

Hanna Kinnunen

**OHJELMISTOKEHITYKSEN KETTERYYS JA SEN
MITTAAMINEN**



JYVÄSKYLÄN YLIOPISTO
TIETOJENKÄSITTELYTIETEIDEN LAITOS
2015

TIIVISTELMÄ

Kinnunen, Hanna

Ohjelmistokehityksen ketteryys ja sen mittaaminen

Jyväskylä: Jyväskylän yliopisto, 2014, 55 s.

Pääaine, tutkimusraportin tyyppi: Tietojärjestelmätiede, pro gradu -tutkielma

Ohjaaja: Luoma, Eetu

Tutkielmassa pyritään luomaan yleiskuva ketteryydestä ohjelmistokehityksessä - sen määritelmästä, laajuudesta ja toteutumisesta. Tutkielmassa käydään läpi ketteryyteen liittyvää tutkimusta pyrkien löytämään yleisesti käytettyjä ketteryyden määritelmiä sekä käsityksiä ketteryyden ilmentymisestä.

Tutkielman jälkipuoliskolla ketteryydelle luodaan mittari, joka perustuu yhteen ketteryyden määritelmään, Agile Manifestoon (Julistus ketterästä ohjelmistokehityksestä). Mittari luodaan survey-tutkimusta varten ja sen tarkoituksena on pyrkiä mittaamaan eri organisaatioiden eroja niiden toiminnan ketteryydessä. Mittaria on kokeiltu suomalaisille ohjelmistoalan yrityksille tehdyssä tutkimuksessa ja tulosten pohjalta arvioidaan ketterien arvojen läsnäoloa yrityksen toiminnassa.

Tuloksia käsiteltiin kolmesta näkökulmasta: ketteryyden määritelmän sisäinen eheys, ketteryyden ja yrityksen markkinoiden välinen yhteys sekä ketteryyden ja yrityksen sisäisten ominaisuuksien välinen yhteys. Agile Manifestoon kuuluu neljä arvoa, joista kehitettiin 16 väittämää ja niistä analyysin pohjalta kolme muuttujaa. Ketteryyden muuttujilla oli jokseenkin vähän yhteyksiä yrityksen ominaisuuksiin. Yritykset, jotka eivät tarjonneet lainkaan tuotteita olivat ketterämpiä kuin yrityksille tuotteita tarjoavat. Yrityksen ominaisuuksista nuori ikä ja pieni henkilöstömäärä korreloivat kaikkien ketteryyden muuttujien kanssa. Ketteryyden muuttujat eivät korreloineet yrityksen liikevaihdon kasvun kanssa ja vain yksi muuttujista (muutosvalmius) korreloi yrityksen kannattavuusprosentin kanssa heikosti.

Asiasanat: Ketterä ohjelmistokehitys, Agile Manifesto, ketteryys, mittari, survey, kyselytutkimus

ABSTRACT

Kinnunen, Hanna

Measuring agility in software development

Jyväskylä: University of Jyväskylä, 2015, 55 p.

Major subject, type of the publication: Information System science, Master's Thesis

Supervisor: Luoma, Eetu

This thesis attempts to portray agility in software development as a whole including its definition, scope and practice. Themes of agility research are introduced in order to find widely accepted definitions of agility as well as views of its manifestation in the development process.

In the second half of the paper an instrument is created to measure agility using Agile Manifesto as the definition of agility. The instrument is created for survey research and its aim is to measure differences in agility between organizations. The measuring instrument has been tested in a research of Finnish software companies and the findings allow the examination of agile values in the operations of the company.

The findings were examined from three viewpoints: integrity of the definition of agility, the connection of agility and the company's market as well as the connection between the company's distinctive qualities and agility. Agile Manifesto retains four values, which were formed into 16 arguments and the results summed into three variables. The agility variables had relatively little connections to the qualities of the companies. Companies that didn't offer products were more agile than companies offering products to other companies. Young age and small staff correlated with all of the agility variables. Agility variables did not correlate with the growth rate of the companies' revenue and only one variable (readiness for change) correlated with profitability.

Keywords: Agile software development, Agile Manifesto, agility, measuring instrument, survey research

KUVIOT

KUVIO 1: Tutkimustulokset: kehitystiimin ominaisuudet, ketteryyden määritelmä, ohjelmistokehityksen lopputulos (Lee & Xia, 2010)	26
KUVIO 2: Ketteryyden luokittelu Conboyn (2009) mukaan	27
KUVIO 3: Viiksilaatikkodiagrammit ketteryyden summamuuttujille (muutosvalmius, dokumentaatio, kanssakäyminen)	42

TAULUKOT

TAULUKKO 1: Ketteryyden kuvauksia (kuvaukset ovat suoria lainauksia tai vapaita suomennoksia)	12
TAULUKKO 2 Agile Manifeston ja ketterien menetelmien periaatteiden vertailu (Agile Alliance, 2001)	18
TAULUKKO 3: Ketteryysväitteiden pääkomponenttianalyysin tulokset.....	36
TAULUKKO 4: Muuttujien kuvaukset.....	37
TAULUKKO 5: Ketteryyttä koskevat väitteet ja niiden vastausten keskiarvot .	40
TAULUKKO 6: Spearmanin järjestyskorrelaatio yrityksen ominaisuuksia kuvaaville muuttujille ja ketteryysmuuttujille	43

SISÄLLYS

TIIVISTELMÄ	2
ABSTRACT	3
KUVIOT	4
TAULUKOT	4
SISÄLLYS.....	5
1 JOHDATUS KETTERYYTEEN.....	7
2 KETTERYYS OHJELMISTOKEHITYKSESSÄ.....	10
2.1 Ketteryyden muodot	10
2.1.1 Ketteryys käsitteenä	11
2.1.2 Ketteryyden historiaa	14
2.1.3 Ketterät menetelmät.....	15
2.1.4 Ketterien periaatteiden vertailu	17
2.2 Ketteryyden tutkimus	19
2.2.1 Laadullista ketteryyden tutkimusta	19
2.2.2 Teoriaa soveltavia tutkimuksia	22
2.2.3 Ketteryyden kriittistä tarkastelua	23
2.2.4 Ketteryyden mittarit	24
2.3 Ketteryydellä parempaa ohjelmistokehitystä.....	27
2.4 Yhteenveto ketteryydestä ohjelmistokehityksessä	29
3 KETTERYYDEN MITTARI JA KYSELYTUTKIMUS	31
3.1 Tavoite ja tutkimusote.....	31
3.2 Mittarin kehittäminen	32
3.2.1 Kysely ja muuttujat	33
3.2.2 Aineiston kerääminen.....	34
3.3 Aineiston käsittely	35
3.3.1 Varsinaisista analyysimenetelmistä.....	37
4 TULOKSET.....	39
4.1 Vastaajajoukko ja vastaukset väitteisiin	39
4.2 Ketteryys ja yrityksen ominaisuudet	42
5 JOHTOPÄÄTÖKSET	44
6 YHTEENVETO	48

LÄHTEET.....	50
LIITE 1: KETTERYYDEN MÄÄRITTELY ARTIKKELEISSA.....	52
LIITE 2: KYSELYLOMAKE	54

1 JOHDATUS KETTERYYTEEN

Ketteryys ei ole käsitteenä helppo määritellä eikä se ole ilmiönä selvärajainen. Ketteryydellä ja ketterillä menetelmillä on omat kannattajansa, joiden ääni on kuulunut voimakkaana ohjelmistokehityksen metodeja koskevassa keskustelussa. Kaikki väitteet ketteryyden eduista eivät ole kuitenkaan selvästi tutkimukselle perustuvia, mikä on voinut hämähäyttää kuulijoiden käsitystä siitä millaisia ilmiöitä sanalla ketteryys kuvataan sekä niistä varsinaisista mitattavista eduista, joita ketteryydellä voidaan saavuttaa. (Erickson, Lyytinen & Siau, 2005.)

Yleisellä tasolla ketteryys voidaan määritellä kyvyksi olla vikkeliäliikkeinen, joustava, eloisa, valpas ja nopea toimimaan. (Erickson ym., 2005; Lyytinen & Rose, 2006). Ketterän ohjelmistokehityksen keskeisenä tarkoituksena on vähentää perinteisiin kehitysmenetelmiin liittyvää raskautta ja pyrkiä nopeasti reagoimaan muuttuvaan ympäristöön ja vaatimuksiin (Erickson ym., 2005). Ohjelmistoja tai tietojärjestelmiä kehittävän organisaation kohdalla tämä tarkoittaisi kykyä vaistota toimintaympäristön muutokset ja liiketoimintamahdollisuudet sekä hyödyntää niitä nopeasti kehitystyössä (Lyytinen & Rose, 2006). Ketterä ohjelmistokehitys on inkrementaalista mahdollistaen vähittäisen julkaisemisen, se perustuu yhteistyölle asiakkaan ja kehitystiimin kesken, se on suoraviivaista ja sen toimintatavat on helppo oppia sekä mukautuvaa eli muutoksia voidaan tehdä myös viime hetkellä (Abrahamsson, Salo, Ronkainen & Warsta, 2002). Ketterästä näkökulmasta katsottuna perinteiset menetelmät ovat liian hidasliikkeisiä ja kaavamaisia eivätkä sovi tapauksiin, joissa tulisi reagoida nopeasti muutoksiin (Erickson ym., 2005). Ketteryyden määritelmää pohditaan tarkemmin tämän tutkielman myöhemmissä luvuissa.

Tässä tutkielmassa kutsutaan ymmärrettävyyden helpottamiseksi perinteisiä ohjelmistokehitysmenetelmiä suunnitteluperustaisiksi (plan-based) (Ceschi, De Panfilis, Sillitti & Succi, 2005) ja ne ymmärretään eräänlaiseksi ajattelutavan vastakohtaksi ketterälle ohjelmistokehitykselle. Suunnitteluperustaiset kehitysmenetelmät keskittyvät ennustettaviin prosesseihin ja niiden pyrkiämyksenä on enemmänkin pyrkiä poistamaan muutostarpeet kuin hyväksyä muutosten vääjäämättömyys. Perinteisessä ohjelmistokehityksessä lähestymis-

tapana on esimerkiksi vesiputous tai spiraalimalli. (Mahapatra, Mangalaraj & Nerur 2005.)

Organisaatiot voivat toteuttaa ketterää ohjelmistokehitystä käyttäen hyväksi ketterää menetelmää (metodia). Ketterät menetelmät pyrkivät toimimaan vaikeasti ennustettavissa tilanteissa käyttäen apunaan ihmisten luovuutta ja kommunikaatiota, iteratiivista kehitystä ja asiakasyhteistyötä. Tunnettuja ketteriä menetelmiä ovat esimerkiksi Scrum ja Extreme Programming (XP). (Mahapatra ym., 2005.) Menetelmät sisältävät erilaisia periaatteita ja työohjeita joiden mukaan toimimalla ohjelmistoja kehittävän organisaation pitäisi pystyä toimimaan ketterästi.

Tässä tutkielmassa pyritään käyttämään termejä ohjelmistoyritys ja ohjelmistokehitys kuvaamaan sitä organisaatiota ja sen toimintaa, jossa ketterää kehitystä toteutetaan. Nämä käsitetään koskevan laajasti ohjelmistojen ja tietojärjestelmien kehitystyötä sen sijaan, että tutkimusalue tältä osin rajattaisiin tiukasti koskemaan vain tietynlaisia organisaatioita tai tuotteita. Ohjelmistokehitystä tekevistä työryhmästä taas käytetään nimeä tiimi. Kyky olla ketterä ilmenee juuri sosiaalisessa ympäristössä ja monet ketterät menetelmät keskittyvätkin juuri tiimin ketterän toiminnan tukemiseen. Esimerkiksi XP-menetelmässä tiimin kooksi on suositeltu 3 – 20 henkilöä eri rooleista, kun taas Crystal metodia voidaan soveltaa jopa sadalle hengelle. Useissa menetelmissä suuria organisaatioita kuitenkin pyritään pilkkomaan pienemmiksi tiimeiksi, joissa ketterälle kehitykselle ominainen keskustelu kasvotusten onnistuu. (Abrahamsson ym., 2002.)

Ohjelmistokehityksen ketteryyteen pyritään usein ketterien menetelmien avulla tarkoituksena pystyä vastaamaan joustavammin asiakkaiden tarpeisiin. Ohjelmistoja kehittävät organisaatiot ovat kuitenkin hyvin erilaisia ja tämän vuoksi myös erilaisten menetelmien käytöstä voi seurata erilaisia tuloksia. Tämän vuoksi onkin kiinnostavaa selvittää kykenevätkö organisaatiot olemaan ketteriä sen sijaan, että tutkittaisiin vain menetelmien käyttöä. Tämä vaatii ketteryyden ja siihen liittyvien käsitteiden määrittelyä, jotta ymmärretään, mitkä organisaatiot ovat ketteriä ja mitkä eivät. Mikäli ketteryys voidaan määritellä yleisellä tasolla ja määritelmästä luodulla mittarilla organisaation ketteryydestä voidaan saada tietoa, on myös mahdollista verrata eri organisaatioita keskenään. Tällainen yleistettävissä oleva tieto auttaa ymmärtämään ketterän toimintatavan vaikutusta ohjelmistokehitykseen.

Tässä tutkielmassa pyritään luomaan yleiskuva ketteryydestä ohjelmistokehityksessä painottaen ketterän kehittämisen alaan liittyvien käsitteiden yhteisiä piirteitä. Tutkimuskysymyksenä ja lisäkysymyksenä on:

- Millä käsitteillä ohjelmistokehityksen ketteryyttä voidaan mitata?
 - Onko yrityksen ketteryydellä yhteyttä yrityksen sisäisiin ominaisuuksiin tai niihin markkinoihin, joilla yritys toimii?

Ketteryyttä lähdetään käsittelemään käymällä ensin läpi ketteryys ohjelmistokehityksen ilmiönä tutustuen ketteryyden määrittelyyn, ketteryyden historiaan ja ketteriin menetelmiin. Seuraavaksi käsitellään ketteryyden

tutkimusta paneutuen myös ketteryyteen liittyvien väitteiden kritiikkiin. Lopuksi verrataan tutkimuksissa löydettyjä hyötyjä Agile Manifeston (Agile Alliance, 2001) arvoihin. Agile Manifeston arvojen pohjalta luodaan tutkimusosiossa kyselytutkimukseen mittari, jolla ketteryyttä mitataan suomalaisissa ohjelmistoyrityksissä. Ketteryyttä tarkastellaan sekä ositettuna yksittäisiksi ketteriksi arvoiksi, että yrityksen ominaisuuksien näkökulmasta.

2 KETTERYYS OHJELMISTOKEHITYKSESSÄ

Ketteryys ei ole käsitteenä yksiselitteinen vaan siihen liittyy laaja kirjo erilaisia tulkintoja. Seuraavien lukujen tarkoituksena on käsitellä tätä kirjoa erilaisista lähtökohdista ja pyrkien näin luomaan kokonaiskuva ilmiöstä sekä siihen liittyvien käsitteiden yhteyksistä. Ensin käydään läpi erilaisia ilmiöitä, joita kuvataan ketteriksi, jonka jälkeen esitellään ketteryyteen liittyviä tutkimuksia. Lopuksi esitellään tapoja, joilla ketteryyttä on pyritty mittaamaan.

Tässä luvussa esitellään ketteryyttä koskevaa kirjallisuutta erityisesti painottuen vertaisarvioituihin tutkimusartikkeleihin sillä katsauksen tarkoituksena on saada kattava kuva ohjelmistokehityksen ketteryyden tutkimuksesta ja nykytilasta. Lähtökohtana kirjallisuuden etsimiseen on käytetty Dybån ja Dingsøyryn (2008) systemaattista kirjallisuuskatsausta, jossa listattiin vuoteen 2007 mennessä julkaistut, alan keskeisistä tietokannoista löytyvät, ohjelmistokehityksen ketteryyttä käsittelevät empiiriset tutkimukset. Artikkelin lähteistä etsittiin tähän tutkielmaan soveltuvia artikkeleita ja tutkijoita, joiden uudempi tuotanto voisi soveltua lähteiksi ja tätä menetelmää jatkettiin taas uusien artikkeleiden kohdalla. Pyrkimys oli löytää mahdollisimman uusia sekä ketteryyden tutkimukselle keskeisiä artikkeleita. Tällä menetelmällä sekä suoraan tietojärjestelmätieteen keskeisiä tietokantoja hyväksikäyttäen etsittiin uusia artikkeleita, kunnes tutkimuskysymyksiin keskeisesti liittyvää uutta tietoa ei enää löydetty. Kriteereinä artikkeleiden valinnalle oli ketteryyden määrittelemisen ja käsittelemisen mahdollisimman yleisellä tasolla ohjelmistokehityksessä sekä empiiriset tutkimukset aiheesta. Yksittäisiä ketteriä menetelmiä sekä muuta kuin ohjelmistokehityksen ketteryyttä koskevia artikkeleita ei juuri otettu mukaan katsukseen.

2.1 Ketteryyden muodot

Ketteryydestä puhuttaessa on hyvä erottaa käsitteellisesti kyky toimia ketterästi ja menetelmä, jolla tämä kyky saavutetaan. Kyky toimia ketterästi ei vaadi ket-

terän menetelmän käyttöä ja toisaalta voi olla, ettei ketterä menetelmä onnistu auttamaan organisaatiota toimimaan ketterästi. Ensin käsitellään ketteryyttä käsitteenä esitellen sen määritelmiä, sitten ketterän ohjelmistokehityksen historiaa, jonka jälkeen esitellään ketteriä menetelmiä ja lopuksi menetelmien taustalla olevia arvoja ja niiden yhteyttä ketteryyden määritelmään.

2.1.1 Ketteryys käsitteenä

Ketteryydestä puhutaan tässä tutkielmassa ohjelmistokehityksen kontekstissa vaikka kyseistä käsitettä käytetäänkin kuvaamaan myös muita liiketoiminnan työtapoja. Conboy (2009) huomauttaa, että tietojärjestelmäkehityksen tutkimuksessa ketteryyttä pidetään varsin uutena käsitteenä, vaikka hallinnon ja tuotannon tutkimuksessa ketteryys, joustavuus ja turhan poistaminen ovat jo vakiintuneita tutkimuskohteita. Ohjelmistokehityksen ketteryys erotetaan myös koko organisaation ketteryydestä, mikä merkitsee koko yrityksen kykyä käsitellä muutoksia (Lu & Ramamurthy, 2011). Ohjelmistokehityksen ja koko organisaation ketteryyden välille ei kaikissa artikkeleissa vedetä tarkkaa rajaa mutta seuraavassa ketteryyden käsitettä pohditaan erityisesti ohjelmistokehitystä koskevien artikkeleiden avulla.

Manifesto for Agile software development eli Ketterän ohjelmistokehityksen julistus julkaistiin vuonna 2001 uudenlaisia ohjelmistokehitysmenetelmiä suosivan ryhmän työpajan tuloksena. Tässä tutkielmassa käytetään julistuksesta sen lyhyttä englanninkielistä nimeä Agile Manifesto, koska se on yleisesti kirjallisuudessa käytetty ja englanninkielisenä mahdollisesti lukijoillekin paremmin tuttu termi kuin suomennos. Monet Agile Manifeston kirjoittajat olivat XP-käytäntöjen edistäjiä ja käyttäjiä (Erickson ym., 2005) mutta myös monet muut ketterät menetelmät olivat työpajassa edustettuina. (Agile Alliance, 2001.) Tämän vuoksi on luonnollista, että erilaiset ketterien menetelmien työohjeet (XP, Scrum, Crystal) vastaavat jossain suhteessa Agile Manifeston periaatteita (Abrahamsson ym., 2008; Mahapatra ym., 2005).

