



TAMPEREEN
AMMATTIKORKEAKOULU

KETTERÄ TOIMITUSSOPIMUS TOIMITTAJAN NÄKÖKULMASTA

Leena Lehtonen

Opinnäytetyö
Joulukuu 2015
Tietojenkäsittelyn koulutusohjelma



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma

LEHTONEN, LEENA:

Ketterä toimitussopimus toimittajan näkökulmasta

Opinnäytetyö 36 sivua
Joulukuu 2015

Ketterät menetelmät ovat vakiinnuttaneet paikkansa ohjelmistotoimituksissa, mutta niihin liittyvät toimitussopimukset eivät ole mukautuneet menetelmän vaatimuksiin samalla, kun menetelmien käyttö on yleistynyt.

Opinnäytetyön tarkoituksena oli pohtia, miten toimitussopimus tulisi laatia, että se tukisi menetelmää parhaalla mahdollisella tavalla, ja miten ketterä toimitussopimus eroaa vesiputousmalliin usein liitetystä kiinteähintaisesta sopimuksesta. Sopimuksellisen ongelman muodostaa toimitustapa, jossa toimituksen tarkkaa sisältöä ei tiedetä sopimuksen tekohetkellä. Ketteristä menetelmistä on paljon kirjallisuutta, mutta niihin liittyvistä sopimuksista on ainakin toistaiseksi kirjoitettu hyvin vähän, siksi työssä käytettiin aineistona sopimusten osalta kokemuksiin ja parhaisiin käytäntöihin perustuvia oppaita ja esimerkkejä.

Ketteriin toimituksiin suositellaan hinnoittelumalliksi tuntiveloitusta asiantuntijatyöstä, sillä tuntiveloitus on samalla tavalla joustava kuin ketterä toimitustapa. Projekti voidaan päättää aiottua aiemmin ilman, että siitä seuraa sopimuksellisia ongelmia. Osapuolten riskiä voidaan pienentää käyttämällä bonusmallia, jossa toimittaja saa paremman tuntihinnan, jos projekti valmistuu aiottua aiemmin, ja toimittaja sitoutuu pienempään tuntihintaan ylimenevältä ajalta, jos projektin tavoitteita ei saavuteta sovitussa ajassa. Ketterä toimitus vaatii tilaajan ja toimittajan välistä jatkuvaa yhteistyötä, mikä käytännössä tarkoittaa asioista sopimista projektin kuluessa, näin projektista tulee vesiputousmallin projektia läpinäkyvämpi eikä sopimukseen tarvitse kirjata toimituksen tarkkaa sisältöä, vaan päätökset sisällöstä tehdään projektin aikana. Ketterä toimitussopimus on sopimus yhteistyöstä ja siihen on tärkeää kirjata mukaan osapuolten vastuut sekä tiimin henkilöt ja heidän roolinsa.

Asiasanat: sopimukset, ketterät menetelmät, projektinhallinta

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Master's Degree Programme in Information System Competence

LEHTONEN, LEENA:
Agile Contract
Vendor's Point of View

Bachelor's thesis 36 pages
December 2015

The use of agile methods has been common and successful for years but still there are very few contractual frameworks available for agile projects.

The purpose of this thesis was to research what kind of contract would be the most suitable for agile projects, and to find out about the differences between agile contracts and traditional fixed-price contracts usually used in waterfall projects. The waterfall model assumes that the client is capable of writing specifications and describing the scope of the project at the beginning so clearly that there is no need to change it later. Agile framework is based on continuous cooperating between the client and the vendor and the scope of the project is not fixed. This feature is problematic from a contractual perspective.

Time and materials contracts are recommended for agile projects. There are different ways to use this contract type and this thesis describes how to apply it to agile methods. For example, paying a bonus to the vendor is one of the methods used when a project is finished before the estimated date of completion.

Key words: contracts, agreements, agile methods, project management

SISÄLLYS

1	JOHDANTO.....	7
2	KETTERIÄ PERIAATTEITA	8
2.1	Agile Manifesto	8
2.1.1	Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja	8
2.1.2	Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota	9
2.1.3	Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja.....	11
2.1.4	Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa.....	12
2.2	Pienin markkinakelpoinen tuote (MVP)	13
2.3	Työn alla olevien töiden määrä (WIP).....	15
2.4	Valmiin määritelmä (DoD)	15
3	KETTERIÄ MENETELMIÄ	16
3.1	Menetelmien esittely	16
3.1.1	Kanban	16
3.1.2	Scaled Agile Framework (SAFe).....	18
3.1.3	Scrum	18
4	KETTERÄT PERIAATTEET SOPIMUKSEN NÄKÖKULMASTA	20
4.1	Tilaajan ja toimittajan välinen yhteistyö ja luottamus	20
4.1.1	Tilaajan tuoteomistaja – liiketoiminnan edustaja.....	21
4.1.2	Kehittyvä tiimi – toimittajan edustus	22
4.1.3	Hyvä kehitysjonon hallinta	23
4.1.4	Pienin markkinakelpoinen tuote.....	23
4.1.5	Hyväksymismenettely	24
4.2	Projektikolmiosta ketterään kolmioon	24
4.2.1	Projektikolmio.....	24
4.2.2	Ketterä kolmio.....	26
5	SOPIMUS JA HINNOITTELU	29
5.1	Puitesopimus ja projektikohtainen sopiminen	29
5.1.1	Yhteistyö, työskentelymalli ja vastuut	29
5.1.2	Immateriaaliset oikeudet työn tuloksiin	30
5.1.3	Tilaajan oikeus vaihtaa henkilöitä ja toimittajaa.....	30
5.2	Hinnoittelu	31
5.2.1	Bonukseen perustuva hinnoittelumalli.....	32
5.2.2	Hybridimalli	32
5.2.3	Keskituntiveloitus asiantuntijatyöstä	33

6 POHDINTA.....	34
LÄHTEET.....	35

ERITYISSANASTO

aika ja materiaalisopimus	asiantuntijatyön hinnoittelumalli, jossa asiantuntijatyön hinta veloitetaan tehtyjen tuntien mukaan, <i>time & material</i>
arviointipiste	työmäärän suhteelliseen kokoon perustuva arvioinnin yksikkö, joka ei perustu todelliseen tuntimäärään, <i>story point</i>
DoD	valmiin määritelmä, <i>definition of done</i>
hukka	työvaihe, joka ei tuota arvoa, <i>waste</i>
iteratiivinen	prosessia toistava työtapa, <i>iterative</i>
Kanban	Lean-periaatteisiin pohjautuva, töiden läpimenon parantamiseen keskittyvä ketterä menetelmä
kehitysjono	kehitystehtävät sisältävä jono, jota hoidetaan jatkuvalla priorisoinnilla, <i>backlog</i>
kiinteähintainen sopimus	vaatimusmäärittelyyn perustuva hinnoittelumalli, jossa koko toimituksen hinta sovitaan etukäteen tehdyn määrittelyn pohjalta, <i>fixed price</i>
Lean	Japanista Toyotalta peräisin oleva johtamisfilosofia, johon ICT-alalla tunnetut ketterät menetelmät perustuvat
MVP	pienin markkinakelpoinen tuote, <i>minimum viable product</i>
SAFe	jatkuvan ketterän kehityksen malli yritystasolle, <i>Scaled Agile Framework</i>
sprintti	kiinteän mittainen tuotteen tai palvelun kehitysjakso, jonka aikana tuotetaan uusi tuotantovalmis versio, <i>sprint</i>
WIP	työn alla olevien töiden määrä, <i>work in progress</i>
WIP-raja	työn alla olevien töiden lukumääräraja, <i>WIP limit</i>

1 JOHDANTO

Opinnäytetyöni tavoite ja toimeksianto on pohtia ketterän ohjelmistotoimituksen sopimistapoja ostajan ja toimittajan välillä, ja lisätä tietämystä aiheesta toimeksiantajani organisaatiossa. Tarkoitus on tutkia sopimiseen liittyviä eroja, kun siirrytään vesiputousmallin projekteista ketteriin toimituksiin. Työssäni pyrin tarkastelemaan asiaa toimittajan näkökulmasta, mutta koska sopimus on kahden osapuolen välinen, tavoitteeni on tuottaa tietoa yhtä lailla myös tilaajan edustajille.

Työni toimeksiantaja on suomalainen ICT-toimittaja, joka toimittaa ohjelmistopalveluja Suomessa finanssialalla toimiville asiakkailleen. Työssäni käytän näistä yrityksistä termejä tilaaja ja toimittaja.

Ketteristä toimituksista on paljon kokemuksia Suomessakin, mutta tilaajan ja toimittajan väliset sopimukset koetaan edelleen hankaliksi. Lause *“I’m agile, but my contract isn’t”* kuvaa hyvin opinnäytetyössäni tutkittavan ongelman: miten tehdä sopimus, joka tukisi ketterää toimintamallia. Haasteen muodostaa ominaisuus, jossa toimituksen tarkkaa sisältöä ei tiedetä vielä sopimusvaiheessa.

Tilaajan sopimusmalli tehdään useimmiten lakimiesten avustuksella ja monesti ohjelmistotalan ketteriä toimintatapoja ei tunneta riittävän hyvin, jotta laadittava sopimus tarjoaisi saman jouston kuin menetelmä. Ongelma on yleisesti tunnettu eikä rajoitu vain työni toimeksiantajaan ja sen asiakassuhteisiin. Suomessa ensimmäinen versio ICT-alan yleisistä sopimusehdoista joissa ketteruus on mukana, on julkaistu syksyllä 2015 IT2015 lisäehtojen muodossa. Puitesopimusten laajempi tarkastelu on rajattu pois työstäni ja keskityn vain projektikohtaiseen sopimiseen.

2 KETTERIÄ PERIAATTEITA

2.1 Agile Manifesto

Yhdysvalloissa vuonna 2001 julkaistu Agile Manifesto, jota pidetään ketterän kehityksen alkulähteenä, ei ole vanhentunut vuosien kuluessa. Päinvastoin se kuvaa edelleen ytimekkäästi keskeisimmät ketterän kehityksen periaatteet.

Seitsemäntoista kokenutta ohjelmistokehityksen asiantuntijaa julkaisivat manifestin, jonka mukaan he kertoivat arvostavansa

Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja
Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota
Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja
Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa
(Agile Manifesto 2001)

Manifesti ei kumoa jäljessä tulevien asioiden arvoa vaan painottaa erityisesti ensimmäisiä. Näiden periaatteiden sisältämän sanoman ymmärtäminen on mielestäni keskeistä myös toimitussopimusta solmittaessa, sillä periaatteet kuvaavat tapaa tehdä työtä. Työskentelymalli, jossa lopputulosta ei sopimusvaiheessa vielä tiedetä, johtaa siihen että kiinteähintainen sopimus ei ole mahdollinen. Toisaalta kun jatkuva yhteistyö ja työn tulosten läpinäkyvyys lisääntyvät tilaajan ja toimittajan välillä, tarve oman selustan varmistamiseen seikkaperäisillä sopimusehdoilla vähenee.

