

Kaisa Jaakonaho

**SOPIMUSMALLIT KETTERÄN LÄHESTYMISTAVAN
MUKAISISSA PROJEKTEISSA**

Tietojärjestelmätieteen kandidaatintutkielma

16. joulukuuta 2009



JYVÄSKYLÄN YLIOPISTO
TIETOIENKÄSITTELYTIETEIDEN LAITOS

TIIVISTELMÄ

Jaakonaho, Kaisa

Sopimusmallit ketterän lähestymistavan mukaisissa projekteissa/ Kaisa Jaakonaho

Jyväskylä: Jyväskylän yliopisto, 2009.

47 s.

Kandidaatintutkielma

Ketterien menetelmien hyödyistä puhutaan paljon kirjallisuudessa, mutta budjetointi ja sopimusmallin valinta on koettu niissä haastaviksi. Perinteiset sopimusmallit vaativat vaatimusmäärittelyn jäädyttämisen projektin alkuvaiheessa, vaikka ketterän lähestymistavan mukaisissa projekteissa vaatimukset voivat muuttua projektin aikana. Perinteisissä sopimusmalleissa riskit eivät jakaannu tasaisesti osapuolten välille, mikäli vaatimukset muuttuvat. Sopimuksen tulisi olla yritysten riskienhallinnan keino, mutta se voi myös häiritä asiakasyhteistyötä ja vähentää luottamusta.

Tässä tutkielmassa tarkastellaan perinteisesti ohjelmistoprojekteissa käytettyjä sopimusmalleja sekä arvioidaan niiden soveltuvuutta ketterän lähestymistavan mukaisiin projekteihin. Lisäksi pyritään löytämään kirjallisuudesta ketterän lähestymistavan mukaisiin projekteihin ehdotettuja vaihtoehtoisia sopimusmalleja sekä niistä saatuja kokemuksia.

Tämän tutkimuksen perusteella voidaan sanoa, että ketterän lähestymistavan mukaisiin projekteihin esitetyistä sopimusmalleista on saatu hyviä kokemuksia ja niillä on voitu luoda luottamusta, mutta myös perinteisiä ja asiakkaille entuudestaan tuttuja sopimusmalleja voidaan edelleen käyttää tietyissä tilanteissa.

AVAINSANAT: sopimusmalli, ketterä menetelmä, riskienhallinta, luottamus

Ohjaaja: Mauri Leppänen
Tietojenkäsittelytieteiden laitos
Jyväskylän Yliopisto

Tarkastaja: Mauri Leppänen
Tietojenkäsittelytieteiden laitos
Jyväskylän Yliopisto

SISÄLTÖ

1	JOHDANTO	5
2	KETTERÄ LÄHESTYMISTAPA	8
	2.1 Vesiputousmallista ketteriin menetelmiin.....	8
	2.2 Ketterän lähestymistavan piirteitä	9
	2.3 Projektin elinkaari	12
3	SOPIMUS, RISKIENHALLINTA JA LUOTTAMUS	15
	3.1 Sopimukset liiketoiminnassa	15
	3.2 Sopimusprosessi.....	16
	3.3 Riskienhallinta.....	18
	3.4 Sopimus riskienhallinnan työvälineenä.....	20
	3.5 Sopimus vastaan luottamus	22
4	SOPIMUSMALLIT OHJELMISTOPROJEKTEISSA	24
	4.1 Sopimusmallin valintaperusteet.....	24
	4.2 Perinteiset sopimusmallit	27
	4.3 Sopimusmallit ketterän lähestymistavan mukaisiin projekteihin	31
	4.4 Yhteenvedo sopimusmalleista	36
5	YHTEENVETO	40
	LÄHTEET	43

1 JOHDANTO

Yhä useampi ohjelmistoyritys käyttää ohjelmistokehityksessä ketteriä menetelmiä. Ketterä lähestymistapa on kehitetty vastaamaan perinteisten ja raskaina pidettyjen prosessimallien, kuten vesiputousmallin ongelmiin. Ketterien menetelmien hyödyistä on keskusteltu paljon kirjallisuudessa (Abrahamsson ym. 2002), mutta suurin este ketterän menetelmän käyttöönotolle on havaittu olevan hinnoitteluun ja sopimusneuvotteluihin liittyvä problematiikka (Schwaber 2004).

Ohjelmistokehitysprojekteissa budjetti laaditaan tavallisimmin vaatimusmäärittelyn pohjalta (Sheta ym. 2008). Vaatimukset on totuttu kirjaamaan tarkasti sopimukseen projektin alkuvaiheessa, sillä tarkoin laaditun sopimuksen on tarkoitus vähentää yritysten riskejä (Warsta & Seppänen, 2000). Nykypäivänä ohjelmistoprojektit tulisi kuitenkin nähdä asiakkaan ja toimittajan välisenä yhteistyönä, jossa tavoitteet ovat yhdensuuntaisia ja molempien osapuolten riskit otetaan huomioon. Ohjelmistoprojekteissa veloitus toteutetaan yleensä joko tunti- tai urakkapohjaisena, joista ensimmäisessä asiakas kantaa riskit, kun taas jälkimmäisessä ohjelmiston toimittaja (Tietotekniikan liitto 2002). Perinteiset sopimusmallit, kuten kiinteä hinta ja aika-ja-materiaali, eivät tue näkemystä ohjelmistoprojektien yhteistyölähtöisyydestä. (Turnerin & Simiesterin 2009).

Ketterän lähestymistavan mukaisissa projekteissa vaatimukset saavat muuttua, mikä usein edesauttaa projektin onnistumista (Leffingwell 2007). Tästä herää kuitenkin kysymys, miten ketterän lähestymistavan mukaisissa projekteissa voidaan laatia sopimus, kun vaatimusmäärittely ja projektin laajuus ovat alussa epämääräisiä? Perinteisissä sopimusmalleissa vaatimukset jäädytetään ja niiden muuttaminen projektin aikana aiheuttaa riskien epätasaista jakaantumista, mikä voi häiritä asiakasyhteistyötä ja vähentää luottamusta. Ketterän

menetelmän käyttö vaatii luottamuksen luomista yritysten välille sopimusneuvottelujen sijaan. Tähän soveltuvaa sopimusmallia voi olla vaikea löytää, varsinkin jos osapuolet eivät tunne toisiaan. (Poppendieck & Poppendieck 2003).

Ketteriin menetelmiin on ehdotettu käytettäväksi muutamia vaihtoehtoisia sopimusmalleja. Poppendieck ja Poppendieck (2003) käsittelevät kirjassaan ketterien menetelmien sopimusneuvotteluihin liittyviä ongelmia ja esittelevät eri sopimusmalleja. Eckefelt ym. (2005) ovat kirjoittaneet erään aloittelevan ketterän yrityksen käyttökokemuksia tavoitehintasopimuksista. Franklin (2008) on kirjoittanut käyttökokemuksia erilaisista sopimusmalleista, joita hiljattain ketteriin menetelmiin siirtynyt yritys on käyttänyt vuosien varrella. Thorup ja Jensen (2009) ovat kirjoittaneet kokemuksia ketterien yhteistyösopimusten käytöstä. Stevens (2009b) arvioi blogissaan kymmenen erilaisen sopimusmallin soveltuvuutta ketterän lähestymistavan mukaisiin projekteihin.

Tässä tutkielmassa on tarkoitus kirjallisuuskatsauksella selvittää, minkälaisia sopimusmalleja ohjelmistoprojekteissa on käytetty ja minkälaisia sopimusmalleja on ehdotettu käytettäväksi ketterän lähestymistavan mukaisiin projekteihin. Lisäksi tutkitaan, minkälaisia kokemuksia sopimusmalleista on saatu ja minkälaisiin tilanteisiin ne soveltuvat parhaiten. Tutkimusongelma voidaan muotoilla seuraavasti: "Millaisia sopimustyyppisiä on esitetty käytettäväksi ketterän lähestymistavan mukaisissa projekteissa ja millaisia kokemuksia niistä on saatu?" Tämä ongelma voidaan jakaa seuraaviin tutkimuskysymyksiin: 1) Mitkä ketterän lähestymistavan piirteet tekevät kustannusten arvioinnin, hinnoittelun ja sopimuksen tekemisen haasteellisiksi? 2) Millaisia sopimustyyppisiä yleisesti ohjelmistoprojektien yhteydessä käytetään ja miten riskit niissä jakaantuvat? 3) Mitkä sopimustyypeistä on esitetty käytettäväksi tai on käytössä ketterän lähestymistavan mukaisissa projekteissa ja onko niistä ollut apua luottamuksen luomisessa?

Tässä tutkielmassa käsitellään vain koko projektia koskevia sopimuksia asiakkaan ja ohjelmiston toimittajan välillä, ei projektin sisällä tehtäviä ohjelmistoyritysten välisiä alihankintasopimuksia.

Seuraavassa luvussa esitellään ketterää lähestymistapaa sekä ketterän lähestymistavan mukaisen projektin elinkaarta. Kolmannessa luvussa käsitellään sopimuksia, luottamusta ja riskienhallintaa ohjelmistoprojekteissa. Luvussa kerrotaan, mihin yritykset tarvitsevat sopimuksia ja millainen on ohjelmistoyrityksen sopimusprosessi. Lisäksi tarkastellaan, miten sopimuksilla voidaan hallita riskejä ja miten luottamuksella voidaan parantaa yritysten välistä yhteistyötä. Neljäs luku käsittelee sopimusmalleja. Luvussa kerrotaan aluksi, miten sopimusmalleja voidaan vertailla, jonka jälkeen esitellään kuusi erilaista ohjelmistoprojekteissa käytössä olevaa sopimusmallia. Neljännen luvun lopussa on yhteenveto sopimusmalleista, jossa pohditaan sopimusmallien soveltuvuutta ketterän lähestymistavan mukaisiin projekteihin. Viides luku on yhteenveto, jossa kerrotaan työn keskeisimmät tulokset.

2 KETTERÄ LÄHESTYMISTAPA

Tässä luvussa kerrotaan, mihin tarpeeseen ketterät menetelmät on kehitetty, millainen on ketterä lähestymistapa ja millaisia ominaispiirteitä ja käytänteitä siihen liittyy. Lisäksi esitellään ketterän lähestymistavan mukaisen projektin elinkaarta. Erityisesti keskitytään ketterän lähestymistavan piirteisiin, jotka tekevät kustannusten arvioinnin, hinnoittelun ja sopimuksen tekemisen haasteellisiksi.