Agile Manifestossa esitetään neljä arvoa, joissa yhdistyy parempi tapa kehittää ohjelmistoja (ks. TAULUKKO 1). Nämä neljä arvoa jaetaan 12 tarkemmaksi periaatteeksi. Agile Manifesto julkaistiin yli vuosikymmen sitten ja sen jälkeen ohjelmistokehityksen alalle on tullut valtava määrä uusia metodeja, työkaluja, tekniikoita ja suositeltuja käytäntöjä (Balijepally, Dingsøyr, Moe & Nerur, 2012). Ketteriä periaatteita voidaan soveltaa sekä koko organisaation että projektitiimin tasolla, jotta sekä tiimit että niiden tuottamat tuotteet olisivat muuttuviin olosuhteisiin mukautuvia (Abrahamsson ym., 2008).

TAULUKKO 1: Ketteryyden kuvauksia (kuvaukset ovat suoria lainauksia tai vapaita suomennoksia)

Lähde	Määritelmä
Agile Alliance 2001 "Agile Manifesto"	"Löydämme parempia tapoja tehdä ohjelmistokehitystä, kun teemme sitä itse ja autamme muita siinä. Kokemuksemme perusteella arvostamme: <ul style="list-style-type: none"> • yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja • toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota • asiakasyhteistyötä enemmän kuin sopimusneuvotteluja • vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa."
Conboy 2009	"Tietojärjestelmäkehityksen metodin jatkuva valmius nopeasti tai luontaisesti aiheuttaa muutosta, ennakoivasti tai reaktiivisesti hyväksyä muutos sekä oppia muutoksesta samanaikaisesti edistäen koetun lisäarvon luomista asiakkaalle (taloudellinen, laatu ja yksinkertaisuus) hyväksikäyttäen sen [metodin] kaikkia osioita ja suhteita ympäristöön."
Erickson, Lyytinen & Siau 2005	'Ketteryys yhdistetään usein sellaisiin käsitteisiin kuin vikkeilyys (nimbleness), notkeus (suppleness), nopeus (quickness), näppäryys (dexterity), eloisuus (liveliness), valppaus (alertness). Sen ytimessä on tarkoitus poistaa mahdollisimman paljon perinteisiin kehitysmenetelmiin liitettyä raskautta ja edistää nopeaa reagoimista muuttuviin ympäristöihin, muuttuviin asiakas / käyttäjävaatimuksiin, aikataulujen kiristymiseen ym.'
Lyytinen & Rose 2006	'Ketteryys voidaan määritellä yleisesti ominaisuutena tai kykynä olla nopea ja vikkela. Tietojärjestelmäkehityksen kontekstissa ketteryys voidaan määritellä organisaation kykynä aistia ja vastata nopeasti teknisiin muutoksiin ja uusiin liiketoimintamahdollisuuksiin. Vastaavasi ketterällä tietojärjestelmiä kehittävällä ja ylläpitävällä organisaatiolla on kyky aistia ja vastata odottamattomiin ympäristön muutoksiin ja hioa taitonsa nopeasti vastaamaan niihin.'
Balijepally ym. 2012	'Ketteryyden ytimessä on kyky nopeasti ja joustavasti luoda ja vastata muutokseen liiketalouden ja teknologian alueella. [...] Pohjimmiltaan nämä periaatteet johdattelvat käyttämään kevyttä menetelmää, joka edistää ohjattavuutta ja nopeaa reagoimista.'
Cockburn 2007	'Ketterä merkitsee olemista tehokas ja helposti ohjattavissa. Ketterä prosessi on sekä kevyt että riittävä. Keveys on keino olla helposti ohjattava. Riittävyys liittyy kykyyn pysyä mukana pelissä.'
Lee & Xia 2010	'Ohjelmistotiimin kyky (suoritusnopeudeltaan ja -laajuudeltaan) tehokkaasti vastata muutoksiin ja sisällyttää muutokset käyttäjävaatimuksissa projektin elinkaaren aikana.'

Agile Manifesto ei ole varsinaisesti teoria, vaikka sitä käsitelläänkin tässä tutkielmassa teorian kaltaisesti. Gummeruksen sanakirjan määritelmä teorialle on "asian tai ilmiön (tieteellisesti) perusteltu selitys" (Gummerus, 2015). Manifesto alkaa sanoilla "Löydämme parempia tapoja tehdä ohjelmistokehitystä,

kun teemme sitä itse ja autamme muita siinä.” (Agile Alliance, 2001). Määritelmän mukaisesti ketteryys siis selittäisi parempaa ohjelmistokehitystä. Agile Manifesto ei kuitenkaan perustu tieteelliselle tutkimukselle vaan sen on luonut asiantuntijaryhmä oman näkemyksensä pohjalta. Manifeston arvojen kehittämisen pohjalla olleet työtavat ovat kuitenkin niin laajasti levinneet ja tutkitut, että Agile Manifestoa voidaan pitää määritelmänä ketteryydelle.

Kuten voidaan huomata, Agile Manifesto tarjoaa listan ketteryyteen liittyvistä arvoista ja periaatteista sekä käsityksen ketterän toiminnan vastakohtasta. Conboyn (2009) taas on luonut tietojärjestelmäkehityksen (ISD) ketteryydelle käsittekuvausten, joka perustuu oletukselle, että joustavuus (flexibility) ja turhan poistaminen (leanness) ovat ketteryyden taustalla olevia käsitteitä ja että niiden määrittäminen kirjallisuuden avulla ja peilaaminen ketteryyttä vasten voivat yhdessä muodostaa pätevän ketteryyden käsitteen (ks. TAULUKKO 1).

Verrattaessa Conboyn määritelmää Agile Manifestoon, voidaan huomata, että jälkimäinen antaa selvästi tarkemmat ohjeet siitä, miten ketterää ohjelmistokehitystä toteuttavan organisaation tulisi toimia. Conboyn määritelmässä on taas mukana ajallinen elementti ja liiketaloudellinen näkökulma on selkeämmin esillä. Conboyn määritelmä on myös luotu käyttäen tieteellisempää lähestymistapaa. Itseasiassa Conboyn (2009) viittaa Agile Manifestoon mainosmateriaalina, joka yksinkertaistaa järjestelmäkehitystä liikaa. Kumpikaan näistä ketteryyden määrittämisestä ei kuitenkaan ole vielä saavuttanut täysin hegemonista asemaa ketteryyden tutkimuksessa (ks. liite 1), minkä vuoksi tutkimukselta puuttuukin tietynlaista teoreettista pohjaa. Conboyn (2009) mukaan artikkeleissa ei käytetä samoja määritelmiä ketteryydestä ja ketteristä menetelmistä ja näillä sanoilla viitataan hyvin erilaisiin asioihin. Tämä myös vaikuttaa ketterien menetelmien soveltamiseen sillä niiden keskeinen merkitys voi olla hankala ymmärtää ja mitata ilman pitävää teoreettista pohjaa (Petter & Yu, 2014). Mikäli ketteryyttä ei osata määritellä, on vaikea selvittää, miten ohjelmistokehitykseen käytettävä menetelmä auttaa kehitystiimin kykyyn saavuttaa ketteryyttä (Baskerville, Madsen & Pries-Heje, 2011).

Agile Manifeston ja Conboyn määritelmän lisäksi taulukossa esitellään viisi muuta kuvausta ketterästä ohjelmistokehityksestä (TAULUKKO 1). Näitä ei varsinaisesti voida kutsua määritelmiksi samalla tavalla kuin Conboyn määritelmää, joka on luotu päättelyketjun tuloksena aiempaan kirjallisuuteen perustuen. Näistä kuvailuista voidaan kuitenkin saada parempi käsitys niistä teemoista, joita ketteryyteen liitetään, kuten iteratiivisuus, joustavuus, nopeus, turhan poistaminen, reagoiminen, kommunikaatio ja asiakasyhteistyö.

Liitteenä olevaan taulukkoon (ks. liite 1) on listattu vaakariveille ilmestymisjärjestyksessä artikkeleita, joissa on kuvattu ketteryyttä ohjelmistokehityksessä tai ketteriä menetelmiä yleisellä tasolla. Pystysarakkeissa on ketteryyteen liittyviä määritelmiä ja käsitteitä (tai käsiteryhmiä), jotka on poimittu ketteryyden kuvauksista (ks. TAULUKKO 1). Väliin jääviin soluihin on merkitty x:llä ilmaukset, jotka esiintyivät kyseisessä artikkelissa. Taulukon tarkoituksena on auttaa hahmottamaan ketteryydestä käytettyjen kuvailuiden yleisyyttä. Taulukkoa tulkittaessa on myös otettava huomioon artikkelien julkaisuvuodet sillä

esimerkiksi Conboyn (2009) määritelmä on julkaistu vasta monien artikkelien kirjoittamisen jälkeen. On myös huomioitava, että lista ketteryyteen liittyvistä käsitteistä ei ole kaiken kattava. Esimerkiksi autonomiset kehitystiimit mainittiin useissa artikkeleissa, vaikkei niitä mainittu ketteryyden kuvauksissa.

Ketteryyteen liittyviä käsitteitä etsittiin yhteensä 20 artikkelista. Niistä 14 sisälsi maininnan Agile Manifestosta. Conboyn (2009) ketteryyden määritelmä mainittiin neljässä kahdeksasta vuoden 2009 jälkeen ilmestyneestä artikkelista. Eniten ketteryyttä kuvattiin reagoimisena muutokseen (17/20) ja seuraavaksi iteratiivisuus ja inkrementaaliset julkaisut (15/20). Nopeus ja asiakasyhteistyö saivat seuraavaksi eniten mainintoja (12/20). Kasvokkainen tai jatkuva kommunikaatio mainittiin 10 kertaa. Joustavuus ja turhan poisto mainittiin molemmat 8 kertaa eli alle puolessa artikkeleista. Tämä on mielenkiintoista sillä juuri näille termeille perustuu Conboyn (2009) määritelmä.

2.1.2 Ketteryyden historiaa

Ketterään ohjelmistokehitykseen liittyy useita ilmiöitä ja käsitteitä, joiden yhteys toisiinsa voi hämärtyä. Seuraavassa on esitelty kronologisesti joitain ketteryyteen liittyviä käsitteitä, jotta niiden suhteet toisiinsa ajallisesti olisi helpompi käsittää.

Ketteryyden juuret näyttävät tulevan valmistavan teollisuuden ja johtamisen aloilta (Balijepally ym., 2012). Ketteryyteen liitettävät käsitteet joustavuudesta ja turhan poistamisesta on liitetty (ajoneuvo-) teollisuuden tuotannon menetelmiin jo vuosisata sitten ja esimerkiksi Leaniin liittyvä ajattelutapa voidaan tunnistaa jo Toyotan autotehtaan 1950-luvun tuotantotavoista (Conboy, 2009). 1980-luvun puolivälissä sanaa Lean käytettiin ensimmäisen kerran tuotannon hallinnan ja myöhemmin tuotekehitysprosessin yhteydessä (Cusumano & Poppendieck, 2012). Leania pidetään ketterän ohjelmistokehityksen pohjana ja esimerkiksi Extreme Programming (XP) ja Scum ovat tekniikoita, joilla Leanista tuttuja turhan työn poistamiseen tähtäviä periaatteita sovelletaan ohjelmistokehitykseen (Ceschi ym., 2005).

Ketterät toimintatavat ohjelmistokehityksessä syntyivät 1990-luvulla vastauksena sellaisiin ympäristön muutoksiin, kuten "Internet-nopeus" (Internet speed), maailmanlaajuinen hyper-kilpailu ja vaatimus dynaamisuudelle (Lyytinen & Rose, 2006). Ohjelmistokehitykseltä vaadittu nopeus ja kompleksisuus yhdessä ympäristömuutosten kanssa johtivat siihen, että projektisuunnitelmat ja vaatimusmäärittelyt olivat usein vanhentuneita ennen kuin projekti saatiin vietyä loppuun. Vastauksena tähän ongelmaan ohjelmistokehittäjät muodostivat työtapoja, jotka hyväksyivät muutokset niiden välttelemisen sijaan. (Cockburn & Williams, 2003.)

Fujitsu kehitti jo vuonna 1993 oman ketterän työkalunsa nimeltä Agile Software Engineering Environment (ASEE). Työkalu kehitettiin, kun ohjelmistokehitystä pyrittiin tekemään maantieteellisesti hajautetulla tiimillä ja tarvittiin verkkopohjainen alusta, jonka avulla voitiin tehdä yhteistyötä ja julkaista ohjelmistoa puolen vuoden välein. (Erickson ym., 2005).

Sana Scrum on lähtöisin rugbyistä ja sen tiedetään ensimmäisen kerran esiintyneen kirjallisuudessa joustavasta ja nopeasta tuotekehitysprosessista puhuttaessa vuonna 1986. Ohjelmistokehitykseen tarkoitetun Scrumin vaiheita on esitelty kirjallisuudessa jo vuonna 1995 (Abrahamsson ym., 2002) mutta laajemmin se määriteltiin vuonna 2001 Swaberin ja Beetlen toimesta (Cusumano & Poppendieck, 2012). XP (Extreme Programming) taas julkaistiin 1990-luvun lopulla ja sitä pidetään yleisesti ketterien ohjelmistokehitysmetodien alkupisteenä, jonka jälkeen ketterän käsitteen alle alettiin kehittää uusia menetelmiä sekä yhdistää myös jo tunnettuja toimintatapoja. (Abrahamsson ym., 2002).

1990-luvulla näitä silloin kevyiksi kehitysmenetelmiksi kutsuttuja metodeja kehitettiin ympäri maailmaa. Vuonna 2001 eri menetelmien edustajia kokoon tui yhteen työpajassa, jossa huomattiin menetelmillä olevan yhteisenä tarkoituksena olevan asiakastytyväisyys ja korkea laatu. Tässä työpajassa kevyitä menetelmiä alettiin kutsua nimellä ketterä (agile) ja työpajan tuloksena syntyi Agile Manifesto. (Cockburn & Williams, 2003).

Vuonna 2005 julkaistussa tutkimuksessa kysyttiin ohjelmistoyritysten projektipäälliköiltä, mitä ketteriä kehitysmenetelmiä he tuntevat. Tuolloin kaikki vastaajista tiesivät ketteristä menetelmistä ja osasivat nimetä XP:n sellaiseksi mutta vain alle kolmasosa kaikista vastaajista tiesi Scrumista. (Ceschi ym., 2005). Tilanne on muuttunut vuoteen 2013 tultaessa, jolloin kaupallisen Yhdysvaltalais tutkimuksen vastaajista 88 % kertoi toteuttavansa ketterää kehitystä ja 73 % ilmoitti menetelmäksi Scrumin ja sen variaatiot (VersionOne, 2014).

2.1.3 Ketterät menetelmät

Ketterien menetelmien tarkoituksena on käyttää ohjelmistokehityksen toimintatavoissa hyväksi väistämätöntä muutosta sen sijaan, että sitä vastustettaisiin. Ketteriä menetelmiä ovat esimerkiksi XP (Extreme Programming), Lean ohjelmistokehitys, Crystal metodit, Scrum, Adaptive Software Development, Feature-driven Development (FDD), Dynamic Systems Development ja Agile Modeling (AM). (Erickson ym., 2005; Balijepally ym., 2012). Monilla ketterillä menetelmillä on keskeiset perusteoksensa tai edistäjänsä, joihin muussa tutkimuksessa viitataan mutta on huomattava, että koska menetelmiä sovelletaan monissa erilaisissa organisaatioissa ja kaupallisissa tarkoituksissa, ei monenkaan menetelmän sisällöstä ole yhtä absoluuttisen oikeaa kuvausta. Monista menetelmistä on myös muotoiltu palvelutuotteita, joita konsulttiyritykset myyvät ketteryyteen tähtääville yrityksille. Menetelmiin voidaankin siis yhdistää monia sellaisia toimintatapoja, joilla ei ole suoraa yhteyttä (Conboyn) ketteryyden määrittämään, kuten esimerkiksi pariohjelmointi XP:ssä (Conboy, 2009).

Seuraavassa esitellään joitain ketteriä menetelmiä, joita vuosien saatossa on kirjallisuudessa käsitelty. Huomioitavaa tässä listassa kuitenkin on, ettei mitään menetelmää tarkoituksella käsitellä kattavasti sillä tarkoituksena on ainoastaan tuoda esiin menetelmien yhteys ketterän kehittämisen periaatteisiin, eikä tehdä käyttöohjetta menetelmän valintaan. Tässä esitellään vain pieni osa kaikista ketteryyteen liitetyistä toimintatavoista eikä valintaa tähän tutkielmaan

ole tehty niiden käyttöasteen vaan käsitellyssä kirjallisuudessa tehtyjen mainintojen määrän perusteella.

XP-menetelmään kuuluu neljä arvoa, jotka ovat viestintä, yksinkertaisuus, palaute ja rohkeus, sekä neljä keskeistä toimintoa, jotka ovat koodaus, testaus, kuuntelu ja virheiden poisto (debuggaus) ja näistä on johdettu vielä 12 toimintatapaa. Yksinkertaistettuna tarkoitus on koodata vain asiakkaan vaatimia asioita eikä pyrkiä ennakoimaan tarpeita tehden mahdollisesti turhaa työtä. (Erickson ym., 2005). XP:ssä keskitytään moniin teknisiin työtapoihin, joista yksi merkittävä on testauslähtöinen ohjelmistokehitys (TDD, test-driven development) (Cusumano & Poppendieck, 2012). Siinä ohjelman testit kirjoitetaan jo ennen varsinaista ohjelmakoodia, millä pyritään siihen, että koodista tulee ymmärrettävämpää ja uutta koodia voidaan lisätä vanhan pohjan päälle (Mahapatra ym., 2005).

Scrum on nykyään niin laajasti tunnettu menetelmä, että monet käsittävät sen tarkoittavan samaa kuin ketterä ohjelmistokehitys (Cusumano & Poppendieck, 2012). Näin ei kuitenkaan ole vaan Scrum on menetelmä, joka keskittyy erityisesti projektinhallintaan ja näin sopii myös käytettäväksi yhdessä muiden ketterien kehitystekniikoiden kanssa. Scrumin mukainen kehitysprosessi perustuu tuotteen työlisterille (product backlog), josta valitaan kullekin sprintiksi (pyrähdykseksi, sprint) kutsutulle iteraatiolle tehtäviä suoritettavaksi. Scrumiin kuuluu erilaisia työkaluja työntekijöiden välisen kanssakäymisen edistämiseksi kuten päivittäiset kokoontumiset (daily scrum), sprintin loppukatselmus (sprint review) ja retrospektiivi. (Johansen, Kautz & Uldahl, 2014.)

Agile Modeling (AM, ketterä mallinnus) on kehitysprosessi, jossa pyritään mallintamaan ja dokumentoimaan tehokkaasti ja ketterästi. Sen periaatteet ovat: yksinkertaisuus, iteratiivinen kehitys, sitkeys, vähittäinen julkaiseminen, tehtävään keskittyminen, laadukkaiden tuotteiden tuottaminen, mallien luominen ja dokumentointi vain tarvittaessa, useat mallit, nopea ja selkeä palaute mallien muutoksista, useamman iteraation takaiset mallien ja dokumenttien hävittäminen. Ketterän mallinnuksen periaatteena on vähentää turhien UML-mallien piirtämistä ja keskittyä malleihin, jotka toimivat suoraan koodauksen pohjana helpottaen sitä. (Erickson ym., 2005).

