

Simone Lehikoinen

TYÖNTEKIJÄN SITOUTUMINEN KETTERILLÄ MENETELMILLÄ TEHTYYN OHJELMISTOKEHITYKSEEN



JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2020

TIIVISTELMÄ

Lehikoinen, Simone

Työntekijän sitoutuminen ketterillä menetelmillä tehtyyn ohjelmistokehitykseen

Jyväskylä: Jyväskylän yliopisto, 2020

Tietojärjestelmätiede, kandidaatin tutkielma

Ohjaaja: Halttunen, Veikko

Tämä tutkielma on kirjallisuuskatsaus työntekijän sitoutumisesta ketterillä menetelmillä tehtyyn ohjelmistokehitykseen. Työntekijän sitoutumisen positiivinen merkitys organisaation liiketoimintaan ja tulokseen on huomattu jo kymmeniä vuosia sitten, mutta vasta viime vuosina työntekijän sitoutumisen vaikutusta tiimityöskentelyyn on alettu tutkimaan. Kirjallisuuskatsaus kuvaa työntekijän sitoutumisen vaikutukset ja haasteet ketterissä menetelmissä. Havainnot osoittavat, että organisaatioiden tulisi panostaa yhä enemmän ketterien ohjelmistokehitysten tiimien väliseen ja sisäiseen yhteistyöhön sekä huomioida edellä mainitut asiat myös tiimien johtamisessa. Tutkielma aloitetaan ohjelmistotuotannon käsitteen avaamisella ja tietojärjestelmän kehittämismenetelmien esittelyllä. Ohjelmistokehityksessä verrataan perinteisiä lineaarisia kehitysmenetelmiä ja ketteriä menetelmiä, joista molemmista metodeista tuodaan yleisimmät esille. Kirjallisuuskatsauksessa esitellään sitoutumisen rakenne, mallit ja tekijät, jotka vaikuttavat yleisesti ihmisten sitoutumiseen työhönsä. Tutkielmasa syvennyttään sitoutumisen merkitykseen ketterien ohjelmistokehitysmenetelmien onnistumisessa ja menestymisessä.

Asiasanat: ketterät menetelmät, työntekijän sitoutuminen, sitoutumisen muodot, ohjelmistokehitys

ABSTRACT

Lehikoinen, Simone

Employee's Commitment to Software Process Improvement via Agile Methods

Jyväskylä: University of Jyväskylä, 2020

Information Systems Science, Bachelor's Thesis

Supervisor: Halttunen, Veikko.

This thesis is a literature review of employee commitment to agile software development. The importance of employee engagement in organization's business and performance has been recognized decades ago, but only in recent years has the research on the impact of employee engagement onto teamwork begun. The literature review describes the effects and challenges of employee engagement in agile methods. The findings show that organizations should invest more and more in collaboration between agile software development teams and team members. This thesis begins with the introduction of the concept of agile software production and the presentation of the methods for developing an information system. The literature review presents the structure, models, and factors that influence people's overall commitment to their work. This research explores the importance of the employees' commitment, commitment's impact on organizations' success and onto the success of agile software development methods.

Keywords: software process improvement, agile methods, employee's commitment, models of commitment.

KUVIOT

KUVIO 1 Vesiputousmalli.....	11
KUVIO 2 V-malli.....	12

TAULUKOT

TAULUKKO 1 Motivaattoritekijät sitoutumisen malleissa	25
TAULUKKO 2 Sitoutumisen prosessi ketterässä ohjelmistokehityksessä.....	27

SISÄLLYS

TIIVISTELMÄ	2
ABSTRACT	3
KUVIOT	4
TAULUKOT	4
SISÄLLYS.....	5
1 JOHDANTO.....	6
2 OHJELMISTOTUOTANNON MENETELMÄT	9
2.1 Ohjelmistotuotannon määritelmä	9
2.2 Ohjelmistotuotannon elinkaarimallit.....	10
2.2.1 Vesiputousmalli.....	10
2.2.2 V-malli.....	11
2.2.3 Spiraalimalli	13
2.3 Ohjelmistokehitys ketterillä menetelmillä	14
2.3.1 Scrum	15
2.3.2 Extreme Programming	17
2.3.3 Dynaaminen ohjelmistokehitysmenetelmä	18
2.3.4 Yhteenveto.....	19
3 TYÖNTEKIJÄN SITOUTUMINEN	21
3.1 Sitoutumisen määritelmä ja sitoutumisen teemat.....	21
3.2 Motivaatio sitoutumisen muodoissa.....	23
3.3 Sitoutuminen ketterien menetelmien näkökulmasta	25
3.3.1 Sitoutuminen Scrumin näkökulmasta.....	27
3.3.2 Sitoutuminen Extreme Programming -kehitysmenetelmän näkökulmasta.....	28
3.3.3 Sitoutuminen dynaamisen ohjelmistokehitysmenetelmän näkökulmasta.....	29
3.3.4 Sitoutuminen ketterässä kehittämisessä: Case Google.....	30
3.3.5 Yhteenveto.....	31
4 POHDINTA JA YHTEENVETO	32
LÄHTEET	35

1 JOHDANTO

Sitoutuminen itsessään on satojen tutkimuskysymysten kohde. Sitoutuminen käsitteenä on herättänyt huomiota organisaation menestymisessä. Kuitenkaan tutkimuksia ketterien menetelmien prosessien parantamiseen ja tulosten kasvattamiseen sitoutumisen kautta ei niinkään ole tehty, vaikka työntekijöiden merkitys on huomioitu arvokkaaksi (Cockburn & Highsmith, 2001). Näin ollen tässä kirjallisuuskatsauksessa keskitytään työntekijöiden sitoutumisen merkitykseen valituissa ketterien ohjelmistokehityksen malleissa.

Ketterät menetelmät tarkoittavat ohjelmistokehityksen mallia, joka perustuu nopeisiin ja lyhyihin iteraatiosykleihin. Ketteriin kehitysmenetelmiin liittyy joustavuus, nopeus, kätevyys, elinvoimaisuus, vireys ja reaktiokykyisyys. Ketteryys tarkoittaa kaiken ylimääräisen ja raskaan poistamista prosesseista, jotka liittyvät perinteisiin ohjelmistokehitysmenetelmiin. Ketteryys edistää nopeaa reagointia muuttuviin ympäristöihin, joita ovat käyttäjien vaatimukset ja nopeutettujen projektien määräaikojen muutokset. Perinteiset vakiintuneet menetelmät ovat liian hitaita, konservatiivisia ja kankeita, mikä toimii perusteluna ketterien menetelmien kehitykselle. (Erickson, Lyytinen & Siau, 2005.)

Toimiva ja menestyvä ketterä ohjelmistokehitys tarvitsee tiimin, joka kykenee toimimaan tehokkaasti ja vastaamaan nopeasti muutoksiin. Tiimin täytyy vähentää informaation siirrosta aiheutuvaa tiedon menetystä sekä päätöksenteon ja toteutuksen välille kuluvaa aikaa. Jotta tiimin työskentely sujuisi odotetulla tavalla, on tärkeää parantaa tiimin positiivista henkeä, kohentaa tiimin ystävällisyyttä, yhteisöllisyyttä ja moraalialia niin, että ihmiset ovat halukkaampia muun muassa välittämään arvokasta tietoa nopeasti toisilleen. (Cockburn & Highsmith, 2001.)

Cockburn ja Highsmith (2001) toteavat, että yksittäisten työntekijöiden taitojen kehittäminen on toki tärkeää, mutta ensisijaisesti ketterissä kehitysmenetelmissä työskentelevien johtajien tulisi painottaa inhimillisiä tekijöitä: ystävällisyyttä, viestintää, lahjakkuutta ja taitoa. Nämä ominaisuudet tulee priorisoida korkealle ketterissä kehitysmenetelmien malleissa. Kyseiset ominaisuudet sekä inhimillisyydet antavat ketterille hankkeille mahdollisuuden menestyä.

Organisaatio on yksi tärkeimpiä yhteisöjä, joihin ihminen voi elämänsä aikana kuulua. Työntekijä viettää työyhteisössä enemmän aikaa kuin muissa yhteisöissä elämänsä aikana. Luonnollisesti työntekijän toimeentulo riippuu organisaation tuloksesta, suorituskyvystä, oman työn tehokkuudesta ja roolista. (Bergami & Bagozzi, 2000.)

Vähitellen vuosikymmenien jälkeen on ymmärretty työntekijöiden merkitys yrityksen suorituskykyyn ja tutkimuksia työntekijän sitoutumisesta ja sitoutumiseen liittyvistä tekijöistä on tehty suuret määrät. 1970-luku oli käännekohta, kun huomio siirrettiin organisatoriseen sitoutumiseen ja sitoutumisen vaikutukseen työntekijän suorituksessa. Tutkimuksissa havaittiin, että työntekijöiden korkean tason sitoutumiset olivat suoraan yhteydessä organisaation positiivisiin tuloksiin kuten poissaolojen vähenemisiin, myöhästymisten tai henkilöstön vaihtuvuuksien vähenemisiin sekä yksilöllisten suorituskykyjen korkeampaan tasoon. (Beck & Wilson, 2000.)

Kuitenkin sitoutuminen itsessään on ollut vaikeasti määriteltävissä, sillä sosiologit käyttivät sitoutumisen termiä analyysissään sekä yksilöllisestä että organisatorisesta käyttäytymisestä. Sitoutumisen käsitettä käytettiin myös laajoissa analyyseissä selittämään erilaisia ilmiöitä kuten vallan, uskonnon, ammatillisen, byrokraattisen ja poliittisen käyttäytymisen malleja. (Becker, 1960.) Tässä kirjallisuuskatsauksessa avataan sitoutumisen käsite ja määritellään sitoutumisen muodot, jotka yhdistetään nykyajan ohjelmistokehityksen malleihin eli ketteriin menetelmiin.

Tutkielman tavoitteena on selventää inhimillisten tekijöiden ja sitoutumisen merkitys paljon käytettyjen ketterien ohjelmistokehityksen mallien menestyksessä ja osoittaa Abrahamssonin (2001) julkaisun tulos oikeaksi: sitoutuminen on tärkein tekijä menestyneessä ohjelmistokehityksessä. Tavoitteena on osoittaa työntekijöiden sitoutumisen positiivinen vaikutus ja suuri merkitys organisaation menestymiseen ketterillä menetelmillä tehdyssä ohjelmistokehityksessä. Tutkimusongelma voidaan siis tiivistää seuraavaan tutkimuskysymykseen: mitkä tekijät vaikuttavat työntekijän sitoutumiseen ketterillä menetelmillä tehdyssä ohjelmistokehityksessä?

Tutkimusmenetelmänä on käytetty systemaattisen kirjallisuuskatsauksen piirteitä työn loogisuuden ja laadun varmistamiseksi. Kirjallisuuskatsaus etenee vaiheittain, jotta työn korkea laatu voidaan varmistaa. Kirjallisuuskatsaus on tehty Breretonin, Kitchenhamin, Budgenin, Turnerin ja Khalilin (2007) julkaisun mukaisesti vaiheittain:

1. Suunnittelu: asetetaan tutkimuskysymykset ja kehitetään protokolla, joka validoidaan.
2. Kirjallisuuskatsaus: etsitään ja valitaan aineisto syvyysasteittain, arvioidaan luotettavuus, eritellään tarvittava tieto ja laaditaan synteesi.
3. Dokumentin tuottaminen: laaditaan raportti ja tehdään validointi.

Aineiston etsimisessä ja valitsemisessa on käytetty erilaisia hakukoneita, kuten Googlea, Yahooa ja Bingiä. Lähdemateriaalien luotettavuus on arvioitu selvittämällä tieteellisten julkaisujen arvostus, julkaisujen tekijöiden tunnettavuus ja

julkaisujen esiintyvyys muissa tieteellisissä tutkielmissa. Lähdemateriaalin validointi on tehty tämän kirjallisuuskatsauksen tavoitteiden pohjalta.

Tutkimuksen tulokset osoittavat, että menestyneimmät projektit tuotetaan sitoutuneiden tiimijäsenten työpanoksella. Tutkimus myös osoittaa yhden menestykseen johdattavan sitoutumisen mallin, jonka sitoutumisen muoto on syvin sekä tuottavin.

Tutkielmassa pohjustetaan aihetta ohjelmistotuotannon käsitteen avaamisella luvussa kaksi sekä tutustumalla ohjelmistotuotannon kehitykseen valituilla perinteisillä malleilla. Tämän jälkeen esitetään valitut ohjelmistokehityksen ketterien menetelmien mallit. Luvussa kolme siirretään huomio sitoutumisen määritelmään ja sitoutumisen malleihin. Luvun kolme aikana syvennyttään sitoutumisen merkitykseen valituissa ketterien ohjelmistokehityksen malleissa. Luvussa neljä esitetään kirjallisuuskatsauksen havainnot ja johtopäätökset.