Jim Highsmith on yksi Agile Manifeston allekirjoittajista, hän ilmaisee ajatuksensa suhteellisen voimakkaasti kirjassaan Agile Project Management. Olen poiminut enimmäkseen juuri Highsmithin (2007) nasevia ajatuksia seuraaviin kappaleisiin, joissa käyn tarkemmin läpi Agile Manifeston periaatteita.

2.1.1 Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja

Työkalut lisäävät tehokkuutta, mutta ilman osaavia ihmisiä, joilla on tilanteeseen nähden oikea tekninen ammattitaito sekä hyvät vuorovaikutustaidot, ei synny tuloksia. Hyvän prosessin tulee nimenomaan tukea tiimiä eikä sanella sen toimintaa. Ketterä työskentely perustuu itseorganisoituvaan tiimiin, tiimin jäsenten itsekuriin, yksilöiden ja

heidän kyvykkyytensä arvostamiseen sekä tasa-arvon pyrkimykseen. Ketteryys on sosiaalista liikettä, jossa voidaan tuntea iloa innovatiivisten tuotteiden luomisesta asiakkaille. (Highsmith 2007, 13-14.)

Highsmith (2007) viittaa kirjassaan siihen, että yrityksissä saatetaan ylistäen kehua kuinka arvokkaita työntekijät yritykselle ovat. Käytännössä kuitenkin työntekijöiden toimintaa rajoitetaan taipumattomilla menettelytavoilla, jolloin prosessista tulee helposti ihmistä tärkeämpi. Perfektionistinen tapa soveltaa menetelmiä käytäntöön johtaa usein kankeuteen ja tehottomuuteen. Ratkaisevaa tiimin työn onnistumiselle onkin ihmisten osaamiseen perustuvat päätökset ja taito soveltaa malleja tilanteeseen sopivalla tavalla eikä niinkään niiden kirjaimellinen noudattaminen.

Kaikki ketterät menetelmät korostavat moniosaajatiimin merkitystä. Tiimit kootaan eri osa-alueiden ammattilaisista ja samalla usein myös eri yritysten edustajista. Ohjelmistokehitystiimissä tarvitaan kehittäjien ja testaajien lisäksi aina myös liiketoiminnan osaaja. Tiimillä on vastuu sekä tehtävästä työstä että toimintatavoista ja sen tulisi huomata toimimattomat työskentelytapansa itse ja pystyä muuttamaan niitä. Työntekijöiden jatkuva oppiminen ja kehittyminen yhteistyön edetessä kuuluvat myös olennaisesti ketterään toimintatapaan.

2.1.2 Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota

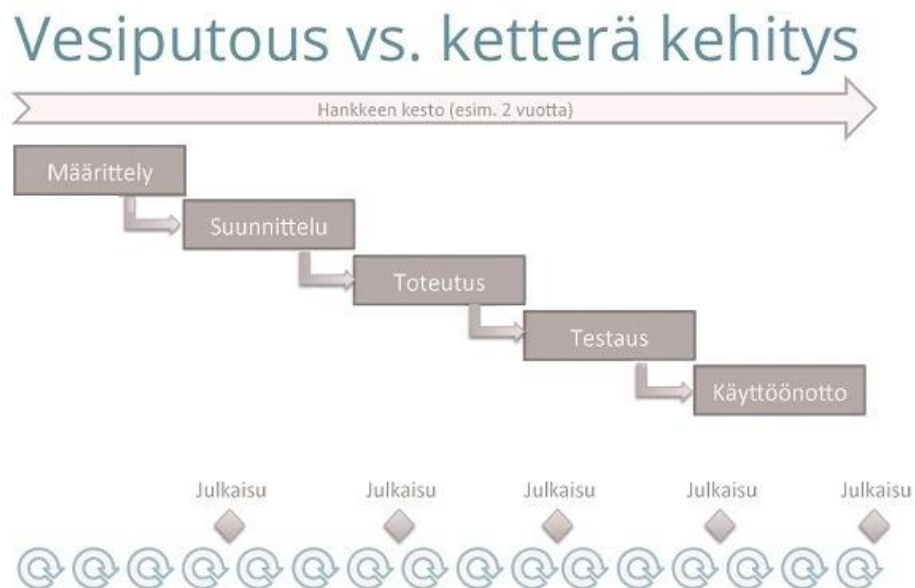
Tilanteessa, jossa asiakas ja tuotepäällikkö kirjoittavat vaatimusdokumentin ja lähettävät sen kehitystiimille, on dokumentista tullut vuorovaikutuksen korvike. Sen sijaan kun asiakas ja kehittäjä yhdessä tekevät määrittelyä ja tuottavat joitakin pysyviä kuvauksia (luonnoksia, piirroksia, muistiinpanoja, käyttäjätarinoita) on dokumentaatio syntynyt vuorovaikutuksen sivutuotteena. Julistus ei kiistä dokumentaation merkitystä, mutta laittaa painopisteen toimivalle ohjelmistolle. (Highsmith 2007, 12.)

Highsmith (2007, 12) mainitsee lyhyesti, että se mikä toimii tuotannossa, on sovellus eikä dokumentti. Olen itse pohtinut tätä perinteisen määrittely- ja toteutusvaiheen erona siinä mielessä, että valmiina pidetty määrittely antaa vielä anteeksi unohtuneita tai epäselviä asioita. Sen sijaan valmis sovellus on aina konkreettinen eikä anna anteeksi oh-

jelmonti- eikä suunnitteluvirheitä. Ketterissä menetelmissä tähän yhdistyy lisäksi valmiin tuotteen loppuasiakkaalle tuottama arvo.

Vesiputousmallissa projektin katsotaan onnistuneen, jos se on toteutunut projektisuunnitelman mukaisesti. Jos tavoitteena kuitenkin on ollut ensisijaisesti tuote, jonka asiakkaat haluavat, tavoite ja projektisuunnitelmaan kirjatut ominaisuudet eivät välttämättä vastaa toisiaan enää tuotteen valmistumishetkellä. (Teikari 2012, 19.)

Vesiputousmallia (*waterfall model*) pidetään ensimmäisenä varsinaisena ohjelmistokehityksen prosessimallina. Se jakaa prosessin lineaarisiin vaiheisiin, jossa edellisen vaiheen tulos on seuraavan vaiheen syöte. Malli perustuu hyvään dokumentaatioon. Nykyisin käytössä olevat ja hyvin tunnetut vaiheet ovat: määrittely, suunnittelu, toteutus, testaus ja käyttöönotto. Kuten alla olevasta kuvasta (kuva 1) näkyy, vesiputousmallilla tehdyn toimituksen kesto saattaa olla hyvinkin pitkä, kun taas ketterillä menetelmillä tuotetaan valmiita osia tasaisesti.



KUVA 1. Lineaarinen vesiputousmalli ja iteratiivinen ketteruus (Sami Lempinen, Celkee Oy)

Innovaatioita haikailevissa yrityksissä olisi hyvä huomata, että iteratiivinen toimintatapa, jota ketterät menetelmät edustavat, tukee myös innovaatioiden syntymistä. Vesiputousmallille tyypillinen, toteutusta edeltävä pitkä määrittelyvaihe saattaa tuottaa hyviä

ideoita, mutta koska niitä ei päästä testaamaan käytännössä eikä niiden toimivuudesta siten saada palautetta loppuasiakkailta, ideoiden todellista arvoa ei tiedetä ennen kuin koko ohjelmisto on valmis. (Highsmith 2007, 12.)

Itse ajattelen, että dokumentaatio, joka palvelee selkeästi jatkokehitystä ja sovelluksen ylläpitoa on tarpeellista tuottaa. Projektin toteuttamisen aikainen dokumentaatio ei välttämättä palvele jatkokehitystä eikä sille sen takia tarvitse asettaa suuria laatuvaatimuksia.

2.1.3 Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja

Projektitiimin päämäärä on tuottaa arvoa loppuasiakkaille. Loppuasiakas voi olla yksilö tai ryhmä, joka käyttää kehitettävää sovellusta tuottaakseen liikevoittoa työnantajalleen tai loppuasiakas voi olla sovellusta käyttävä yksityishenkilö. Tuotteen arvon ja sitä kautta hankkeen onnistumisen määrittää lopulta nimenomaan loppuasiakas. Siksi tilaajan ja toimittajan välistä yhteistyötä ei saisi varjostaa sopimukselliset erimielisyydet vaan kehitystiimissä tilaajan edustajalla on tärkeä rooli koko kehitystyön ajan. (Highsmith 2007, 13.)

Tuotteen kehittymiselle annetaan vapaus muotoutua työn aikana sellaiseksi, että se parhaiten vastaa asiakkaan tarvetta. Tarkkaa yksityiskohtaista määrittelyä ei tarvita, koska yksityiskohtia hiotaan jatkuvassa yhteistyössä tilaajan ja toimittajan edustajien kesken. Tämän vuoksi asiakasyhteistyö on suuressa roolissa. Jatkuva yhteistyö myös parantaa olennaisesti tilaajan mahdollisuutta seurata toimittajan työskentelyä.

Loppujen lopuksi onnistunut projekti ei synny sopimuksesta vaan ihmisten välisestä hyvästä yhteistyöstä, läpinäkyvyydestä ja luottamuksesta. Sopimusneuvottelu tässä yhteydessä viittaa laillisen sopimuksen lisäksi myös osapuolten sopimukseen jatkuvasta yhteistyöstä, oppimisesta ja kehittymisestä. Tämä kohta ymmärretään usein väärin siten, että se koskisi vain laillista sopimusta. (Arbogast, Larman & Vodde, 2012, 7.)

2.1.4 Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa

Kaikkiin projekteihin liittyy riskejä ja epävarmuustekijöitä, joita ei voi etukäteen hallita. Ketterässä projektissa haetaan tasapainoa, kuinka paljon projekti kestää tätä epävarmuutta. Tutkiva tyyli, jossa kokeillaan erilaisia vaihtoehtoja, on kustannuksiltaan kallis verrattuna mukautuvaan tyyliin, jossa suunnitelmia ja mahdollisesti myös arkkitehtuuria kehitetään jatkuvasti työn edetessä. Kumpikaan tyyli ei silti ole automaattisesti toistaan parempi vaan tilanne ratkaisee, miten projektissa kannattaa edetä. Kaikissa tilanteissa visio, jota kohti pyritään, pitää olla selkeä. (Highsmith 2007, 11.)

Omaan sovelluskehitystaustaan perustuen ymmärrän hyvin pyrkimyksen pois vesiputousmallin tarkasta etukäteen tehdystä määrittelystä. Muistan hyvin turhautumisen tunteen määriteltäessä tarkkoja yksityiskohtia laajaan kokonaisuuteen vaiheessa, jossa en hahmottanut vielä kunnolla mitä olimme tekemässä. Tämä voi tietysti olla henkilökohtainen ominaisuus, osa ihmisistä lähtee mieluummin liikkeelle yksityiskohdista. Joka tapauksessa toteutuksen yhteydessä, kun työn tulokset alkavat olla konkreettisia, on huomattavasti helpompi havaita, minkälaisia ratkaisuja tarvitaan ja mitä mahdollisesti vielä puuttuu. Ketterien menetelmien lyhyet kehitysjaksot tukevat tätä hyvin, sillä sovelluksen osien valmistuessa tiimin jäsenten ymmärrys ja osaaminen kasvavat ja tiimi käyttää saamansa kokemuksen hyväkseen saman tien.