Crystal perheeseen kuuluu useita metodeja (kirkas, oranssi...) joista valitaan sopiva metodi käyttöön projektin koon ja kriittisyyden perusteella. Kaikille perheen metodeille on yhteistä inkrementaalinen kehittäminen, kommunikaatio ja yhteistyö sekä mahdollisuus käyttää yhtäaikaaisesti erilaisia kehityskäytäntöjä kuten XP ja Scrum. (Abrahamsson ym., 2002).

Lean kehittämisen keskiössä ovat turhan työn vähentäminen ja arvon tuottaminen asiakkaalle. Lean ohjelmistokehitys seuraa seitsemää periaatetta, jotka ovat: optimoi kokonaisuus, eliminoi turha työ, rakenna sisäistä laatua, opi jatkuvasti uutta, toimita nopeasti, sitouta kaikki sekä paranna suoritusta jatkuvasti (Cusumano & Poppendieck, 2012).

Kanban on ohjelmistokehitykseen tuotantoteollisuudesta muunnettu järjestelmä, joka sopii käytettäväksi yhdessä esimerkiksi Leanin kanssa. Siinä työn virta on kuvattuna taululle, jossa yksittäiset tehtävät liikkuvat kehityksen vai-

heesta toiseen järjestyksessä. Järjestelmä perustuu työmäärän rajoittamiselle kussakin vaiheessa niin, että työn kulku (flow) pysyy tasaisena jatkuvasti eikä synny pullonkauloja. Kanban voidaan laajentaa helposti koskemaan myös laajempaa arvoketjua, jolloin mukana on tehtäviä myös muuhun liiketoimintaan kuin ohjelmistokehitykseen, esimerkiksi markkinointiin, liittyen. (Cusumano & Poppendieck, 2012).

2.1.4 Ketterien periaatteiden vertailu

Sekä Agile Manifestoon että ketteriin menetelmiin liittyy keskeisiä periaatteita, joiden avulla ohjelmistokehityksestä pitäisi tulla ketterää. Kaikkien näiden yksityiskohtainen läpikäyminen ei ole tämän tutkielman puitteissa perusteltua. Yhteydet Agile Manifeston ja ketterien menetelmien välillä ovat kuitenkin tärkeitä huomioida ketteryyden kokonaisuuden hahmottamiseksi. Seuraavassa taulukossa (ks. TAULUKKO 2) on lueteltu kaikki Agile Manifeston neljä arvoa ja 12 periaatetta sekä niitä vastaavat Lean menetelmän periaatteet ja XP menetelmän arvot. Periaatteet eivät vastaa toisiaan yksi yhteen, minkä vuoksi jotkin periaatteet esiintyvät useita kertoja.

TAULUKKO 2 Agile Manifeston ja ketterien menetelmien periaatteiden vertailu (Agile Alliance, 2001)

Arvo	Agile Manifeston periaatteet	Leanin periaatteet	XP: arvot
Toimivaa ohjelmistoa Yksilöitä ja kanssakäymistä	Liiketoiminnan edustajien ja ohjelmistokehittäjien tulee työskennellä yhdessä päivittäin	Sitouta kaikki	Viestintä
	Projektit rakennetaan motivoituneiden yksilöiden ympärille joille tarjotaan puitteet ja tuki jotta he saavat työn tehtyä	Sitouta kaikki	Rohkeus
	Kasvokkain keskustelu on tehokkain tapa jakaa tietoa kehitystiimille ja sen sisällä		Viestintä
	Parhaat arkkitehtuurit, vaatimukset ja suunnitelmat syntyvät itseohjautuvissa tiimeissä	Sitouta kaikki	Palaute
	Tiimi arvioi säännöllisesti, kuinka parantaa tehokkuuttaan ja mukauttaa toimintansa sen mukaisesti	Sitouta kaikki, paranna suoritusta jatkuvasti	Palaute
	Ketterät menetelmät kannustavat kestävään toimintatapaan ja saavutettu työtahti pitäisi pystyä ylläpitämään	Paranna suoritusta jatkuvasti	
	Edistymisen ensisijainen mittari on toimiva ohjelmisto	Eliminoi turha työ, toimita nopeasti	Palaute
	Oleellista on yksinkertaisuus eli tekemättä jätetyn työn maksimointi	Eliminoi turha työ	Yksinkertaisuus
	Teknisen laadun ja ohjelmiston hyvän rakenteen jatkuva huomioiminen edesauttavat ketteryyttä	Rakenna sisäistä laatua, toimita nopeasti	Palaute
	Vastuuta asiakasyhteistyötä	Asiakkaan tyydyttäminen toimittamalla tämän tarpeet täyttäviä versioita ohjelmistosta aikaisessa vaiheessa ja säännöllisesti	Optimoi kokonaisuus, opi jatkuvasti uutta
Versioita toimivasta ohjelmistosta toimitetaan säännöllisesti, parin viikon tai kuukauden välein		Opi jatkuvasti uutta, toimita nopeasti	
Vastaamista muutok-työtä	Muuttuvien vaatimuksien vastaanottaminen kehityksen myöhäisessä vaiheessa, ketterät menetelmät hyödyntävät muutosta asiakkaan kilpailukyvyyn edistämiseksi		Palaute

Taulukosta on huomattava, että jokaisen Agile Manifeston periaatteen kohdalle ei löydetty Leanin periaatetta tai XP:n arvoa. Manifeston periaatteet eivät myöskään jakaudu tasaisesti kaikkien sen arvojen mukaan vaan painotus on selvästi yksilöissä ja kanssakäymisessä.

2.2 Ketteryyden tutkimus

Balijepally, Dingsøy, Moe ja Nerur tekivät vuonna 2012 selvityksen ketteryyden tutkimuksen tilasta vuosina 2001–2010 ja huomasivat, että artikkelien määrä kasvoi tuona aikana vuosi vuodelta ollen yhteensä 1551 artikkelia koko aikana. He valitsivat Agile Manifeston julkaisuvuoden tutkimuksensa lähtöpisteeksi sillä juuri tuolloin ketterien menetelmien käyttöönotto ja niiden tutkiminen alkoivat lisääntyä. Vuosikymmenen alussa tutkimuksesta puuttui vielä tarkkuutta teoreettisesti sekä metodien osalta ja tutkimus oli keskittynyt menetelmien käyttöönoton tutkimiseen. Ketteryyden tutkimuksen teoreettiset lähtökohdat ovat hyvin moninaisia eikä monenkaan kohdalla tutkimuksia ole tehty yhtä enempää, poikkeuksena tiedon hallinta, persoonallisuus, kompleksiset adaptiiviset järjestelmät ja sosiaalinen fasilitaatio. Kartoituksen mukaan vuosina 2001–2010 Yhdysvalloissa julkaistiin eniten (338) akateemisia artikkeleita liittyen ketteryyteen kun taas Suomessa julkaistiin maailman neljänneksi eniten (94). (Balijepally ym., 2012.)

Ensimmäinen ketterää ohjelmistokehitystä koskeva kirjallisuuskatsaus on tiettävästi julkaistu Suomessa VTT:llä vuonna 2002 (Dybå & Dingsøy, 2008). Katsauksessa käsitellään kymmentä eri ohjelmistokehityksen menetelmää, jotka täyttävät enemmän tai vähemmän ketteryyden määritelmän. Määritelmän mukaan menetelmän tulee olla inkrementaalinen, yhteistyötä korostava, helposti käytettävä ja mukautuva. Vertailun perusteella useimmat ketterät menetelmät eivät näyttäneet soveltuvan kaikkiin tuotteen elinkaaren vaiheisiin erityisesti sen alussa ja lopussa. Aikana, jolloin Agile Manifeston julkaisusta oli kulunut vasta vuosi, löydettiin jonkin verran tukea väitteille ketterien menetelmien tehokkuudesta ja soveltuvuudesta moniin tilanteisiin, mutta hyvin vähän empiriaan perustuvia tutkimustuloksia. (Abrahamsson ym., 2002.)

Kuten seuraavissa luvuissa esitetyistä tutkimusten kuvauksista voidaan huomata, ei ketteryyden tutkimus useinkaan perustu teorialle tai teorian sovellettavuudesta aiheeseen ei välttämättä löydy todisteita, kuten esimerkiksi Abrahamsson ym. totesivat artikkelissaan (2008). Balijepally ym. tutkivat 2001 - 2010 ketteryydestä julkaistujen 452 artikkelin otsikot ja huomasivat, ettei suuri osa ketteryyden tutkimuksesta perustu teorialle. (Balijepally ym. 2012.) Tutkimuksista suurin osa näyttää myös olevan case-tutkimuksia ja suurin otos eri organisaatioita yhdessä tutkimuksessa oli 20 Ceschin ym. tutkimuksessa. Dybån ja Dingsøyin systemaattisessa kirjallisuuskatsauksesta selviää, että vuoteen 2008 mennessä tehdyistä tutkimuksista 39 % oli yhden casen ja 33 % useamman casen tutkimuksia muiden, erityisesti kvantitatiivisten metodien, jäädessä alakynteen. (Dybå & Dingsøy, 2008).

2.2.1 Laadullista ketteryyden tutkimusta

Vuoteen 2008 mennessä julkaistuista ketteryyttä empiirisesti tutkineista artikkeleista yli 70 % oli case-tutkimuksia (Dybå & Dingsøy, 2008). Seuraavassa esitel-

lään neljä eri lähtökohdista tehtyä case-tutkimusta, joissa on löydetty ketterien menetelmien käyttämisen etuja ja haasteita. Viimeisenä esitellään kaksi grounded theory -menetelmällä tehtyä tutkimusta, joissa haastatteluja tehtiin ilman ennako-odotuksia.

Johansen, Kautz ja Uldahl (2014) tutkivat Scrumin koettua vaikutusta tuottavuuteen tapaustutkimuksessa, jossa toteutettiin 11 strukturoitua haastattelua yhdessä kohdeorganisaatiossa. Tutkimuksessa kysyttiin seitsemän indikaattorin avulla arviota Scrumin vaikutuksesta verrattuna aiempaan perinteiseen työskentelytapaan sekä Scrumin vaikutusta tuottavuuteen. Seitsemän indikaattoria olivat: keskeytysten määrä, kehityskierrosten jatkuva määrä, virheiden toistaminen, määräaikojen pitävyys, virheiden (bug) korjaamiseen kuluva aika, keskeytyksettä jatkuva työaika ja työntekijöiden suoritus. Erityistä parannusta koettiin tapahtuneen työntekijöiden suorituksessa sillä työntekijät ottivat enemmän vastuuta omasta panoksestaan. Parannusta koettiin myös kehityskierrosten lyhenemisestä, koska turhaa työtä epäkelpojen julkaisujen parissa tuli tehdä vähemmän. Vastaajat kokivat myös, että työ oli helpompi saada valmiiksi määräajassa, kun koko kehitysprosessi oli pilkottu pienemmiksi osatehtäviksi. (Johansen ym., 2014.)

Ceschi, De Panfilis, Sillitti ja Succi (2005) tutkivat sitä, kuinka hyvin projektipäälliköt hallitsivat ihmisiä suunnittelupohjaisia ja ketteriä menetelmiä käyttävissä yrityksissä. Tutkijat käyttivät kyselyn luomisessa lähestymistapana kolmetasoista mittausjärjestelmää: konseptuaalinen, operationaalinen, mitattava (goal, question, metric, GQM). He pyrkivät varmistamaan vastausten eheyden ja oikeellisuuden testaamalla lomaketta laajasti (jopa suuremmalla joukolla kuin lopullinen otos, joka oli 20 vastaajaa) sekä kaksivaiheisella vastaamisella, jossa haastattelukysymyksiin vastaamisen lisäksi vastaajat myös hyväksyivät haastattelusta tehdyn litteroinnin. Tutkimuksen tuloksena saatiin tietää, että sekä suunnittelupohjaisia että ketteriä menetelmiä käyttävissä yrityksissä suurimmaksi ongelmaksi koettiin kehitysprojektin saaminen valmiiksi kokonaisuudessaan määräaikaan mennessä. Tutkimuksen perusteella näyttää siltä, että ketterien menetelmien käyttöönotolla voidaan parantaa erityisesti yrityksen asiakassuhdetta ja muita hyötyjä ovat laadun, vaatimushallinnan ja työtiimin tyytyväisyyden paraneminen. Keskeisimpiä ongelmia ketterien menetelmien käyttöönotossa ovat vaikeus ennustaa tulevia kuluja sekä uusien konseptien mukanaan tuomat ongelmat silloin kun organisaation on hankala mukautua muutokseen. (Ceschi ym., 2005.)

Conboy, Coyle, Wang ja Pikkarainen (2011) tekivät case-tutkimuksen 17 organisaatiossa, jotka olivat käyttäneet ketteriä menetelmiä yli kolmen vuoden ajan. Tutkimuksessaan he pyrkivät löytämään ihmisten työskentelyyn liittyviä haasteita ja niiden ratkaisuja. Ketteriin menetelmiin liittyvät haasteet tyypiteltiin yhdeksään luokkaan: kehittäjien pelko osaamisen puutteiden paljastumisesta, kehittäjien osaamiselle asetetut laajemmat vaatimukset, sosiaalisen vuorovaikutuksen lisääntyminen, liiketoiminta-alueen huono tuntemus, ketterien arvojen ja periaatteiden huono ymmärtäminen, motivaation puute, hajautettu päätöksenteko, työpanoksen arviointi ja palkitseminen sekä rekrytoinnin on-

gelmat. Tekstissä esiteltiin myös organisaatioissa kehitettyjä ratkaisuja näihin ongelmiin ja painotettiin sitä, että ongelmien syntyminen ja niiden ratkaisut riippuvat paljon organisaatiosta itsestään. (Conboy ym., 2011.)

Conboy, Lang ja McHugh (2012) tutkivat ketteriä menetelmiä (iteraation suunnittelu, päivittäiset kokoukset, iteraation retrospektiivi) käyttävän kehitystiimin sisäistä luottamusta toisiinsa kolmen case-yrityksen (25 haastattelua) avulla. Kaikissa yrityksissä ketterien menetelmien käyttö lisäsi työskentelyn läpinäkyvyyttä tiimin sisällä ja muulle organisaatiolle. Ketterä menetelmä johti myös vastuunoton ja yhteisen vastuunkannon lisääntymiseen. Myös kommunikointi lisääntyi yli välttämättömän tason, minkä koettiin lisäävän luottamusta samoin kuin tiedon jakaminen ja palaute. Luottamuksen saavuttamiselle haasteita loivat tiimin ulkopuolisten kommentit, ryhmäpaine, tuoteomistajan rooli ja saavutettavuus, työmäärän arviointi ja retrospektiivin merkitys. Haasteista huolimatta ketterät menetelmät auttoivat tiimejä toimimaan paremmin yhdessä ja lisäsivät luottamusta (Conboy ym., 2012).

van Vliet ja van Waardenburg (2013) tutkivat Grounded theory -menetelmällä ihmisten välistä vuorovaikutusta kahdessa suuryrityksessä, joissa yritystä ohjattiin suunnitteluperustaisin menetelmin, mutta ohjelmistokehitystä toteutettiin ketterin menetelmin. Tutkijat haastattelivat 21 ketterien menetelmien käyttäjää yrityksissä ja pyrkivät löytämään aineistosta toistuvia teemoja sekä peilaamaan sitä aiempaa tutkimusta vasten. Tutkimuksen perusteella löydettiin kaksi pääteemaa, jotka olivat IT-alueen monimutkaistuminen sekä liiketoimintapuolen vähäinen osallistuminen. IT-alueen monimutkaistumisen helpottamiseksi löydettiin kolme strategiaa: yhteinen merkityksellisyyden kokemus tehtävälle työlle, yhteisten toimintalinjojen luominen hanketasolla sekä projektipäällikön työn mahdollistaminen (erityisesti poistamassa kitkaa kahden työtavan välillä). Liiketoimintapuolen osallistumisen parantamiseksi ehdotettiin seuraavia asioita: liiketoiminnan sidosryhmien ajatustapaan vaikuttaminen, liiketoimintaan liittyvän tiedon ohjaaminen tiimille tuotteen omistajan (Product Owner) kautta sekä yhtenäisen linjan muodostaminen tiedon ja vaatimusten osalta liiketoiminnan tasolla. (van Vliet & van Waardenburg, 2013.)

Baskerville, Madsen ja Pries-Heje (2011) tutkivat grounded theory-menetelmällä ohjelmistokehityksen muutosta neljänä eri ajankohtana vuosina 1999, 2001, 2003 ja 2008 käyttäen hiukan erilaisia tutkimusasetelmia. Ensimmäisessä tutkimuksessa 1999 korostui Internet-ajan mukanaan tuoma aikapaine sekä muuttuvat ja tuntemattomat vaatimukset. Toisessa tutkimuksessa 2001 korostuivat uusien työtapojen, kuten asiakkaan osallistumisen käyttöönotto ja vakiintuminen. Kolmannessa tutkimuksessa 2003 nousi esiin taloudellinen muutos ja ohjelmistokehitys osana liiketoimintaratkaisuja. Viimeisessä tutkimuksessa vuonna 2008 esiin nousi ketteriä menetelmiä käyttävän ohjelmistokehityksen ja suunnitteluperustaisia menetelmiä käyttävän muun organisaation välinen jännite. Tutkijat löysivät eri vuosien väliltä toistuvan kaavan, jossa ensin tapahtuu muutoksia ympäristössä ja sitä seuraa vaihe, jossa prosessit muuttuvat. Käyttämällä Conboyn (2009) luokittelua, tutkijat määrittelivät kahden ensimmäisen tutkimuskerran kuuluneen esi-ketterään (pre-agility) aikaan, sillä

niiden aikana organisaatiot eivät täyttäneet ketteryyden vaatimuksia. Kaksi jälkimmäistä tutkimuskertaa taas toteutettiin Agile Manifeston julkistamisen jälkeen. Tutkijat ennustivat myös jälki-ketterän (post-agile) ajan olevan tulossa, eli rajat ketteriä menetelmiä käyttävien kehitystiimien ja muun suunnitelmaaperustaisesti toimivan organisaation välillä murtuvat. (Baskerville ym., 2011)

Jo kuuden tutkimuksen esittelyn pohjalta voidaan huomata, että ketteryyteen liittyvät tutkimuskysymykset laadullisessa tutkimuksessa ovat hyvin laaja-alaisia, vaikka yhteneväisyyksiäkin löytyy. Vastuun ottaminen omasta työstä ilmeni tuottavuutta lisäävänä Johansenin ym. (2014) artikkelissa ja luottamusta lisäävänä Conboyn ym. (2012) artikkelissa. Maininta liiketoimintapuolen tai tuoteomistajan vähäisestä osallistumisesta löytyy Conboyn ym. (2012) sekä van Vlietin ja Van Waardenburgin (2013) artikkeleista. Kahdesta Grounded theory-menetelmällä toteutetusta tutkimuksesta löytyi yhtäläisyys jännitteestä ketterän kehitystiimin ja suunnitelmaaperustaisesti toimivan muun organisaation välillä. Varsinaisesti vertailtavia tuloksia artikkeleista ei kuitenkaan löydy sillä ainoastaan Conboyn (2009, ks. luku 2.2.4) ja Baskervillen ym. (2011) artikkeleissa käytettiin samaa luokittelua eivätkä näissäkään tulokset ole vertailukelpoisia.

2.2.2 Teoriaa soveltavia tutkimuksia

Suurin osa ketteryyden tutkimuksesta ei vaikuta perustuvan teorialle. Tutkimuksessa on kuitenkin sovellettu teoriaa jonkin verran ja suosituimmat teoriat vuosina 2001 - 2010 liittyivät tiedonhallintaan ja persoonallisuuteen. (Balijepally ym., 2012.) Seuraavassa esitetään neljä tutkimusta, joissa on pyritty sovelta- maan erilaisia teorioita ketterään ohjelmistokehitykseen.

