



OULUN YLIOPISTO
UNIVERSITY of OULU

TEKNILLINEN TIEDEKUNTA

**Hyvät käytänteet ja haasteet kommunikoinnissa tiimin
jäsenten välillä ketterässä ohjelmistoprojektissa**

Joonas Huttu

TUOTANTOTALOUDEN KOULUTUSOHJELMA

Kandidaatintyö

Elokuu 2017



OULUN YLIOPISTO
UNIVERSITY of OULU

TEKNILLINEN TIEDEKUNTA

**Hyvät käytänteet ja haasteet kommunikoinnissa tiimin
jäsenten välillä ketterässä ohjelmistoprojektissa**

Joonas Huttu

Ohjaaja(t): K.A.

TUOTANTOTALOUDEN KOULUTUSOHJELMA

Kandidaatintyö

Elokuu 2017

TIIVISTELMÄ

OPINNÄYTETYÖSTÄ Oulun yliopisto Teknillinen tiedekunta

Koulutusohjelma (kandidaatintyö, diplomityö) Tuotantotalouden koulutusohjelma		Pääaineopinnojen ala (lisensiaatintyö)	
Tekijä Huttu, Joonas		Työn ohjaaja yliopistolla Aaltonen K, apulaisprofessori	
Työn nimi Hyvät käytänteet ja haasteet kommunikoinnissa tiimin jäsenten välillä ketterässä ohjelmistoprojektissa			
Opintosuunta	Työn laji Kandidaatintyö	Aika Elokuu 2017	Sivumäärä 30 s.
Tiivistelmä			
<p>Kandidaatintyön ensisijaisena tavoitteena oli saada vastaus tutkimuskysymykseen, millaisia haasteita ja hyviä käytäntöjä voidaan tunnistaa tiimin sisäisessä kommunikoinnissa ketterässä ohjelmistoprojektissa. Tällä työllä oli myös tarkoitus tuoda yhteen ketterien menetelmien ja kommunikoinnin eri teorioita, käytäntöjä ja tutkimustuloksia. Lisäksi tavoitteena oli saada itselleni syvempää ymmärrystä ketterien menetelmien toiminnasta ja kommunikoinnin teoriasta.</p> <p>Työ tehtiin kuvailevana kirjallisuuskatsauksena mahdollisimman laajan kuvan aikaansaamiseksi. Aineistoksi kirjallisuuskatsausta varten pyrittiin ensisijaisesti löytämään tieteellisiä artikkeleita ketteristä menetelmistä ja kommunikoinnista. Tieteellisten artikkeleiden lisäksi ketteristä menetelmistä käytettiin lähteinä Agile Manifestoa ja sen kirjoittajien kirjallisuutta ja teoriaa ketteristä menetelmistä. Työssä käsitellään kommunikoinnin teoriaa ja ketteriä menetelmiä yhdessä ja erikseen. Näiden perusteella muodostetaan johtopäätökset tutkimuskysymykseen vastaamista varten.</p> <p>Kommunikoinnin hyviksi käytänteiksi löytyi avoimuus ja palautteen antaminen, kasvokkain kommunikointi, tarinalaput ja taulu, työskentely mahdollisimman lähekkäin, aktiivinen kuuntelu ja avoimet kysymykset, verbaalisen ja non-verbaalisen kommunikoinnin pitäminen linjassa keskenään, oikeiden ihmisten valitseminen tiimiin sekä päivittäiset tapaamiset ja uuden tiedon jakaminen. Merkittävimmiksi haasteiksi kommunikoinnille löytyi virtuaalitiimi ja huono toimiston rakenne, riittävä informaation jakaminen, aiemmat tottumukset itsenäisestä työskentelystä, häiriöt kommunikointikanavassa ja henkilökohtaiset kommunikoinnin suodattimet, faktat, tunteet ja mielipiteet, tiimin jäsenten vaihtuvuus sekä kommunikoinnin hukka. Tulokset ovat pääosin hyödynnettävissä käytäntöön sellaisenaan. Merkittävin hyöty tuloksista voi olla ohjelmistokehityksen alalla, jossa kommunikointi tiimin sisällä on merkittävässä roolissa lopputuotteen kannalta. Kuitenkin aiheesta olisi vielä tarvetta tehdä lisää tieteellistä tutkimusta.</p>			
Muita tietoja			

ABSTRACT FOR THESIS

University of Oulu Faculty of Technology

Degree Programme (Bachelor's Thesis, Master's Thesis) Industrial Engineering and Management		Major Subject (Licentiate Thesis)	
Author Huttu, Joonas		Thesis Supervisor Aaltonen K, assistant professor	
Title of Thesis Good practices and challenges in communication within the team in an agile software project			
Major Subject	Type of Thesis Bachelor's Thesis	Submission Date August 2017	Number of Pages 30 p.
Abstract <p>The main goal for this bachelor's thesis was to get an answer to the question: What kind of challenges and good practices can be identified in communication within the team in an agile software project? This thesis was also supposed to bring together different theories, practices and research results of agile methods and communication. In addition, one goal was to get myself a deeper understanding of the functioning of agile methods and the theory of communication.</p> <p>The thesis was carried out as a descriptive literature review to gain as much perspective about the subject as possible. Primarily for the literature review was chosen scientific articles about agile methods and communication. Besides scientific articles the Agile Manifesto and the theories and literature of the Manifesto's writers were used as sources about agile methods. In the thesis, the theory of communication and agile methods are dealt with first separately and then together. Based on these theories are done the conclusions to answer the research question.</p> <p>Good practices in communication were found openness, giving feedback, communicating face-to-face, story cards and the wall for them, working as close to each other as possible, active listening, open-ended questions, keeping the verbal and non-verbal communication in line with each other, choosing the right people to the team, daily meetings and sharing new information. The most significant challenges that were found for communication are working in a virtual team, bad structure of the office, sharing enough information, previous independent working habits, interference in the communication channel, personal communication filters, one's facts, feelings and opinions, turnover of members in the team and loss in communication. The results of this thesis are mostly exploitable to practice. The most significant benefit of the results is in the field of software development, where communication within the team plays a major role in terms of the end product. However there still should be carried out more scientific research about the subject.</p>			
Additional Information			

SISÄLLYSLUETTELO

SISÄLLYSLUETTELO.....	5
1 JOHDANTO.....	5
2 KETTERÄ OHJELMISTOPROJEKTI	6
2.1 Ketterä kehitys yleisesti.....	7
2.2 Ketterän ohjelmistoprojektin eri muotoja.....	12
2.3 Ketterän ohjelmistoprojektin tiimin jäsenet	15
3 KOMMUNIKOINTI KETTERÄSSÄ OHJELMISTOPROJEKTISSA.....	18
3.1 Kommunikoinnin perusteet tiimissä	18
3.2 Käytännön haasteet tiimin sisäisessä kommunikoinnissa ketterissä projekteissa	19
3.3 Hyvät käytänteet tiimin sisäisessä kommunikoinnissa ketterissä projekteissa.....	21
4 POHDINTA.....	23
5 YHTEENVETO.....	25
Lähdeluettelo	26

1 JOHDANTO

Ohjelmistoprojektit eroavat huomattavasti perinteisistä, lineaarisesti etenevistä projekteista. Suurin ero syntyy itse projektin lopputuloksesta, sillä projektin alussa suunniteltu lopputulos voi olla hyvinkin erilainen kuin valmis ohjelmisto projektin päätyttyä. Tämä johtuu pääasiassa vaatimusmäärittelyjen muuttumisesta projektin edetessä. Tällöin perinteiset projektimallit voivat olla suurissa vaikeuksissa, kun ei voida edetä suunnitelman mukaisesti. Tämä on ollut yksi merkittävimmistä syistä ketterien menetelmien syntymiseen. Ketterät menetelmät eroavat huomattavasti perinteisistä projektimalleista sopeutumiskykynsä vuoksi. Tällöin kaikilla tiimin jäsenillä tulee olla koko ajan ajantasainen tieto projektin tilasta ja uusimmista vaatimuksista ohjelmiston suhteen. Näin ollen luonnollisesti myös kommunikointi, eli kahden tai useamman ihmisen välinen informaation vaihtaminen, eroaa merkittävästi perinteisiin projekteihin nähden.

Tässä kandidaatintyössä käsittelen kommunikoinnin haasteita ja hyviä käytäntöjä tiimin sisällä ketterässä ohjelmistoprojektissa. Työ on tehty kuvailevana kirjallisuuskatsauksena mahdollisimman laajan näkökulman saamiseksi. Tutkimuskysymyksenäni on: Millaisia haasteita ja hyviä käytäntöjä voidaan tunnistaa tiimin sisäisessä kommunikoinnissa ketterässä ohjelmistoprojektissa?