Myös tilaajan puolella on usein sama ongelma, että ihmisten on vaikea hahmottaa kokonaisuutta siinä vaiheessa, kun vasta suunnitellaan ja ideoidaan ratkaisua heidän liiketoiminnalliseen ongelmaansa tietojärjestelmän avulla. Seuraavalla sivulla olevassa kuvassa on käytetty visuaalisuutta web-suunnittelussa (kuva 2). Sovelluksen ulkoasua voidaan lähteä kuvaamaan myös tarkoitukseen sopivalla kehitysvälineellä, jolloin saadaan nopeasti näkyviä tuloksia ja niihin perustuvaa palautetta asiakkaalta. Molemmissa tilanteissa suunnittelua ja päätöksiä tehdään työn edetessä eikä pitäydytä aiemmin laaditussa valmiissa suunnitelmassa.



KUVA 2. Visuaalinen web-suunnittelu (Taali 2013)

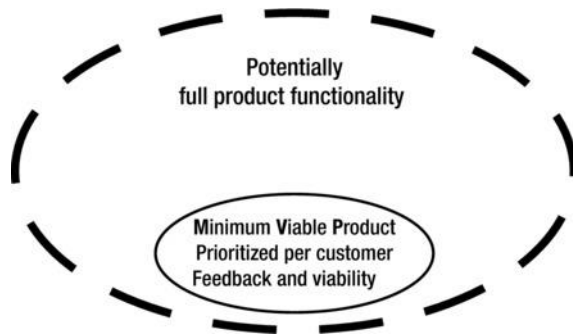
2.2 Pienin markkinakelpoinen tuote (MVP)

Tähän ja kahteen seuraavaan lukuun olen poiminut muutaman käsitteen, jotka liittyvät olennaisena osana ketteriin viitekehyksiin. Viitekehysellä tarkoitan tunnettuja ketteriä menetelmiä kuten Scrum tai SAFe. Nämä viitekehykset sisältävät lukuisan joukon menetelmiä ja prosesseja, joita projekteissa voidaan käyttää.

Ketterissä hankkeissa suositellaan liikkeelle lähtemistä ominaisuuksiltaan niin karsituilla ohjelmistoversiolla kuin mahdollista. Englanninkielinen termi Minimum Viable Product (MVP) on käytetympi kuin sen suomenkielinen vastine, pienin markkinakelpoinen tuote, joka sekin on hyvin kuvaava. Käyttöönoton jälkeen tuotteen kehittämistä jatketaan iteroiden jatkuvasti asiakaspalautteen ohjaamaan suuntaan. (Teikari 2012, 27.)

Toisin sanoen käsitteellä MVP tarkoitetaan mahdollisimman yksinkertaista sovelluksen versiota, jonka käyttämisellä kuitenkin on loppukäyttäjän kannalta jo tietty tarkoitus. Visio sisältää ominaisuuksia, joita sovellukseen voidaan lisätä ja kehittää myöhemmin. Asia tulee mielestäni hyvin esiin seuraavan sivun kuvasta (kuva 3).

Tuotteen kehittäminen MVP:n pohjalta perustuu mielellään pienen, mutta visionäärisen asiakasjoukon palautteeseen. Palautteen perusteella arvioidaan, onko tuote niin kysytty ja haluttu kuin odotettiin. Asiakaspalautetta tarvitaan vain sen verran, että saadaan selville ollaanko tuotteen kehittämisessä oikealla tiellä. Kokeilut ja epäonnistumiset ovat sallittuja, mutta niiden kustannukset halutaan pitää mahdollisimman pieninä. (Ries 2009.)



KUVA 3. Pienin markkinakelpoinen tuote (Moreira 2013)

Loppujen lopuksi useimmilla sovelluksilla pyritään tuottamaan vastinetta sijoitetulle pääomalle ja MVP:n toteuttaminen mahdollistaa rahallisen tuoton sekä muut odotetut hyödyt jo aikaisessa vaiheessa. Useat käyttöönotot ja mahdolliset suunnanmuutokset aiheuttavat kustannuksia, mutta vastaavasti niiden avulla on mahdollista välttää suuremmat riskit.

MVP:n pitäisi erottua kilpailijoista ja olla käyttäjilleen hyödyllinen. Sen toiminnallisuus voidaan ottaa tarjouspyynnön pohjaksi, mutta visio on kuitenkin listattuja ominaisuuksia tärkeämpi. MVP:n pitäisi ketterien periaatteiden mukaan sisältää projektin riskipitoisimmat osat, tällä tarkoitetaan, että toteutukseen valittavia ominaisuuksia arvioidaan arvon ja riskin suhteen perusteella. Ensimmäiseksi toteutettavien osuuksien listalle priorisoidaan ominaisuus, jolla on sekä suuri arvo että suuri riski, mikäli tällainen tehtävä tunnustetaan. Menestystä on odotettavissa silloin, kun tilaaja kertoo toimittajalle mikä tekee tulevasta tuotteesta voittajan, koska tällöin toiminta lähtee liikkeelle loppuasiakkaalle tuotettavasta arvosta. (Järvenpää & Kovanen, 10-12.)

2.3 Työn alla olevien töiden määrä (WIP)

Kanban, kuten muutkin Lean-ajatteluun perustuvat ketterät menetelmät, esittelee käsitteen WIP, työn alla olevien töiden määrä (*work in progress*). Kerrallaan työn alla olevien töiden määrän rajoittamisella pyritään vahvistamaan töiden tehokasta virtausta vaiheesta toiseen, esimerkiksi toteutuksesta testaukseen ja lopulta valmistumista asiakkaalle arvoa tuottavaksi. Tiimin tulee pyrkiä löytämään itselleen optimaalinen WIP-raja, jota enempiä töitä ei kannata ottaa yhteen sprinttiin. Rajalla pyritään varmistamaan ennen kaikkea töiden valmistuminen kehitysjakson aikana. (Stellman & Greene 2014.)

Tärkeä periaate, joka nykyisessä työkuultuurissa ei välttämättä ole itsestään selvä, vaan hyvinkin tavallista on, että henkilö vaihtaa jatkuvasti tehtävästä toiseen ja yrittää edistää useita asioita rinnakkain. Jatkuva työn keskeytyminen kuitenkin tosiasiasa vie aikaa ja vähentää työntekijän tehokkuutta.

2.4 Valmiin määritelmä (DoD)

Yksi hyvistä peruseriaateista on määritellä, milloin jokin asia on valmis. Jos näin ei tehdä, eri osapuolille syntyy helposti erilaisia oletuksia siitä, mitä valmis tarkoittaa. Valmiin määritelmästä käytetään kirjainyhdistelmää DoD (*definition of done*).

Valmiin määritelmä voidaan tehdä erikseen eri osille ja vaiheille esimerkiksi siten, että käyttäjätarinoille (*user story*), kirjoitetaan hyväksymisehdot, jotka tulee täyttää ennen kuin tarinan voidaan todeta olevan valmis toteutettavaksi, testattavaksi tai hyväksyttäväksi. Käyttäjätarinat ovat dokumentaatiota sovelluksen toiminnasta ja arvosta, jota niiden toiminta tuottaa. (SAFe 2015a.)

Seuraavassa luvussa oleva kuva Kanban-työkalusta (kuva 4) havainnollistaa myös valmiin määritelmää, koska siinä yksittäinen työvaihe on jaettu osiin Työn alla (*In Progress*) ja Tehty (*Done*). Kun työn tila suunnittelu- ja analysointivaiheessa on Tehty, työ voidaan siirtää toteutusvaiheeseen. Tehty-tilaan siirtäminen tarkoittaa, että suunnittelu- ja analysointivaiheen hyväksymisehdot on täytetty. Hyväksymisehdot on hyvä kirjoittaa yhdessä liiketoiminnan edustajan kanssa.

3 KETTERIÄ MENETELMIÄ

3.1 Menetelmien esittely

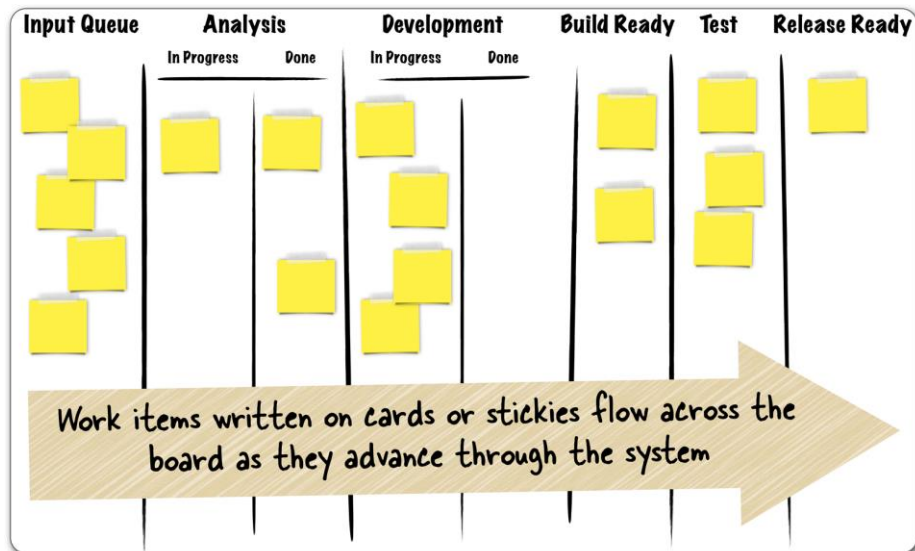
Esittelen tässä luvussa lyhyesti ketteristä menetelmistä ne, joihin työssäni viitataan: Kanban, Scaled Agile Framework ja Scrum. Koska työni tarkastelunäkökulma liittyy sopimuksiin, en kuvaa menetelmiä tarkasti. Kaikki kolme menetelmää ovat keskenään erilaisia ja eri tarkoitukseen syntyneitä, mutta kaikkien taustalta löytyy alun perin Japanista Toyotan tehtailta peräisin oleva Lean-ajattelu. Lean on johtamismenetelmä, joka kasvattaa asiakkaalle tuotettavaa arvoa eliminoimalla hukkaa (*waste*) (Smolander, Maglyas & Nikula 2012, 41).

Hukalla tarkoitetaan esimerkiksi työvaiheita, jotka eivät tuota mitään erityistä arvoa loppuasiakkaalle. Tätä voidaan ajatella myös niin, että ketterissä projekteissa tekemättä jätettävän työn maksimointi on olennaista. Menetelmissä pyritään siihen, että lopputuoteseen ei tuoteta ominaisuuksia, joilla on vain vähän käyttöä.