2.1 Vesiputousmallista ketteriin menetelmiin

Uusi internetaika (Internet time) loi tarpeita uudentlaisille ohjelmistokehitysmenetelmille. Perinteinen vesiputousmalliin (waterfall model) perustuva ohjelmistokehitys, jossa projektit viedään läpi määrättyssä ajassa ja vaatimukset lyödään lukkoon projektin alkuvaiheessa, ei soveltunut internetajan uusiin haasteisiin. Internetajalle tyypillisiä piirteitä ovat mm. sovellusten nopea toteuttaminen, asiakaskeskeisyys, vaatimusten muuttuminen nopeassa tahdissa, laadun kriteerien vaihtelevuus ja tuotteen saaminen markkinoille mahdollisimman nopeasti. Myös aikataulun tai budjetin ylittyminen havaittiin olevan ohjelmistokehitysprojekteissa (software development project) tavallista, eivätkä ohjelmistojen laatu ja toiminnallisuudet usein vastanneet asiakkaan odotuksia. (Baskerville ym. 2003). Aikataulun ja kustannusten arvioinnin haasteellisuuden on sanottu johtuvan projektien ainutlaatuisuudesta sekä ohjelmistojen kompleksisuudesta (Reel 1999). Ongelmia aiheuttaa myös se, etteivät asiakkaat osaa määritellä vaatimuksia kovinkaan tarkasti projektin alkuvaiheessa ja ymmärrys asiakkaan tarpeista on ohjelmistokehittäjille jatkuva prosessi (Davis 1994). Tämän vuoksi uudenlaiset kevyemmät menetelmät alkoivat kiinnostaa ohjelmistoalan ammattilaisia 1990-luvun lopussa. Ketterät menetelmät korostavat yksilöitä ja vuorovaikutusta,

toimivaa ohjelmistoa, tiivistä yhteistyötä asiakkaan kanssa sekä jatkuvaa muutosvalmiutta. (Abrahamsson ym. 2002).

Viime vuosina on kehitetty useita ketteräksi luokiteltavia menetelmiä. Osa niistä on hyvin käytännönläheisiä ja tarkkoja sääntöjä sisältäviä, kun taas osa käsittelee projektinhallintaa ja yleisiä linjanvetoja ohjelmistokehitykseen. Tunnetuimpia ketteriä menetelmiä ovat Scrum (Schwaber ym. 2002), Extreme Programming (Beck 1999) ja Lean (Poppendieck & Poppendieck 2003). Yhteistä näille menetelmille on kehitys iteraatioissa, yleensä muutaman viikon sykleissä, toimivien ohjelmaosien toimittaminen asiakkaalle mahdollisimman aikaisessa vaiheessa sekä tiivis asiakasyhteistyö. Tässä tutkielmassa ei käsitellä menetelmiä yksityiskohtaisemmin, vaan perehdytään asiakasyhteistyötä ja sopimusneuvotteluja sekä muutostenhallintaa koskevien ketterän lähestymistavan arvojen ja periaatteiden problematiikkaan.

2.2 Ketterän lähestymistavan piirteitä

Agile-allianssin manifesti (Agile Alliance 2001) luettelee neljä arvoa, jotka kiteyttävät ketterän lähestymistavan periaatteet ja toimii samalla ketterien menetelmien virallisena määritelmänä:

Yksilöitä ja vuorovaikutusta enemmän kuin prosesseja ja työkaluja

Toimivaa sovellusta enemmän kuin kokonaisvaltaista dokumentaatiota

Asiakasyhteistyötä enemmän kuin sopimusneuvotteluita

Muutokseen reagoimista enemmän kuin suunnitelman noudattamista

Ketterää menetelmää käyttävät kehittäjät ymmärtävät myös oikealla puolella olevien käsitteiden merkityksen ohjelmistokehitysohjelmassa, mutta arvostavat enemmän vasemmalla puolella olevia. Yksilöiden ja vuorovaikutuksen arvostus näkyy ketterien menetelmien käytänteissä vahvana tiimityöskentelynä. Ketterän lähestymistavan mukaisissa projekteissa korostetaan osaavia yksilöitä, hyvää tiimihenkeä ja vuorovaikutteista työympäristöä. Tiimeille annetaan suuri vastuu niin asiakasyhteistyöhön kuin sovelluksen tekniseen toteuttamiseenkin. (Abrahamsson ym. 2002). Johdon tehtävä on eliminoida organisaation häiritsevyyttä ja antaa tiimin saavuttaa tavoitteet parhaaksi katsomallaan tavalla (Leffingwell 2007). Toimiva sovellus edellyttää koodin pitämistä mahdollisimman yksinkertaisena, mikä helpottaa testausta ja vähentää dokumentoinnin tarvetta (Abrahamsson ym. 2002). Testaukset tehdään koodin kirjoittamisen yhteydessä ja valmiit ohjelmakomponentit integroidaan välittömästi osaksi sovellusta. Ohjelmistokehitys on tiimin sisällä läpinäkyvää, jolloin jokainen tiimin jäsen tietää missä vaiheessa projekti on ja mitä on vielä tekemättä. (Leffingwell 2007). Ketterän lähestymistavan mukaisissa projekteissa pyritään julkaisemaan toimivia sovelluksen osia lyhyissä, noin neljän viikon sykleissä. Liiketoiminnan näkökulmasta asiakkaalle pyritään tuottamaan arvoa heti projektin alusta alkaen jokaisella iteraatiokierroksella. (Abrahamsson 2002). Toiminnot pyritään priorisoimaan siten, että tärkeimmät toiminnot julkaistaan ensimmäisenä. Aikaisen julkaisemisen ansiosta asiakkaalta saadaan palautetta nopeammin (Leffingwell 2007). Asiakasyhteistyö on ketterissä menetelmissä suuressa roolissa, ja jatkuvan kommunikaation avulla pyritään luomaan vahva luottamussuhde asiakkaan ja kehittäjien välille. Muutostenhallintaa on pyritty parantamaan ketterissä menetelmissä siten, että projekteja viedään eteenpäin olettaen muutosten ilmaantumista. Asiakkaan ei tarvitse määrittää vaatimuksia alussa kovinkaan tarkasti, ja ne saavat tarkentua tai muuttua projektin aikana. Sopimukset laaditaan niin, että niihin voidaan tehdä muutoksia. (Abrahamsson ym. 2002). Suunnitelmia ei tehdä vain alussa vaan ennen jokaista

iteraatiokierrosta. Suunnittelu on systemaattista ja rutiininomaista, mutta myös helpompaa, sillä julkaisupäivät ovat tiedossa ja toiminnot voidaan priorisoida niiden mukaan. Projektin onnistumista mitataan sillä, miten paljon pystytään tuottamaan asiakkaalle arvoa, eikä sen mukaan kuinka hyvin pysyttiin aikataulussa. (Leffingwell 2007).

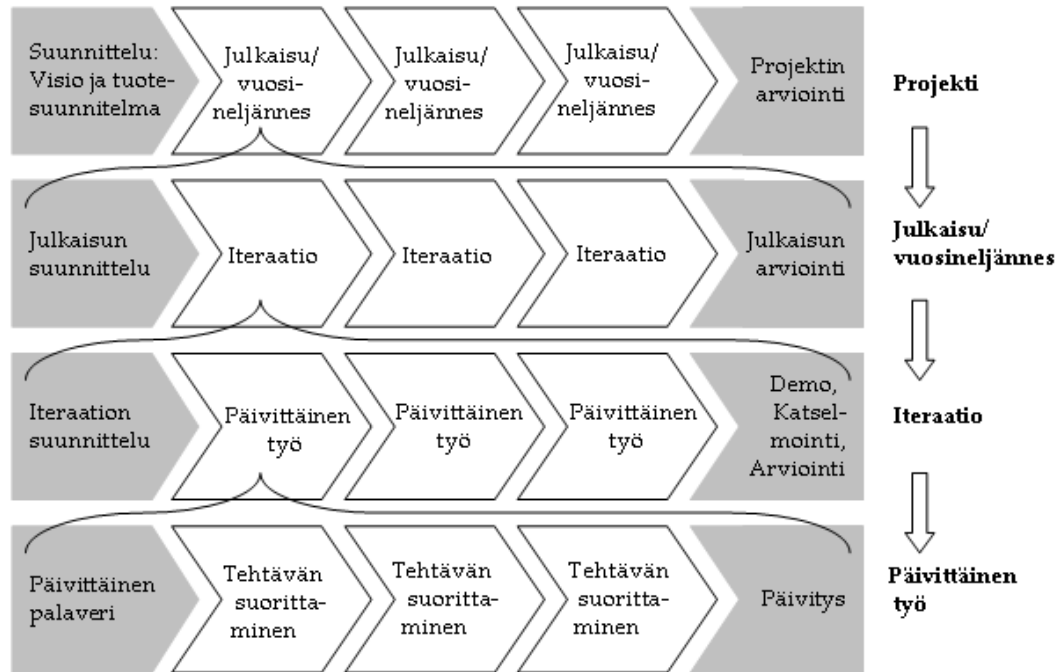
Ketterien menetelmien hyödyistä on keskusteltu paljon kirjallisuudessa (Abrahamsson ym. 2002), mutta hinnoittelu- ja budjetointiongelmiin ne eivät anna ratkaisua. Ketterän lähestymistavan mukaisissa projekteissa arvostetaan enemmän muutoksiin reagoimista kuin suunnitelmien noudattamista. Ohjelmistoprojekteissa budjetti kuitenkin arvioidaan tavallisesti asiakkaan antamien vaatimusten perusteella. Vaatimusmäärittelyn avulla voidaan arvioida projektin laajuus, jota voidaan mitata esimerkiksi toimintojen lukumäärän perusteella. Toiminnoista voidaan erotella vielä tarkempia lukumääriä, kuten sovellukseen tarvittavat sivut, tiedostot, dialogit, tietokannan taulut, viestit tai raportit. Työtunnit, aikataulu ja hinta saadaan laskettua, kun tarvittavat toiminnot on tarkkaan määritelty. (Sheta ym. 2008). Ketterän lähestymistavan mukaisissa projekteissa ohjelmiston toimintovaatimuksia ei kiinnitetä projektin alussa, vaan ne voivat muuttua, mikä tekee budjetoinnin ja sitä kautta hinnoittelun ja sopimusten teon hankalaksi perinteisillä keinoilla. (Abrahamsson ym. 2002)

Ketterän lähestymistavan mukaisissa projekteissa arvostetaan enemmän asiakasyhteistyötä kuin sopimusneuvotteluja. Sopimuksen, jossa määritellään tarkkaan sopijaosapuolten vastuut ja velvollisuudet, on kuitenkin tarkoitus vähentää sekä asiakkaan että toimittajan riskejä. Yritysten sisällä voi tapahtua monia muutoksia projektin aikana ja ainoa asia, joka lopulta ratkaisee, on sopimuksen sisältö. (Poppendieck & Poppendieck 2003). Vaikka asiakkaat ymmärtäisivät ketterän lähestymistavan hyödyt, on vaikea laatia sopimus, joka sallii muutoksia. Molempia sopimusosapuolia tyydyttävää ratkaisua on vaikea

löytää, sillä perinteiset sopimusmallit ovat usein joko asiakkaan tai toimittajan eduksi, eivätkä riskit jakaannu niissä tasaisesti. (Eckfeldt ym. 2005). Ketterän lähestymistavan suosion kasvu on luonut tarpeen uudentlaisille sopimusmalleille (Abrahamsson ym. 2002). Näitä sopimusmalleja tarkastellaan luvussa 4. Sopimusmallit ohjelmistoprojekteissa.