Aluksi käyn läpi ketterien menetelmien toimintatapoja ja periaatteita, joiden avulla luon viitekehyksen toimintaympäristölle. Tämän jälkeen käyn hieman läpi kommunikoinnin teoriaa, josta jatkan kirjallisuudesta paljastuneisiin haasteisiin ja hyviin käytänteisiin kommunikoinnissa ketterässä ohjelmistoprojektissa. Lopuksi pohdin kirjallisuudesta selvinnyttä lopputulosta ja millaisia jatkotutkimuskohteita aiheesta olisi.

2 KETTERÄ OHJELMISTOPROJEKTI

Projekteja on ollut olemassa jo vuosisatojen ajan, mutta ohjelmistoprojektit ovat suhteellisen uusi projektien muoto. Kuitenkin jo varhaisessa vaiheessa huomattiin, ettei ohjelmistoprojektien hallinta onnistunut aiemmin hyviksi todetuilla projektinhallintateorioilla. (Cline et al. 2015, s. 3) Yksi merkittävä syy tähän epäonnistumiseen oli teorioiden pohjautuminen oletukseen, että projektien vaatimusmäärittelyt voitaisiin tietää täysin jo projektin alkuvaiheilla. Kuitenkin ohjelmistoprojektit ovat ennalta-arvaamattomampia, sillä todellisuudessa suurin osa vaatimusmäärittelyistä ja muutoksista selviää vasta projektin aikana. (Abbas et al. 2008)

Ohjelmistoprojektilla tarkoitetaan kaikkia niitä työvaiheita, joiden päämääränä on toteuttaa projektisopimuksen mukaiset ehdot. Lisäksi ohjelmistoprojektilla on selkeä aikataulu, päämäärä, rajoitteet ja budjetti. Ohjelmistoprojekti voi olla itsenäisen ohjelmiston kehittäminen asiakkaalle tai osa suurempaa projektia. (IEEE standard for software project management plans, 1988) Perinteisesti ohjelmistoprojektit toimivat tayloristisen tavan mukaisesti, jossa suositaan roolien jakamista tiimin jäsenille ja näin ollen työn jakamista tiimin sisällä (Chau & Maurer, 2004). Ohjelmistoprojektille voidaan nähdä myös kaksi erillistä tavoitetta. Ensimmäinen on saada aikaan toimiva ohjelmisto vaatimusten mukaisesti. Toisena tavoitteena on puolestaan valmistautua projektilla tulevia tilanteita varten. (Cockburn, 2006, s. 219)

Ensimmäisten vuosikymmenien aikana ohjelmistokehitys muuttui jatkuvasti byrokraattisemmaksi, hitaammaksi ja kalliimmaksi. Tämä johtui muodollisten menetelmien suosimisesta, jolloin kehitettiin paljon erilaisia apuvälineitä kuten malleja, metodeja, ohjelmointikieliä ja tekniikoita. Muodollisten menetelmien suosimisen taustalla oli oletus, että ohjelmistokehitys olisi tekninen aikaansaannos, kuten muut tekniikan alat. (Appelo, 2011, s. 19) Lopulta kuitenkin nämä menetelmät eivät olleet riittäviä, minkä seurauksena ohjelmistotuotanto kehittyi merkittävästi. Tässä oli osaltaan vaikuttamassa ketterien menetelmien kehittäminen. (Cline et al., 2015, s. 3) Osasyynä tähän kehitykseen on myös ollut talouden kehittyminen tietoyhteiskunnan aikana. Tietoyhteiskunnan kehityksen seurauksena on yrityksille syntynyt tarve nopeaan reagointiin muutokseen ja muutoksen luomiseen kovassa kilpailussa menestymiseksi.

Lisäksi ohjelmistoja alettiin hyödyntää laajasti uusien tuotteiden kehittämisessä ja ohjelmistoista tuli merkittävä osa yritysten liiketoimintastrategiaa. (Highsmith, 2002, s. 3)

Ajatukset ketteristä menetelmistä oli esitetty jo ensimmäisen kerran Japanissa 1980-luvulla lean-ajattelun parissa, jossa yhtenä pääperiaatteena on hukan vähentäminen (Highsmith, 2004, s. 34). Varsinaisesti ketterien menetelmien periaatteet koottiin yhteen kuitenkin vasta Agile Manifestossa vuonna 2001. Yksi merkittävimmistä syistä Manifeston tekemiseksi oli kehittää sellaisia ohjelmistotuotannon käytäntöjä, jotka auttaisivat tuottamaan mahdollisimman hyviä ohjelmistoja pitämällä ihmisiä tärkeimpänä tekijänä ohjelmistotuotannossa. Toisena merkittävänä tekijänä Manifeston tekemisessä oli raskaan dokumentoinnin merkittävä vähentäminen ja mahdollisimman kevyesti toimiminen. (Highsmith, 2001)

2.1 Ketterä kehitys yleisesti

Ohjelmistokehitys on ketterää kehittämistä, kun se on inkrementaalista, suoraviivaista ja mukautuvaa sekä kehittäjät ja asiakas toimivat tiiviissä yhteistyössä (Abrahamsson et al., 2002, s. 98). Highsmith (2004, s. 16) puolestaan määrittelee ketteryyden kahdessa osassa: kykyinä luoda muutosta ja vastata muutokseen tavoitellen voiton tekemistä myrskyisässä liiketoimintaympäristössä samalla tasapainoillen joustavuuden ja vakauden välillä. Cockburnin (2006, s. 222) mukaan ketteryys on tehokasta ja helposti käsiteltävissä olevaa. Ketterä prosessi on hänen mukaansa puolestaan sekä kevyt että riittävä. Riittävyys takaa helpon käsiteltävyyden ja keveys tehokkuuden. Vaikka isommalla tiimillä on hankalampi seurata ketterien menetelmien periaatteita kuin pienellä tiimillä, niin tiimin tulisi silti pyrkiä mahdollisimman lähelle ketterien menetelmien ideaalitalanteita. Tällaisia ideaalisia tilanteita ovat kahdesta kahdeksaan henkilöä työskentelemässä samassa huoneessa, paikan päällä olevat käytettävyydsiantuntijat, kuukausittaiset inkrementit, täysin automatisoidut testit ja kokeneet kehittäjät. Kuukausittaisilla inkrementeillä tarkoitetaan kuukauden mittaisia ajanjaksoja, jonka aikana tehdään uusi osa ohjelmistoa, joka lisätään osaksi jo aiemmin tehtyä ohjelmistoa. Näin ohjelmiston aiemmin tehdyt osat tulisivat säilyä koko ajan toimintakykyisenä. (Cockburn, 2006, s. 222-224)

Agile Manifestossa määritellään ketterälle kehittämiselle neljä pääsääntöä. Ketterissä menetelmissä arvostetaan yksilöitä ja vuorovaikutusta enemmän kuin prosesseja ja työkaluja, toimivaa ohjelmistoa enemmän kuin kattavaa dokumentointia, yhteistyötä asiakkaan kanssa enemmän kuin sopimusneuvotteluja ja muutokseen vastaamista enemmän kuin tietyn suunnitelman seuraamista. Vaikka prosesseja ja työkaluja, dokumentointia, sopimusneuvotteluja sekä suunnitelman seuraamista arvostetaan vähemmän kuin perinteisissä menetelmissä, niin ne ovat kuitenkin arvokkaita osia projektissa. (Beck et al., 2001a)

Ketterä kehitys on ohjelmistokehitystapa nopeaan ohjelmiston kehittämiseen ympäristössä, jossa vaatimukset muuttuvat nopeasti. Tätä ketteryyttä varten tulee kehityksessä käyttää sellaisia toimintaperiaatteita, joilla saavutetaan riittävä kuri tiimin työskentelyssä ja jotka tarjoavat riittävän palautteen toiminnasta. Lisäksi ohjelmiston suunnittelussa tulee käyttää periaatteita, joilla ohjelmisto olisi mahdollisimman joustava ja ylläpidettävissä oleva. Toimivaan ketterään kehitykseen tulee lisäksi tietää ne suunnittelumallit, joiden avulla pystytään saavuttamaan edellä kuvattu tasapaino joustavuuden ja ylläpidettävyyden suhteen. (Martin, 2003, s. iv)

Ketterässä kehittämisessä kaikkia ohjelmistokehittämisen toimintoja tehdään koko projektin ajan sen sijaan että tietyt toiminnot alkavat vasta tietyssä vaiheessa projektia. Esimerkiksi palautetta annetaan jatkuvasti ja kehittäminen on jatkuvaa. Virheet korjataan saman tien sellaisen ilmettyä eikä odoteta tiettyyn vaiheeseen pääsemistä ensin. Näin toimitaan, koska ajan myötä virhe voi kasvaa suuremmaksi ja sen korjaaminen myöhemmin voi olla merkittävästi työläämpää ja haastavampaa. (Subramanian & Hunt, 2006, s. 3)

Ketteriä menetelmiä yhdistävänä Agile Manifesto pohjautuu seuraaviin kahteentoista periaatteeseen (Beck et al., 2001b):

1. Tärkein asia on asiakkaan tyydyttäminen toimittamalla aikaisessa vaiheessa ja jatkuvasti arvokasta ohjelmistoa.
2. Toivotetaan muutokset tervetulleiksi jopa kehityksen loppupuolella sekä hyödynnetään muutosta asiakkaan kilpailueduksi ketterissä prosesseissa.