3.1.1 Kanban

Kanban on erityisesti töiden läpimenon parantamiseen keskittyvä menetelmä, joka myös visualisoi tiimin työtilanteen havainnollisen Kanban-työtaulun avulla. Kanban-työtaulun käyttö lisää projektin läpinäkyvyyttä. Jokainen tiimin työ sijoitetaan työtaululle, josta yhdellä vilkaisulla voidaan havaita, missä vaiheessa mikin työ on. Yksinkertaisimmillaan Kanban-työtaulu sisältää sarakkeet Aloittamatta, Työn alla ja Tehty. Yksittäinen tehtävä siirretään työtaululla vaiheesta toiseen sen mukaan, kun tiimin jäsen ottaa sen työstettäväkseen. Työtaulun avulla tiimillä on mahdollisuus analysoida töidensä kokonaistilannetta ja tunnistaa ongelmat töiden virtauksessa vaiheesta toiseen. (Stellman & Greene 2014.)

Kanbania käytetään usein yhdessä jonkin toisen viitekehyksen kanssa. Scrumin ja Kanbanin yhdistelmästä on kehitetty oma menetelmänsä, joka on nimeltään Scumban. Kanban soveltuu myös hyvin järjestelmän ylläpitoon liittyvien töiden hallintaan, eikä ole siten pelkästään projektien työväline.



KUVA 4. Kanban-taulu (Stellman & Greene 2014)

Yksilötasolla pyrkimys tehostaa töiden läpimenoa tarkoittaa yhden tehtävän hoitamista seuraavaan vaiheeseen ennen uuden tehtävän aloittamista. Tunnettu Kanbanin lausahdus onkin 'Lopeta aloittaminen, aloita lopettaminen'. Jos tiimin jäsen yrittää edistää useita tehtäviä rinnakkain, hän todennäköisesti vaihtaa työtehtävää useita kertoja päivittäin, vaihtamiseen kuuluva aika on ketterien menetelmien näkökulmasta hukkaa, sillä tehtävän vaihtoon saattaa kulua jopa enemmän aikaa kuin tehtävän tekemiseen kuluisi.

Kanban-taulun muodostamiseen on olemassa valmiita ohjelmistoja, esimerkiksi JIRA sisältää oman JIRA Kanban board -nimellä tunnetun taulun. Taulu voidaan yhtä hyvin piirtää projektihuoneen seinälle käsin, mikäli koko tiimi työskentelee samassa toimipisteessä. Tiimien hajauttamien eri puolille maailmaa ja etätöet ovat lisääntyneet niin voimakkaasti, että monissa tiimeissä tarvitaan kuitenkin tekninen työväline, että taulu palvelisi kaikkia tiimin jäseniä jatkuvasti ja ajantasaisesti.

Kanban-menetelmään liittyy muutamia sääntöjä, jotka tähtäävät yksittäisen tehtävän läpimenon optimoimiseen prosessissa, aiemmin esitelty WIP-raja on yksi niistä. Rajalla varmistetaan sitä, että jokainen työ tulee tehtyä loppuun asti eikä tiimin jäsen pelkästään aloita tehtäviä, jotka sitten jäävät keskeneräisiksi. Rajan avulla työt saadaan virtaamaan prosessissa eteenpäin, muuten taulu täyttyy keskeneräisistä töistä.

3.1.2 Scaled Agile Framework (SAFe)

Scaled Agile Framework tähtää pidempi aikaisen kuin vain yhden projektin mittaisen kehitysorganisaation rakentamiseen. Kyseinen yritystasolle suunniteltu ketterä viitekehys onkin liian raskas otettavaksi käyttöön vain yhden projektin tarkoituksiin. Samoja ajatuksia näyttää esiintyvän laajemminkin, sillä on havaittu, että yksittäisen projektin käynnistämiseen ja resursointiin kuuluu paljon aikaa verrattuna tilanteeseen, jossa osaa-va ja valmis kehitystiimi tuottaa uusia ominaisuuksia säännöllisin väliajoin.

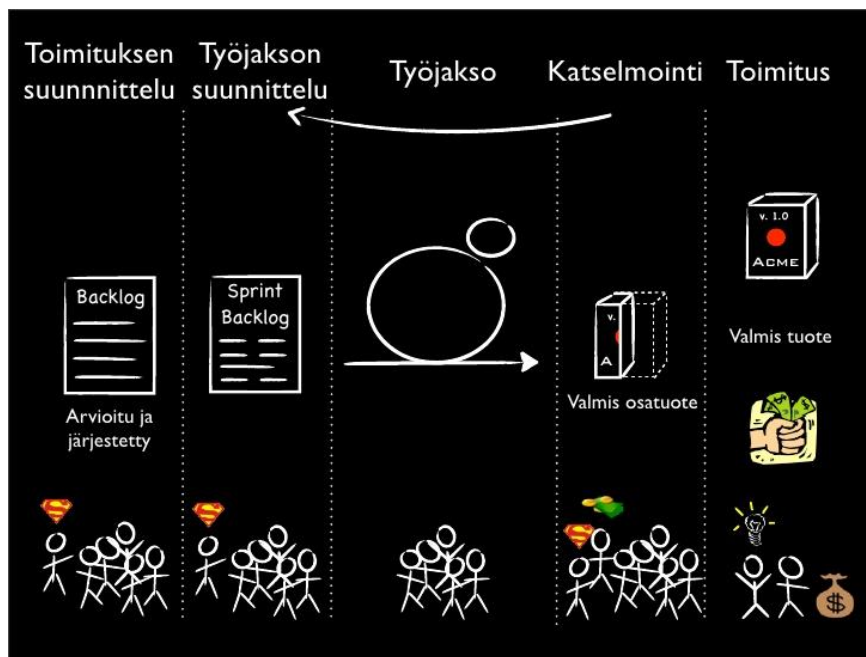
Monissa suurissa ja keskisuurissa yrityksissä ohjelmistoilla on niin paljon riippuvuuksia toisiinsa, että yksittäinen ketterä projekti ei ole mahdollinen, mikäli näitä riippuvuuksia ei pystytä hoitamaan varsinaisen ketterän kehitysprojektin kanssa samassa tahdissa. Yritystasolle skaalautuvat mallit kuten SAFe ovat lähteneet liikkeelle tämän ongelman ratkaisemisesta. SAFe-mallia kehitettiin aikanaan aktiivisesti Suomessa Nokia Mobile Phones -yhtiössä Dean Leffingwellin johdolla, Leffingwell on SAFe-mallin luoja. Nokia Networks:n puolella kehitettiin samoja ongelmia ratkova LeSS-niminen malli. (Auer ym. 2013, 26; SAFe 2015a; Tikka & Nyman 2015.)

SAFEn sivuston mukaan Nordean ensimmäisissä kokemuksissa Scaled Agile Frameworkin käytöstä Tukholman toimipisteessä 2014 - 2015 korostuu nimenomaan yhteistyön lisääntyminen yli tiimirajojen ja sen merkitys työtehon parantumisessa. Kuten edellä on mainittu, SAFe tarjoaa ratkaisuja ohjelmistojen riippuvuuksien hallitsemiseen. Jotta tiimit eivät kasvaisi liian suuriksi, ohjelmiston eri osuuksia on käytännössä usein tehtävä lukuisissa eri tiimeissä. Tiedonkulun parantuminen tiimien välillä yhteisten kokousten ja tuoteomistajien avulla koettiin Nordeassa erityisen onnistuneeksi. Samoin Tukholmassa oltiin tyytyväisiä mallin käyttöönottamiseen projektin johtamisessa, sillä tiimitason toiminta ei yksin riitä vaan kulttuurin muutos koskettaa yhtä lailla projektin johtoa. Muun muassa kehitysjonon jatkuva aktiivinen hoitaminen asettaa velvoitteita ICT:n johtamiselle. (SAFe, 2015b.)

3.1.3 Scrum

Scrum on tunnetuin ketterä menetelmä, se on viitekehys, jossa työ jaksotetaan 2-4 viikon pituisiin jaksoihin (*Sprint*). Jokaisen jakson aikana valmistuu tuotantokelpoinen

osatuote. Scrum sisältää paljon sääntöjä, jonka takia kehystä on alettu myös kritisoida. Alla olevassa kuvassa (kuva 5) havainnollistetaan työn jaksottaista etenemistä scrum-tyyliin. Kuvassa näkyvät henkilöt ovat tuoteomistaja ja muu tiimi, katselmointiin osallistuu lisäksi tiimin ulkopuolinen liiketoiminnan edustaja ja viimeisessä kohdassa tilaaja ja ideoija saavat idealleen tuoton. Myös muissa ketterissä menetelmissä on otettu scrum-sanasto käyttöön, esimerkiksi scrummasterin ja tuoteomistajan rooleja käytetään yleisesti ketterissä viitekehyksissä.



KUVA 5. Scrum-projektin eteneminen (Tanninen & Taipale 2009)

Kuvassa koko palkkio maksetaan toimittajalle vasta sitten, kun tuote on kokonaan valmis. Käytännössä kuitenkin suosittu tapa on laskuttaa jokaisen sprintin päätteeksi, kun osatoimitus on hyväksytty. Yksinkertaisessa hinnoittelumallissa todellinen iteraation toimituksen hinta veloitetaan ja maksetaan. Monimutkaisemmat mallit, kuten luvussa 5.2.1 esitelty bonukseen perustuva hinnoittelumalli, voivat sisältää ehtoja, jolloin onnistuneesta projektista maksettava bonus maksetaan vasta lopuksi. (Arbogast, Larman & Vodde 2012, 25.)

4 KETTERÄT PERIAATTEET SOPIMUKSEN NÄKÖKULMASTA

4.1 Tilaajan ja toimittajan välinen yhteistyö ja luottamus

Elisa Rahoitus Oy:n toimitusjohtaja Janne Niemensivu toteaa Houston Inc:n Ketterän kehityksen ostajan oppaassa, että ”Perinteinen tapa ostaa ohjelmistoprojekteja on sellainen, jossa halutaan ostaa määritelty ominaisuus kiinteällä hinnalla”. Niemensivu kertoo lisäksi, että perinteinen tarkkaan määrittelyyn perustuva malli ei toimi hyvin ostajan näkökulmasta. Vaikka pyydettyjen tarjousten oletetaan olevan vertailukelpoisia, todellisuudessa ne eivät yleensä ole, vaan määrittelyjä tulkitaan toimittajien keskuudessa eri tavoin. Samasta syystä johtuen tarjouskilpailua ei koeta reiluksi myöskään toimittajien puolella, koska myös he uskovat tarjousten sisällössä olevan eroja. (Teikari 2012, 20.)

Niemensivun mukaan ostajan kannattaakin myöntää, että täydellistä määrittelyä on mahdoton tehdä, eivätkä toimittajat pysty lukemaan rivien välistä, mitä ostaja todellisuudessa haluaa. Niemensivu kertoo, että ketterät hankkeet ovat onnistuneet Elisa Rahoituksessa perinteisiä paremmin, koska projektitoimitus on tilaajalle läpinäkyvämpi. Perinteisen mallin toimituksissa ongelmat paljastuvat vasta testausvaiheessa, kun koko toteutus on jo tehty. Kun kysymyksessä on monimutkainen ohjelmisto, pitäisi ostajan ymmärtää että alkuvaiheessa ei tiedetä mitä lopulta halutaan. Ketterissä menetelmissä tämä on nimenomaan projektitoimituksen luonnollinen lähtökohta. (Teikari 2012, 21.)