2 OHJELMISTOTUOTANNON MENETELMÄT

Ohjelmistotuotanto sisältää kaikki vaiheet ja menetelmät tietokoneohjelmiston tai -ohjelman suunnittelusta ohjelmiston ylläpitoon. Projektinhallinta, käytetyt johtamisen menetelmät sekä valmistukseen liittyvät prosessit ovat osa ohjelmistotuotantoa. (Ghezzi, Jazayeri & Mandrioli, 2002)

Tietokoneohjelmistojen analyysi, suunnittelu, ohjelmointi, testaus, käyttöönotto sekä ylläpito etenevät loogisesti ja kaavamaisesti, vaikka ohjelmistotuotannossa käytettäisiin erilaisia lähetystapoja ja menetelmiä. Jokaiselle menetelmälle yhteistä on alustava tutkimus, analyysi, muotoilu, käyttöönotto sekä testaus ja ylläpito. (Ghezzi ym., 2002.)

Tässä luvussa esitellään ohjelmistotuotannon rakenne ja ohjelmistokehityksen elinkaaren vaiheet sekä yleisimmät mallit niin perinteisistä kuin ketteristä kehitysmalleista. Luvun päätteeksi lukijalla on selkeä kuva ohjelmistokehityksestä niin perinteisillä kuin ketterillä malleilla.

2.1 Ohjelmistotuotannon määritelmä

Pressman ja Maxim (2014) tiivistävät ohjelmistotuotannon määritelmän olevan tekniikan moitteetonta käyttöä ja rakentamista, jolla saadaan kustannustehokkaasti luotua ohjelmisto, joka on luotettava ja toimii vaivattomasti oikeilla laitteilla.

Pressmanin ja Maximin (2014) mukaan ohjelmistotuotanto on kerroksellista teknologiaa, jossa jokaisen organisaation jäsenen tulee olla sitoutunut laatu-määrittäisiin. Pohja, johon teoria nojaa, tulee laadullisesta näkökulmasta ja jatkuvasta kehittämisen kulttuurista. Kyseinen kulttuurinäkökulma johtaa automaattisesti yhä kustannustehokkaampiin lähestymistapoihin ohjelmistotuotannon kehittämisessä.

Laatusitoumuksen jälkeen toinen kerros Pressmanin ja Maximin (2014) mukaan on prosessin viitekehys. Ohjelmistotuotannon prosessin viitekehys yhdistää teknologian ja suunnittelun sekä tukee aikataulussa pysymistä. Prosessin

viitekehysmalli ohjaa myös organisaation johtoa kontrolloimaan, tukemaan ja tarkastamaan välitavoitteiden ja laatustandardien saavuttamista.

Pressmanin ja Maximin (2014) kerrosmallissa toiseksi viimeisenä kuvataan organisaation metodit, jotka tarjoavat tekniset näkökulmat ja ratkaisut ohjelmistotuotannon valmistamiselle. Kuten Ghezzi ym. (2002) mallissaan esittää perusmenetelmät ohjelmistotuotannolle, Pressmanin ja Maximin (2014) metodikerrokseen sisältyy myös kommunikaatio, vaatimusten analyysi, mallinnus, ohjelman koodaus, testaus sekä tuki.

Pressmanin ja Maximin (2014) kerrosmallissa viimeinen kerros sisältää organisaation työkalut, jotka tarjoavat osittaisen tai kokonaan automatisoidun tuen prosesseille ja metodeille ohjelmistotuotannossa.

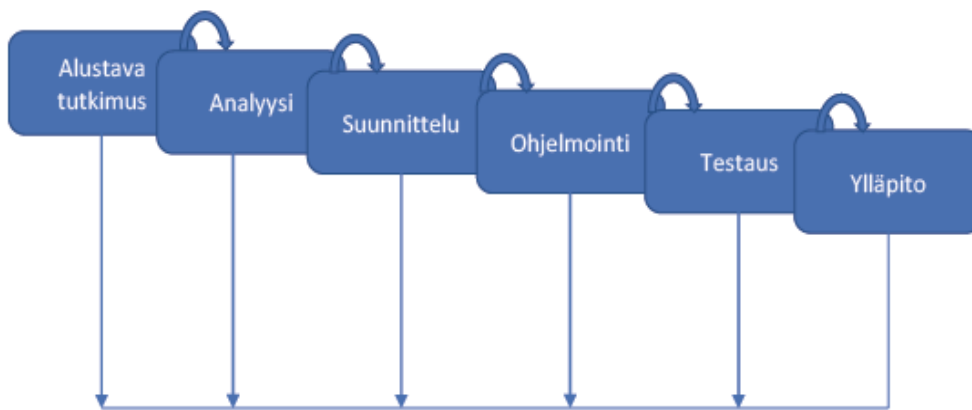
2.2 Ohjelmistotuotannon elinkaarimallit

Ohjelmistotuotannon elinkaarimalleihin kuuluvat määritellyt eli perinteiset mallit sekä ketterät mallit. Perinteisillä malleilla tarkoitetaan muun muassa vesiputousmallia, josta ohjelmistokehitys on saanut alkunsa. Muita perinteisiä malleja ovat esimerkiksi V-malli ja spiraalimalli. Spiraalimalli on toiminut avainasemassa ketterien menetelmien kehitykselle. (Paetsch, Eberlein & Maurer, 2003.)

Paetschin ym. (2003) mukaan ketterät menetelmät johdettiin perinteisistä malleista, kun teknologia kehittyi. Perinteiset mallit olivat liian kankeita organisaation projekteihin. Ketterillä menetelmillä pyrittiin vastaamaan yhä nopeammin kehittyvään markkinaan sekä tyydyttämään paremmin asiakkaan muuttuvat halut ja tarpeet. Ketteriä menetelmiä on lukuisia, mutta tässä kirjallisuuskatsauksessa esiteltävät menetelmät ovat Scrum, dynaaminen ohjelmistokehitysmenetelmä (Dynamic Systems Development Method eli DSDM) sekä Extreme Programming eli XP. (Paetsch ym., 2003.) Ketteriin menetelmiin syvennyttään kolmannessa pääluvussa.

2.2.1 Vesiputousmalli

Vesiputousmalli oli aivan ensimmäisiä selkeästi määriteltyjä järjestelmäkehitysmetodeja, joiden juuret ulottuvat aina 1960-luvulle saakka. Vesiputousmallin mukaan määrittelemättömien prosessien käyttö on lineaarista, mutta näkemyksen luonne on muodostunut suurimmaksi ongelmaksi. Kyseinen malli ei määrittele kuinka organisaation tulisi vastata odottamattomiin tapahtumiin. (Boehm, 1988.) Vesiputousmalli havainnollistetaan kuviossa 1.



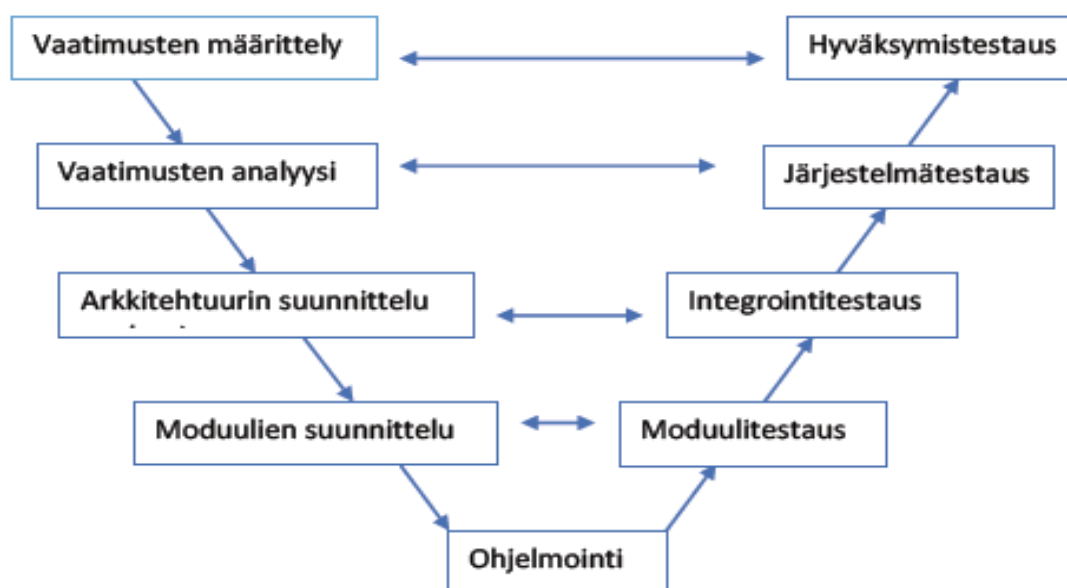
KUVIO 1 Vesiputousmalli.

Ghezzin ym. (2002) mukaan vesiputousmallin vaiheita ovat alustava tutkimus, analyysi, suunnittelu, ohjelmointi, testaus ja ylläpito. Alustava tutkimus koostuu hankkeeseen liittyvistä kehitysehdotuksista ja hankkeen rajoista (mukaan lukien hylätyksi tulleet vaihtoehdot). Tavoitteet ja vaatimukset täytyvät olla tarkkaan määritellyt. Vaihtoehtoja on tarkasteltava teknisestä sekä taloudellisesta näkökulmasta. (Ghezzi ym., 2002.)

Analyysi sisältää teknisten yksityiskohtien tarkastelun. Vesiputousmallin onnistumiselle on elinarvoisen tärkeää, että asiakas osallistuu ohjelmistokehityksen analyysiosioon aktiivisesti. Ainoastaan tällöin asiakkaan tavoitteet ja vaatimukset voidaan vielä toteuttaa. Vesiputousmallin suunnitteluvaiheessa sovelluksen, laitteistojen ja osajärjestelmien yksityiskohdat suunnitellaan tarkkaan. (Ghezzi ym., 2002.) Ghezzin ym. (2002) mukaan toteutus tapahtuu joko mallin mukaisesti ylhäältä alaspäin tai alhaalta ylöspäin. Normaalisti viimeiset vaiheet mallissa ovat järjestelmän testaus sekä ylläpito. Toisessa tapauksessa näistä vaiheista aloitetaan.

2.2.2 V-malli

Boehmin (1988) mukaan V-mallin kehitysmenetelmän ydin on ohjelmiston päävaatimuksissa. Tämä tarkoittaa sitä, että vaatimusten määrittelyt, jotka kehittyvät suunnitteluvaiheessa, ovat hyväksytyjä asiakkaan ja kehittäjän puolesta. V-malli lisää vesiputousmalliin testauksen tasot, mikä helpottaa testauksen suunnittelua jo kehitysvaiheissa. Jokaisen testauksen tasolla on sitä vastaava suunnittelun taso. V-malli on havainnollistettu kuviossa kaksi.



KUVIO 2 V-malli.

Boehmin (1988) mukaan vaatimusmäärittelyt konkretisoituvat suunnitteluvaiheessa ja päättyvät hyväksytyyn testiin. Validointitoimia, joita esiintyy koko ohjelmiston elinkaaren ajan, ovat esimerkiksi tuotteen suunnittelun simulointi. Suunnittelun aikana voidaan jo havaita, jos tuote ei yllä perustason suorituskykyvaatimukseen (todentaminen) tai suorituskykyvaatimukset ovat liian tiukat kaikkien kustannustehokkaiden ratkaisujen jälkeen, joten niitä on sen seurauksena muutettava (validointi).

Boehmin (1988) mukaan avainasiat onnistuneessa V-mallissa ovat iteraatiotodentamisvaiheessa, validointivaiheessa, vaatimusmäärittelyvaiheessa sekä suunnitteluvaiheessa. Tämä tarkoittaa sitä, että todentamis- ja validointivastuussa oleva henkilö analysoi vaatimusmäärittelyt ja niihin liittyvät haasteet sekä raportoi vaatimusmäärittelyistä vastuussa olevalle henkilölle. Boehmin (1988) mukaan vaatimusmäärittelystä vastuussa oleva asiantuntija rajaa ongelman lähteen ja kehittää ratkaisun. Projektipäällikkö ja asiakas hyväksyvät tarjotut ratkaisuehdotukset, jotka saattavat muuttaa vaatimusten päälinjan. Todentamis- ja validointivastuun kantaja analysoi uudet määritykset ja raportoi uusista haasteista, jos tarpeellista. Tässä vaiheessa voidaan myös käyttää simulointia, sillä se on riskitön ja halpa tapa havainnollistaa osa tulevasta järjestelmästä. Sykli jatkuu niin pitkään, että kaikkiin ongelmakohtiin on saatu ratkaisu ja kaikki vaatimukset on täytetty ohjelmiston rakentamisessa. (Boehm, 1988).

Boehmin (1988) mukaan V-malli sisältää neljä tärkeää kriteeriä teknisille vaatimuksille ja suunnittelulle. Niitä ovat täydellisyys, testattavuus, toteutettavuus sekä johdonmukaisuus. Määrittely on täydellinen siltä osin, kun kaikki sen osat ovat valmiita ja jokainen osa on täysin kehittynyt.