3. Toimitetaan toimivaa ohjelmistoa säännöllisesti muutamasta viikosta muutaman kuukauden välein. Mieluiten ohjelmiston toimittaminen tapahtuisi mahdollisimman lyhyin aikavälein.
4. Kehittäjien ja liiketoimintaihminen tulee koko projektin ajan toimia päivittäisessä yhteistyössä.
5. Projektien tulee rakentua motivoituneiden yksilöiden ympärille, joihin luotetaan sekä tarjota heille tarvittava ympäristö ja tuki.
6. Tehokkain tapa kommunikoida tiimille ja tiimin sisällä on kasvokkain tapahtuva kommunikointi.
7. Etenemisen ensisijainen mittari on toimiva ohjelmisto.
8. Ketterät prosessit edistävät kestäväää kehittämistä. Sponsoreiden, käyttäjien ja kehittäjien tulisi pystyä pitämään tasainen tahti loputtomasti.
9. Ketteryyttä edistää jatkuva huomio tekniseen erinomaisuuteen.
10. Tekemättömän työn maksimoimiseen eli yksinkertaisuuteen tulee pyrkiä.
11. Itseorganisoituvat tiimit tuottavat parhaat arkkitehtuurit, suunnittelun ja vaatimukset.
12. Tiimin tulee säännöllisin väliajoin reflektoida omaa toimintaansa, miten parantaa työskentelynsä tehokkuutta ja toimia syntyneiden parannusehdotusten mukaisesti.

Nämä periaatteet syntyivät yhdessä itse Agile Manifeston kanssa, sillä tuolloin suurena ongelmana oli, etteivät ohjelmistonkehitystiimit kyenneet vastaamaan muutoksiin ja toimimaan nopeasti jatkuvasti kasvavien prosessien vuoksi (Martin, 2003, s. 4). Agile Manifeston syntymisen taustalla oli myös tarve saada ohjelmistokehitykseen metodologia, joka keskittyisi vain olennaisiin asioihin ohjelmistokehityksessä. Lisäksi ohjelmistokehitysprosessit olivat raskaita, mutta saavuttivat vähäisiä tuloksia. Tiimin jäsenillä tulee ymmärtää myös itse ketterä metodologia ja syyt sen taustalla pelkän ohjelmiston ymmärtämisen lisäksi. Vasta tällöin ketterien menetelmien käyttäminen onnistuu kunnolla ja muutokset saadaan tehtyä tehokkaasti. (Subramaniam & Hunt, 2006, s. 2, 17)

Vaikka Agile Manifeston (2001) mukaan dokumentointi tulisi pitää minimissään, sitä tulisi kuitenkin tehdä riittävästi ja riittävän ajoissa, jotta myöhemmin tarvittava tieto ei häviä heti projektin jälkeen jonkin henkilön mukana. Riittävä dokumentointi määräytyy

sen mukaan, minkä verran dokumentoinnin saaja tarvitsee tietoa pystyäksään suorittamaan seuraavan tehtävänsä. Kaikki tämän määrän ylittävä dokumentointi on turhaa. Dokumentointia tehdessä kannattaa käyttää keveintä ja laiskinta mahdollista menetelmää. Lisäksi dokumentointi voi olla vajavaista, kunhan ei dokumentoi liikaa. Paras dokumentaatio tehdystä työstä on kuitenkin itse ohjelma. (Cockburn, 2006, s. 219-221)

Myös Martin (2003, s. 5) näkee riittävän dokumentoinnin olevan välttämätöntä ohjelmiston onnistumiseksi. Hänen mukaan dokumentoinnissa tulisi keskittyä logiikkaan ja vain korkean tason rakenteisiin. Kuitenkaan dokumentointia ei tulisi tehdä kuin vain välittömään ja merkittävään tarpeeseen. (Martin, 2003, s. 5) Dokumentoinnin lisäksi tiedon häviämistä voidaan estää antamalla oma koodi säännöllisin väliajoin kollegojen tarkastettavaksi. Tällöin tiimistä vähintään yksi toinen kehittäjä käy koodin läpi, jotta koodi olisi myös ulkopuolisen ymmärrettävissä. Toinen tapa varmistaa useamman henkilön ymmärrys koodista on järjestää yksikkötestejä. Yksikkötestien avulla koodia saadaan jaettua pienempiin ja paremmin hallittaviin osiin, jolloin yksittäisten koodin osien merkitys tulee paremmin esille. (Subramaniam & Hunt, 2006, s. 17-18)

Ketterässä toiminnassa tulee pyrkiä toimimaan mahdollisimman yksinkertaisesti, jottei byrokraattisiin tehtäviin kulu turhaa aikaa. Kuitenkin rakenteita tarvitaan ketterissäkin menetelmissä sopivissa määrin pitämään toiminta koossa. (Highsmith, 2004, s. 72) Joustavuuden ja rakenteen tasapainoilussa suurimpana ongelmana on, että miten saadaan määriteltyä konteksti mahdollisimman kapeasti, mutta samalla oppien ja pystyen mukautumaan uusiin tilanteisiin. Ketteryydessä kuitenkin tulee enemmän luottaa tiimin joustavuuteen kuin suunnitteluun. Liian vähäinen suunnittelu voi kuitenkin tehdä ketteryyden toimimattomaksi ilman kunnollista ohjausta. (Highsmith, 2002, s. 33-34)

Tavallisesti ohjelmistoprojekteissa, jossa asiakas ei ole jatkuvasti mukana voi kehittäjillä olla oletuksena, etteivät asiakkaat aina välttämättä täysin tiedä mitä haluavat ohjelmistolta. Tällä oletuksella voidaan pahimmillaan päätyä tilanteeseen, jossa kehittäjä vaatii täsmällisen määritelmän kehitettävästä ohjelmistosta heti alussa. Täsmällisen määrittelyn avulla kehittäjä siirtää vastuuta asiakkaalle, mikäli ohjelma ei ole sellainen kuin asiakas lopulta haluaisi. Toinen merkittävä mahdollinen oletus kehittäjillä on, ettei

asiakas ole riittävän kauaskatseinen. Tällöin voi käydä niin, että kehittäjä tekee ohjelmaan omia lisäyksiään siinä oletuksessa, että ne auttaisivat asiakasta tulevaisuudessa ja toisivat ohjelmalle enemmän joustoa. (Highsmith, 2002, s. 61)

Agile Manifeston mukaisesti ketterissä menetelmissä asiakas on merkittävässä asemassa ohjelmiston kehittämisessä (Beck et al., 2001b). Asiakas on tässä tilanteessa se henkilö tai ryhmä, joka saa kehitettävästä ohjelmistosta arvoa liiketoiminnalleen (Highsmith, 2004, s. 13). Asiakkaat ovat ohjelmiston kehittämisessä mukana koko projektin ajan monella tavalla. Heidän tehtäviinsä kuuluu käyttäjäkertomusten tekeminen, ohjelmiston arviointi, vaatimusmäärittelyjen asettaminen ja toiminnallisten testien tekeminen. (Rico et al., 2009, s. 35) Projektin aikana jokaisen iteraation alussa asiakkaalla on merkittävä rooli, sillä hän päättää suuntaviivat ja prioriteetit tulevalle iteraatiolle. Näitä suuntaviivoja ja asetettujen vaatimusten toteutumista asiakas seuraa koko projektin ajan ja antaa palautetta kehittäjille ohjelmistosta. (Hazzan & Dubinsky, 2014, s. 27-28)

Päätöksenteossa asiakas on ratkaisevassa roolissa, sillä asiakkaan tulisi saada päättää kaikesta, joka on kriittisessä roolissa asiakkaan liiketoiminnan kannalta. Tällöin suunnittelijoiden ja kehittäjien tulee tarjota asiakkaalle riittävästi tietoa eri vaihtoehtojen hyvistä ja huonoista puolista liiketoiminnan kannalta. Lisäksi eri vaihtoehtojen hinnasta, aikataulusta ja vaikutuksista tulee keskustella asiakkaan kanssa tarkasti, sillä asiakkaan päätös pohjautuu kyseisiin tietoihin. Päätökset, jotka eivät puolestaan vaikuta asiakkaan liiketoimintaan tulee tehdä kehitystiimissä. Nämä päätökset ja niiden syyt tulee kirjata ylös mahdollista myöhempää tarkistamista varten, kuitenkin niin, ettei kirjaaminen käy liian raskaaksi. (Subramaniam & Hunt, 2006, s. 47-48)