Suomalainen tutkijaryhmä julkaisi vuonna 2008 tutkimuksen, jossa tutkittiin kommunikaatiota Scrum ja XP-menetelmät käyttöönotaneissa ohjelmistokehitystiimeissä (yhden yrityksen kaksi projektia). Tutkimusmetodina käytettiin tapaustutkimusta ja tietoa kerättiin puolistrukturoiduilla haastatteluilla sekä havainnoinnilla. Käyttöön otetuista ketteristä käytännöistä esimerkiksi sprintin suunnittelulla, päivittäisillä tapaamisilla ja avoimella toimistotilalla oli positiivisia vaikutuksia tiimin sisäiselle kommunikaatiolle. Tiimin kommunikaatiossa sidosryhmien kanssa löytyi enemmän ketteristä käytännöistä johtuvia ongelmia, kuten erilaiset odotukset pidetyille kokouksille ja dokumentaatiolle. Tutkimuksessa pyrittiin käyttämään koordinaatioteorian riippuvuussuhteita apuna kommunikaation luokittelussa mutta tutkimuksen perusteella huomattiin, että ohjelmistokehityksen kommunikaatiossa on myös muita riippuvuuksia kuin kehityksessä on kuvattu. (Abrahamsson ym., 2008.)

Lyytinen ja Rose (2006) tutkivat tietojärjestelmiä kehittävien organisaatioiden ketteryyttä selvittämällä niiden kykyä nopeasti huomioida ja hyväksikäyttää arvoketjussa syntyviä innovaatioita. Eryityisesti tutkimuskohteena oli organisaation kyky oppia löytämällä ja hyväksikäyttämällä ympäristön innovaatioita. Tutkimus toteutettiin 2000–2003 pitkittäistutkimuksena seitsemässä case-yrityksessä, jotka kaikki kehittivät ketterästi verkkopohjaisia järjestelmiä eri toimialoille. Yritysten kyvykkyksiä tutkittiin viiden kehitysprosessin ta-

voitteen näkökulmasta: nopeus, innovatiivinen sisältö, kustannukset, riskit ja laatu. Tutkittavat henkilöt kuvasivat toimintaansa eri ajanhetkinä ja tutkijat löysivät jonkin verran todisteita teorialle, jonka mukaan, kun yksi viidestä tavoitteesta nousee muita tärkeämmäksi, tulee joistain muista tinkiä (vrt. projektin tuloskolmio). Tutkimus toteutettiin aikana, jolloin verkkopohjaisten järjestelmien kehitys oli murroksessa ja vaatimukset yrityksen nopealle reaktiokyvyille sekä sen tuottamille tuotteille muuttuivat selvästi neljän vuoden aikana. (Lyytinen & Rose, 2006.)

Yu ja Petter (2014) tutkivat XP- ja Scrum-menetelmien käytäntöjä käyttäen näkökulmana kognitiivisen psykologian teoriaa jaetuista mentaalista malleista (shared mental models). Tiimissä syntyy tehtäviin ja tiimityöhön liittyviä malleja ja niiden syntyminen tapahtuu neljässä vaiheessa, jotka ovat tietoisuus, oppiminen, ymmärtäminen ja toteuttaminen. XP:stä tuttu järjestelmän metafora auttaa yhteisten mallien luomisessa tietoisuuden ja oppimisen vaiheissa sillä sen avulla tiimi tunnistaa tiedon, joka puuttuu tai ei ole kaikilla tiimin jäsenillä. Scrumin päivittäiset seisten pidetyt kokoontumiset auttavat tiimiä erityisesti tehtävien ymmärtämisen ja toteuttamisen vaiheissa lisäten yhteisymmärrystä tehtävien prioriteeteista ja ongelmista. Jatkuvasti läsnä oleva asiakkaan edustaja auttaa myös ymmärtämisen ja toteuttamisen vaiheissa lisäten yhteisymmärrystä tehtävien oikeellisuudesta. Artikkelit ei esittele minkäänlaista empiiristä aineistoa, joka kuvaisi yhteisten mentaalisten mallien syntymistä todellisessa tilanteessa. Kirjoittajat kuitenkin suosittelevat jaettujen mentaalisten mallien soveltamista arvioitaessa ketterien menetelmien käytön tiimille tuomaa arvoa. (Petter & Yu, 2014).

Kuten voidaan huomata niin Abrahamsson ym. sekä Lyytinen ja Rose pyrkivät soveltamaan teorioita tapaustutkimuksessa ja molemmissa todettiin, ettei teoria täysin sovellu ketteryyden tutkimukseen. Petter ja Yu taas käyttivät kognitiivisen psykologian teoriaa eräänlaisena viitekehiksenä ketteryyden arvioimiseen mutta heidän artikkelinsa ei kuulunut empiiristä osuutta, joten heidän päätelmiensä soveltuvuudesta ketteryyden tutkimuksessa ei ole näyttöä.

2.2.3 Ketteryyden kriittistä tarkastelua

Ketterien menetelmien luomisesta, edistämisestä ja jalkauttamisesta ovat olleet menetelmien käyttöönoton alkuaikoina vastuussa lähinnä niitä työssään käyttävät asiantuntijat ja konsultit (Conboy, 2009; Agile Alliance 2001). Menetelmien saadessa suosiota, myös niiden tutkimus on lisääntynyt vuosi vuodelta (Dybå & Dingsøyr, 2008) ja myös ketteryyden väitettyjä etuja vastaan löytyy todisteita. Seuraavassa on esitelty kolme artikkelia, joissa lähtökohdaksi on otettu ketteryyteen liittyvä väite, jota on lähdetty tarkastelemaan järjestelmällisesti.

France, Rumpe ja Turk (2005) tutkivat Agile Manifeston 12 periaatetta kriittisesti etsien niiden takana olevia oletuksia. Heidän mukaansa kehitettävän ohjelman tyyppi voi olla sellainen, ettei sitä voida ymmärrettävästi esittää asiakkaalle tai jakaa pieniin kehitettäviin osiin. Kommunikaatio kasvokkain ei aina

onnistu asiakkaan kanssa tai edes tiimin sisällä sillä aika, paikka ja muut prioriteetit voivat olla esteenä ja toisaalta tekstipohjainen kommunikaatio luo tallennettavaa ja siirrettävää tietovarantoa. Muutosten hyväksyminen myöhäisessäkin vaiheessa voi antaa mahdollisuuden tuottaa asiakkaalle lisäarvoa mutta ketterällä menetelmällä tuotettu ohjelmisto ei eroa vesiputousmallilla tuotetusta virheiden hintavuuden suhteen silloin, kun ne vaativat suuria muutostöitä suureen määrään koodia. Oletuksena on, että kehitystiimin jäsenet kykenevät itse ohjaamaan omaa työtään ja arvioimaan sitä, mikä voi joillekin tiimeille olla liian haasteellista. Oletus siitä, että koodia voisi jatkuvasti päivittää ja samalla varmistaa sen laatua ei välttämättä sovi hyvin suurille tai kriittisille järjestelmille. Vaatimus yksinkertaisuudesta voi johtaa siihen, että samaa koodia kirjoitetaan uudestaan, koska ”pahan päivän varalle” ei saa koodata ja toisaalta iteratiivinen kehittäminen saattaa jopa sotia yksinkertaisuutta vastaan, koska useiden muutosten myötä järjestelmä muuttuu entistä monimutkaisemmaksi. (France, Rumppe & Turk, 2005.)

Vuonna 2005 julkaistussa tapaustutkimuksessa Merisalo-Rantanen, Rossi ja Tuunanen tutkivat kuinka XP on otettu käyttöön ja kuinka sitä sovelletaan kehitystyössä. He käsittelivät kahta yritystä, joissa käytettiin XP:n menetelmiä mutta kummassakaan XP:tä ei ollut otettu käyttöön hallitusti tai kaikilta osin. Tutkimuksen tulosten mukaan XP:n käytännöt näyttivät olevan samoja kuin vanhat työtavat, joita aikaisemmin on pidetty talon sisällä räätälöityinä menetelminä. Tutkijat päättelivätkin, että ketteryys on ”vanhaa viiniä uudessa pullossa”. (Merisalo-Rantanen, Rossi & Tuunanen, 2005.)

Balijepally, Mahapatra, Nerur ja Price (2009) tekivät kontrolloidun kokeen pariohjelmoinnin eduista erityisesti lähtökohtana tutkia tieteellisesti siihen liitettyjä ylistäviä väitteitä ja oletusta, että ryhmän yhteistoiminta tuottaisi parempia tuloksia kuin sen jäsenet yksinään. Kokeeseen osallistui 120 jonkin verran ohjelmointitaustaa omaavaa opiskelijaa, jotka jaettiin tekemään tehtäviä yksin (muodostettiin vertailuparit) ja parityönä. Tulosten mukaan parin tuottaman ohjelmiston laatu oli parempi kuin yksin työskennelleistä vertailupareista ohjelmointitaidoiltaan huonomman henkilön mutta työn jälki ei ollut parempaa kuin vertailuparin paremman ohjelmoijan yksin tekemä työ. Ohjelmointitehtävän kompleksisuus vaikutti ohjelmiston laatuun riippumatta siitä, tehtiinkö työtä yksin vai pareissa. Pareissa työskennelleet koehenkilöt olivat tyytyväisempiä työnsä laatuun kuin yksin työskennelleet. Parien luottamus suorituksiensa oli myös parempi kuin vertailuparien huonompien jäsenten mutta ei parempi kuin parin paremman jäsenen luottamus. Tämän tutkimuksen valossa väitteet pariohjelmoinnin eduista eivät siis pidä paikkaansa. (Balijepally ym., 2009.)

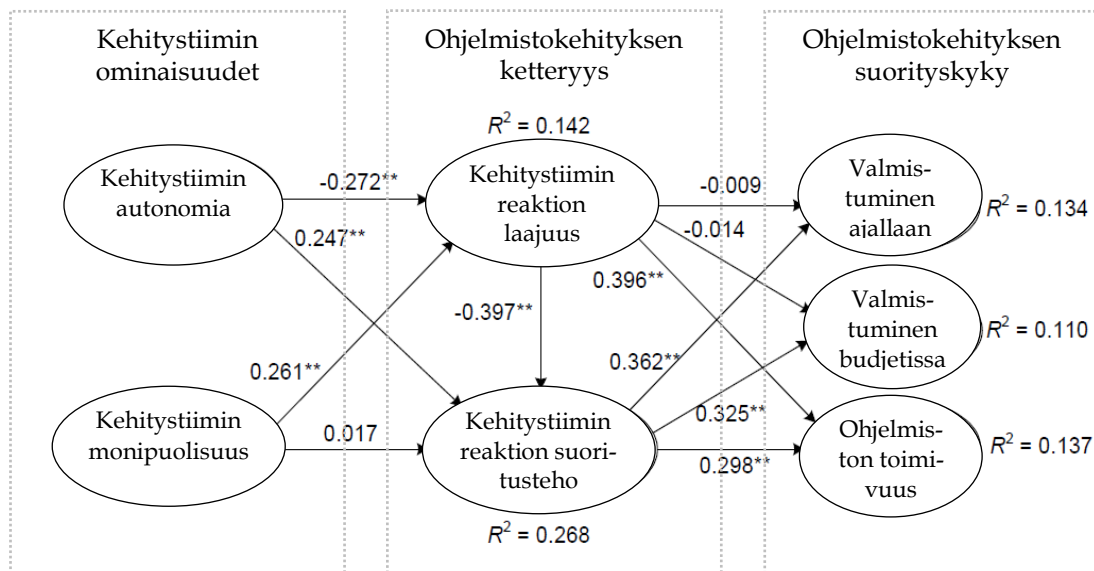
2.2.4 Ketteryyden mittarit

Ketteryys on yhä enenevässä määrin tutkittu ala mutta sen tutkimuksesta puuttuvat edelleen yleisesti hyväksytyt määritelmät ketteryydelle sekä todistettava teoria ketteryyden vaikutuksesta ohjelmistokehitykseen. Teorian ja määritel-

män puuttuessa on myöskin vaikea löytää uskottavaa yleistettävyyttä liittyen mihinkään laadullisen tutkimusten tuloksena tehtyihin päätelmiin. Jotta voitaisi esimerkiksi väittää, että ketteruus parantaa asiakastytyvääsyyttä, olisi ensin voitava määritellä se millainen työskentely on ketterää ja millainen ei, tämä määritelmä pitäisi pystyä soveltamaan erilaisiin ohjelmistokehitystä toteutaviin yhteisöihin ja lopulta mittaamaan sekä yhteisöjen ketteryyden tasoa että näiden asiakkaiden tyytyväisyyden tasoa luotettavasti.

Ketteryyden tutkimuksessa on sovellettu joitain mittareita, joilla ketteryyden vaikutusta ohjelmistokehitykseen on pyritty mittaamaan. Tuottavuutta ketterässä ohjelmistokehityksessä voidaan mitata monilla tavoilla, kuten laskemalla tuotetun koodin määrää suhteutettuna henkilömäärään ja aikaan, mittaamalla valmiin ohjelmiston sisältämien toiminnallisuuksien määrää painottaen kriittisyyttä sekä mittaamalla tuotettua lisäarvoa verrattuna taloudellisiin panoksiin (Johansen ym., 2014). Johansen ym. (2014) käyttivät seitsemää Scrummenetelmään liittyvää indikaattoria selvittämään ketterän kehityksen koettuja hyötyjä. Projektipäällikön kykyä hallita ihmisiä ketterässä ja suunnitteluperustaisessa työskentelyssä on mitattu projektin osa-alueiden kuten asiakassuhteen, vaatimushallinnan ja kulujen suhteen (Ceschi ym., 2005). Abrahamsson ym. (2008) taas ovat käyttäneet mittareidensa lähtökohtana ketterän menetelmän eri työtapoja ja pyrkineet mittaamaan kommunikaation muutosta niiden valossa. Näissä esimerkeissä ei kuitenkaan mitata varsinaisesti ketteryyttä vaan ketterien menetelmien osien vaikutusta valittuun teemaan.

Lee ja Xia (2010) loivat oman määritelmänsä ketteryydelle ja sovelsivat sitä monimenetelmäisessä tutkimuksessa, jonka tarkoituksena oli selvittää riippuvuuksia ja merkityksiä kehitystiimin ominaisuuksien, ketteryyden määritelmän ja projektin lopputuloksen välillä (ks. TAULUKKO 1). Tutkimus aloitettiin luomalla ketteryydelle määritelmä, jossa ohjelmistokehitys on ketterää, mikäli se vastaa muutokseen tehokkaasti kulutettujen resurssien sekä tuotetun ohjelmiston suhteen. Tutkimuksen ensimmäisessä vaiheessa käytettiin alustavia haastatteluita ja muita tekniikoita tutkimusmallin ja mallin taustalla olevan mittausinstrumentin hiomiseksi. Toisessa vaiheessa tehtiin laaja kyselytutkimus IT-alan projektipäälliköille (kyselyn tulokset on merkitty oheiseen malliin, ks. kuvio 1), Määrällisen tutkimuksen jälkeen toteutettiin vielä 17 haastattelua koskien kymmentä kyselyyn osallistunutta projektia sen selvittämiseksi, mistä saadut tulokset johtuvat. Tulostensa pohjalta tutkijat ehdottavat, että projektille pitäisi ensin asettaa tulostavoitteet ajan, budjetin ja tuotoksen suhteen ja sitten kulkea mallissa taaksepäin määritellen kuinka paljon, eri ketteryyden muotoja tarvitaan ja lopulta kuinka autonominen ja monimuotoinen tiimi tarvitaan. (Lee & Xia, 2010.)



Huom * $p < .05$; ** $p < .01$

KUVIO 1: Tutkimustulokset: kehitystiimin ominaisuudet, ketteryyden määritelmä, ohjelmistokehityksen lopputulos (Lee & Xia, 2010)

Conboy (2009) lähestyi tutkimuksessaan ketteryyttä tietojärjestelmien kehittämisessä yleisellä tasolla, luoden ensin kirjallisuuden pohjalta ketteryydelle määritelmän (ks. TAULUKKO 1) ja tutkimalla sen pohjalta tehdyn ketteryyden luokittelun avulla kahta toteutunutta projektia ja niiden työtavoissa toteutunutta ketteryyttä. Luokittelu on kolmeosainen ja se sisältää ehdot ketterälle tietojärjestelmän kehitysmenetelmälle (ks. kuvio 2). Luokittelua käytettiin hyväksi tutkittaessa kahden Scrum/XP-menetelmiä käyttäneen projektin ketteryyttä. Tutkimuksessa huomattiin, että monet ketterien menetelmien käytännöistä eivät todellisuudessa edistäneetkään työskentelyn ketteryyttä mutta toisaalta ketteryyttä voitiin tunnistaa myös sellaisista työtavoista, jotka eivät olleet osana ketterää menetelmää. (Conboy, 2009.)

- 1) Ollakseen ketterä tietojärjestelmäkehityksen osan (mikä tahansa tietty osa menetelmää) pitää edistää vähintään yhtä seuraavista
 - muutoksen luominen
 - toiminta muutoksen aiheuttamiseksi ennen muutosta
 - muutokseen reagointi
 - muutoksesta oppiminen
- 2) Ollakseen ketterä tietojärjestelmäkehityksen osan pitää edistää vähintään yhtä seuraavista eikä se saa vähentää niistä ainuttakaan
 - havaittu taloudellisuus
 - havaittu laatu
 - havaittu yksinkertaisuus
- 3) Ollakseen ketterä, tietojärjestelmäkehityksen osan pitää olla jatkuvasti valmis toisin sanoen minimaalinen aika ja kustannus valmistella komponentti käyttöä varten.

KUVIO 2: Ketteryyden luokittelu Conboyn (2009) mukaan

Tässä luvussa esitellyistä tutkimuksista Conboy (2009) sekä Lee ja Xia (2010) ovat toteuttaneet tutkimusta, joissa pyrittiin mittaamaan ketteryyttä eikä muita siihen liittyviä ilmiöitä. Kummisakin tutkimuksissa tutkijat olivat luoneet ketteryydelle oman määritelmänsä ja kehittäneet tutkimusmallinsa sen päälle. Conboyn luokittelua sovellettiin ainoastaan case-tutkimuksessa eikä sen soveltuvuudesta laajempaan määrälliseen tutkimukseen ole vielä tietoa. Lee ja Xia taas tekivät laajan määrällisen tutkimuksen mutta heidän käyttämänsä ketteryyden määrittely on hyvin erilainen muihin määrittelyihin verrattuna (ks. TAULUKKO 1 ja liite 1).

2.3 Ketteryydellä parempaa ohjelmistokehitystä

Agile Manifesto alkaa suomennettuna seuraavasti: ”Löydämme parempia tapoja tehdä ohjelmistokehitystä, kun teemme sitä itse ja autamme muita siinä.” (Agile Alliance, 2001). Agile Manifeston arvojen ja periaatteiden voidaan nähdä olevan suuntaviivoja juuri tätä parempaa ohjelmistokehitystä kohti. Tämän osion tarkoituksena on käydä läpi aiempien lukujen sisällön valossa tätä väitettä.

Tekeekö yksilöiden ja kanssakäymisen arvostaminen kehityksestä parempaa kuin menetelmien ja työkalujen painottaminen? Johansen ym. (2014) tutkivat tuottavuutta ja huomasivat, että Scrumin käytön koettiin parantavat työntekijöiden työsuoritusta ja vähentävän turhan työn määrää. Abrahamsson ym. (2008) taas tutkivat kommunikaatiota ketteriä menetelmiä käyttävissä tiimeissä ja huomasivat, että menetelmillä oli positiivinen vaikutus erityisesti tiimin sisäisessä viestinnässä. Balijepally ym. (2009) tutkivat pariohjelmointia, jonka tarkoituksena on juuri lisätä jatkuvaa keskustelua ohjelmoijien välillä. Tällä toiminta-

tavalla ei näyttänyt olevan vaikutusta valmiin työn laatuun vaikka työtä tehtiinkin tiiviimmässä yhteistyössä.

