ja järjestäytyminen. Nämä ovat ketteryuden kannalta tärkeimmät elementit hajautettujen kehitystiimien toiminnan kannalta. Seuraavassa kuviossa (KUVIO 4) on tuotu esille nämä ketteryuden pääulottuvuudet.



KUVIO 4 Ketteryuden kolme ulottuvuutta (Sarker, Munson, Sarker & Chakraborty, 2009).

Seuraavat alakohdat käsittelevät Sarkerin ja Sarkerin (2009) tuomia havaintoja ketteryuden ulottuvuuksien eri elementeistä.

### 3.4.1 Resurssin ketteryys

Resurssin ketteryuden muoto juontaa juurensa hajautetun tietojärjestelmän kehitystiimin käytettävissä olevista ihmisistä ja teknisistä resursseista. Ihmisperusteinen ketteryys perustuu monitaitoisiin ja joustaviin ihmisiin, jotka työskentelevät eri paikoissa. Tärkeä näkökohta on se, että projektitiimillä on kyky suurentaa ja pienentää ryhmän kokoa missä tahansa tietyssä paikassa projektista nousevien vaatimusten mukaan. Tämän resurssin taktiikka sisältää muun muassa huolella valittujen kehityskeskusten sijainnit ja tiimin jäsenten kierron eri paikkojen ja yksiköiden välillä. Tiimin uudelleenkonfiguroitavuus ja hajautettu päätöksenteko mahdollistavat monitaitoisten henkilöiden palkkaamisen, yhteisten oletusten esittämisen kaikissa paikoissa, tiedon epätasapainon minimoimisen eri alueilla ja ennakkotapaamisten ja kiertojen järjestämisen.

Teknologiaperusteinen ketteryys sisältää myös erilaisia tarvittavia komponentteja ja taktiikoita, kuten vertailukelpoisen ja yhteensopivan ICT-infrastruktuurin kussakin paikassa. Päämääränä on muun muassa se, että ylläpidetään standardoitua teknologia-alustaa, joka mahdollistaa tiedon yhdenmukaisen saatavuuden kaikissa paikoissa sekä yhteisten kehitysalustojen ylläpitämisen ja monipuolisen viestintäteknologian soveltamisen erilaisiin tilanteisiin (Sarker & Sarker, 2009, s. 444, 447-448, 455).

### 3.4.2 Prosessin ketteryys

Prosessin ketteryys syntyy tiimin projektia ohjaavasta järjestelmäkehitysmenettelystä, sen toimintaympäristön seurannasta ja järkevistä rutiineista mahdollisten kriisien ennakoimiseksi ja työtavoista, jotka mahdollistavat yhteistyön aikavyöhykkeiden välillä. Prosessin ketteryys voidaan jakaa vielä kolmeen ulottuvuuteen, jotka liittyvät menetelmiin, ajallisiin tekijöihin ja ympäristötietoisuuteen.

Menetelmiin perustuva ketteryys koskettaa perinteisten tietojärjestelmien kehittämismenetelmien ja -käytäntöjen mukauttamisen hajautettuun kontekstiin, kuten perinteisten menetelmien ja ketterien metodologioiden yhdistelmän. Se koskettaa myös ketterien metodologioiden huolellista hallintaa hajautetussa yhteydessä. Tämä tarkoittaa positiivisen kuvan luomista ja kaikkien tiimin jäsenten kouluttamista ketteriin toimintatapoihin sekä niihin liittyviin tekniikoihin, päivittäisten tapaamisten, stressin ja työelämän epätasapainon hallintaa.

Toinen ulottuvuus (ajalliset tekijät) tässä kategoriassa koskettaa kykyä tehdä yhteistyötä eri paikkojen välillä ilman merkittäviä ajallisia viivästyksiä. Siinä hyödynnetään mahdollisuuksien mukaan myös aikavyöhykkeitä. Tällä tarkoitetaan esimerkiksi yötyön tuomaa hyötyä, joka yleensä menetetään, mutta aikavyöhykkeiden ansiosta tuo aika voidaan käyttää tuottavasti maailman toisessa päässä. Se lisää projektin ketteryyttä.

Vaikka aikavyöhykkeiden sanotaan aiheuttavan viivästyksiä vastauksissa, hajautetut tiimit voivat osallistua usein synkronisiin kokouksiin pyrkiessään nopeaan ja välittömään palautteeseen ja siten välttämään väärinkäsitykset. Joissakin tilanteissa työaikojen siirtäminen voi olla vaihtoehto ketteryyden ylläpitämiseksi, mutta se ei ole kuitenkaan pitkällä tähtäimellä kannatettavaa.

Kolmas tämän kategorian ulottuvuuksista (ympäristötietoisuus) liittyy tiimin järjestelmälliseen kykyyn ennakoida, tunnistaa ja reagoida projektissa tapahtuviin muutoksiin. Se tarkoittaa pitkälti asiakas- ja kumppaniorganisaatioiden seuraamista, vaistojen tarkkuutta, tutkitun tiedon nopeaa välitystä eteenpäin ja päätöksentekoa hajautetussa tiimissä. Hankkeeseen liittyvien tärkeiden sidosryhmien kanssa on hyvä rakentaa ja ylläpitää läheistä suhdetta (Sarker & Sarker, 2009, s.448-452, 455).

### 3.4.3 Osapuolten väliseen vuorovaikutukseen liittyvä ketteryys

Järjestäytymisen (engl. linkage) ketteryys syntyy hajautetun tiimin ja hankkeen sidosryhmien vuorovaikutussuhteiden luonteesta. Tällainen luonne koostuu

kulttuurisista ja kommunikatiivisista elementeistä. Kulttuurinen näkökulma pohjautuu siihen, missä määrin tiimin jäsenten kulttuurit kohtaavat ja kietoutuvat saumattomasti yhteen siten, että yhteisen ymmärryksen ja keskinäisten suhteiden kehittymiselle on vain vähän esteitä. Kulttuurinen vastavuoroisuus kuljettaa projektia nopeammin, vaikka haasteena on nähty muun muassa kielten ymmärtäminen. Kieltäytymistä yhteistyöstä tietyissä tilanteissa ei voida hyväksyä, olivatpa syyt mitkä tahansa, ei myöskään turhien lupauksen antamista, vaikka se olisi joissakin kulttuureissa tapana.

Tällaisissa tilanteissa tiimien hitautta voidaan pyrkiä lieventämään avoimuudella. Hajautetun työn lähtötilanteessa on hyvä ymmärtää se, että työskentely tapahtuu eri kulttuureita edustavien henkilöiden kanssa. Olisi virhe olettaa, että työskentely tapahtuisi juuri sillä tavalla kuin on omassa kotimaassaan totuttu. Lisäksi erilaiset kulttuurikompetenssikoulutukset voivat tukea hajautettuja tiimejä.

Kulttuurisen vastavuoroisuusperusteisen ketteryuden toinen elementti liittyy yhteiseen työkuulttuuriin koko organisaatiossa. Se ei ole riippuvainen sijainnista. Yhteisen työkuulttuurin varmistamiseksi ensisijainen periaate on se, että kaikki käyvät läpi samanlaisen koulutuksen. Toinen tärkeä periaate olisi varmistaa se, että tällaiseen hajautettuun tiimiin palkatut työntekijät omaksuvat ammatillisen lähestymistavan vuorovaikutuksessa muiden kulttuurien jäsenten kanssa. Jäsenten on hyvä valmistautua siihen, että työ voi sisältää mahdollisia kulttuuripoikkeavuuksia.

Kyky kommunikoida saumattomasti hajautetun tiimin jäsenten kesken liittyy kommunikaatioyhteyteen, mikä luo perustan ketteryydelle. Hajautetussa työssä erilaiset tietokonevälitteiset viestintävälineet ovat isossa roolissa, mikä mahdollistaa ketteryyden muodon monella eri tavalla. Onnistunut viestintä edistää ketteryyttä, kun se johtaa tiimin jäsenet tuntemaan, että he voivat olla jatkuvasti läsnä toistensa kanssa. Se on tärkeää ketteryyden varmistamiseksi, koska hajautetusta työstä puuttuu kasvotusten työskentely, mikä usein hankaloittaa tehokasta suoraa viestintää, koordinoitua ja yhteenkuuluvuutta.

Osa tiimin jäsenistä voidaan jaotella kahteen eri sijaintiin. Kummassakin päässä on pidettävä toiset ajan tasalla päivittäisillä toimilla. Hajautettu malli toimii parhaiten silloin, kun nämä avainhenkilöt työskentelevät tiiviisti yhdessä. Virtuaalisen "avaruuden" jakaminen mahdollistaa ylläpitämään yhteisymmärryksen.

Toinen kommunikatiiviseen suhteeseen perustuvan ketteryyden komponentti liittyy kansainvälisten kommunikaatiosuhteiden kypsyyteen. Tiimin on helpompi sopeutua vaatimuksissa tapahtuviin muutoksiin, kun se on kypsempi. Kypsyyttä voi kehittää sosiaalisen yhteyden rakentamisessa, tiimin luottamuksen rohkaisemisessa erilaisissa kehittämisharjoituksissa ja sosiaalisten toimintojen avulla.

Hajautetun tiimin jäsenten ja asiakkaiden välisen yhteistyön tulee olla tiivistä ketteryyden varmistamiseksi. Tietoisilla pyrkimyksillä voidaan kehittää yhteistyötä tehokkaammaksi. Saumattomia yhteyksiä asiakkaisiin kyetään parantamaan luomalla avoimia viestintälinjoja (Sarker & Sarker, 2009, s.452-455).

## **4 KETTERÄN OHJELMISTOKEHITYKSEN KRIITTISET MENESTYSTEKIJÄT AIEMMAN TUTKIMUKSEN VALOSSA**

Tässä luvussa määritellään, mitä ovat ketterän ohjelmistokehitysprojektin onnistuminen ja kriittiset menestystekijät tieteellisiä artikkeleita apuna käyttäen. Luvussa esitellään myös aiemmista tutkimuksista esille nousseita ketterän ohjelmistokehityksen hypoteettisia menestystekijöitä sekä kriittisiksi todettuja menestystekijöitä. Aiempien tutkimuksien tuloksia voidaan pitää yleisenä kehikkona, johon tullaan vertaamaan tässä tehtävän tutkimuksen tuloksia. Hajautettu malli on yksittäinen tapa toteuttaa ketterää ohjelmistokehitystä.

### **4.1 Ketterän ohjelmistokehitysprojektin onnistumisen määritelmä**

Misran, Kumarin ja Kumarin (2009) mukaan projektipäällikön näkökulmasta esitetyt aika, kustannukset ja laatu ovat projektin onnistumisen kolme avainkomponenttia. Nämä onnistumisena pidettävät tekijät ovat harkitsemisen arvoisia missä tahansa ohjelmistokehitysprojektissa. Tutkijoiden mukaan ohjelmistokehityksen päätavoitteena on kehittää korkealaatuisia ohjelmistoja, joissa on käytetty vähemmän aikaa ja kustannuksia (Misra, Kumar & Kumar, 2009).

Chow'n ja Caon (2008) mukaan ketterän ohjelmistokehitysprojektin onnistumiseen liittyvät laatu, laajuus, aika ja kustannukset. Laatu perustuu toimivan tuotteen toimittamiseen. Laajuus huomioi asiakkaan kaikkien vaatimusten täyttämisen. Oikea-aikaisuus tarkoittaa toimitusta ajallaan. Kustannukset perustuvat siihen, että toimitus on tehty arvioitujen kustannusten ja vaivan puitteissa (Chow & Cao, 2008).

Misra ym. (2009) määrittelevät onnistuneen ketterän ohjelmistokehitysprojektin käyttämällä seuraavaa viittä kriteeriä:

- Pienemmät toimitusaikataulut

- Lisääntynyt sijoitetun pääoman tuotto (ROI)
- Lisääntynyt kyky vastata nykyisiin asiakkaan vaatimuksiin
- Lisää joustavuutta vastataksaan muuttuviin asiakasvaatimuksiin
- Parannetut liiketoimintaprosessit

Nämä kriteerit ovat muodostaneet tutkimuksen varsinaisen tutkimusmuuttujan ”onnistumisen” komponentit. Ne kuvaavat minkä tahansa projektin onnistumisen kolmea tärkeää ulottuvuutta: lyhyempi kesto, pienemmät kustannukset ja parempi laatu (Misra, Kumar & Kumar, 2009).

## 4.2 Kriittisten menestystekijöiden määrittely

Tieteellisissä artikkeleissa nähdään puhuttavan sekä menestystekijöistä että kriittisistä menestystekijöistä. Chow’n ja Caon (2008) artikkelissa tutkijat kertovat kriittisten menestystekijöiden määrittelyn alkuvaiheista. Rochart (1979) kehitti ensimmäisenä kriittisen menestystekijän (engl. The Critical Success Factor, CSF) lähestymistavan, jonka avulla tunnustetaan ja mitataan organisaation suorituskykyä. Myöhemmin kriittisen menestystekijän määritelmä tarkentui ja se vakiintui.

Kriittinen menestystekijä on määritelty ”rajalliseksi määräksi tekijöitä”, joiden avulla riittävät tulokset takaavat onnistuneen kilpailukyvyn yksilölle, yksikölle tai organisaatiolle. Kriittisillä menestystekijöillä on vain muutama avaintekijä, joiden avulla asioiden on mentävä oikein liiketoiminnan kukoistamiseksi ja johtajien tavoitteiden saavuttamiseksi (Chow & Cao, 2008).

Kriittiset menestystekijät ovat yritysten johtajien työkalu. Stankovic, Nikolic, Djordjevic ja Cao (2013) tuovat esille artikkelissaan lisäksi sen, että kriittinen menestystekijä on menetelmä, jonka avulla johtajat voivat selvittää, mikä tieto on heille parhaiten sopiva, jotta he voivat menestyksellisesti saavuttaa tavoitteensa yhä monimutkaisemmassa maailmassa. Sanotaan, että kaikilla hyvillä johtajilla, useimmiten alitajuisesti, on omat kriittiset menestystekijät, joita he hyödyntävät hallitakseen uran kulkua (Stankovic, Nikolic, Djordjevic & Cao, 2013). Kriittiset menestystekijät on kuvailtu myös kysymyksiksi, jotka asianmukaisella ratkaisulla lisäävät huomattavasti projektin onnistumismahdollisuuksien todennäköisyyttä (Ahimbisibwe, Cavana & Daellenbach, 2015, s. 15).

## 4.3 Ketterän ohjelmistokehityksen menestystekijät

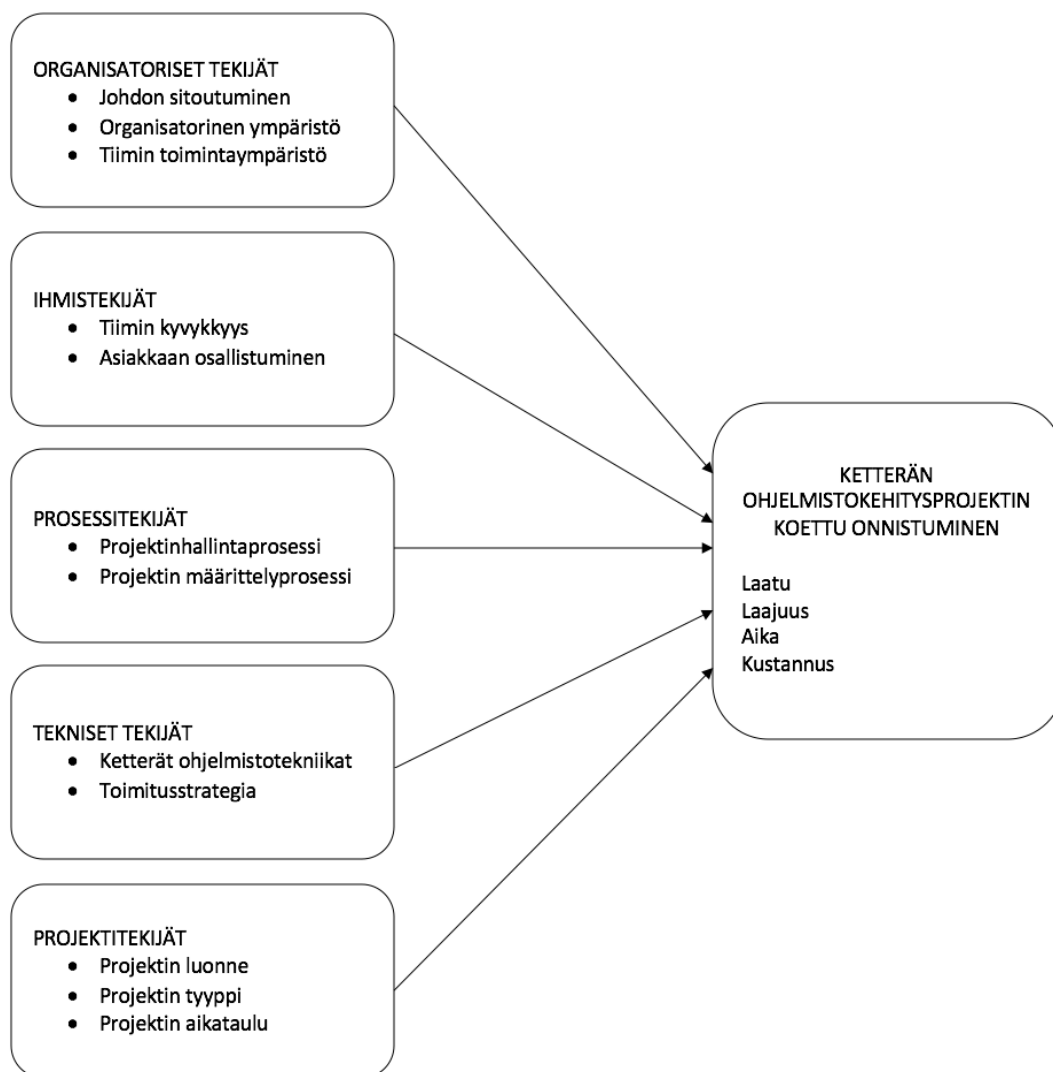
Ketterän ohjelmistokehityksen menestystekijöitä on kuvattu tieteellisissä artikkeleissa erilaisilla viitekehyksillä. Monet tutkimukset ovat keskittyneet enemmän ohjelmistokehitysprojektien tuloksiin kuin itse ohjelmistojen kehittämisprosessiin (Ahimbisibwe, Cavana & Daellenbach, 2015, s. 15).

Chow ja Cao (2008) kertovat artikkelissaan, että ohjelmistoprojektien kriittisten menestystekijöiden havaitaan liittyvän perustavanlaatuisiin projektinhal-



lintatekniikoihin tai liiketoimintastrategian ja ohjelmistosuunnittelun yhdistelmään. Aiemmissä tapaustutkimuksissa ja tutkimusteorioissa käsitellään onnistumista tai epäonnistumista ketterissä toteutuksissa ja ketterissä ohjelmistokehitysprojekteissa. Näiden asioiden tarkastelu on oleellista tunnistettaessa ketterien ohjelmistokehitysprojektien mahdollisia menestystekijöitä. Epäonnistumiset auttavat ymmärtämään, kuinka välttää tietyt sudenkuopat, jotka ovat kriittisiä projektin onnistumiselle (Chow & Cao, 2008).

Chow ja Cao (2008) esittävät tutkimusartikkelissaan viitekehyyksen menestystekijöistä ketterässä ohjelmistokehityksessä. Tämä tutkimus on myös paljon käytetty lähde muissa jatkotutkimuksissa. Kirjallisuudessa esiintyvä menestykseen liittyvä tutkimus perustuu pääosin tapaustutkimuksiin, metatietoihin tai ketterien projektien ja käytäntöjen kokoelmiin ja havaintoihin. Jotkut tutkimukset ovat keskittyneet ketterään toteutukseen suurissa organisaatioissa tai ketterien menetelmien skaalaamiseen suurissa hankkeissa. Tutkijoiden Chow'n ja Caon kokoamat menestystekijät esitellään seuraavassa tutkimusmallissa (KUVIO 5).

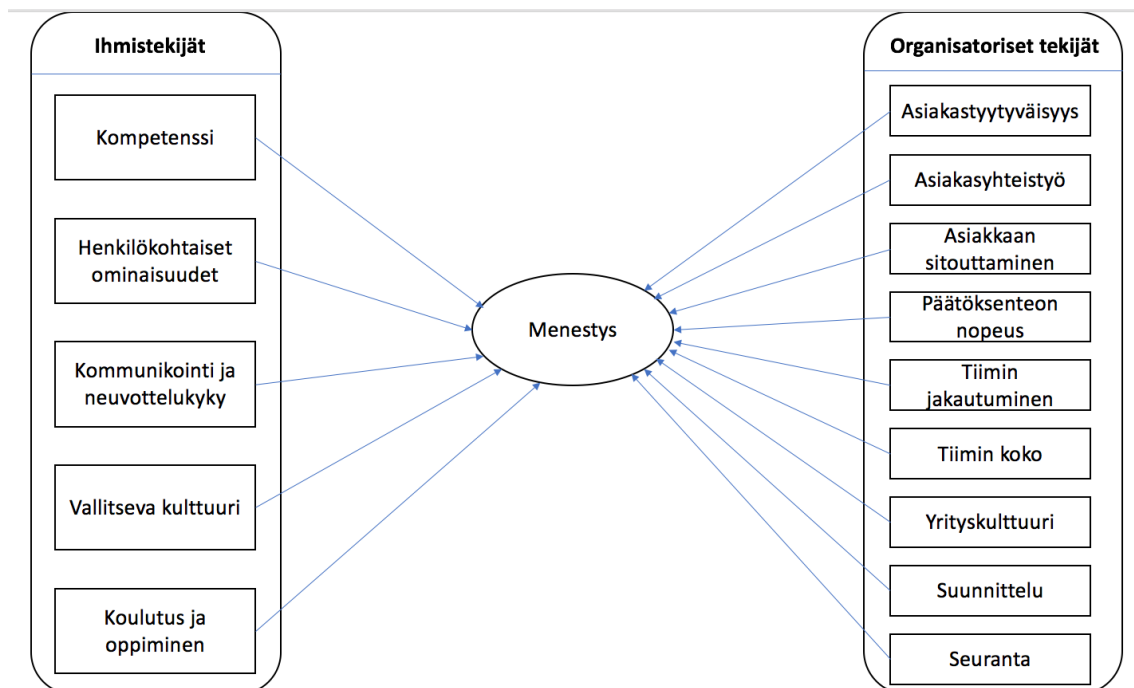


KUVIO 5 Chow'n ja Caon esittämät ketterän ohjelmistokehityksen hypoteettiset menestystekijät (Chow & Cao, 2008).

Kuten kuviosta näkyy, menestystekijät on jaettu eri ulottuvuuksiin: organisatoriset tekijät, ihmistekijät, prosessitekijät, tekniset tekijät ja projektitekijät. Chow'n ja Caon (2008) esittämässä kuviossa on yhdistetty mahdollisia tekijöitä, jotka voivat vaikuttaa ketterän ohjelmistokehitysprojektin onnistumiseen tai epäonnistumiseen. Tässä tehdyssä tutkimuksessa tiedot eivät edusta kaikkia ketterinä pidettyjä menetelmiä.

Tutkimuksessa kahdestatoista tekijästä todettiin kuuden olevan kriittisiä menestystekijöitä. Nämä kriittiset menestystekijät ovat toimitusstrategia, ketterät ohjelmistotekniikat, tiimin kyvykkyys, projektinhallintaprosessi, tiimin toimintaympäristö ja asiakkaan osallistuminen. (Chow & Cao, 2008).

Myös Misran ym. (2009) tutkimusartikkelissa esitetään hypoteettiset menestystekijät, jotka on jaettu seuraavien tekijöiden alle: organisatoriset tekijät ja ihmistekijät. Seuraavassa kuviossa (KUVIO 6) on esitelty hypoteettiset menestystekijät, jotka ovat jakautuneet näiden kahden tekijän alle.



KUVIO 6 Misran ym. esittämät ketterän ohjelmistokehityksen hypoteettiset menestystekijät (Misra ym., 2009)

Tutkimuksessa käytetty tutkimusmenetelmä on tyypillinen johtamistietojärjestelmien kyselypohjaisessa tutkimuksessa. Kolmivaiheisen tutkimuksen ensimmäinen vaihe sisälsi tutkimuskysymyksen muotoilun, hypoteettisen kehyksen rakentamisen ja tutkimuskysymykseen perustuvien hypoteesien kehittämisen. Toisessa vaiheessa tiedonkeruu tapahtui kyselytutkimuksen avulla. Viimeisessä vaiheessa analysoitiin kyselyssä kerätyt tiedot.