Asiakkaiden mielestä ketterät menetelmät ovat paras tapa toteuttaa ohjelmistoprojekti. Kuitenkin ketterien menetelmien käytäntöjä noudattamalla päädytään tilanteeseen, että asiakas työskentelee paljon enemmän vuorokaudessa kuin perinteisessä ohjelmistokehitysprojektissa. Kyseinen työskentelymäärä on kestävämpi, jolloin voi syntyä merkittäviä ongelmia pitkissä ja kovan paineen alaisissa projekteissa. Asiakkaan työaikaa saadaan kuitenkin lyhennettyä merkittävästi lisäämällä kehitystiimin ja asiakkaan intensiivistä työskentelyä, tehostamalla asiakkaan todellista osallistumista ja tehostamalla koko tiimin toimintaa. (Martin et al., 2010) Asiakas kommunikoi tiimille

mieluiten sähköpostitse tai joissain tilanteissa puhelimitse sekä selkeistä että epäselvistä asioista projektin iteraatioiden aikana. Kuitenkin, mitä kevyempiä kommunikointimenetelmiä käytetään tiimin ja asiakkaan välisessä kommunikoinnissa, sitä enemmän ohjelmistossa on vikoja. Asiakkaan ja tiimin välillä suositeltavinta olisi mahdollisimman intensiivinen ja merkityksellinen kommunikointi. Tämän avulla väärinymmärrykset tiimin ja asiakkaan välillä pienenevät ja samalla vikojen määrä ohjelmistossa pienenee. Kuitenkin kommunikointiongelmien seurauksena aiheutuneet ongelmat ovat pienempi riski ketterissä menetelmissä kuin perinteisessä ohjelmistokehityksessä. Tämä johtuu lyhyistä ja jatkuvista iteraatioista, jolloin viat saadaan huomattua ja korjattua mahdollisimman pian. (Korkala et al., 2006)

Varsinaisia kriittisiä menestystekijöitä ketterän ohjelmistoprojektin onnistumiseen on löydetty kolme. Nämä ovat oikea toimitusstrategia, oikeanlainen ketterien ohjelmistokehitysmenetelmien käyttäminen ja korkeatasoinen tiimi. Lisäksi lähes kriittisiksi menestystekijöiksi voidaan laskea hyvä ketterän ohjelmistoprojektin hallintaprosessi, hyvin ketteriin menetelmiin suhtautuva tiimiympäristö ja vahva asiakkaan osallistuminen. (Chow & Chao, 2008)

Abrahamssonin et al. (2010) mukaan ketteristä menetelmistä puuttuu todellinen tuki projektinhallinnalle. Tämän lisäksi ei ole perusteltu ohjelmistokehityksen elinkaaren pituutta, vaikka se vaihtelee eri ketterien menetelmien välillä. Lisäksi kirjallisuutta hallitsevat käytännön sijaan abstraktit ohjeet. Tarjolla ei myöskään ole mekanismeja, joilla ottaa käyttöön eri ketteriä menetelmiä käytännössä. Yhtenä haasteena on myös, että ketterien menetelmien tutkimus on yleensä tehty ratkaisua tukevalla tavalla ja empiirisiä todisteita menetelmistä on vähän. (Abrahamsson et al., 2010)

2.2 Ketterän ohjelmistoprojektin eri muotoja

Ensimmäinen nykyaikaisen ketterän menetelmän käytäntö oli Extreme Programming (XP) (Cline et al., 2015, s. 11). XP:n keskeisimpinä osina ovat tekniikka ja hyvät suhteet. Tämän menetelmän avulla pyritään muuttamaan koodaajien työskentelytapaa vähemmän itsenäiseksi ja enemmän yhteistyön tekemiseksi. XP:ssä tärkeässä roolissa ovat erinomaiset ohjelmointitaidot sekä selkeä kommunikointi ja tiimityöskentely. XP:lle

tunnusomaista on keveys, eli ei tehdä mitään ylimääräistä, joka ei tuota asiakkaalle arvoa. Lisäksi tämä menetelmä on hyvin skaalautuva kaikenkokoisille tiimeille, se pohjautuu käsittelemään rajoitteita ohjelmistokehityksessä ja sopeutuu sekä epämääräisiin että nopeasti muuttuviin vaatimuksiin. (Beck, 2004, s. 1-3)

XP:ssä asiakkaalla tarkoitetaan henkilöä, jolla on määräysvalta kehitettävän ohjelmiston ominaisuuksista. Asiakkaan tulisikin toimia mahdollisimman lähellä kehitystiimiä, mieluiten jopa samassa huoneessa, sillä asiakas on osa tiimiä. Toimivaa ohjelmistoa toimitetaan kahden viikon välein sidosryhmille, jotka arvioivat ohjelmiston. Jokaisen kahden viikon aikainen työskentely pohjautuu iteraation alussa asiakkaan kanssa sovittujen vaatimusten toteuttamiseksi. Ohjelmiston kehittämisessä asiakas määrittelee hyväksyttämistestejä, joiden avulla määritellään tietyn osan toimivuus. Ohjelman läpäistyä tällainen testi, tulee sen olla siitä eteenpäin aina toimiva ja osa lopullista ohjelmistokokonaisuutta. Itse koodaaminen tapahtuu pareittain saman työpisteen äärellä, jolloin toinen kirjoittaa koodia ja toinen tarkistaa koodia virheiden varalta. Kirjoittamisen jälkeen koodia testataan mahdollisimman pian ja mikäli koodi ei läpäise testiä, tulee se välittömästi korjata, jotta se läpäisisi testauksen mahdollisimman pian. Koodi itsessään on koko tiimin yhteisessä käytössä ja kaikki kehittäjät integroivat jatkuvasti kirjoitettuja ohjelman osia osaksi lopullista ohjelmistoa. Itse työympäristön tulisi olla avoin ja tiimin jäsenet kuuloetäisyydellä toisistaan. Tahdin tulisi olla tasainen ja tiimi ei saisikaan työskennellä ollenkaan yliaikaa kuin aivan välttämättömissä tilanteissa. XP:n menetelmillä tehtävän ohjelmiston tulisi olla mahdollisimman yksinkertaista. Siinä tulisi välttää saman koodin kirjoittamista useampaan kertaan ja jättää rakenteiden tekeminen niin myöhään kuin mahdollista. Tällöin ohjelmiston pitäisi pysyä mahdollisimman kevyenä ja yksinkertaisena. (Martin, 2003, s. 11-16)

Scrum on ketterä ohjelmistokehityksen menetelmä, jossa toimitaan 30-päivän mittaisissa Sprinteissä. Jokaisen Sprintin aikana tehdään toimiva ohjelmisto. (Highsmith, 2002, s. 243) Scrum:n mukaisessa projektissa on kolme eri roolia: tuotteen omistaja, tiimi ja Scrum-mestari. Tuotteen omistajalla tarkoitetaan henkilöä, joka määrää kehitettävän ohjelmiston vaatimuksista ja projektin tavoitteista. Tiimi puolestaan kehittää itse ohjelmiston ja toimii itseohjautuvasti. Scrum-mestari on taas vastuussa itse Scrum:n menetelmien implementoinnista ja niiden opettamisesta tiimille. Scrumissa jokainen

Sprint lähtee liikkeelle Product Backlog:sta, johon on kirjattu kaikki systeemin vaatimukset. Tämän perusteella aletaan suorittaa 30-päivän iteraatiota. Sprint:n aikana järjestetään päivittäin 15-minuutin tapaaminen, jossa kaikki päivittävät muille tilanteen oman työskentelynsä suhteen. Sprint:n päätteeksi esitellään ohjelmiston uudet ominaisuudet tuotteen omistajalle ja muille halukkaille sidosryhmille. Esittelyn päätteeksi on selvillä, mitä seuraavaksi tulee tehdä ohjelmistoon. Tämän jälkeen Scrum-mestari pitää tiimin kanssa tapaamisen, jossa käsitellään edellisen Sprint:n toiminta ja miten toimintaa tulisi kehittää seuraavaa Sprint:ä varten. Tapaamisen jälkeen suunnitellaan seuraava Sprint Product Backlog:n ja viimeisimmän esittelyn perusteella. Samalla kaavalla jatketaan kehittämistä, kunnes projekti päättyy. (Schwaber, 2004, s. 6-10)

Crystal metodologia jakautuu useampiin eri metodologioihin, jotka soveltuvat käytettäväksi eri tilanteissa ja on nimetty eri värien mukaan. Yhtenäistä näissä Crystal:n eri metodologioissa on, että kaikki jakavat samat arvot ja ovat mukautuvia kesken toiminnan. Crystal metodologian perusidea on, että ensisijaisesti pyritään ohjelmistokehityksessä toimittamaan hyödyllistä ja toimivaa ohjelmistoa ja toissijaisena päämääränä on valmistautua tulevia ohjelmistokehityksen tilanteita varten. Crystal:n metodologioissa on kaksi yhteistä sääntöä. Ensimmäinen on, että kehittäminen tapahtuu korkeintaan neljän kuukauden mittaisissa inkrementeissä. Toinen sääntö on, että tiimin tulee pitää inkrementtiä ennen ja jälkeen refleктоiva työpaja. (Cockburn, 2006, s. 335, 337, 339)