Boehm (1988) jatkaa, että järjestelmä voi olla niin kehittynyt, että se täyttää toiminnalliset vaatimukset ja suorituskykyvaatimukset. Tämä merkitsee myös sitä, että tarkasti määritetty järjestelmä on ylläpidettävissä, riittävän lu-

tettava ja taitavasti ohjelmoitu pitämään järjestelmän elinkaaren taseen positiivisena. Tärkeintä on tunnistaa ja ratkaista suorituskykyvaatimuksia koskevat haasteet ennen kuin joukko ohjelmistokehitykseen liittyviä työntekijöitä sitoutuu yksityiskohtaiseen suunnitteluun eli moduulisuunnitteluun. (Boehm, 1988.)

2.2.3 Spiraalimalli

Boehmin (1988) artikkelin mukaan spiraalimallilla toteutetussa ohjelmistokehityksessä ensimmäiseen iteraatioon käytetään kaksi työkuukautta ja toiseen jopa 12 työkuukautta. Spiraalimallia voi pitää realistisena lähtökohtana suurten tietojärjestelmien kehittämiseen riskivetoisuutensa vuoksi. Boehm (1988) esittelee spiraalimallin tyypilliset vaiheet:

- valmistettavan tuoteosan kehitystavoitteet (esim. suorituskyky, toiminnallisuus, kyky mukautua muutoksiin)
- vaihtoehtoiset keinot kyseisen tuoteosan toteuttamisessa (esim. suunnitelma A, suunnitelma B, uudelleenkäyttö, osto)
- vaihtoehtojen soveltamisesta aiheutuvat rajoitukset (esim. kustannukset, aikataulu, käyttöliittymä)

Boehmin (1988) mukaan seuraava vaihe on arvioida vaihtoehtoja tavoitteisiin nähden ja huomioida mahdolliset rajoittavat tekijät. Usein tässä prosessissa tunnistetaan epävarmuustekijät, jotka ovat merkittäviä riskejä ohjelmistokehityksen onnistumiselle. Kun riskejä arvioidaan, seuraava vaihe määräytyy jäljellä olevien riskitekijöiden mukaan.

Jos suorituskyvyn tai käyttöliittymän riskit hallitsevat voimakkaasti ohjelmien kehitystä tai sisäisen rajapinnan hallintaa, seuraava vaihe voi olla niin kutsuttu evoluutiokehitys: minimaalisella ponnistuksella määritellään tuotteen yleinen luonne, prototyypin suunnittelu seuraavalla tasolla ja edelleen yksityiskohtaisempi prototyypikehittely suurimpien riskien ratkaisemiseksi. Jos kyseinen prototyyppi on toiminnallisesti hyödyllinen ja riittävän vahva, se voi toimia vähäriskisenä perustana tulevaisuuden tuotekehitykselle. Myöhemmät riskipainotteiset vaiheet olisivat sarja evoluution kehittämiä prototyyppejä. (Boehm, 1988.)

Jos aiemmat prototyypikehityksen toimet ovat jo ratkaisseet kaikki suorituskykyyn, käyttöliittymän riskeihin, ohjelmien kehittämiseen ja rajapinnan valvontaan liittyvät ongelmat, seuraava vaihe seuraa vesiputousnäkökulmaa. Jokaista ohjelmistomäärittelyn tasoa seuraa validointivaihe, ja suunnitelmien laatiminen seuraavalle kehityskierrokselle. (Boehm, 1988.)

Spiraalimallin huomattava ominaisuus on se, että jokainen kehityssykli päättyy tarkastelujaksoon, johon osallistuvat ensisijaisesti tuotteen kanssa työskentelevät ihmiset tai organisaatiot. Tarkastelu kattaa kaikki edeltävän syklin aikana kehitetyt tuotteet, mukaan luettuna seuraavan kierroksen suunnitelmat ja tarvittavat resurssit niiden toteuttamiseksi. Tarkastelun päätavoitteena on

varmistaa, että kaikki osapuolet sitoutuvat keskenään tiettyyn lähestymistapaan seuraavassa vaiheessa. (Boehm, 1988.)

2.3 Ohjelmistokehitys ketterillä menetelmillä

Highsmith ja Cockburn (2001) painottavat nykyajan teollisuuden ja teknologian kehityksen vauhdin nopeutta, johon organisaatioiden tulee vastata pitääkseen vakaan aseman markkinoilla. Highsmithin ja Cockburnin (2001) mukaan yhä useammin ohjelmistokehityksessä suunnitelman seuraaminen jää toisarvoiseksi ja asiakkaan tarpeiden tyydyttäminen tulee ensisijaisena.

Nopeasti muuttuviin ympäristövaatimuksiin vastaaminen on luonnollista ketterien menetelmien malleilla, sillä niin ohjelmistojen kehityksen kuin kehitykseen kuuluvien prosessien odotetaan toimivan yhä nopeammin ja joustavammin. Nykyään ohjelmistotkin ovat yhä monimutkaisempia ja vaativampia. Asiakkaiden dynaamisiin vaatimuksiin pyritään vastaamaan nopealla toimituksella ja kehityskierroksella. Ketterät menetelmät ovat luonteeltaan joustavia, tuotteliaampia, valmiita vastaamaan käyttäjän vaatimuksiin sekä asiakassitoutuneisuudeltaan parempia. (Matharu, Mishra, Sing & Upadhyay, 2015.)

Abrahamssonin, Salon, Ronkaisen ja Warstan (2002) mukaan ketterillä menetelmillä tehty ohjelmistokehitys on mukautunut vastaamaan uudenlaisiin liikemaailman vaatimuksiin nopean ja ketterän ajattelun ja tekemisen kautta. Laadukasta tulosta ei synny pelkästään tarkkaan hiotun suunnitelman ja alun vaatimusten spesifioidulla keräämisellä vaan reagoimisella teknisiin muutoksiin, asiakkaan tarpeiden muutoksiin sekä mahdolliseen tiimin vaihtuvuuteen. Tällöin ryhmätyöskentelyn ja vuorovaikutussuhteiden merkitys kasvaa.

Highsmith ja Cockburn (2001) korostavat ketterien menetelmien tiimijäsenten vuorovaikutustaitoja ja työskentelyä hyvässä hengessä, sillä niillä varmistetaan onnistunut ohjelmistokehitys. Ketterillä menetelmillä tehty ohjelmistokehitys on avoin prosessi, johon työntekijät voivat tehdä myöhäisessäkin vaiheessa muutoksia – näihin muutoksiin eri tiimien jäsenet ehtivät vielä hyvin reagoimaan (Abrahamsson ym., 2002).

Williams ja Cockburn (2003) lisäävät, että jokaisella ketterällä kehitysmenetelmällä on samantyylliset arvot, jotka tekniikan alan ammattilaiset vuonna 2001 kokosivat ketterien menetelmien manifestiin:

- Yksilöt ja vuorovaikutus tulevat ennen prosesseja ja välineitä
- Toimiva ohjelmisto on arvokkaampi kuin kattava dokumentaatio
- Asiakasyhteistyö on arvokkaampi kuin sopimusneuvottelut
- Muutokseen reagoiminen on tärkeämpää kuin suunnitelman seuraaminen

Ketterien menetelmien toimintatyyli muodostuu lyhyistä iteraatioista, joiden aikana tuotetta kehitetään vaatimusten mukaan niin, että aina iteraation päätteenä saadaan asiakkaalle esiteltyä toimiva tuote. Riippuen käytetystä ketteräs-

tä menetelmästä iteraatiosykli saattaa kestää tunnin, päivän, viikon tai jopa kuukauden ajan. (Abrahamsson ym., 2002.)

Abrahamssonin, Warstan, Siposen ja Ronkaisen (2003) mukaan analyytiset viitekehykset on suunniteltu tarkastelemaan ja ohjaamaan jo olemassa olevien ketterien menetelmien arviointia. Viitekehykset käsittävät kuusi näkökantaa:

- 1 Projektinhallintatuen
- 2 Ohjelmistokehityksen elinkaaren kattavuuden
- 3 Konkreettisten ohjeiden saatavuuden sovellukselle
- 4 Sopeutumiskyvyn varsinaisessa käytössä
- 5 Tutkimustavoitteet
- 6 Empiiriset näytöt

Tulokset osoittavat, että olemassa olevat menetelmät kattavat useat elinkaaren vaiheet, mutta suurin osa ketteristä menetelmistä ei kuitenkaan anna riittävästi tukea ohjelmistokehityksen projektin hallinnalle. Vain muutamat menetelmät tarjoavat konkreettista ohjeistusta ratkaisujen tukemiseksi tai soveltamiseksi erilaisissa kehitystilanteissa. (Abrahamsson ym., 2003.)

Ketterät menetelmät hajottavat tiimi- ja ryhmätyöskentelyn vuoksi perinteiset päätöksenteon rakenteet, sillä ketterissä menetelmissä joko yksi asiantuntija tai asiantuntijaryhmä vastaa omasta kehitysalueestaan. Päätöksenteko ei tällä tavoin kokonaan säily johtajilla vaan delegoituu kunkin ammattilaisen osaamisen mukaisesti. (Williams & Cockburn, 2003.)

Ketterissä menetelmissä asiakas on aktiivinen ja läsnä eri kehitysvaiheissa. Asiakkaan aktiivisuuden taso riippuu käytetystä menetelmästä. Asiakasyhteistyö tarkoittaa, että jokainen ohjelmistokehitykseen liittyvä henkilö (rahoittaja, asiakas, käyttäjä ja kehittäjä) ovat samassa tiimissä. Jokaisen kokemuksen ja asiantuntijuuden sulautuminen yhteen hyvän tahdon kautta antaa mahdollisuuden nopeille suunnanmuutoksille, kestävämmille tuloksille sekä edullisemmalle suunnittelulle. (Highsmith & Cockburn, 2001.) Highsmithin (2002, 4) mukaan vuoden 1995 Goldmanin, Nagelin ja Preissin määritelmä ketteryydestä on erittäin osuva:

Agility is dynamic, context-specific, aggressively change embracing, and growth-orientated. It is not about improving efficiency, cutting costs, or battenning down the business hatches to ride out fearsome competitive "storms". It is about succeeding and about winning: about succeeding in emerging competitive arenas, and about winning profits, market share, and customers in the very center of the competitive storms many companies now fear.

2.3.1 Scrum

Schwaber (2004), yksi Scrumin luojaista, vakuuttaa kyseisen ketterän menetelmän yksinkertaisine prosesseineen sopivan monimutkaisten ohjelmistokehitysten johtamiseen. Brhelin, Methin, Maedchen ja Werderin (2015) mukaan Scrumin käyttäjäkeskeinen suunnittelu parantaa loppukäyttäjän tarpeiden ymmär-

tämistä, mutta haasteena kyseisessä mallissa on itse sovelluksen käytettävyyden testaaminen. Yksinkertaisuudessaan Scrum voi olla hämmentävä, sillä tarkkaa ohjekirjaa prosessien etenemiseen ei ole. Kuitenkin yleisten kokemusten mukaan Scrumin avulla pystytään rakentamaan kustannustehokkaasti ja nopeasti laadukas järjestelmä. (Schwaber, 2004.) Schwaber (2004) listaa Scrumin edut seuraavasti:

- monimutkaisten projektien johtaminen
- tehokas odottamattomien tai nopeasti muuttuvien tapahtumien hallinta
- komentoketjun yksinkertaistaminen itsenäisillä kehitystiimeillä
- selkeämpien tietojen ja palautteen saanti asiakkailta
- 30-päivän iteraatiosykli ("rakenna ja lanseeraa") varmistavat, että asiakas saa tuotetoimituksen aiemmin
- ohjelmistokehityksen suunnitteluun käytetyn ajan ja resurssien vähentyminen
- väärinkäytösten välttäminen säännöllisellä tarkastelulla, raportoinnilla sekä hankkeiden hienosäädöllä
- moninaisten tiimien tukeminen laaja-alaisissa hankkeissa riippumatta geologisesta sijainnista
- sijoitetun pääoman tuoton maksimointi

Kyseisessä ketterässä kehitysmenetyksessä kehittäjät tapaavat palaverin merkeissä joka päivä. Tapaaminen kestää noin 15 minuuttia ja kattavammat iteraatioarviot käydään läpi jokaisen 30-päiväisen iteraatio- eli kehityssyklin lopussa. (Highsmith & Cockburn, 2001).

Schwaber (2004) täsmentää iteraation alkavan tiimien tapaamisesta, jossa käydään läpi, mitä tiimien tulee tehdä ja mitkä ovat tiimien tavoitteet. Tämän jälkeen tiimit vetäytyvät omilleen, jossa ne työskentelevät parhaimman tuloksen saavuttamiseksi. Iteraation loppuvaiheessa tiimit tapaavat taas, jossa ne esittelevät kehitystoimenpiteen toimivuuden, jotta sidosryhmät (asiakkaat ja muut ryhmät) voivat tarkastaa ohjelmistokehityksen ajantasaiset mukautukset.