Tutkimuksesta selvisi korrelaatioanalyysin perusteella, että seuraavat tekijät liittyvät merkittävästi menestykseen: asiakastyytyväisyys, asiakasyhteistyö, asiakkaan sitouttaminen, päätöksenteon nopeus, yrityskulttuuri, seuranta, henkilökohtaiset ominaisuudet, vallitseva kulttuuri sekä koulutus ja oppiminen.

Pelkästään tutkimustulosten perusteella voitiin havaita, että seuraavat tekijät eivät ole merkittävästi yhteydessä menestykseen: tiimin jakautuminen, tiimin koko, suunnittelu, tekninen pätevyys (kompetenssi) sekä kommunikointi ja neuvottelukyky (Misra ym., 2009).

#### **4.4 Havainnot hajautetun ketterän ohjelmistokehitysprojektin onnistumisesta ja menestystekijöistä**

Hajautetun ketterän ohjelmistokehityksen menestystekijöihin liittyviä tieteellisiä tutkimuksia on vasta vähän. Moni tutkimus on keskittynyt enemmän ketterän ohjelmistokehityksen yleiseen malliin ja sen menestystekijöihin.

Hajautettuja projektitiimejä ja ulkoistamista on paljon nähtävissä globaalisti, koska monet organisaatiot ovat kansainvälisesti toimivia. Se, että hajautettu työ ja ketterät toimintatavat on koettu hyödyllisiksi globaalisti eri organisaatioissa, liittyy paljolti koettuun onnistumiseen ja hyödyllisyyteen liiketoimintaprosessien kehittämisessä. Tutkimusalalta löytyy kuitenkin aukko sille, kuinka voidaan määritellä onnistunut hajautettu ketterä ohjelmistokehitysprojekti ja mitkä ovat ne tekijät, joilla siihen päästään.

Tutkijoiden Shrivastavan ja Rathodin (2015) mukaan organisaatiot yhdistävät ketterän lähestymistavan ja hajautetun ohjelmistokehityksen, jotta projektissa voidaan onnistua lyhyemmässä ajassa ja vähemmällä kustannuksilla. Se auttaa hyödyntämään sekä ketterän että hajautetun ohjelmistokehityksen etuja, mutta sisältää merkittäviä haasteita ja riskejä. Tutkijoiden artikkeli perustuu riskitekijöihin, jotka pitää ottaa huomioon, jotta voidaan taklata sudenkuopat ja onnistua projekteissa (Shrivastava & Rathod, 2015).

Stankovic ym. (2013) tuovat esille artikkelissaan yrityksen kasvavan tarpeen tutkia globaalia hankintaa. Siihen liittyvät hajautetut ohjelmistoprojektit, joissa maantieteellisesti, ajallisesti ja sosiaalis-kulttuurisesti hajaantuneet tiimit kohtaavat lisähaasteita, kun ne yrittävät toteuttaa ketteriä arvoja ja periaatteita onnistuneesti. Aiempaan tutkimukseen perustuen ketteriä periaatteita on mahdollista soveltaa onnistuneesti hajautetussa ympäristössä, vaikka näitä kahta voidaan pitää vastakkaisina ääripäinä (Stankovic ym., 2013).

Koska ketterää ohjelmistokehitystä on tutkittu pitkään, sieltä löytyy viitteitä eri tapoihin toteuttaa ohjelmistokehitystä. Bavanin (2009) mukaan ketterä ohjelmistokehitys hajautetussa ympäristössä ei tarkoita minkään erityisen menetelmän, kuten Scrumin, askel askeleelta toteutusta, jolla on korkeat odotukset oikea-aikaisesta toimituksesta. Se tarkoittaa oikeanlaista yhteistyötä hajautettujen tiimien ja tiimin jäsenten välillä ketterien periaatteiden saavuttamiseksi ja niiden hyväksi toimivan metodologian laatimista.

Monet projektit kärsivät seuratessaan ketterää menetelmää hajautetussa ohjelmistokehityksessä, koska ydintiimin hyväksymä menetelmä ohjaa muuta ryhmää. On kuitenkin osoitettu, että hajautetut ketterät ohjelmistokehitysprojektit voivat myös onnistua, kun tiimi pyrkii määrittelemään itselleen sopivan metodologian.

Tiimin metodologian määrittely tapahtuu avoimen viestinnän ja pienten muutosten avulla, jotta asiat toimisivat odotetusti. Vielä tärkeämpänä pidetään sitä, että menetelmä, joka toimii yhdessä hajautetussa ketterässä ohjelmistokehitysprojektissa, ei välttämättä toimi toisessa. Tämä johtuu siitä, että vaikka periaatteet pysyvät ennallaan, toteutuksen yksityiskohdat vaihtelevat projektien välillä (Bavan, 2009, s. 75). Täten voidaan todeta, että hajautettu ketterä ohjelmistokehitys on enemmän sovellettua kuin paikallisesti toteutettu. Soveltamisella tarkoitetaan tässä sitä, ettei hajautettu kehitystiimi noudata kirjaimellisesti ketteriä menetelmiä tai periaatteita.

## 5 EMPIIRINEN TUTKIMUS

Tässä luvussa käydään läpi empiirisen tutkimuksen prosessia, tarkoitusta ja valittua tutkimusmenetelmää. Sen lisäksi esitellään aineistonkeruumenetelmä, jonka jälkeen tulee haastattelurunko ja kysely. Tässä luvussa kerrotaan myös siitä, kuinka haastateltavat on valittu, miten aineisto analysoitiin ja mitä tarkoitavat reliabiliteetti ja validiteetti laadullisen tutkimuksen kannalta.

### 5.1 Tutkimusprosessi

Tutkimuksen kulku on monivaiheinen. Hirsjärvi, Remes ja Sajavaara (2009) toteavat, että tutkimuksen tekemiseen liittyy viisi askelta: aiheen valinta, tiedon kerääminen, materiaalin arvioiminen, ideoiden, tuloksien ja muistiinpanojen järjestäminen sekä tutkielman kirjoittaminen. Suunnitelmallista ja tavoitteellista tutkimusta voidaan kutsua luovaksi prosessiksi, joka etenee askeleiden myötä. Tutkimuksia on monenlaisia, ja myös tutkijat eroavat toisistaan omalla toiminnallaan.

Jokainen tutkimuksen askel sisältää erilaisia toimia ja tutkijan tekemiä valintoja. Ensimmäinen askel sisältää aihepiirin valinnan, sen rajaamisen, ohjaajan tuen sekä aikataulun suunnittelun. Toisen askeleen aikana tutkija etsii informaatiolähteet, lukee, haastattelee ja kerää aineistoa sekä tekee muistiinpanoja.

Kolmas askel sisältää aineiston kriittisen tarkastelun, materiaalin hyödyllisyyden ja sopivuuden arvioimisen, aineiston karsimisen sekä mahdollisen lisäaineiston keräämisen. Neljännen askeleen aikana tutkija pyrkii järjestämään aineistoa loogisesti sekä analysoi ja tulkitsee sitä. Tämän jälkeen on tuloksien toteaminen. Viimeisen eli viidennen askeleen toimet ovat kirjoittaminen, tekstin muokkaaminen, viimeistely ja tarkistaminen (Hirsjärvi, Remes & Sajavaara, 2009, s. 63-65).

## 5.2 Empiirisen tutkimuksen tarkoitus

Hirsjärven ym. (2009) mukaan tutkimuksella on aina jokin tarkoitus tai tehtävä. Tarkoitus perustuu tutkimusstrategisten valintojen ohjaamiseen. Tutkimuksen tarkoituksen luonnehtimiseen liittyy yleensä neljä eri piirrettä: kartoittava, selittävä, kuvaileva ja ennustava. Tutkimus voi sisältää enemmän kuin yhden tarkoituksen ja se voi muuttua tutkimuksen edetessä. Seuraavassa taulukossa (TAULUKKO 3) on esitetty tutkimuksen tarkoituksen eri piirteitä.

TAULUKKO 3 Tutkimuksen tarkoitus (mukaillen Hirsjärvi ym., 2009)

Tutkimuksen tarkoitus
<p><b>1. Kartoittava</b></p> <ul style="list-style-type: none"> <li>○ Katsoa mitä tapahtuu</li> <li>○ Etsiä uusia näkökulmia, löytää uusia ilmiöitä</li> <li>○ Selvittää vähän tunnettuja ilmiöitä</li> <li>○ Kehittää hypoteeseja</li> </ul>
<p><b>2. Selittävä</b></p> <ul style="list-style-type: none"> <li>○ Etsiä selitystä tilanteeseen tai ongelmaan, tavallisimmin kausaalisten suhteiden muodossa (= syy-seuraussuhteita etsivä)</li> <li>○ Tunnistaa todennäköisiä syy-seurausketjuja</li> </ul>
<p><b>3. Kuvaileva</b></p> <ul style="list-style-type: none"> <li>○ Esittää tarkkoja kuvauksia henkilöistä, tapahtumista tai tilanteista</li> <li>○ Dokumentoida ilmiöistä keskeisiä, kiinnostavia piirteitä</li> </ul>
<p><b>4. Ennustava</b></p> <ul style="list-style-type: none"> <li>○ Ennustaa tapahtumia tai ihmisten toimintoja, jotka ovat seurauksena ilmiöstä</li> </ul>

Tämän tutkimuksen tarkoitus on olla sekä kuvaileva että selittävä. Tutkimuksessa esitetään tarkkoja kuvauksia haasteista, joita hajautettu ketterä tiimi kokee ja pyritään selittämään, mistä ne johtuvat. Hajautetun ketterän ohjelmistokehityksen kriittiset menestystekijät ovat niitä tekijöitä, joihin hajautetun kehitystiimin tulisi keskittyä ja sisällyttää toimintaansa, jotta projekti onnistuisi. Tässä tutkimuksessa pyritään selittämään, millä kriittisillä menestystekijöillä onnistumiseen päästään.

## 5.3 Tutkimuksen lähestymistavan valinta

Tutkimus jaetaan määrälliseen eli kvantitatiiviseen ja laadulliseen eli kvalitatiiviseen tutkimukseen. Hirsjärvi ym. (2009) esittävät kirjassaan, että näistä kahdesta tutkimustavasta on keskusteltu jo pitkään. Vaikka joskus nämä nähdään vastakkaisina lähestymistapoina, niitä on pidettävä toisiaan täydentävinä.

Molempia lähestymistapoja voidaan käyttää rinnakkain. Laadullinen tutkimus voi edeltää määrällistä tutkimusta ja päinvastoin. Kvantitatiiviselle eli määrälliselle tutkimukselle keskeisiä piirteitä ovat muun muassa hypoteesien esittäminen, johtopäätökset aiemmista tutkimuksista, aiemmat teoriat, perusjoukon määrittely ja siitä otettava otos sekä aineiston saattaminen tilastollisesti käsiteltävään muotoon (Hirsjärvi ym., 2009, s. 135-137, 140).

Tässä tutkimuksessa lähestymistavaksi on valittu laadullinen tutkimus. Hirsjärven ym. (2009) mukaan laadullisessa tutkimuksessa ollaan yleensä kiinnostuneita asioista, joita ei voi yksinkertaisella tavalla mitata määrällisesti. Lähtökohtana tällaiselle tutkimukselle on todellisen elämän kuvaaminen. Tutkimuksen tavoitteena on tutkia kohdetta mahdollisimman kokonaisvaltaisesti, mikä tarkoittaa sitä, että tieto kerätään ja kootaan luonnollisissa ja todellisissa tilanteissa. Erilaiset arvot muovaavat sitä, kuinka ymmärrämme tutkimaamme ilmiötä. Sen vuoksi tutkija ei voi sanoutua irti arvolähtökohdista. Tavoitteena on mieluummin löytää tai paljastaa tosiasioita kuin todentaa jo olemassa olevia (totuus) väittämiä.

Tiedon lähteenä suositaan ihmistä. Ihminen tutkijana on riittävän joustava vaihteleviin tilanteisiin. Laadullisen tutkimuksen tiedon hankinnassa on mahdollista käyttää myös lomakkeita tai testejä. (Hirsjärvi ym., 2009, s. 160-161, 164) Tässä tutkimuksessa on hyödynnetty myös kyselyä tietojen keräämiseksi.

Kuten Hirsjärvi ym. (2009) toteavat laadullisessa tutkimuksessa tutkija pyrkii paljastamaan odottamattomia seikkoja, minkä vuoksi käytetään induktiivista analyysia. Tutkimuksen tarkoituksena ei ole teorian ja hypoteesien testaaminen, kuten määrällisessä tutkimuksessa. Pyrkimyksenä on aineiston monitahoinen ja yksityiskohtainen tarkastelu. Sen vuoksi kvalitatiivinen lähestymistapa soveltuu paremmin tässä tehtävään tutkimukseen.

Laadullista tutkimusta on mahdollista toteuttaa erilaisilla metodeilla, joista kerrotaan lisää seuraavassa osiossa. Metodien lähtökohtana on se, että tutkittavien näkökulmat ja ”ääni” pääsevät esille. Laadullisessa tutkimuksessa on tärkeää kohderyhmän tarkoituksenmukainen valinta. Tällöin satunnaisotoksen menetelmä ei ole oleellinen. Tutkimuksen kohteena ovat tavallisimmin ihmiset, ryhmät tai organisaatiot (Benbasat, Goldstein & Mead, 1987, s. 370).

Joustavasti toteutettavan tutkimuksen edetessä myös tutkimussuunnitelma muotoutuu olosuhteiden mukaisesti (Hirsjärvi ym., 2009, s. 164). Tämän pro gradu -tutkielman osalta tutkimus on ollut tutkijalle myös tietynlainen oppimismatka. Teemahaastatteluiden myötä erilaisten lisäkysymysten esittäminen on lisääntynyt kattavan aineiston saamiseksi.

## 5.4 Tiedonkeruumenetelmä

Laadullisessa tutkimuksessa aineistoa voidaan kerätä muun muassa haastatteluilla, havainnoimalla, kyselyillä tai olemassa olevia dokumentteja hyödyntämällä (Darke, Shanks & Broadbent, 1998, s. 275). Kuten Hirsjärvi ym. (2009) toteavat tiedonkeruussa suositaan ihmistä instrumenttina. Sen vuoksi haastattelu on ainutlaatuinen tiedonkeruumenetelmä, koska siinä ollaan tutkittavan kanssa

suorassa kielellisessä vuorovaikutuksessa. Haastattelusta tekee joustavan se, että siinä voidaan säädellä aineiston keruuta tilanteen edellyttämällä tavalla. Vastaaajia voidaan myötäillä ja aiheiden järjestyksen muuttaminen on mahdollista. Tämä auttaa tulkitsemaan enemmän kuin tyyppillinen kysely (Hirsjärvi ym. 2009, s. 204-205).

Tässä tehtävän tutkimuksen tavoitteena oli saada ymmärrys siitä, mitkä ovat hajautetun ketterän ohjelmistokehityksen kriittiset menestystekijät ja miten ne suhteutuvat koettuihin haasteisiin. Haastattelut perustuivat täten haastatteluvien omiin kokemuksiin ja näkemyksiin. Sen vuoksi haastattelun käyttö on perusteltua tässä tutkimuksessa. Tutkija on kiinnostunut toisten ihmisten tarinoista.

Menetelmän valinnassa on tärkeää Hirsjärven ja Hurmen (2008) mukaan seuraavat kriteerit: tehokkuus, taloudellisuus, tarkkuus ja luotettavuus. Haastattelu mahdollistaa vastauksien taustalla olevien motiivien selvittämisen. (Hirsjärvi & Hurme, 2008, s. 34).

Hajautettua ketterää ohjelmistokehitystä ja sen kriittisiä menestystekijöitä on vasta vähän tutkittu. Syynä on mahdollisesti kahden täysin eri lähtökohdan, ketterän kehittämisen yhdistäminen hajautettuun työhön. Aiemman kirjallisuuden tapaustutkimuksista selviää, että ketterien käytäntöjen soveltaminen maantieteellisesti hajautettuun työskentelytapaan, on projektin luonteesta riippuen haasteellista. Moni kansainvälistynyt organisaatio kuitenkin toteuttaa tällaista toimintaa haasteista huolimatta. Hajautetun ketterän ohjelmistoprojektin onnistumisen määrittely ja kriittisten menestystekijöiden tunnistaminen täyttävät tutkimusaukkoa tällä tutkimusalalla.

Haastattelu auttaa selventämään vastauksia ja syventämään saatavia tietoja (Hirsjärvi & Hurme, 2008, s. 35). Jotta tutkimuksesta saataisiin vastaus tutkimusongelmaan, haastattelun aikana on mahdollista pyytää perusteluja esitettyihin kokemuksiin (Hirsjärvi & Hurme, 2008, s.35).

Tutkimuksen kohdeorganisaatio ottaa mittaa myös hajautuksen tuomista haasteista ketterässä ohjelmistokehityksessä. Tämä aihe voi sen vuoksi olla arka ja vaikea. Siitä syystä tutkimus on tehty täysin anonymina.

Yhtenä vaihtoehtoisena menetelmänä olisi voinut ajatella pelkkää kyselylomaketta. Hirsjärven ja Hurmen (2008) mukaan sitä käytetään usein helppouden vuoksi. Se ei olisi kuitenkaan mahdollistanut kuvaavien esimerkkien saamista, mikä tukee tutkittavan ilmiön ymmärtämistä (Hirsjärvi & Hurme, 2008, s. 36).

Myers ja Newman (2007) esittävät artikkelissaan eri haastattelutyyppejä, joita ovat strukturoitu eli lomakehaastattelu, puolistrukturoituhaastattelu, strukturoimaton eli avoin haastattelu ja ryhmähaastattelu. Strukturoidussa haastattelussa on kokonainen käsikirjoitus, joka on valmisteltu etukäteen, eikä improvisoinnille ole tilaa. Se on kyselyn kaltainen. Puolistrukturoitua ja strukturoitua haastattelua pidetään epätäydellisenä käsikirjoituksena, jossa tutkija on saattanut laatia joitakin kysymyksiä etukäteen, mutta improvisointi on tarpeen (Myers & Newman, 2007, s. 4).

Teemahaastattelu rinnastetaan usein puolistrukturoituun haastatteluun. Hirsjärven ym. (2009) mukaan se on lomakehaastattelun ja avoimen haastattelun välimuoto. Tälle haastattelutyypille on ominaista se, että haastattelun aihe-



piiri eli teema-alueet ovat tiedossa, mutta kysymysten tarkka muoto ja järjestys eivät ole niin suunniteltuja (Hirsjärvi ym., 2009, s. 208).

Tässä tehtävään tutkimukseen eri haastattelutyypeistä teemahaastattelu soveltui parhaiten tutkimusongelman ratkaisemiseksi. Ennalta määritellyt teema-alueet tukivat tutkijaa pysymään tutkittavassa aiheessa ja täten vastaamaan tutkimusongelmaan. Strukturoitu haastattelu olisi voinut johdatella liikaa haastateltavaa, eikä tällöin olisi saatu esille omia kokemuksia, perusteltuja mielipiteitä ja esimerkkejä. Tämä haastattelumuoto olisi saattanut jättää ulkopuolelle niitä asioita, jotka ovat tutkimuksen kannalta tärkeitä. Myös täysin avoin haastattelu olisi voinut koitua suureksi haasteeksi, koska fokus aiheeseen olisi voinut karata haastattelujen aikana.

Teemahaastattelun avulla pystyttiin keskittymään syvemmälle koettuihin haasteisiin, joita nousi pitkin haastattelua esille. Myöskin kriittisten menestystekijöiden selvittämisessä saatiin perusteluja ja samalla oli mahdollista kyseenalaistaa niitä suhteessa koettuihin haasteisiin.

Viimeisen teeman kohdalla jokaisen haastateltavan kanssa arvioitiin lisäksi kirjallisuuden esittämiä menestystekijöitä kyselyn avulla. Haastateltava täytti haastattelun aikana paperisen kyselylomakkeen, jota käytiin samalla läpi keskustelumaisesti perustelujen saamiseksi. Kyselyn käyttö koettiin tärkeäksi, sillä sen avulla pyrittiin löytämään vielä lisää uusia kriittisiä menestystekijöitä. Lisäksi kyselyn käyttö oli perusteltua, koska sen tuomia tuloksia pystyi vertaamaan myös teemahaastatteluiden tuloksiin. Vertailussa huomioitiin muun muassa sitä, nousiko kyselyn tuloksista samoja kriittisiä menestystekijöitä kuin haastatteluissa. Kysely koettiin myös hyväksi, koska sen avulla tunnistettiin sellaisia kriittisiä menestystekijöitä, joita haastateltava ei osannut sanoa haastatteluhetkellä.

Kyselyssä käytettiin vastausvaihtoehtoina *täysin eri mieltä, jokseenkin eri mieltä, ei samaa eikä eri mieltä, jokseenkin samaa mieltä, täysin samaa mieltä sekä ei sovelleta / en tiedä*. Haastateltavilla oli tässä kohtaa niin sanottu ”vapaa sana”. Jokainen kirjallisuuden esittämä kriittinen menestystekijä avattiin tarkemmin haastateltavalle. Näin vältettiin myös mahdolliset väärinymmärrykset.

Hirsjärven ym. (2009) mukaan haastattelu voidaan toteuttaa yksilöhaastatteluna, parihaastatteluna tai ryhmähaastatteluna. Yksilöhaastattelut ovat tavallisimpia. Parihaastattelu on yksi ryhmähaastattelun muodoista. Ryhmähaastattelua pidetään tehokkaana tiedonkeruumuotona, koska silloin tietoja kootaan usealta henkilöltä yhtä aikaa (Hirsjärvi, 2009, s. 210). Eri vaihtoehdoista teemahaastattelut pidettiin yksilöhaastatteluina, koska se mahdollisti keskusteluiden pitämisen vapautuneina, luontevina ja anonyymeina.

## 5.5 Haastattelurunko ja kysely

Haastatteluiden tavoitteena oli se, että haastateltavat pystyisivät kertomaan mahdollisimman vapaamuotoisesti kokemuksistaan ja näkemyksistään hajautetun ketterän ohjelmistokehityksen suhteen. Hirsjärven ja Hurmen (2008) mukaan haastattelurunkoa suunniteltaessa ei ole niin tärkeää yksityiskohtainen

kysymysluettelo vaan enemmänkin teema-alueuettelo. Haastattelukysymyksien on tarkoitus kohdistua näihin eri teema-alueisiin. Teema-alueet ovat haastattelijan muistilista. Ne ovat keskustelun kannalta tärkeitä ohjaavia kiintopisteitä. Haastateltava voi jatkaa ja syventää keskustelua niin pitkälle kuin tutkimusintressit edellyttävät teema-alueiden pohjalta (Hirsjärvi & Hurme, 2008, s. 66-67).

Tutkielman teemahaastattelunrunko (LIITE 1) muodostui viidestä alueesta:

- Taustatiedot
- Hajautettu ketterä ohjelmistokehitys
- Koetut haasteet
- Onnistumisen määrittely
- Kriittiset menestystekijät

Ensimmäinen eli taustatiedot-osio keskittyi selvittämään haastateltavan kokemusta ja taustaa ketterästä ohjelmistokehityksestä (projektien määrä, kokemus vuosissa jne.). Lisäksi haastateltavalta kysyttiin, millaisissa tiimeissä hän on työskennellyt ja millaisia vastuita hänellä on ollut. Toinen osio keskittyi hajautettuun ketterään ohjelmistokehitykseen. Haastatteluiden avulla selvitettiin, kuinka sitä toteutetaan kohdeorganisaatiossa, mitkä ovat odotukset hajautetulle ketterälle tiimille ja miten he kommunikoivat.