Onko ainoastaan toimivaan ohjelmistoon keskittyminen parempi ohjelmistokehityksen kannalta kuin varmistaa myös kattava dokumentaatio? Lyytisen ja Rosen (2006) tutkimus käsitteli ketteriä menetelmiä toivotun tuloksen näkökulmasta ja tutkimuksessaan he löysivät viitteitä siihen, että nopeus, innovatiivinen sisältö, kustannukset, riskit ja laatu vaikuttavat toinen toisiinsa niin, yhden noustessa prioriteetiksi, on muista tingittävä. Jos siis esimerkiksi dokumentaation koetaan olevan laadukkaan ohjelmiston merkki, niin on huomattava, että sen tuottaminen voi vaatia enemmän aikaa ja rahaa sekä sen tekeminen voi viedä aikaa innovatiivisen sisällön luomiselta. Toisaalta France ym. (2005) huomauttavat, että on myös ohjelmistoja, joiden kehittämisen etenemistä ei voida kommunikoida asiakkaalle esimerkiksi käyttöliittymän avulla, jolloin dokumentaatio on arvokas viestinnän väline, mikäli asiakasyhteistyötä arvostetaan.

Takaako asiakasyhteistyö varmemmin parempaa ohjelmistokehitystä kuin sopimusneuvottelujen käyminen? Ceschi ym. (2005) tulivat omassa ketterää ja suunnittelupohjaista menetelmää vertailevassa tutkimuksessaan siihen tulokseen, että ketteristä menetelmistä on erityistä hyötyä asiakassuhteen parantamisessa, sillä yhteistyötä oli enemmän. Abrahamsson, Haikara, Pikkarainen, Salo ja Still (2008) taas totesivat ketterien menetelmien käytön huonontavan kommunikaatiota kehittäjien ja asiakkaiden välillä erityisesti erilaisten odotusten vuoksi. Conboy ym. (2012) sekä van Vliet ja Van Waardenburg (2013) huomasivat tutkimuksissaan, että asiakkaan edustaja, ei aina osallistunut ketterän tiimin toimintaan toivotulla tavalla, mikä taas vuorostaan aiheutti ongelmia.

Onko ohjelmistokehitys parempaa, jos vastataan muutokseen suunnitelmassa pitäytymisen sijaan? France ym. (2005) totesivat kriittisessä arvioinnissaan, että muutosten hinta riippuu menetelmästä riippumatta aina muutoksen vaatiman korjaustyön määrästä. Jos siis projektin lopussa tehdään muutosvaatimus, jonka vuoksi suuria osia ohjelmistosta on uusittava, niin se vie aina aikaa ja tulee näin kalliiksi. Tältä vältytään, jos asiakkaalta saadaan palautetta jatkuvasti siitä, mikä ohjelmistossa tuo tälle lisäarvoa mutta aina ohjelmisto ei ole myöskään sellainen, että se voitaisi asiakkaalle kesken kehityksen ymmärrettävästi selittää. Toisaalta Lee ja Xia (2010) määrittelivät ketterän ohjelmistokehityksen juuri vastaamisena muutokseen tehokkaasti kulutettujen resurssien sekä tuotetun ohjelmiston suhteen. Tällä taas oli positiivisia tilastollisia yhteyksiä projektin lopputulokseen.

Ketteryyteen liittyviä tutkimuksia on julkaisu vuoteen 2015 mennessä jo runsaasti mutta silti vaikuttaa siltä, että vastauksen löytäminen siihen, tekeekö ketteruus ohjelmistokehityksestä parempaa, on vaikea löytää. Tutkimusta tehdään monista eri lähtökohdista eikä vielä mikään yksittäinen tapa mitata ketteryyttä ole saanut suurempaa suosiota.

2.4 Yhteenveto ketteryydestä ohjelmistokehityksessä

Ketteryydelle ei ole olemassa yhtä akateemisen yhteisön yhteisesti hyväksymää määritelmää, mikä vaikuttaa aiheen käsittelyyn ja tutkimukseen. Koska ketteryyden määrittelemine on hankalaa, on myös vaikea määritellä, mitkä ohjelmistokehityksen työmenetelmistä ja niiden soveltamistavoista ovat ketteriä ja mitkä eivät. Ketterien menetelmien käytänteet ovat toisaalta olleet yleinen tapa tutkia ketteryyttä ja suuri osa tutkimuksesta käsittelee ketteryyttä käytettyjen menetelmien eikä esimerkiksi toiminnan pohjalla olevien periaatteiden ta-solla.

Tämän kirjallisuuskatsauksen perusteella ei voida yksiselitteisesti kuvata sitä, miten ketteruus ohjelmistojen ja tietojärjestelmien kehityksessä määritellään. Agile Manifesto on laajasti siteerattu ja tunnettu kuvaus siitä, millaisia asioita ketterässä ohjelmistokehityksessä arvostetaan. Tämä ei kuitenkaan tee julistuksesta yleisesti tutkimuksen parissa hyväksyttyä määritelmää. Toisaalta Conboy (2009) on luonut kirjallisuuden pohjalta oman määritelmänsä tietojärjestelmäkehityksen ketteryydelle ja soveltanut sitä tutkimukseen. Conboyn määritelmässä korostuvat muutos ja lisäarvon luominen kun taas Agile Manifestossa korostuu vastakkainasettelu ”parempien ohjelmistokehityksen tapojen” (Agile Alliance, 2001) ja suunnitelmaperustaisten ohjelmistokehityksen työtapojen välillä. Agile Manifeston ja Conboyn määritelmässä on kuitenkin enemmän samaa kuin eroavaisuuksia ja Agile Manifeston kaksi viimeistä teesiä koskevatkin asiakasyhteistyötä ja vastaamista muutokseen kuten Conboynkin määritelmä.

Agile Manifeston kirjoittajista suuri osa edusti erilaisia ketteriksi kutsuttuja menetelmiä ja Manifesto olikin eräänlainen kokoelma näille työtavoille yhteisiä periaatteita. Eri ketterien menetelmien välillä työohjeet voivat olla hyvinkin erilaisia mutta menetelmistä löytyy myös selvästi yhteistä arvopohjaa, kuten luvussa 2.1.4 tehdystä vertailusta voidaan huomata. Ketteriä menetelmiä myös sovelletaan eri organisaatioissa hyvin eri tavoin ja työtapoja voidaan keräillä eri menetelmistä tai jättää toteuttamatta sellaisia menetelmän osia, jotka omassa organisaatioissa tuntuvat hankalilta toteuttaa. Kun halutaan selvittää, onko yrityksen toiminta ketterää, onkin tärkeämpää selvittää, ovatko yrityksen toiminnan motivaationa olevat periaatteet tai arvot ketteriä, eikä niinkään sitä, sovelletaanko yrityksessä ketteräksi kutsuttua menetelmää.

Ketteryyden tutkimuksen perinne on verrattain lyhyt ja alkaa varsinaisesti vasta 2000-luvun vaihteesta. Koska ketteryydelle ei ole yhtä yleisesti hyväksyttyä määritelmää tai teoriaa, on sen tutkimus myös hyvin kirjavaa. Suuri osa käytetyistä tutkimusmenetelmistä on case-tutkimuksia ja vain harvoissa niistä on sovellettu olemassa olevaa teoriaa. Monissa tutkimuksissa vertailtiin ketterien menetelmien käytäntöjä esimerkiksi ajallisesti aiempaan tilanteeseen tai toiseen ryhmään, joka toteutti samankaltaista tehtävää eri käytännöillä. Yleistettäviä vastauksia esimerkiksi kysymyksiin ketterien menetelmien tai ketterien periaatteiden hyödyistä tai haitoista ei juuri ole saatu. Tämä johtuu sovellettu-

jen teorioiden ja luotujen mittarien vähydestä, minkä vuoksi tutkimuksia toistetaan harvoin uusissa konteksteissa sekä siitä, että harvoissa tutkimuksissa on verrattu useita organisaatioita keskenään.

Ketteryyden tutkimuksen keskeisenä ongelmana on siis teoreettisen tarkkuuden puuttuminen, ketteriksi kutsuttujen käytäntöjen ja varsinaisen ketterän toiminnan rajan hämärtyminen sekä eri organisaatioiden ketteryyden vaikea vertailtavuus. Lee ja Xia (2010) sekä Conboy (2009) ovat luoneet omat ketteryyden määritelmänsä, joita he ovat käyttäneet pohjana ketteryyden mittaamiselle. Näyttää kuitenkin siltä, että muissa tutkimusartikkeleissa viitataan useammin Agile Manifestoon, mikä voi viitata siihen, että sen sisältämät ketteryyteen liittyvät käsitteet ovat yleisemmin hyväksytyjä kuvaamaan ketteryyttä. Kirjallisuuden perusteella voidaankin osin vastata tutkimuskysymykseen; ohjelmistokehityksen ketteryyttä voidaan mitata mutta mittarin pohjalla olevien käsitteiden valinta on tehtävä tarkoin. Jos ketteryyden mittari perustuu käsitteille, joiden merkityksen yleisesti koetaan liittyvän ketterään toimintaan, on mahdollista saada mittaustulokseksi tietoa myös eri organisaatioiden välisistä eroista.

3 KETTERYYDEN MITTARI JA KYSELYTUTKIMUS

Aiemmin toteutetusta ketteryyden tutkimuksesta puuttuu paljolti keinoja yksiselitteisesti verrata ohjelmistoja kehittävien organisaatioiden ketteryyden määrää toisiinsa. Tämän tutkimuksen tarkoituksena on selvittää, millä käsitteillä ketteryyttä voidaan mitata ohjelmistokehityksessä. Lisäksi halutaan selvittää, onko ketteryydellä yhteyttä yrityksen sisäisiin ominaisuuksiin tai niihin markkinoihin, joilla yritys toimii. Tutkittaviksi ketteryyden käsitteiksi on valittu Agile Manifesto ja sen pohjalta luodaan mittari kyselytutkimukselle.

3.1 Tavoite ja tutkimusote

Kirjallisuuden perusteella voidaan huomata, että ketteryydelle ei ole vakiintunut vain yhtä kaiken kattavaa määritelmää vaan erilaisia käsityksiä ketteryydestä ohjelmistokehityksessä on useita, vaikkakaan ne eivät ole toisiaan pois sulkevia. Aiemmissa tutkimuksissa ketteryyttä on mitattu menetelmätasolla ja organisaatioiden ketteryyden eroja ei ole tutkittu paljon. Tämän menetelmäosion tarkoituksena on luoda ketteryyden määritelmän (Agile Manifesto) pohjalta mittari, joka mittaa yrityksen toiminnan ketteryyttä. Tutkimuksella halutaan selvittää, ovatko suomalaiset ohjelmistoyritykset ketteriä ja selittääkö ketteryyden mitattua määrää jotkin yrityksen ominaisuudet. Eroja ketteryydessä halutaan selvittää yrityksen sisäisten ominaisuuksien ja sen markkinoiden näkökulmasta. Ketteryyttä tarkastellaan myös osissa sen eri arvojen mukaisesti.

Tässä tutkimuksessa pyritään selvittämään, voidaanko organisaation ohjelmistokehityksen ketteryyttä mitata Agile Manifeston käsitteillä ja toisaalta voidaanko organisaatioiden ketteryyden eroja löytää. Tämän vuoksi ketteryyden mittari on luotava niin, että se on yleispätevä ja soveltuu monien eri organisaatioiden tutkimiseen. Koska tarkoituksena on tilannekuvan tekeminen yritysten eroista, ei tutkimuksessa tarvitse ottaa huomioon ohjelmistokehitysmenetelmien muutosta eri aikoina havaintoyksiköissä. Tämän vuoksi tutkimusmenetelmäksi valikoitui kvantitatiivinen survey-tutkimus.

Agile Manifesto alkaa sanoilla ”Löydämme parempia tapoja tehdä ohjelmistokehitystä, kun teemme sitä itse ja autamme muita siinä.” (Agile Alliance, 2001) eli julistuksen mukaan ketteryys siis selittää parempaa ohjelmistokehitystä. Tässä tutkielmassa tutkitaan Agile Manifeston arvojen mukaisen toiminnan läsnäoloa ohjelmistokehityksessä mutta ei vielä arvioida ohjelmistokehityksen laatua. Manifeston arvojen kehittämisen pohjalla olleet työtavat ovat niin laajasti levinneet ja tutkitut, että Agile Manifestoa voidaan pitää määritelmänä ketteryydelle. Tutkittavan ilmiön ollessa näin selvästi rajattu sisällöltään, on sen tutkimiseen valittava teoriaa testaava tutkimusote. Kysely sopii tällaiseen tutkimusaiheeseen hyvin, sillä kysymyksiä on vähän ja vastaajia tarvitaan paljon, jotta voidaan tehdä havaintoja tilastollisin menetelmin. Valmiit vastausvaihtoehdot sopivat tutkimukseen siksi, että Agile Manifesto sisältää valmiin ketteryyden luokituksen eli neljä osa-aluetta, joissa kaikissa on vastakohtapari. (Järvinen & Järvinen, 2011.)

3.2 Mittarin kehittäminen

Churchill (1979) on esittänyt kahdeksanvaiheisen metodin mittarin kehittämiseksi (Järvinen & Järvinen, 2011), jota seuraten ketteryyden mittarin luomisen vaiheet tässä esitellään. Ensimmäisessä vaiheessa tulee määrittää yläkäsitteen ala. Tässä tutkimuksessa tarkoituksena oli tutkia ketteryyttä ja tämä vaati ketteryyden käsitteen määrittelyä. Tähän tutkimukseen määrittelyksi valittiin Agile Manifesto sillä se määrittelee selvästi ketterään ohjelmistokehitykseen liittyvän toimintatavan sekä sen vastakohtan. Näin ollen tutkittavan käsitteen ala voitiin rajata Agile Manifestossa esitettyihin vastapareihin olettaen, että Manifeston kuvaamat arvostettavat asiat todella kuvaavat ketteryyttä ohjelmistokehityksessä. Agile Manifesto myös sisältää valmiiksi neljä väittämää ääripäineen eli erityistä osien generointia ei tarvinnut tehdä.

Ketteryyttä koskeva kysymyspatteri rakennettiin 16 väittämästä – neljä väittämää koskien kutakin Agile Manifeston neljää arvoa: (a) yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja, (b) toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota, (c) asiakasyhteistyötä enemmän kuin sopimusneuvotteluja, (d) vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa. Väitteissä pyrittiin käyttämään samaa sanastoa kuin Agile Manifestossa ja niissä pyrittiin myös huomioimaan toiminnan tuottavuuteen ja tulevaisuuden suunnitteluun liittyvät näkökulmat. Väittämiin annettiin vastausvaihtoehdot, joissa vastaajan tuli arvioida kuuluvatko hänen yrityksensä työtavat lähemmäs ketterää vai suunnitelmapohjaista ohjelmistokehitystapaa. Ääripäiden välillä oli viisikohtainen asteikko, jossa keskellä oli neutraali vaihtoehto ja vastaaja pystyi myös jättää vastaamatta yksittäisiin väitteisiin. Eri arvoja koskevat kysymykset sekoitettiin keskenään ja osa vastausvaihtoehdoista käännettiin ääripäiden suhteen päinvastaisiksi. Liitteenä olevaan kysymyslomakkeeseen on lisätty luettavuuden parantamiseksi kysymyksille juokseva numerointi ja arvoa koskeva kirjain (ks. liite 2).

Churchillin (1979) mukaan seuraavaksi tulisi kerätä havaintoaineisto pieneltä koehenkilöjoukolta, jotta saataisiin selville, kuuluvatko kehitetyt kysymykset todella yläkäsitteen alaan. Tätä vaihetta ei toteutettu näin useastakin syystä. Ensinnäkin Ohjelmistoyrityskartoitus on kohdistettu kaikille Suomalaisille ohjelmistoalan yrityksille, eli koeorganisaatiot olisi etsittävä ulkomailta sellaisen koejoukon löytämiseksi, jolle lopullista kyselyä ei lähetettäisi ja tällöin taas kulttuuriaspekti olisi pitänyt huomioida koko kyselyssä. Toisaalta, koska kysymykset olivat osa suurempaa tutkimusprojektia tuli niiden kehittäminen tapahtua projektin aikataulun mukaisesti. Näin ollen kysymysten validiteettia testattiin pyytämällä ketteryteen perehtyneitä tutkijoita ja yrittäjäjärjestön edustajia yhdistämään kyselyn väittämät Agile Manifeston neljään arvoon. Tällä tavalla pyrittiin selvittämään sitä, voidaanko väittämien olettaa kuvaavan yläkäsitettä eli ketteryyttä. Saadun palautteen perusteella osaa väittämistä muutettiin niin, että ne selkeämmin kuvaisivat vastaavaa Agile Manifeston arvoa. Tästä syystä mittaria ei myöskään voitu pudistaa tilastollisin menetelmin.

Tässä luodun mittarin rakennevaliditeettia on jokseenkin hankala arvioida, sillä se mittaa laadullisia ominaisuuksia, joista eksaktin todellisen tiedon saaminen on hankalaa. Siksi mittari sopiikin vain vastanneiden yritysten välisten erojen tutkimiseen. Toisaalta mittarin tuloksia ei voida myöskään verrata aiempien mittarien tuloksiin, sillä vertailuun soveltuvia mittareita ei löytynyt kirjallisuuskatsauksen yhteydessä. Näin ollen Churchillin vaatimus normien kehittämisestä mittaustulosten perusteella on jäätävä seuraavaan tutkimukseen. (Järvinen & Järvinen, 2001.)

3.2.1 Kysely ja muuttajat

Varsinainen havaintoaineisto kerättiin osana Ohjelmistoyrityskartoitusta, joka sisälsi myös muita kuin ketteryteen liittyviä kysymyksiä. Kysely oli yhteensä kahdeksansivuinen, josta ketteryttä käsitteli viimeinen kysymysosio. Ketteryyttä mittaavat väitteet oli sijoitettu samalle sivulle kyselyssä ja ne oli sekoitettu aihealueiltaan ja ketterä-suunnitelmaperustainen ääripäiden suhteen.

Ketteryyttä koskevia väitteitä oli 16 ja niiden vastaukset annettiin viisiporraisella asteikolla, jonka toisessa päässä oli ketterä ja toisessa päässä suunnitelmaperustainen vastaus. Näistä tehtiin summamuuttujia, joita voidaan analysoida ordinaalimuuttujina. Selitystä ketteryden ilmenemiselle tutkitaan kolmen yrityksen markkinoita kuvaavan nominaalimuuttujan ja neljän yrityksen sisäisiä ominaisuuksia kuvaavan intervallimuuttujan avulla. Nämä kysymykset sijaitsivat kyselyn alussa.

Ketteryyden taustalla on ajatus tuotekehityksestä, joka tapahtuu markkinoiden ehdoilla. Tämän vuoksi ketteryttä tarkastellaan kolmen yrityksen kohdemarkkinoita koskevan selittävän muuttujan valossa. Kaikki muuttajat ovat kahden tai kolmen ryhmän nominaalimuuttujia, sillä tarkemmalla ryhmittelytasolla jotkin ryhmät saattoivat jäädä liian pieniksi tilastollisen analyysin näkökulmasta. Liiketoimintamalli-muuttuja koostuu kahdesta ryhmästä: tuoteyritykset ja palveluyritykset. Päätuote-muuttuja erottelee yritykset päätuotteen

mukaan kuluttaja- ja yrityssovelluksia tuottaviin yrityksiin sekä yrityksiin joilla ei ole tuotteita. Kansainvälinen-muuttuja koostuu kahdesta ryhmästä, joista toisella on kansainvälistä liiketoimintaa ja toisella ei.