Scrumissa keskeisessä asemassa on tiimin lisäksi kaksi eri roolia: tuoteomistaja ja Scrum-mestari. Tuoteomistaja on henkilö, joka on vastuussa tuotteen backlogin hallinnomisesta, jotta ohjelmistokehityksen arvo saataisiin maksimoitua. Tuoteomistaja edustaa ohjelmistokehityksen projektin sidosryhmiä. Tiimi taas koostuu niistä ryhmän henkilöistä, jotka ovat vastuussa järjestelmän kehityksestä jokaisessa iteraatiossa. Scrum-mestari on henkilö, joka on vastuussa ketterän menetelmän prosesseista, oikeasta toteutuksesta sekä tiimin etujen maksimoimisesta. (Schwaber, 2004.)

Scrumissa on käytettävissä backlogit, jotka sisältävät tuotteen toiminnallisuudet, ominaisuudet, ja niillä myös hallitaan ohjelmistokehityksen etenemistä. Tuotteen backlog sisältää listan liiketoiminnallisista ja teknisistä tuotteen piirteistä. Kyseisen backlogin priorisoi tuoteomistaja. Tuotteen backlog on aloituspiste, jossa tuoteomistaja valitsee seuraavassa sprintissä tuotteen julkaistavat ominaisuudet. Tuoteomistajan päätös johtaa toiseen sprintin backlogiin. Sprin-

tissä julkaistaan sen iteraation aikana kehitelty tuote, jolloin sidosryhmät päässevät analysoimaan järjestelmää kokonaisuutena, jonka jälkeen prosessi alkaa alusta. (Schwaber, 2004.)

2.3.2 Extreme Programming

Extreme Programming -menetelmä (XP) on yksi arvostetuimpia ketteriä kehitysmalleja, jonka on kehittänyt Beck, Cunningham ja Jeffries (Highsmith, 2002). Menetelmän tavoite on vähentää muutoksista aiheutuvia kuluja läpi ohjelmistokehitysprojehtin. Kun Scrum keskittyy hallitsemaan ja johtamaan ohjelmistohankkeita, Extreme Programming keskittyy määriteltyyn ohjelmointimalliin, johon sisältyy pariohjelmointi, koodausstandardit, testauskehitys ja jatkuva integraatio. (Mahnic & Drnovscek, 2005.)

Paetsch ym. (2003) esittävät XP:n arvon muodostuvan yksinkertaisuudesta, viestinnästä, palautteesta ja rohkeudesta. Se toimii tuomalla koko tiimin yhteen yksinkertaisilla käytännön harjoitteilla. Riittävän palautteen ansiosta tiimi näkee missä vaiheessa ohjelmistokehitystä he ovat. XP ei niinkään korosta vaatimusmäärittelyn tekniikoita yksityiskohtaisesti vaan yleisesti ohjelmistokehitysprosessia ja mitä tämän prosessin aikana tulee tuottaa. Erityisesti suunnittelu- vaiheessa keskiössä ovat aivoriihet sekä priorisointi. (Paetsch ym., 2003.)

Matharun ym. (2015) mukaan Extreme Programming -menetelmässä käytetään kertomuskortteja töiden jouduttamiseksi. Kertomuskortti on käyttäjän kuvaus ominaisuudesta tai vaatimuksesta, joka tarjoaa arvoa asiakkaalle. Ohjelmistokehittäjät jakavat jokaisen tarinakortin sarjaksi pienempiä tehtäviä, jotka myöhemmin priorisoidaan asiakkaan toimesta. Extreme Programmingiin kuuluu myös yksinkertaisuus. Suunnittelu ja itse ohjelmointikoodi on aluksi mahdollisimman yksinkertaisesti tehty, jotta asiakkaan toivomat toiminnallisuudet saadaan lisättyä helposti ja jotta ohjelmointikehitystiimin on helppo ymmärtää suunnittelua ja koodia. (Matharu ym., 2015.)

Paetsch ym. (2003) kirjoittavat julkaisussaan, että ennen kuin kertomuskortteja voidaan käyttää, asiakkaiden tulee tietää, miten he haluavat järjestelmän toimivan. Tämä prosessi nähdään aivoriihityöskentelynä. Tietyn toimintam ominaisuuden pohtiminen lisää ideoita ja käyttäjätarinoita. Jokainen tarina käydään avoimesti läpi ennen täytäntöönpanoa. Aluksi kehittäjät selvittävät asiakkailta haluttuja järjestelmän yksityiskohtia, jotta kehitystiimi voi arvioida toteutuksen.

Matharu ym. (2015) tähdentävät, että XP:n ominaisiin työtapoihin kuuluu erittäin tiivis kommunikaatio asiakkaan kanssa jokaisessa iteraatiossa, jossa työtehtävät kirjoitetaan ennen ohjelmointia. Suunnittelulla taataan paras tulos ja laatu. XP:ssä käytetään myös pariohjelmointia, joka takaa dynaamisen työskentelyn ja tehokkaamman viestinnän.

Ohjelmistokehitystiimin tulee tuottaa ensimmäinen tuote viikkojen sisällä. Tiimin tulee kehitellä yksinkertaisia ratkaisuja, jotta tulevaisuudessa muutoksia olisi vähemmän ja ne olisivat helpommin käsiteltävissä. Tavoitteena on parantaa tuotteen laatua jatkuvasti, jotta seuraavan vaiheen toteutus tulisi edulli-

semmäksi. Tuotetta testataan jatkuvasti, jotta mahdolliset tulevat kulut voidaan minimoida. XP:n iteraatiosykli kestää kahdesta kolmeen viikkoa. (Highsmith & Cockburn, 2001.)

2.3.3 Dynaaminen ohjelmistokehitysmenetelmä

Dynamic Systems Development Method (DSDM) on karkeasti suomennettu dynaamiseksi ohjelmistokehitysmenetelmäksi (Highsmith & Cockburn, 2001). Voigtin (2004) mukaan DSDM perustuu yhdeksään perusperiaatteeseen:

1. käyttäjän aktiiviseen läsnäoloon (välttämättömyys),
2. tiimien valtuuksiin tehdä päätöksiä,
3. toimitukseen keskittymiseen,
4. tarkoin määrättyihin kriteereihin hyväksytyille toimituksille,
5. iteratiiviseen ja inkrementaaliseen kehitykseen (pakollinen),
6. palautettavissa oleviin muutoksiin (kaiken on oltava palautettavissa),
7. kehitysprosessin elinkaareen integroituihin testauksiin,
8. yhteistoiminnalliseen lähestymistapaan sekä
9. yhteistyökykyiseen lähestymistapaan.

Voigt (2004) jatkaa artikkelissaan, että suurimmassa osassa ketteriä kehitysmalleja myös DSDM:n kustannustehokkaalle toteutukselle tärkeintä on käyttäjän tai asiakkaan aktiivinen läsnäolo. Näin voidaan välttyä ylimääräisiltä virheiltä.

Voigtin (2004) mukaan DSDM-kehitysmenetelmissä suositellaan käytettävän pientä tarkasti valikoitua käyttäjäryhmää jatkuvasti; toisin kuin tarkasteluistunnoissa, joita hyödynnetään muun muassa Scrumissa. Jatkuvuus on yksi keskeisiä teemoja DSDM:n jokaisessa vaiheessa. Jotta mahdollisilta ylimääräisiltä siirtymävaiheilta, hidastuksilta sekä ylimääräisiltä kuluilta säästyttäisiin, tiimin tulisi päättää itsenäisesti käytännön vaatimuksista, toimintojen kehityksen elinkaarista, toimintojen ja määritysten priorisoinnista sekä teknisten ratkaisujen yksityiskohdista. Lyhyet iteraatiosykliet varmistavat, että virheet havaitaan nopeasti ja ne voidaan korjata helposti. (Voigt, 2004.)

Voigt (2004) jatkaa, että DSDM tukee ensisijaisesti liiketoiminnan tarpeiden tyydyttämistä. Se vaatii suunnittelun aikataulutusta. Myös DSDM-kehitysmenetelmissä on ratkaisevan tärkeää määritellä avainkysymykset, jotka edellyttävät vankkaa suunnittelua. DSDM jaetaan pieniin ominaisuusosa-alueisiin, jotta ohjelmistokehityksen kompleksisuus pysyy hallittavissa. Jokaisessa lanseerauksessa lisätään uusia ominaisuuksia siihen asti, että vaatimusmäärittelyt on saavutettu. (Voigt, 2004.)

Voigt (2004) esittää, että Scrumissa kuin myös DSDM:ssä vaaditaan testausta aivan kehityspolun alusta lähtien, ja testitulosten dokumenttejakin tarkastetaan kontrolliryhmän toimesta. Avoin ja rohkaiseva viestintä on avainasemassa teknisen tiimin ja muun henkilöstön välisessä tuottavassa yhteistyössä. Ilman positiivista ilmapiiriä, luottoa tai rehellisyyttä, DSDM:n kehitysmallin

on vaikeaa saavuttaa haluttuja tuloksia, sillä vaatimusten kerääminen ja myöhemmin rehellisen palautteen saaminen testituotteista on vaikeaa. (Voigt, 2004.)

Voigtin (2004) mukaan DSDM vaatii ainoastaan kuuliaisuutta yhdeksään peruseriaatteeseen. Nämä toimet voivat olla enemmän tai vähemmän ketteriä, riippuen tilanteesta ja rajoitteista, joita työn aikana kohdataan. Abrahamsson, Warsta, Siponen ja Ronkainen (2003) vahvistavat, että DSDM on itsenäinen metodi, joka tarjoaa täydellistä tukea koko kehitystyön elinkaaren vaiheissa.

DSDM:n ajatuksena on, että ”suunnitelma ei aina selviä ensimmäisestä kehityspäivästä”. Tämän vuoksi DSDM:n yhteydessä suositaan käytettävien muitakin menetelmiä. Loppujen lopuksi DSDM ei pyri ratkaisemaan aivan jokaista kehityskohdetta. Jos yhdeksää periaatetta ei voida toteuttaa, dynaamista ohjelmistokehitysmenetelmää ei suositella käytettävän. (Voigt, 2004.)

Abrahamsson kollegoineen (2003) kiteyttää, että DSDM voidaan pitää ensimmäisenä todella ketteränä ohjelmistokehitysmenetelmänä, kun projektissa on tarkoitus ensin korjata resursseja ja ajankäyttöä, jonka jälkeen vastaavasti säätää tuotteen toiminnallisuuksia eikä niinkään aloittaa niistä.

2.3.4 Yhteenveto

Luvussa 2.1 esiteltiin ohjelmistotuotannon määritelmä ja luvussa 2.2 esiteltiin ohjelmistotuotannon perinteiset elinkaarimallit, joihin koko ohjelmistotuotannon kehityksen prosessit ja nykyaikaiset kehitysmenetelmät perustuvat. Luvut 2.1 ja 2.2 nostavat esille kattavasti ohjelmistotuotannon kehityksen historian ja mallien kehityksen kaaren. Luvut antavat kattavan pohjan seuraavassa luvussa käsiteltäviin tutkielmaan valittuihin ketteriin kehitysmenetelmiin. Luvuissa tuodaan esille, kuinka teknologian, markkinoiden ja kulttuurin muutos loivat tilanteen, jossa perinteiset mallit eivät enää pystyneet vastaamaan tarpeeksi nopeasti markkinoiden ja asiakkaiden tarpeisiin ja tämän takia kehitysmenetelmien oli muututtava.

Luvussa 2.3. esiteltiin yleisimmät ohjelmistokehityksen ketterät menetelmät. Ketterät kehitysmenetelmät luotiin vastaamaan nopeasti asiakkaiden tarpeisiin ja parempaan asiakkaiden tarpeiden ymmärtämiseen tuomalla asiakas osaksi kehitysprosesseja sekä vastaamaan markkinoiden ja teknologian nopeisiin muutoksiin. Luvussa tuodaan myös esille, että ketterien kehitysmenetelmien yhtenäisyys piilee nopeissa iteraatioissa, joissa kuluttaja tai asiakas on tiiviisti mukana, joustavuudessa, tuottavuudessa sekä valmiudessa vastata muutoksiin. Dokumentaatio ei myöskään ole ketterissä menetelmissä suuressa roolissa toisin kuin perinteisissä malleissa. Tällä tavoin on pyritty pitämään korkea laatu muissa kehityksen aspekteissa aikarajoineen. Tutkielmaan valittuja yleisimpiä ketteriä ohjelmistokehitysmalleja ovat Scrum, Extreme Programming sekä Dynaaminen ohjelmistokehitysmenetelmä.

Luvusta 2.3 voidaan siis päätellä, että nopea prototyypin tuotto, testaus, tiimivetoisuus, viestintä ja asiakkaan tiivis mukanaolo kehityksessä ovat tehokkaita ja tarpeellisia ratkaisuja nykyajan haasteisiin ohjelmistokehityksessä. Ohjelmistokehitysala törmää jatkuvasti kompleksisempiin ohjelmistohaasteisiin,

dynaamisiin asiakasvaatimuksiin, pienempiin budjetteihin ja tiukkoihin aikatauluihin. Ketterät menetelmät pystyvät vastaamaan näihin haasteisiin kustannustehokkuudellaan ja lisäarvon tuotollaan.