Kolmas osio liittyi koettuihin haasteisiin, joita hajautus aiheuttaa ketterälle ohjelmistokehitykselle. Neljäs osio käsitti hajautetun ketterän ohjelmistokehitysprojektin onnistumista, ja kuinka sitä mitataan. Viimeisessä ja eli viidennessä osiossa keskityttiin kriittisiin menestystekijöihin, eli mitkä ovat ne tekijät, joihin hajautetun tiimin tulee keskittyä ja sisällyttää toimintaansa, jotta projektissa onnistutaan. Haastateltavan annettiin itse kertoa omat näkemyksensä, minkä jälkeen käytiin läpi vielä tieteellisessä kirjallisuudessa esitettyjä menestystekijöitä kyselyn avulla. Kyselyssä esitetyt menestystekijät arvioitiin asteikolla 1-5.

## 5.6 Haastateltavien valinta ja haastattelujen toteutus

Haastateltavien valinta vaatii suunnittelua. Kuten Hirsjärvi ja Hurme (2008) toteavat, tutkijan tulisi haastatella niin monta henkilöä kuin on välttämätöntä, jotta tämä saa tarvitsemansa tiedon. Jos valitaan vain pieni määrä havaintoyksiköitä, niin määrällinen tutkimus ei ole tarpeeksi pätevä valinta tutkimusmenetelmäksi. Jos haastateltavia on kuitenkin liikaa, aineistosta voi olla vaikeaa tehdä syvällisiä tulkintoja (Hirsjärvi & Hurme, 2008, s. 58). Aineiston riittävyttä voidaan arvioida saturaatiopisteen avulla. Se tarkoittaa sitä, että aineistoa voidaan ajatella olevan tarpeeksi, jolloin uudet haastattelut eivät tuo olennaista lisäarvoa tutkimukselle (Saaranen-Kauppinen & Puusniekka, 2006).

Tätä tutkielmaa varten tehtävässä tutkimuksessa kaikki haastateltavat valittiin organisaatiosta, joka toimii monikansallisten ohjelmistoprojektien parissa. Haastateltaviksi etsittiin henkilöitä, joilla on kokemusta hajautetusta ketterästä

ohjelmistokehityksestä. Tutkimuksen tavoitteena oli saada 5-9 haastattelua riittävän aineiston keräämiseksi. Lopullinen haastateltavien määrä oli kahdeksan.

Kaikki haastattelut toteutettiin yksilohaastatteluina kasvotusten ja ne nauhoitettiin litterointia varten. Haastattelujen kestot vaihtelivat noin 42 minuutista noin 71 minuuttiin (1h 11min). Keskimääräinen kesto haastatteluille oli siis noin 55 minuuttia.

Haastattelujen alussa käytiin läpi vielä tutkimuksen aihe, tavoitteet sekä käsiteltävät teemat. Lisäksi haastateltavalle varmistettiin se, että haastattelut pidetään täysin anonyymeina. Haastattelujen kulku vaihteli teemahaastattelun joustavuuden vuoksi. Haastateltavalle annettiin tilaa kertoa kokemuksistaan sekä haastateltava esitti lisäkysymyksiä syvemmän tiedon keräämiseksi.

Haastattelut pidettiin suljetussa rauhallisessa tilassa ilman häiriötekijöitä. Haastateltavat olivat haastateltavalle ennestään tuntemattomia henkilöitä, kuten tutkijat Myers ja Newman (2007) suosittelevat artikkelissaan. Tällöin tilanne on aito eikä vääristelty.

Laadullisen haastatteluprosessin yksi sudenkuopista on kielen moniselitteisyys. Se tarkoittaa sitä, että kysymysten esittäminen ja vastausten saaminen on paljon vaikeampi tehtävä kuin miltä se aluksi saattaa vaikuttaa. Puhuttaessa sanoissa on aina epäselvyyttä riippumatta siitä, kuinka huolellisesti sanomme kysymykset. Tämä korostuu usein, jos haastattelua tehdään toisen osapuolen vieraalla kielellä (Myers & Newman, 2007, s. 4-5). Tämän sudenkuopan välttämiseksi haastattelukysymysten esittämistä suunniteltiin ja harjoiteltiin sekä haastattelut pidettiin haastateltavalle ja haastattelijalle yhteisellä äidinkielellä.

## 5.7 Aineiston analysointi

Hirsjärven ja Hurmen (2008) ohjeiden mukaisesti teemahaastattelussa kerätty aineisto on yleensä runsas. Kuitenkaan kaikkea materiaalia ei ole välttämättä tarpeen analysoida, eikä tutkija pysty välttämättä kaikkea materiaalia edes hyödyntämään. Laadullisen aineiston analyysi aloitetaan jo aineistoa kerättyäessä, kuten tämän tutkielman tehdyssä tutkimuksessa (Hirsjärvi & Hurme, 2008, s. 135). Analyysin teko jatkui aineiston purkamisessa eli litteroinnin aikana.

Aineiston litterointiin ei ole Hirsjärven ja Hurmen (2008) mukaan yksiselitteistä ohjetta. Se, kuinka tarkasti litterointi tulee tehdä, riippuu tutkimustehtävästä ja tutkimusotteesta. Yhden tunnin kestävän nauhoitetun haastattelun litteroinnin on arvioitu kestävän 4-6 tuntia.

Aineiston luokittelu ja sen koodaaminen muodostettuihin luokkiin ei ole analyysin lopullinen tavoite, vaan pikemminkin ne ovat välivaiheita analyysin rakentamisessa. Aineiston yhdistely tarkoittaa sitä, että tutkija pyrkii löytämään joitakin säännönmukaisuuksia tai samankaltaisuuksia luokkien välillä. Tarkastelussa löydetään usein myös säännönmukaista vaihtelua ja muista poikkeavia tapauksia.

Eri luokkien yhdistämisessä on erilaisia prosessoinnin vaiheita. Ensimmäinen vaihe liittyy lähtöaineistoon, joka on yleensä muokkaamaton, esimerkiksi tietokoneelle litteroituna. Toisen vaiheen aikana aineisto on luokitettu, eli

lähtöaineisto on induktiivisesti muokattu ja ryhmitelty luokiksi. Tällaista aineiston käsittelyä pidetään joskus riittävänä (Hirsjärvi & Hurme, 2008).

Tässä tehdyssä tutkimuksessa litteroinnin jälkeen tietoa analysoitiin pääosin käyttämällä suunnattua sisällönanalyysia. Hsiehin ja Shannonin (2005) mukaan suunnattu sisällönanalyysi perustuu teorioihin ja aiempiin tutkimustuloksiin, ja ne ohjaavat koodien määrittämistä. Koodit määritetään ennen tietojen analysointia ja niiden aikana. Tämän lähestymistavan tavoitteena on validoida tai laajentaa käsitteellisesti teoreettista kehystä tai teoriaa (Hsieh & Shannon, 2005, s. 1281, 1286).

Suunnatun sisällönanalyysin pääasiallinen vahvuus on se, että olemassa olevaa teoriaa voidaan tukea ja laajentaa. Tässä tutkimuksessa muodostettiin aiempaan teoriaan perustuen avainteemat, joita Hsieh ja Shannonin (2005) mukaan kutsutaan myös koodeiksi eli aihealueiksi. Teemahaastattelussa käytetty haastattelurunko perustui avaintemoihin, jotka muodostettiin hyödyntämällä aiempaa teoriaa. Tämä johdatteli myöhemmin koodien muodostumiseen aineiston analysoinnissa.

Tutkimusaineiston analysoinnissa käytettiin Microsoft Exceliä. Litteroidut haastattelut jaoteltiin eri teemoihin, joissa hyödynnettiin myös värikoodausta selkeyttämään yhtäläisyyksiä ja eroavaisuuksia. Esimerkiksi kriittisten menestystekijöiden tunnistamisessa toimittiin seuraavasti: ensimmäinen vaakarivi vastasi ensimmäistä haastattelua "H1", toinen vaakarivi toista haastattelua "H2" jne. Ensimmäiselle pystyriville otsikoitiin ensimmäinen esille tullut kriittinen menestystekijä ja siihen liittyvä haastateltavan suora lainaus sen osoittamalle vaakariville. Haastattelussa mainitut kaikki kriittiset menestystekijät otsikoitiin pystyriveille vasemmalta oikealle.

Analysointia jatkettiin samalla tavalla jokaisen haastateltavan kohdalla. Kun joku haastateltavista oli nostanut esille saman kriittisen menestystekijän kuin aiemmissa haastattelussa oli käynyt ilmi, suora lainaus kirjattiin sille kuuluvalla pystyriville oikean haastateltavan kohdalle. Tämän avulla pyrittiin löytämään samankaltaisuuksia ja eroavaisuuksia vastauksissa sekä mahdollisia uusia teemoja.

Tiettyjen teemojen alle muodostui alateemoja, jotka pystyttiin yhdistämään pääteemaan. Näiden teemojen perusteella alettiin analysoida ja selvittää eri näkemysten välisiä yhtäläisyyksiä ja eroja.

Hsiehin ja Shannonin (2005) mukaan tutkimustulokset voivat antaa joko tukea tai eriäviä näkemyksiä teorialle ja aiemmille tutkimuslöydöksille. Tämän tutkimuksen pohjalta tunnistettiin kriittisiä menestystekijöitä, jotka ovat linjassa aiemman tutkimuksen kanssa. Sen lisäksi tässä tutkimuksessa löydettiin lisää kriittisiä menestystekijöitä aiempien tutkimustuloksien rinnalle sekä tunnistettiin sellaisia kriittisiä menestystekijöitä, jotka ovat ristiriidassa aiempien tutkimustuloksien kanssa.

Aineistonanalyysin jälkeen tutkimustulokset kirjoitettiin puhtaaksi. Tuloksia vertailtiin aiempaan tutkimukseen. Tämän tutkimuksen tarkoituksena oli antaa lisää tietoa kohdeorganisaatiolle siitä, mitkä ovat koetut haasteet hajaautuksessa ketterässä ohjelmistokehityksessä, sekä mitkä ovat ne kriittiset menestystekijät, jotka ketterän tiimin tulisi sisällyttää toimintaansa onnistuakseen hajautetuissa ohjelmistokehitysprojeekteissa.

## 6 TUTKIMUSTULOKSET

Tässä luvussa esitellään empiirisen tutkimuksen tulokset valittujen teemojen kautta, jotka ilmenevät otsikoista. Ensimmäiseksi esitellään haastateltavien taustatiedot ja kokemuksia ketterästä ohjelmistokehityksestä. Sen jälkeen esitellään, kuinka hajautettua ketterää ohjelmistokehitystä on toteutettu kohdeorganisaatiossa. Kolmanneksi esitellään koetut haasteet ja neljänneksi haastateltavien määrittelemä hajautetun ketterän ohjelmistokehitysprojektin onnistuminen. Lopuksi käydään läpi tunnistetut kriittiset menestystekijät sekä kyselyn tulokset.

### 6.1 Haastateltavien taustatiedot

Tässä tutkimuksessa haastateltavia oli yhteensä kahdeksan ja he kaikki työskentelevät monikansallisessa ohjelmistoyrityksessä. Haastattelujen alussa kartoitettiin haastateltavien taustaa ja kokemusta ketterästä ohjelmistokehityksestä. Haastateltaviin kuului Scrum Mastereita, softa- ja IT-kehittäjiä, Data Scientist, Release Train Engineer (RTE) ja projektipäällikköjä. Monet näistä tehtävämikkeistä on yleisesti käytettäviä rooleja Scrum- tai SaFe-viitekehityksessä. Haastateltaviksi valikoitui satunnaisesti eniten Scrum Mastereita ja softa- / IT-kehittäjiä.

Haastateltavien kokemus ketterästä ohjelmistokehityksestä vaihteli 1,5 vuodesta 11 vuoteen. Pelkkää ohjelmistokehitystä oli kertynyt parhaimmillaan 20 vuoden takaa. Osalla haastateltavista oli myös kokemusta vesiputousmallista ennen ketterää kehittämistä. Erinäisiä hajautettuja ohjelmistokehitysprojekteja oli ehtinyt kertyä haastateltaville 2-10.

Haastateltavien työkokemukseen on kuulunut erilaisia työvastuita, joita olivat projektipäällikkö, IT-analyst (tiimin jäsen), Scrum Master, valmentava Scrum Master, menetelmäkehittäjä, linjaesimies, IT-kehittäjä, Business Analyst, ohjelmistokehittäjä, ylitason arkkitehtuuri suunnittelija, tuoteomistaja (engl. Product Owner), Agile Coach ja konsultti. Osaa työvastuista oli vaikeaa kuvaila käsitteenä, kuten eräs haastateltava totesi.

”Siinä on vähän tommoista kolmea roolia. On semmoista PO:ta, draivaajaa ja sit tämmöistä ylätason arkkitehtuuri duunia.”

Yleisimmin käytetyt ketterät menetelmät olivat Scrum ja Kanban sekä SAFe organisaatiotasolla. Näitä menetelmiä moni haastateltava on noudattanut by the book -periaatteella eli ohjeiden mukaisesti. Muita kokeiltuja ja sovellettuja menetelmiä olivat myös Scrumban, Lean ja LeSS. Lisäksi osa haastateltavista on myös toteuttanut ketterää ohjelmistokehitystä noudattamalla ketteriä periaatteita ilman tarkemmin asetettua menetelmää ja tarkempia rooleja.

Haastateltavien hajautettujen tiimien koot vaihtelivat. Tiimien koot olivat 5-10 henkilöä. Haastattelussa nousi esille myös pari hyvinkin isoa tiimiä, joihin kuului noin 30 henkilöä per tiimi. Tiimit oli hajautettu eri tavoin. Yleisin jakautuminen oli Suomeen ja Intiaan. Tämän lisäksi tiimejä on jakautunut Puolaan ja eri Pohjoismaihin. Eräs haastateltava kuvasi tiimiänsä seuraavasti:

”Mul on yks tiimi, joka on kokonaan hajautettu. Tällä hetkellä mun kanssa toimistossa istuu kaksi ihmistä. Sitten mulla on kaksi ihmistä, jotka tekevät pääsääntöisesti toisesta kaupungista. Ovat kerran viikossa konttorilla. Sitten mulla on kaksi tuotepäällikköä, joista toinen on tässä Helsingissä ja toinen Tukholmassa. Ja sitten meillä on kehittäjiä Intiassa.”

## 6.2 Hajautettu ketterä ohjelmistokehitys

Tässä luvussa keskitytään hajautetun ketterän ohjelmistokehityksen toteuttamiseen kohdeorganisaatiossa. Sen lisäksi kerrotaan, kuinka hajautettu ketterä tiimi kommunikoi, ja mitkä ovat odotukset tiimille.

### 6.2.1 Toteutus

Tässä osiossa esitetään tuloksia haastateltavien näkemyksistä ja kokemuksista siitä, miten hajautettua ketterää ohjelmistokehitystä on toteutettu. Hajautettua ketterää ohjelmistokehitystä toteutetaan pitkälti sovelletusti.

”Kyl sitä toteutetaan periaatteessa ihan samalla tavalla kuin tämmöisessä lokaalissa tiimissä. Meillä on vaan vähä eri työvälaineet.”

Organisaatiotasolla on käytössä SAFe, jonka luoja on Dean Leffinwell. SaFe on todistettujen, integroitujen periaatteiden, käytäntöjen ja osaamisen tietämyskan- ta liiketoiminnan ketteryyden saavuttamiseksi Lean-, Agile- ja DevOps-ohjelmien avulla (SaFe, 2020). Organisaatiotasolla tutkimuskohteessa on käytetty SaFe-viitekehystä ketterän kehityksen implementoinnissa. Viitekehysten tar- koituksena on tukea viestintää ja suunnittelua.

SaFe tukee sitä, mitä tiimin sisällä tai tiimien välillä tapahtuu. Viitekehys määrittää, miten tiimit keskustelevat ja suunnittelevat keskenään. Suunnittelu sisältää muun muassa sen, ovatko tiimit riippuvaisia toisistaan, ja millainen

riippuvuusketju on kyseessä. Samalla harkitaan sitä, onko se ylipäättänsä mahdollista. Kaikki, mitä tiimin sisällä tapahtuu, jää pitkälti muun muassa Scrum Masterin ja tiimin vastuulle.

”Scaled Agile Framework on se iso [viitekehys] siinä ympärillä. Se mitoittaa sen, miten tiimit suhteutuu toisiinsa.”

”SAFe on ehkä enemmän semmoinen jättiorganisaatioiden tapa koittaa olla ketterä mun näkökulmasta. Mut, mitä lähemmäksi tiimiä mennään ja mitä lähempänä ite on toiminut tiimiä, niin se on ollut enemmän jotain just Scrum ja Kanban väliltä.”

Lähes kaikkien haastateltavien vastauksissa ilmeni varsinaisista ketteristä menetelmistä Scrum, jota he käyttävät. Yksi haastateltavista mainitsi, että he ovat siirtymässä Scrumban-menetelmän käyttöön. Eräs toinen haastateltavista kertoi, että heidän tiiminsä ei noudata mitään tiettyä menetelmää, vaan he ovat hyvin uskollisia ketterille periaatteille.

”Sinänsä mä vedän nyt Kanban-tiimiä. Me tehtiin päätös Kanbaniin siirtymisestä. Siinä oli montakin eri syytä. Mut joo, Kanban tai Scrumban, tai miksi sitä halutaan nyt kutsua semmoista hybridiä.”

Kaikissa haastatteluissa nousivat esille erilaiset Scrum-seremoniat, joita tiimi kohtaa päivä-, viikko- ja kuukausitasolla. Eri tiimit kuitenkin pitävät seremonioita aikataulullisesti eri tavalla. Lisäksi yksi isompi tiimi ei noudata seremonioita by the book -ajatuksella, vaan enemmänkin periaatteita noudattaen.

Esimerkkinä Scrum-seremonioista eräs haastateltavista nosti esille kahden viikon kestävän iteraation eli sprintin, joka alkaa suunnittelulla ja loppuu sprintin katselmointiin ja retrospektiiviin. Lisäksi tiimillä oli useamman kerran sprintissä kehitysjonon laatua parantava kokous, jossa he katsovat, mitä heillä on työn alla ja mitä heillä on tulossa.

Tiimeillä on aina PI-suunnittelutapahtuma (engl. Program Increment Planning) 12-14 viikon välein. Sen tarkoituksena on suunnitella ja käydä läpi isommalla tasolla liiketoiminnan ominaisuudet. Business Analystien avustuksella on käyty läpi, mihin järjestelmiin tarvitaan muutoksia ja minkälaisia muutoksia tarvitaan. PI-suunnittelutapahtumassa aletaan muodostaa story-tason eli käyttäjätarinan backlogit eli kehitettävät asiat.

”Kerran kolmessa kuukaudessa me tavataan kasvotusten, me keskustellaan ja me juhlimaan meidän onnistumisia isona ryhmänä. Me opitaan siitä, mitä me ollaan tehty väärin tai mikä on mennyt mokaan. Ja sit me suunnitellaan se, mitä me tehdään seuraavat kolme kuukautta.”

”Se on parin päivän semmoinen aika intensiivinen [tapahtuma], missä on useampi agile-tiimi samassa.”

”PI planningissä käydään isommalla tasolla bisneksen featuret läpi, jotka on sellaisia isompia otsakkeita.”

PI-suunnittelutapahtumassa on mukana useampi tiimi. PI:n kesto on noin kolme kuukautta ja PI-suunnittelutapahtumassa suunnitellaan kyseisen jakson työt kerralla. Tiimien väliset riippuvuudet selvitetään ja suunnitellaan. Haastateltavat käyttivät käsitettä ”juna” (engl. Agile Release Train, SaFe framework), joka koostuu useammasta tiimistä.

Tavoitteena on, että kaikki osallistuvat PI-suunnittelutapahtumaan. Osalla haastateltavien hajautetuista tiimeistä on ollut tavoitteena, että PI-suunnittelutapahtumat pidettäisiin kasvotusten. Se ei kuitenkaan ole aina mahdollista esimerkiksi kustannus- tai käytännön syistä johtuen.

PI-suunnittelutapahtuman lisäksi tiimeillä on sprintin suunnittelu (engl. Sprint Planning) ja kehitysjonon laatua parantava kokous (engl. Backlog Refinement). Kehitysjonon laatua parantavassa kokouksessa käydään läpi, missä vaiheessa tiimi on sillä hetkellä ja mitä on tulossa. Kokous pidetään yleensä keran sprintissä.

”Aina niillä viikoilla, kun ei ole sprint planning, meillä on backlock refinement ja siellä katsotaan, et nämä tarinat, mitä olisi tulossa seuraavaan sprinttiin mukaan, että onko niitten speksit kunnossa.”

”Meillä on sprint planning eli silloin kun alkaa sprintti eli jaetaan tehtävät taululle, meillä on JIRA board käytössä, mihin jaetaan tehtävät. Sieltä sitten jokainen saa yksitellen ottaa niitä tehtäviä, kun se sprintti alkaa.”

Lisäksi hajautetuille ketterille tiimeille säännöllinen kohtaaminen on päivittäinen kokous (engl. Daily). Sen pituus on noin 15 minuuttia. Sen tarkoituksena on keskittyä siihen, kuinka tiimin toteutus etenee ja ovatko kaikki ajan tasalla. Päiväkokouksen merkitys korostui myös suuremmassa hajautetussa tiimissä, jonka ketterä toiminta perustui periaatteisiin. Heille päivittäinen kokoontuminen tarkoitti sitä, että muissa yksiköissä työskentelevät ovat perillä raportointimielessä siitä, mitä keskitetyimmässä ohjelmointitiimissä tapahtuu.

”Me käytetään aina 15 minuuttia päivässä siihen, et me ymmärretään, mitä kaikki muut tekee ja miten se, mitä mä teen vertautuu muiden tekemisiin.”

”Daily stand-up, se onkin minun suosikkini, koska daily on mun mielestä tärkein osa minkä tahansa toimivan tiimin tekemistä. Mun mielestä daily stand-up ei saisi olla semmoista aikaraportointia, vaan enemmänkin sen pitäisi keskittyä siihen, että miten se meidän toteutus etenee.”

Sprintit päättyvät lopulta retrospektiiviin, joka on koettu jossain määrin haasteelliseksi kommunikoinnin osalta hajautuksen vuoksi. Retrospektiivissä käydään läpi kulunutta sprinttiä, mikä on sujunut hyvin ja mikä huonosti. Lisäksi tiimi pyrki tunnistamaan kehitystarpeita.

”Retro on oleellinen osa mitä tahansa agile kehitystä, koska siinä aina opitaan menneistä virheistä, mutta tämmöisissä etäretroissa se on pelottavan helppoa, kuinka sinne jäädään vaan hiljaisiksi ja sit loppuu se keskustelu ’Thank you for this session’ ja sit se oli siinä. Sieltä vois tulla erittäin hyviä näkökulmia, mutta se saattaa jäädä kokonaan piiloon.”



”Kuinka hyvin siellä tulee asioita ja ollaanko tartuttu niihin, mitä ollaan otettu action ponteiksi edelliseltä kerralta. Ja onko niihin tullut muutosta ja ehkä semmoinen yleinen ilmapiiri.”

Kuten aiemmin tässä osiossa on kerrottu, eräs haastateltava työskentelee isossa hajautetussa tiimissä, joka on uskollinen ketterille periaatteille. Tiimissä on kokeneita työntekijöitä, mikä on mahdollistanut haastateltavan mukaan joustavamman tavan tehdä ketterää ohjelmistokehitystä.

”Me ei noudateta mitään nimettyä menetelmää. Mutta ollaan hyvin uskollisia tämmöisille ketterille periaatteille. Jos miettii tämmöistä alkuperäistä Agile Manifestoa, niin hyvin sen mukaisesti tehdään asioita. Hyvin paljon sovelletaan ja meillä on kuitenkin sillä tavalla kokenut tiimi, että ihmisillä on 10-15 vuotta kokemusta softakehityksestä. Niin ei meillä oo ollut kauheasti tarvetta lukee kirjasta, miten kannattaisi tehdä.”

Tämä iso hajautettu tiimi on pyritty koordinoimaan siten, että vierekkäin istuvat henkilöt keskittyvät yhteen asiaan. Syynä tällaiseen järjestelyyn on tiimin suuri koko ja hajautus. Tällä tavalla on myös varmistettu tarvittavan kommunikation määrä, joka on ehdoton hajautetuille tiimeille.