Muita menetelmiä ovat esimerkiksi Feature-Driven Development (FDD), Dynamic Systems Development Method (DSDM), Lean Development ja Adaptive Software Development (ASD). Nämä eroavat hieman aiemmin esitetyistä ketteristä menetelmistä, mutta noudattavat kuitenkin ketterien menetelmien periaatteita. DSDM on jatkoa Rapid Application Development:lle ja keskittyy erityisesti nopeisiin muutoksiin ja ratkaisujen kehittämiseen. FDD:llä on puolestaan minimalistinen näkökulma ja siinä kehitys tapahtuu nimensä mukaisesti ominaisuus kerrallaan. Lean Development on eniten strategia-orientoitunut ketterä menetelmä, jossa korostuu muutoksen näkeminen mahdollisuutena. ASD on filosofisempi ketterä menetelmä, jossa myös muutoksen välttämisen sijaan pyritään hyötymään siitä. (Highsmith, 2002, s. 251, 273, 287, 310)

2.3 Ketterän ohjelmistoprojektin tiimin jäsenet

Ketterissä menetelmissä tiimit ovat yleensä pieniä ja vaikka tiimi olisikin iso, niin se on usein jaettu useisiin pieniin, korkeintaan kymmenen hengen ryhmiin. Tiimit työskentelevät samassa huoneessa ja muokkaavat jatkuvasti yhdessä samaa koodia. (Subramaniam & Hunt, 2006, s. 4) Tiimi tulisi luoda ennen toimintaympäristöä. Tällöin tiimillä on mahdollisuus luoda itse toimintaympäristönsä, jossa se pystyy parhaiten toimimaan. (Martin, 2003, s. 4-5)

Suurimpana haasteena ohjelmistokehitystiimin muodostamisessa on oikeiden henkilöiden valinta. Tämä johtuu pitkälti itsessään työn luonteesta, sillä ohjelmiston kehittäminen on tietopohjaista työtä. Tämä mahdollistaa tilanteet, että kaksi kehittäjää, jotka omaavat saman taidon, voivat selviytyä ohjelmistonkehityksestä kahdella täysin eri tavalla. (Ruhe & Wohlin, 2014, s. 91) Ketterässä tiimissä jäsenten tärkeimpiin ominaisuuksiin kuuluvat ystävällisyys, lahjakkuus, taito ja kommunikointikyky (Cockburn & Highsmith, 2001).

Tiimin kyvykkyys ja itsekuri kertovat siitä, onko tiimissä oikeat henkilöt. Jäsenten valinnan lisäksi myös asiakkaan valikoinnilla on merkitystä. Tiimin ei tulisi ottaa vastaan sellaista projektia, johon asiakas ei kykene osallistumaan riittävästi. Tiimin sisäinen tai asiakkaan ja tiimin välinen riittämätön yhteistyö johtaa todennäköisesti projektin epäonnistumiseen. (Highsmith, 2004, s. 109-110)

Tiimi on lisäksi itseorganisoituva ja sen jäsenet vastaavat itse omasta työstään sekä koordinoivat työn ryhmän kesken (Highsmith, 2004, s.64). Itseorganisoitumiseen kuuluu myös suhteellisen vapaa päättäminen työntekotavasta, sillä vain lopputuloksella on merkitystä (Medinilla, 2012, s. 75). Itseorganisoituvan tiimin toiminnan edellytyksenä ovat seuraavat asiat: oikeiden ihmisten saaminen tiimiin, selvästi ilmaistu tuotteen visio, tiimin rajoitteiden ja roolin selvä ilmaiseminen, kannustaminen eri tiimien väliseen vuorovaikutukseen ja tiedonkulkuun, osallistuvan päätöksenteon helpottaminen, vastuullisuuden vaatiminen tiimiltä sekä tiimin ohjaaminen kontrolloinnin sijaan. Tuotteen vision ja rajoitteiden selkeä ilmaiseminen on tärkeässä roolissa, sillä sen avulla työntekijät pystyvät paremmin tekemään päätöksiä jokapäiväisessä työssään ja paremmin

ymmärtämään omaa rooliaan projektissa. Jatkuvan muutoksen vuoksi visiota ja rajoitteita tulee kuitenkin jatkuvasti päivittää ja tuoda tiimin jäsenille selvästi esille muutokset. Vuorovaikutus ja vapaa tiedonkulku tiimin sisällä ovat avainasemassa laadukkaiden tulosten saavuttamisessa. Näiden lisäksi tiimin ytimessä on muiden tiimin jäsenten kunnioittaminen ja heihin luottaminen. Tämä takaa toimivan keskustelun ja tiedon jakamisen tiimin jäsenten välillä. Osallistavalla päätöksenteolla tarkoitetaan yhteistyössä eri henkilöiden kanssa tiimissä tehtävää päätöksentekoa, joka on merkittävä osa todellisen yhteistyön tekemisessä. Vastuun kantamisen vaatimisella luodaan luottamusta yhteistyölle. Ohjaamalla tiimiä saadaan pidettyä tiimin suunta kehityksessä oikeana koko ajan samalla, kun tiimi toimii mahdollisimman itsenäisesti. Johtajan tehtäväksi jää tällöin ohjaamisen lisäksi yksilöiden valmentaminen ja tiimiin sopeutumattomien henkilöiden poistaminen tiimistä. (Highsmith, 2004, s. 65-71)

Ketterissä menetelmissä vaaditaan merkittävää tasapainoilua joustavuuden ja vakauden välillä. Tämä puolestaan vaatii henkilöiltä hyvää improvisointikykyä ja sopeutumista epämääräisyyteen. Tähän kykenevät henkilöt tarvitsevat organisaatiolta mukautuvaa toimintakulttuuria, minimaalisen määrän sääntöjä tiimin toiminnassa ja intensiivistä yhteistyötä projektiyhteisön sisällä. Mukautuvalla toimintakulttuurilla pystytään sopeutumaan muutokseen ja sääntöjen puute puolestaan kannustaa itseohjautuvuuteen. (Highsmith, 2004, s.17)

Ketterissä menetelmissä projektinhallinta on enemmän yhteistyöllä toimivaa johtajuutta kuin käskyillä hallitsemista. Lisäksi projektipäälliköiden työ keskittyy enemmän ylemmän tason johtamiseen kuin yksittäisten ja pienten asioiden johtamiseen. (Cockburn & Highsmith, 2001) Tämän seurauksena päätöksenteko erityisesti pienemmistä asioista on siirretty esimiehiltä alaisille. Päätöksenteon siirtäminen puolestaan synnyttää vahvaa luottamusta esimiesten ja alaisten välille. (Highsmith, 2004, s. 38)

Tiimin sisällä kritiikki tulisi kohdistaa ideoihin eikä ihmisiin. Tällöin henkilöt todennäköisemmin ehdottavat jatkossakin uusia ideoitaan, joista voi olla merkittävää hyötyä. Tiimissä tulisi olla useita todella lahjakkaita ihmisiä, jotta työskentely olisi mahdollisimman tehokasta. Kuitenkin näiden henkilöiden tulee pystyä lahjakkuudestaan huolimatta työskentelemään yhdessä, jotta tiimi säilyy tuottavana. Ketterä kehitys vaatii

tiimin jäseniltä sitoutumista ja oikeaa asennetta toimimisessa ketterien menetelmien mukaisesti. (Subramaniam & Hunt, 2006, s. 3, 12, 19, 20)

Tiimin toiminnassa pyritään jatkuvasti löytämään ratkaisua esille tuleviin ongelmiin pyrkien koko ajan päämäärää kohti (Subramaniam & Hunt, 2006, s. 13). Näihin päämääriin pääsemiseksi tiimin tulee käyttää jossain määrin erilaisia työkaluja. Kuitenkin työkalujen käyttäminen tulisi aloittaa mahdollisimman yksinkertaisilla ja pienillä työkaluilla. Vasta huomattuaan todellisen tarpeen paremmille ja isommille työkaluille tulisi tiimin harkita niiden hankkimista. (Martin, 2003, s. 4-5)

Ketterässä tiimissä kannustetaan kysymään epäselvistä asioista ja oppimaan uutta. Eräs menetelmä tiimin jäsenten tiedon lisäämiseksi ja kehittämiseksi ovat ns. brown-bag tapaamiset. Näissä kerran viikossa yksi tiimin jäsen jakaa tietoaan jostain tietystä asiasta, joka kiinnostaa tiimiä. Tapaaminen kannattaa pitää mieluiten lounastauolla, jottei se mene päällekkäin muiden asioiden kanssa ja mieluiten jonain muuna päivänä kuin maanantaina tai perjantaina, jolloin kuuntelijat keskittyvät parhaiten. (Subramaniam & Hunt, 2006, s. 32-33)