Huomioitava on kuitenkin, että tutkielmaan on valittu vain muutama ketterä kehitysmenetelmä tutkielman luonteen ja rajallisten resurssien vuoksi. Tutkielmasta puuttuu valideja ketteriä kehitysmenetelmiä, kuten Kanban tai Rapid-application development (RAD). Tutkielmassa ei oteta kantaa käytettyjen ketterien kehitysmenetelmien toteutumisesta sellaisenaan organisaatioissa.

3 TYÖNTEKIJÄN SITOUTUMINEN

1950-luvulta lähtien on julkaistu satoja artikkeleita sitoutumisen konseptiin liittyen, kun sitoutumisen huomattiin vaikuttavan organisatoriseen käyttäytymiseen ja organisaation menestymiseen. Sitoutuminen nousi pinnalle työntekijän ja organisaation välisiä suhteita käsittelevissä tutkimuksissa. Uskottiin, että sitoutunut työntekijä olisi yritykselle voimavara sen vuoksi, että yrityksen liikevaihto ja suorituskyky kasvoivat sekä työntekijän poissaolot vähenivät. (Abrahamsson, 2002.)

Kuitenkaan edellä mainitut edut eivät ole ainoita voimavaroja, joita yritys voi saada sitoutuneesta työntekijästä. Sitoutunut työntekijä pysyy organisaatiossa sen vahvempien ja heikompien kausien läpi, työskentelee organisaatiolle täydellä teholla koko päivän, suojaa yrityksen omaisuutta sekä jakaa yhtiön uskomukset ja tavoitteet (Meyer & Allen, 1997.) Sitoutunut työntekijä on myös onnellinen ja tyytyväinen, hän hyödyntää vapaasti resurssejaan halutun tuloksen saavuttamiseksi ja jopa rikkoo sääntöjä tarpeen vaatiessa. Vaikka edellä mainittuja ja monia muita luonnehdintoja sitoutumisesta on esitetty julkisesti, vain muutama on validoitu ja hyväksytty. (Abrahamsson, 2002.)

Luvussa kolme esitetään sitoutumisen määritelmä ja sitoutumisen eri muodot. Luvun päätteeksi syvennytään työntekijän sitoutumiseen eri ketterien menetelmien ohjelmistokehitysmalleissa.

3.1 Sitoutumisen määritelmä ja sitoutumisen teemat

Mitä sitoutuminen on? Sitoutumisen konsepti on Abrahamssonin (2002) mukaan hankala määritellä tarkkaan, sillä tulkintoja aiheelle löytyy lukemattomia. Sitoutumisen erilaisia määritelmiä tutkimuksista löytyy 25 erilaista. Näin moninainen selvitys yhdelle käsitteelle johtuu tarkan sitoutumismallin puutteesta. Sitoutumistutkimuksissa termiä käytettiin yleisesti viittaamaan anteeksiantamiseen. Brown (1996) kiteyttää sitoutumisen seuraavasti: sitoutuminen tiettyyn kokonaisuuteen on erillinen ja monisäikeinen ilmiö, joka saattaa poiketa toisista

sitoutumisen malleista, riippuen miten tietyt tekijät (jotka liittyvät kaikkiin sitoumuksiin) ovat havaittu ja arvioitu yksilön toimesta.

Brownin (1996) mainitsemat tekijät ovat yhteisiä kaikille sitoutumisen muodoille. Näitä ovat keskittyminen, ehdot ja vahvuus. Kaikella sitoutumisella on tarkoitus: sitoutumisen kohde ja saavutettava päämäärä. Kohde voi olla organisaatio, työryhmä, projektin tavoite tai idea. Sitoutumisen vahvuus vaihtelee sitoutumisen kohteen ja henkilökohtaisen merkityksen mukaan.

Pitkään on tieteellisissä julkaisuissa ehdotettu, että sitoutuminen on jatkuva muuttuja. Yksilöitä pidetään enemmän tai vähemmän sitoutuneina sen sijaan, että he olisivat yksinkertaisesti ja selkeästi sitoutuneita tai sitoutumattomia. (Kiesler, 1971.)

Abrahamsson (2002) korostaa, että Meyerin ja Allenin (1991) havainnot ehdotetuista määritelmistä heijastavat kolmea yleistä teemaa: emotionaalinen suuntautuminen, kustannusperusteinen ja moraalinen velvollisuus. Meyer ja Allen (1991) esittävät, että sitoutuminen on psykologinen liitos, joka voi olla kolmea muotoa: affektiivinen, normatiivinen ja jatkuvuustyyppinen. Näitä muotoja voidaan myös pitää perustana sitoutumisen käsitteelle ja myöhemmille malleille, jotka aiheuttavat kiintymistä (Becker, 1992).

Kelmanin (1958) tutkimus asenteesta ja käyttäytymisestä osoittaa sitoutumisen olevan kolmea eri mallia: vaatimustenmukaisuus tai instrumentaalinen osallistuminen, joka liittyy erityisiin ulkoisiin etuihin, tunnistautuminen ja tunnistaminen, joka perustuu haluun liittyä johonkin yhteisöön sekä osallisuus, joka perustuu yksittäisten ja organisatoristen arvojen yhtenevyyteen.

Abrahamsson (2002) ehdottaa, että noudattaminen tai instrumentaalinen sitoutuminen on eräänlainen sitoutumisen vastakohta ja sen sisällyttäminen perinteisen sitoutumisen määritelmään vain sekoittaisi konseptia entisestään. Sen sijaan tunnistautuminen, tunnistaminen ja osallisuus olisivat sitoutumisen peruskäsitteitä. Nämä mekanismit edesauttaisivat affektiivisen sitoutumisen muodostumista. Meyerin ja Allenin (1991) sitoutumisen muodot on tunnistettu organisatorisen sitoutumisen tutkimuksissa, ja ne ovat myös joustavia malleja muihin sitoumustavoitteisiin nähden. Työntekijä, olkoon hän sitten yritysjohtaja tai ohjelmistokehittäjä, voi olla sitoutunut tiettyyn hankkeeseen neljässä sitoutumisen muodossa. Näitä ovat affektiivinen sitoutuminen, jatkuva sitoutuminen, normatiivinen sitoutuminen ja instrumentaalinen sitoutuminen.

Affektiivinen sitoutuminen on Abrahamssonin (2002) mukaan sitoutumisen malleista selvin. Affektiivinen sitoutuminen viittaa tekijän kiinnittymiseen, tunnistamiseen, ja osallistumiseen asianomaisessa yhteisössä. Se sisältää joukkoon kuulumisen tunteen ja psykologisen sitoutumisen tunteen kohteeseen (Hartmann, 2000). Tämä on kokonaisuus, joka voi koostua organisaatiosta, hankkeesta, ohjaajasta ja työtovereista. Mikä tahansa, mikä antaa yhteisöllisyyden tunteen ja näin ollen luo merkityksen työntekijälle. (Abrahamsson, 2002.) Meyer ja Allen (2004) kiteyttävät, että affektiivisessä sitoutumisessa työntekijä pysyy organisaatiossa, koska hän haluaa.

Jatkuvasta sitoutumisesta Abrahamsson (2002) esittää, että sitoutumisen jatkuvuus viittaa tietoisuuteen mahdollisista kustannuksista. Nämä kustannuk-

set voivat olla taloudellisia tai ei-taloudellisia, vaihtoehtojen puutteista johtuvia tai sitovia organisatorisia järjestelyjä; kuten palkkarakenne, jossa johtajan suorituskyky on kytköksissä menestymiseen. Tästä voidaan sanoa, että ensisijainen sitoutumisajuri on jatkuva sitoutuminen. Toisaalta vaihtoehtoja on muitakin, mutta palkkarakenne yleensä vaatii sitoutumista myös jatkossa. (Hartmann, 2000.) Meyer ja Allen (2004) tiivistävät, että sitoutumisen jatkuvuudessa työntekijä jää organisaatioon, koska hän tuntee, että hänen pitäisi jäädä.

Normatiivinen sitoutuminen heijastaa velvollisuutta jatkaa jäsenyyttä yhteisessä yksikössä. Normatiivisen sitoutumisen käsite selitettiin alun perin ”sisäiseksi normatiiviseksi paineiksi, jotka toimivat tavalla, joka täyttää organisatoriset tavoitteet ja edut”. Toinen yleisesti käytetty termi on moraalinen sitoutuminen (Jaros, Jermier, Koehler & Sincich, 1993). Jos normatiivinen sitoutuminen on hallitseva sitoutumisen muoto, se voi kestää vain niin kauan kuin sisäinen tai ulkoinen paine on olemassa. (Abrahamsson, 2002.) Meyerin ja Allenin (2004) mukaan normatiivisen sitoutumisen kautta työntekijä jää organisaatioon, sillä hänestä tuntuu, että hänen on pakko jäädä.

Instrumentaalinen sitoutuminen viittaa Abrahamssonin (2002) mukaan osallistumiseen erityisistä ulkoisista syistä. Se on vähiten perinteinen sitoutumisen muoto siinä mielessä, kuinka sitoutumista ajatellaan yleisesti. Tässä tapauksessa sitova vaikutus liittyy ulkoisiin palkkioihin. Instrumentaalinen sitoutuminen osoittaa, että hankkeen tai organisaation tavoitteet eivät kiinnitä työntekijää henkilökohtaisen tunnustuksen saamisen toivossa vaan pikemminkin halu saavuttaa muita hyötyjä ja välttää rangaistusta. Tämän tyyppinen sitoutuminen eroaa affektiivisista, jatkuvista ja normatiivisista muodoista siinä, että mieli on suunnattu kohti etuja ja hyötyjä eikä itse tavoitetta. Jos instrumentaalinen sitoutuminen on hallitseva malli, sen kestävyys yhdistetään instrumentaaliin osallistumismenettelyyn liittyviin arvoihin. Ketterän ohjelmistokehityksen kontekstissa edut voidaan tulkita kehitystyön aloitteen hyödyiksi. Palkkio voi olla muutaakin kuin rahallista arvoa. (Abrahamsson, 2002.)

Abrahamsson (2002) huomauttaa, että sitoutumista on yritetty määrittellä ja konseptoida monella tapaa jo usean kymmenen vuoden ajan. Tutkimuksilleen eniten näyttöä ovat saaneet Meyer ja Allen (1991) sekä Brown (1996), joita pidetäänkin sitoutumiskäsitteen periaatteen perustana. Abrahamsson (2002) määrittelee sitoutumisen seuraavasti: sitoutuminen on kiintymyksen tila, joka määrittelee tekijän (esimerkiksi yksilö, ryhmä tai organisaatio) ja entiteetin (sitoutumisen kohde) välisen suhteen. Nämä suhteet voivat ottaa erilaisia muotoja tai malleja (affektiivinen, jatkuva, normatiivinen ja instrumentaalinen), jotka jakavat tiettyjä yhteisiä näkökohtia (painopiste, vahvuus, ehdot ja kestävyys) kaikissa sitoutumisen muodoissa. (Abrahamsson, 2002.)

3.2 Motivaatio sitoutumisen muodoissa

Motiiveista ja motivaatiosta puhutaan tutkimuksissa paljon. Niiden merkitys on todistettu olevan suurimpia työntekijän tuottavuuteen ja laadun johtamiseen

vaikuttavia tekijöitä, joita on edelleen hankala ohjata. Motivaation puute taas on selvästi nähty monissa ohjelmistokehitykseen liittyvissä projekteissa epäonnistumisen suurimmaksi syyksi. Jotkin ohjelmistokehityksen alan tutkimukset viittaavat siihen, että perinteiset lähestymistavat motivaatioon kyseisellä alalla saattavat olla vanhentuneita. Ne ovat keskittyneet palkintoihin ja tunnustuksiin, kun taas jotkut asiantuntijat ovat tunnistaneet ohjelmistoinsinöörien persoonallisuusprofiilin olevan erilainen juurikin työn luonteen vuoksi. Kyseisiä työntekijöitä sitovatkin haastavat tekniset ongelmat ja niiden ratkaiseminen. (Beecham, Baddoo, Hall, Robinson & Sharp, 2008.)

Motiivi ja motivaatio ovat kuitenkin sitoutumiseen liittyviä tekijöitä; ilman henkilökohtaista motivaatiotakin työntekijä voi olla sitoutunut ainakin hetkellisesti (Beecham ym., 2008). Näin ollen motivaatio voi sisältyä Meyerin ja Allenin (1991) mallin kolmeen muotoon: sitoutumisen jatkuvuuteen, normatiiviseen tai instrumentaaliseen sitoutumiseen (Abrahamsson, 2002). Sharpin, Baddoon, Beechamin, Hallin ja Robinsonin (2009) tutkimuksessa löydettiin ohjelmistoinsinöörin motivaattoritekijät, jotka havainnollistetaan taulukossa 1.

Taulukko 1 on koottu tutkielman tekijän toimesta Sharpin ym. (2009) motivaattoritekijöistä sekä Abrahamssonin (2002) listaamista sitoutumisen muodoista. Taulukko 1 osoittaa eri motivaattoritekijöiden kuuluvan aina johonkin sitoutumisen malliin.