2.3 Projektin elinkaari

Ohjelmistokehityksessä on tapana käyttää jotakin prosessimallia (process model) tai menetelmää (method) ohjaamaan projektin elinkaarta (project lifecycle). Menetelmästä riippumatta jokainen ohjelmistoprojekti sisältää samantapaisia vaiheita ja tehtäviä. Näitä ovat esimerkiksi määrittely, suunnittelu, toteutus, testaus, integrointi, käyttöönotto, koulutus ja ylläpito. Menetelmä tai prosessimalli määrittelee, millainen on projektin elinkaari eli miten tehtävien suorittaminen ajoitetaan projektissa, miten palautetta kerätään ja millaisia kontrollitoimenpiteitä käytetään. (Green & DiCaterino 1998). Tunnetuimpia elinkaarimalleja ovat vesiputousmalli, prototyypimalli, inkrementaalinen malli ja iteratiivinen malli (Haikala & Märijärvi 2000). Ketterän lähestymistavan mukainen projekti toteutetaan iteraatioissa ja jokainen iteraatio sisältää uusien toimintojen julkaisemiseen tarvittavia tehtäviä. Peräkkäiset julkaisut ja iteraatiot pilkkovat projektin helpommin hallittaviin osiin. Tätä havainnollistaa Sligerin ja Broderickin (2008) esittämä kuva ketterän lähestymistavan mukaisen projektin elinkaaresta (Kuva 1).



Kuva 1. Ketterän lähestymistavan mukaisen projektin elinkaari (Sliger & Broderick 2008, s. 38).

Ketterän lähestymistavan mukaisen projektin elinkaari alkaa tuotteen suunnittelulla, johon kuuluu yleinen kuvaus siitä mitä pitäisi saada aikaan (product backlog), mitkä toiminnot julkaistaan ensimmäisenä ja millä aikataululla. Tässä vaiheessa tehdään usein myös kartoitusta, liiketoimintanalyysia, standardointia, koulutusta, tutkimusta, asennuksia sekä sovitaan julkaisutahdista ja sopimusmallista. (Sliger & Broderick 2008).

Julkaisua ennen tehdään korkean tason suunnitelma ja julkaisuun vaadittavat toiminnot jaetaan toteutettavaksi noin kuuden viikon iteraatioissa. Jokaista iteraatiokierrosta ennen suunnitellaan asiakkaan kanssa mitä tehdään ja miten. Iteraatiolle määritellään aloitus- ja lopetusaika ja selvä yksityiskohtainen tehtävistä (iteration backlog). Jokaisen iteraatiokierroksen jälkeen iteraation onnistuminen arvioidaan tiimin sisällä sekä esitellään iteraation tuotokset

asiakkaalle. Asiakkaalle voidaan demota jotakin ohjelmiston osaa ja saada siitä palautetta. (Sliger & Broderick 2008).

Julkaisu voidaan tehdä myös jokaisen iteraatiokierroksen jälkeen, mutta se ei aina ole järkevää. Leffingwellin (2007) mukaan tiheä julkaisutahti tuo kilpailuetua, välitöntä asiakaspalautetta ja parempaa mahdollisuutta reagoida muutoksiin, mutta siihen liittyy myös paljon haasteita. Myyntiosastoa ja asiakkaita täytyy kouluttaa useammin, markkinointimateriaalia täytyy uudistaa jatkuvasti, yhteensopivuutta asiakkaan järjestelmien kanssa täytyy varmistaa ja niin edelleen. Tämän vuoksi julkaisu tehdään usein muutaman iteraatiokierroksen jälkeen. Iteraation jälkeen voidaan tehdä niin sanottu yrityksen sisäinen julkaisu. Julkaisuvaihtoehdot on otettava huomioon sopimusmallin valinnassa. Joka tapauksessa julkaisuja tehdään ketterän lähestymistavan mukaisissa projekteissa useammin kuin perinteisillä elinkaarimalleilla.

Iteraation tehtävälista jaetaan päivittäisiksi työtehtäviksi, joita on helppo hallita. Tiimit pitävät tavallisesti päivittäin lyhyen palaverin pysyäkseen perillä iteraation etenemisestä. Jokaisen iteraation jälkeen asiakkaalle esitellään aikaansaadut tuotokset ja tiimin kesken arvioidaan iteraation onnistumista. (Sliger & Broderick 2008).

3 SOPIMUS, RISKIENHALLINTA JA LUOTTAMUS

Yritykset tarvitsevat sopimuksia, sillä ne ovat ennen kaikkea väline hallita yrityksen riskejä. Pelkkä sopimus ei kuitenkaan aina riitä riskien ehkäisemiseen ja yhteistyön onnistumiseen vaaditaan sopijaosapuolten välille myös luottamusta. Tässä luvussa kerrotaan, mihin yritykset tarvitsevat sopimuksia, millainen on yrityksen sopimusprosessi ja miten se kytkeytyy yrityksen muihin prosesseihin. Lisäksi selvitetään, miten sopimuksilla voidaan hallita riskejä ja miten luottamus auttaa asiakkaan ja toimittajan välisen yhteistyön onnistumisessa.

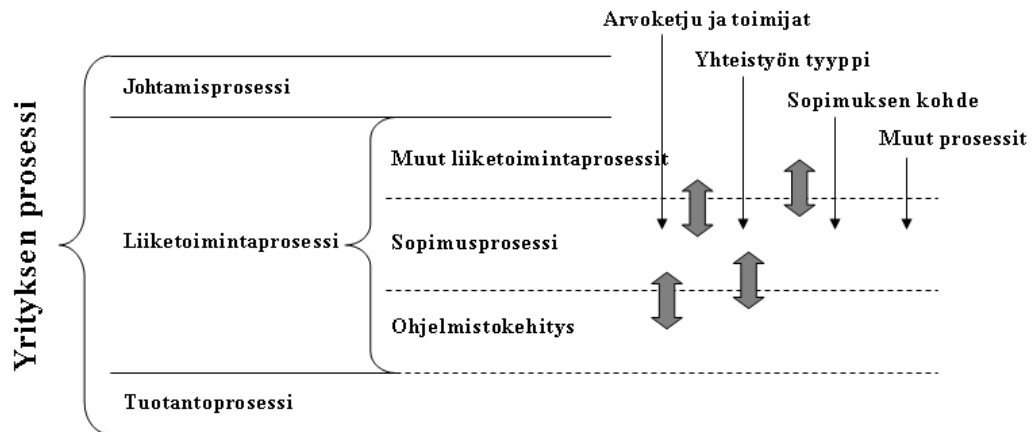
3.1 Sopimukset liiketoiminnassa

Sopimus on sitova toimi kahden tai useamman osapuolen välillä (O'Sullivan & Sheffrin 2003). Sopimusehdoilla määritellään osapuolten oikeuksia ja velvollisuuksia sekä jaetaan riskit osapuolten välillä. Sopimuksia käsitellään sopimusoikeudessa tyypikohtaisesti, jolloin niihin sovelletaan sopimustyyppien mukaisia oikeusperiaatteita. Sopimukset voidaan jaotella esimerkiksi käsitepareihin kuluttajasopimus–liikesopimus, yksilöllinen sopimus–vakiosopimus sekä kertasopimus–kestosopimus. Tässä tutkielmassa käsitellään pääasiassa yksilöllisiä kertaluonteisia liikesopimuksia. Tavallisesti liikesopimuksissa osapuolet eivät ole yksityishenkilöitä, joten sopijaosapuolia kohdellaan tasavertaisina, kun taas kuluttajasopimuksissa kuluttaja on aina heikompi osapuoli yritykseen verrattuna. Liikesopimuksen osapuolilta voidaan edellyttää syvällisempää sopimusoikeuden tuntemusta sekä riskinottovalmiutta. Riskialtis liikesopimus, joka voi johtaa joko suuriin voittoihin tai yrityksen konkurssiin, on täysin hyväksyttävä, eikä negatiivinen lopputulos ole syy sopimuksen sovitteluun. (Hemmo 2003). Tästä syystä sopimuksen tekemiseen täytyy kiinnittää erityistä huomiota liiketoiminnassa.

Ohjelmistokehitysprojekteissa sopimuksen tekeminen ei ole helppoa, sillä sopimuksen tulee täyttää sekä laillis-tekniset että liiketoiminnalliset tarpeet, mutta myös ohjelmistotekniset seikat. (Warsta & Seppänen 2000). Sopimuksia voidaan tarkastella sekä liiketoiminnan kannalta että sopimusoikeudelliselta kannalta (Nysten-Haarala 1998). Tässä tutkielmassa ei tarkemmin keskitytä sopimusoikeudellisiin seikkoihin, vaan tarkastellaan sopimukseen liittyviä liiketoiminnallisia ja ohjelmistoteknisiä näkökulmia.

3.2 Sopimusprosessi

Kaikki yrityksen toiminta pyritään nykypäivänä kuvaamaan prosesseina. Näin ollen myös sopimuksen tekeminen on yksi yrityksen monista prosesseista. Hammer ja Champy (1993) ovat määritelleet prosessin joukoksi toimintoja, joka vastaanottaa syötteitä ja jonka tuloksena syntyy asiakkaan arvostama tuote. Warsta ja Seppänen (2000) ovat jakaneet ohjelmistoyrityksen prosessin (company process) kolmeen pääprosessiin: johtamisprosessi (management process), liiketoimintaprosessi (business process) ja tuotantoprosessi (work process). Sopimusprosessi (contract process) on osa liiketoimintaprosessia, joka voidaan jakaa kolmeen aliprosessiin: sopimusprosessi, ohjelmistokehitys ja muut liiketoimintaprosessit. Kuten kuvasta 2 nähdään, muilla liiketoimintaprosesseilla on vaikutusta sopimusprosessin kulkuun. Muita vaikuttavia elementtejä ovat arvoketju ja toimijat, yhteistyön tyyppi, sopimuksen kohde ja muut prosessit. Ketterän lähestymistavan mukaisissa projekteissa arvoketjun muodostavat kehitystiimi ja asiakkaat. Arvoa tuotetaan jokaisella iteraatiokierroksella ja yhteistyö asiakkaan kanssa on tiivistä. Nämä seikat vaikuttavat ketterän ohjelmistoyrityksen sopimusprosessiin.