Ketterän toimituksen sopimusta solmittaessa kummallakaan osapuolella ei ole tarkasti selvillä mikä tulee olemaan toimituksen sisältö kokonaisuutena ja sanotaankin, että ketterän sopimuksen tulisi perustua tilaajan ja toimittajan väliseen luottamukseen. Luottamus on kuitenkin sopimuksen kannalta abstrakti käsite ja siksi hankala kirjata (Auer ym. 2013, 31). Luottamuksen tulee olla molemmin puolista, toimittajan on luotettava tilaajaan ja tilaajan toimittajaan. On kuitenkin ymmärrettävää, että tilaajan kannalta olennainen kysymys on edelleen, kuinka paljon projekti tulee maksamaan. Projektin hinnoittelua käsittelemme tarkemmin luvussa 5. Seuraavissa luvuissa esittelen Suomessakin hyväksi havaittuja keinoja luottamuspulan helpottamiseen.

4.1.1 Tilaaajan tuoteomistaja – liiketoiminnan edustaja

”Ketterän ohjelmistokehitysprojektin ostaminen vaatii enemmän sitoutumista verrattuna perinteiseen vesiputousmalliseen projektiin. Ostajan täytyy varata resursseja projektin käyttöön.” (Teikari 2012, 34). Käytännössä hyväksi havaittu malli on tuoteomistajan (*Product owner*) roolin hoitaminen tilaaajan puolelta. Ketterässä projektissa tuoteomistaja priorisoi tiimin työt, tuo liiketoiminnan terveiset tiimille, ja on samalla tilaaajan edun valvoja sekä hyväksyy tai hylkää valmiit ominaisuudet. Tuoteomistajan lisäksi projektiin saatetaan tarvita erillinen liiketoiminnan asiantuntija. (Teikari 2012, 34.)

SAFe-mallissa asia on ratkaistu asiakkaan tuotehallinnalla, joka on vastuussa tuotteen kehitysjonon määrittelemisestä ja priorisoinnista. Mallia käytetään, kun on kysymys laajoista kokonaisuuksista, jolloin yhden henkilön osaaminen ei oletettavasti riittäisi kukaan tuoteomistajan roolin hoitamiseen, ja siksi sitä vastaa tuotehallinta, joka yleensä koostuu useammasta henkilöstä. Sitä vastoin tuoteomistaja on SAFe:ssa toimittajan rooli. (SAFe 2015a.)

Useiden suomalaisten asiantuntijoiden yhdessä kirjoittamassa kirjassa, Ketterää kehitystä, mainitaan nimenomaan vakuutusyhtiö esimerkkinä tilanteesta, jossa yhden henkilön liiketoimintaosaaminen ei todennäköisesti riitä tuoteomistajan rooliin. Vakuutusyhtiössä järjestelmien toiminnallisuutta määrittelevät lait, vakuutusehdot, EU-standardit ynnä muut, tämän tyyppisessä tilanteessa tuoteomistajan tueksi onkin hyvä koota asiantuntijajoukko liiketoiminnan puolelta. (Auer ym. 2013, 32.)

Kuten edellä mainitsin SAFe-malli ja sen sisältämä tuotehallinta tukevat hyvin esimerkiksi vakuutusyhtiön tilannetta, jossa tarvitaan monipuolista liiketoiminnan asiantunte-
musta. Joka tapauksessa sekä tilaaajan että toimittajan edun mukaista on riittävä liiketoiminnan edustus tiimissä, liiketoiminnan edustajien tulee olla päivittäin käytettävissä projektin töihin. Toimittajalle on erityinen riski, jos asiaa ei ole huomioitu sopimuksessa, sillä mahdolliset ongelmat ketterässä toimituksessa saatetaan tulkita yksinomaan toimittajan heikkoudeksi. Siksi vastuut tulee kirjata sopimukseen selkeästi, kumpi osapuoli vastaa mistäkin ja onko jollakin osuudella osapuolten yhteisvastuu (Teikari 2012, 36).

Sopimuksessa kuvataan osapuolten vastuut useimmiten sanallisesti, mutta alla on esitelty tapa käyttää matriisia (kuva 6).

Esimerkki vastuumatriisista

	Ostaja	Toimittaja
Liiketoiminnan vaatimukset	X	
Toteutus		X
Toiminnallisesta laadusta ja laajuudesta sopiminen	X	X
Hyväksyminen	X	

KUVA 6. Vastuumatriisi (Teikari 2012, 35)

4.1.2 Kehittyvä tiimi – toimittajan edustus

Toimittajan puolelta tärkeitä rooleja ovat scrummaster ja tekninen arkkitehti. Puhtaassa Scrumissa ei ole arkkitehdin roolia, mutta sen sijaan SAFe-malli palauttaa arkkitehdin jälleen projekteihin. Scrummaster on toimittajan puolen edunvalvoja ja ohjaa tiimin työskentelyä, hän takaa tiimilleen työrauhan ja hoitaa yhteydet tiimin ulkopuolisiin tahoihin. Arkkitehdillä ei ole sopimuksellista edunvalvojan roolia vaan hän on johtava sovelluskehittäjä. Koska arkkitehdin rooli on kuitenkin tärkeä projektin onnistumisen kannalta, myös hänet on hyvä kirjata sopimukseen mukaan. (Teikari 2012, 35; SAFe 2015a.)

Ketteryydessä on olennaista, että ainakin kehitystehtävissä olevat tiimin jäsenet ovat mukana sadan prosentin työpanoksella, sillä siten voidaan taata ketterä työskentely. Koska tiimi sitoutuu toimittamaan nopeassa rytmissä, tilaajan tulee ostaa tiimin jäsenten koko työaika siksi ajaksi, kun he työskentelevät projektissa. Nopea toimitusrytmi tarkoittaa 2-4 viikon välein valmistuvaa testattua ominaisuutta. On selvää, että tämä nopean toimittamisen malli ei voi toteutua, jos yksi tai useampi henkilöistä siirretään muihin

tehtäviin kesken sprintin tai alun perinkin oletetaan, että henkilö jakaa työaikaansa usealle projektille viikon aikana.

SAFe määrittelee tiimin kooksi 5-9 henkilöä mukaan lukien tuoteomistajan ja scrummasterin. Tiimillä on mallin mukaan yksi tuoteomistaja, joka voi työskennellä korkeintaan kahdessa SAFe-tiimissä samanaikaisesti. Scrummasterin rooli voi olla jaettu maksimissaan kahden tai kolmen tiimin kesken, rooli voi olla myös osittainen rooli jollekin tiimin jäsenistä. (SAFe 2015a.)

4.1.3 Hyvä kehitysjonon hallinta

Kehitysjono (*Backlog*) on lista ominaisuuksista, jotka projektissa on suunniteltu toteutettavan. Kehitysjono elää jatkuvasti ja priorisoinnilla tarkoitetaan töiden järjestämistä sen mukaan, missä järjestyksessä ne halutaan otettavan jonosta työnalle. Tilaajan hyvä kehitysjonon hallinta on toimittajan työskentelyn edellytys, sillä se mahdollistaa tasaisen työkuorman ja antaa tiimille edellytykset tehdä sisällöllisesti oikeita asioita.

Myös muuten hyvin onnistuneissa ketterissä projekteissa on törmätty ongelmaan, jossa projektia ei osata lopettaa oikeaan aikaan. Budjettia saattaa olla jäljellä ja tämän vuoksi jatketaan työskentelyä ottamalla kehitysjonosta toteutettavaksi ominaisuuksia, joiden merkitys on pieni. Ketteriin sopimukseen suositellaan tapaa, jossa tilaaja voi päättää projektin jokaisen sprintin lopuksi. Koska toimittaja on kiinnittänyt työntekijänsä tiimiin alkuperäisen suunnitelman mukaan, toimittajan riskiä voidaan vähentää ketteriin menetelmiin sopivalla hinnoittelumallilla, joka palkitsee kustannusten alituksesta (ks. 5.2 Hinnoittelu). Hyvän kehitysjonon hallinnan ansiosta tällainen tilanne on ennakoitavissa hyvissä ajoin etukäteen (Järvenpää & Kovanen, 17, 19).

4.1.4 Pienin markkinakelpoinen tuote

Yksi mahdollisuus rakentaa tilaajan ja toimittajan välistä luottamusta on tehdä sopimus ensin vain MVP:n toteutuksesta (ks. luku 2.4). Jos sen työmäärä on riittävän suppea, voi kysymykseen tulla perinteinen kiinteähintainen sopimus tälle osuudelle. (Järvenpää & Kovanen, 17.)

Sopimusmielessä näin päästään kokeilemaan, miten yhteistyö tilaajan ja toimittajan välillä sujuu. Toteuttamisessa voidaan edetä myös siten, että tehdään oma kehitysjono pelkästään MVP:n töille. Inkrementaalinen eli pienin lisäyksin kasvava toimittamisen malli tukee silloin molemmin puolisen luottamuksen syntymistä. Jokaisen kehitysjakson (*Sprint*) päätteeksi voidaan tarkistaa, syntyikö asiakkaalle arvoa. MVP:n toteuttamisen jälkeen on tilaisuus arvioida molemmin puolin, halutaanko yhteistyötä jatkaa.

4.1.5 Hyväksymismenettely

Ketterien periaatteiden mukaan toimittaja tuottaa tuotantoon siirtokelpoisen version jokaisen sprintin päätteeksi, jokaisen version tulee olla testattu ja dokumentoitu. Tilaaajan hyväksymistestaus voidaan tehdä sprinttikohtaisesti, mutta käytännössä on usein järkevää kerätä useamman sprintin tuotokset liiketoiminnallisiin kokonaisuuksiin sekä toimittajan suorittamaa integraatiotestausta että tilaajan hyväksymistestausta varten. Varsinkin, jos tuotantoon siirtopäivät ovat kiinteitä ja niitä on vain muutama vuoden aikana. (Auer ym. 2013, 32-33.)

4.2 Projektikolmiosta ketterään kolmioon

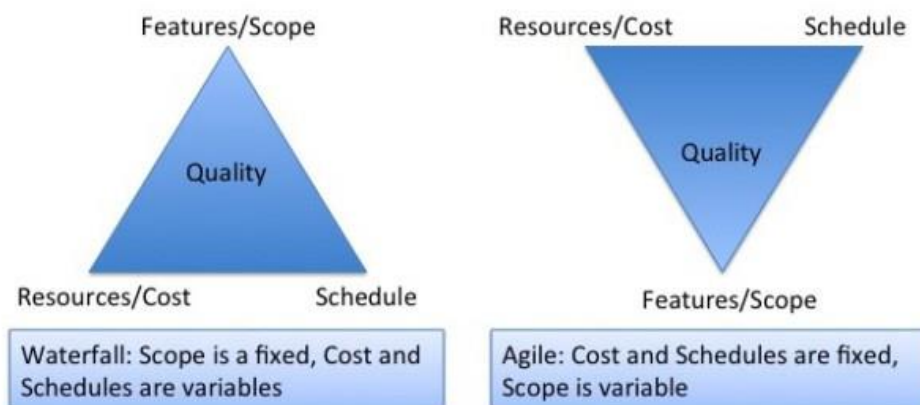
Kirjassa Ketterää kehitystä kuvataan siirtymistä vesiputousmallin sopimuksesta ketterään sopimukseen muun muassa näin:

Perinteinen projektimyynä johtaa usein vaatimusten lukitsemiseen etukäteen, jolloin looginen tarve vaatimusten keskinäiselle priorisoinnille poistuu. Ilman priorisointia tuotamme väistämättä turhia ominaisuuksia eli hukkaa. Ongelman voi välttää sopimuksella, jossa ostetaan kehitystiimin työaikaa etukäteen määritellyn sisällön sijaan. (Auer ym. 2013, 29.)