”Me ollaan pyritty suunnittelemaan järjestelmäarkkitehtuuri ja se koko set-up niin, että esimerkiksi ne henkilöt, jotka istuu siellä Köpiksessä, niin keskittyy tämän osan tekemiseen. Eikä sillain, että on joku juttu, mikä pitää tehdä ja sit meillä on viides paikkaa ihmiset tekemässä sitä. Koska siinä on sitten se haaste, että jos viidestä eri lokaatiosta tehdään jotain yhtä samaa, niin siinä vaatisi ihan hirveästi sitä kommunikointia, jotta se onnistuisi hyvin. Niin me ollaan pyritty minimoimaan tän jatkuvan kommunikoinnin tarve sillä tavalla, että suunniteltu se järjestelmä niin, että me pystytään palastelemaan se hajautetulle tiimille.”

Kyseinen tiimi ei noudata yhtä uskollisesti säännöllisiä seremonioita. Haastateltava on sitä mieltä, ettei päivittäinen 15 minuutin kokous muuta heidän toimintaansa sen ketterämmäksi. Ketteristä seremonioista heillä on kuitenkin käytössä päivittäinen kokous (daily), koska muut sijaintipaikan työntekijät haluavat raportointimielessä kuulla, mitä keskitetyimmässä ohjelmointia toteuttavassa ydintiimissä tapahtuu. Sen lisäksi heillä on viikoittain tuotteen kehitysjonon työstöä (engl. Product Backlog Refining) koskeva, tunnin kestävä, tarkentava priorisointisessio.

”Tarkennetaan tavoitteita ja prioriteetteja. Tilanteet tuntuu nyt elävän ihan viikkotasolla. Se, mikä oli viime viikolla hirveän tärkeätä, ei seuraavalla viikolla enää jostain syystä ollutkaan. Ja pitää taas sopii uudestaan, mitä me nyt ollaan tekemässä. Me ei edes pyritä tällä hetkellä tekeen mitään kuukauden tason etenemissuunnitelmaa ja splittaa sitä vaikka johonkin sprintteihin. Vaan me joka viikko yhdessä katsotaan, et viime viikolla asioiden tekemisen tärkeysjärjestys oli tämä ja työn alla olevat taskit olivat nämä. Ollaanko me tästä edelleen samaa mieltä?”

”Se on enemmänkin semmoinen, et meitä on kaikki kynnelle kykenevät ja halukkaat. Käydään vaan ylätasolla se läpi. Ja sit, jos me siellä tunnustetaan, et tehtävä X onkin tärkeä, mut kukaan ei oikeastaan tiedä, mitä se pitää sisällään. Niin sit vaan siinä so-

vitaan, että tätä pitää nyt tarkentaa ennen kuin lähdetään koodaamaan. Mut, et miten sitä tarkennetaan, riippuu ihan taskista, eikä meillä ole siinä mitään tiettyä strukturoidua prosessia.”

Kaikissa hajautetuissa tiimeissä työskentelevien on tärkeää muistaa pitää toimivat ”välineet” mukana. Tämä tarkoittaa muun muassa ladattuja kuulokkeita, joiden on oltava mukana erinäisten palavereiden ja äkillisten yhteydenottojen vuoksi. Se on vaatimus, josta ei voi poiketa.

Useammassa haastattelussa nousi esille, että aikaeroista johtuen seremonioiden pitäminen vaatii enemmän suunnittelua. Jokaisen hajautetun tiimin jäsenen sijainti ja aika tulee huomioida seremonioita suunniteltaessa. Eräässäkin tiimissä on huomioitava kolme eri aikavyöhykettä. Suomen ja Intian välinen aikaero on talviaikaan 3,5 tuntia. Sen lisäksi on kuitenkin vielä huomioitava saman tiimin ruotsalaiset kehittäjät, jolloin intialaisten ja ruotsalaisten välinen aikaero on 4,5 tuntia.

”Mulla on Exceli, johon mä sitten isken tällaisia palavereita. Meillä pitää olla kahden viikon syklissä näitä näin monta. Tämän verran aikaa mä tarvitsen siihen. Missä minulla on sellainen kohta tässä päivässä, että minä voin palaverin järjestää? Koska käytännössä tällain talviaikaan meillä on vaan ihan rehellisesti sanottuna kolme tuntia semmoista kunnollista aikaa tehdä töitä yhdessä.”

”Jotenkin tällain talvella, kun Intian aikaero ja Ruotsin aikaero on sellaiset, että me käytännössä pystytään toimiin yhdessä ainoastaan aamupäivisin meidän [Suomen] aikaan. Juuri tällä hetkellä Intiassa kello on 15:42, mut sit Ruotsissahan kello on vasta 11:12. Se on nyt todella vaikeata, mutta kesäaikaan se aikaero on Intiassa vähän omanlaisensa. Niin se heidän kesäaikansa ja meidän kesäaika tekee sen, et se aikaero on paljon pienempi.”

Aikaeroista ja tiimin hajautuksesta johtuen myös dokumentaation määrä ja soveltaminen on yleisempää kuin lokaalissa tiimissä. JIRA on yksi niistä työkaluista, johon tiimi dokumentoi eli kirjoittaa enemmän projektin kulkuun liittyviä asioita. Lisäksi haastatteluissa nousi esille se, että dokumentaatiota saattaa tapahtua siellä täällä sähköposteissa tai erinäisissä keskusteluissa, mitä pidetään haasteena tarvittavan tiedon löytämiseksi.

”Me joudutaan laittamaan ehkä vähän enemmän tavaraa tai tekstiä, selosteita, liitteitä kuin tommoinen lokaalitiimi, koska meillä ei oo sitä, et me voitais koko aika olla katsekontaktissa.”

”Sähköpostiketjuja ja viestiketjuja on hirveästi ja sit jotenkin niistä semmoinen kirjanpito on ehkä itselle työlästä. Sitten jotenkin, missäs olikaan ja kuka olikaan se tyyppi, joka tästä puhui ja tällaisia. Paljon puhutaan, että olisiko tiimin kesken joku nettisivu, minne dokumentoidaan kaikki. Jonkun verran sitä tehdään, mutta en koe, että kauhean systemaattisesti.”

”Ehkä hajautetussa tiimissä koen, että vähän tää just dokumentaation puoli ja sitten tällainen refaktoirointi ja testaus niin ehkä siinä se korostuu, et kun ollaan vähän erillään.”

## 6.2.2 Tiimin kommunikointi

Selkeästi hajautetuissa tiimeissä kommunikointi on merkittävä asia, mutta samalla myös monimutkainen. Kommunikoinnissa tulee ottaa huomioon erilaisia seikkoja. Hajautetussa tiimissä koetaan menetettävän paljon niitä hyötyjä, mitä kasvokkain tapahtuvassa kommunikoinnissa tapahtuu. Työntekijät ovat koneen kanssa mukana kaikissa palavereissa.

”[Kommunikointi] Se on monimutkainen asia. Ensinnäkin siinä tulee niin monta [asiaa], sulla on kulttuurierot, sulla on aikaerot.”

Hajautettujen ketterien tiimien kommunikointi tapahtuu eri viestintätyökalujen avulla. Esimerkiksi päivittäisessä kokouksessa hajautetuilla ketterillä tiimeillä on käytössä pääasiassa eri chat-työkaluja, kuten Microsoftin Skype ja Teams. Sähköpostin käyttö on usean haastateltavan mukaan vähäistä tai vähentynyt. Muita tärkeitä työkaluja kommunikoinnissa ja tiedon jakamisessa ovat JIRA ja Trello.

”Kyl Skype ja Teams on ne mun primäärit välineet, mitkä mulla on koko ajan auki.”

”Meillä ei ole valkotaulua, jonka eteen me voisimme mennä pitämään dailyn. Me avaamme Skype-yhteyden ja katsomme JIRAn eli sen työkalun board ja käydään läpi, mitä siellä on. Ihan samalla tavalla kaikki muutkin palaverit, niin me pidetään ne onlineina.”

”Ja sit meil on Trello tämmöiseen backlog ylläpitoon. Trello on semmoinen web-työkalu. Se on hyvin simppele. Se on niin kun listoja ja listoilla on taskeja ja sitä voi sit vähän käyttää miten tykkää.”

Tiimin jäsenten kesken otetaan pääosin äänipuheluita yhteyden varmistamiseksi. Videopuhelut olisivat toivottuja eri palavereissa, mutta ne ovat olleet haasteellisia huonojen yhteyksien vuoksi. Monet hajautetut tiimit ovatkin tulleet siihen lopputulokseen, että he käyttävät pelkkiä äänipuheluita useista videokokeiluista huolimatta.

Lisäksi tapaamisiin kuuluu näytön jakaminen muille. Hajautetuille tiimeille JIRA-työkalu on tärkeä apuväline havainnoimaan heidän etenemistään. Se korvaa fyysisen ”valkotaulun”. Osassa tiimeistä jäseniä työskentelee samassa toimipisteessä. Tästä huolimatta tiimin jäsenet osallistuvat etätapaamisiin jokainen omalta työpisteeltään. Yhteisen neuvotteluhuoneen varaaminen osalle porukkaa ei ole yhden haastateltavan mukaan välttämättä niin käytännöllistä, koska kokouksia on paljon. Eräs haastateltavista oli vahvasti myös sitä mieltä, että jos hajautettuun ohjelmistokehitykseen on ryhdytty, niin kaikkien tulee olla hajautettuina työskennellessään.

”Eli tämmöisessä hajautetussa tiimissä, niin mä koen, että on parempi sitten, että kaikki on hajautetusti kuin se, että osa porukkaa olisi yhdessä, koska meillä tulee helposti semmoisia klikkejä.”

Kaksi haastateltavaa kertoi, että he ovat tukeutuneet kommunikoinnissa yhteen niin sanottuun viestinvälittäjään, joka on ollut esimerkiksi toinen Scrum Master, kehittäjä, testaaja tai Business Analyst.

”Scrum Master -roolissa mä katson, että mulla täytyy olla joku proxy, joka on siellä paikan päällä. Mä oon yhden henkilön kanssa täysin synkassa. Eli mulla on yks huone täällä, hänellä on se yks toinen huone siellä tuhannen sadan kilometrin päässä. Niin meillä on joku toivo siitä, et me saadaan asia pysymään kasassa.”

Toisen haastateltavan tiimiin kuuluva PO istuu erillään kaikista muista. Haastateltavan mukaan yhteyden saaminen PO:hon on parantunut huomattavasti viestinvälittäjän myötä.

”Tilanne on huomattavasti parantunut, kun me saatiin se [business] analyst siihen tiimiin, joka istuu täällä. Hän on nyt periaatteessa sitten meidän first point of contact näissä PO asioissa ja hänen tehtävänsä on ikään kuin tuottaa meille nyt vaatimuksia, ja hän saa sitten tavaransa PO:lta.”

Useammassa haastattelussa nousi esille lisäksi ad hoc -keskustelut. Haastattelujen perusteella nämä ad hoc -keskustelut ovat työntekijöille arkea kiireellisten tapausten käsittelemiseksi.

”Me joudutaan aika usein ottaa semmoisia ad hoc -puheluita. Ne pingaa ’call?’ ja sitten me soitellaan toisillemme.”

Eräs haastateltavista on sitä mieltä, että yhteinen ryhmächat olisi paljon parempi kuin, että yksittäiset ihmiset ottavat Skype-keskusteluita keskenään. Haastateltava uskoo, että sen myötä tiimihenki kohenisi. Lisäksi työntekijä voisi etsiä mahdollisia avoimien kysymyksien vastauksia yhteisestä keskustelusta, jos tämä on saman kysymyksen äärellä.

Isojen tiimien eli noin 30 hengen porukoissa Skype-puheluja on pyritty järjestämään niin, että tapaamiseen osallistuvat vain ne, joiden täytyy olla mukana. Tällä on pyritty myös siihen, että se olisi enemmän henkilöihin perustuva.

Hajautetuille ketterille tiimeillä kasvotusten tehtävät tapaamiset ovat vähäisiä. Ne ovat kuitenkin toivottuja silloin tällöin. Yhdessä tiimissä on aiemmin pyritty kuukausittaiseen kohtaamiseen kasvotusten, mutta näistä on myöhemmin luovuttu organisaation muutosten vuoksi.

”Mä en oo nähnyt heitä ketään. Ainoastaan äänen perusteella ollaan kommunikoinnissa.”

”Mä en oo käynyt Intiassa, mutta intialaiset on käynyt täällä kyllä useammankin kerran. Täältä siirretään tietoa sinne, plus että he siirtävät omaa osaamistaan tänne meille. On ollut semmoisia 2-3 kuukauden jaksoja, kun sieltä on tullut joku. Tässä oli syksyllä yks kaveri ja se oli ihan mukava kokemus kyllä.”

### 6.2.3 Odotukset hajautetulle ketterälle tiimille

Tuloksista nousi selkeästi esille se, että tiimiltä odotetaan valmiita toiminnollisuuksia (engl. features) ja tuloksia. Monesta haastattelusta kävi ilmi, että tiimin tavoitteena on saada tuotettua se, mitä on pyydetty ja mihin tiimi on lupautunut.

”Meiltähän odotetaan, että me tehdään featureita valmiiksi. Käytännössä se tarkoittaa sitä, että tavallaan PO:n kautta meille syötetään vaatimuksia. Sitten meiltä odotetaan, että se, mitä me ollaan luvattu siellä PI planningissä, niin se saadaan sitten toteutettua siihen pisteeseen asti, että se lähtee meidän tiimiltä pois.”

”Meillähän tulee tietysti backlogille tehtäviä. Odotukset on, et me saadaan ne aina tiettyyn aikaan mennessä tehtyä.”

Kokeneimmilta tiimeiltä ei enää odoteta sitä, että niiden täytyisi noudattaa tarkasti jotain nimettyä ketterää menetelmää, vaan pikemminkin tiimiltä odotetaan uskollisuutta ketteryyden pääperiaatteille. Kehittyneeltä tiimiltä odotetaan täten myös kykyä soveltaa näitä ketteriä periaatteita projekteissa.

”Mä uskon siihen, et sitten kun tiimi on uusi, niin silloin on tosi hyvä, että on tällaisia erilaisia seremonioita. Mutta, kun tiimi on tottunut toimii keskenään, niin niitä on voinut alkaa vähän vapautuneemmin soveltaa.”

Tähän toimintamalliin liittyy se, että tavoitteet tehdään selviksi, ja tiimille annetaan vapaus ja vastuu toimia.

”Me ei pyritä mitenkään ‘top-down’ sanomaan, että teillä pitää olla sitten yksi henkilö nimettynä Scrum Masteriksi ja sitten pitää tähän järjestelmään merkkää taskit ja joka aamu muuten pitää olla daily. Ei me tämmöisiä seremonioita jakseta mikromanageroida. Vaan niin kun tavoitteet selväksi ja vapaus sille tiimille toimia.”

”Sen pitäisi olla enemmän sitä, et ymmärretäänkö me meidän tavoitteet siihen suuntaan kuin siinä sprintissä pitäisi mennä.”

Tiimin toiminnalta odotetaan myös sitä, että yhteistyö toimii, eikä siihen pitäisi joutua käyttämään liikaa ylimääräistä energiaa. Lisäksi tiimiltä odotetaan ja toivotaan sitä, että hajautettu tiimi tekee työtä yhtä lailla kuin lokaalitiimi. Yhdessä haastattelussa nousi esille myös se, että tiimissä muistettaisiin antaa palautetta toisille jäsenille.

## 6.3 Haasteet

Hajautuksen tuomia haasteita ketterään ohjelmistokehitykseen selvitettiin haastattelujen avulla. Haastatteluissa nousi esille eniten kasvotusten tapahtuvan viestinnän hyötyjen menetys ja kommunikointiin liittyvät haasteet sekä tekniset että kulttuurilliset haasteet. Näiden lisäksi muita merkittäviä haasteita olivat

myös se, etteivät tiimin jäsenet tunteneet toisiaan kovin hyvin. Myös kielihaasteet, väärinymmärrykset, riippuvuudet muista tiimeistä, etäisyys, aikaero, epäselvät tavoitteet, informaation kulku ja tiimin jäsenten välinen yhteistyö koettiin haasteina.

### 6.3.1 Kasvotusten tapahtuvaan työskentelyyn liittyvien hyötyjen menetys

Kasvotusten tapahtuvaan työskentelyyn liittyvien hyötyjen menetys oli isoin haaste. Tämän mainitsivat melkein kaikki haastateltavat. Menetettyt hyödyt liittyivät paljon kommunikointiin ja siihen, kuinka hyvin tiimin jäsenet tuntevat toisensa. Lisäksi hyötyjen menettäminen on vaatinut lisätyötä siihen, että kommunikointia tapahtuu tarpeeksi tiimin jäsenten välillä.

*”Tää on ihan mielenkiintoinen asia, että jos meillä on yksikin henkilö tiimistä, joka ei istu samassa paikassa kuin kaikki muut, niin menetetäänkö me kaikki face-to-face -hyödyt vai menetetäänkö me vaan osa? Mun mielipide on, et me saatetaan jopa menettää kaikki hyödyt siitä.”*

#### Kommunikointi

Koettuun hyödyn menettämiseen liittyvät paljon sosiaalinen kanssakäyminen ja kommunikointi. Haastatteluista kävi ilmi, että kommunikoinnin määrä on vähäisempää sekä se, että siitä puuttuu osittain tavallinen jutustelu (engl. small talk). Haastatteluista kävi ilmi, että jutustelulla tarkoitetaan rentoa keskustelua, esimerkiksi kahvikupin äärellä, jolloin nähdään myös eleet ja kehonkieli.

*”Ihmiset ovat sosiaalisia eläimiä, jos lähdetään ihan siltä tasolta. Kaikki katsekontaktit, keskustelut, kaikki se, että sä menet juomaan kahvia jonkun kanssa ja kysyt ’Hei mitä sä teit viikonloppuna?’. Tää on asia, mitä Skypessä ei tehdä.”*

*”Kommunikointiongelmia ovat suurimpia ja ihan ollut näissä jo kautta linjan kaikkien tiimien kanssa. Kun kuitenkin ketterä ohjelmistokehitystiimi perustuu sellaiseen ajatukseen, että ollaan täällä yhtenä tiiminä lähellä toisiamme tekemässä tätä juttua.”*

*”Kun se kaveri istuu siinä vieressä, sä näet sen tilanteen, milloin sä voit häiritä sitä. Sitten, kun sä otat yhteyden tuonne Intiaan, sä et näe. Sä voit häiritä jotakin tosi pahasti. Kaverilla voi olla huono hetki tai jotain tällaista. Totta kai se on tämmöinen käytännön ongelma.”*

Hajautetun tiimin koordinoiminen vaatii enemmän suunnittelua ja aikataulutusta. Sen vuoksi eri seremoniat (päivittäiset ja viikoittaiset kokoukset) tukevat säännöllistä kanssakäymistä. Se ei kuitenkaan lisää tavallisen jutustelun määrää, jota normaalisti syntyisi lokaalin tiimin yhteisessä työskentelytilassa.

#### Vähäinen jäsenten välinen tuttavuus

Tuloksista nousi esille myös se, että joissakin hajautetuissa tiimeissä kaikki jäsenet eivät tunne toisiaan kovin hyvin. Osa haastateltavista koki tämän asian haasteena. Sosiaaliset kontaktit, joissa puhutaan työajan ulkopuolisista aiheista, jäävät vähemmälle, ja täten kollegoiden välinen suhde ei kehity samalla tavalla kuin kasvotusten.

Varsinkin uuden työntekijän sopeutuminen tiimiin voi olla haasteellista, jos kasvotusten kohtaaminen ei ole mahdollista. Se voi vaikuttaa myös siihen, kuinka hyvin työntekijä tulee jatkossa toimeen muiden kanssa, mikä voi taas vaikuttaa projektin kulkuun.

”Ihmiset ei välttämättä tunne toisiaan niin hyvin. Tavallaan se, että sä et tarkalleen tiedä mitään, mitä tää henkilö oikeasti ajattelee. Jos se on pelkkä se Skype-ikoni, josta kuulee aina silloin tällöin, niin kyllä siinä menetetään paljon.”

”Meillä on Helsinki-porukka ja sitten on muut. Helsinki-porukka on hyvin kavereita keskenään. Porukka käy töiden jälkeen yhdessä syömässä. Sen lisäksi, et on työkavereita, niin porukka on ihan siviilissäkin kavereita. Toisin kuin näitten muitten alueiden kanssa. Selvä ero.”

### 6.3.2 Tekniset haasteet

Yksi merkittäväksi koettu haaste liittyy teknisiin vaikeuksiin. Hajautettu ketterä ohjelmistokehitys vaatii paljon teknistä tukea, jotta tiimi pystyy toimimaan. Sen sijaan, että tiimin jäsen voisi kävellä työtoverinsa työpisteelle, hän joutuu avaamaan jonkin viestintäpalvelun. Moni tekninen haaste liittyy yhteyksien katkeamiseen. Lisäksi videoyhteydet ovat hajautetuilla tiimeillä harvassa huonojen yhteyksien vuoksi. Eräs haastateltavista totesi, että yhteyden katkeaminen kriittisellä hetkellä on enemmän sääntö kuin poikkeus.

”Loppupeleissä yhteydet Intiaan oli niin huonot, että he eivät päässeet Skypeen tai se Skype ei vaan toiminut. He joutuivat aina soittamaan tämmöisestä konferenssipuhelimesta, siis lankapuhelinnumerosta Skypen numeroon. Se aiheutti sen, että sieltä kuulu hirveä melu taustalta. Siinä on monia haasteita ihan näissä teknisissä asioissa.”

”Meillä on huone- ja ääniongelmia. Ei oo kerta eikä kaksi, kun me sanotaan esimerkiksi dailyn aikana, et ’Hei nyt sä puhut ihan kuin sä olisit Marsissa’. Eli siis se äänenlaatu heikkenee. Me menetetään siitä informaatiosta ihan varmasti tosi paljon. Se on jotenkin niin käsittämätöntä. Mä en ymmärrä, miksi me ei voida pistää enemmän paukkuja näihin yhteyksiin, että ne saataisiin paremmiksi. Koska jos tää on meidän päätös, et me tehdään tällein hajautetusti töitä, niin sit meillä pitäisi olla työkalut, jotka tukee sitä hajautettua työtä.”

”Haasteista tuli mieleen se, että etäyhteydet ei aina toimi. Se on haasteena mun mielestä ihan jokaisessa kokouksessa. Eli aina jonkun headset (suom. kuulokkeet) pettää ja joku ei pääse edes liittyä ja jonkun puhe ei kuulu.”

Hajautettujen tiimien tekniset haasteet eivät rajoitu vain yhteyksiin tiimin jäsenten välillä. Hajautetuille tiimeille JIRA-sovellus on tärkeä työkalu työskentelyn suunnitteluun, seuraamiseen ja raportointiin. Tiedot ovat kokonaan sähköisesti saatavilla.

”Työvälineisiin liittyviä haasteita on se, että JIRA, jonka mä mainitsin, on siis koko yrityksen yhteinen työkalu, joka ei toimi aina. Voi olla, että se on ihan kokonaan down. Tai sitten voi olla, niin kuin meille kävi maanantaina, että me yritettiin tehdä plänäystä, niin aina vuorotellen joku Skypessä sitten jakoi näyttönsä, kenellä sattui

JIRA juuri sillä hetkellä toimimaan. Eli me menetettiin ihan valtavasti aikaa, kun me kyseltiin, kenellä toimii JIRA ja kuka haluaa jakaa. Ja kun se on meidän ainoa tärkeä työkalu siinä palaverissa.”

Lisäksi Skype ja Teams -työkalujen välillä on ollut eroavaisuuksia. Erään haastateltavan mukaan Teams on toiminut Skypeä paremmin etäyhteyksissä. Useampi tiimi onkin siirtynyt käyttämään Teamsiä Skypen sijaan.

Osa haastateltavista kokee haasteeksi myös sen, että eri sovellusten käyttö on liian rajattua organisaatioissa. Yksi haastateltavista kyseenalaistikin sitä, ettei ymmärrä, miksi organisaatio on tehnyt niin tiukan linjan. Haastatteluiden perusteella työntekijöillä olisi intoa ja halua käyttää muitakin työkaluja työssään. Siinä, mitä työkaluja voidaan käyttää, maiden välillä on eroja. Tähän vaikuttaa muun muassa se, että eri maissa työskennellään eri verkoissa.

”Niin se, mitä mä voin käyttää on tosi rajallista valitettavasti. Se on oma haasteensa. Meillä olisi ihan valtavasti kaikkia kivoja työkaluja käyttää tuolla, mutta kun me ei voida käyttää.”