Kuva 2. Ohjelmistoyrityksen pääprosessit ja sopimusprosessiin vaikuttavat tekijät (Warsta & Seppänen 2000, 241).

Sopimusprosessi kattaa koko sopimuksen teon elinkaaren ensimmäisestä yhteydenotosta sopimuksen viimeistelyyn ja toteutumiseen saakka. Sopimusprosessin aikana yritysten tulee vastata seuraaviin sopimusta koskeviin peruskysymyksiin (Warsta & Seppänen 2000):

- 1) Miksi sopimus tehdään?
- 2) Milloin sopimus tehdään ja mistä siinä sovitaan?
- 3) Ketkä ovat sopijaosapuolet?
- 4) Mitkä ovat taloudelliset seikat sopimuksessa?
- 5) Mikä sopimusmalli on sopiva kyseisessä tilanteessa?
- 6) Miten jatkosta sovitaan?

Ohjelmistoprojekteissa sopimuksella voidaan varmistaa molemminpuolinen ymmärrys ja juridinen pohja projektin onnistumiselle ja hallinnalle.

Sopimuksessa sovitaan keskeiset vaatimukset, mikä lisää osapuolten realismia. Sopimuksen taloudellisia seikkoja ovat mm. maksut ja maksuehdot. Toimittamiseen liittyen sovitaan yleensä mm. maksuaikataulusta, ohjelmiston ylläpidosta, omistus- ja tekijänoikeuskysymyksistä, sopimuksen muuttamismenettelystä, sopimusrikkomusten käsittelystä ja riitatilanteiden ratkomisesta. Lisäksi on sovittava myös kehittämistyössä tarvittavien ohjelmistojen, laitteistojen, tietoliikenneyhteyksien hankkimisesta, hallinnasta ja vastuusta. (Tietotekniikan liitto 2002).

Tässä tutkielmassa keskitytään erityisesti sopimusprosessin viidenteen peruskysymykseen: Mikä sopimusmalli on sopiva kyseisessä tilanteessa? Tutkielmassa tarkastellaan, mitkä seikat tulisi ottaa huomioon sopimusmallin valinnassa ketterän lähestymistavan mukaisessa projektissa. Sopimusmallin valintaan liittyviä kysymyksiä käsitellään luvussa 4. Sopimusmallit ohjelmistoprojekteissa.

3.3 Riskienhallinta

Riskienhallinnalla voidaan tarkoittaa abstraktia toimintaa, jossa etukäteen määritellään mahdolliset liiketoimintaa koskevat uhat ja mahdollisuudet. Toisaalta riskienhallinta voi sisältää konkreettisia toimia, joilla yritystä uhkaavia vaaroja pyritään torjumaan ja niistä aiheutuvia menetyksiä pyritään minimoimaan. (Heikkala 2009). Project Management Body of Knowledge (PMBOK) (IEEE Computer Society 2004) jakaa projektinhallinnan yhdeksään tietämysalueeseen, joista riskienhallinta on yksi. PMBOK:n (2004) mukaan riskienhallintaan kuuluu riskienhallinnan suunnittelu, riskien tunnistaminen, riskien laadullinen ja määrällinen analysointi, vastatoimenpiteiden suunnittelu sekä riskien seuranta ja kontrollointi. Riskien tunnistaminen tarkoittaa todennäköisimpien projektia uhkaavien riskien tunnistamista ja niiden

erityispiirteiden dokumentointia. Riskien analysointi tarkoittaa havaittujen riskien arvioimista. Esimerkiksi voidaan arvioida riskin vaikutusta projektin lopputulokseen ja toisaalta riskiin liittyviä mahdollisuuksia. Riskien analysoinnin tarkoitus on erottaa ja priorisoida ne riskit, jotka vaativat vastatoimenpiteitä. Riskien vastatoimenpiteiden suunnitteluun kuuluu mahdollisuuksien parantaminen ja uhkiin reagoiminen. Uhkia voidaan yrittää välttää, niiden vaikutusta voidaan pienentää, ne voidaan siirtää toiselle osapuolelle tai ne voidaan hyväksyä. Riskien seuranta ja kontrollointi tarkoittaa tunnistettujen riskien seurantaa ja uusien riskien jatkuvaa tunnistamista ja analysointia. (IEEE Computer Society 2004).

Ohjelmistokehitysprojektit ovat luonteeltaan hyvin riskialttiita ja epävarmoja. Uuden tuotteen kehittäminen vaatii ohjelmistokehittäjiltä paljon itseohjautuvuutta, oppimista ja päällekkäisten kehitysvaiheiden hallinnointia. Schwaberin ja Beeblen (2002) mukaan ohjelmistoprojekteissa riskejä aiheuttavat yleisimmin seuraavat tilanteet; asiakas on tyytymätön lopulliseen tuotteeseen, kaikkia sovelluksen toimintoja ei saada valmiiksi tai kehittäjät laiminlyövät suunnittelun ja arvioinnin. Riskejä aiheutuu myös, jos ongelmia ei selvitetä ajoissa tai projekti on resursseihin nähden liian työläs tai vaatimukset muuttuvat. (Schwaber & Beeble 2002).

Cadle ja Yeates (2008) luokittelevat ohjelmistoprojekteille tyypilliset riskit kuuteen eri kategoriaan: liiketoiminnalliset riskit, asiakassuhderiskit, vaatimusriskit, suunnittelu- ja resurssiriskit, tekniset riskit ja alihankintariskit. Hyvin laaditulla sopimuksella ja oikealla sopimusmallilla oikeassa tilanteessa, voidaan välttää useimpia näistä riskeistä. Toisaalta sopimus voi itsessään synnyttää riskejä, mikäli se on tilanteeseen sopimaton tai epämääräinen. (Cadle & Yeates 2008).

Ketterän lähestymistapa ei tyypillisesti sisällä erillistä riskienhallintavaihetta tai ennalta määrättyä menettelytapaa hallita riskejä. Sen sijaan riskienhallinta toteutetaan implisiittisesti ja käytännönläheisesti. Ketterät menetelmät käyttävät empiiristä prosessia valvomaan projektin etenemistä, jolloin riskien ajatellaan vähenevän itsestään. Tiimin oletetaan tunnistavan riskit projektin aikana ja tärkeimmät toiminnot toteutetaan ensimmäisenä. (Nelson ym. 2008). Riskienhallintakeinona voidaan myös pitää tiivistä asiakasyhteistyötä, nopeaa julkaisutahtia ja lyhyitä iteraatiokierroksia (Abrahamsson ym. 2002). Myös valitun sopimusmallin tulisi tukea näitä riskienhallintakeinoja ketterän lähestymistavan mukaisissa projekteissa.

3.4 Sopimus riskienhallinnan työvälineenä

Sopimuksen tekeminen on yksi konkreettinen riskienhallintakeino. Mikäli osapuolten välillä vallitsee täydellinen keskinäinen luottamus ja osapuolet ymmärtävät tarkasti mistä he sopivat, ei kirjallista sopimusta tarvita. Liiketoiminnassa tällainen tilanne on kuitenkin erittäin harvinainen. Kirjallinen sopimus on välttämätön, sillä yritykset haluavat suojata itsensä ennalta arvaamattomien tilanteiden varalta. Sopimusten tekemisen tärkein arvo onkin, että se mahdollistaa yhteistyön myös täysin tuntemattomien osapuolten kesken. (Warsta & Seppänen, 2000).

Sopimukseen liittyy osapuolten positiivisia tavoitteita, mutta usein myös negatiivisten tapahtumien mahdollisuuksia, jotka voivat vaarantaa sopimukselle asetetut tavoitteet. Sopimuksen ensisijainen tavoite on välttää erimielisyyksistä johtuvat konfliktit ja tehokkain keino tähän on sopimusehtojen huolellinen laatiminen tulkintaongelmien välttämiseksi. Ohjelmistoprojekteissa tämä voisi tarkoittaa projektin laajuuden, laatuvaatimusten ja hinnan tarkkaa määrittämistä. Ulkopuolisten

häiriötekijöiden vaikutusta pitäisi ennakoida erilaisin olosuhdemuutoksia koskevin ehdoin. Esimerkiksi asiakkaan liiketoiminnassa tapahtuvat muutokset voivat aiheuttaa vaatimusten muuttumisen ja tarpeita sopimusehtojen uudelleenmäärittämiselle. Sopimusehdoista tulisi käydä ilmi, miten tällaisessa tilanteessa toimitaan ja kuka maksaa tästä aiheutuvat kulut. Suoritusajan pidentämisellä voidaan välttää viivästyemisestä aiheutuvia riskejä. Lisäksi sopimukseen voidaan lisätä erilaisia positiivisia kannustimia sekä sanktioita sopimusehtojen rikkomisesta. (Hemmo 2005).

Osapuolten riskinottovalmius säätelee suurelta osin sitä, kuinka tarkasti sopimus laaditaan. Täydellisen aukottoman sopimuksen tekemiseen kuluu paljon yrityksen resursseja. Jos sopimukseen joudutaan tekemään useita muutoksia, voi sopimussuhteen hallinnoiminen tulla kalliiksi. *Transaktiokustannus* (transaction cost) on kustannus, joka syntyy sopimuksen suunnittelusta, sovittelusta ja sopimuksen noudattamisen valvomisesta. (Turner & Simister 2001). Ääripäänä voidaan pitää sopimustekniikkaa, jossa hyvin epätodennäköisetkin häiriöt ja muutostekijät otetaan sopimuksessa huomioon. Tällaisen sopimuksen laatiminen synnyttää paljon niin sanottuja ennakkollisia transaktiokustannuksia. Sopimus, jossa pyritään liiallisesti hallitsemaan riskejä ja lisätään vain omaa asemaa vahvistavia ehtoja, voi pahimmassa tapauksessa johtaa sopimusneuvotteluiden päättymiseen, sillä neuvottelukumppani pitää sopimusta toista suosivana. (Hemmo 2005). Toinen vaihtoehto on tehdä sopimus nopealla aikataululla ja suurpiirteisesti, mikä voi olla taloudellisesti kannattavampaa, mutta samalla riskialtista. Niin sanotut jälkikäteiset transaktiokustannukset voivat olla suuria, jos tuotteessa havaitaan virheitä tai sopimusriitoja joudutaan selvittämään oikeudessa. (Kanninen & Määttä 1996). Toisena ääripäänä voidaan pitää sopimustekniikkaa, jossa yksinkertaisesti luotetaan siihen, että yhteistyö sujuu ja sopimusosapuolet

täyttävät velvollisuutensa ilman sopimusehtojen tarkkaa määrittämistä (Hemmo 2005).