4.2.1 Projektikolmio

Projektikolmio kuvaa mainiosti vesiputousmallin hyvät ja huonot puolet. Kehitettävän tuotteen ominaisuudet, projektin kustannukset ja aikataulu kiinnitetään ja niistä tehdään kiinteähintainen sopimus. Kustannukset ja aikataulu eivät ole alun perin tarkoitettu

muuttujiksi, mutta käytäntö on kuitenkin osoittanut, että yleensä kustannukset ja aika joustavat, kun ominaisuuksista pidetään kiinni (kuva 7). Nykyisin vesiputousmallin projekteissa noudatetaan tiukkaa muutoksenhallintaa, koska muuten projektin aikataulu saattaa venyä ja kustannukset kasvaa hallitsemattomasti. Muutoksenhallinnalla pystytään osoittamaan, mihin aika ja raha ovat kuluneet. Yleensä vesiputousmallissa kaikki määrittelyyn tehdyt muutokset maksavat tilaajalle erikseen.



KUVA 7. Vesiputousmallin ja ketteryyden joustavuuden ero (Vaidya 2014)

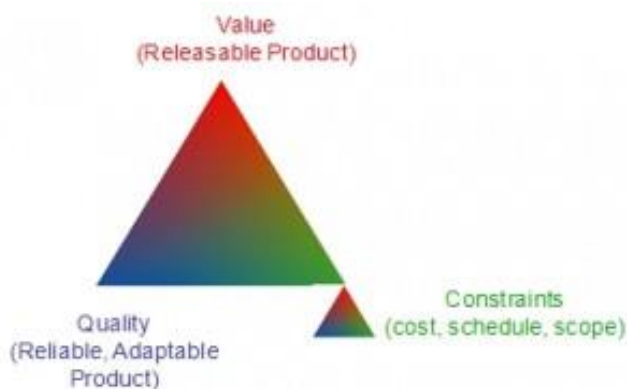
Ketteryydessä projektikolmio kääntyy ylösalaisin, kuten kuvassa 7 on esitetty, ja tuotteen ominaisuudet ovat joustava osuus (Vaidya 2014). Tässä piilee ketterien menetelmien hienous, sillä projektissa tehdään vain tarpeellinen eikä käytetä aikaa ja rahaa ominaisuuksiin, joilla ei saavuteta liiketoiminnallista arvoa. Riski siihen, että tilaaja ei saisi investoinnilleen riittävästi vastinetta, on olemassa, mutta toisaalta mahdollisuus havaita kehityksen huono suunta tulee vastaan niin aikaisin, ettei tilanteen pitäisi kehittyä pitkälle. Tilaaja näkee työn tulokset paljon aiemmin kuin vesiputousmallin toimituksessa ja ominaisuuksia valitaan työn alle jatkuvasti priorisoiden. Sen sijaan, jos kiinteällä hinnalla on ostettu joukko ominaisuuksia, niin uudet hinta- ja aikatauluneuvottelut ovat vääjäämättä edessä, jos halutaan muuttaa jotakin mistä on alun perin sovittu.

Viime aikoina ohjelmistokehityksessä on keskitytty jatkuvasti kasvattamaan nopeutta, jolla markkinakelpoinen ominaisuus saadaan tuotettua. Tämä on alkanut herättää epävarmuutta siitä, tingitäänkö samalla sovelluksen teknisestä laadusta aikataulu- ja kustannuspaineiden alla. Niin sanottu tekninen velka tarkoittaa juuri sitä, että on oikaistu

tekniseen arkkitehtuuriin liittyvistä ratkaisuista. Tekninen velka voi tulla myöhemmin esiin erilaisina ongelmina sovelluksessa, esimerkiksi odottamattomina virheinä siinä vaiheessa, kun ohjelmistoa jatkokehitetään. Ohjelmiston sisältämä tekninen velka kasvattaa siten ylläpito- ja kehityskustannuksia. Varsinaisen kehitystyön jälkeen jatkuva pitkä testaus- ja integrointijakso saattaa myös olla oire teknisestä velasta ja viittaa mahdollisesti huonoihin käytäntöihin. (Järvenpää & Kovanen, 6-9; Smolander 2015.)

4.2.2 Ketterä kolmio

Jim Highsmith vie kolmion uudelle tasolle ja haluaa kuvata sillä nimenomaan laadun ja arvon tuottamista (kuva 8). Highsmithin ketterässä kolmiossa (*Agile triangle*) arvoa edustaa julkaisukelpoinen tuote ja laatua vastaa luotettavasti toimiva sekä muokattava ja joustava sovellus. Ketterän kolmion kolmas kulma sisältää edelleen ominaisuudet, kustannukset sekä aikataulun ja nämä kolme yhdessä muodostavat selkeät rajoitukset työille. (Highsmith 2010).



KUVA 8. Ketterä kolmio (Highsmith 2010)

Highsmithin (2010) mukaan tuotteen ominaisuudet (*scope*) ovat huono mittari projektin hallintaan, sillä tutkimusten perusteella ohjelmistot sisältävät yli 50 prosenttia sellaista toiminnallisuutta, jota ei koskaan käytetä tai käytetään erittäin harvoin. Osa tästä on toki ominaisuuksia, jotka on pakko ottaa mukaan, kuten esimerkiksi vuoden vaihteen kirjanpito. Harvoin käytettyjen ominaisuuksien suuren määrän vuoksi olisikin parempi arvioida projektin tuottamaa arvoa.

Laadun osalta Highsmith (2010) korostaa, että tekninen velka pitäisi pystyä pitämään mahdollisimman pienenä. Joustavat suunnitteluratkaisut mahdollistavat sekä ennakoitua että ennakoimattomat tulevaisuuden muutokset. Käytännössä moni ketterä tiimi joutuu kohtaamaan ristiriidan tässä suhteessa, koska projektijohto tai asiakkaat saattavat vaatia joustavuutta ja ketteryyttä samanaikaisesti suunnitelman noudattamisen kanssa. Kun suunnitelmalla tarkoitetaan projektikolmion mukaista kolminaisuutta ominaisuudet, aikataulu ja kustannukset, joiden lähtökohtainen tarkoitus ei ole joustaa, paradoksi on Highsmithin (2010) mukaan valmis. Ketterä kolmio korostaa siksi projektin todellisia tavoitteita, kuten asiakasta ilahduttavan arvon tuottamista niin laadukkaasti, että kehitystyö nopeutuu.

Kolmesta rajoituksesta vain yksi voi olla tärkein ja yleensä se on aikataulu. Ketterä toimitushan rytmitetään sprinttien ja julkaisujen (*release*) tahtiin ja tuotteen ominaisuudet joustavat, siksi aikataulu on useimmiten tärkein rajoitus. Etenkin SAFE-mallissa aikataulu tiedetään pitkälle etukäteen ja julkaisujen päivämäärät noudattavat sovittua tahtia. Sen sijaan julkaisujen sisältöä ei tiedetä etukäteen.

Yleisestä keskustelusta löytyy mielipiteitä, joiden mukaan projektikolmio muuttuu siten, että vain yksi kolmesta tekijästä kiinnitetään (Madden 2014). Jos ketterän kolmion rajoituksia pohditaan tästä näkökulmasta, kustannukset nousevat keskeiseksi tekijäksi kuitenkin jokaisessa kohdassa (taulukko 1), siksi luotan enemmän aiemmin esittelemääni ajattelutapaan, jossa nimenomaan ominaisuudet (*scope*) joustavat ja kustannusten suuruusluokka tiedetään. Liiketoiminnassa on kuitenkin aina jokin raja, kuinka paljon ICT-hankkeeseen ollaan valmiita sijoittamaan. Lisäksi uskon, että Highsmithin (2010) esiin ottamat laatu ja arvo, ovat niitä ominaisuuksia, joiden perusteella toimittajat lopulta kilpailevat keskenään, kun toimitussopimukset perustuvat yhä enemmän tilaajan ja toimittajan väliseen luottamukseen ja yhteistyöhön. Tietysti myös hinnalla kilpaillaan, mutta siinä toimittajat pyrkivät arvioimaan yleistä hintatasoa.

Taulukon 1 avulla olen pohtinut sitä, että vain yksi projektikolmion ominaisuus olisi kiinnitetty. Yleensä projektiin kiinnitetään aina henkilöt, ja jos kustannukset on kiinnitetty ominaisuus, voidaan laskea kuinka pitkään tiimi pystyy työskentelemään ennen kuin budjetti ylittyy, näin saadaan projektille myös aikataulu ja siten kaksi kolmesta ominaisuudesta tiedetään. Toisinpäin, jos aikataulu on kiinnitetty ja tiimin henkilöt ovat

tiedossa, voidaan henkilöiden tuntiveloituksen perusteella arvioida projektin kokonaiskustannukset. Tässäkin tilanteessa projektin kustannukset ja aikataulu ovat tiedossa olevia ominaisuuksia. Jos sen sijaan ominaisuudet kiinnitetään, myös jonkinlainen työmääräarvio on mahdollista antaa sekä samalla työmäärään perustuva kustannusarvio. Myös tällöin kustannukset voidaan arvioida ja niin pitää ollakin, ei tilaajan voida olettaa sitoutuvan, ellei toimituksen kustannustaso ei ole selvillä, vaikkakaan lähteessä ei tuoda tätä kovin selkeästi esiin.

TAULUKKO 1. Jos vain yksi kolmesta tekijästä kiinnitetään (Madden 2014).

Kustannukset kiinnitetty	Aikataulu kiinnitetty	Ominaisuudet kiinnitetty
<ul style="list-style-type: none"> • Projektilla on kiinteä budjetti ja toimittaja sitoutuu siihen, että budjettia ei ylitetä • Kiinteä budjetti tarkoittaa sitä, että työ lopetetaan silloin, kun projektille varattu rahasumma on käytetty 	<ul style="list-style-type: none"> • Jos tilaajan täytyy saada toimitus tietyllä aikavälillä, kustannukset ja toteutettavat ominaisuudet joustavat • Kiinnitetty aikataulu tarkoittaa, että tiimi työskentelee sovitun ajan 	<ul style="list-style-type: none"> • Mahdollinen, jos tilaajalla on vaatimusmäärittely olemassa • Ominaisuudet kiinnitetty, kustannukset ja aika joustavat -> vesiputousmalli
Suhde muihin ominaisuuksiin:		
Kun tiimin koko ja budjetti tiedetään, saadaan projektille myös aikataulu	Kun tiimi on kiinnitetty aikavälille, voidaan kustannusarvio projektista kuitenkin antaa	Määrittelyyn ja sitä kautta työmäärään perustuva kustannusarvio voidaan antaa
Projektin onnistuminen edellyttää tilaajalta		
Hyvää kehitysjonon hallintaa, jotta toteutukseen saadaan ensimmäisenä tärkeimmät ominaisuudet ja jätetään sivuun toistaiseksi 'kiva saada' ominaisuudet	Hyvää kehitysjonon hallintaa, jos halutaan lisää ominaisuuksia, silloin toki kustannukset joustavat, koska projektiin on otettava lisää ihmisiä	Tilaja ymmärtää, että kustannukset ja aika voivat kasvaa, kun varmistetaan kaikkien määriteltyjen ominaisuuksien mukaan saaminen

5 SOPIMUS JA HINNOITTELU

5.1 Puitesopimus ja projektikohtainen sopiminen

Ketteriin toimituksiin suositellaan mallia, jossa käytetään yleisempää puitesopimusta ja tilannekohtaista projektisopimusta. Jos ketteryyttä ei ole puitesopimuksessa mukana, mutta sopimus sallii projektikohtaisen sopimisen, voidaan myös projektisuunnitelmaa liitteineen käyttää sopimusasiakirjana tilaajan ja toimittajan välillä. (Auer ym. 2013, 32.)