”Se, miten me oltaisiin onnistuttu vielä paremmin niin olisi se, että meillä olisi ollut vielä enemmän vapautta valita meidän työkaluja. Esimerkiksi täällä on todella harmillista, että se on rajoitettua.”

### 6.3.3 Kulttuurierot

Monesta haastattelusta kävi ilmi, että kulttuurierot vaikuttavat yhteiseen tapaan työskennellä. Se on saanut aikaan monia haastavia hetkiä tiimin jäsenten välillä. Puhetapa, kielenkäyttö ja äänensävyt vaikuttavat tunnelmaan tapaamisissa.

Kielihaasteiden ohella myös kulttuurierot ovat haastateltavien mukaan merkittävä haaste. Joissain tiimeissä, joissa ihmiset ovat työskennelleet pitkään yhdessä, on kansainvälistyminen aiheuttanut epäluottamusta uusista työntekijöistä kohtaan ja tuonut haasteita, joissa eri käyttäytymistavat ovat koetuksella. Se on näkynyt jopa tyykeytenä toisia kohtaan.

Haastatteluista kävi ilmi, ettei kaikissa tiimeissä ole rakentunut täysin yhteistä tapaa työskennellä. Eri kulttuurit vaikuttavat paljon ihmisten tapaan käyttäytyä ja toimia. Joissakin kulttuureissa työskentely voi olla hyvinkin suoraviivaista ja toimintaorientoitunutta, kun taas toisissa projektien eteneminen on ollut hidasta, koska tiimin sisällä on ollut liikaa aikaa vievää keskustelua. Silloin päätöksen tekeminen on ollut vaikeampaa.

Henkilöiden omat kokemukset, muiden kokemukset ja mielipiteet sekä yleistyksiset ja ennakkokäsitykset vaikuttavat siihen, kuinka eri tiimin jäsenten työtapakulttuuriin suhtaudutaan. Esimerkiksi eurooppalaisissa kulttuureissa keskustelun ja päätöksenteon prosesseissa on nähty eroja eri maissa olevien kollegoiden kesken. Pääsääntöisesti nostettiin esiin se, että Ruotsissa on keskusteluvampi tapa yleisempää kuin Suomessa. On myös koettu, että toisissa kulttuureissa kysymiseen voi liittyä kynnyks, joka johtuu mahdollisesti kulttuuriin liittyvästä häpeästä tai kasvojen menetyksestä.



Tuloksista kävi myös ilmi se, että joissakin hajautetuissa tiimeissä on pyritty harjoittamaan tiettyjä käytös- ja toimintatapoja, esimerkiksi uskallusta esittää avoimia kysymyksiä. Yksi haastateltavista kertoi, että hän kysyi ensimmäisten kuukausien aikana jokaiselta kehittäjältä erikseen joka päivä, onko jotain, jossa hän voi auttaa ja onko mielessä avoimia kysymyksiä. Harjoitusten pyrki- myksenä oli se, että aina pitää kysyä ja aina saa kyseenalaistaa.

### 6.3.4 Kielihaasteet

Koska haastateltavat työskentelivät kansainvälisissä hajautetuissa tiimeissä, ei työkielenä ollut välttämättä oma äidinkieli. Pääasiallisena työkielenä näissä hajautetuissa tiimeissä toimi englanti. Kielihaasteiden taso vaihteli tiimeittäin. Kaikissa tiimeissä se ei kuitenkaan ollut ongelma, mikä tuli ilmi haastatteluissa.

”Nimenomaan tässä hajautetussa ja tässä et me ollaan monen eri valtion alueella, me puhutaan eri kieliä. Se on ihan täyttä tuskaa, että ne pystyy kommunikoimaan keskenään. Se on sellainen haaste, jota mä en ite ollenkaan tajunnut, et se voi olla niin paha se tilanne.”

”No meillä on haasteita siinä, kun vaikka kaikki puhuu englantia, niin sit kuitenkin tuntuu, että tommoisissa kun käydään mailikeskusteluja tai chattia, niin se ei kuitenkaan oo hirveen luontevaa kaikille ihmisille.”

### 6.3.5 Muita esille nousseita haasteita

Tuloksista nousivat esille myös seuraavat haasteet: väärinymmärrykset, riippuvuudet muista tiimeistä, etäisyys, aikaero, yhteinen suunta ja ymmärrys tavoitteesta ja tiimin sisäinen yhteistyö. *Väärinymmärrykset* ovat usein johtuneet haastateltavan sanoin ”rikkinäisestä puhelimesta”. Tämä tarkoittaa sitä, että viesti muuttuu matkanvarrella osallisten välillä. Informaation kulku koetaan erään haastateltavan mukaan jopa primäärinä haasteena.

”Väärinymmärrykset on isoja. Vaikka mä tekisin kaikkeni. Siinä on aina joku mahdollisuus, et he ovat ymmärtäneet väärin tai mä oon puhunut englantia liian huonosti tai yhteys on katkennut kriittisellä hetkellä.”

”Sitten on semmoisia, et jos sä et pääse sinne kokoukseen, niin joku meistä laittaa mailia perään. Helpommin tulee semmoista rikkinäistä puhelinta. Ja helpommin tulee sitä toisteisuutta. Samat asiat kokoustetaan useampaan kertaan.”

Myös tiimin jäsenten välinen *etäisyys* koettiin vaikeana. Ongelma on se, ettei tiedetä muuta kuin, mitä linjoja pitkin kuullaan. Etäisyydestä johtuvat haasteet linkittyvät pitkälti myös kommunikaatioon ja viestin välitykseen.

”Kyllähän tosiaan on se, että he ovat niin kaukana ja meillä ei oo mitään muuta käsitystä kuin mitä puheluissa tulee ilmi. Ei oo käsitystä edes tämän kehittäjän ammattitaidosta. Tuntuu, että pitää selittää hyvin perusasioitakin jopa. Ei oo semmoista näkemystä siitä, että tapahtuuko siellä mitään.”

”Tää, että kaveri istuu vieressä, niin totta kai se on helpompaa tehdä sitä hommaa kuin se, että ottais Skypellä yhteyden sinne Intiaan ja yrittää sitten selittää. Toki me aina jaetaan työpöytä ja samalla kerrotaan, että mikä on ongelman nimi, mutta se että nyt esimerkiksi, kun tää yks kaveri oli täällä pari kuukautta syksyllä, niin oli se niin paljon helpompaa samalla vaan koputtaa olkapäästä, että ’Katoppas nyt tätä näin. Mikä tää homma on?’. Haasteena on kyllä nimenomaan tämä etäisyys. Ei se ole niin helppoa kuin se, et kaveri istuu vierellä.”

Monet hajautetut tiimit ovat riippuvaisia toisista tiimeistä, ja tällöin hajautus näkyy tiimin jäsenten lisäksi myös eri tiimien välillä. Tiimeillä voi olla riippuvuuksia myös oman yksikön ulkopuolella. *Riippuvuusongelmat* ovat näkyneet luottamuksen puutteena.

”Tässäkin tapauksessa, missä me nyt työskennellään, niin meillä on paljon riippuvuuksia, joihin me ei pystytä suoraan vaikuttamaan. Elikkä meidän pitää uskoa heidän lupauksiinsa, että joku asia on esimerkiksi valmis sprintissä kolme, jonka jälkeen me voidaan sitten itse työskennellä neljä, viisi, kuusi ja sitten se on valmis lopussa. Mutta kokemus on osoittanut, että näihin ei nyt välttämättä voi luottaa.”

”Meillä on hirveästi riippuvaisuuksia muista tiimeistä. Yleensä aina tulee jotain hidastuksia. Joku toinen tiimi ei oo vielä valmis tai tulee muutoksia kesken työnteon ja tämmöisiä.”

Joidenkin tiimien kohdalla *aikaerot* vaikuttivat haastavasti työarjen suunnitteluun. Joillekin aikaero oli sen verran pieni, että sen kanssa pärjättiin ilman suurempaa ongelmaa. Haastatteluista kävi ilmi, että aikaerot on huomioitava suunniteltaessa tapaamisia.

”Se on tällä hetkellä 3,5 tuntia. Ei ihan mihin aikaan tahansa pysty. Joko meidän pitäisi tulla tänne todella aikaisin tai heidän pitäisi olla vastaavasti myöhään siellä. Kyl siinä on omat haasteensa.”

Kahdessa haastattelussa nousi esille haaste siitä, onko tiimissä aina *yhteinen suunta ja käsitys tavoitteesta*. Toisen haasteltavan vastauksesta kävi ilmi, että tiimin suuresta koosta johtuen joidenkin työntekijöiden rooli on pienentynyt, mikä on johtanut siihen, että he ovat jossain määrin irrallaan siitä, mitä ydinporukassa tehdään. Toinen puolestaan vetosi haastattelussaan siihen, että vähäinen dokumentointi vaikuttaa yhteiseen suuntaan ja käsitykseen tavoitteesta.

”Jonkun verran on ongelmia sen kanssa, että kun ollaan agileja ja ei hirveästi dokumentoida ja muuta vastaavaa, niin lähdetään vähän niin kun soitellen sotaan, ettei tiedä ihan tarkkaa, mitä nyt ihan tarkasti ollaan tekemässä. Sitten siitä tulee yllätyksiä, että ’ai tää tarkoittikin tämmöistä’ tai sitten vaikuttaa aikatauluihin.”

Yksi haastateltavista korosti *tiimin jäsenten välistä yhteistyötä* ja sitä kuinka haastavaa se on hajautuksen myötä.

”Ja toiseksi isoin, mitä nyt voi vielä tarkentaa... softakehitys on aktiviteettina semmoinen, että... se on vähän kuin yrittäis jotain maalausta rakentaa sillain, että yks maalaa sitä Tanskasta ja yks maalaa sitä Suomesta ja ne ei juttele ihan hirveästi tois-

tensa kanssa eikä varsinaisesti näe sitä lopputulostakaan. Ihan hirveätähän siitä syntyy. Et kyllähän sitä pitäisi tehdä yhdessä tässä pöydän ääressä. Luova aktiviteetti pitäisi tehdä yhdessä.”

Haasteita on monia, kuten tuloksista ilmeni. Ketterästi työskentelevän tiimin hajauttaminen vaikuttaa paljon niihin hyötyihin, mitä kasvotusten tapahtuva työskentely antaa.

## **6.4 Hajautetun ketterän ohjelmistokehitysprojektin onnistumisen ja sen mittaaminen**

Tässä osiossa kerrotaan, miten kohdeorganisaatiossa määritellään hajautetun ketterän ohjelmistokehitysprojektin onnistuminen sekä se, miten sitä mitataan. Onnistumisen määrittely johdattelee myös viimeiseen käsiteltävään teemaan, jossa kerrotaan haastateltavien esittämät kriittiset menestystekijät.

### **6.4.1 Hajautetun ketterän ohjelmistokehitysprojektin onnistumisen määrittely**

Moni haastateltava kertoi, että hajautetun ketterän ohjelmistoprojektin onnistuminen liittyy asiakastytyväisyyteen, tuotantoon viemiseen, tavoitteisiin pääsemiseen ja aikataulussa pysymiseen. Myös arvioitujen kustannusten puitteissa toteutettu valmis projekti nähtiin onnistumisena.

”No joo sillä tavalla, että kyl me ylpeillään sillä, kuinka usein me esimerkiksi pystytään tekemään uusia ominaisuuksia, kuinka usein me tuodaan uutta kamaa kuluttajille.”

”Me ollaan tyytyväisiä, jos me saadaan asiat tuotantoon.”

Lisäksi tavoitteisiin pääseminen oli selkeästi useammalle haastateltavalle merkki siitä, että projektissa on onnistuttu. Tavoitteeseen pääsemiseen liittyy myös vahvasti se, että pysytään sovitussa aikataulussa ja ollaan sitoutuneita.

”Mä tykkään vähän sellaisista kovista arvoista onnistumisen suhteen, että sit päästään siihen tavoitteeseen, mitä tavoitellaan ja pysytään aikataulussa. Se on ennen kaikkea se ensimmäinen juttu.”

Valmiin projektin ja tavoitteiden rinnalla onnistumisena koettiin tyytyväiset asiakkaat ja myös se, että heitä on paljon. Tähän liittyy sekä asiakasyrityksen että loppukäyttäjän tyytyväisyys, riippuen siitä, kenelle ollaan tuottamassa.

”Jos asiakastytyväisyys on korkea projektin jälkeen, niin tietenkin me voidaan todeta, että projekti on onnistunut.”

”Yleensä ne iloiset ihmiset löytyy sieltä liiketoiminnan tai sitten ihan sieltä asiakasrapinnasta.”

Eräs haastateltavista nosti esille myös sen, että projekti on menestynyt, jos on pysytty budjetissa. Lisäksi haastateltava totesi, että menestynyt projekti on myös sitä, että se on onnistuttu tuottamaan halvemmalla, koska tiimi on ollut hajautettu ja siinä on hyödynnetty ulkoistamista.

”Mikä on se juttu ‘happy case’ projektin päättymisessä? Se on varmaa se, et jos me ollaan onnistuttu tekemään se halvemmalla tai nopeammin tai ehkä paremmin kuin mitä me oltaisiin tehty pelkästään, jos meillä olisi vain on-site -tiimi.”

Useammassa haastattelussa korostettiin tiimin yhteistyötä. Onnistunut yhteistyö on myös merkki siitä, että tiimihenki on parantunut ja sitä on enemmän. Tämä tarkoittaa esimerkiksi sitä, että työntekijät uskaltavat pyytää toisiltaan apua ongelmiinsa. Toimiva yhteistyö on perustana pitkäkestoisen tiimin rakentamiselle.

Yhteistyön merkitys korostuu myös eri tiimien välillä puhuttaessa riippuvuuksista. Onnistuakseen se vaatii riippuvuuksien koordinoitua menestyksekkäästi.

”... niin ne riippuvuudet muihin tiimeihin tavallaan hanskataan siinä PI plänäyksessä. Eli jos meitä pyydetään tekemään joku ominaisuus X, niin me ei pystytä sitä yksin tekemään meidän yhdellä tiimillä, vaan me tarvitaan siihen muita tiimejä, jotka käytännössä melkein kaikki on jossain muualla paikkakunnalla tai muualla maassa. Niin silloin siellä PI plänäyksessä näiden muiden tiimien kanssa sovitaan, että heillä on samat tavoitteet.”

”Mutta sitten, jos rakennetaan tällaista pitkäkestoisempaa tiimiä, jonka tarkoitus on olla monia vuosia yhdessä tai saada se putki toimimaan niin, et se on jatkuvaa kehitystä, niin sit yks tosi tärkeä onnistumisen mittari mun mielestä on se, että tiimi kokee onnistuvansa yhdessä, ollaan tiimiydytty, juhlietaan yhdessä ja tehdään asioita yhdessä.”

Eräs haastateltavista kuvasi onnistumista myös sillä, että tiimi on kyennyt hoitamaan jonkin ylimääräisen tehtävän tai, että se on pystynyt ratkaisemaan jonkin tuotannon ongelman.

”Enenevässä määrin nyt on ruvettu kokemaan onnistumisena semmoinen, että on ollut joku ongelma ja siitä on uskallettu puhua ääneen, ja sit tiimi on pystynyt ratkaisemaan sen keskenään.”

Lisäksi yhdessä haastattelussa tuli esille kaksi merkittävää onnistumiseen liittyvää asiaa. Toinen liittyy siihen, että tiimin toiminnan hyödyllisyyteen uskotaan, ja että he saavat jatkaa edelleen toimintaansa, mikä ei ole aina välttämättä itsensänselvyys. Toinen liittyy siihen, että tiimi on saanut näkyvyyttä yrityksen sisällä.

”Ja sit tietysti tämmöinen perinteinen merkki onnistumisesta isossa korporaatiossa on, et mikäli jos joku toimitusjohtaja tai joku mainitsee meistä jossain julkisessa esitelmässä, niin se on tietysti semmoinen sulka hattuun sisäisesti isoissa korporaatioissa. On semmoista näkyvyyttä myös ylemmässä johdossa.”

Oikein määritellyt tavoitteet koettiin myös yhtenä onnistumisen merkinä. Tämä nousi esille eräässä haastattelussa, jossa haastattelija korosti tavoitteiden oikein määrittämistä ja niihin panostamista.

”Myös onnistumisena näen sen, että työtehtävät on määriteltyjä eli se PI ei mee pelkään suunnitteluun ja puhumiseen, vaan ikään kuin planningissä ollaan onnistuttu määrittämään semmoiset työtehtävät, jotka pystytään toteuttamaan niitten sprinttien aikana.”

#### 6.4.2 Onnistumisen mittaaminen

Onnistumisen mittaaminen voidaan jakaa seuraaviin kategorioihin: rahallinen mittari, ketterät (agilet) metriikat, tiimityytyväisyys, epäsuorat arvon mittarit ja asiakastyytyväisyys. Tuloksien perusteella ketterät (agilet) metriikat olivat eniten käytössä.

”Mä haluan aina puhua niistä agileista metrikoista. Paljon me ollaan tuotettu? Monta julkaisua me ollaan tehty ajanjaksossa? Monta user storya me ollaan devattu alusta loppuun? Monta feature on? Mikä on se business value?”

”No meidän RTE (engl. Reliese Train Engineer) mittaa onnistumista silleen, että paljonko me toimitetaan story pointseja.”

Rahalliset mittarit auttavat seuramaan käytettyä budjettia. Se ei ole kuitenkaan erään haastateltavan mukaan niin toivottava mittari, koska on liian helppoa katsoa sitä, mitä on kulutettu tietyssä ajanjaksona versus siihen, mitä tiimi on saanut lopulta aikaan.

”Budjetti saattaa olla ainoa meidän semmoinen rahallinen mittari, koska kaikille ei voida laskea sitä, paljonko se tuottaa niin kun loppupeleissä.”

Toisen haastateltavan mukaan tiimi on käynyt keskustelua siitä, pitäisikö heidän miettiä enemmän tavoitteitaan generoidun rahan kannalta. Tässä tiimissä on käytetty vielä epäsuoran arvon mittareita. Ajatus siitä, sopivatko rahalliset mittarit mittaamaan onnistumista, on tässäkin riippuvainen siitä, kohdistuuko tuotanto loppukäyttäjien vai sisäisen liiketoiminnan kehittämiseen.

”Nythän meillä on ollut hyvin tommoisia epäsuoria arvomittareita: käyttäjien lukumäärä, käyttäjien tyytyväisyys, päivitysten lukumäärä vuodessa ja näkyvyys ylemmän johdon slideissa.”

Tiimin tyytyväisyyttä mitataan erään haastateltavan mukaan joka iteraation päätyttyä. Hyväksi koettu iteraatio ja siitä koitunut hyvä mieli on merkki onnistumisesta. Myös toinen haastateltava mainitsi haastattelussa anonymistia palauttehetkestä, joka pidetään aina retrospektiivin yhteydessä. Kaikki tiimin jäsenet voivat antaa palautetta, jotka puretaan jälkikäteen yhdessä.

”Meillä on joka iteraation päätteeksi tämmöinen kysely, kuinka happy olet tämän iteraation päättyessä. Ja se näkyy ihan selvästi, jos siellä on ollut joku [ongelma].”

Asiakastyytyväisyyttä on mitattu epäsuorilla arvomittareilla. Asiakaspalaute on toiminut mittarina loppukäyttäjille menevien palveluiden osalta. Eräs haastateltava kuitenkin kokee sen jossain määrin myös ongelmana luotettavuuden näkökulmasta. Asiakastyytyväisyyttä seurataan myös sillä, paljonko jokin palvelu on saanut tähtiä loppukäyttäjiltä.

”Asiakaspalautteessa on se ongelma, että sitä ei koskaan tiedä, että kuka sitä palautetta antaa. Onko 90 % ihmisistä ihan tyytyväisiä vai vihaisia vai miksi ei koskaan kuulu niistä yhtään mitään. Se porukka ihmisiä, jotka antaa palautetta, sehän on ihan murto-osa näistä kaikista käyttäjistä.”

Haastatteluihin perustuen yhteenvedona voidaan todeta, että onnistumiseen liittyvät paljon valmiiden ohjelmistojen tuottaminen, tavoitteisiin pääseminen, aikataulussa pysyminen sekä asiakastyytyväisyys. Voidaan siis todeta, että haastateltavien onnistumisen määritelmät ovat lähempänä Misran ym. (2009) määrittelemää onnistunutta ohjelmistokehitysprojektia. Haastatteluiden perusteella voidaan myös todeta, että yleisimmin käytetyt onnistumisen mittarit ovat ketterät (engl. agile) metriikat ja epäsuorat arvomittarit.

## 6.5 Haastateltavien esittämät hajautetun ketterän ohjelmistokehityksen kriittiset menestystekijät

Tässä osiossa kerrotaan haastateltavien esittämät hajautetun ketterän ohjelmistokehityksen kriittiset menestystekijät. Tämä on myös viimeinen käsiteltävä teema. Teemahaastattelutilanteessa haastateltavalle kerrottiin, miten ohjelmistokehityksen kriittiset menestystekijät on määritelty tieteellisessä kirjallisuudessa. Alkuun haastateltavalle annettiin oma vapaus kertoa näkemyksensä siitä, mitkä ovat ne menestystekijät, joilla onnistumiseen päästään. Tämän jälkeen haastateltavalle annettiin täytettäväksi kysely, johon oli kerätty kirjallisuudesta poimitut menestystekijät.

Oheisessa kappaleessa esitetään ensin vapaamuotoiset vastaukset, jonka jälkeen tulevat kyselyn tulokset perusteluineen. Haastattelujen perusteella tärkeimmät kriittiset menestystekijät painottuivat ihmistekijöihin. Haastatteluissa nousi esille erityisesti kommunikointi, tiimin sisäinen yhteistyö, yhteinen suunta ja ymmärrys sekä asiakkaan osallistuminen. Lisäksi muita merkittäviä kriittisiä menestystekijöitä ovat tiimin kyvykkyys, johdon tuki, tekninen pätevyys, henkilökohtaiset ominaisuudet, seuranta sekä koulutus ja oppiminen.

### Tiimin sisäinen yhteistyö

Haastateltavat kokivat, että tiimin sisäinen yhteistyö on kriittinen menestystekijä, jotta onnistumiseen päästään. Lähes jokainen haastateltava mainitsi juuri tämän menestystekijän haastattelussaan. Tiimin jäsenten väliseen synergiaan

vaikuttavat paljon halu tehdä yhteistyötä, keskinäinen vuorovaikutus sekä tiimihengen ylläpitäminen.

Yhteistyön rakentamisessa voi mennä aikaa. Se vaatii alkuun sen, että työntekijät tutustuvat toisiinsa. On tärkeää, että työntekijät oppivat ymmärtämään toistensa erilaiset toiminta- ja käytöstavat. Pienenkin huomion osoittaminen voi edistää tiimin sisäistä yhteistyötä. Ollaan kiinnostuneita toisista ja annetaan aikaa yhteisille keskusteluille.

”Meidänhän ei tarvitse toisiamme rakastaa, mutta meidän pitää pystyä tekemään töitä yhdessä ja jos me vielä pidämme toisistamme, niin sehän on aina parempi. Halu tehdä yhteistyötä. Halu auttaa toisia.”

”Siinä mielessä kyl se tiimityöskentely ja sen hiominen, niin kyl se ehdottomasti on olennaista siinä onnistumisessa.”

Hyvä yhteistyö vaatii sen, että hajautetussa tiimissä varmistetaan jokaisen läsnäolo. Tämä siis tarkoittaa sitä, että jokainen jäsen on otettu mukaan. Se vaatii oikeanlaista tiimin koordinoitua, joka on usein Scrum Masterin vastuulla.

Koordinoinniksi voidaan laskea myös se, että pidetään huoli hajautettuun tiimiin kuuluvista kulttuuri- ja aikaeroista. Monessa haastattelussa tuli esille virtuaalitalaisuuden fasilitointi ja sen taitaminen. Sitä haastateltavat kuvasivat osallistamisen ja kulttuurin näkökulmasta.