3.5 Sopimus vastaan luottamus

Fukuyaman (2008) mukaan yritysten välisen yhteistyön odotetaan sujuvan paremmin, jos sopimuksen lisäksi osapuolten välillä vallitsee luottamus. *Luottamus* voidaan määritellä usealla eri tavalla tieteenalasta riippuen. Kaikissa määritelmissä on kuitenkin yhteisenä tekijänä ajatus siitä, että luottamus tarkoittaa luottajan uskoa siihen, ettei luottamuksen kohde petä. Markkinoinnin ja tietojärjestelmätieteiden tutkimuksissa on usein käytetty kolmea käsitettä tarkasteltaessa luottamuksen muodostumista. Nämä kolme käsitettä ovat pätevyys, hyvántahtoisuus ja rehellisyys. Pätevyydellä tarkoitetaan kykyä tehdä se, mitä on luvattu. Ohjelmistokehityksessä tämä tarkoittaa esimerkiksi sitä, että sovellus toimitetaan aikataulussa ja sellaisella osaamistasolla, mitä projekti vaatii. Hyvántahtoisuus tarkoittaa, että molemmat osapuolet toimivat toisia hyödyttävällä tavalla. Käytännössä tämä voi olla esimerkiksi hyödyllisten tietojen jakamista ilman korvausvaatimuksia. Rehellisyys on ihmisten vilpittömyyttä ja uskottavuutta sekä sitä, että henkilö pitää lupauksensa ja jakaa totuudenmukaista tietoa. (Tiainen ym. 2004).

Usein ajatellaan, että sopimuksella voidaan korvata toimittajan ja asiakkaan välisen luottamuksen puute ja sopimuksella tulisi estää kaikki keinot, joilla osapuolet voisivat käyttää toista osapuolta hyväksi (Poppendieck & Poppendieck 2003). Sopimusmalleja tarkasteltaessa kuitenkin huomataan, ettei pelkkä sopimus aina riitä. Vaikka sopimusehdot määriteltäisiin kuinka tarkasti, on esimerkiksi laatuvaatimukset ja käytettävyys vaikea kirjoittaa yksiselitteisesti. Ohjelmistoprojekteissa myös sopimuksen kohde eli sovellus ja sen toiminnot tai prosessi, jolla ohjelmisto kehitetään voivat olla epävarmoja,

eikä sopimusehtoja yksinkertaisesti pystytä määrittämään tarkasti. Jos vahingollisia toimia voitaisiin rajoittaa ennemmin parantamalla asiakassuhteen laatua kuin laatimalla tarkkoja sopimuksia, voitaisiin Hemmon (2005) mukaan huomattavasti tehokkaammin optimoida liiketoiminnan keskinäisiä hyötyjä. Luottamuksen ja yhteistyön avulla, projektit olisivat kokonaisuudessaan tehokkaampia, joustavampia ja halvempia. Sabherwalin (1999) mukaan odottamattomat muutokset voidaan hoitaa sekä sopimukseen kirjoitetuilla rakenteellisilla kontrollitoimilla että luottamuksesta syntyvällä psykologisella sopimuksella yritysten henkilöstön välillä.

Heiskasen ym. (2008) mukaan hyvässä asiakassuhteessa asiakkaan tulisi voida luottaa toimittajaan kolmella eri osa-alueella: suoritus, hinta ja käytös. Suoritus kattaa tuotteen sisällön sekä toimitusaikataulun. Asiakkaan täytyy voida luottaa, että hinta pysyy sovitunlaisena ja osapuolilla on keskinäinen ymmärrys siitä, että oma toiminta voi vahingoittaa toista. Luottamus käytökseen on suhteessa havaintoihin, joita tehdään henkilöstön käytöksestä asiakassuhteessa ilmenevissä ongelmatilanteissa.

Miten sitten voidaan luottaa toiseen yritykseen? Ohjelmistoprojektit ovat usein kertaluonteisia, eikä asiakas sen vuoksi tunne toimittajayrityksen osaamista, toimintatapoja ja arvoja. Mölleringin (2005) mukaan luottamus ja kontrolli ovat saman kolikon eri puolet. Luottamuksen olemassaoloon tarvitaan kontrollia ja päinvastoin. Oikeanlainen sopimusmalli oikeassa tilanteessa voi edesauttaa yhteistyön sujumisesta ja sitä kautta luottamuksen luomista. (Poppendieck & Poppendieck 2003).

4 SOPIMUSMALLIT OHJELMISTOPROJEKTEISSA

Ohjelmistoprojekteissa tehdään kahdenlaisia sopimuksia, koko projektia koskien asiakkaan ja ohjelmiston toimittajan välillä, ja ohjelmistoprojektin sisällä kahden ohjelmistotoimittajan välillä. Tässä tutkielmassa käsitellään ensiksi mainittuja. Tässä luvussa esitellään, mistä näkökulmista sopimusmalleja voidaan ketterän lähestymistavan mukaisissa projekteissa tarkastella ja millä perusteella sopimusmalli voidaan valita tiettyyn tilanteeseen. Luvussa esitellään perinteisesti ohjelmistoprojekteissa käytetyt sopimusmallit sekä ketterän lähestymistavan mukaisiin projekteihin käytettäväksi esitetyt sopimusmallit. Sopimukset on tässä jaoteltu perinteisiin ja ketterän lähestymistavan mukaisiin projekteihin ehdotettuihin, sillä jälkimmäisiä usein verrataan kirjallisuudessa kiinteä hinta- ja aika-ja-materiaali -sopimusmalleihin. Lisäksi kerrotaan, minkälaisia kokemuksia sopimusmalleista on saatu ketterän lähestymistavan mukaisissa projekteissa ja minkälaisia vaikutuksia sopimusmallin valinnalla on ollut asiakassuhteen laatuun ja luottamuksen luomiseen.

4.1 Sopimusmallin valintaperusteet

Perinteisen käsityksen mukaan sopimusmallin valinnan tulee perustua arvioon, kuinka suuri riski projektiin liittyy toimittajan näkökulmasta. Turnerin ja Simiesterin (2009) mukaan nykypäivänä ohjelmistoprojektit tulisi nähdä asiakkaan ja toimittajan välisenä yhteistyönä, jossa tavoitteet ovat yhdensuuntaisia ja molempien osapuolten riskit otetaan huomioon. Tämä mukailee myös Agile Manifeston (2001) arvoa, joka korostaa asiakasyhteistyötä enemmän kuin sopimusneuvotteluja.

Sopimusmallin valintaa voidaan perustella myös transaktiokustannusten suuruudella. Valitun sopimusmallin on tutkittu vaikuttavan transaktiokustannuksiin eli sopimuksen tekemiseen, kontrollointiin ja muokkaamiseen kuluviin resursseihin ja sitä kautta asiakkaalle muodostuvaan lopulliseen hintaan. (Turner & Simister 2009). Ketterän lähestymistavan mukaisissa projekteissa sopimukset laaditaan niin, että niihin voidaan tehdä muutoksia. Muutosten tekeminen saattaa lisätä projektin transaktiokustannuksia, mutta Turnerin ja Simisterin (2009) mukaan tämä ei kuitenkaan ole olennaista, sillä transaktiokustannukset ovat aina melko pieniä suhteessa ohjelmistotuotteen kokonaishintaan. Turnerin ja Simisterin (2009) mukaan hyvä tapa valita sopimusmalli olisi arvioida sopimusmallin vaikutusta osapuolten motivaatioon ja tehokkuuteen ja sitä kautta muodostuvaan kokonaishintaan. Valinnan tulisi tällöin perustua kahteen ulottuvuuteen; epävarmuus tuotteessa ja epävarmuus prosessissa.

Highsmithin (2002) mukaan ennen ohjelmistoprojektin aloittamista ja sopimuksen laatimista on vastattava kolmeen kysymykseen. Ensimmäisenä tulisi määrittää, minkälaisia ominaisuuksia ongelma-alueeseen liittyy, mikä olisi paras tapa kehittää tuote ja miten saadaan aikaan hyvä asiakassuhde. Jos näihin kolmeen kysymykseen saadaan ratkaisu, ei valitulla sopimusmallilla ole merkitystä. Sopimuksen laatimisen sijaan yritysten tulisi panostaa ketterän lähestymistavan mukaiseen toimitustapaan, asiakasyhteistyöhön ja luottamuksen luomiseen. (Highsmith 2002).

Stevens (2009a) on blogissaan listannut neljä seikkaa, jotka tulee ottaa huomioon arvioitaessa sopimusmallin soveltuvuutta ketterän lähestymistavan mukaiseen projektiin:

1. *Rakenne*: Miten sopimus on jäsennetty eli miten tuotteen toimitus ja asiakkaan laskutus hoidetaan?
2. *Muutokset*: Miten sopimuksessa käsitellään vaatimusten muuttuminen?
3. *Riskit*: Miten riskit ja palkkiot jaetaan asiakkaan ja toimittajan välillä?
4. *Asiakassuhde*: Mitä asiakassuhdetyyppejä sopimus edustaa? Asiakassuhdetyyppejä ovat kilpailu (competitive, my win is your loss), yhteistyö (cooperation, win-win), välinpitämätön (indifferent, I don't care-you lose) ja riippuvainen (dependent, heads-I-win-tails-you-lose).

Kilpaileva asiakassuhde tarkoittaa tilannetta, jossa asiakkaat haluavat käyttää projektiin mahdollisimman vähän rahaa, mutta saada siitä mahdollisimman suuren hyödyn, kun taas toimittaja haluaa tehdä mahdollisimman vähän työtä mahdollisimman suurella palkkiolla. Tässä yhteydessä puhutaan usein nollasumma-peleistä, jossa kumpikin osapuoli lopulta häviää. Yhteistyö-asiakassuhteessa kummallakin osapuolella on yhteiset tavoitteet ja ymmärrys siitä mitä tuotetaan ja paljonko siitä maksetaan. Välinpitämätön asiakassuhde voi tarkoittaa, että asiakkaalla on valta, koska toimittaja tekee usein myönnytyksiä miellyttääkseen asiakasta. Toimittaja voi taas käyttää asiakkaan tietämättömyyttä hyväksi laskuttamalla asiakasta liikaa. Riippuvainen asiakassuhde tarkoittaa, että osapuolet ovat riippuvaisia toisistaan ja projektin päättyminen liian aikaisin olisi jommallekummalle vahingollista. (Stevens 2009b).