Yksi ICT-alan tunnetuimmista suomalaisista puitesopimuksista on nimeltään IT2010 Yleiset sopimusehdot. Näiden yleisten ehtojen käyttäminen vaatii yritykseltä käyttöoikeuslisenssin ostamisen vuosittain. Syyskuussa 2015 ehtoihin julkaistiin lisäosa nimellä IT2015 EKT, Erityisehtoja ohjelmistojen toimituksista ketterillä menetelmillä. Aiemmin ketteryyttä ei ole ollut mukana suomenkielisissä puitesopimuksissa. Yleisissä ehdoissa ovat valmiina monet lakiin perustuvat asiat kuten salassapito, ylivoimainen este, tietoturva ja henkilötietojen käsittely. Uudet lisäehdot sisältävät lisäksi ketterien käsitteiden määritelmiä, immateriaaliset oikeudet ja takuun. (IT2010; IT2015.)

Houston Inc:n Vesa Teikarin (2012, 42) mukaan projektikohtaiseen sopimukseen kannattaa sisällyttää lähinnä sopimuksen osapuolet, vastuut, hinnat ja maksuehdot. Lisäksi työtavat, tiimi ja työkalut ovat hyvä olla sopimuksessa. Varsinkin, jos tiimi on hajautettu, palaverikäytännöistä ja muista työtapoista sopiminen on hyödyllistä, sillä tokihan matkustaminen nostaa projektin kustannuksia, jos sitä ei ole alun perin huomioitu. Ketterän kehityksen ostajan oppaassaan Teikari ei sen sijaan suosittele juridisten lausekkeiden ottamista mukaan projektisopimukseen, koska ne menevät helposti ristiin puitesopimuksen kanssa. Projektisopimuksen tarkoitus on sopia mitä kyseisessä projektissa on tarkoitus tehdä ja sitä voidaan pitää työkaluna asian hallinnoimiseen.

5.1.1 Yhteistyö, työskentelymalli ja vastuut

Kuten aiemmin olen todennut (ks. 4.1), yhteistyöstä sovittaessa ketterässä sopimuksessa kannattaa sopia erityisesti vastuut. Vastuista voidaan sopia käytettävään menetelmään

liittyvien roolien perusteella, esimerkiksi sopimalla, että tilaajan tuoteomistaja vastaa kehitysjonon hallinnasta ja toimittajan scrummaster ja tiimi vastaavat kehitystyön toteuttamisesta. Sen sijaan muutoksenhallintaa sopimuksessa ei tarvita, koska sisällöstä sovitaan osapuolten kesken työn edistyessä. Myöskään IT2015 lisäehdoissa ei mainita muutoksenhallintaa (IT2015).

Kaikkien toimitusten ei kuitenkaan tarvitse olla muodoltaan ketteriä. Jos toimeksianto on lakimuutosten tekeminen vanhaan järjestelmään (*legacy*) voi kiinteähintainen sopimus palvella edelleen hyvin, koska muutokset ja aikataulu ovat ulkopuolisen tahon määräämiä eikä niistä voida joustaa. Etukäteen annetun aikataulun vuoksi määrittely on joka tapauksessa tehtävä riittävän tarkasti, että työmäärä ja sen vaatimat henkilöresurssit voidaan arvioida. Tällaisessa tilanteessa osapuolet voivat hyvin päätyä kiinteähintaiseen sopimukseen, koska sille on kaikki edellytykset olemassa. Lakimuutokset ovat erityinen poikkeus, jossa toimituksen sisältöön ei voida vaikuttaa.

5.1.2 Immateriaaliset oikeudet työn tuloksiin

Immateriaalisista oikeuksista, joihin tekijänoikeudet kuuluvat, on lisäehdoissa mainittu, että oikeudet toimitukseen, julkaisuihin ja dokumentaatioon kuuluvat toimittajalle (IT2015, 10.1). Asiakkaalla on kuitenkin vapaa kopiointioikeus sekä oikeus käyttää toimitusta, julkaisuja ja dokumentaatiota pohjana jatkotyölle, mutta asiakkaalla ei ole oikeutta myydä tai muutenkaan luovuttaa niitä kolmannelle osapuolelle muutoin kuin edellä mainittua tarkoitusta varten. (IT2015, 10.2).

Puitesopimuksessa pyritään turvaamaan molempien sopijapuolien oikeuksia. Myös kirjassaan *Ketterää kehitystä* IT-ammattilaiset suosittelevat huomioimaan sopimuksessa immateriaaliset oikeudet ohjelmistokoodiin. Yllä olevan kaltainen puitesopimus kuitenkin riittänee hyvin, jos sellainen on käytettävissä. (Auer ym. 2013, 31.)

5.1.3 Tilaajan oikeus vaihtaa henkilöitä ja toimittajaa

Ketteriin sopimukseen liittyy yleisesti tilaajan oikeus vaihtaa kehityspalvelun toimittajaa. Tämä liittyy esimerkiksi tapaan tehdä sopimus julkaisu kerrallaan, mutta oikeus on lail-

linen muutenkin. Julkaisu kerrallaan sopimiseen voidaan päätyä erityisesti silloin, kun tilaajan ja toimittajan välinen suhde on uusi. Lisäehtojen (IT2015) kohdassa 6.6 todetaan, että ellei toisin ole kirjallisesti sovittu, asiakkaalla on mahdollisuus päättää projekti ilmoittamalla siitä kirjallisesti toimittajalle 14 päivää etukäteen.

Ketterässä toimituksessa tilaajalle annetaan yleensä myös oikeus vaihtaa kehitystyöhön osallistuva henkilö, jos tilaaja on tyytymätön yksittäisen henkilön työpanokseen. Lisäehdot (IT2015) eivät velvoita sopimaan nimetyistä avainhenkilöistä, mutta jos henkilöistä on sovittu, niin ilman toisen sopijapuolen suostumusta avainhenkilöä ei voi vaihtaa ellei kysymys ole sopijapuolesta riippumattomasta syystä kuten työntekijän irtisanoutumisesta (IT2015, 5.3).

Mielestäni tässä ja edellisessä luvussa kuvatut lain mukaiset tilaajalle suotuisat oikeudet vaikuttavat siihen, että toimittajan etu on pyrkiä laadukkaisiin toimituksiin, siten että yhteistyö tilaajan kanssa saadaan toimimaan parhaalla mahdollisella tavalla. Käytännössä toimittajan pitää jatkuvasti osoittaa olevansa luottamuksen arvoinen kumppani.

5.2 Hinnoittelu

Ketteriin toimituksiin parhaiten sopivana hinnoittelumallina pidetään yleisesti tuntiveloitusta asiantuntijatyöstä (*time & material*) (Järvenpää & Kovanen, 17). Tässä mallissa tilaaja maksaa tiimin henkilöiden työstä toimittajan käyttämien tuntien mukaan. Koska ketterien periaatteiden mukaan sisältö joustaa ja iteraatioiden sisältöä ohjaa jatkuva asiakaspalaute, kiinteähintainen sopimus veisi pohjan ketterältä toimitustavalta. Kiinteähintainen sopimus vaatii tarkan etukäteen tehdyn määrittelyn ennen kuin toimittaja voi sitoutua ennalta sovittavaan kokonaishintaan. Tämä ei kuitenkaan tarkoita sitä, että koko ketterän toimituksen hinta jäisi arvailujen varaan. Kaikissa ketterän ohjelmistokehityksen viitekehyksissä toimitus sovitaan jollekin tietylle aikavälille (toisin sanoen aika on projektissa kiinnitetty ominaisuus) ja henkilöt kiinnitetään projektiin täksi ajaksi. Näin ollen maksimihinta toimitukselle saadaan helposti laskettua henkilöiden tuntiveloitukseen perustuen. (Auer ym. 2013, 30-31.)

Ei ole kuitenkaan vain yhtä tapaa käyttää tuntiveloitusta. Seuraavat luvut antavat ideoita siihen, miten tuntiveloitusta voidaan soveltaa ketterissä projekteissa useammalla eri

tavalla suoraviivaisen henkilön rooliin ja tehtyihin tunteihin perustuvan tuntiveloituksen lisäksi.

5.2.1 Bonukseen perustuva hinnoittelumalli

Bonusmalli on nimensä mukaisesti toimittajaa palkitseva malli, jossa houkutin on onnistumisesta saatava porkkana. Mallilla tarkoitetaan tuntihintaan perustuvaa sopimusta, jossa toimittaja sitoutuu normaalia tuntihintaansa alempaan veloitukseen tietyn toimituksen osalta ja sopimukseen kirjataan vastapainoksi bonussumma, jonka tilaaja maksaa, jos erikseen sovittavat bonukseen oikeuttavat ehdot täyttyvät. Jos bonustavoite sen sijaan ei täyty, toimittajan on tyytyminen normaalia veloitustaan alempaan tuntihintaan sopimuksen mukaisesti. Toimittajan tulee arvioida oman riskinsä suuruus etukäteen. Haastavaa mallissa on tavoitteiden asettaminen ja mittarien löytäminen siten, että päästään oikeudenmukaiseen sopimuksen kummankin osapuolen kannalta. Mittarit voidaan sopia tapauskohtaisesti vaikka niinkin, että ne ovat sprinttiin sidottuja ja bonus maksetaan täysimääräisenä vain, jos tavoitteet saavutetaan kaikkien sprinttien osalta. (Teikari 2012, 41.)

Yksinkertaisempi versio bonusmallista on sitoa bonuksen maksu alkuperäiseen hintaarvioon: jos toimittaja alittaa budjetin, tilaaja maksaa sovitun bonuksen ja toisaalta, jos alkuperäinen budjetti ylittyy, toimittaja veloittaa alemman tuntihinnan budjetin ylittävältä osalta. (Järvenpää & Kovanen, 17).