Ketteryyden yhteyttä yrityksen ominaisuuksiin on myös mielenkiintoista tarkastella. Ketteryyden määrää tutkitaan verrattuna yrityksen ikään, joka on laskettu yrityksen perustamisvuodesta. Tämä tieto saatiin kyselyn ulkopuolisesta tietokannasta. Yrityksen koko voidaan määritellä henkilöstön tai liikevaihdon perusteella mutta näistä ensimmäinen on mielekkäämpi tämän tutkimuksen yhteydessä. Ketteryyden arvot liittyvät juuri henkilöstöön eikä yrityksen kaupankäynnin määrä kerro sen organisaatiosta. Henkilöstön määrä pyydettiin ilmoittamaan vuoden 2013 lopun tilanteen mukaan. Ketteryyteen on myös liitetty monia väitteitä sen eduista, kuten tämän tutkielman alussa todettiin. Tämän vuoksi on mielenkiintoista myös verrata yrityksen ketteryyttä sen kannattavuuteen. Yrityksille laskettiin intervalliasteikollinen Kannattavuusprosentti-muuttuja käyttämällä näiden ilmoittamia tietoja vuoden 2013 liikevaihdosta ja liikevoitosta. Ketteryyden yhteyttä yrityksen kasvuun taas tutkitaan Liikevaihdon muutos -muuttujalla. Tiedot on haettu Orbis-tietokannasta (Bureau van Dijk, 2015) ja niiden yritysten kohdalla, joista tietokantatietoa ei ollut, käytettiin näiden itse ilmoittamia tietoja. Muuttujan arvo on liikevaihdon muutosprosentti kolmen vuoden ajalta.

3.2.2 Aineiston kerääminen

Tämän tutkimuksen aineisto kerättiin osana Ohjelmistoyrityskartoitusta kesällä 2014. Tutkimuksen toteuttaneet tutkijat olivat Aalto ja Jyväskylän yliopistoista ja lisäksi yhteistyökumppaneina olivat Ohjelmistoyrittäjät ry, Teknologiateollisuus ry sekä TIVIA ry. Ohjelmistoyrityskartoitus on toteutettu lähes vuosittain vuodesta 1997 lähtien ja kyselyn kysymykset muodostuvat vuosittain samankaltaisina pysyvistä taustakysymyksistä sekä vaihtuvista teemoista. Vuoden 2014 teemana oli joustavuus ja ketteryyden mittaaminen oli osa tätä teemaa. (Luoma, 2014.)

Ketteryyttä koskevat kysymykset olivat osa laajempaa kyselylomaketta ja myös muut tässä tutkielmassa käytetyt muuttujat perustuvat kyselyssä annettuihin vastauksiin. Kysely lähetettiin vastaajille postitse ja tieto kyselystä lähetettiin myös sähköpostilla. Lomake tarjottiin vastaajille suomeksi, ruotsiksi ja englanniksi. Kyselyn kohderyhmänä oli yrityksen ylin johto eli postituslistat koostuivat lähinnä toimitusjohtajista nimen ollessa tiedossa. Kyselyyn oli mahdollista vastata palauttamalla paperilomake tai vaihtoehtoisesti täyttämällä vastaava lomake verkossa. Päällekkäisiä ja asiattomia vastauksia valvottiin yrityskohtaisin tunnistenumeroin, jotka toimitettiin vain kohderyhmänä olleille yrityksille. Kysely lähetettiin 5050 suomalaiselle ohjelmistoyritykselle (toimialakoodi 62) ja vastauslomakkeita palautettiin 500.

3.3 Aineiston käsittely

Kaikki kyselyyn saadut vastaukset eivät olleet tämän tutkimuksen kannalta hyväksyttäviä. Osassa palautetuista lomakkeista kehitysmenetelmiä koskeviin kysymyksiin ei vastattu, joten nämä vastaukset karsittiin pois. Ketteryyttä koskevista kysymyksistä pyrittiin muodostamaan summamuuttujia niin, että käytössä olisi kaikista kysymyksistä koottu muuttujia sekä pienemmistä kysymysryhmistä jokaista ketteryyteen liittyvää arvoa kuvaava muuttuja. Summamuuttujien muodostamisessa käytettiin apuna faktorianalyysia, Cronbachin alfoja ja vieraita havaintoja (outliers) etsittiin boxplot-analyysin avulla.

Kysymyksissä esitetyt väittämät eivät välttämättä sopineet hyvin mikroyrityksille tai ainakaan yhden henkilön yrityksille. Erityisesti Agile Manifeston ensimmäinen kohta, jossa arvostetaan kanssakäymistä, oli muodostettu kysymyksiksi, joissa oletettiin, että ohjelmistoa kehittää enemmän kuin yksi ihminen kerralla. Väittämät oli myös aseteltu niin, että niissä oletettiin yrityksellä jo olevan jonkinlaisia käytäntöjä. Tämän vuoksi aineistoon eivät sovi myöskään aivan uudet yritykset. Lopulta mukaan otettiin siis vain vähintään kahden henkilön yritykset, jotka olivat toimineet vähintään kaksi vuotta.

Väitteille tehtiin Harman single factor -testi kyselymetodin luoman vääristymän löytämiseksi (Common method bias /variancy). Kaikkien kysymysten kohdalla yksi rotatoimaton faktori selitti 24,1 % vaihtelusta. Tarkoittaa, että yksi faktori selittää vain noin neljäsosan varianssista ja tämän testin pohjalta näyttääkin siltä, ettei mittausinstrumentti itsessään aiheuta liikaa vääristymää vastauksiin.

Kaikille 16 väitteelle tehtiin pääkomponenttianalyysi (Principal Component Analysis), jonka tarkoituksena oli vähentää muuttujia varsinaista analyysiä varten. Analyysissä faktorien määrää ei rajoitettu vaan eigenarvon rajaksi valittiin 1. Rotaatiometodina käytettiin Obliminia (Kaiser Normalization). Analyysin perusteella löydettiin neljä faktoria (ks. TAULUKKO 3). Ensimmäiseen komponenttiin kuuluivat väitteet 8, 13, 16, 6, 14 ja 1 joista kuitenkin viimeisen kommunaliteetti oli niin alhainen (.302), ettei sitä kannata ottaa summamuuttujaan. Toiseen komponenttiin tulivat väitteet 9, 7 ja 2, jotka kaikki liittyivät dokumentaatioon. Kolmanteen komponenttiin kuuluivat väitteet 12, 10, 4 ja 3, joista 4 ei liittynyt yksilöitä ja kanssakäymistä koskevaan arvoon. Neljänteen komponenttiin tulivat väitteet 11, 15 ja 5 näiden kommunaliteetit jäivät kuitenkin alhaisiksi eivätkä kaikki liittyneet samaan arvoon.

Pääkomponenttianalyysin perusteella löydettyjen komponenttien sisäistä eheyttä voidaan myös tutkia Cronbachin alfojen avulla. Ensimmäisen komponentin viiden väittämän (8, 13, 16, 6, 14) Cronbachin alfa oli ,789. Väitteen 1 mukaan ottaminen olisi huonontanut alfaa. Toisen komponentin (9, 7, 2) alfa oli ,692. Kolmannen komponentin (12, 10, 3) alfa oli ,685. Coleman ja Multon (2010) ovat muiden muassa esittäneet, että sopiva raja hyvälle Cronbachin alfalalle olisi $>.700$. Tähän kuitenkin vaikuttaa tarkasteltavien yksiköiden määrä, joka tässä jää vähäiseksi sillä osa kysymyksistä jouduttiin jättämään pois. Toisaalta

Coleman ja Multon (2010) myös toteavat, että riittävä alfa riippuu suurelta osin siitä, mihin tuloksia käytetään. Heidän mukaansa alfa, joka on .600-.800 voi olla riittävä, kun tarkastellaan ryhmien välisiä eroja, kuten tässä tutkimuksessa on tarkoitus eli toinen ja kolmas komponentti voidaan hyväksyä. Neljännen komponentin alfa kuitenkin jää vain ,374:ään, mikä on jo liian alhainen ja lisäisy komponentin hylkäämiselle.

TAULUKKO 3: Ketteryysväitteiden pääkomponenttianalyysin tulokset

Kysymys	Komponentti				Kommunaliteetti
	1	2	3	4	
8 C	,788	-,056	-,159	,179	,617
13 D	,712	,123	,141	-,325	,655
16 C	,600	,094	,254	,015	,573
6 B	,565	,268	,057	,116	,530
14 C	,553	,278	,055	,149	,540
1 D	,446	-,081	,202	,063	,302
9 B	-,040	,843	-,154	-,102	,667
7 B	,186	,714	-,092	,183	,662
2 B	,196	,644	,177	-,069	,578
12 A	-,022	,029	,832	-,105	,680
10 A	,236	-,177	,635	,128	,575
4 C	-,003	,029	,628	,267	,512
3 A	,142	-,039	,612	-,222	,458
11 D	,195	-,120	-,116	,680	,503
15 D	,022	,100	,078	,595	,412
5 A	-,217	,360	,358	,416	,495

Tässä tutkimuksessa käytettävät muuttujat on esitetty seuraavassa taulukossa (ks. TAULUKKO 4). Ketteryyttä koskevia muuttujia on kolme, yrityksen markkinoita koskevia muuttujia kolme ja yrityksen ominaisuuksia koskevia muuttujia neljä.

TAULUKKO 4: Muuttujien kuvaukset

Muuttujan nimi	Kuvaus	Tyyppi
Kanssakäyminen	kolme väitettä (12, 10, 3), jotka koskevat arvoa "yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja" α ,685	Ordinaali
Dokumentaatio	kolme väitettä (9, 7, 2), jotka koskevat arvoa "toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota" α ,692	Ordinaali
Muutosvalmius	kolme väitettä (8, 16, 14), jotka koskevat arvoa "asiakasyhteistyötä enemmän kuin sopimusneuvotteluja", yksi väite (13) koskien arvoa "vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa" sekä yksi väite (6), joka koski vastaamista muutokseen dokumentaation kannalta. α ,789	Ordinaali
Liiketoimintamalli	Tuoteyritykset ja palveluyritykset	Nominaali
Päätuote	Kuluttaja- ja yrityssovellus tai ei tuotetta (liiketoimintamallin mukaisesti palveluyritys voi myös tarjota tuotteita)	Nominaali
Kansainvälinen	on / ei kansainvälistä liikevaihtoa	Nominaali
Ikä	Yrityksen ikä vuosina	Riippumaton Intervalli
Henkilöstömäärä	Yrityksen ilmoittama henkilöstön määrä	Intervalli
Kannattavuus	liikevoitto/liikevaihto * 100	Intervalli
Liikevaihdon muutos	Yrityksen liikevaihdon muutos kolmen vuoden ajalta prosentteina, tiedot on yhdistetty Orbis-tietokannasta ja yritysten ilmoittamista tiedoista	Intervalli

3.3.1 Varsinaisista analyysimenetelmistä

Järvinen ja Järvinen (2011) luettelevat parametrisille menetelmille seuraavat ehdot:

- havainnot ovat toisistaan riippumattomia
- havainnot on tehty normaalisti jakautuneesta populaatiosta
- havaintopopulaatioiden varianssit ovat samat
- muuttujien mitta-asteikot ovat ainakin intervalliasteikkoja (ei nominaali, eikä ordinaaliasteikkoja)

Tämän listan viimeinen ehto ohjaa suoraan käyttämään jakaumasta riippumattomia testejä. Ketteryyden summamuuttujat näyttävät intervalliasteikollisilta mutta niiden tiedot on kerätty ordinaaliasteikollisilla vastausvaihtoehdoilla.

Yrityksen markkinoita kuvaaville nominaalimuuttujille tulee valita jakaumasta riippumaton testi selvittämään yrityksen ominaisuuksien vaikutusta ketteryuteen. Kaikille esitetyille muuttujille sopii Kruskal-Wallis-testi. Tämä soveltuu normaalisti jakautumattomille ja erikokoisille ryhmille silloinkin, kun ryhmiä on enemmän kuin kaksi (päätuote-muuttuja). Testi tulee tehdä kuitenkin samanlaista jakaumaa noudattaville joukoille, mitä testattiin jakaumasta riippumattomalla Levenen testillä, joka käyttää järjestysnumeroita ryhmien vertailuun. Liiketoimintamallin jakauma oli yhtenevä kaikilla ketteryuden muuttujilla (kanssakäyminen sig. ,395, dokumentaatio sig. ,715, muutosvalmius sig. ,383). Päätuotteen ryhmien välillä jakaumat olivat myös riittävän yhtenevät (kanssakäyminen sig. ,764, dokumentaatio sig. ,673, muutosvalmius sig. ,965). Kansainvälinen-muuttujan kohdalla ryhmien jakaumien väliset erot olivat myös riittävän vähäisiä (kanssakäyminen sig. ,197, dokumentaatio sig. ,904, muutosvalmius sig. ,691).

Yrityksen ominaisuuksia kuvaavien intervallimuuttujien (ikä, henkilöstö, kannattavuus, liikevaihdon muutos) normaalijakaumaa tutkittiin myös Shapiro-Wilk testillä. Kaikkien p-arvo oli ,000 eli muuttujat eivät ole missään nimessä normaalijakautuneita.

4 TULOKSET

Tutkimuksen tulokset esitellään ensin kuvaamalla vastaajajoukko ja ketteryyssuuttujien tulokset. Seuraavaksi käsitellään ketteryyttä yrityksen markkinoiden ja sisäisten ominaisuuksien näkökulmasta.

4.1 Vastaajajoukko ja vastaukset väitteisiin

Kysely oli kohdistettu kaikille Suomen ohjelmistoalan yrityksille (toimialaluokitus 62 - Ohjelmistot, konsultointi ja siihen liittyvä toiminta). Kyselylomake lähetettiin kirjeenä yhteensä 5050 (suomen ja ruotsin kieliseen) yritykseen toimitusjohtajan nimellä sen ollessa tiedossa. Kaikki lomakkeet eivät kuitenkaan tavoittaneet yrityksiä vaan palautuivat lähettäjälle. Kyselyyn saatiin yhteensä 500 vastausta, joista osa ei täyttänyt tämän tutkimuksen vaatimuksia. Ainoana kannustimena vastaamiseen oli yritykselle annettu mahdollisuus saada sähköpostitse raportti yrityksen vastauksista suhteessa muihin vastaajiin.

Tähän tutkimukseen hyväksyttiin ainoastaan vastaukset, joissa kaikkiin ohjelmistokehitysmenetelmiä koskeviin kysymyksiin oli vastattu. Lisäksi vastaajista karsittiin alle kaksivuotiaat ja yhden hengen yritykset. Yhteensä vastauksia jäi aineistoon 157.

Ketteryyttä koskevien 16 väitteen vastaukset annettiin asteikolla 1-5, jossa 1 vastasi suunnitelmaperustaisinta ja 5 ketterintä ääripäätä. Seuraavassa taulukossa (TAULUKKO 5) on esitetty kaikki väitteet sekä niiden keskiarvot. Taulukossa on merkitty myös muuttujiin kuuluvien väitteiden kohdalle se summa-
muuttuja, johon väite kuuluu myöhemmässä analyysissä. Suluissa on merkitty niiden arvojen kirjaimet, joita varten kysymys luotiin (a: yksilöitä ja kansakäymistä enemmän kuin menetelmiä ja työkaluja, b: toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota, c: asiakasyhteistyötä enemmän kuin sopimusneuvotteluja, d: vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa) (Agile Alliance, 2001).

TAULUKKO 5: Ketteryyttä koskevat väitteet ja niiden vastausten keskiarvot

Numero ja muuttuja	Ketterä väite	Suunnitteluperustainen väite	Keskiarvo
Kehitystiimimme työtavat on järjestetty niin, että...			
1 (d)	...tiimit ovat jatkuvasti valmiita muutoksiin vaatimuksissa.	...vaatimukset voidaan toteuttaa alussa sovitusti muutospaineista huolimatta.	3,68
2 Dokumentaatio (b)	...dokumentaation tekeminen on mahdollisimman vähäinen osa työtehtäviä.	...dokumentaation tekeminen on keskeinen osa työtehtäviä...	3,25
3 Kanssakäyminen (a)	...sitä ohjaavat tiimin jäsenten yhdessä sopimat työtavat ja tavoitteet.	...sitä ohjaavat yrityksessä käytettävät toimintatavat ja johdon asettamat tavoitteet.	3,45
Kehitystiimimme kommunikoivat pääasiassa...			
4 (c)	...asiakkaan kanssa helpoimmalla mahdollisella tavalla aina tarvittaessa.	...asiakkaan kanssa sovitun toimintamallin ja aikataulun mukaisesti.	3,66
5 (a)	...tiimin jäsenten kesken vapaasti haluamallaan tavalla.	...tiimin jäsenten kesken yrityksessä määritellyjä keinoja sovitusti käyttäen.	3,66
Ohjelmistokehitysprojektimme todennäköisemmin...			
6 Muutosvalmius (b)	...epäonnistuu, jos toimimme liikaa suunnitelmien ja dokumentaation ehdoilla muuttuviin tilanteisiin reagoimisen sijaan.	...epäonnistuu, jos teemme liikaa muutospäätöksiä suunnitelmassa pitäytymisen ja dokumentoinnin sijaan.	3,37
7 Dokumentaatio (b)	...onnistuu, koska toimivan ohjelmiston lisäksi syntyy vähäinen dokumentaatio.	...onnistuu, koska toimivan ohjelmiston lisäksi syntyy kattava dokumentaatio.	3,00
Ohjelmistokehitysprojektiemme tuottavuuden kannalta on tärkeää, että...			
8 Asiakasyhteistyö (d)	...projekti voi vastata asiakkaan muuttuviin tarpeisiin kehitystyön aikana.	...projekti voi vastata tilaukseen asiakkaan alkuperäisen toiveen mukaisesti.	3,85
9 Dokumentaatio (b)	...projekti keskittyy vain ohjelmiston tuottamiseen.	...projekti tuottaa sekä toimivan ohjelmiston että laadukkaan dokumentaation.	2,79
Kehitystiimien työntekijöiden...			
10 Kanssakäyminen (a)	...työtehtävät valikoituvat heille oman osaamisen mukaan joustavasti.	...työtehtävät valikoituvat heille työntekijäprofiilin mukaan esimiehen päätöksellä.	3,92
11 (d)	...osaamisen kehittämisessä keskitytään heidän kykynsä reagoida muuttuviin tarpeisiin.	...osaamisen kehittämisessä keskitytään heidän kykynsä suunnitella kattavasti ja ennakoivasti.	3,15
12 Kanssakäyminen (a)	...työ on tuottavampaa, kun he muodostavat omat työtapansa yhdessä tiimin kanssa.	...työ on tuottavampaa, kun he käyttävät yrityksen yhteisiä toimintatapoja.	3,54