Näitä neljää kriteeriä, rakenne, muutokset, riskit ja asiakassuhde, tullaan käyttämään myös tässä tutkielmassa sopimusmallien jäsentämiseen ja vertailemiseen, sillä niiden avulla voidaan kattavimmin arvioida sopimusmallin soveltuvuutta ketterän lähestymistavan mukaisten projekteihin. Ketterän lähestymistavan mukaisissa projekteissa sopimusmallin tulisi olla

rakenteeltaan iteratiivista ohjelmistokehitystapaa tukeva ja muutokset salliva. Riskit ja palkkiot tulisi jakaantua tasaisesti ja sopimuksen tulisi edustaa asiakassuhdetyypiltään yhteistyötä.

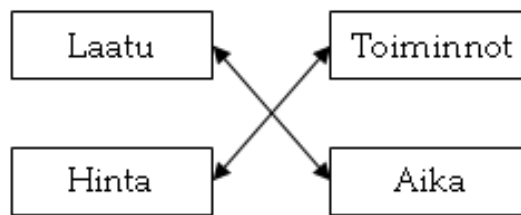
4.2 Perinteiset sopimusmallit

Ohjelmistoprojekteissa on perinteisesti käytetty seuraavia sopimusmalleja:

Kiinteähintaisessa (fixed price) sopimusmallissa asiakas maksaa vaatimusmäärittelyä vastaavasta tuotteesta täsmälleen ennalta sovitun hinnan toimituksen jälkeen. (Franklin 2008). Sopimukseen määritellään usein projektin laajuus, hinta ja aikataulu. Kiinteähintainen sopimus miellyttää usein asiakkaita, sillä asiakas saa tarkan sopimuksen ja hinnan ja toimittaja ottaa kaiken vastuun riskeistä. (Poppendieck & Poppendieck 2003). Jos aikataulu tai kustannukset ylitetään, joutuu toimittaja maksamaan ylimenevät kustannukset (Thorup & Jensen 2009).

Kiinteä hinnoittelu yhdistetään usein vesiputousmalliin, joten se saa osakseen samankaltaista kritiikkiä. Franklinin (2008) mukaan vaatimusten mukainen tuote ei aina ole laadukas tai käytettävä, eikä asiakas välttämättä saa sitä mitä halusi. Kiinteä hinnoittelu voi myös tulla asiakkaalle kalliiksi ja pahimmassa tapauksessa johtaa projektin epäonnistumiseen. Myös Thorupin ja Jensenin (2009) mukaan jompikumpi on tavallista toimittajan ammattitaidosta riippuen. Kokenut toimittajayritys tiedostaa kiinteän hinnoittelun riskit ja sen vuoksi käyttää niissä erilaisia itsepuolustuskeinoja. Kokenut toimittaja osaa ottaa huomioon kustannusten ja aikataulun ylittymismahdollisuuden asiakkaalle tekemässään tarjouksessa, jolloin hinta on korkeampi. Muutostenhallinta toteutetaan niin, että kaikesta työstä, mitä ei ole mainittu sopimuksessa, veloitetaan erikseen. Vähemmän kokenut toimittaja taas arvioi projektin aikataulun liian tiukaksi, ei vaadi tarpeeksi selkää vaatimusmäärittelyä, eikä

osaa veloittaa muutoksista. Virhearvioinnit johtavat siihen, että kun asiakkaan vaatimukset lisääntyvät, aikarajat tulevat vastaan ja kustannukset kasvavat, joudutaan tinkimään tuotteen laadusta. (Thorup & Jensen 2008). Samat ongelmat pätevät sopimusmalleihin kiinteä aika (fixed time) ja kiinteä laajuus (fixed scope), joita ei tässä tutkielmassa tarkemmin käsitellä. Ohjelmistoprojektin muuttujien suhdetta kuvaa hyvin Stevensin (2009b) blogissaan esittämä kuva (Kuva 3), jossa ohjelmistoprojektissa tasapainottelevat neljä muuttujaa; toiminnot, aika, hinta, ja laatu. Esimerkiksi järjestelmässä edellytettävien toimintojen lisääntymisen pitäisi vaikuttaa lisäävästi myös hintaan ja aikaan, jotta laatu pysyisi vakiona.



Kuva 3. Neljä muuttujaa ohjelmistoprojektissa (Stevens 2009b).

Kiinteähintaisissa projekteissa asiakassuhde muodostuu usein kilpailuksi tai välinpitämättömäksi. Asiakkaat haluavat käyttää projektiin rahaa vain sen mitä on sovittu, vaikka vaatimukset ohjelmiston toimintojen suhteen usein kasvavat projektin edetessä. Toimittaja taas haluaa tehdä mahdollisimman vähän työtä mahdollisimman suurella palkkiolla. Suhteesta voi tulla myös välinpitämätön, sillä asiakkaan on helppo käyttää hyväksi toimittajaa, joka haluaa asiakkaan olevan tyytyväinen. (Stevens 2009b). Kiinteä hinnoittelu ei yleensä sovellu ketterän lähestymistavan mukaisiin projekteihin, sillä se ei varaudu muutoksiin. Toisaalta kiinteää hinnoittelua voidaan käyttää, mikäli projektin

laajuus on pieni, vaatimukset tiedetään varmasti ennalta ja riskit ovat minimaaliset. (Franklin 2008).

Aika-ja-materiaali (time and materials) -sopimusmallissa asiakasta veloitetaan ennalta määrättyllä tunti hinnalla käytettyjen työtuntien mukaan. Lisäksi asiakas maksaa kaikki ohjelmistoprojektista syntyneet materiaalikustannukset, kuten laitteet ja ohjelmistot, joita tietyn projektin toteuttamiseen tarvitaan. Aika-ja-materiaali sopimusmalli on kehitetty epävarmuuden hallitsemiseksi, mutta se ei silti poista riskejä, vaan siirtää ne toimittajalta asiakkaalle. (Poppendieck & Poppendieck 2003). Aika-ja-materiaali -sopimukset ovat yleensä hyviä toimittajan kannalta, sillä asiakas ottaa kaiken vastuun riskeistä. Ilman aikarajoitteita ja tarkkaa budjettia työtä tehdään toimittajan aikataulujen ja intressien mukaisesti. Kun toimiva ohjelmisto saadaan toimitettua, asiakasta voidaan laskuttaa joka ikisestä projektiin käytetystä työtunnista. (Eckfeldt ym. 2005). Asiakas ei tiedä, käyttääkö toimittaja aikansa tehokkaasti, mikä voi johtaa erilaisiin yhteistyötä häiritseviin kontrollitoimiin (Thorup & Jenssen 2009).

Mikäli lopputulos ei miellytä asiakasta tai hän haluaa muuttaa vaatimuksia projektin aikana, ei toimittajalle synny riskejä vaan lisää tuloja. Mitä enemmän toimittajalle tulee lisätyötä, sitä enemmän hän voi veloittaa asiakasta. Tätä voidaan jatkaa, kunnes asiakkaan vaatimukset loppuvat tai hän ei enää halua maksaa enempää. Jälkimmäinen voidaan käytännössä ajatella projektille asetettuna hintakattona. Hintakatto aiheuttaa usein sen, että toimittaja pyrkii laskuttamaan hintakatoksi määritellyn hinnan, vaikka projekti olisi voitu toteuttaa todellisuudessa nopeammin ja halvemmalla. (Stevens 2009b). Ainoa riski toimittajalle on projektin päättymisen liian nopeasti, sillä toimittaja joutuu ennakoimaan myös omia resurssitarpeitaan. Aika-ja-materiaali -sopimusmallissa päädytään usein selvittelemään, mitkä kustannukset kuuluvat asiakkaalle maksettavaksi ja mitkä eivät, mikä lisää transaktiokustannuksia.

(Poppendieck & Poppendieck 2003). Asiakassuhteesta muodostuu usein välinpitämätön (Stevens 2009b).

Aika-ja-materiaali -sopimusten heikkoudet korostuvat projekteissa, jotka toteutetaan vaiheittain. Ketterän lähestymistavan mukaiset projektit toteutetaan iteraatioina, jolloin toimivia sovellusosia voidaan tuottaa alusta asti. Tämä vähentää asiakkaan riskejä, sillä asiakas voi määrittellä tärkeimmät toiminnot toteutettaviksi alussa. (Poppendieck & Poppendieck 2003). Aika-ja-materiaali -sopimusmalli voi soveltua ketterän lähestymistavan mukaisiin projekteihin, mikäli asiakas osaa priorisoida vaatimuksensa tarpeeksi hyvin tai asiakkaalla on paremmat mahdollisuudet kantaa riskit (Stevens 2009b).

Monivaiheinen sopimus (multi-stage contract) on kehitetty hallitsemaan epävarmuutta ja riskejä. Monivaiheisia sopimuksia on kahdenlaisia; sopimukset, jotka johtavat lopulta kiinteä hintaiseen sopimukseen, ja sopimukset, jotka säilyttävät monivaiheisuuden alusta loppuun. Ensimmäiseksi mainitussa projekti alkaa usein yhdellä tai kahdella lyhyellä sopimuksella, joiden tavoitteena on oppia asiakkaasta sekä ongelma-alueesta ja tällä tavoin luoda luottamusta ja vähentää riskejä. Tämän jälkeen kiinnitetään vaatimukset, hinta ja aikataulu. Monivaiheinen sopimus, joka johtaa kiinteähintaiseen sopimukseen, sisältää samat riskit kuin kiinteähintainen sopimus, mikäli vaatimuksiin halutaan muutoksia. (Poppendieck & Poppendieck 2003).

Sopimus, joka säilyy monivaiheisena, soveltuu hyvin iteratiivisesti toteutettavaan projektiin. Tällöin jokaiselle iteraatiokierrokselle voidaan tehdä oma kiinteähintainen sopimus. Muutokset ovat siis mahdollisia myöhemmille iteraatioille, mutta täysin riskittömästä mallista ei tässäkään ole kyse. Suurin riski on, että toinen osapuolista lopettaa asiakassuhteen kesken projektin. Sopimusmallin yhteydessä puhutaankin molemminpuolisesta monopolista, jossa kumpikin osapuoli on riippuvainen toisesta. Asiakasyritys saattaa

esimerkiksi käyttää paljon resursseja projektin alussa määrittelyyn ja suunnitteluun. Jos toimittaja vetäytyy projektista, asiakkaalle syntyy määrittelytyöstä turhia kustannuksia. Toimittaja taas voi tehdä asiakkaan kanssa määrittelytyötä tappiolla siinä toivossa, että ennen pitkää asiakkaan kanssa syntyy pitkäkestoinen kannattava projekti. Jos näin ei tapahdu, syntyy toimittajalle turhia kustannuksia. Poppendieckin ja Poppendieckin (2003) mukaan hyvä tapa estää asiakkaan vetäytyminen olisi ketterän lähestymistavan mukaisesti tuottaa asiakkaalle arvoa jokaisella iteraatiokierroksella ja priorisoida toiminnot siten, että tärkeimmät tehdään ensimmäisenä. Toinen tapa Highsmithin (2002) mukaan olisi luoda luottamusta, jolloin osapuolet voivat luottaa siihen, että suhde jatkuu, mikäli odotukset täytetään.