TAULUKKO 1 Motivaattoritekijät sitoutumisen malleissa

Motivaattorit	Sitoutumisen mallit
Palkkiot ja kannustimet	Instrumentaalinen
Kehittämistarpeiden huomioiminen	Jatkuva
Työn vaihtelevuus ja mukautuvuus	Normatiivinen
Urapolku mahdollisuus	Jatkuva
Vastuullisuus ja voimaantuminen	Instrumentaalinen ja affektiivinen
Hyvä johtaminen	Affektiivinen
Yhteenkuuluvuuden tunne ja tuki	Affektiivinen
Työn ja muun elämän tasapaino	Affektiivinen
Työ menestyksekkäässä yrityksessä	Jatkuva
Työntekijöiden osallistuvuus	Affektiivinen ja jatkuva
Palaute	Jatkuva ja instrumentaalinen
Tunnustus	Instrumentaalinen
Tasa-arvoisuus	Jatkuva
Luottamus ja arvostus	Affektiivinen
Teknisesti haastava työ	Normatiivinen ja jatkuva
Vakaa työympäristö	Normatiivinen ja jatkuva
Identifioiminen työtehtävien kautta	Normatiivinen
Itsenäisyys	Jatkuva
Asianmukaiset työolot	Jatkuva
Merkittävän työn tekeminen	Affektiivinen
Riittävät resurssit	Normatiivinen

3.3 Sitoutuminen ketterien menetelmien näkökulmasta

Boehmin ja Papaccion (1988) mukaan kaikista houkuttelevimmat yksittäiset strategiat ohjelmistojen tuottavuuden parantamiseksi ovat:

1. Kirjoittamalla vähemmän koodia: uudelleenkäyttämällä ohjelmistojen komponentteja, kehittämällä ja käyttämällä korkealaatuisia koodikie-
liä.
2. Saamalla parhaimman suorituksen irti työntekijöistä: paremmalla
ihmisten johtamisella, hankkimalla oikeanlaisia työntekijöitä, käyt-
tämällä kannustimia ja parantamalla työympäristöä.
3. Välttämällä uudelleensuunnittelua: paremmalla riskienhallinnalla,
prototyypin kehittämisellä, inkrementaalaisella kehityksellä, tieto-
koneavusteisella ohjelmistosuunnittelulla ja moderneilla ohjelmointi-
käytänteillä.
4. Kehittämällä ja käyttämällä integroitua tukiympäristöjä.

Shore ja Martin (1989) tutkimuksen mukaan asenteet organisaatiota kohtaan, kuten organisatorinen sitoutuminen, ovat lähekkäin yhteydessä organisaation tuloksiin. Kofman ja Senge (1993) korostavat, että organisaatiot ovat yhteiskun-

nan vaikuttavimpia yhteisöjä; muutoksista johtuvat suurimmat huolenaiheet ovat organisaatiokulttuurin toiminnan häiriöt ja viestintäongelmat. Sitoutuminen on siis sidoksissa suorassa suhteessa organisaation suorituskykyyn ja toimintaan (Shore & Martin, 1989).

Moe, Dingsoyr ja Dybå (2008) mainitsevat tutkimuksessaan, että ohjelmistokehityksen prosessien onnistuminen riippuu merkittävästi tiimin suorituskyvystä. Kaksi tärkeää tekijää tiimin maksimaalisen suorituskyvyn saavuttamiseksi ovat palaute ja toimiva viestintä. Ketterä kehitystyö perustuu tiimityöhön, toisin kuin yksilöivät työt, joita on esimerkiksi suunnittelukeskeinen kehitystyö. Suunnittelukeskeisen mallin mukainen tiimi koostuu usein itseohjautuvista itsenäisistä ammattilaisista kuten Scrumissa saattaa olla.

Cockburn ja Highsmith (2001) mainitsevat ketterän ohjelmistokehityksen työntekijöiden olevan tarpeeksi hyviä, jos he voivat vaivattomasti suoriutua tehtävistään jokaisessa vaiheessa. Jos he eivät ole riittävän taitavia, ei prosessi itsessään korjaa heidän riittämättömyyttään: "ihmisillä on valta prosesseissa osaamisen vuoksi".

Cockburn ja Highsmith (2001) jatkavat, että käyttäjien ja johtajien tuen puute voi lakkauttaa koko kehitystyön: "politiikalla on valta ihmisissä". Riittämätön tuki voi estää taitavia työntekijöitä tekemästä työtään. Ihmiset voivat työskennellä yhdessä hyvällä viestinnällä ja vuorovaikutuksella huomattavan korkealla laatutasolla. Tämä nousee esiin aivoriihi- ja ongelmanratkaisuistunnoissa XP:n kehitysmallissa. Ketterät projektitiimit keskittyvät lisäämään sekä yksilöllisiä valmiuksia että yhteistyön tasoja menestyäkseen. Ketterät menetelmät on luotu hyödyntämään jokaisen yksilön ja kunkin tiimin ainutlaatuisia vahvuuksia. (Cockburn & Highsmith, 2001.)

Ketterät kehitysmenetelmät eivät kuitenkaan ole jokaista organisaatiota varten. Jos yritys sopeuttaa ketteriä menetelmiä prosessikeskeiseen ja yhteistyökyvyttömään organisaatioon, tulee kehitystyö todennäköisesti epäonnistumaan. Parhaita tuloksia saadaan, jos ketterät kehitysmenetelmät on käytössä ihmiskeskeisessä yhteistyöhön perustuvassa organisaatiokulttuurissa. (Cockburn & Highsmith, 2001.)

Williams ja Cockburn (2003) listaavat kolme riskialuetta, jotka voivat heikentää ohjelmistokehityksen ketteryyttä:

- Ongelmat, jotka johtuvat ketterästä lähestymistavasta: skaalautuvuus, kriittisyys, muotoilun yksinkertaisuus, henkilöstön vaihtuvuus ja osaaaminen.
- Riski, joka johtuu suunnitelmalähtöisestä lähestymistavasta: uudet vaatimukset, jatkuvat muutokset, nopeiden tulosten tarve, henkilöstön osaaaminen.
- Yleiset ympäristötekijät: tekniset epävarmuustekijät, erilaiset sidosryhmät ja monimutkaiset järjestelmät.

Edellä mainitut haasteet voidaan voittaa hyvällä tiimihengellä, vuorovaikutustaidoilla ja hyvällä henkilöstöjohtamisella, jota ketterissä kehitysmenetelmissä tarvitaan (Williams & Cockburn, 2003).

Sitoutuminen on todettu jokaisessa ketterässä kehitysmenettelmissä kriittiseksi menestystekijäksi. Sitoutuminen tavoitteeseen tai tulokseen, psykologinen tunne yhteenkuuluvuudesta, positiivinen ilmapiiri ja yhteisöllisyys ovat niin tärkeitä menestyksen kulmakiviä, että ketteriin kehitysmalleihin on luotu erilliset roolit pitämään huolta juuri näistä menestyksen avaimista. (Williams & Cockburn, 2003.)

Abrahamsson (2002) mukaan sitoutumisen verkostoihin kuuluu kolme erilaista näkökulmaa: strateginen, toiminnallinen ja yksilöllinen. Sitoutumisen verkostoja voidaan tavata eri ohjelmistokehityksen vaiheissa. Taulukkoon 2 on tutkielman tekijä yhdistänyt eri sitoutumisen mallit ja Abrahamssonin (2002) Sitoutumisen rooli ohjelmistokehityksessä -taulukon. Taulukossa 2 havainnollistetaan sitoutumisen verkostot, sitoutumista edistävät tekijät sekä sitoutumisen mallit ohjelmistokehityksen eri vaiheissa.

TAULUKKO 2 Sitoutumisen prosessi ketterässä ohjelmistokehityksessä

Sitoutumisen verkostot	Sitoutumisen mallit	Prosessin aloitus	Toimenpiteet	Prosessin lopetus
		Edistävät tekijät	Edistävät tekijät	Edistävät tekijät
Strateginen	Jatkuva, Instrumentaalinen, Normatiivinen	Psykologinen, Toiminta, Projekti	Projekti	Projekti, Psykologinen
Toiminnallinen	Jatkuva, Instrumentaalinen, Normatiivinen	Toiminta, Projekti	Projekti	Psykologinen, Sosiaalinen
Yksilöllinen	Affektiivinen	Psykologinen	Psykologinen	Psykologinen

Sitoutuminen toiminnallisessa verkostossa, joka sisältää projektin, ja jossa prosessin lopetus on sosiaalinen, voi työntekijä tuntea vastuuta projektista ja sen etenemisestä. Niin kauan kuin ohjelmistokehitysprojekti on käynnissä, on työntekijän velvollisuus suorittaa työtehtävänsä tunnollisesti loppuun saakka. Tässä tapauksessa sitoutumisen malli on normatiivinen. Toisaalta työntekijä saattaa ymmärtää, mitä kustannuksia mahdollisen ohjelmistokehitysprojektin keskeyttäminen voi tuoda, jolloin sitoutumisen malli on jatkuva. (Abrahamsson, 2002.)

Brhel ja kumppanit (2015) kiteyttävät, että organisatorinen tuki ja kulttuurin muutos täytyy huomioida ohjelmistokehityksen prosesseissa, ihmisten välisessä yhteistyössä sekä sosiaalisen tyytyväisyyden ja tuotteliaisuuden ylläpitämisessä. Organisatorinen tuki ja kulttuuri ovat kriittisiä tekijöitä tiimityöskentelyn menestyksessä.

3.3.1 Sitoutuminen Scrumin näkökulmasta

Liian usein ohjelmistokehityksen tiimeille on annettu täysi vastuu tuotteen lanseeraamisesta, kun johtoryhmälle (projektin vetäjät ja prosessien johtajat) on annettu päätösvalta. Vastuiden kautta tiimit työskentelevät tiiviimmin yhdessä, ja ovat valmiimpia innovoimaan ja reagoimaan muutoksiin. Ketterissä mene-

telmissä painotetaan tiimien pätevyyttä prosessien sijaan. (Cockburn & Highsmith, 2001.) Cockburnin ja Highsmithin (2001) artikkelissa mainitaan usean haastatellun ketterän menetelmän tiimin saaneen johtajiltaan syytöksiä vaatimusmäärittelyjen puutteista verrattaessa alkuperäiseen suunnitelmaan. Näissä tapauksissa ketterän kehityksen tiimit olivat vastanneet asiakkaan toiveisiin teknologisista alustan muutoksista, jotta asiakas saisi lisää arvoa ja tyydytystä. Tässä tapauksessa tiimien toimet katsottiin ristiriitaisiksi, mikä saattoi vähentää sitoutumisen halua tiimissä tulevaisuudessa.

Brhel kollegoineen (2015) esittää artikkelissaan, että selkeät vastuun jaot ovat Scrum-tiimissä tarpeen, mikä vaikuttaa myös tuoteomistajan tyytyväisyyteen. Artikkelissa nostetaan myös esille, että käytettävyyssiantuntijan tulee olla aktiivinen osa tiimiä. Käytettävyyssiantuntijalla tulee olla selvästi määritellyt vastuut ja auktoriteettiasema. Edellä mainitut toimenpiteet vaikuttavat ratkaisevasti tiimin sisäiseen kommunikaatioon, rohkeuteen, sitoutumiseen sekä tiimityön lopputulokseen.

Cockburnin ja Highsmithin (2001) tutkimuksessa huomattiin, että ketterät menetelmät pärjäsivät paremmin kuin ne menetelmät, jotka sitoivat työntekijän tiukkoihin raameihin moraalisesti. Tämä tarkoittaa sitä, että johtajatasolla tulee ymmärtää, että lopulta ei ole väliä, kuka tekee päätöksiä. Yhteistyö ja informaation jakaminen ovat tärkeämpää, jotta tärkeät päätökset saadaan tehtyä nopeasti. Tiimijäseniin ja tiimien väliseen yhteistyöhön tulee luottaa.

Scrumissa Scrum-mestari on vastuussa etujen maksimoimisesta ja kehitystyön toteutuksesta. Tässä tapauksessa Scrum-mestari huoltaa jäsenten välisiä suhteita, jotta menestys taataan. Tämä tarkoittaa sitä, että yksi henkilö toimii kommunikaation ja hyvien suhteiden luoja. Scrum-mestari edesauttaa informaation jakamista, mutta ei saa toimia esteenä kommunikaatiolle, jos informaation jakaminen toimii luontevasti tiimin jäseniltä asiakkaan suuntaan. Kuten edelläkin mainitaan, ihmiskeskeisyys ja nopea reagointikyky ovat avaintekijöitä menestyvässä ketterässä kehitysmenetelmässä. (Schwaber, 2004.)

3.3.2 Sitoutuminen Extreme Programming -kehitysmenetelmän näkökulmasta

Extreme Programming eli XP perustuu sosiaaliseen muutokseen. Se on vanhoista toimintatavoista luopumista ja omaan tuottavuuteen tutustumista, mikä voi tuntua aralta ja paljaalta. Hyvät suhteet johtavat hyvään liiketoimintaan. Tuottavuus ja itsevarmuus ovat yhteydessä ihmissuhteisiin työpaikalla yhtä paljon kuin työntekijä koodiinsa tai muihin työtehtäviin. XP-kehitysmenetelmä tarvitsee sekä taitavaa tekniikkaa että hyviä ihmissuhteita ollakseen menestynyt ohjelmistokehitysprojekti. (Beck & Andres, 2004.)