3 KOMMUNIKOINTI OHJELMISTOPROJEKTISSA

KETTERÄSSÄ

3.1 Kommunikoinnin perusteet tiimissä

Kommunikointi pohjautuu yleisesti malliin, jossa on neljä komponenttia: viestin lähde, lähettäjä, vastaanottaja ja määränpää. Informaatio itsessään liikkuu lähettäjältä vastaanottajalle jotain tiettyä kanavaa pitkin. Kanavaan kuitenkin kohdistuu ulkopuolista melua, joka voi vaikuttaa viestin välittymiseen ja pahimmillaan estää viestin välittymisen. (Goldie & Pinch, 1991, s. 1) Lisäksi henkilön omat kommunikoinnin suodattimet voivat häiritä viestin vastaanottamista. Näitä suodattimia ovat ennakkoluulo, tiedonpuute, aiempi kokemus, tiedon ylikuormitus, kiinnostuksen tai motivaation puute ja epäluottamus viestin lähettäjään. (Parker, 2009, s. 7)

Kommunikointikanava on se tapa, jolla viesti välittyy sen lähettäjältä vastaanottajalle. Kommunikointikanavat voivat olla esimerkiksi yksi-, kaksi- tai monisuuntaista kommunikointia, teknologian välittämää, synkronoitua tai synkronoimatonta, yksilö- tai ryhmäkommunikointia, paperitulosteena tai sähköisessä muodossa, pysyvä tai ohimenevä, virallinen tai epävirallinen sekä suppea tai runsas. Priesley:n paradoksin mukaisesti kommunikointikanavien määrä heikentää kommunikoinnin laatua. Näin ollen teknologian kehittyminen ei aina välttämättä takaa kommunikoinnin kehittymistä. (Eunson, 2007, s. 8, 22, 23)

Tiimin menestymisen määrää ensisijaisesti tiimin jäsenten motivaatio ja ihmisten arvostus (Wong, 2007, s. 16). Tiimin tärkeänä ominaisuutena on, että yksilöillä on yhteinen päämäärä, johon pyritään yhteistyössä tehdyillä päätöksillä ja toiminnalla. Päätösten tekemisessä ja aktiviteeteissa on tärkeää tiedon ja resurssien jakaminen tiimin jäsenten välillä. Näin ollen nähdään, että kommunikointi tiimin sisällä on tärkeässä roolissa koko tiimin toiminnan ajan ohjaamassa toimintaa. (Dickinson & McIntyre, 1997, s. 21)

Tiimityöskentely koostuu seitsemästä komponentista: tiimin orientaatiosta, tiimin johtamisesta, kommunikoinnista, seurannasta, palautteesta, tukitoiminnoista ja

koordinoinnista. Tiimin orientaatio tarkoittaa asennetta, joka tiimin jäsenillä on toisiaan ja tiimin tehtäviä kohtaan. Tiimin johtaminen tarkoittaa rakenteen, tuen ja suunnan tarjoamista muille tiimin jäsenille. Lisäksi johtajuuteen kuuluu muiden tiiminjäsenten huolenaiheiden kuunteleminen ja tehtävien vaatimusten selittäminen muille tiimin jäsenille. Kommunikoinnilla tarkoitetaan kahden tai useamman henkilön välistä informaation vaihtamista. Kommunikoinnin tarkoituksena on lisäksi vahvistaa tai selventää viestin saaminen. Seurannalla tarkoitetaan toisten tiimin jäsenten toiminnan seuranta, joka mahdollistaa palautteen antamisen ja tukemisen. Palautteeseen kuuluu palautteen saaminen ja antaminen tiimin jäsenten välillä. Palaute itsessään on informaatiota toisen henkilön suoriutumisesta. Tukitoiminnalla tarkoitetaan muiden tiimin jäsenten avustamista parempaa tehtävistä suoriutumisesta varten. Tiimin jäsenten tulisi olla halukkaita hankkimaan ja antamaan tukea tilanteen vaatiessa. Koordinoinnilla tarkoitetaan, että tiimin jäsenet saavat tehtävänsä hoidettua integroidusti ja oikea-aikaisesti. Tämä tarkoittaa myös, että tiiminjäsenten toiminta vaikuttaa myös muiden tiimin jäsenten toimintaan esimerkiksi tarvittavan tiedon kautta. (Dickinson & McIntyre, 1997, s. 25)

3.2 Käytännön haasteet tiimin sisäisessä kommunikoinnissa ketterissä projekteissa

Yleisesti kommunikoinnissa haasteita aiheuttavat faktat, tunteet, henkilökohtaiset arvot ja mielipiteet. Faktat ovat objektiivista informaatiota, mutta ne tulevat esteiksi kommunikaatiossa, kun toisia faktoja otetaan huomioon ja toisia ei tai jos jotain tiettyä faktaa painottaa erityisesti. Tunteet vaikuttavat puolueellisesti keskustelun lopputulokseen, sillä tällöin keskusteluun sisältyvät henkilön omat kiinnostuksen kohteet. Henkilökohtaisista arvoista vaikuttavat päätöksentekoon konfliktiin ja ilmapiiriin liittyvät arvot. Henkilön omat mielipiteet vaikuttavat kommunikointiin esim. vaihtoehtoista ja päätöksistä keskusteltaessa. (Parker, 2009, s. 24)

Kommunikoinnissa omat haasteensa asettaa hukka. Kommunikoinnille on tunnistettu viisi eri hukan muotoa: osallistumisen puute, yhteisen ymmärryksen puute, vanhentunut informaatio, rajoitettu pääsy informaatioon ja hajallaan oleva informaatio. Näitä hukan

muotoja olisi tärkeä tunnistaa tiimin toiminnassa. Tunnistamisen avulla näitä on mahdollista vähentää ja näin ollen parantaa kommunikointia. (Korkala & Maurer, 2014)

Yhtenä haasteena voi olla tiimin jäsenten aiempi tottuminen itsenäiseen työhön, jonka seurauksena jäsenet keskittyvät pääosin vain omiin osioihinsa projektissa. Tämän seurauksena he tekevät itsenäisesti päätöksiä ja suunnitelmia sekä näkevät esiintyvät ongelmat ominaan. Lisäksi Moen et al. (2010) tekemässä tutkimuksessa oli havaittavissa, että vähäisen tiimiorientaation seurauksena tiimin jäsenillä ei ollut tarkkaa tietoa toistensa tekemisestä ja näin ollen toisten suoriutumisen seuranta ei onnistunut. Tätä haastetta lisäsi myös merkittävän erikoistunut osaaminen tiimin jäsenillä ja tätä osaamista vastaava työnjako. (Moe et al., 2010)

Haasteena kommunikoinnissa voi olla myös toimiston fyysinen rakenne. Mikäli tiimin jäsenet työskentelevät kaukana toisistaan, aiheutuu tästä useita negatiivisia asioita informaation välittymisessä. Huono toimiston rakenne voi johtaa kysymysten kysymättä jättämiseen, heikentyneeseen informaation havaitsemiseen ja siirtämiseen sekä taustalta havaittavan informaation menettämiseen. Taustalta havaittavalla informaatiolla tarkoitetaan esimerkiksi henkilöiden välistä keskustelua, johon on ulkopuolisella henkilöllä mahdollista päästä mukaan alitajuntaisella kuuntelulla. Parhaimmillaan tällä ulkopuolisella kuulijalla on ratkaisu taustalla keskustelun kohteena olevaan asiaan, jolloin ongelman ratkeaminen nopeutuu ulkopuolisen kuulijan auttamalla keskustelijoita omalla tietämyksellään. Nämä kaikki edellä mainitut negatiiviset asiat tulevat näkyviin toimiston rakenteesta, mikäli tiimin jäsenet työskentelevät kaukana toisistaan. (Cockburn, 2006, s. 111)

Tiimillä voi tulla haasteita riittävässä informaation jakamisessa. Tavoitteina informaation jakamisella tiimissä on mm. löytää konsensus jäsenten välille ja jakaa jäsenten asiantuntemus ja tietämys. Tiedon jakaminen antaa mahdollisuuden saavuttaa perehtyneempi päätös kuin ryhmän jäsenten itsenäisesti tekemä päätös. Tiimin jäsenet usein kuitenkin epäonnistuvat jakaessaan tietojaan ryhmälle, sillä keskustelua dominoi tieto, jonka kaikki ryhmän jäsenet tietävät ja joka tukee henkilön aiempia mieltymyksiä. Tämän seurauksena ei usein saada kaikkea olennaista informaatiota jaettua, sillä keskustelu on harvoin systemaattista ja tasapainoista tutkimista. (Stasser & Titus, 1985)