Monivaiheisen sopimuksen transaktiokustannukset voivat kasvaa suuriksi, jos sopimus joudutaan tekemään jokaiselle iteraatiokierrokselle erikseen. Tämän vuoksi tehdään usein pääsopimus ja jokainen työtilaus toteutetaan sen mukaisesti. (Pitette 2001).

4.3 Sopimusmallit ketterän lähestymistavan mukaisiin projekteihin

Ketterän lähestymistavan mukaisiin projekteihin on esitetty käytettäväksi muutamia vaihtoehtoisia sopimusmalleja (Poppendieck & Poppendieck 2003). Nämä sopimusmallit ovat kehitetty vastaamaan edellä esiteltyjen sopimusmallien ongelmiin ja jakamaan yhteistyöstä aiheutuneet menetykset ja voitot tasaisesti osapuolten kesken. Tasapuolista, reilua ja riskitöntä sopimusta ei ehkä koskaan voida saavuttaa, sillä kuten aiemmassa kohdassa todettiin, vaaditaan sopimuksen lisäksi osapuolten välille myös luottamusta. Näiden sopimusmallien tarkoitus on toimia hyvänä pohjana toimivan yhteistyösuhteen kehittämiseen osapuolten välille. Sopimusmalleista on olemassa useita eri variaatioita, mutta tässä tutkielmassa esitellään kolme rakenteeltaan toisistaan

poikkeavaa sopimusmallia. Nämä ovat tavoitehintasopimus, tavoiteaikasopimus (Poppendieck & Poppendieck 2003) ja yhteistyösopimus (Thorup & Jensen 2009).

Tavoitehintasopimus (target-cost contract) -sopimusmallissa asiakas ja toimittajat määrittävät yhdessä hinnan, mitä projekti kokonaisuudessaan, muutokset mukaan lukien, saisi maksaa. Jos tavoitehintaa ylitetään, toimittaja ja asiakas maksavat ylimenevät kustannukset puoliksi. Jos tavoitehintaa alitetaan, osapuolet jakavat ansaitut säästöt keskenään. Toimittaja ei siis voi saada lisävoittoa ylimenevästä ajasta, kuten aika-ja-materiaali -sopimuksissa. (Poppendieck & Poppendieck 2003).

Tavoitehintaan perustuvassa ohjelmistokehityksessä sovitaan ensin, mitä projektissa tulisi saada aikaiseksi. Tähän voidaan käyttää esimerkiksi asiakkaan tekemiä yleisen tason käyttäjätarinoita, joista voidaan arvioida, kuinka kauan haluttujen toimintojen toteuttamiseen menee aikaa. (Eckfeldt ym. 2005). Tarkkoja yksityiskohtia ei vaadita, sillä sopimusmallissa voidaan ottaa huomioon projektin aikana tapahtuvat muutokset (Poppendieck & Poppendieck 2003). Tämän jälkeen asiakas ja toimittaja sopivat projektin tavoitehinnan, johon voidaan sisällyttää satunnaisuusprosenttiosuus ennalta arvaamattomien tilanteiden varalle sekä asennuksiin, kokouksiin ja käyttöliittymäsuunnitteluun menevä aika (Eckfeldt ym. 2005).

Sopimusmalli kannustaa asiakkaita pitämään vaatimusmäärittelyn realistisena ja toimittajia alittamaan projektin kustannukset tavoitteisiin nähden. Tavoitehintaan pääseminen vaatii sekä kehittäjien että käyttäjien sitoutumisen projektiin, mikä edesauttaa halutunlaisen tuotteen syntymistä ja projektin onnistumista. (Poppendieck & Poppendieck 2003). Asiakkaalle ehto kulujen tai säästöjen jakamisesta on kannustava. Eckfeldtin ym. (2005) kirjoittamassa käyttökokemusraportissa mainitaan kolme syytä, miksi eräs asiakas piti

tavoitehintasopimuksesta: 1) Toimittajan voiton vaarantaminen saa toimittajan sitoutumaan projektiin. 2) Mahdollisuus vaikuttaa kustannusten suuruuteen yksinkertaistamalla vaatimuksia ilman, että laatu kärsii. 3) Toimittaja ei voi pitkittää projektia vain laskuttaakseen enemmän. (Eckfeldt ym. 2005). Toimittajalla ei tavallisesti ole intressejä alittaa tavoitehintaa ilman sovittua erityispalkkioita. Toimittajan motivoimiseksi on olemassa kaksi tapaa (Poppendieck & Poppendieck 2003):

- Kustannukset ja kiinteä palkkio (cost plus fixed fee): Toimittajan saamaa voittoa ei lasketa mukaan tavoitehintaan. Toimittaja saa ennalta sovittun kiinteän palkkion, kun tuote on onnistuneesti toimitettu. Jos tavoitehintaa on ylitetty, kulut jaetaan ja toimittajan voitto on pienempi. Jos tavoitehintaa alitetaan, säästöt jaetaan, jolloin toimittajan voitto on suurempi.
- Voittoa, jos kustannuksia ei ylitetä (profit not to exceed): Tavoitehintaa sisältää toimittajan voitto-osuuden. Kun tavoitehintaa ylitetään, toimittaja tekee työtä omakustannushintaan. Alittaessaan tavoitehinnan, toimittaja saa ennalta sovittun lisäpalkkion.

Tavoitehintasopimuksissa muutoksia voidaan hallita yhdessä asiakkaan kanssa. Sopimusmalli kannustaa molempia osapuolia harkitsemaan vaatimusten muuttumisen vaikutusta sekä hintaan että aikaan ja näin ollen pohtimaan asiakkaan todellisia tarpeita verrattuna kustannuksiin. (Eckfeldt ym. 2005). Asiakassuhdetta voidaan pitää yhteistyönä (Stevens 2009b).

Tavoitehintasopimukset soveltuvat Eckfeldtin ym. (2005) käyttökokemusten perusteella sekä suuriin että pieniin projekteihin. Pienemmissä projekteissa aikataulun ylittämisestä aiheutuvat kustannukset ovat kuitenkin merkittävämpiä kuin esimerkiksi vuoden kestävässä projektissa. Pienemmissä projekteissa ei tämän vuoksi suhtauduta muutoksiin suopeasti, sillä niissä ei ole

aikaa arvioida muutosten vaikutusta kustannuksiin yhtä tarkasti kuin suurissa. (Eckfeldt ym. 2005). Vaatimukset eivät voi muuttua oleellisesti, sillä tavoitehintaan on tällöin mahdoton päästä ja toimittajan voitto on pienempi, mikäli tavoitehintaa ylitetään (Stevens 2009b).

Tavoiteaikasopimus (target-schedule contract) - sopimusmalli on rakenteeltaan tavoitehinnan kaltainen, mutta sopimukseen määritellään tarkka aikaraja ja kustannukset jätetään avoimeksi. Tavoiteaikasopimusta käytetään tilanteissa, joissa aikataulu on tärkeämpi kuin hinta. Riskien välttämiseksi vaatimukset priorisoidaan yleensä siten, että välttämättömimmät toiminnot ja tärkeimmät ohjelmaosat julkaistaan ensimmäiseksi. Tarvittavat resurssit, kuten työvoima ja kustannukset, määritellään aikataulun mukaan niin, että tuote voidaan toimittaa aikataulussa. Internet ajalle on tyypillistä, että tuote on saatava markkinoille hyvin nopealla aikataululla. Tällöin kustannukset voivat olla sivuseikka. Tilanteissa, joissa aika on todellakin tärkein kriteeri projektin onnistumiselle, toimittajalla on mahdollisuus esimerkiksi palkata lisätyövoimaa tavoiteajan alittamiseksi. Jos aikataulu on kiinnitetty, resurssit pysyvät samana, eikä projekti vaadi suurempia hankintoja, tavoiteaikasopimus ja tavoitehintasopimus ovat sama asia. (Poppendieck & Poppendieck 2003).

Tavoiteaikasopimukseen liittyy riski, että asiakas ei todellisuudessa tarvitse tuotetta sovittuun päivämäärään mennessä. Tällöin asiakas ei välttämättä sitoudu projektiin toivotulla tavalla ja tavoiteajan ylittyessä toimittaja joutuu tekemään työtä halvemmalla. (Stevens 2009b).

Ketterän lähestymistavan mukaisissa projekteissa on aikataulu suuressa roolissa. Tiedyt toiminnot sovitaan tehtäväksi tietyn ajanjakson sisällä. Tavoiteaikasopimus antaa mahdollisuuden toteuttaa projekti iteraatioissa ja nopealla julkaisutahtilla. Tämän vuoksi tavoiteaikasopimus sopii sellaisiin

ketterän lähestymistavan mukaisiin projekteihin, joissa aika on hintaa tärkeämpi.

Yhteistyösopimus (collaborative contract, hybrid contract) on kehitetty tukemaan ketteriä menetelmiä sekä tehokasta yhteistyötä asiakkaan ja toimittajan välillä. Sopimusmallissa hinta muodostuu kahdesta osasta, aika-ja-materiaali - tyyppisestä tuntihinnasta sekä tuotteen valmistumishinnasta (completion price). Jälkimmäinen maksetaan vasta, kun tietyt kriteerit tuotteen suhteen on täytetty. Aikataulu ei kuulu kriteereihin kuten muissa sopimusmalleissa, vaan tärkein kriteeri on lisäarvon tuottaminen. (Thorup & Jensen 2009).

Yhteistyösopimukseen kirjoitetaan asiakkaan määrittämät vaatimukset vapaamuotoisesti muutamalla kappaleella. Tämän jälkeen vaatimukset jaetaan osiin (milestone), jotka sisältävät tietyt toiminnot. Jokainen osa täytyy voida toteuttaa ja toimittaa erikseen. Asiakas maksaa tehdystä työstä sovitun tuntihinnan mukaan, joka on usein alhaisempi kuin normaali aika-ja-materiaali -perusteinen tuntihinta. Jokaiselle tuotteen osalle sovitaan myös kiinteä hinta, joka laskutetaan asiakkaalta vasta kun toiminnot ovat vaatimusten mukaisia ja tuotteen osa on asiakkaalla käytössä. Koko projektille sekä jokaisen tuotteen osan toteuttamiselle voidaan tehdä alustava aikataulu, mutta aikarajoja voidaan tarvittaessa muuttaa. (Thorup & Jensen 2009).