5.2.2 Hybridimalli

Hybridimallilla tarkoitetaan erilaisten hinnoittelumallien yhdistämistä samaan toimitukseen. Tällä toimittaja pyrkii antamaan tilaajalle mahdollisuuden todeta kyvykkyytensä ja luotettavuutensa. Yhteistyön alkuvaiheessa on mahdollista tehdä sopimus julkaisu kerrallaan tai sopia kiinteä hinta toimituksen ensimmäiselle osalle, jonka pitää silloin olla riittävän pieni arvioitavaksi kiinteähintaisena. Myöhemmin kun tilaajan ja toimittajan välinen luottamus on alkanut muodostua, voidaan siirtyä käyttämään asiantuntijatyön tuntiveloitusta. Tuntiveloituksen etu on helppous ja joustavuus, koska projektitoi-

mituksen sisällön (*scope*) muuttuessa sopimusta ei tarvitse muuttaa kuten vesiputousmallissa piti. (Järvenpää & Kovanen, 17.)

Unboxed Consulting Britanniasta on esitellyt mallin, jossa tehdään projektin ensimmäiset kolme sprinttiä aikaperusteisella hinnoittelulla. Tällä tarkoitetaan, että tiimi työskentelee tietyn ajan ja sen jälkeen arvioidaan tulevaa hinnoittelua tehtyjen tuntien ja aikaan saatujen tuotosten perusteella. Arvioinnissa käytetään ketterissä menetelmissä tunnettuja arviointipisteitä (*story points*), ja yhden pisteen hinta lasketaan toteutuneiden pisteiden ja niihin käytetyn ajan perusteella. Tämän perusteella toimittajan on mahdollista luvata toimittavansa vähintään tiettyä pistemäärää vastaava määrä valmista toteutusta jokaisessa kehitysjaksoissa, ja siten myös toimittajan kannalta tärkeä minimiveloitus saadaan muodostettua. Arvioinnin lähtökohdat muutaman sprintin jälkeen ovat huomattavasti paremmat kuin projektin alkaessa, koska ne perustuvat kokemukseen. (Järvenpää 2010.)

5.2.3 Keskituntiveloitus asiantuntijatyöstä

Kokemusten perusteella tuntihintaan perustuva veloitus asiantuntijatyöstä, on käytännössä osoittautunut ketteriin menetelmiin parhaiten sopivaksi laskutusperusteeksi. Perinteisesti toimittajat laskuttavat roolipohjaisesti siten, että esimerkiksi projektipäälliköllä sekä vanhemmalla ja nuoremmalla sovelluskehittäjällä on kullakin roolinsa mukainen tuntihinta, ja toimittaja laskuttaa henkilöiden tekemien työtuntien perusteella. Samoin voidaan toimia ketterissä projekteissa. Toisaalta, koska ketteryydessä ostetaan tiimin työaikaa, mielenkiintoinen malli perinteisen rinnalla voisi olla tiimin keskituntihinta, jossa ei eritelläkään tehtyjä työtunteja roolin mukaan, vaan lasketaan henkilöiden keskituntihinta ja laskutetaan kaikki tiimin tekemät työtunnit tuolla hinnalla.

Hinnoittelumalli myötäilisi silloin ketteriä periaatteita, joissa korostetaan tiimin hyvää yhteistyötä. Mallin etuna olisi myös kustannusten hyvä ennustettavuus, jos tiimin jäsenet työskentelisivät niin ikään ketterien periaatteiden mukaan täysipäiväisesti kyseisessä projektissa. Tiimin jatkuvan kehittymisen avulla myös tehokkuus lisääntyisi ja laskuttaminenkin olisi helpompaa. Se kummalle osapuolelle malli kulloinkin olisi edullisempi, riippuisi tiimin kokoonpanosta, mutta merkittäviä eroja tuskin syntyisi tarkkaan tuntiveloitukseen verrattuna.

6 POHDINTA

Sopimuksen kannalta ensisijainen ongelma on, että ICT-projekteja on totuttu ostamaan kiinteällä hinnalla ja mieluusti ostajan puolella edelleenkin toimittaisiin näin, vaikka toimitustapa on muuttunut. Tilaajan on hankala luottaa toimitukseen, jossa sisältöä ei ole ennalta määrätty ja jonka hinnoittelumalliksi suositellaan tuntiveloitusta asiantuntijatyöstä. Tällainen ostamisen malli saatetaan tilaajan puolella kokea siten, että toimittaja pääsee laskuttamaan hallitsemattomasti.

Olen päätenyt siihen, että ostaakseen ketterän toimituksen on ennen kaikkea tunnettava menetelmä ja luotettava siihen. Molempien sopimuksen osapuolten tulee varmistaa omalta osaltaan riittävä menetelmäosaaminen, jonka jälkeen sopiminen tuntiveloitukseen perustuvasta asiantuntijatyöstä pitäisi olla luontevampaa. Sopimuksessa on erityisen tärkeää kuvata kummankin osapuolen vastuut. Toimittaja ei voi selviytyä ketterästä toimituksesta ilman tilaajan sitoutumista jatkuvaan yhteistyöhön, vaan toimittajan näkökulmasta katsottuna ketterän projektin onnistuminen edellyttää muun muassa tilaajan hyvää kehitysjonon hallintaa. Nimenomaan tilaajan riski on projektin läpinäkyvyyden ja pienemmissä osissa valmistumisen vuoksi vesiputousmallia pienempi. Toimittajan tulee jatkuvasti osoittaa olevansa tilaajan luottamuksen arvoinen.

Hyvällä sopimuksella luodaan projektille onnistumisen edellytykset myös sillä, että projektiin kiinnitetään henkilöt, joilla on tiiminä riittävä osaaminen tulevasta tehtävästä suoriutumiseen. Nopeasta toimitusrytmistä johtuen ketterässä tiimissä toimiminen vaatii yleensä henkilön kiinnittämistä projektiin täysipäiväisesti. Toimittajan taloudellista riskiä voidaan pienentää palkitsevalla hinnoittelumallilla (bonusmalli), jossa toimittaja saa paremman tuntihinnan, jos ohjelmisto valmistuu ja sopimus päätetään arvioitua aiemmin ja vastaavasti, jos aikataulua joudutaan venyttämään, toimittaja kantaa vastuunsa veloittamalla halvemman tuntihinnan sopimukseen kirjatulla tavalla.

Tutkiessani aiheeni teoreettista taustaa havaitsin, että tällä hetkellä paljon huomiota keraava digitalisaatio, vaatii eri tahojen hyvää yhteistyötä tulevaisuudessa, joten tälläkin alueella ketterät periaatteet ja menetelmät kuulostavat ajankohtaisilta. Jatkossa mukaan projekteihin voi tulla myös laitetoimittajien edustajia. Onkohan tulossa uusia haasteita sekä ketterille toimituksille että niiden sopimuksille?

LÄHTEET

Agile Manifesto. 2001. Luettu 30.7.2015.

<http://www.agilemanifesto.org/iso/fi/manifesto.html>

Arbogast, T., Larman, C. & Vodde B. 2012. Practices for Scaling Lean & Agile Development. Luettu 3.12.2015. <http://www.agilecontracts.org/>

Auer, A., Auer, L., Heinäsmäki, M., Hölttä M., Kalliala, E., Laanti, M., Laine, K., Lekman, L., Miinalainen, P., Naski, H., Piiparinen, T., Puhakka, H., Pyhäjärvi, M., Pääkkönen, T., Räisänen, S., Sora, H., Taipale, M., Talvio, J., Tanninen, A., Toikkanen, T., Toivola, T., Toro, K., Valsta, A., Väyrynen, V., Weissenberg, M. Ketterää kehitystä. 2013. Finn Lectura.

Highsmith, J. 2007. Agile Project Management. Fourth printing. Addison-Wesley.

Highsmith, J. 2010. Beyond Scope, Schedule, and Cost: The Agile Triangle. Luettu 19.8.2015. <http://jimhighsmith.com/beyond-scope-schedule-and-cost-the-agile-triangle/>

IT2010 Sopimusehdot. Teknologiainfo Teknova. Luettu 16.8.2015.

<http://www.it2010.fi/>

IT2015 EKT. Erityisehtoja ohjelmistojen toimituksista ketterillä menetelmillä. Luettu 11.10.2015.

http://www.it2010.fi/system/files/IT2010_PDF/fi/EsikatselaIT2015_EKT_Suomi2015.pdf

Järvenpää, J. & Kovanen, P. Ohjelmistokehityksen ostajan pikaopas. 1. Painos. Vincit.

Järvenpää, J. 2010. Ketterä hinnoittelumalli. Blogi. Luettu 15.11.2015.

<https://www.vincit.fi/blog/kettera-hinnoittelumalli/>

Lempinen, S. Celkee Oy. Luettu 29.11.2015.

<http://lapinyhteinenkirjasto.blogspot.fi/2013/06/iteraatioilla-ihannetuloiksi.html>

Madden, D. 2014. I'm Agile But My Contract Isn't: How to Align Contracts with Agile Software Development Teams. Blogi. Luettu 30.7.2015.

<http://www.stridenyc.com/blog/im-agile-but-my-contract-isnt-how-to-align-contracts-with-agile-software-development-teams/>

Moreira, M. 2013. Being Agile: Your Roadmap to Successful Adoption of Agile. E-kirja. Apress.

Ries, E. 2009. Minimum Viable Product. Luettu 2.8.2015.

<http://www.startuplessonslearned.com/2009/08/minimum-viable-product-guide.html>

SAFe. Scaled Agile Framework. 2015a. Luettu 15.8.2015.

<http://www.scaledagileframework.com/>

SAFe. Scaled Agile Framework . Nordea case study. 2015b. Luettu 21.8.2015.

<http://www.scaledagileframework.com/nordea-case-study/>

- Smolander, K., Maglyas, A. & Nikula, U. 2012. Lean Solutions to Software Product Management Problems. Artikkele. Luettu 19.8.2015.
http://www.researchgate.net/publication/260649306_Lean_Solutions_to_Software_Product_Management_Problems
- Smolander, K. 2015. Overfly fast software development creates technical debt. Artikkele. Luettu 19.8.2015. <http://www.n4s.fi/2015magazine/article11/>
- Stellman, A. & Greene, J. 2014. Learning Agile. Understanding Scrum, XP, Lean, and Kanban. E-kirja. O'Reilly Media, Inc.
- Taali, M. 2013. Kuvitellen. Blogi. Luettu 25.1.2015.
<http://www.kuvitellen.fi/blogi/vieraileva-bloggaaja-marko-taali-kuinka-minusta-tuli-graafinen-fasilitoija>
- Tanninen, A. & Taipale, M. 2009. Scrum ei riitä. Luettu 4.10.2015
<http://www.slideshare.net/aritanninen/scrum-is-not-enough>
- Teikari, V. 2012. Don't panic – Ketterän kehityksen ostajan opas. Helsinki: Houston Inc.
- Tikka, A. & Nyman, R. 2015. Agile Scaling frameworks LeSS and SAFe have a history with Nokia. What problem were they solving and how? Blogi. Luettu 19.8.2015
<http://gosei.fi/blog/less-safe-comparison/>
- Vaidya, A. 2014. Adding Capacity to an Agile Project. Blogi. Luettu 18.8.2015.
<https://agilesutra.wordpress.com/2014/03/03/adding-capacity-to-an-agile-project/>