Mikäli tiimi ei pääse työskentelemään fyysisesti yhdessä, voidaan puhua virtuaalitiimistä. Tällainen tilanne aiheuttaa tiimin työskentelylle huomattavasti haasteita ja lisää projektin kustannuksia. Nämä johtuvat pääosin siitä, että ideoiden siirtämiseen kuluu enemmän aikaa ja vaivaa. Lisäksi virtuaalitiimissä työskentely aiheuttaa merkittäviä menetyksiä mahdollisuuksissa, kun joihinkin olennaisiin kysymyksiin ei saada vastausta. (Cockburn, 2006, s. 224) Kuitenkaan informaation siirtäminen kasvokkain ei välttämättä ole yhtään tehokkaampaa kuin virtuaalisesti. Tällöin virtuaalisen tiedonsiirron toimimisen edellytyksenä on kommunikoinnin jäsenten kärsivällisyys, peräänantamattomuus, sinnikkyys, joustavuus ja kärsivällisyys. Lisäksi kommunikoiden tulee tuntee käytettävät työkalut ja teknologia hyvin, jotta kommunikointi niillä olisi tehokasta. Eri teknologioiden avulla kommunikointi kuitenkin vaatii niillä tapahtuvien kommunikointimenetelmien opettelemista, mitkä eroavat selvästi kasvokkain kommunikoinnista. (Warkentin et al., 1997)

3.3 Hyvät käytänteet tiimin sisäisessä kommunikoinnissa ketterissä projekteissa

Ketterän tiimin toiminnan kannalta keskeisessä osassa kommunikoinnissa ovat tarinalaput ja taulu tarinalapuille. Tarinalaput ovat pieniä lappuja, joihin on kirjoitettu lyhyesti jostain tietystä asiasta. Laput voivat olla värikoodattuja ja niissä voi olla esimerkiksi asiakaskertomuksia, arvioita sekä luonnoksia ja suunnitelmia. Taulu tarinalapuille on puolestaan seinällä oleva taulumainen alue, johon tarinalappuja kiinnitetään. Taulun avulla voi helposti esimerkiksi seurata projektin edistymistä tai ohjelmistossa olevien vikojen tilannetta. (Sharp & Robinson, 2010)

Päivittäiset tapaamiset tiiminjäsenten kanssa on myös hyvä tapa kommunikoida. Tällöin on mahdollista päästä paremmin perille muiden tiimin jäsenten tekemisistä ja samalla parantaa suhteita muihin tiimin jäseniin. Tapaamisen tulisi tapahtua kasvokkain ja olla kuitenkin tehokas, jottei siihen kulu liikaa aikaa. (Medinilla, 2012, s. 44)

Tiimin oppimisen kannalta on suotavaa, että tiimin sisäinen kommunikointi olisi mahdollisimman avointa (So, 2010, s. 152). Tähän sisältyy osaltaan palautteen antaminen, joka on usein vaatimuksena onnistuneelle kommunikoinnille (Appelo, 2011,

s. 274). Palautteen antamisessa tulisi olla täsmällinen ja keskittyä tilanteen kuvailemiseen eikä arviointiin. Lisäksi palaute tulisi antaa mahdollisimman pian tapahtuman jälkeen ja palautteen antajan tulisi kuunnella palautteen saajan näkökulmaa asiasta. (Parker, 2009, s. 25)

Keskustellessa kasvokkain tulisi pyrkiä tehokkaaseen viestin lähettämiseen. Tällöin tulisi lähettäjän ainakin ymmärtää itse lähettämänsä viesti, olla täsmällinen ja pitää verbaalinen sekä non-verbaalinen kommunikointi linjassa keskenään käyttämällä useita viestintätapoja ja räätälöimällä viesti vastaanottajalle. Viestin lähettäjän tulee esittää viesti omaksi, esimerkiksi minä-persoonan käytöllä. Lisäksi viesti on hyvä räätälöidä tilanteeseen sopivaksi ja sopivissa määrin tulisi suosia metaforia sekä analogioita. (Parker, 2009, s. 18)

Puhumisen lisäksi tehokkaimmat kommunikoijat kysyvät hyviä, avoimia kysymyksiä ja ovat aktiivisia kuuntelijoita. Aktiivisena kuuntelijana tavoitteena on ymmärtää toisen viesti sellaisena kuin hän sen tarkoitti. Kuuntelua henkilö voi osoittaa olemalla läsnä ja pyrkimällä muotoilemaan vastaus vasta kun toinen on sanonut asiansa loppuun saakka. Kysymykset puolestaan tulisi olla avoimia, jotta keskustelu olisi avoimempaa ja informaatiota välittyisi mahdollisimman paljon. (Parker, 2009, s. 16, 22)

4 POHDINTA

Kirjallisuuskatsauksen perusteella ketterissä menetelmissä tehokkaimmaksi ja parhaaksi väitetty kommunikointitapa on kasvokkain kommunikointi. Tämän lisäksi kirjallisuuden perusteella kommunikoinnin tulisi tapahtua avoimesti ja tiiviissä yhteistyössä tiimin jäsenten kesken. Kommunikoinnissa erityisen toimivaksi menetelmäksi ketterissä ohjelmistoprojekteissa Sharp ja Robinson (2010) näkivät tutkimuksensa perusteella tarinalaput ja taulun, johon näitä lappuja laitettiin. Näiden avulla pystytään korvaamaan dokumentointia ja samalla yleisesti seuraamaan projektin etenemistä. Tällöin informaatio on helposti kaikkien saatavilla, mutta informaation tuottaminen ei kuitenkaan ole liian vaivalloista.

Aiemmin esitellyn kirjallisuuden perusteella kommunikoinnissa tiimin sisällä on merkittävänä haasteena kaikkien tiimin jäsenten pysyminen ajan tasalla projektista ja riittävän dokumentoinnin määrän määrittäminen. Nämä vaativat hyvin avointa kommunikointiympäristöä ja hyvää yhteisymmärrystä tiimin jäsenten välillä. Lisäksi oman haasteensa asettaa ketterien menetelmien nopea eteneminen ja tämän seurauksena mahdollisesti syntyvä stressi tiimin jäsenille. Kiire varmasti aiheuttaakin jossain määrin tasapainoilua laajan kommunikoinnin ja omien työtehtävien suorittamisen välillä. Molemmat ovat kriittisessä osassa ketterässä ohjelmistoprojektissa, mutta ketterien menetelmien mukaisesti kommunikointi on korostetummassa osassa. Kuitenkaan kirjallisuuden perusteella ei selvinnyt, missä suhteessa tulisi keskittyä kommunikointiin ja omien työtehtävien suorittamiseen.

Yhtenä haasteena kommunikoinnin kannalta kirjallisuuden perusteella on myös tiimin kokoaminen. Erityisen haastavaa on oikeiden ihmisten saaminen projektiin ja mahdollinen tiimin jäsenten vaihtuvuus projektin alkuvaiheessa. Tämä voi pahimmillaan aiheuttaa epävarmuutta tiimin sisällä, joka heikentää kommunikoinnilta vaadittavaa avoimuutta. Tämä on erityinen haaste projektipäällikölle, joka on viime kädessä vastuussa projektin henkilöstöstä.

Kirjallisuuden perusteella virtuaalinen kommunikointi eri teknologioiden kautta on vaativampaa ja mahdollisesti tehottomampaa kuin kasvokkain kommunikointi.

Kuitenkaan kirjallisuudesta ei selvinnyt, onko mahdollista yhdistää kasvokkain kommunikointia ja virtuaalista kommunikointia ketterässä tiimissä. Esimerkkinä tällaisesta tilanteesta voisi olla videoyhteys, jolloin henkilöllä ei välttämättä tarvitsisi nousta oman työpisteensä äärestä ja keskeyttää tekemistään. Tässä olisi vielä potentiaalista jatkotutkimuskohdetta. Kaikki hyvät käytänteet ja haasteet kommunikoinnissa tiimin sisällä ketterässä ohjelmistoprojektissa on esitetty alla olevassa taulukossa 1.

Hyvät käytänteet	Haasteet
Kasvokkain kommunikointi	Virtuaalitiimi ja huono toimiston rakenne
Avoin kommunikointi ja palautteen antaminen	Riittävä informaation jakaminen
Työskentely lähekkäin	Aiemmat tottumukset itsenäisestä työskentelystä
Aktiivinen kuuntelu ja avoimet kysymykset	Häiriöt kommunikointikanavassa ja henkilökohtaiset kommunikoinnin suodattimet
Verbaalisen ja non-verbaalisen kommunikoinnin oleminen linjassa keskenään	Faktat, tunteet ja mielipiteet
Oikeiden ihmisten valinta tiimiin	Tiimin jäsenten vaihtuvuus
Tarinalaput ja taulu	Hukka
Päivittäiset tapaamiset ja uuden tiedon jakaminen	

Taulukko 1. Yhteenveto hyvistä käytänteistä ja haasteista kommunikoinnissa tiimin sisällä ketterässä ohjelmistoprojektissa.

5 YHTEENVETO

Kandidaatintyössä tarkasteltiin tiimin sisäistä kommunikointia ketterissä ohjelmistoprojekteissa pyrkien löytämään haasteita ja hyviä käytäntöjä kommunikoinnille. Aluksi työssä käytiin läpi ketteriä menetelmiä yleisesti sekä näihin liittyviä käytäntöjä. Erityistä painotusta tässä oli yksittäisten tiimin jäsenten toiminnalla tiimin toiminnan tavoitteilla.