Etätapaamisessa haasteena on se, että jotkut henkilöt voivat jäädä pimentoon eli he saattavat olla linjalla vain hiljaa. Eräs haastateltava kuvailee haastattelussaan, että on huomannut ihmisten käytöksessä eroja, kun kyseessä on etätapaaminen. Työntekijät ajattelevat, että he ovat vain Skype-yhteydellä mukana, jolloin heidän ei tarvitse kommentoida samalla tavalla kuin kasvokkain tapauksissa. Ongelmana on se, että kaikkien mielipiteet eivät tule kuulluiksi. Mielipiteillä on arvoa. Tällöin se on fasilitointitehtävä, jolloin esimerkiksi Scrum Master pyrkii saamaan jokaisen mukaan.

”Mä sanon, että aika paljon on kiinni siitä, minkälainen Scrum Master ja Product Owner tämmöisellä hajautetulla tiimillä on. Heidän pitää pystyä pitämään näpeissänsä se tiimi. Scrum Masterin osalta mä oon itse kokenut, että noitten kokousten fasilitointi on saada kaikki 'saman pöydän' ääreen. Ja joku huolehtii siitä, jos joku on useamman viikon poissa, niin sille infotaan samoista asioista.”

### **Kommunikointi**

Yhteistyö vaatii paljon kommunikointia, joka myös nousi merkittäväksi kriittiseksi menestystekijäksi. Kommunikointi on erään haastateltavan mukaan erittäin tärkeää muun muassa siksi, että hajautetussa mallissa niin sanottu ”rikkinäinen puhelin” on yleinen ongelma, kuten aiemmin todettiin tämän tutkielman haasteet-osiossa. Sen taklaamiseksi kommunikointia täytyy olla paljon, jotta asiat saadaan esitettyä ja määriteltyä selkeästi.

”Kauheasti puhunut kommunikoinnista. Se varmaan on se kaikista tärkein. Sitä on riittävästi, ja et se kommunikaatio on avointa.”

Erään haastateltavan mukaan kommunikointia ja toisen ymmärtämistä helpottaa paljon se, että näkee toisen henkilön. Videopuhelut ovat merkittävä tekijä kommunikoinnin edistämiseksi.

*”Se on ihan käsittämätön juttu, miten paljon helpompaa siitä puheesta on saada selvää, kun sä näet sen toisen henkilön. Se on ihan uskomatonta.”*

Kuten Misran ym. (2009) artikkelissa todetaan, kommunikoinnilla ja neuvottelutaidolla on oleellinen rooli ketterässä ohjelmistokehityksessä. Tehokas kommunikointi on tunnistettu yhdeksi tärkeimmistä tekijöistä ketterien käytäntöjen onnistumiseksi. Täten yksi ketteryyden tärkeistä vaatimuksista on nopea ja tehokas viestintä kehittäjien, toiminnan, tuen, asiakkaiden ja liiketoiminta-alueiden välillä.

### **Yhteinen suunta ja ymmärrys tavoitteista**

Tiimin jäsenten ymmärrystä ja näkemystä yhteisestä suunnasta ja tavoitteista pidettiin tärkeänä onnistumisen kannalta. Yhteisen suunnan varmistamiseksi tulee tiimin jäsenillä olla tarkka käsitys siitä, miten toteutusta tehdään.

*”Me voidaan tehdä asioita sillein, et kokeillaan, laitetaan testiin ja katsotaan, toimiiko se siellä. Mut, jos tää toinen osa tiimistä ei tiedä tästä, he luottavat siihen, et siellä on nyt jotain todella hyvää ja he rakentavat sen perusteella. Niin sehän saattaa ihan täysin tuhota sen. Sanotaan, että saattaa tuhota tän tiimin yhtenäisyydenkin.”*

Yhteisten tavoitteiden varmistamiseksi niiden dokumentointi on hajautetussa kontekstissa merkityksellistä. Tavoitteet on hyvä olla luettavissa. Eräs haastateltava kokee, että hajautetussa työssä olisi hyvä nähdä tavoitteet paperilla, koska bisnesihmisten tuki ei ole jatkuvasti saatavilla, ja tällöin tavoitteiden tarkentaminen matkan varrella on haasteellisempaa.

### **Asiakasyhteistyö**

Tuloksista nousi selkeästi myös esille se, että asiakasyhteistyö on edellytys ja kriittinen menestystekijä sille, että hajautetussa ketterässä ohjelmistokehitysprojektissa voidaan onnistua. Asiakasyhteistyöllä on vaikutusta aiemmin mainittuun yhteiseen suuntaan ja ymmärrykseen tavoitteista. Lisäksi tällaisessa hajautetussa tiimissä asiakkaan tavoittaminen on tärkeää.

Erään haastateltavan mukaan olisi hyvä, että asiakas olisi lähellä edes yhtä hajautetun tiimin osaa. Tällöin se osa tiimistä pystyy jakamaan tietoa kaikille muille osapuolille. Myös palautteen saaminen on suorempaa, kun tiimi voi näyttää suorituksen tuloksen viikoittain asiakkaalle.

*”Sieltä tulee hyvin suoraa palautetta yleensä, koska bisnesihmiset ajattelee sen rahallisen näkökulman kautta, et mikä tuottaa. Tämä on se asia, mitä IT ei yleensä koskaan näe.”*

*”Niin sekin on myös yks niistä onnistumistekijöistä, että meidän Product Ownerit tietää, mitkä on niitä tärkeimpiä töitä, millä me tehdään se asiakas [loppukäyttäjä] tyytyväiseksi. Me tehdään oikeita asioita oikeaan aikaan. Kyllä se mulla tuossa tiimissä aika hyvin toimii. Mut sekin on mun mielestä tosi tärkeitä, että tällein hajautevasti, kun me ollaan, niin just se et se Product Owner on tavoitettavissa.”*



Tärkeää on, että tuoteomistaja tietää, mitkä ovat niitä oleellisia töitä ja osaa ajoittaa ne järkevästi kehitystiimille. Kuten haastateltava totesi, tuoteomistajan tavoittaminen on äärimmäisen tärkeää erityisesti silloin, kun on kyse hajaute- tusta tiimistä. Silloin sitä voidaan kutsua onnistuneeksi asiakasyhteistyöksi. Yhteistyö tuoteomistajan kanssa on tärkeä edellytys muun tiimin toiminnalle.

”Product Owner edustaa meille asiakasta. On se asiakkaan ääni. Se on kyllä ihan eh- doton asia ja mun mielestä se toimii meillä tosi hyvin, että meidän Product Ownerit on tavoitettavissa. On todella harvoin sellainen tilanne, että tiimiläinen ottaisi mi- nuun yhteyttä ja sanoisi, että hän ei saa Product Owneria kiinni.”

”PO onnistuu allokoidaan omaa aikaansa kaikkiin featureihin yhtä paljon ja tasaises- ti. Ja näkee myös sen, että päästäänkö niihin tavoitteisiin vai pitääkö se feature siirtää ikään kuin toiselle PI:lle.”

### **Henkilökohtaiset ominaisuudet, tekninen pätevyys sekä koulutus ja oppi- minen**

Muita haastatteluista esiin nousseita kriittisiä menestystekijöitä ovat henkilö- kohtaiset ominaisuudet, tekninen pätevyys sekä koulutus ja oppiminen. Kuten aiemmin on todettu, onnistunut tiimin jäsenten välinen yhteistyö vaatii työntekijöiltä tiettyjä ominaisuuksia, kuten yhteistyöhalukkuutta ja kykyä tulla toi- meen toisten kanssa.

”Ehdottomasti se, että meillä on päteviä tekijöitä. Ylipäättänsä tällöisten perustaus- tan ja koulutuksen lisäksi tietynlaiset kognitiiviset kyvyt. Se, että osaako jotain tiettyä asiaa, niin ei haittaa, kun sen voi sitten opetella. Mutta, että on tosi fiksua ja terävää porukkaa oikealla asenteella ja sillä tavalla vielä hyviä tyyppejä, että tulee toimeen keskenään ja kaikilla on hauskaa töissä. Se on ehdottomasti mun mielestä tärkein.”

”Sit on se, että meillä on monen vuoden kokemus. Me tiedetään, mitä jokainen osaa.”

Haastatteluista kävi myös ilmi, että oppimishalu ja kompetenssi auttavat onnis- tumaan projekteissa. Uuden oppimista tapahtuu paljon päivittäisessä työssä, ja se vaatii monen haastateltavan mukaan sen, että työntekijä on oppimishaluinen.

”Koulutus ja oppiminen on asia, mikä mun näkökulmasta kuuluu siihen henkilön omaan parantamiseen tai omaan oppimiseen. Tietenkin se, et hän on oppimishalui- nen, niin huomattavasti helpottaa sitä. Jos projektilla on jotain vaateita tai tiimillä on jotain vaateita, niin se varmasti auttaa sitä.”

”Ei voi olla semmoisia ihmisiä tiimissä, jotka kuvittelee olevansa jo valmiita ja ei tar- vii ikinä enää oppia mitään uutta.”

”Yleensäkin ottaen IT-alalla aina pitää olla halukas oppimaan uusia asioita. Me ol- laan saatu myös ihan koulutuksia kyllä. Mut toki se on hirveästi myöskin sitä itse- opiskelua.”

Tekninen pätevyys ei ole kuitenkaan monenkaan haastateltavan mielestä tär- kein edellytys sille, että projektissa onnistutaan. Yksittäisiä asioita on mahdol-

lista oppia. Tietynlainen älykkyys ja aiemmin mainittu oppimishalu tukevat kokonaisuutta.

”Varsinkin silloin, jos meillä on erittäin spesifi teknologia, niin silloin se on tärkeitä, et meillä on oikeasti joku ammattilainen siinä. Koska jos meidän pitää kehittää sen tiimin kyvykkyys siinä, niin se on iso juttu.”

”Mun mielestä hajautetussakin tiimissä on kuitenkin tilaa näille junnuille. Mut on tärkeitä, että ne on samassa paikassa sen kanssa, kenen kanssa opettelee. Kaikkien ei tarvii olla senior-tasoisia, mutta ehdottomasti sellaisia pitää olla.”

”Vaikka meillä vaihtuu tiimin jäsen, niin kyllä meillä aika hyvä tämä opastus on. Osataan neuvoa, ja uusi jäsen pääsee aika nopeasti kärryille kyllä. Ei tarvii olla ihan ekspertti työskennelläkseen tässä tiimissä.”

Tuloksista huomasi sen, että teknistä pätevyyttä (kompetenssi) ja muita henkilökohtaisia ominaisuuksia, kuten kommunikaatiotaitoja, yhteistyöhalua ja kognitiivisia kykyjä punnittiin onnistumisen kannalta. Kumpi siis on merkittävämpi? Tuloksista voidaan todeta, että henkilökohtaisilla ominaisuuksilla on isompi paino onnistumisessa kuin kompetenssilla, koska jos työntekijällä on oppimishalua, niin tällä on myös mahdollisuus lisätä teknistä pätevyyttään kouluttumalla.

”Varsinkin toi, et jos meillä on henkilö, jolla on hyvät kommunikaatiotaidot ja on ekspertti tässä jossain asiassa niin sehän on jackpot. Niitä henkilöitä on ikävä kyllä erittäin harvassa. Sanotaan, jos meillä on henkilö, jolla on hyvät kompetenssit, mutta huonot kommunikointitaidot... Tää on ehkä vähän vaarallinen kysyä. Vaikka sulla on tiimi täynnä nuoria ja suhteellisen kokemattomia henkilöitä, niin kyllä siitä 'koutsamalla' saa ihan pätevän ja ajalla.”

### **Tiimin kyvykkyys ja johdon tuki**

Haastatteluissa nousi esille tiimin kyvykkyyteen vaikuttavia tekijöitä. Chow'n ja Caon (2008) artikkelissa on todettu, että tiimin kyvykkyyteen liittyy se, että projekteilla on johtajat, jotka tuntevat ketterän prosessin sekä heillä on kevyt tai mukautuva johtamistyyli. Johdon sitoutumisessa pidetään tärkeänä sitä, että ymmärretään, mitä ketterässä ohjelmistokehityksessä tapahtuu.

”Jos johto ei oo sitoutunut siihen, et se projekti pitäisi olla olemassa, niin sillä projektilla ei ole hirveästi mitään toivoa. Ja tiimiläiset sen aavistaakin.”

” [Johdon sitoutuminen] Jos se puuttuu, niin me ollaan huonolla tiellä. Sanotaanko, että mä näen sen niin, että silloin, kun johdon sitoutuminen puuttuu, silloin se johto ei myöskään osaa ohjata niitä tiimejä. Eli silloin siellä tulee vähän sitä semmoista 'juupas eipäs' tai sitten, että puututaan asioihin, kun ei ymmärretä, kun ei haluta antaa tiimille vastuuta. Silloin johto ei oo sitoutunut siihen ketterään menetelmään, jonka mukaan se tiimi itse parhaiten tietää, miten työt pitää tehdä.”

Hajautettu tiimi tarvitsee onnistuakseen sellaisen johdon, joka hyödyntää kevyempää johtamisotetta. Eräänkin haastateltavan mukaan mikromanageroiminen ei ole ketterässä ympäristössä olennaista. Tähän liittyy myös vallitseva kulttuuri, joka kattaa muun muassa sen, että ihmisiin luotetaan. Lisäksi kyseisen haas-

tateltavan tiimi ei ole riippuvainen muista yksiköistä, mikä helpottaa menestymisessä.

”Semmoinen ylemmän johdon tuki, sillä tavalla kuitenkin, että siinä annetaan vapaus toimia. Se ei ole liian mikromanageroitua, mitä helposti just hajautetuissa ympäristöissä tapahtuu. Jos sä alat johtaa sillä Excelillä sitä ylhäältä alaspäin, niin sit ha-joo se malli ihan täysin. Ei se ole enää ketterää ja se ei vaan toimi ollenkaan.”

”Meillä on ollut viime aikoina paljon vapautta toimia. Meille ei ole sanottu tän yksikön ulkopuolelta kauhean selkeästi, et nyt teidän pitää tehdä tämä ja tämä juttu. Vaan meille on annettu kasa rahaa ja näissä asioissa pitäisi tapahtuu hyviä asioita, että saa suorittaa. Semmoinen tietynlainen autonomia ja toisaalta vastuu tuloksista. Me voidaan periaatteessa sopii, et nyt on tämmöinen idea ja siinä porukassa ilman mitään riippuvuuksia ulkopuolelta tehdä se alusta loppuun ja pistää ulos asiakkaille.”

### Seuranta

Monissa haastatteluissa nousi esille niin kutsuttu seuranta, johon lasketaan agilet metriikat. Vaikka päiväkokouksen pitäminen vaati enemmän suunnittelua, se oli monelle kuitenkin sellainen, josta on pidettävä kiinni.

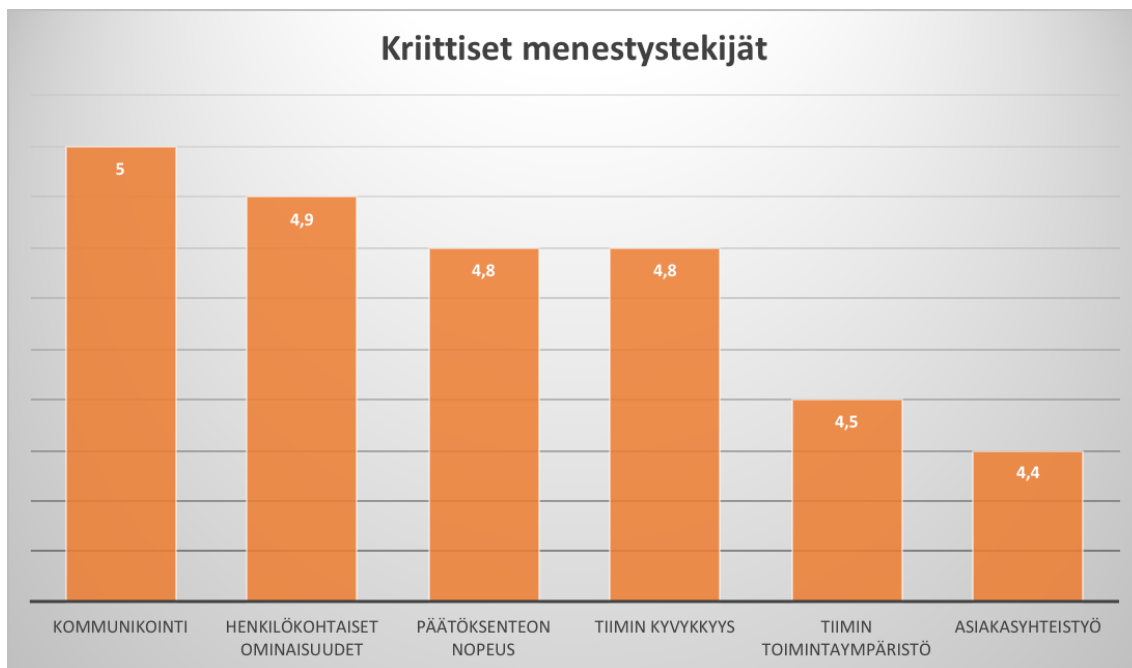
”Seuranta ja nyt tässä iso suuri varoitus, et on hyvää seurantaa ja on erittäin huonoa seurantaa. Metriikka, agilet metriikat, jes iso peukku! Semmoinen mikromanageraus-tason seuranta, ei todellakaan! Siitä ei kyl seuraa mitään hyvää.”

”Tää on yks sellainen asia, jossa mä on suhteellisen tiukka Scrum Master. Mun vaatimustaso, ihan riippumatta, missä henkilö istuu, et dailyissä ollaan aina. Se on semmoinen asia, mistä mä en anna luistaa. Daily-aika on pyhitetty.”

## 6.6 Kysely kirjallisuuden esittämistä menestystekijöistä

Teemahaastatteluissa oli mukana myös pienimuotoinen kysely (LIITE 2), johon oli kerätty tieteellisestä kirjallisuudesta menestystekijöitä. Haastateltavat saivat arvioida seuraavan asteikoin mukaisesti eri menestystekijöitä: *täysin eri mieltä, jokseenkin eri mieltä, ei samaa eikä eri mieltä, jokseenkin samaa mieltä, täysin samaa mieltä tai ei sovelleta / en tiedä*. Haastateltavilla oli mahdollisuus kommentoida vapaamuotoisesti eri menestystekijöitä antamansa arvioinnin lisäksi.

Kyselyyn kerättyjä menestystekijöitä oli yhteensä 24. Kyselyyn otetut menestystekijät ovat peräisin Chow ja Caon (2008) sekä Misran ym. (2009) artikkeleista, joita on myös käsitelty aiemmin tässä tutkielmassa. Kyselyn perusteella kriittisiksi menestystekijöiksi nousivat selkeästi kommunikointi, henkilökohtaiset ominaisuudet, päätöksenteon nopeus, tiimin kyvykkyys, tiimin toimintaympäristö ja asiakasyhteistyö. (KUVIO 8). Kuvioista nähtävät luvut tarkoittavat keskiarvoa. Asteikko määräytyi 1-5.



KUVIO 7 Kyselyn tulokset

Kuviosta voi nähdä, että jokainen haastateltava oli täysin samaa mieltä siitä, että *kommunikointi* on kriittinen menestystekijä. Kommunikointi on lähes poikkeuksetta merkittävin kriittinen menestystekijä hajautetussa ketterässä ohjelmistokehityksessä.

”Mun työ, jos sitä ajatellaan, niin sehän on pelkkää kommunikointia.”

”Se kommunikointi on kaiken A ja O. Se on äärimmäisen tärkeätä. Se vaan, että meiltä puuttuu se kasvokkain.”

Kyselyssä toiseksi merkittävin kriittinen menestystekijä on *henkilökohtaiset ominaisuudet*. Tähän liittyy vahvasti tiimin jäsenen kokemuksen ja pätevyyden lisäksi sellaisia piirteitä, kuten yhteistyökykyinen, vastuuntuntoinen, oppimishaluinen ja halu työskennellä muiden kanssa (Misra ym., 2009). Moni haastateltava kommentoi olevansa täysin samaa mieltä tai pitävänsä sitä erittäin tärkeänä.

Päätöksenteon nopeus ja tiimin kyvykkyys koettiin yhtä tärkeiksi tekijöiksi. Vaikka moni haastateltava pitää *päätöksenteon nopeutta* merkittävänä tekijänä onnistumisen kannalta, niin se koettiin myös haasteeksi. Päätöksenteon nopeuteen liittyy projektin kannalta nopeasti tehtävät päätökset lyhyessä ajassa.

”Joo onhan se tärkeätä, et saada heti tietoa, mut valitettavasti joistakin asioista ei olla saatu selkeää kuvaa.”

*Tiimin kyvykkyys* liittyy siihen, että jäsenet ovat motivoituneita ja sitoutuneita projektin onnistumiseen. Myös projektin johto tuntee ketterät periaatteet sekä käytänteet ja pystyy hyödyntämään sitä johtamisotteessaan. Tiimin kyvykkyyteen liittyy myös asianmukainen tekninen koulutus projektin aihepiiriin ja ketteriin prosesseihin (Chow & Cao, 2008).

”Kyl mä oon täysin samaa mieltä. Aika paljon mekin ollaan nyt pidetty erilaisia koulutuksia ja käyty ihan Scrum-perusteita läpi ja muita. Sekin ehkä nimenomaan hajautetussa mallissa vielä korostuu. On tosi tärkeätä, että ihmiset ymmärtää, miksi me ollaan niissä sprintin planningeissä ja mitä sieltä olisi tavoitteena saada ulos.”

Kyselyssä myös *tiimin toimintaympäristö* koettiin kriittisenä menestystekijänä onnistumiselle. Tiimin toimintaympäristöllä tarkoitetaan sitä, että kehitystiimi työskentelee itseorganisoituvasti sekä tiimi luottaa omaan kykyynsä ratkaista ongelmia autonomisesti. Projektit pyritään tekemään pienellä ryhmällä, eikä niissä olisi useita riippumattomia ryhmiä (Chow & Cao, 2008).

”Kyl mä oon täysin samaa mieltä. Pystytään kommunikoimaan keskenään ja päättää pienistä. Tietenkin on eri taseisia asioita.”

”Kysytään helposti toisiltamme tukea. Tai sitten kysytään joiltain toisilta tiimeiltä tukea. Mun mielestä meidän tiimiläiset on semmoisia, et ne tosiaan aktiivisesti kyselee ja pyytää sitä apua. Niin kuin mä alussa sanoin, se on aika paljon henkilökysymys. Mut meidän tiimi onneksi on semmoinen, et siel on kaikki aktiivisia.”

*Asiakasyhteistyöhön* liittyy vahvasti se, että asiakas tekee läheistä yhteistyötä kehitystiimin kanssa. Asiakas on motivoitunut, aktiivinen ja pitää itseään vastuunalaisena projektin osallistujana (Misra ym., 2009). Asiakasroolit vaihtelevat riippuen siitä, kenelle tiimi on tekemässä. Suurimmalla osalla haastateltavien tiimeistä asiakkaana toimii jokin sisäinen liiketoimintayksikkö, kuten bisnesihmiset, jolloin myös tuoteomistaja voidaan nähdä asiakkaan roolissa. Vain muuttaman haastateltavan tiimi tekee suoraan loppukäyttäjille.

”Meille on hirveän tärkeitä nää asiat, et me käydään itse meidän tuotepäällikön kanssa vastaamassa aina, jos joku antaa meille yhden tähden arvostelun. Ja meidän sovelluksessa on myös tällainen palautekanava, mistä voi lähettää suoraan sähköpostia meille. Se ei mee mihinkään meidän yleiseen asiakaspalveluun, vaan se tulee oikeasti meille, mistä me se luetaan ja pyritään aina myöskin toteuttaa asiakkaitten toiveita mahdollisuuksien puitteissa. Mä koen sen tosi hyväksi asiakasyhteistyöksi.”

”Se on ihan selkeä homma, et sitä tarvitaan. Määrittelythän pitää tulla asiakkaalta. Sitten, kun devaaja alkaa tehdä sitä hommaa, niin saattaa tulla tarkentavia kysymyksiä tai muutenkin jotain kysymyksiä liittyen siihen tekemiseen. Silloin tarvitaan bisneksen kommenttia. Aika harvoin määrittelyt ovat ihan täydellisiä, et tiedetään ihan tasan tarkkaa, mitä pitää tehdä. Useasti tarvitaan lisäkysymyksiä bisnespuolelta.”