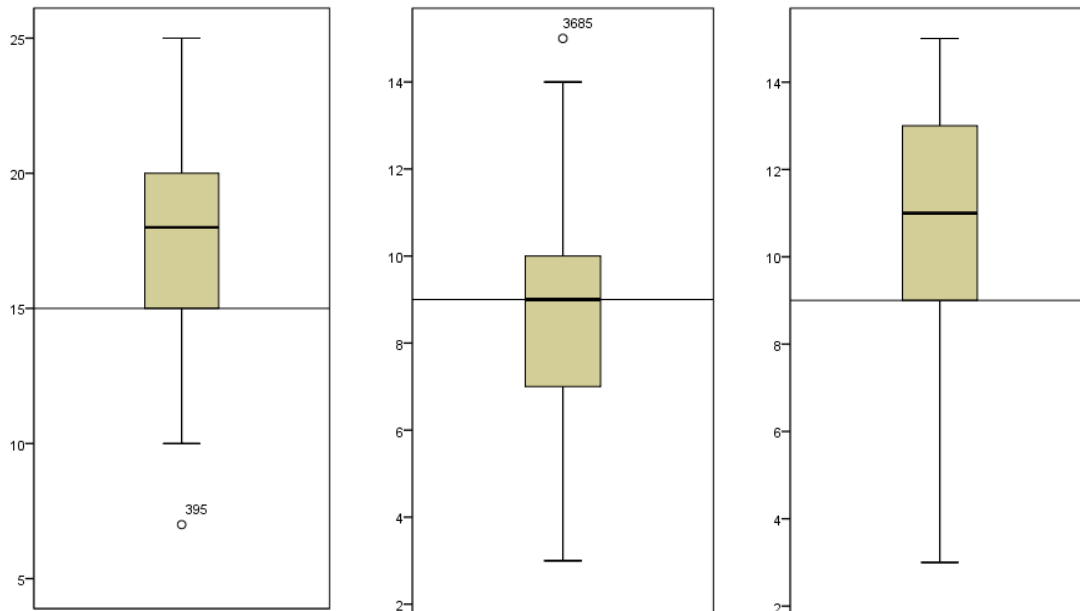
(jatkuu)

Taulukko 5 (jatkuu)

Projekteillemme on tunnusomaista, että...			
13 Muutosvalmius (d)	...ne ovat tuottavampia, kun muutamme suunnitelmiamme joustavasti muuttuvien vaatimusten mukaisesti kehitystyön aikana.	...ne ovat tuottavampia, kun laadimme projektin alussa toimivat suunnitelmat ja pidämme niistä kiinni.	3,52
14 Asiakasyhteistyö (c)	...projektin tavoite voi muuttua kesken projektin asiakkaalta saadun palautteen perusteella.	...projektin tavoite on määritelty etukäteen asiakkaan kanssa projektisuunnitelmaan ja toimitamme suunnitellun ohjelmiston.	3,47
15 (d)	...ne valitaan toteutettaviksi pääasiassa juuri sen hetkisten mahdollisuuksien ja tarpeiden mukaisesti.	...ne valitaan toteutettaviksi pääasiassa pidemmän aikavälin strategisten suunnitelmien mukaisesti.	3,23
16 Asiakasyhteistyö (c)	...projektin lopputulokset sovitaan pääpiirteittäin jättäen tilaa asiakkaan kanssa sovittaville muutoksille.	...projektin lopputulokset sovitaan tarkasti määritellen projektin kulku ennen sen käynnistämistä.	3,82

Väittämien asteikossa numero kolme oli neutraali ja yhtä kaukana ketterästä ja suunnitteluperustaisesta. Ainoa lähemmäs suunnitteluperustaista kuin ketterää osunut keskiarvo oli väitteellä 9, joka koski dokumentaatiota. Lähimpänä neutraalia keskiarvoa (+- 0,3) olivat väitteet 2 (dokumentaatio), 7 (dokumentaatio), 9 (dokumentaatio), 11 (ei mukana muuttujissa), 15 (ei mukana muuttujissa). Näistä kolme ensimmäistä dokumentaatiota koskevaa kysymystä sopivat Cronbachin alfojen mukaan yhteen ja niistä muodostettiinkin summamuuttuja. Kaksi muutosvalmiutta koskevaa kysymystä taas eivät olleet lainkaan mukana summamuuttujassa. Kaikkein lähimpänä ketterää asteikon päätä olivat keskiarvot väitteissä 8 (asiakasyhteistyö), 10 (kanssakäyminen) ja 16 (asiakasyhteistyö).

Ketteryyttä koskevista väitteistä muodostettiin kolme summamuuttujaa faktorianalyysin ja Cronbachin alfojen avulla. Seuraavat viiksilaatikkodiagrammit (KUVIO 3) näyttävät summamuuttujien jakauman. Kuvioihin on piirretty neutraali keskikohta jakauman hahmotuksen helpottamiseksi. Kanssakäyminen ja muutosvalmiusmuuttujien jakaumat painottuvat selvästi keskikohdan yläpuolelle eli 75 % summatuista vastauksista on ketterän puolella. Dokumentaatiomuuttuja taas jakautuu neutraalin keskikohdan ympärille niin, että keskiarvoltaan enemmän suunnitelmaperustaisia vastauksia oli enemmän kun taas ketterälle puolelle osuvat vastaukset olivat lähimpänä neutraalia.



Muutosvalmius

Dokumentaatio

Kanssakäyminen

KUVIO 3: Viiksilaatikkodiagrammit ketteryyden summamuuttujille (muutosvalmius, dokumentaatio, kanssakäyminen)

4.2 Ketteryys ja yrityksen ominaisuudet

Tutkimuksen tulokset asettuvat kahden ääripään välille kuten Agile Manifestossa esitetyt neljä arvoa ketterälle ohjelmistokehitykselle. Seuraavissa tuloksissa käsitellään vain muodostettuja summamuuttujia, sillä yksittäinen kysymys ei voi antaa luotettavaa kuvaa ketteryydestä. Ketteryyden yhteyttä tutkitaan ensin suhteessa yrityksen markkinoihin liittyviin muuttujiin ja sen jälkeen suhteessa yrityksen sisäisiin ominaisuuksiin.

Ketteryyden yhteyttä yrityksen ominaisuuksiin tutkittiin jakamalla yritykset eri tavoin ryhmiin sen markkinoiden ominaisuuksien perusteella. Liiketoimintamalli-muuttujassa ryhmät olivat tuotteita tarjoavat yritykset (N 75) ja palveluyritykset (N 80). Kansainvälinen muuttuja jakaa yritykset niihin, joilla on säännöllistä tai satunnaista kansainvälistä liikavaihtoa (N 91) ja niihin, joilla ei ollut sitä vastaushetkellä (N 63). Päätuotemuuttuja jakoi yritykset kolmeen ryhmään: kuluttajasovellukset (N 12), yrityssovellukset (N 113) sekä yritykset, joilla ei ollut tuotteita (N 31).

Liiketoimintamalli-, kansainvälinen- ja päätuotemuuttujat testattiin Kruskal-Wallis-testillä, jossa nollahypoteesina oli ryhmien havaintojen sijaintilukujen yhdenmukainen jakautuminen. Testit tehtiin jokaisen ketteryyden muuttujan ja markkinoihin liittyvän muuttujan kanssa erikseen.

Tarkastellaan ensin liiketoimintamallin yhteyttä ketteryyden muuttujiin. Kaikkien muuttujien (muutosvalmius sig. ,398, dokumentaatio sig. ,723 ja kanssakäyminen sig. ,195) kohdalla erot ryhmien välillä eivät olleet tarpeeksi suuria ollakseen tilastollisesti merkitseviä. Kansainvälisyysmuuttujan kohdalla eroja ei

myöskään löytynyt minkään ryhmien väliltä (muutosvalmius sig. ,891, dokumentaatio sig. ,747, kanssakäyminen sig. ,555). Yrityksen päätuotteen kohdalla eroa ei löytynyt dokumentaation (sig. ,460) tai kanssakäymisen kohdalla (sig. ,548). Sen sijaan muutosvalmius-muuttujan kohdalla ryhmien välinen ero oli tilastollisesti merkitsevä (sig. ,036). Tarkasteltaessa päätuotemuuttujan ryhmiä erikseen, voidaan huomata, että merkitsevä ero löytyy ryhmien "ei tuotetta" ja "yritystuote" väliltä (p-arvo: asymp. sig. 2-tailed ,020).

Ketteryyden muuttujia verrattiin myös yrityksen sisäisiin ominaisuuksiin. Seuraavassa taulukossa (ks. TAULUKKO 6) on esitetty intervallimuuttujien järjestyskorrelaatiot ketteryysmuuttujien kanssa. Yrityksen ikä korreloi merkitsevästi negatiivisesti kaikkien kolmen ketteryyttä koskevan muuttujan kanssa. Henkilöstömäärä korreloi negatiivisesti merkitsevästi muutosvalmius- ja kanssakäyminen-muuttujien kanssa mutta heikommin dokumentaatio-muuttujien kanssa. Kannattavuus korreloi heikosti positiivisesti muutosvalmiuden kanssa mutta ei muiden ketteryysmuuttujien kanssa. Liikevaihdon muutos ei korreloinut minkään ketteryysmuuttujan kanssa.

TAULUKKO 6: Spearmanin järjestyskorrelaatio yrityksen ominaisuuksia kuvaaville muuttujille ja ketteryysmuuttujille

Spearman	Muutosvalmius	Dokumentaatio	Kanssakäyminen
Ikä	-,221 **	-,247 **	-,238 **
Henkilöstömäärä	-,209 **	-,187 *	-,247 **
Kannattavuus	,187 *	,086	,023
Liikevaihdon muutos	,030	,082	-,098

5 JOHTOPÄÄTÖKSET

Tämän tutkimuksen tarkoituksena oli selvittää millä käsitteillä ohjelmistoyritysten ketteryyttä voidaan mitata. Ketteryydelle myös pyrittiin löytämään selittäviä tekijöitä yrityksen sisäisistä ominaisuuksista ja markkinoista. Tutkimus ei varsinaisesti perustunut aiemmalle tutkimukselle, minkä vuoksi johtopäätöksiä on tehtävä varoen liikaa yleistämistä ja hyväksyen tutkimuksen lähtökohtien puutteet. Tutkimus ei myöskään testaa teoriaa vaan pyrkii vasta todistamaan määritelmän soveltuvuuden ohjelmistokehitykselle.

Tässä tutkielmassa käytetyn kyselylomakkeen väitteet perustuvat Agile Manifestolle, sillä se vaikutti perustellulta lähtöpisteeltä ketteryyden mittaamiseksi. Väitteissä oli kuitenkin omat ongelmansa, minkä vuoksi kyselyn vastauksia on tulkittava varovasti. Osa näistä ongelmista olisi ehkä voitu välttää, mikäli väitteitä olisi testattu enemmän ennen varsinaisen aineiston keräämistä. Tähän ei kuitenkaan ollut mahdollisuutta, minkä vuoksi tämän kyselyn tuloksia tulee tarkastella myös mittarin puhdistamisen näkökulmasta.

Agile Manifesto muodostuu neljästä arvosta, joista kustakin muodostettiin kyselyyn neljä väitettä. Tarkoituksena oli pyrkiä muodostamaan summamuuttujia koskien kutakin väitettä. Pääkomponenttianalyysin tuloksena saatiin kuitenkin vain kolme komponenttia, joista ensimmäinen koostuu useampaa arvoa koskevista väitteistä. Tähän voi olla syynä se, että asiakasyhteistyötä ja muutokseen vastaamista koskevat väitteet koskivat vastaajien kokemuksen mukaan samanlaisia periaatteita toimintatavoissa. Toisaalta kiinnostavaa on myös pohdittava miksi dokumentaatiota ja kanssakäymistä koskevat väitteet eivät latautuneet samalle komponentille.

Muutosvalmiusmuuttujan syntyminen useammasta arvosta on mielenkiintoista ketteryyden määritelmien näkökulmasta. Ensimmäisen komponentin viisi väitettä nimittäin koskevat asiakasyhteistyötä ja muutosvalmiutta, jotka ovat yhtymäkohdat myös Agile Manifeston (2001) ja Conboyn (2009) määritelmän välillä. Myös Lee ja Xia (2010) määrittivät ketteryyden muutosvalmiuden kautta. Tämä siirtää huomion määritelmien validiuteen nykypäivänä sillä Agile Manifesto on selvästi muita määritelmiä aiemmin luotu.

Dokumentaatiota koskeviin väitteisiin annetut vastaukset olivat selvästi suunnitelmaperustaisempia kuin muihin arvoihin liittyvät väitteet ja vastausten keskiarvot keskittyivät selvästi neutraalin vaihtoehdon ympärille. Tämä voi kieliä siitä, ettei vastaaja kokenut väitteen ääripäiden sopivan kuvaamaan liiketoimintaansa. Myös sanan "dokumentaatio" ymmärtäminen sekä dokumentaation erilainen merkitys erilaisten ohjelmistotuotteiden kohdalla voivat olla syytä erilaisiin vastauksiin. Ohjelmistokehityksen näkökulmasta sillä tarkoitetaan kehityksen edistymisen kommunikoinniseksi sekä sen ohjaamiseksi tarkoitettuja dokumentteja mutta se voi merkitä myös asiakkaalle tarkoitettuja käyttöohjeita, ohjelmakoodin merkintöjä tai tulevia projekteja varten arkistointikäyttöön tarkoitettua materiaalia. Toisaalta myös joihinkin ketteriin menetelmiin liittyy vahvasti dokumentaatio (esim. Test driven development, TDD) ja sen tuottaminen ketterästi ja turhaa välttäen. On myös mahdollista, että dokumentaation asema ohjelmistokehityksessä on ollut erilainen vuonna 2001, jolloin Agile Manifesto on julkaistu ja siksi se on ollut osa Manifestoa. Nykyään kuitenkin internet ja pilvipalvelut ovat paljon keskeisempi osa kaikenlaisia projekteja, mikä voi vähentää tarvetta erillisten ohjelmistokehitysprojektin etenemistä kuvaavien dokumenttien tekemiselle, koska viestintätavat ovat muuttuneet.

Yrityksistä voitiin löytää hyvin vähän eroja perustuen niiden markkinoihin. Ei siis voida esimerkiksi sanoa, että kansainväliset markkinat tai liiketoimintamalli aiheuttaisivat painetta toimia ketterämmin. Yrityksen tuotteiden kohderyhmällä on toisaalta yhteys yrityksen ketteryyteen. Tässä lähimmäs ketterää sijoittuivat yritykset, joilla ei ollut tuotetta, mikä voidaan yhdistää ajatuksen paleluyritysten luontaisesta ketteryydestä. Esimerkiksi konsultoinnissa projektin kulkua ei usein kuulukaan päättää tiukasti etukäteen vaan se muotoutuu työn edetessä. Näiden yritysten kohdalla voidaan myös pohtia sitä, tehdäänkö yrityksissä todella ohjelmistokehitystä ja voidaanko Agile Manifestoa soveltaa niihin siis lainkaan.

Ketteryys korreloi negatiivisesti yrityksen iän ja henkilöstömäärän kanssa. Ketteryys on siis suurempaa silloin, kun yrityksen henkilöstömäärä on pienempi ja se on ikävuosiltaan nuorempi. Tämä toteutui kaikkien ketteryyden muuttujien kanssa vaikkakin dokumentaation ja henkilöstömäärän yhteys oli heikompi. Henkilöstömäärän yhteys ketteryyteen voi johtua siitä, ettei pienissä yrityksissä välttämättä ole erillistä johtoporrasta käsittelemässä asiakkaita ja tekemässä strategioita vaan työntekijöiden työnkuvat ovat laajempia ja vaikutusmahdollisuudet toimintatapoihin ovat paremmat. Tämän tuloksen pohjalta näyttää myös siltä, että monissa ketterissä menetelmissä suositeltu pieni itsenäinen kehittäjätiimi on suurta organisaatiota parempi, kun pyritään ketteryyteen. Korrelointi yrityksen iän kanssa voi taas johtua siitä, ettei yritykselle ole vielä muodostunut suunnitelmaperustaisia toimintamalleja vaan toimitaan vielä vastaantulevien mahdollisuuksien mukaan.

Yrityksen taloudelliseen tilanteeseen liittyvät muuttujat kannattavuusprosentti ja liikevaihdon muutos kolmen vuoden aikana eivät korreloineet ketteryyden muuttujien kanssa yhtä lukuun ottamatta. Muutosvalmiusmuuttuja korreloi heikosti positiivisesti kannattavuusprosentin kanssa. Tästä ei pidä suoraan

vetää johtopäätöstä, että yrityksen tulos olisi parempi, mikäli se toimii ketterästi mutta korrelaatio antaa kuitenkin viitteitä siitä, että ketteryydellä voisi olla yhteys yrityksen taloudelliseen menestykseen. Koska kyseessä oli kuitenkin ainoa korrelaatio tutkitusta kuudesta ketteryyden ja taloudellisten tunnuslukujen välillä, ei voida sanoa, että suunnitelmaperustaisuus olisi ohjelmistoliiketoiminnalle huono toimintatapa. Vahvan korrelaation puuttuminen herättääkin kysymyksen siitä, mihin ketteryyden suosio perustuu. Jatkuvasti lisääntyvä tutkimus (Balijepally ym., 2012) ja ketterien menetelmien käyttö (Version one, 2014) voisivat antaa kuvan ilmiöstä, joka mullistaa ohjelmistokehityksen mutta viitteitä sen vaikutuksista ohjelmistoliiketoiminnalle ei saatu tämän tutkimuksen tuloksista. Lee ja Xia (2010) tosin löysivät omassa tutkimuksessaan yhteyden ohjelmistokehitystiimin suoritustehon (nopeuden) ja projektin budjetissa pysymisen väliltä. Ei siis tule vielä kokonaan kieltää ketteryyden taloudellisia vaikutuksia, vaikka ne eivät tässä olleet ilmeisiä.

Yhdysvalloissa ja Euroopassa toteutetussa kaupallisessa tutkimuksessa 88 % vastaajista kertoi heidän yrityksessä käytettävän ketteriä ohjelmistokehitysmenetelmiä (Version One, 2014). Kun tätä lukua verrataan siihen, että tässä tutkimuksessa kahdessa muuttujassa vastauksista 75 % oli ketteriä ja kolmannessa vain 50 % on pohdittava mistä tämä ero johtuu. Luonnollisesti vastaajajoukko, kyselyn taustamotivaatiot ja tekotapa ovat hyvin erilaisia, minkä vuoksi kyselyjen tulokset ole suoraan vertailukelpoisia. Tämä selvä ero kuitenkin herättää kysymyksiä siitä, johtaako ketteräksi kutsutun työmenetelmän käyttö ketterään toimintaan. Tämän saman ristiriidan löysi myös Conboy (2009), jonka tutkimustuloksissa ketteryyttä edistäviä työtapoja otettiin käyttöön myös ketterän menetelmän ulkopuolelta, kun taas kaikki menetelmän työtavat eivät tuke-neet ketteryyttä.

Kyselyn vastauksia tulkitessa on myös otettava huomioon vastaajan näkemys ohjelmistokehityksestä, sillä kysely oli kohdistettu yritysten toimitusjohtajille. Baskerville ym. (2011) sekä van Vliet ja van Waardenburg (2013) löysivät tutkimuksissaan jännitteitä ketterästi toimivien ohjelmistokehitystiimien ja suunnitteluperustaisesti toimivan muun organisaation välillä. Tämä havainto voisi tässä tutkimuksessa ilmentyä niin, että ohjelmistokehityksen ulkopuolella oleva toimitusjohtaja pitää esimerkiksi dokumentteja ja sopimusneuvotteluita tärkeämpänä kuin ohjelmistokehittäjät. Tämä siis voi selittää suurempien yritysten vähäisempää ketteryyttä.