Kuten aiemmin tutkielmassa mainittiin, XP:n filosofia menestyvästä kehitysmenetelmästä sisältää kommunikaatiota, palautetta, yksinkertaisuutta, rohkeutta ja arvostusta. XP:n toimintatapoihin kuuluu myös aivoriivet, pariohjelmointi ja kertomuskortit. Kaikki edellä mainitut tekijät ovat inhimillisyyteen liittyviä. (Beck & Andres, 2004.) Työntekijöillä on henkilökohtaisia mieltymyk-

siä, mutta jokainen XP:n vaatimasta menestystekijästä on läheisessä suhteessa motivaattoritekijöihin (Sharp ym., 2009), jotka kuuluvat kukin affektiiviseen, instrumentaaliseen, normatiiviseen tai jatkuvaan sitoutumiseen.

Beck ja Anders (2004) korostavat, että XP-menetelmän menestys on riippuvainen aktiivisesti mukana olevien henkilöiden läheisestä yhteistyöstä. Menestys on riippuvainen myös käytänteistä, jotka toimivat sekä pitkäaikaisissa että lyhytaikaisissa ohjelmistokehitysprojekteissa tiimin jäsenten vuorovaikutuksessa.

Beck ja Anders (2004) mainitsevat julkaisussaan, että XP-malli oli hänen pyrkimyksensä sovittaa ihmiskuntaa yhteen. Hän pyrki tuomaan esille tuottavuuttaan omassa ohjelmistokehitysprojektissaan, ja jakaa muille käytännön kokemuksensa. Beck alkoi huomata, että mitä inhimillisemmin hän käsitteli itseään ja muita, sitä tuotteliaammiksi he kaikki tulivat. Menestyksen avain ei ole itsemääräämistä vaan hyväksymistä, että olemme ihmisiä liiketoimintamaailmassa, jossa tehdään kauppaa ihmiseltä toiselle. (Beck & Anders, 2004.)

3.3.3 Sitoutuminen dynaamisen ohjelmistokehitysmenetelmän näkökulmasta

Thelen ja Smith (1996) lähestyvät ihmisen kehitystä uskomalla, että henkinen elämä ja sen kehitys ovat jatkuvaa kaiken biologisen kasvun muodoissa ja toiminnassa. Tällä lähestymistavalla he jatkavat ketterien kehitysmenetelmienkin kanssa. Ihminen voi suoriutua lukemattomista aktiviteeteista: yksinkertaisista tai kompleksisista.

Perinteisen projektipäällikön ja DSDM-ohjelmistokehityksen projektipäällikön ero on se, että DSDM-projektipäällikkö keskittyy luomaan ja ylläpitämään positiivista suhdetta asiakkaisiin ja tiimien jäseniin (Cockburn & Highsmith, 2001). Kofman ja Senge (1993) esittävät, että vahvan sitoutumisen rakentamiseksi tiimien sisällä tulisi rakentaa yhteisöllistä sitoutumista. Yhteisöllisyyden rakentaminen aktiviteettien ja käytännön kokeilujen kautta avaisi ovia vuorovaikutukselle. Vuorovaikutus on suurimpia menestyksen avaimia ketterissä kehitysmenetelmissä. Ketterissä kehitysmenetelmissä, kuten DSDM:ssä, positiivinen ilmapiiri tuottaa tulosta ja tämä vaatii avoimuutta sekä luottamusta tiimijäsenten välillä. (Cockburn & Highsmith, 2001.)

DSDM-mallissa itsessään nostetaan esille affektiivisten sitoutumispiirteiden tärkeyttä. Työntekijää pyritään sitouttamaan työhönsä kulttuurin, työkavereiden ja asiakkaiden eli yhteisöllisyyden ja yhteenkuuluvuuden kautta. Sitoutumisen malli on tällöin psykologinen. DSDM-mallissa korostetaan yhteenkuuluvuuden tunnetta ja yhteisöllisyyttä, joita affektiiviseen sitoutumiseen kuuluu. Sitoutuminen on DSDM-mallissa erittäin tärkeää, mikä näkyy ketterän kehitysmenetelmän projektipäällikön roolissa ja tavoitteissa.

3.3.4 Sitoutuminen ketterässä kehittämisessä: Case Google

Rozowsky (2015) kertoo, kuinka Google aloitti kaksivuotiset tutkimukset tiimien menestymisestä. Tutkimuksessa selvitettiin miksi jotkut tiimit menestyvät hyvin ja toiset heikommin. Tutkimusta johti Googlen henkilöstöanalytiikan osaston johtaja Abeer Dubey kollegansa Julia Rozowskyn kanssa. Kahdessa vuodessa Dubeyn ja Julia Rozowskyn vetämä tiimi haastatteli 180 Googlen työtiimiä. Tutkimuksessa selvitettiin tiimien keskinäistä ja sisäistä toimintaa. Googlen ennakko-odotukset olivat väärät, kun he odottivat, että kunkin alan huippuosajista koostuneet tiimit olisivat menestyneimpiä. Tutkimuksen lopussa he löysivät työpsykologien ja sosiologien tutkimuksia ryhmadynamiikasta, kuten perinteisistä normeista sekä kirjoittamattomista säännöistä, jotka vaikuttavat siihen, miten hyvin ihmiset työskentelevät toistensa kanssa. (Rozowsky, 2015.) Rozowskyn (2015) arvion mukaan työskentelyyn vaikuttivat viisi olennaista tekijää:

- Luottamus: tiimin jäsenet saivat tehtyä työtehtävät ajoissa ja tavoitteiden mukaisesti
- Rakenne ja selkeys: tavoitteet ja roolit olivat kaikkien tiedossa
- Tarkoitus: työllä oli henkilökohtainen merkitys jokaiselle ryhmän jäsenelle
- Vaikuttavuus: jokainen jäsen uskoi työllään olevan vaikutusta päämäärään
- Psykologinen turvallisuus: tiimin jäsenet uskalsivat kommentoida, kysyä ja kyseenalaistaa toisiaan pelkäämättä tulevansa nolatuksi tai saavansa kokea muita seuraamuksia

Tutkimuksessa selvisi, että tiimi toimii menestyksekkäimmin, kun tiimin jäsenet voivat ottaa riskejä, luottaa toisiinsa, jokaisella on selkeä tavoite ja jokaisen tiimin jäsenen työ tuntuu merkittävältä.

Googlen tutkimus sai vahvistusta, kun johtamisen professorit Jasmine Hu ja Robert Liden julkaisivat tutkimuksen 67 eri tiimin toimintatavoista kuudessa eri yrityksessä. Päällimmäisenä nousivat sosiaalisten arvojen merkitys sekä tiimin jäsenten halu vaikuttaa toisten elämään positiivisesti. Hun ja Lidenin tulokset vain vahvistivat Googlen tutkimuksen tuloksia. (Dubey, 2015.)

Googlen tutkimustuloksissa korostettiin, että kaikista tärkein ja kriittisin työkalu viidestä avaintekijästä on psykologinen turvallisuus. Ihmiset ovat vastahakoisia sitoutumaan, jos on olemassa mahdollisuus, että henkilön käytös voi vaikuttaa negatiivisesti siihen, kuinka muut näkevät hänen pätevyytensä, tietoisuuden tai positiivisuuden. Kyseinen ajattelutapa on ihmiselle luontainen, mutta se saattaa olla haitallinen tiimityöskentelylle. Tärkeämpää psykologisessa turvallisuudessa on kuitenkin se, että sitoutuneet työntekijät epätodennäköisemmin jättävät työpaikkansa, he ovat valmiimpia omaksumaan monipuoliset ja eriävät ideat työkavereiltaan, he ovat tuotteliaampia ja heidät luokitellaan

paljon useammin tehokkaiksi työntekijöiksi muihin verrattaessa. (Rozovsky, 2015.)

3.3.5 Yhteenveto

Luvussa 3.1 käsiteltiin sitoutumisen määritelmät sekä sitoutumisen mallit. Luvussa nousi esille sitoutumisen määritelmän haasteellisuus, sillä siihen on monta eri näkökulmaa riippuen tutkimuksen tekijästä. Sitoutumisen malleihin pystyttiin kiteyttämään yleisimmät ja yhtenäiset sitoutumisen muodot eri tasoihin (affektiivinen, normatiivinen, jatkuva ja instrumentaalinen sitoutuminen).

Luvussa 3.2 käsiteltiin motivaattoritekijät sitoutumisen malleissa. Motivaatio kuuluu aina johonkin sitoutumisen malliin, mutta työntekijä voi olla sitoutunut ilman motivaatiotakin. Motivaattoritekijöistä suurin osa liittyy psykologiseen näkökulmaan eli affektiivisen sitoutumisen malliin.

Luvussa 3.3 tuotiin esille sitoutuminen ketterien kehitysmenetelmien näkökulmasta ja alaluvuissa syvennyttiin sitoutumiseen jokaisessa tutkielmaan valitussa ketterässä kehitysmenetelmässä (Scrum, XP ja DSDM).

Jokaisessa ketterässä kehityksen mallissa vuorovaikutustaidot, positiivinen ilmapiiri ja avoimuus nousivat esille vahvasti. Sitoutumisen prosessit voidaan nähdä jokaisessa eri sitoutumisen muodoissa.

Sitoutuminen on todettu jokaisessa ketterässä kehitysmenetelmässä kriittiseksi avaintekijäksi menestymisessä. Sitoutuminen tavoitteeseen tai tulokseen, psykologinen tunne yhteenkuuluvuudesta, positiivinen ilmapiiri ja yhteisöllisyys ovat niin tärkeitä menestyksen kulmakiviä, että ketteriin kehitysmalleihin on luotu erilliset roolit pitämään huolta juuri näistä menestyksen avaimista.

Case Googlessa havaitut tekijät toistavat tässä kirjallisuuskatsauksessa esille nousseita osa- tai avaintekijöitä menestyvään tiimityöhön ketterissä kehitysmenetelmissä. Yksilöllisellä sitoutumisella, oli se sitten mikä sitoutumisen muoto tahansa, on suuri merkitys ja vaikutus ryhmädynamiikkaan sekä ketterän kehitysmenetelmän onnistumiseen ja menestymiseen.

4 POHDINTA JA YHTEENVETO

Tämän kirjallisuuskatsauksen tulosten mukaan kaikkein johdonmukaisimpia suhteita on todettu olevan affektiivisen sitoutumisen mallin ja erilaisten suoritusindeksien välillä. Yksilön tunnistautumisen voima, yhteenkuuluvuus ja osallistuvuus tietyssä organisaatiossa on luonnehdittu kolmella tavalla. Voimakas usko organisaation tavoitteisiin ja arvoihin, valmius nähdä vaivaa organisaation ja muiden työntekijöiden puolesta sekä voimakas halu säilyä osana organisaation yhteisöä, ovat suurimpia tekijöitä affektiivisessä sitoutumisessa. Affektiivisellä sitoutumisella on todettu olevan suurin ero työntekijän työskentelyyn, työskentelyn tehokkuuteen ja onnistumiseen. Yritysjohtajien tulisikin kiinnittää erityistä huomiota affektiivisen sitoutumisen tunnistamiseen. Kuinka tällainen sitoutuminen kehittyy? Tunnistamisen avulla organisaation johto voi vahvistaa työntekijän sitoutumisen yritykseen, mikä todistetusti parantaa organisaation tehokkuutta ja tulosta. Affektiivinen sitoutuminen on suoraan sidoksissa yrityksen henkilöstökulttuuriin.

Ketteriä ohjelmistokehitysmenetelmiä luonnehditaan nopeudella, ketteryydellä, reaktiivisuudella ja herkkyydellä. Tällainen toimintatyyli vaatii joustavuutta, avoimuutta, kärsivällisyyttä, luottamusta sekä arvostusta tiimien ja tiimin jäsenten välillä. Oli käytettävä ketterä kehitysmenetelmä mikä tahansa, se vaatii myös paljon tiimien johtajilta. Vaikka päätöksentekoprosessit olisivatkin delegoituneet tiimeille, niin aina on yhden henkilön oltava vastuussa ohjelmistokehityksen edistymisestä. Tämä vaatii johtajilta henkilöstöjohtamisen ja osaamisen johtamisen taitoja. Ihmisten johtamiseen ja työntekijöiden suhteisiin sekä asiakassuhteisiin on panostettu jo itse ketterien kehitysmenetelmien mallien luomisessa, sillä ihmisten suhteet ja sitoutuminen ovat vuosien varrella osoitettu vaikuttavan erittäin voimakkaasti yrityksen menestymiseen. Kuten DSDM:n, Scrumin tai XP:n ketterissä kehitysmenetelmissä, on ketterän tiimin johtajilla rooli luoda hyviä suhteita ja ylläpitää positiivista ilmapiiriä.