## 7 TULOSTEN TULKINTA JA POHDINTA

Tämän tutkielman tavoitteena oli selvittää hajautetun ketterän ohjelmistokehityksen kriittisiä menestystekijöitä ja koettuja haasteita. Eli mitkä ovat ne tekijät, joihin hajautetun ketterän kehitystiimin tulisi keskittyä ja sisällyttää toimintaansa, jotta projekti onnistuisi? Ja mitä haasteita hajautus tuo ketterään ohjelmistokehitykseen? Tässä osiossa käsitellään myös sitä, miten kriittiset menestystekijät ovat suhteessa koettuihin haasteisiin.

Empiirinen osuus toteutettiin laadullisena tutkimuksena. Aineistonkeruu tapahtui teemahaastattelujen ja kyselyn avulla. Tutkimuksen kohteena oli yritys, joka toimii monikansallisten ohjelmistoprojektien parissa.

### 7.1 Tuloksien vertailu aiempiin tutkimuksiin - Haasteet

Tuloksista kävi ilmi pitkä lista haasteita, joita ovat kommunikointi, vähäinen jäsenten välinen tuttavuus, tekniset haasteet, kulttuurierot, kielihaasteet, väärinymmärrykset, riippuvuudet, etäisyys, aikaero, yhteinen ymmärrys ja suunta tavoitteista, informaation kulku ja tiimin sisäinen yhteistyö.

Aiemmissä tutkimuksissa on todettu, ettei ketteriä menetelmiä ja erityisesti ketteriä käytäntöjä voida suoraan soveltaa hajautettuun ohjelmistokehitykseen (Paasivaara ym., 2009). Aiemmat tutkimukset ovat osoittaneet, että hajautettu ketterä ohjelmistokehitys on tuonut esille seuraavanlaisia haasteita: kulttuurilliset ja kielelliset erot, luottamus ja sitoutuminen, tekniset haasteet, viestintää koskevat haasteet, tiedonhallintaan liittyvät kysymykset, tiimien koordinaatio ja tehokas tiedon jakaminen niiden välillä, suunnittelu ilman tietynlaista arkkitehtuuria tai oikein määriteltäviä vaatimuksia (Layman ym., 2006; Paasivaara & Lassenius, 2014; Alzoubi ym., 2016).

Suora viestintä on yksi ketterän toiminnan lähtökohdista. Aiemmat tutkimukset ovat osoittaneet, että viestintää koskevat haasteet on tunnistettavissa hajautettujen tiimien ja asiakkaan välillä (Alzoubi ym. 2016). Tässä tutkielmassa tehdyn tutkimuksen tuloksien perusteella kommunikointiin liittyvät haasteet koskevat sekä hajautetun tiimin sisäistä että ulkopuolista viestintää muiden

hajautettujen tiimien kanssa. Kommunikoinnilla on vaikutusta informaation kulkuun sekä yhteiseen ymmärrykseen ja suuntaan tavoitteista. Tehoton ja epä-säännöllinen viestintä vaihtuvien vaatimusten yhteydessä voi aiheuttaa ongelmia. Tämä löydös tukee aiempaa teoriaa ja täydentää sitä korostaen tiimin sisäistä viestintää ja sen vaikutusta hajautetun tiimin yhteiseen ymmärrykseen ja suuntaan tavoitteista.

Kommunikointiin liittyy paljon tavallista jutustelua jäsenten välillä. Tulosten perusteella tämä vaikuttaa siihen, miten läheisiä tiimin jäsenet ovat. Tavallinen jutustelu on selkeästi vähäisempää hajautetuissa tiimeissä.

Tämän tutkimuksen perusteella voidaan myös vastata aiemmin esitettyyn kysymykseen ”Ovatko maantieteelliset, kulttuurilliset ja ajalliset etäisyydet viestinnän ja yhteistyön keskeisinä esteinä?”. Haastatteluiden perusteella voidaan todeta, että ne eivät ole esteitä, mutta joissakin tiimeissä ne koetaan isoina haasteina ja vaativat sen vuoksi enemmän tiimin koordinoitua ja harjoitusta.

Tuloksista voi päätellä, että kulttuurinen etäisyys tuo lisähaasteita, mikä myös tukee aiempaa teoriaa. Kulttuurierot vaikuttavat käytöstapoihin ja toimintamalleihin. Toiset tiimin jäsenet voivat olla hyvinkin suoraviivaisia ja johdovetoisia, kun taas toiset voivat olla varovaisia kysymyksiensä esittämisessä, päätöksiensä tekemisessä ja vaikeuksien myöntämisessä ääneen. Ajalliset etäisyydet vaativat enemmän tiimin koordinoitua ja dokumentointia. Viestimisessä on huomioitava työajat ja mahdolliset pyhä- ja lomapäivät. Ei ole syytä väheksyä sitä seikkaa, että koko hajautettu tiimi voi kohdata esimerkiksi vain aamupäivisin tiettyyn aikaan. Se rajoittaa hajautettua tiimiä toimimasta itseohjautuvasti, eikä se ei ole kovin joustava tapa työskennellä. Se vaatii myös paljon järjestelyä ja jokaisen huomioimista, jotta kaikki pääsevät mukaan. Maantieteellinen etäisyys aiheuttaa sen, että henkilöiden välinen tuttavuus toisiaan kohtaan on vähäistä, ja tapaamisten sopiminen on välillä haasteellista.

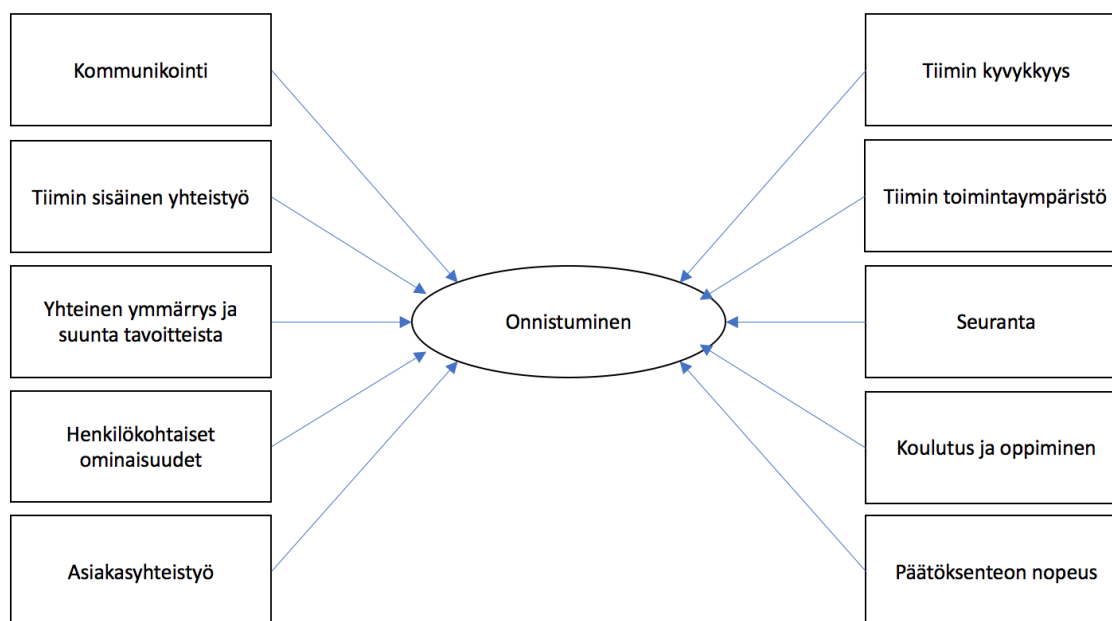
Tekniset haasteet nousivat tutkimustuloksissa voimakkaasti esille. Ne ovat myös yhteydessä muihin haasteisiin tai saattavat jopa tulosten perusteella olla muiden haasteiden aiheuttaja. Yhteyksien katkeaminen on yleinen ongelma, mikä saa aikaan sen, että informaatiosta menetetään paljon.

## 7.2 Tuloksien vertailu aiempiin tutkimuksiin - Kriittiset menestystekijät

Tutkimuksessa selvitettiin haasteiden lisäksi ne kriittiset menestystekijät, joihin tiimin tulisi keskittyä ja sisällyttää toimintaansa, jotta projekti onnistuisi. Tutkimustuloksien perusteella hajautetun ketterän ohjelmistokehitysprojektin onnistumiseen liittyvät asiakastyytyväisyys, tuotantoon vieminen, tavoitteisiin pääseminen, aikataulussa pysyminen ja arvioitujen kustannusten puitteissa toteutettu valmis projekti.

Kahdeksan teemahaastattelun pohjalta esille nousivat seuraavat kriittiset menestystekijät: *kommunikointi, tiimin sisäinen yhteistyö, henkilökohtaiset ominaisuudet, yhteinen ymmärrys ja suunta tavoitteista, asiakasyhteistyö, tiimin kyvykyys, koulutus ja oppiminen sekä seuranta*. Näiden lisäksi kriittisiä menestystekijöitä

ovat kyselyn perusteella *päätöksenteon nopeus* ja *tiimin toimintaympäristö*. (KUVIO 9)



KUVIO 8 Hajautetun ketterän ohjelmistokehityksen kriittiset menestystekijät

Tässä tutkielman osassa tutkimustuloksia verrataan pääosin Chow'n ja Caon (2008) sekä Misran ym. (2009) tutkimuslöydöksiin. Näissä tutkimuksissa ei ole erikseen eroteltu sitä, että kaikki kyselyyn vastanneet tekisivät vain hajautettua ketterää ohjelmistokehitystä. Tähän tutkielmaan liittyvässä tutkimuksessa jokainen haastateltava kuului osaksi hajautettua tiimiä.

Aiempien tutkimuksien tuloksia voidaan pitää yleisenä kehikkona, johon verrataan. Vertaus tehdään sen vuoksi, että hajautettua ketterää ohjelmistokehitystä on tarkoitus toteuttaa pitkälti samalla tavalla kuin kasvotusten tapahtuvaa. Se on kuitenkin enemmän sovellettua. Hajautettu malli on yksittäinen tapa toteuttaa ketterää ohjelmistokehitystä. Tämä antaa mahdollisuuden verrata, mitkä asiat korostuvat hajautetussa ketterässä ohjelmistokehityksessä, mitkä ovat ristiriidassa aiempien tuloksien kanssa tai mitkä ovat uudet lisäykset aiempiin tuloksiin.

Tutkimustulosten vertailua on nähtävissä seuraavassa taulukossa 4. Taulukkoon on nostettu myös Stankovicin ym. (2013) tutkimuslöydökset. Myöhemmin tässä luvussa on otettu huomioon myös muita tutkimuksia ja sieltä nousseita menestykseen vaikuttavia tekijöitä. Näissä tutkimuksissa on ollut laajempi tutkimuskohde, kuten tietojärjestelmäkehitys (engl. information system development, ISD).



TAULUKKO 4 Tutkimustuloksien vertailu aiempiin tutkimuksiin

Ohjelmistokehityksen kriittiset menestystekijät	Chow & Cao (2008)	Misra ym. (2009)	Stankovic ym. (2013)	Tämän tutkimuksen tulokset (2020)
Asiakastyytyväisyys		x		
Asiakasyhteistyö / Asiakkaan osallistuminen	x	x		x
Asiakkaan sitouttaminen		x		
Henkilökohtaiset ominaisuudet		x		x
Ketterät ohjelmistotekniikat	x			
Kommunikointi				x
Koulutus ja oppiminen		x		x
Projektin aikataulu			x	
Projektin luonne			x	
Projektin määrittelyprosessi			x	
Projektinhallintaprosessi	x		x	
Päätöksenteon nopeus		x		x
Seuranta		x		x
Tiimin kyvykkyys	x			x
Tiimin sisäinen yhteistyö				x
Tiimin toimintaympäristö	x			x
Toimitusstrategia	x			
Vallitseva kulttuuri		x		
Yhteinen ymmärrys ja suunta tavoitteista				x
Yrityskulttuuri		x		

Voidaan todeta, että tutkimustulokset ovat osittain linjassa aiempien tutkimuslöydöksiensä kanssa. Samoja tunnistettuja kriittisiä menestystekijöitä ovat tiimin kyvykkyys, tiimin toimintaympäristö, asiakasyhteistyö, päätöksenteon nopeus, henkilökohtaiset ominaisuudet sekä koulutus ja oppiminen. Tuloksissa todetut kriittiset menestystekijät ovat kuitenkin lähimpänä Misran ym. (2009) esittämiä.

Merkittävin ero tämän tutkimuksen ja Misran ym. (2009) esittelemien tutkimustulosten välillä on kommunikointi, joka nousi kärkeen tämän tutkimuksen tuloksissa kriittisimmäksi menestystekijäksi hajautetussa ketterässä ohjelmistokehityksessä. Misran ym. (2009) tuloksissa todettiin, että kommunikointi ja neuvottelukyky on yksi vähiten yhteydessä oleva tekijä menestykseen.

Ketterän tiimin kommunikointi ei ole niin muodollista, vaan se on enemmänkin spontaania, esimerkiksi lyhyitä keskusteluita chatin välityksellä. (Lindsjörn, Sjøberg, Dingsøy, Bergersen ja Dybå, 2016). Monessa haastattelussa nousivatkin esille ad hoc-keskustelut, jotka ovat osa tiimin päivittäistä kommunikointia.

Taipaluksen, Seppäsen ja Pirhosen (2020) teettämän tuoreen tutkimuksen tuloksien perusteella avoin viestintä on sellainen selviytymismekanismi, joka vähentää eri epävarmuustekijöitä tietojärjestelmien kehittämisessä. Avointa viestintää on syytä olla sekä kehitysorganisaation sisällä että asiakasorganisaation kanssa. Se kasvattaa luottamusta, mikä on keskeinen osa onnistumista tietojärjestelmien kehittämisessä. Tutkimuksen mukaan luottamusta lisää se, että pystytään tunnustamaan mahdolliset ongelmakohdat ja epävarmuustekijät (Taipalus, Seppänen & Pirhonen, 2020).

Tämä tutkimus on toteutettu monikansallisessa ohjelmistoyrityksessä, joka tekee uusia tuotoksia niin sisäiselle bisnekselle kuin loppukäyttäjille. Asiakkaan rooli vaihtelee hajautetuissa tiimeissä. Saumaton asiakasyhteistyö on olennaista, jotta ketterissä projekteissa voidaan menestyä.

Asiakasyhteistyön voidaan sanoa olevan merkittävä tekijä onnistumisessa kaikenlaisessa tietojärjestelmäkehityksessä. Taipaluksen ym. (2020) tutkimus osoittaa, että läheinen yhteistyö asiakkaan kanssa sekä projektin alussa että sen aikana, on todella tärkeää tarpeettoman työn ja psykologisten ongelmien välttämiseksi. Se tarkoittaa sitä, että kehitystiimillä ja asiakkaalla on oltava tarpeeksi aikaa ensimmäisten vaatimusten esittelylle. Kehitystiimin on ymmärrettävä, mitä toimintoja heiltä odotetaan kehitettävän. Sen vuoksi asiakkaan on selitettävä myös liiketoiminta-alue kehitystiimille, jotta tiimi ymmärtäisi myös, miksi toimintoa tarvitaan.

Tämän jälkeen kehitystiimin on kartoitettava järjestelmän pääasialliset loppukäyttäjärühmät. Heidän tulisi olla säännöllisesti mukana kehityksen eri vaiheissa, kuten osallistumisessa suunnitteluun (Taipalus ym., 2020).

Tässä tutkimuksessa löydettiin lisää kriittisiä menestystekijöitä aiempien tutkimustuloksien rinnalle. Näitä ovat tiimin sisäinen yhteistyö sekä tiimin yhteinen ymmärrys ja suunta tavoitteista. Tiimin sisäisen yhteistyön rakentaminen vaatii riittävää kommunikointia jäsenten välillä sekä ymmärrystä toisten erilaisista toiminta- ja käytöstavoista.

Myös Lindsjörnin ym. (2016) artikkelissa korostetaan sitä, että muutosten reagoimiseksi tiimin jäsenten tulisi työskennellä tiiviimmin yhdessä, kommunikoida useammin ja olla tietoisia muiden tiimin jäsenten työskentelystä sekä kyetä jakamaan työmäärä henkilöiden välillä.

Tiimin yhteiseen ymmärrykseen ja suuntaan tavoitteista liittyy tarkka käsitys siitä, mitä ollaan tekemässä, ja miten toteutusta tehdään. Tämä vaatii oikeanlaista koordinointia, jonka on osoitettu olevan Stroden, Huffin, Hopen ja Linkin (2012) mukaan kriittinen menestystekijä onnistuneissa ohjelmistoprojekteissa. Koordinoinnin varmistamiseksi ketterän tiimin toiminnan tulisi sisältää tietyt toimet, jotka edistävät projektin ja tavoitteen yhteistä ymmärrystä (Strode, Huff, Hope & Link, 2012).

Tässä tehdyn tutkimuksen tuloksien perusteella yhteisen suunnan ja tavoitteen varmistamiseen liittyy vahvasti myös riittävä dokumentointi ja seuranta, eli eri metriikoilla seurataan ketterän tiimin toiminnan kehittymistä. Nämä metriikat, kuten päivittäinen palaveri (engl. daily), ad hoc-keskustelut ja jokaisen iteraation alussa pidettävä suunnittelutapahtuma ovat Stroden ym. (2012) mukaan juuri niitä toimintoja, joita koordinoinnin tulisi sisältää (Strode, Huff, Hope & Link, 2012).

### **7.3 Hajautetun ketterän ohjelmistokehityksen kriittiset menestystekijät suhteessa koettuihin haasteisiin**

Tutkimustulokset osoittavat, että osa tunnistetuista kriittisistä menestystekijöistä koetaan myös haasteina. Kommunikointi on selkeästi kriittisin menestystekijä

niin haastatteluiden kuin kyselyn perusteella. Toimiakseen tehokkaasti hajautettu tiimi tarvitsee paljon kommunikointia jäsenten välillä.

Kommunikointiin voidaan yhdistää seuraavia haasteita: tiimin jäsenten välinen vähäinen tuttavuus, väärinymmärrykset, tiimin sisäinen yhteistyö ja informaation kulku. Mitä hajautettu tiimi tarvitsee onnistuneeseen kommunikointiin? Toimivia teknisiä ratkaisuja, kuten katkeilemattomat videoyhteydet. Tekniset haasteet ovat selkeästi iso ongelma kohdeorganisaatiossa. Huonot yhteydet johtavat katkeilevaan informaatioon, mikä taas voi aiheuttaa väärinymmärryksiä. Työntekijöiden videokeskustelut voivat myös auttaa ymmärtämään toista paremmin sekä vähentämään kielihaasteita, kuten eräs haastateltavista totesi.

Tiimin sisäistä toimivaa yhteistyötä pidetään merkittävänä tekijänä onnistumisessa. Se koetaan myös haasteena johtuen esimerkiksi siitä, että tiimi toimii hajautetusti, ja liiallinen dokumentointi ei kuulu ketterään toimintaan. Ohjelmistokehitystä voidaan pitää luovana aktiviteettina. Kuten eräs haastateltavista kuvaili, se on kuin maalausta tekisi. Luova aktiviteetti vaatii paljon yhteistyötä.

Hajautetussa ketterässä ohjelmistokehityksessä yhteinen ymmärrys ja suunta siitä, mitä ollaan tekemässä ja mitä kohti menossa, voi vaatia enemmän dokumentointia. Jotakin pitää olla mustaa valkoisella. Ei ole myöskään tarkoituksenmukaista se, että vain osa tiimistä tietää, mitä tehdään, ja muut seuraavat perässä ja uskovat hienoon lopputulokseen. Onnistunut projekti vaatii kaikkien osallistumista, saumatonta yhteistyötä sekä ymmärrystä yhteisestä suunnasta ja tavoitteesta.

Hyvä yhteistyö edellyttää työntekijöiltä sellaisia ominaisuuksia kuin yhteistyökyky ja halu työskennellä muiden kanssa. Ilman näitä henkilökohtaisia ominaisuuksia hajautetussa tiimissä voidaan kohdata paljon vaikeuksia. Eri kulttuureista saapuvat henkilöt on osattava ottaa huomioon, ja on ymmärrettävää, etteivät kaikki toimi tavalla, johon on itse tottunut. Kulttuurierot koetaan haasteena hajautetussa ketterässä ohjelmistokehityksessä, minkä vuoksi yhteistyökyky on äärimmäisen tärkeää. Myös rehellisyys kuuluu merkittävänä osana henkilökohtaisiin ominaisuuksiin, mitä toivotaan jokaiselta työntekijältä. Rehellisyyden lisäksi kyky myöntää epätietoisuus sekä uskallus pyytää apua helpottavat tiimin yhteistä toimintaa.

Etäisyys haasteena aiheuttaa myös sen, ettei välttämättä tiedetä muuta kuin, mitä "linjoja" pitkin kuullaan. Sen vuoksi seuranta ketterillä metriikoilla auttaa varmistamaan ajantasaisen tiedon ja työntekijöiden tilanteen. Säännölliset seremoniat päivä-, viikko- ja kuukausitasolla ovat olennaisia. Niillä voidaan myös pitää huoli tarvittavasta kommunikoinnin määrästä ja varmistaa, että kaikilla on samat tiedot.

Haastatteluista kävi ilmi, että riippuvuudet muista tiimeistä ovat tuoneet oman haasteensa. Ketterässä ohjelmistokehityksessä tuetaan yleensä projekteja, joissa ei ole useita riippumattomia ryhmiä. Kuitenkin organisaatiossa SAFemallin myötä "junassa" voi olla useampi tiimi. Hajautetuissa tiimeissä uskotaan itseorganisoiutuvaan työskentelyyn, mutta se voi vaikeutua, jos riippuvuuksia on liikaa. Saumatonta asiakasyhteistyötä, tavoitteiden määrittäminen sekä yhteinen ymmärrys ja suunta tavoitteista helpottavat riippuvuuksien välistä yhteistyötä.

Hajautettu ketterä ohjelmistokehitys on mahdollista vaikeuksista huolimatta. Se vaatii paljon soveltamista ja organisointia. Normaalisti jaettavat työtehtävät lokaalitiimissä eivät välttämättä toimi hajautetussa tiimissä. Eri tehtävien painottaminen eri toimipisteisiin tiettyjen henkilöiden vastuulle voi olla tarpeen. Silloin kommunikoinnin määrä pystytään pitämään riittävänä. Onko tämä kuitenkin ketteryyden tarkoitus? Hyödynnetäänkö koko käytettävissä oleva kapasiteetti ja tarjolla oleva kompetenssi? Hajautettu ketterä ohjelmistokehitys perustuu ihmisiin ja heidän väliseen toimintaansa sekä teknisiin resursseihin.

Tämä tutkimus osoittaa todeksi sen, mitä on epäilty. Ketterä ohjelmistokehittäminen hajautetussa kontekstissa on haasteellista. Moni hajautetun ketterän ohjelmistokehityksen kriittinen menestystekijä koetaan myös haasteena, minkä vuoksi se edellyttää jatkotutkimusta.

## 7.4 Tutkimuksen luotettavuus ja rajoitteet

Tämän tutkimuksen tarkoituksena oli lisätä ymmärrystä siitä, mitä hajautettu ketterä ohjelmistokehitys tarkoittaa, ja mitkä ovat sen kriittiset menestystekijät suhteessa koettuihin haasteisiin. Aineistonkeruumenetelmiksi valikoituivat teemahaastattelu ja kysely, jotta saataisiin syvällisempää ymmärrystä aiheesta. Tämä tukee mahdollista jatkossa tehtävää laajempaa määrällistä tutkimusta, jossa voidaan selvittää lisää hajautetun ketterän ohjelmistokehityksen kriittisiä menestystekijöitä.

Aihetta lähdettiin valottamaan kirjallisuuskatsauksella. Tutkimuksen kannalta oli tärkeää tutustua siihen, miten ketterä ohjelmistokehitys on määritelty tieteellisessä kirjallisuudessa, sekä millaista tutkimusta on tehty hajautetusta ketterästä ohjelmistokehityksestä. Tämän lisäksi tuli tutustua siihen, mitä kriittisiä menestystekijöitä on jo todettu aiemmissa tutkimuksissa. Tunnetuimmat tutkimukset sijoittuvat noin kymmenen vuoden taakse, mikä myös tarkoittaa sitä, että alalla on tapahtunut kehitystä, ja asiat ovat voineet muuttua.