Ketterien menetelmien tarkastelun jälkeen tarkasteltiin kommunikoinnin teoriaa. Kommunikoinnin teorian ja ketterien menetelmien teorioiden avulla koottiin yhteen hyviä käytäntöjä ja haasteita kommunikoinnissa tiimin sisällä ketterissä menetelmissä. Tähän sisältyi olennaisimmat käytänteet kirjallisuuden pohjalta. Näiden avulla merkittävimiksi haasteiksi kommunikoinnille löytyi toimiston rakenne, kommunikoinnin hukka, aiemmat työskentelytottumukset, yleiset kommunikoinnin haasteet ja virtuaalitiimin toiminta. Hyviksi käytänteiksi puolestaan selvisi yleisesti hyvät kommunikointitavat, kasvokkain kommunikointi, avoin kommunikointi, tarinalaput ja nämä kokoava taulu sekä päivittäiset tapaamiset.

Aihetta olisi syytä vielä tutkia merkittävästi. Ketteristä menetelmistä on tehty kohtalaisen vähän tieteellisiä tutkimuksia, joka osaltaan aiheuttaa tarvetta jatkotutkimuksille. Kommunikointia on tutkittu paljon, mutta voisi olla hyvä tutkia enemmän kommunikointia ketterissä menetelmissä. Tämä johtuu ketterien menetelmien tuoreudesta ja poikkeuksellisuudesta projektitoiminnassa, joka luonnollisesti johtaa erilaisiin toimintatapoihin.

LÄHDELUETTELO

Abbas N., Gravell A. & Wills G., 2008. Historical roots of agile methods: Where did "Agile thinking" come from? Teoksessa: Concas G., Damiani E., Scotto M. & Succi G. (toim.) Agile processes in software engineering and extreme programming. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 94-103. ISBN 978-3-540-73100-9

Abrahamsson P., Oza N., & Siponen M. T., 2010. Agile software development methods: A comparative review. Teoksessa: Dingsøyr T., Dybå T. & Moe N. B. (toim.) Agile Software Development – Current Research and Future Directions. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 31-59. ISBN 978-3-642-12574-4

Abrahamsson P., Salo O., Ronkainen J., & Warsta J., 2002. Agile software development methods - Review and analysis. Espoo: VTT, 107 s. ISBN 951-38-6009-4

Appelo J., 2011. Management 3.0 - Leading agile developers, developing agile leaders. Indianapolis: Addison Wesley Professional, 413 s. ISBN 978-0-321-71247-9

Beck K. & Andres C., 2004. Extreme programming explained - Embrace change 2. painos. Boston, MA: Addison-Wesley, 189 s. ISBN 0-321-27865-8

Beck K., Beedle M., van Bennekum A, Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., Martin R. C., Mellor S., Schwaber K., Sutherland J. & Thomas D., 2001a. Manifesto for agile software development. Saatavissa: <https://agilemanifesto.org/> [viitattu 30.3.2017].

Beck K., Beedle M., van Bennekum A, Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., Martin R. C., Mellor S., Schwaber K., Sutherland J. & Thomas D., 2001b. Principles behind the agile manifesto. Saatavissa: <http://agilemanifesto.org/principles.html> [viitattu 29.3.2017].

Chau T., & Maurer F., 2004. Knowledge sharing in agile software teams. Teoksessa: Lenski W. (toim.) Logic versus Approximation. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 173-183. ISBN 3-540-22562-5

Chow T. & Cao D., 2008. A survey study of critical success factors in agile software projects. The Journal of Systems and Software, 81 (6), S. 961-971.

Cline A., 2015. Agile development in the real world. New York: Apress, 297 s. ISBN 978-1-4842-1678-1

Cockburn A., 2006. Agile software development - The cooperative game. 2 painos. Boston, MA: Pearson Education, 467 s. ISBN 0-321-48275-1

Cockburn A. & Highsmith J., 2001. Agile software development, the people factor. Computer, 34 (11), S. 131-133.

Dickinson T. L. & McIntyre R. M., 1997. A conceptual framework for teamwork measurement. Teoksessa: Brannick M. T., Salas E. & Prince C. (toim.) Team performance assessment and measurement - Theory, methods, and applications. Mahwah, New Jersey: Lawrence Erlbaum Associates, S. 19-44. ISBN 1-4106-0205-2

Eunson B., 2007. Communication in the workplace. Milton, Queensland, Australia: John Wiley & Sons Australia, 148 s. ISBN 978-0-7314-0650-0

Goldie C. M. & Pinch R. G. E., 1991. Communication theory. Cambridge, United Kingdom: Cambridge University Press, 210 s. ISBN 978-0-521-40456-3

Hazzan O. & Dubinsky Y., 2014. Agile anywhere - Essays on agile projects and beyond. Cham: Springer, 72 s. ISBN 978-3-319-10157-6

Highsmith J., 2001. History: The agile manifesto. Saatavissa: <http://agilemanifesto.org/history.html> [viitattu 30.3.2017].

Highsmith J., 2002. Agile software development ecosystems. Boston, MA: Addison-Wesley, 404 s. ISBN 0-201-76043-6

Highsmith J., 2004. Agile project management - Creating innovative products. Boston, MA: Addison-Wesley, 277 s. ISBN 0-321-21977-5

IEEE Std 1058, 1998. IEEE Standard for Software Project Management Plans. New York: IEEE-SA Standards Board, 12 + 8 s.

Korkala M., Abrahamsson P. & Kyllönen P., 2006. A case study on the impact of customer communication on defects in agile software development. Teoksessa: Chao J., Cohn M., Maurer F., Sharp H. & Shore J. (toim.) Agile '06. New York: IEEE Computer Society, S. 76-88. ISBN 978-0-7695-2562-4

Korkala M. & Maurer F., 2014. Waste identification as the means for improving communication in globally distributed agile software development. The Journal of Systems and Software, 95, S. 122-140.

Martin A., Biddle R. & Nobble J., 2010. An Ideal Customer: A grounded theory of requirements elicitation, communication and acceptance on agile projects. Teoksessa: Dingsøy T., Dybå T. & Moe N. B. (toim.) Agile Software Development – Current Research and Future Directions. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 31-59. ISBN 978-3-642-12574-4

Martin R. C., 2003. Agile software development - Principles, patterns, and practices. Upper Saddle River, NJ: Pearson Education, 529 s. ISBN 0-13-597444-5

Medinilla A., 2012. Agile management - Leadership in an agile environment. Berlin, Heidelberg: Springer Berlin Heidelberg, 184 s. ISBN 978-3-642-28909-5

Moe N. B., Dingsøy T. & Dybå T., 2009. A teamwork model for understanding an agile team: A case study of a scrum project. Information and Software Technology, 52 (5), S. 480-491.

Parker G. M., 2009. Team communication - 20 essential aids. Amherst, MA: HRD Press, 43 s. ISBN 978-1-59996-194-1

Rico D. F., Sayani H. H., Sone S. & Sutherland, J. V., 2009. The business value of agile software methods - Maximizing ROI with just-in-time processes and documentation. Fort Lauderdale, FL: J. Ross Publishing, 241 s. ISBN 978-1-60427-031-0

Ruhe G. & Wohlin C., 2014. Software project management in a changing world. Berlin, Heidelberg: Springer Berlin Heidelberg, 477 s. ISBN 978-3-642-55035-5

Schwaber K., 2004. Agile project management with scrum. Redmond, WA: Microsoft Press, 163 s. ISBN 978-0-7356-3600-2

Sharp H. & Robinson H., 2010. Three 'C's of agile practice: Collaboration, co-ordination and communication. Teoksessa: Teoksessa: Dingsøy T., Dybå T. & Moe N. B. (toim.) Agile Software Development – Current Research and Future Directions. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 61-85. ISBN 978-3-642-12574-4

So C., 2010. Making software teams effective - How agile practices lead to project success through teamwork mechanisms. Frankfurt am Main: Peter Lang AG, 195 s. ISBN 978-3-653-00451-9

Stasser G. & Titus W., 1985. Pooling of unshared information in group decision making: Biased information sampling during discussion. *Journal of Personality and Social Psychology*, 48 (6), S. 1467-1478.

Subramaniam V. & Hunt A., 2006. Practices of an agile developer - Working in the real world. Raleigh, NC: Pragmatic Bookshelf, 189 s. ISBN 978-0-974-51408-6

Warkentin M. E., Sayeed L. & Hightower R., 1997. Virtual teams versus face-to-face teams: An exploratory study of a Web-based conference system. *Decision Sciences*, 28 (4), S. 975-996.

Wong Z., 2007. Human factors in project management - Concepts, tools, and techniques for inspiring teamwork and motivation. San Francisco: Jossey-Bass, 351 s. ISBN 978-0-7879-9629-1