Tutkimuksia liiketoiminnan prosesseista ja protokollista sitoutumisen pohjalta on tehty viime vuosina jo enemmän. Liiketoiminnan prosesseilla ja protokollilla, jotka pohjautuvat sitoutumisen muotoihin, tarkoitetaan sopimukseen perustuvia suhdemuotoja sekä analyysia, joilla löydetään oikeat toimenpi-

teet tavoitteiden saavuttamiseksi. Sitoutumisen muotoja ja niitä tukevia liiketoiminnan prosesseja tutkitaan ja tulisi tutkia yhä enemmän, sillä näiden suuri merkitys organisaation menestymiseen on viimeinkin huomattu ja noteerattu.

Vasta aivan viime vuosina on tutkimuksissa löydetty selvästi tehokkaan, sitoutuneen ja menestyneen tiimityöskentelyn avaintekijät. Tuloksista voidaan siis päätellä, että kyseiset kriittiset menestystekijät pohjaavat nimenomaan affektiiviseen sitoutumisen malliin, joita ovat:

- luottamus
- vaikuttavuus
- tarkoitus
- rakenne ja selkeys
- psykologinen turvallisuus

Edellä mainitut menestystekijät ovat avaintekijöitä ketterän ohjelmistokehitysmallin menestyksessä. Kyseisillä avaintekijöillä on suurin positiivinen vaikutus tiimityöskentelyyn, mikä taas näyttelee erittäin suurta roolia jokaisessa ketterässä kehitysmenetyksessä. Affektiivisen sitoutumisen mallin avaintekijöistäkin on löydetty syvin ja vahvin sitoutumisen muoto, joka on psykologinen turvallisuus. Aiheesta on tehty vasta muutamia tieteellisiä tutkimuksia, joten tulevaisuudessa psykologista turvallisuutta ja sen vaikutusta ketteriin kehitysmenelmiin tulisi tutkia lisää.

Tässä kirjallisuuskatsauksessa tarkasteltiin perinteisiä ohjelmistokehityksen malleja, jotta ohjelmistokehityksen mallien muuttumista nykyaikaa vastaaviksi voitaisiin ymmärtää. Tutkielmassa tarkasteltiin ketteriä kehitysmenetelmiä, joista valittiin vain kolme yleisintä mallia (Scrum, Extreme Programming sekä dynaaminen ohjelmistokehitysmenelmä) tutkimuksen luonteen ja resurssien puutteen vuoksi. Kirjallisuuskatsauksessa tutkittiin myös työntekijän sitoutumista, sitoutumisen malleja sekä sitoutumisen malleja valittujen ketterien kehitysmenetelmien näkökulmasta.

Tutkimusongelma tiivistettiin seuraavasti:

- Mitkä tekijät vaikuttavat työntekijän sitoutumiseen ketterillä menetelmillä tehdyssä ohjelmistokehityksessä?

Tutkielman toteutus perustui johdannossa esiteltyyn systemaattisen kirjallisuuskatsauksen malliin. Tutkimusongelman asettamisen ja tutkimuksen suunnittelun jälkeen aloitettiin lähdemateriaalin koostaminen. Lähdemateriaalit koottiin systemaattisen kirjallisuuskatsauksen protokollan mukaisesti eri hakukoneiden ja avainsanojen avulla. Tuoreen lähdemateriaalin löytäminen oli haastavaa, sillä ketteriä kehitysmenetelmiä ei ollut tutkittu viimeisiin vuosiin kovin aktiivisesti kirjallisuuskatsausta aloitettaessa. Omat haasteensa toivat myös lähdemateriaalien vähyys työntekijän sitoutumisesta ketterien kehitysmenetelmien näkökulmasta. Suoraan kyseisestä näkökulmasta tehtyjä tutkimuksia ja laadukkaita lähteitä oli vain kourallinen. Kuitenkin sitoutumista yle-

sesti ja työntekijän sitoutumista työhön sekä organisaatioon on tutkittu paljon, jolloin laadukasta lähdemateriaalia kuitenkin aiheeseen liittyen löytyi hyvin.

Tutkielman aineisto koostuu ohjelmistokehityksen perinteisistä malleista, ketteristä kehitysmenetelmistä, sitoutumisen muodoista ja työntekijän sitoutumisesta omaan työhön käsittelevistä tieteellisistä julkaisuista ja kirjoista. Tutkielman tulokset pohjautuvat kirjallisuuskatsauksen lähdemateriaaliin sekä näistä tehtyihin johtopäätöksiin.

Tämän kirjallisuuskatsauksen tavoitteena oli löytää yleisimpien käytettyjen ketterien kehitysmenetelmien yhteiset piirteet tiimien toiminnan näkökannalta ja esittää tärkeimmät sitoutumisen mallit. Kirjallisuuskatsauksen tavoitteen pääpiste oli osoittaa toimivin sitoutumisen malli sekä sitoutumisen merkitys tiimityöskentelyssä ja ketterissä kehitysmenetelmissä. Saadut tulokset olivat odotettuja ja tutkimusprosessi täytti sille asetetut tavoitteet.

Kirjallisuuskatsauksessa ei kuitenkaan aiheajauksen vuoksi tutkittu syvimmän sitoutumisen mallin (affektiivinen sitoutumisen muoto) ja ketterien kehitysmenetelmien liiketoimintaprosesseja ja protokollia, jotka voisivat tukea ja edesauttaa niin työntekijöiden sitoutumista kuin tiimityöskentelyn menestymistä. Tutkielmassa ei myöskään tutkittu kaikkia yleisiä ketteriä kehitysmalleja eikä totuudenmukaista organisaatioiden ketterien kehitysmenetelmien käyttöönottoa ja hyödyntämistä. Tämän vuoksi jatkotutkimukset ovat tarpeen. Sitoutumisen mallit kaikissa yleisimmissä ketterissä ohjelmistokehitysmenetelmissä tulisi tutkia laajemmin ja syvällisemmin. Tänä päivänä yritysten toimintamallit perustuvat yhä enemmän ketteryyteen ja itseohjautuvaan tiimityöskentelyyn. Tutkimukset ovat tarpeen myös johtoryhmän näkökulmasta: kuinka sitoutuminen tunnistetaan, otetaan huomioon taktisissa operatiivisissa päätöksissä ja kuinka sitä vahvistetaan ketterissä kehitysmenetelmissä.

LÄHTEET

- Abrahamsson, P. (2002). *The Role of Commitment in Software Process Improvement*. Academic Dissertation to be presented with the assent of the Faculty of Science for public discussion in Kajaaninsali, Oulu, June 14-14, (s. 29–150).
- Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. (2002). *Agile Software Development Methods: Review and Analysis*. Espoo: VTT Publications 478.
- Abrahamsson, P., Warsta, J., Siponen, M. & Ronkainen, J. (2003). New Directions on Agile Methods: A Comparative Analysis. *Proceedings of the 25th International Conference on Software Engineering*, Portland, OR, USA, May 3-10, 2003. IEEE.
- Bagozzi, R. & Bergami, M. (2000). Self-categorization, Affective Commitment and Group Self-esteem as Distinct Aspects of Social Identity in the Organization. *The British Journal of Social Psychology*, 39(4), 555-577.
- Beck, K. & Wilson, C. (2000). Development of Affective Organizational Commitment: A Cross- Sequential Examination of Change with Tenure. *Journal of Vocational Behavior*, 56, 114-136.
- Beck, K. & Andres, C. (2004). *Extreme Programming Explained: Embrace Change*. (2. uud. painos). Boston: Addison-Wesley Professional.
- Becker, H. (1960). Notes on the concept of commitment. *American Journal of Sociology*, 66, 32-40.
- Becker, T. (1992). Foci and Bases of Commitment: Are They Distinctions Worth Making? *Academy of Management Journal*, 35, 232-244.
- Beecham, S., Baddoo, N., Hall, T., Robinson, H. & Sharp, H. (2008). Motivation in Software Engineering: A Systematic Literature Review. *Information and Software Technology*, 50(9-10), 860-878.
- Boehm, B. (1988). A Spiral Model of Software Development and Enhancement. *IEEE Computer*, 21(5), 61-72.
- Boehm, B. & Papaccio, P. (1988). Understanding and Controlling Software Costs. *IEEE Transactions on Software Engineering*, 14(10), 1462-1477.
- Brereton, P., Kitchenham, B., Budgen, D., Turner. & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *The Journal of Systems and Software*, 80(4), 571-583.
- Brhel, M., Maedche, A., Meth, H. & Werder, K. (2015). Exploring principles of user-centered agile software development: A literature review. *The Journal of Information and Software Technology* 61, 163-181.
- Brown, R. (1996). Organizational Commitment: Clarifying the Concept and Simplifying the Existing Construct Typology. *Journal of Vocational Behavior* 49, 230-251.
- Cockburn, A. & Highsmith, J. (2001). Agile Software Development, the People Factor. *IEEE Computer*, 36(11), 131-133.

- 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.
- Ghezzi, C., Jazayeri, M. & Mandrioli, D. (2002). *Fundamentals of Software Engineering*. New Jersey: Prentice Hall.
- Goldman, S., Nagel, R. & Preiss, K. (1995). *Agile Competitors and Virtual Organizations: Strategies for Enriching the Customer*. New York: A division of International Thomson Publishing Inc.
- Dubey, A. (2015, 12, marraskuuta). Google Searches Itself to Build More Productive Teams. Haettu 19.4.2020 osoitteesta <https://www.voanews.com/silicon-valley-technology/google-searches-itself-build-more-productive-teams>
- Hartmann, L. (2000). Organizational commitment: A multi method scale analysis and test of effects. *The International Journal of Organizational Analysis* 8(1): 89-108.
- Highsmith, J. (2002). *Agile Software Development Ecosystems*. Boston: Pearson Education, Inc.
- Highsmith, J. & Cockburn, A. (2001). Agile Software Development: The Business of Innovation. *IEEE Computer* 34(9), 120-127.
- Jaros, S. Jermier, J., Koehler, J. & Sincich, T. (1993). Effects of Continuance, Affective, and Moral Commitment on the Withdrawal Process: An Evaluation of Eight Structural Equation Models. *Academy of Management Journal* 36(5): 951-955.
- Kelman, H. (1958). Compliance, Identification and Internalization: Three Processes of Attitude Change. *Journal of Conflict Resolution* 2: 51-60.
- Kiesler, C. (1971). *The Psychology of Commitment: Experiments Linking Behavior to Belief*. New York: Academic Press.
- Kofman, F. & Senge, P. (1993). Communities of Commitment: The Heart of Learning Organizations. *Organizational Dynamics*, 22(2), 5-23.
- Mahnic, V. & Drnovscek, S. (2005, January). *Agile Software Project Management with Scrum*. Haettu 19.4.2020 osoitteesta https://www.researchgate.net/publication/228967959_Agile_Software_Project_Management_with_Scrum
- Matharu, G., Mishra, A., Singh, H. & Upadhyay, P. (2015). Empirical Study of Agile Software Development Methodologies: A Comparative Analysis. *ACM SIGSOFT Software Engineering Notes*, 40(1).
- Meyer, J. & Allen, N. (1991). A Three-Component Conceptualization of Organizational Commitment. *Human Resource Management Review*, 1(1), 61-89.
- Meyer, J. & Allen, N. (1997). *Commitment in the Workplace: Theory, Research, and Application*. Thousand Oaks: Sage Publications.
- Meyer, J. & Allen, N. (2004). *TCM Employee Commitment Survey: Academic User Guide 2004*. Canada.
- Moe, N., Dingsoyr, T. & Dybå, T. (2008). Understanding Self-Organizing Teams in Agile Software Development. *Proceedings of the 19th IEEE Australian*

- Conference on Software Engineering*. Perth, Australia, March 26-28, 2008. IEEE.
- Paetsch, F., Eberlein, A. & Maurer, F. (2003). Requirement engineering and agile software development. *Proceedings of the 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*. Linz, Austria, June 11-11, 2003. IEEE.
- Pressman, R. & Maxim, B. (2014). *Software Engineering: A Practitioner's Approach*. (8. uud. painos). New York: McGraw-Hill Higher Education.
- Rozovsky, J. (2015, 17 marraskuuta). *The Five Keys to a Successful Google Team*. Haettu 19.4.2020 osoitteesta <https://rework.withgoogle.com/blog/five-keys-to-a-successful-google-team/>
- Schwaber, K. (2004). *Agile Project Management with Scrum*. Washington: Microsoft Press.
- Sharp, H., Baddoo, N., Beecham, S., Hall, T. & Robinson, H. (2009). Models of motivation in software engineering. *Information and Software Technology*, 51(1), 219-233.
- Shore, L. & Martin, H. (1989). Job Satisfaction and Organizational Commitment in Relation to Work Performance and Turnover Intentions. *Human Relations* 42(7), 625-638.
- Thelen, E. & Smith, L. (1996). *A Dynamic Systems Approach to the Development of Cognition and Action*. London: A Bradford book.
- Voigt, B. (2004, tammikuu). *Dynamic System Development Method*. Haettu 19.4.2020 osoitteesta <https://tinyurl.com/yb88bldl>
- Williams, L. & Cockburn, A. (2003). Guest Editors' Introduction: Agile Software Development: It's about Feedback and Change. *IEEE Computer Society*, 36, 39-43.