Kirjallisuuskatsaus pyrittiin tekemään mahdollisimman kattavasti resursseihin nähden. Se antaa lukijalle lisää ymmärrystä sekä johdattelee empiiriseen tutkimukseen.

Aineistonkeruumenetelmän valinta oli resursseihin nähden järkevä. Ennen varsinaisia haastatteluita, haastattelurunko käytiin harjoitusmielessä mentorin kanssa läpi. Sen myötä haastattelurunkoa vielä tarkennettiin tutkittavaan aiheeseen sopivaksi. Haastattelut toteutettiin kahden viikon aikana. Jokainen haastattelu pidettiin kasvotusten ja ne nauhoitettiin litterointia ja jälkiselvittelyä varten.

Tutkimusaineisto analysoitiin suunnatun sisällönanalyysin avulla. Aiempaan teoriaan ja tutkimukseen perustuen teemahaastattelun haastattelurungolle muodostettiin avainteemat. Näiden teemojen valossa alettiin analysoida tutkimusaineistoa.

Tutkija ei tuntenut ennestään haastateltavia. Lisäksi tutkijan vähäinen kokemus ketterästä ohjelmistokehityksestä minimoi ennakkoluulojen vaikutuksen

tuloksiin tutkimusaineistoa analysoitaessa. Hsiehin ja Shannonin (2005) mukaan suunnatun sisällönanalyysin heikkoutena on myös se, että teoriaan hyvin perehtynyt tutkija saattaa lähestyä tutkimusaineistoa puolueellisesti. Tämä johtaa usein siihen, että tutkimustulokset tukevat pikemminkin teoriaa kuin olisivat ristiriidassa sen kanssa. Tätä rajoitetta pyrittiin minimoimaan haastatteluiden nauhoittamisella ja litteroinnilla. Haastatteluissa käytettiin avoimia kysymyksiä, jotta vältettiin vastauksien liiallista rajaamista. Tutkimuksen läpinäkyvyyttä ja luotettavuutta tuettiin käyttämällä haastateltavien suoria lainauksia tutkielmassa.

Tutkimuksen luotettavuutta arvioidaan tavallisimmin reliabiliteetin ja validiteetin näkökulmasta. Hirsjärven ja Hurmen (2008) mukaan aineiston luotettavuus riippuu sen laadusta. Reliabiliteetin todentaminen on laadullisessa tutkimuksessa ongelmallisempi kuin määrällisessä tutkimuksessa, koska tutkimuksen keskeiset vaiheet eivät ole niin yksinkertaisesti toistettavissa.

Reliaabelius eli mittaustulosten toistettavuus perustuu ajatukseen siitä, että haastateltaessa samaa henkilöä saadaan kahdella tutkimuskerralla sama tulos. On kuitenkin otettava huomioon se, että ihmiselle on ominaista ajassa tapahtuva muutos. Toinen tapa reliaabeliuksen määrittelyssä on se, että tulos on reliaabeli. Se tarkoittaa sitä, että kaksi arvioitsijaa päätyy samanlaiseen tulokseen. On kuitenkin huomioitava se, että jokainen yksilö tekee omien kokemustensa perusteella tutkittavasta kohteesta oman tulkintansa (Hirsjärvi & Hurme, 2008, s. 185-186). Tässä tutkielmassa on pyritty kertomaan avoimesti ja tarkasti, kuinka tutkimusaineistoa on kerätty, ja kuinka sitä on käsitelty ja analysoitu.

Kolmas tapa ymmärtää reliaabelius liittyy siihen, että kahdella rinnakkaisella tutkimusmenetelmällä saadaan sama tulos. On kuitenkin epätodennäköistä, että kahdella menetelmällä päädytään täsmälleen samaan tulokseen, mikä johtuu muun muassa ihmisen käyttäytymisen riippuvuudesta kontekstiin. Tässä tutkimuksessa käytettyjen aineistonkeruumenetelmien (teemahaastattelu ja kysely) tuloksissa oli nähtävissä yhteneväisyyttä. Kysely toimi ennemminkin täydentävänä menetelmänä haastatteluille.

Voidaan todeta, että reliabiliteetti ottaa kantaa tuloksien toistettavuuteen ja siihen, voidaanko olettaa, että tulokset ovat samat, vaikka tutkimus toistettaisiin, tutkija vaihdettaisiin tai toista menetelmää käytettäisiin (Hirsjärvi & Hurme, 2008, s. 185-186).

Toinen tunnettu tutkimuksen arviointiin liittyvä käsite on validius eli pätevyys. Validiteetista puhuttaessa on kyse siitä, onko tutkimus pätevä; onko se perusteellisesti tehty, ovatko saadut tulokset ja tehdyt päätelmät ”oikeita” (Saaranen-Kauppinen & Puusniekka, 2006). Validius perustuu tutkimusmenetelmän kykyyn mitata juuri sitä, mitä on tarkoituskina (Hirsjärvi ym., 2009, s. 231). Validius on Hirsjärven ja Hurmen (2008) mukaan tullut tutuksi kvantitatiivisesta tutkimuksesta.

Tutkimustulokset ovat linjassa myös aiempien tutkimuksien kanssa. Tutkimuksessa tuli esille myös uusia näkökulmia ja eroavaisuuksia aiempiin tutkimuksiin nähden.

Tämä tutkimus rajattiin yhteen monikansalliseen ohjelmistoyritykseen resursseista johtuen. Sen vuoksi se antaa vain yhden näkökulman tutkittavaan aiheeseen. Kohdeorganisaation koko, sisäinen kulttuuri ja johtamismalli voivat

vaikuttaa tuloksiin. Lisäksi ketterästä ohjelmistokehityksestä saatu vuosien kokemus voi vaikuttaa tuloksiin. Kertynyt kokemus vaihteli haastateltavien kesken huomattavasti, mikä taas on voinut vaikuttaa heidän näkemyksiinsä ja kokemuksiinsa. Osalla haastateltavista on kokemusta sekä lokaalissa että hajautetussa tiimissä työskentelystä. Se on voinut antaa haastateltavalle laajempaa näkökulmaa näiden kahden työskentelytavan eroihin.

## 7.5 Jatkotutkimusaiheita

Hajautettu ketterä ohjelmistokehitys on aiheena laaja, minkä vuoksi tässä tutkielmassa tutkimusotanta on resursseista johtuen suhteellisen pieni. Hajautetusta työstä ja ulkoistamisesta löytyy tutkimusta yli 20 vuoden takaa. Myös ketterän ohjelmistokehityksen tutkimuksen ruuhkaisimmat vuodet sijoittuvat noin kymmenen vuoden taakse. Ketterä ohjelmistokehitys on silti suhteellisen uutta monille yrityksille.

Aiemmat hajautetun ketterän ohjelmistokehityksen tutkimukset liittyvät pitkälti haasteiden tunnistamiseen. Kuten on todettu ketterät menetelmät ja erityisesti ketterät käytännöt eivät sovellu suoraan hajautettuun ohjelmistokehitykseen. Sitä kuitenkin toteutetaan muun muassa resurssi- ja kansainvälistymissyistä. Ketterä ohjelmistokehitys hajautetussa ympäristössä on kehittyvää ja monelle yritykselle nykyinen linjaus. Sen vuoksi menestystekijöiden tunnistaminen on oleellista. Niiden sisällyttäminen hajautetun tiimin toimintaan auttaa onnistumaan ohjelmistoprojekteissa. Lisäksi se auttaa taklaamaan erilaisia ongelmakohtia ja haasteita.

Jatkotutkimukset voisivat liittyä saman aihepiirin tutkimiseen syvällisemmin. Hajautetun ketterän ohjelmistokehityksen kriittisten menestystekijöiden ja haasteiden välinen tutkimusalue vaatii edelleen syventymistä.

Laajempi laadullinen tutkimus eri alan yrityksissä ja useampi haastateltava voisivat auttaa löytämään lisää mahdollisia menestystekijöitä. Sen jatkoksi voisi toteuttaa laajan määrällisen kyselyn uusien kriittisten menestystekijöiden tunnistamiseksi. Jatkotutkimuksissa olisi hyvä olla mukana vielä laajemmin maantieteellisesti hajautettuja tiimejä sekä eri johtamismalleilla toimivia yrityksiä.

Tässä tutkimuksessa nousi esille paljon tiimin toimintaan liittyviä tekijöitä. Näiden syvällisempi tutkiminen tiimin näkökulmasta voisi auttaa ymmärtämään lisää koettuja haasteita. Kun mahdolliset sudenkuopat on tunnistettu, niiden taklaaminen oikeilla menestystekijöillä mahdollistaa onnistumisen.

## 8 YHTEENVETO

Tässä tutkielmassa tutkittiin hajautettua ketterää ohjelmistokehitystä sekä sen kriittisiä menestystekijöitä ja koettuja haasteita. Tutkimuskohteena oli yritys, joka toimii monikansallisten ohjelmistokehitysprojektien parissa. Tutkimusaihe oli ajankohtainen kohdeyritykselle, koska hajautettua ketterää ohjelmistokehitystä pyritään siellä kehittämään jouhevammaksi, ja kyseinen työskentelytapa on osoittautunut merkittäväksi tavaksi tehdä ohjelmistokehitystä. Tässä tutkimuksessa tutkimuskysymyksenä käytettiin:

Mitkä ovat hajautetun ketterän ohjelmistokehityksen kriittiset menestystekijät ja miten ne suhteutuvat koettuihin haasteisiin?

Tavallisimmin ketterän ohjelmistokehityksen määrittelyssä käytetään Agile Manifeston julkaisua. Sen 12 periaatetta näkyvät eri ketterien menetelmien toiminnassa. Hajautettua työtä ja ulkoistamista alettiin tutkia yli 20 vuotta sitten. Tietotekniikan merkitys on vahvistunut organisaatioissa. Yritykset keskittyvät nyt enemmän ydinosaamiseen ja ne ovat luopuneet monipuolistamisstrategioistaan. Ulkoistaminen on mahdollistanut kustannussäästöt ja pääsyn suurempiin työvoimavaroihin.

Tutkimuksen keskeisten tulosten selvittämiseksi oli tärkeää ymmärtää, mitä hajautetusta ketterästä ohjelmistokehityksestä jo tiedetään empiirisesti, kuinka saavuttaa ketteryys hajautetussa ohjelmistokehityksessä sekä mitä aiempi tutkimus sanoo ketterän ohjelmistokehityksen onnistumisesta ja kriittisistä menestystekijöistä. Osakysymysten vastaamisessa käytettiin tutkimusmenetelmänä kirjallisuuskatsausta. Lähteitä kerättiin ja analysoitiin vastauksen selvittämiseksi. Tässä tutkielmassa käytetty lähdeaineisto koostuu tieteellisistä artikkeleista, kirjallaisista sekä dokumentoiduista tapaustutkimuksista.

Ketterät periaatteet pyrkivät epävirallisiin prosesseihin koordinoinnin helpottamiseksi, ja hajautettu ohjelmistokehitys on riippuvainen tyypillisesti muodollisista toimintaperiaatteista. Näiden lähtökohtien yhdistäminen tuottaa erilaisia haasteita, jotka liittyvät tavallisimmin maantieteellisen, kulttuurisen ja ajallisen etäisyyden tuomiin esteisiin viestinnässä ja yhteistyössä.

Hajautetussa ketterässä ohjelmistokehityksessä on koettu monia haasteita, joista kommunikointi nousi eri lähteissä esille. Myös tämän tutkielman laadullinen tutkimus osoitti, että kommunikointi on suuri haaste.

Päätutkimuskysymykseen vastaamisessa käytettiin empiiristä tutkimusta. Aineiston kerääminen toteutettiin laadullisena tutkimuksena, mikä sisälsi kahdeksan teemahaastattelua ja kyselyn. Tutkimusaineiston käsittelyssä käytettiin suunnattua sisällönanalyysia. Empiirisen tutkimuksen perusteella tunnistettiin monia haasteita sekä kymmenen kriittistä menestystekijää. Haasteisiin kuuluvat kommunikointi, vähäinen työntekijöiden välinen tuttavuus, tekniset haasteet, kulttuurierot, kielihaasteet, väärinymmärrykset, riippuvuudet, etäisyys, aikaero, yhteinen ymmärrys ja suunta tavoitteista, informaation kulku ja tiimin sisäinen yhteistyö.

Teemahaastatteluiden ja kyselyn perusteella tunnistettiin seuraavat kriittiset menestystekijät: kommunikointi, tiimin sisäinen yhteistyö, henkilökohtaiset ominaisuudet, yhteinen ymmärrys ja suunta tavoitteista, asiakasyhteistyö, tiimin kyvykkyys, koulutus ja oppiminen, seuranta, päätöksenteon nopeus ja tiimin toimintaympäristö. Tuloksista huomasi, miten moni kriittinen menestystekijä koettiin ja nähtiin myös haasteena. Kommunikointi ja tiimin jäsenten välinen yhteistyö hajautetussa työssä vaatii selkeästi paljon panostusta, harjoitusta ja oikeat ominaisuudet omaavia henkilöitä. Tekninen tuki on merkittävässä osassa, jotta tiimit pystyvät työskentelemään mutkattomasti yhdessä ohjelmistoprojektien parissa. Tekniset haasteet vaikuttavat olevan kuitenkin enemmän sääntö kuin poikkeus. Tämä näkyi monien haastateltavien kohdalla suurena pettymyksenä ja tietynlaisena luottamuksen menetyksenä.

Empiirinen tutkimus on linjassa aiempien tutkimuksien kanssa. Tuloksista nousi esille monia haasteita, jotka liittyvät ihmisiin ja heidän välisiin suhteisiin ja toimintaansa. Myös kriittiset menestystekijät painottuivat paljolti ihmistekijöihin.

Moni ketteryuden vaikuttava tekijä lähtee liikkeelle asianmukaisten resurssien valitsemisesta ja järjestämisestä projektiin. Ihmisiin ja teknologiaan perustuvaa ketteryttä pidetään merkittävänä tekijänä, mikä nousee esille eri haasteiden voittamisessa ketteryuden ylläpitämiseksi hajautetussa ohjelmistokehitysprojektissa. Täten kohdeorganisaation tulisi ottaa tämä asia huomioon ja järjestää se asianmukaiselle tasolle.



## LÄHTEET

- Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile software development methods: Review and analysis. arXiv preprint arXiv:1709.08439.
- Ahimbisibwe, A., Cavana, R. Y., & Daellenbach, U. (2015). A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies. *Journal of Enterprise Information Management*, 28(1), 7-33.
- Alzoubi, Y. I., Gill, A. Q., & Al-Ani, A. (2016). Empirical studies of geographically distributed agile development communication challenges: A systematic review. *Information & Management*, 53(1), 22-37.
- Bavani, R. (2009, December). Critical success factors in distributed agile for outsourced product development. In *Proceedings of CONSEG-09: International Conference on Software Engineering* (pp. 75-79).
- Beck, K. (1999). Embracing change with extreme programming. *Computer*, 32(10), 70-77.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). Manifesto for agile software development.
- Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The case research strategy in studies of information systems. *MIS quarterly*, 369-386.
- Cadle, J., & Yeates, D. (Eds.). (2008). *Project management for information systems* (5th ed.). Pearson education.
- Carmel, E., & Agarwal, R. (2001). Tactical approaches for alleviating distance in global software development. *IEEE software*, 18(2), 22-29.
- Chow, T., & Cao, D. B. (2008). A survey study of critical success factors in agile software projects. *Journal of systems and software*, 81(6), 961-971.
- Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information systems research*, 20(3), 329-354.
- Darke, P., Shanks, G., & Broadbent, M. (1998). Successfully completing case study research: combining rigour, relevance and pragmatism. *Information systems journal*, 8(4), 273-289.

- Dibbern, J., Goles, T., Hirschheim, R., & Jayatilaka, B. (2004). Information systems outsourcing: a survey and analysis of the literature. *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, 35(4), 6-102.
- Dingsøy, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development.
- Dorairaj, S., Noble, J., & Malik, P. (2012, August). Knowledge management in distributed agile software development. In *2012 Agile Conference* (pp. 64-73). IEEE.
- Erickson, J., Lyytinen, K., & Siau, K. (2005). Agile modeling, agile software development, and extreme programming: the state of research. *Journal of Database Management (JDM)*, 16(4), 88-100.
- Estler, H. C., Nordio, M., Furia, C. A., Meyer, B., & Schneider, J. (2014). Agile vs. structured distributed software development: A case study. *Empirical Software Engineering*, 19(5), 1197-1224.
- Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems*, 15(2), 200-213.
- Hirsjärvi, S. & Hurme, H. (2008). *Tutkimushaastattelu: teemahaastattelun teoria ja käytäntö*. Helsinki: Gaudeamus.
- Hirsjärvi, S., Remes, P. & Sajavaara, P. (2009). *Tutki ja kirjoita*. (15. uud. painos). Helsinki: Tammi.
- Holmstrom, H., Conchúir, E. Ó., Agerfalk, J., & Fitzgerald, B. (2006, October). Global software development challenges: A case study on temporal, geographical and socio-cultural distance. In *2006 IEEE International Conference on Global Software Engineering (ICGSE'06)* (pp. 3-11). IEEE.
- Hsieh, H. F., & Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative health research*, 15(9), 1277-1288.
- Ikonen, M., Pirinen, E., Fagerholm, F., Kettunen, P., & Abrahamsson, P. (2011, April). On the impact of kanban on software project work: An empirical case study investigation. In *2011 16th IEEE International Conference on Engineering of Complex Computer Systems* (pp. 305-314). IEEE.
- Korkala, M., & Abrahamsson, P. (2007, August). Communication in distributed agile development: A case study. In *33rd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO 2007)* (pp. 203-210). IEEE.

- Korkala, M., & Maurer, F. (2014). Waste identification as the means for improving communication in globally distributed agile software development. *Journal of Systems and Software*, 95, 122-140.
- Kotlarsky, J., & Oshri, I. (2005). Social ties, knowledge sharing and successful collaboration in globally distributed system development projects. *European Journal of Information Systems*, 14(1), 37-48.
- Layman, L., Williams, L., Damian, D., & Bures, H. (2006). Essential communication practices for Extreme Programming in a global software development team. *Information and software technology*, 48(9), 781-794.
- Lee, G., & Xia, W. (2010). Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly*, 34(1).
- Lindsjörn, Y., Sjöberg, D. I., Dingsøy, T., Bergersen, G. R., & Dybå, T. (2016). Teamwork quality and project success in software development: A survey of agile development teams. *Journal of Systems and Software*, 122, 274-286.
- Lindstrom, L., & Jeffries, R. (2004). Extreme programming and agile software development methodologies. *Information systems management*, 21(3), 41-52.
- Lyytinen, K., & Rose, G. M. (2006). Information system development agility as organizational learning. *European Journal of Information Systems*, 15(2), 183-199.
- Matharu, G. S., Mishra, A., Singh, H., & Upadhyay, P. (2015). Empirical study of agile software development methodologies: A comparative analysis. *ACM SIGSOFT Software Engineering Notes*, 40(1), 1-6.
- 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.
- Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and organization*, 17(1), 2-26.
- Olsson, H. H., Conchúir, E. Ó., Ågerfalk, P. J., & Fitzgerald, B. (2008). Two-stage offshoring: An investigation of the Irish bridge. *MIS quarterly*, 257-279.
- Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2009, July). Using scrum in distributed agile development: A multiple case study. In 2009 Fourth IEEE

International Conference on Global Software Engineering (pp. 195-204). IEEE.

- Paasivaara, M., & Lassenius, C. (2014). Communities of practice in a large distributed agile software development organization—Case Ericsson. *Information and Software Technology*, 56(12), 1556-1577.
- Ramesh, B., Cao, L., Mohan, K., & Xu, P. (2006). Can distributed software development be agile?. *Communications of the ACM*, 49(10), 41-46.
- Rising, L., & Janoff, N. S. (2000). The Scrum software development process for small teams. *IEEE software*, 17(4), 26-32.
- Saaranen-Kauppinen, A., & Puusniekka, A. (2006). KvaliMOTV - Menetelmäopetuksen tietovaranto [verkkójulkaisu]. Tampere: Yhteiskuntatieteellinen tietoarkisto [ylläpitäjä ja tuottaja]. <<https://www.fsd.uta.fi/menetelmaopetus/>>. (Haettu 9.6.2020.)
- Sadath, L., Karim, K., & Gill, S. (2018, June). Extreme programming implementation in academia for software engineering sustainability. In 2018 Advances in Science and Engineering Technology International Conferences (ASET) (pp. 1-6). IEEE.
- SaFe. (2020). Welcome to Scaled Agile Framework® 5.0! Haettu 9.8.2020 osoitteesta <https://www.scaledagileframework.com/about/>
- Sarker, S., Munson, C. L., Sarker, S., & Chakraborty, S. (2009). Assessing the relative contribution of the facets of agility to distributed systems development success: an Analytic Hierarchy Process approach. *European Journal of Information Systems*, 18(4), 285-299.
- Sarker, S., & Sarker, S. (2009). Exploring agility in distributed information systems development teams: An interpretive study in an offshoring context. *Information Systems Research*, 20(3), 440-461.
- Schwaber, K. & Sutherland, J. (2017). Scrum Guide. Haettu 21.5.2020 osoitteesta <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>
- Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017, May). SCRUM model for agile methodology. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 864-869). IEEE.
- Shrivastava, S. V., & Rathod, U. (2015). Categorization of risk factors for distributed agile projects. *Information and Software Technology*, 58, 373-387.

- Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D. B. (2013). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *Journal of Systems and Software*, 86(6), 1663-1678.
- Strode, D. E., Huff, S. L., Hope, B., & Link, S. (2012). Coordination in co-located agile software development projects. *Journal of Systems and Software*, 85(6), 1222-1238.
- Sutherland, J., & Schwaber, K. (2007). *The Scrum Papers. Nuts, Bolts and Origins of an Agile Process.*
- Taipalus, T., Seppänen, V., & Pirhonen, M. (2020). Uncertainty in information system development: Causes, effects, and coping mechanisms. *Journal of Systems and Software*, 110655.
- Vlaar, P. W., van Fenema, P. C., & Tiwari, V. (2008). Cocreating understanding and value in distributed work: How members of onsite and offshore vendor teams give, make, demand, and break sense. *MIS Quarterly*, 32(2), 227-255.
- VersionOne. (2019). 13<sup>th</sup> Annual State of Agile Report. Haettu 21.5.2020 osoitteesta <https://stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508>
- Williams, L., & Cockburn, A. (2003). Agile software development: it's about feedback and change. *IEEE Computer*, 36(6), 39-43.
- Ågerfalk, P. J., Fitzgerald, B., & Slaughter, S. A. (2009). Introduction to the special issue—flexible and distributed information systems development: state of the art and research challenges. *Information systems research*, 20(3), 317-328.

## LIITE 1 HAASTATTELUKYSYMYKSET

### TAUSTATIEDOT

1. Haastateltavan tehtävänimike
2. Haastateltavan kokemus vuosissa ja kuinka monessa hajautetussa ketterässä ohjelmistokehitysprojektissa tämä on ollut mukana?
3. Minkälaisia kokemuksia haastateltavalla on ketteristä menetelmistä tai periaatteista ohjelmistokehityksessä?
4. Minkälaisissa ja minkä kokoisissa tiimeissä haastateltava on työskennellyt? Minkälaisia vastuita haastateltavalla on ollut?

### HAJAUTETTU KETTERÄ OHJELMISTOKEHITYS

5. Miten hajautettua ketterää ohjelmistokehitystä on toteutettu? Mitä projekteissa on käytetty (menetelmät, periaatteet)?
6. Mitkä ovat tavoitteet/odotukset hajautetulle ketterälle tiimille?
7. Miten hajautettu ketterä tiimi kommunikoi?

### HAASTEET

8. Minkälaisia haasteita hajautus aiheuttaa ketterään ohjelmistokehitykseen? Miten ne näkyvät?

### ONNISTUMINEN

9. Mitä hajautetun ketterän ohjelmistokehitysprojektin onnistuminen tarkoittaa kohdeorganisaatiossa?
10. Miten onnistumista mitataan?

### KRIITTISET MENESTYSTEKIJÄT

11. Mitkä ovat ne menestystekijät, joilla onnistumiseen päästään? Perustelut.
12. Laadullinen kysely: käydään läpi kirjallisuuden esittämiä menestystekijöitä ja kuullaan, mitä haastateltava niistä ajattelee.