Tässä tutkielmassa esitetty mittari vaatii vielä kehittämistä, ennen kuin sillä voidaan saada luotettavia tuloksia yritysten ketteryydestä. Ilmeisin ongelma on analyysistä pääkomponenttianalyysin ja Cronbachin alfojen vuoksi poisjätetyt väitteet. Kysymysten sisältöä tulisikin tarkentaa ainakin niin, että ne eivät ota liiallisesti kantaa ohjelmistoa kehittävän yksikön kokoon, väitteet edustavat ehdottomasti kahta ääripäätä ja kysymykset edustavat tiukasti vain yhtä arvoa kerrallaan.

Tässä tutkielmassa ketteryyden mittari luotiin tiukasti Agile Manifeston pohjalta. Ketteryydelle on kuitenkin muitakin määritelmiä ja voisikin olla perusteltua muokata mittaria niin, että esimerkiksi Conboyn (2009) esittämä ajal-

linen ulottuvuus sekä lisäarvon tuottaminen asiakkaalle sisältyisivät kysymyksiin selvemmin.

Ketteryyden mittarin kehittäminen on tärkeää, jos halutaan saada vertailtavaa tutkimustietoa ketteryyden vaikutuksesta ohjelmistokehitysprosessiin. Ilman keinoa mitata ketteryyttä, ei voida esimerkiksi tehdä päätelmiä siitä sopiiko ketteryys paremmin esimerkiksi tietynlaisille markkinoille tai onko se yritykselle taloudellisesti kannattavaa.

Tässä tutkimuksessa ei saatu selville, mitkä yritykset tekevät parempaa ja mitkä huonompaa ohjelmistokehitystä. Agile Manifeston alkusanat ovat ”Löydämme parempia tapoja tehdä ohjelmistokehitystä, kun teemme sitä itse ja autamme muita siinä” (Agile Alliance, 2001). Jos määritelmä hyvälle ohjelmistokehitykselle olisi käytössä, voitaisi Agile Manifeston arvojen toteuttamista peilata sitä vasten. Näin voitaisi saada tietoa siitä, onko ketteryydellä todella vaikutusta ohjelmistotuotantoon vai onko se vain yhden aikakauden ihanne, joka väistyy kohta seuraavan muotivillityksen tieltä.

6 YHTEENVETO

Tässä tutkielmassa pyrittiin luomaan yleiskuva ketteryydestä ohjelmistokehityksessä painottaen ketteryyttä toimintatapojen taustalla vaikuttavana arvona. Tämän tutkimuksen tarkoituksena oli selvittää millä käsitteillä ohjelmistokehityksen ketteryyttä voidaan mitata. Ketteryydelle myös pyrittiin löytämään selittäviä tekijöitä yrityksen sisäisistä ominaisuuksista ja markkinoista. Ensin tutkielmassa käsiteltiin ketteryyteen liittyviä käsitteitä sekä ketteryyden tutkimusta. Tutkielman toisessa osassa luotiin mittari ketteryyden mittaamiseksi ja kehitettiin mittaria määrällisessä tutkimuksessa.

Ketteryydelle löydettiin kirjallisuudesta tarkempia määritelmiä sekä yksittäisiä kuvailevia sanoja kuten joustavuus ja vikkeläliikkeisyys. Selvempiä määritelmiä olivat esimerkiksi ohjelmistokehitykselle tarkoitettu Agile Manifesto (julistus ketterästä ohjelmistokehityksestä) (Agile Alliance, 2001) sekä tietojärjestelmien kehittämiseksi tarkoitettu Conboyn (2009) määritelmä. Näillä määritelmillä oli paljon yhteistä mutta myös selviä eroja. Yhteyksiä ketteriin menetelmiin oli myös helppoa löytää menetelmien arvoista ja periaatteista, jotka ovat menetelmän työtapojen takana.

Ketteryys ja ketterät menetelmät yhdistettiin ohjelmistokehitykseen 1990-luvulla. 2000-luvun alussa suosituimpia menetelmiä oli XP, kun taas tälle vuosikymmenelle tullessa Scrum vaikuttaa olevan suosituin. Ketterät menetelmät pitää nähdä välineinä, joilla pyritään ketteryyteen, eikä pidä olettaa, että jonkin menetelmän soveltaminen automaattisesti johtaisi kykyyn toimia ketterästi.

Ketteryydellä on ohjelmistokehityksessä jo kahden vuosikymmenen historia mutta tutkimuksen näkökulmasta aihe on vielä nuori. Ketteryyttä koskevia tieteellisiä artikkeleita julkaistaan vuosi vuodelta enemmän mutta painotus on edelleen teoriaa luovassa tutkimuksessa eivätkä tieteelliset määritelmät tai teorit ole juuri levinneet ketteryyden tutkijoiden keskuudessa. Ketteryyden tutkimuksessa painottuvat erityisesti case-tutkimukset ja niistä monet tutkivat juuri ketterien menetelmien käyttöä. Ketteryyden tutkimiseen on myös sovellettu erilaisia teorioita mutta sopivaa mallia ei ole esitelty ketteryyden vaikutuksen tutkimiseen ohjelmistokehityksessä. Pisimmällä ketteryyden mittaamisessa vaikuttaa olevan Conboyn (2009) sekä Leen ja Xian (2010) tutkimukset.

Tässä tutkimuksessa ketteryyden määritelmäksi valittiin Agile Manifesto. Tutkimusongelmana oli, millä käsitteillä ketteryyttä ohjelmistokehityksessä voidaan mitata ja tämän selvittämiseksi tuli muodostaa mittari määritelmän pohjalta. Jotta voitaisi löytää eroja yritysten ketteryydessä ja verrata sitä yrityksen ominaisuuksiin, käytettiin menetelmänä määrällistä kyselytutkimusta. Agile Manifeston pohjalta luotiin 16 väitettä vastausvaihtoehtoiseen. Kysely lähetettiin 5050 suomalaiselle ohjelmistoyritykselle ja tähän tutkimukseen hyväksyttiin 157 vastausta.

Pääkomponenttianalyysin avulla vastauksista löydettiin kolme komponenttia, joista muodostettiin summamuuttujat: muutosvalmius, dokumentaatio, kanssakäyminen. Yrityksen ulkoisia ominaisuuksia kuvaamaan käytettiin kolmea yrityksen markkinoihin liittyvää muuttujaa: liiketoimintamalli, päätuote ja kansainvälisyys. Sisäisiä ominaisuuksia kuvasivat ikä, henkilöstömäärä, kannattavuus ja liikevaihdon kasvu.

Muutosvalmius- ja kanssakäyminen-muuttujien jakaumat asettuivat pääosin neutraalin keskikohdan ketterälle puolelle kun taas dokumentaatiomuuttuja jakautui tasaisemmin neutraalin ympärille. Yhteyksiä ketteryyden ja yrityksen ominaisuuksien välillä löytyi jokseenkin vähän. Yrityksen markkinoiden ominaisuuksista ainoastaan päätuote nosti esiin eron yritysten välillä, joista ne joilla ei ollut tuotetta olivat ketterämpiä kuin yrityksille tuotteita tuottavat. Yrityksen sisäisistä ominaisuuksista ikä ja henkilöstömäärä korreloivat negatiivisesti kaikkien ketteryydsmuuttujien kanssa, eli nuoremmat ja henkilöstöltään pienemmät yritykset vaikuttivat ketterämmiltä. Yrityksen taloudellisista ominaisuuksista kannattavuus korreloi hiukan muutosvalmiusmuuttujan kanssa mutta muita korrelaatioita ei löytynyt.

Ketteryyttä koskevat muuttujat eivät pääkomponenttianalyysin jälkeen latautuneetkaan neljälle komponentille kuten Agile Manifeston perusteella olisi voinut kuvitella. Asiakasyhteistyöhön ja muutoksiin reagoimiseen liittyvät väitteet latautuivat samalle komponentille, mistä voidaan huomata yhteys Conboyn (2009) ketteryyden määritelmään. Dokumentaatioon liittyvän muuttujan vastaukset olivat taas paljon suunnitelmaperustaisempia kuin kahden muun ketteryydsmuuttujan, mikä herättää kysymyksiä dokumentaation asemasta ohjelmistokehityksessä nykyään sekä sen liittymistä ketteryyteen.

Tämän tutkimuksen tutkimuskysymyksenä oli ”millä käsitteillä ohjelmistokehityksen ketteryyttä voidaan mitata?” Erilaisia käsitteitä löydettiin kirjallisuudesta paljon, ja analyysin perusteella päädyttiin kokeilemaan Agile Manifestoa mittarin pohjana. Vaikka luotu kysymyspatteri vaatiikin hiomista, saatiin tästä tutkimuksesta mielenkiintoisia tuloksia. Myös lisäkysymykseen ”onko yrityksen ketteryydellä yhteyttä yrityksen sisäisiin ominaisuuksiin tai niihin markkinoihin, joilla yritys toimii?” saatiin vastauksia. Tulosten perusteella näyttää siltä, että Agile Manifesto on hyvä lähtökohta ketteryyden mittaamiselle, mutta myös muiden määritelmien näkökulmia on hyvä sisällyttää mittariin.

LÄHTEET

- Abrahamsson, P., Haikara, J., Pikkarainen, M., Salo, O. & Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13(3), 303-337.
- Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. (2002). Agile software development methods: Review and analysis. *VTT Publications*, 478.
- Agile Alliance. (2001). Manifesto for agile software development. Haettu 7.12.2014 osoitteesta <http://agilemanifesto.org/>
- Balijepally, V., Dingsøyr, T., Moe, N. & Nerur, S. (2012). A decade of agile methodologies: Towards explaining agile software development. *The Journal of Systems and Software*, 85, 1213-1221.
- Balijepally, V. Mahapatra, R., Nerur, S. & Price, K. (2009). Are Two Heads Better than One for Software Development? *The Productivity Paradox for Pair Programming. MIS Quarterly* 33(1), 91-118.
- Balijepally, V. & Nerur, S. (2007). Theoretical reflections on Agile Development Methodologies. *Communications of the ACM* 50(3), 79-83.
- Baskerville, R., Madsen, S. & Pries-Heje, J. (2011). Post-agility: What follows a decade of agility? *Information and Software Technology* 53, 543-555.
- Bureau van Dijk. (2015). Haettu 3.9.2015 osoitteesta <http://www.bvdinfo.com/en-gb/our-products/company-information/international-products/orbis>
- Ceschi, M., De Panfilis, S., Sillitti, A. & Succi, G. (2005). Project management in plan-based and agile companies. *IEEE Software* 22(3), 21-27.
- Churchill, G. (1979). *Market research : methodological foundations*. Hinsdale (IL): Dryden.
- Cockburn, A. (2007). *Agile Software development: The cooperative game*. Boston, USA : Pearson Education Inc.
- Cockburn, A. & Williams, L. (2003) Agile Software development : it's about feedback and change. *IEEE Computer*, 3(5), 39-43.
- Coleman, J. & Multon, K. (2010). Coefficient Alpha. Teoksessa N. Salkind (suunn. & toim.), *Encyclopedia of Research Design* (s. 160 - 164). CA, USA : SAGE Publications.
- Conboy, K. (2009). Agility from first Principles : Raconstructing the Concept of Agility in Information Systems Development. *Information Systems Research*, 20(3), 329-354.
- Conboy, K., Coyle, S. Wang, X. & Pikkarainen, M. (2011). People over Process : Key Challenges in Agile Development. *IEEE Software*, 28(4), 48-57.
- Conboy, K., Lang, M. & McHugh, O. (2012) Agile Practices : The Impact of Trust in Software Project Teams. *IEEE Software* 29(3), 71-76.
- Cusumano, M. & Poppendieck, M. (2012). Lean Software Development : A Tutorial. *IEEE Software*, 29(5), 26-32.

- Dybå, T. & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50, s833-859.
- Erickson, J., Lyytinen, K. & Siau, K. (2005). Agile Modeling, Agile Software Development, and Extreme Programming : The state of Research. *Journal of Database Management*, 16(4), 88-100.
- France, R., Rumpe, B. & Turk, D. (2005). Assumptions Underlying Agile Software-Development Processes. *Journal of Database Management*, 16(4), 62-87.
- Gummerus. (2015). *MOT Gummerus Uusi suomen kielen sanakirja*. Gummerus Kustannus oy.
- Johansen, T., Kautz, K. & Uldahl, A. (2014). The Perceived Impact of the Agile Development and Project Management Method Scrum on Information System and Software Development Productivity. *Austrasian Journal of Information Systems*, 18(3).
- Järvinen, A. & Järvinen, P. (2011). Tutkimustyön metodeista. Tampere: Opinpajan kirja.
- Lee, G. & Xia, W. (2010). Toward agile : an integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly*, 34(1), 87-114.
- Lu, Y & Ramamurthy, K. (2011). Understanding the link between information technology capability and organizational agility : an empirical examination. *MIS Quarterly*, 35(4), 931-954.
- Luoma, E. (2014) Software industry survey. *Blogikirjoitus*. <http://www.softwareindustrysurvey.fi/node/76.html>
- Lyytinen, K & Rose, G. (2006). Information system development agility as organizational learning. *European Journal of Information Systems*, 15, 183-199.
- Mahapatra, R., Mangalaraj, G. & Nerur, S. (2005) Challenges of migrating to Agile Methodologies. *Communications of the ACM*, 48(5), 73-78.
- Merisalo-Rantanen, H., Rossi, M. & Tuunanen, T. (2005). Is Extreme Programming Just Old Wine in New Bottles: A Comparison of Two Cases. *Journal of Database Management*, 16(4), 41-61.
- Petter, S. & Yu, X. (2014). Understanding agile software development practices using shared mental models theory. *Information and Software Technology* 56(2014), 911-921.
- VersionOne. (2014). 8th annual State of Agile Survey. Haettu 23.11.2014 osoitteesta <http://stateofagile.versionone.com/>
- van Vliet, H. & van Waandenburg, G (2013). When agile meets the enterprise. *Information and Software Technology* 55(2013), 2154-2171.

LIITE 1: KETTERYYDEN MÄÄRITTELY ARTIKKELEISSA

Artikkeli	Agile Manifesto (2001)	Conboyn määritelmä (2009)	Iteratiivisuus, inkrementaalisuus, useat julkaisut	Joustavuus (flexibility)	Nopeus (quick, fast, nimble)	Turhan poisto (leanness, parsimony)	Reagointi muutokseen (respond to change, adaptive)	Kasvotusten /jatkuva kommunikaatio	Asiakasyhteistyö
Abrahamsson, Salo, Ronkainen & Warsta, 2002	x		x		x	x	x	x	x
Cockburn & Williams, 2003	x		x				x		
Ceschi, De Panfilis, Sillitti & Succi, 2005						x	x		x
Merisalo-Rantanen, Rossi & Tuunanen, 2005	x		x	x				x	x
Erickson, Lyytinen & Siau, 2005	x			x	x	x	x		
France, Rumpe & Turk 2005	x		x		x		x	x	x
Lyytinen & Rose 2006			x		x		x		
Mahapatra, Mangalaraj & Nerur, 2005	x		x	x	x	x	x	x	x
Balijepally & Nerur, 2007			x	x	x		x	x	x
Dybå & Dingsøy 2008	x		x	x	x	x	x		
Abrahamsson, Haikara, Pikkarainen, Salo, & Still, 2008	x		x				x	x	
Balijepally, Mahapatra, Nerur & Price, 2009	x		x				x	x	x
Conboy, 2009		X		x	x	x	x		
Lee & Xia, 2010	x		x		x		x	x	x

Baskerville, Madsen & Pries-Heje, 2011	x	x			x		x		
Cusumano & Poppendieck, 2012	x		x	x		x			x
Balijepally, Dingsøyr, Moe & Nerur, 2012	x	x	x	x	x	x	x		x
Conboy, Lang & McHugh, 2012	x		x					x	
van Vliet & van Waandenburg, 2013		x			x		x		x
Petter & Yu, 2014		x	x				x	x	x
Yhteensä	14/ 20	4/8	15	8	12	8	17	10	12

LIITE 2: KYSELYLOMAKE

KEHITYSMENETELMÄT

Kehitystiimimme työtavat on järjestetty niin, että...

1 d	...tiimit ovat jatkuvasti valmiita muutoksiin vaatimuksissa.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...vaatimukset voidaan toteuttaa alussa sovitusti muospaineista huolimatta.
2 b	...dokumentaation tekeminen on keskeinen osa työtehtäviä.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...dokumentaation tekeminen on mahdollisimman vähäinen osa työtehtäviä.
3 a	...sitä ohjaavat tiimin jäsenten yhdessä sopimat työtavat ja tavoitteet.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...sitä ohjaavat yrityksessä käytettävät toimintatavat ja johdon asettamat tavoitteet.

Kehitystiimimme kommunikoivat pääasiassa...

4 c	...asiakkaan kanssa helpoimalla mahdollisella tavalla aina tarvittaessa.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...asiakkaan kanssa sovitun toimintamallin ja aikataulun mukaisesti.
5 a	...tiimin jäsenten kesken yrityksessä määriteltyjä keinoja sovitusti käyttäen.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...tiimin jäsenten kesken vapaasti haluamallaan tavalla.

Ohjelmistokehitysprojektimme todennäköisemmin...

6 b	...epäonnistuu, jos toimimme liikaa suunnitelmien ja dokumentaation ehdoilla muuttuviin tilanteisiin reagoimisen sijaan.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...epäonnistuu, jos teemme liikaa muospäätöksiä suunnitelmassa pitäytymisen ja dokumentoinnin sijaan.
7 b	...onnistuu, koska toimivan ohjelmiston lisäksi syntyy kattava dokumentaatio.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...onnistuu, koska toimivan ohjelmiston lisäksi syntyy vähäinen dokumentaatio.

Ohjelmistokehitysprojektiemme tuottavuuden kannalta on tärkeää, että...

8 c	...projekti voi vastata tilaukseen asiakkaan alkuperäisen toiveen mukaisesti.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...projekti voi vastata asiakkaan muuttuviin tarpeisiin kehitystyön aikana.
9 b	...projekti keskittyy vain ohjelmiston tuottamiseen.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...projekti tuottaa sekä toimivan ohjelmiston että laadukkaan dokumentaation.

Kehitystiimien työntekijöiden...

10 a	...työtehtävät valikoituvat heille oman osaamisen mukaan joustavasti.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...työtehtävät valikoituvat heille työntekijäprofiilin mukaan esimiehen päätöksellä.
11 d	...osaamisen kehittämisessä keskittään heidän kykyynsä suunnitella kattavasti ja ennakoivasti.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...osaamisen kehittämisessä keskittään heidän kykyynsä reagoida muuttuviin tarpeisiin.
12 a	...työ on tuottavampaa, kun he muodostavat omat työtapansa yhdessä tiimin kanssa.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...työ on tuottavampaa, kun he käyttävät yrityksen yhteisiä toimintatapoja.

Projekteillemme on tunnusomaista, että...

13 d	...ne ovat tuottavampia, kun muutamme suunnitelmiamme joustavasti muuttuvien vaatimusten mukaisesti kehitystyön aikana.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...ne ovat tuottavampia, kun laadimme projektin alussa toimivat suunnitelmat ja pidämme niistä kiinni.
14 c	...projektin tavoite on määritelty etukäteen asiakkaan kanssa projektisuunnitelmaan ja toimitamme suunnitellun ohjelmiston.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...projektin tavoite voi muuttua kesken projektin asiakkaalta saadun palautteen perusteella.
15d	...ne valitaan toteutettaviksi pääasiassa pidemmän aikavälin strategisten suunnitelmien mukaisesti.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...ne valitaan toteutettaviksi pääasiassa juuri sen hetkisten mahdollisuuksien ja tarpeiden mukaisesti.
16 c	...projektin lopputulokset sovitaan pääpiirteittäin jättäen tilaa asiakkaan kanssa sovittaville muutoksille.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	...projektin lopputulokset sovitaan tarkasti määritellen projektin kulku ennen sen käynnistämistä.