Thorupin ja Jensenin (2009) mukaan hinnan määrittäminen on haastavaa yhteistyösopimuksissa. Jos tuntihinta on pieni ja valmistumishinta on suuri, sopimus muistuttaa kiinteä hintaista sopimusta. Jos taas tuntihinta on korkea ja valmistumishinta pieni, sopimus muistuttaa aika-ja-materiaali -sopimusmallia. Yhteistyösopimukseen saattaa siten liittyä perinteisten sopimusmallien kaltaisia riskejä, mikäli sopivaa tasapainoa ei löydetä. Jos aika on tärkeä kriteeri, kannattaa valita korkea valmistumishinta, sillä se motivoi sekä kehittäjiä että asiakkaita tehokkaampaan yhteistyöhön ja tuote saadaan valmiiksi nopeammin.

Jos tarkkaa aikarajaa ei ole ja asiakkaan vaatimukset eivät ole selvät, kannattaa asettaa korkeampi tuntihinta. Tämä varmistaa toimittajalle suuremman voiton, mutta antaa myös asiakkaalle mahdollisuuden muuttaa vaatimuksia helpommin kuin kiinteähintaista sopimusta muistuttavalla hinnoittelulla. (Thorup & Jensen 2009.)

Franklinin (2008) mukaan yhteistyösopimuksen suurin haaste on jakaa toiminnot osa-alueisiin, jos vaatimukset ovat vielä epävarmoja. Sopimusmalli parantaa kuitenkin asiakasyhteistyötä, sillä asiakkaalta saadaan palautetta jokaisen osa-alueen toimituksen yhteydessä, vaikka vaatimukset olisivat jo kiinnitetty. (Franklin 2008).

Yhteistyösopimus sopii ketterän lähestymistavan mukaisiin projekteihin, sillä se tukee iteraatioissa toteutettavaa ohjelmistokehitystä. Tuotteen osat voidaan priorisoida ja tuote voidaan toimittaa asiakkaalle pienissä osissa. Iteraatioissa toteutettava projekti mahdollistaa myös vaadittujen toimintojen siirtämistä seuraavalle kierrokselle, mikäli ne jäävät keskeneräisiksi. (Stevens 2009b).

4.4 Yhteenveto sopimusmalleista

Alla olevassa taulukossa (Taulukko 1) esitellään sopimusmallien olennaisimmat erot ja tilanteet, joissa sopimusmalli soveltuu ketterän lähestymistavan mukaiseen projektiin.

Taulukosta nähdään, että kokenut toimittajayritys voi käyttää kiinteää hinnoittelua ketterän lähestymistavan mukaisen projektin ollessa pienimuotoinen, esimerkiksi yhden iteraatiokierroksen pituinen, ja vaatimusten ollessa selkeät. Toimittajan täytyy kuitenkin osata arvioida kustannukset tarkasti ja kantaa riskit. Asiakassuhteesta voi tulla kilpaileva, mikäli asiakkaan vaatimukset lisääntyvät projektin aikana, tai välinpitämätön, mikäli asiakas

käyttää toimittajan asiakaspalvelutahtoa hyväksi tai vastavuoroisesti toimittaja laskuttaa asiakasta aivan kaikesta, mitä sopimuksessa ei ole mainittu.

Suurempaan ketterän lähestymistavan mukaiseen projektiin voisi harkita monivaiheista sopimusta, missä jokaiselle iteraatiokierrokselle tehdään oma kiinteähintainen sopimus. Jokaiseen tehtyyn sopimukseen, liittyy kiinteähintaisen sopimuksen riskit, mutta ne ovat pienemmät, sillä projekti pilkotaan pieniin, helpommin hallittaviin kokonaisuuksiin. Tavallisesti asiakassuhteesta voi tulla monivaiheisella sopimusmallilla riippuvainen, mutta tuottamalla asiakkaalle arvoa jokaisella iteraatiokierroksella ketterän lähestymistavan mukaisesti, tämä voidaan välttää.

Ketterän lähestymistavan mukaisissa projekteissa voidaan hyvin käyttää aika- ja materiaali -sopimusta, mikäli asiakas siihen suostuu. Toimittajan ja asiakkaan välille vaaditaan tässä mallissa paljon luottamusta, sillä toimittajan on helppo laskuttaa asiakasta epärehellisesti. Vaatimukset saavat muuttua, mutta asiakkaalla tulee olla mahdollisuudet kantaa tästä aiheutuva riskit. Vaatimukset tulisi voida priorisoida riskien välttämiseksi niin, että tärkeimmät toiminnot tehdään ensin.

Tavoitehinta-sopimuksista on saatu hyviä kokemuksia erityisesti suurissa ketterän lähestymistavan mukaisissa projekteissa. Tällöin vaatimuksia ei tarvitse tietää kovin tarkasti etukäteen. Tässä mallissa vaatimusten muuttuminen ja budjetista poikkeaminen vaikuttaa yhtäläillä sekä asiakkaaseen että toimittajaan, jolloin riskit voidaan jakaa tasaisesti ja asiakassuhteesta muodostuu yhteistyönomainen. Asiakkaan todellisia tarpeita voidaan arvioida yhdessä ja suunnitella ne jokaiselle iteraatiokierrokselle erikseen. Sopimusmallissa joudutaan kuitenkin määrittämään erikseen toimittajaa motivoivat kannustimet.

Jos projektin aikataulu on kiireellinen, on tavoiteaika hyvä vaihtoehto. Koska kustannuksia ei kiinnitetä, voidaan resurssitarpeet sovittaa aikatauluun ja ohjelmiston laajuuteen. Vaatimusten muuttuminen vaikuttaa kuitenkin siihen, voidaanko tavoiteaikaan päästä. Tätä voidaan arvioida asiakkaan ja toimittajan välisellä yhteistyöllä. Yhteistyön onnistuminen riippuu siitä, onko aika todellisuudessa asiakkaalle tärkeä kriteeri vai yrittääkö asiakas vain saada työn toteutettua halvemmalla, mitä kauemmin se kestää. Tällöin asiakassuhteesta voi tulla välinpitämätön.

Myös yhteistyösopimusmallista on saatu hyviä kokemuksia ketterän lähestymistavan mukaisissa projekteissa ja se on sopimusmalleista kaikista joustavin. Sopimusmalli tukee iteraatioissa toteutettavaa ohjelmistokehitystä tuottaen asiakkaalle arvoa alusta asti sekä varmistaen toimittajalle tasaisen voiton. Vaatimukset voivat muuttua, sillä asiakas kantaa riskit tuntihintaisesta osasta. Toimittajaa motivoi kiinteähintainen valmistumispalkkio, jolloin ohjelmiston osa valmistuu todennäköisemmin aikataulussa. Mikäli aika on tärkeä kriteeri, kannattaa määrittää suuri valmistumispalkkio ja pieni tuntihinta. Jos taas vaatimukset ovat epäselvät eikä tarkkaa aikataulua ole määritetty, kannattaa suosia suurta tuntihintaa ja pientä valmistumispalkkiota.

Sopimusmalli	Rakenne	Vaatimukset	Riskit	Asiakassuhde	Soveltuvuus KL: aan
Kiinteä hinta	Asiakas maksaa sovitun summan	Eivät voi muuttua	Kokematon toimittaja kantaa riskit, kokenut osaa suojautua	Kilpaileva tai välinpitämätön	Soveltuu, mikäli vaatimukset ovat selvät ja projektin laajuus on pieni
Aika-ja-materiaali	Asiakas maksaa käytettyjen työtuntien mukaan	Saavat muuttua	Asiakas kantaa riskit	Välinpitämätön	Soveltuu, mikäli vaatimukset ovat priorisoitavissa ja asiakkaalla on mahdollisuus kantaa riskit
Monivaiheinen sopimus	Projektissa tehdään monta pienempää kiinteähintaista sopimusta	Saavat muuttua, mikäli monivaiheisuus säilyy koko projektin ajan	Jompikumpi kantaa, "Molemmipuolinen monopoli"	Riippuvainen	Soveltuu, jos osapuolet eivät tunne toisiaan ja sopimus säilyttää monivaiheisuuden koko projektin ajan.
Tavoitehintaa	Asiakas ja toimittaja jakavat tavoitehinnasta poikkeavat kustannukset tai säästöt	Eivät voi muuttua oleellisesti	Riskit ja palkkiot jaetaan asiakkaan ja toimittajan kesken	Yhteistyö	Soveltuu suuriin projekteihin, toteutus iteraatioissa
Tavoiteaika	Aika kiinnitetty, kustannukset ja vaatimukset eivät.	Eivät voi muuttua oleellisesti	Riskit ja palkkiot jaetaan asiakkaan ja toimittajan kesken	Yhteistyö tai välinpitämätön	Soveltuu, mikäli aikataulu on kustannuksia tärkeämpi
Yhteistyösopimus	Asiakas maksaa tuntihinnan ja valmistumishinnan	Saavat muuttua	Jompikumpi kantaa riippuen tuntihinnasta ja valmistumishinnasta	Yhteistyö	Soveltuu

Taulukko 1: Sopimusmallien vertailu (KL = ketterä lähestymistapa).

5 YHTEENVETO

Ketterien menetelmien hyödyt ovat monelle ohjelmistokehittäjälle tuttuja, mutta asiakkaan kanssa on koettu olevan haastavaa tehdä sopimus, joka tyydyttää molempia osapuolia ja tukee samalla ketterää ohjelmistokehitystä. Tämän työn tavoite oli löytää ne sopimusmallit, jotka sopivat parhaiten ketterän lähestymistavan mukaisiin projekteihin sekä selvittää, voiko oikealla sopimusmallilla luoda luottamusta yritysten välille. Ensimmäinen tutkimuskysymys oli: Mitkä ketterän lähestymistavan piirteet tekevät kustannusten arvioinnin, hinnoittelun ja sopimuksen tekemisen haasteelliseksi? Ketterän lähestymistavan mukaisissa projekteissa vaatimukset saavat lähtökohtaisesti muuttua projektin aikana ja ketterissä menetelmissä keskitytään enemmän asiakasyhteistyöhön kuin sopimusneuvotteluihin. Lisäksi julkaisutahti on ketterissä menetelmissä nopeampi kuin perinteisissä ohjelmistoprojektien elinkaarimalleissa. Nämä piirteet ovat luoneet tarpeita uudentlaisille sopimusmalleille. Sopimuksen pitäisi olla rakenteeltaan iteraatioissa toteutettavaa ohjelmistokehitystä tukeva, riskien pitäisi jakautua tasaisesti osapuolten välille ja muutosten tulisi olla mahdollisia. Tutkielmassa selvisi, että sopimus on keino hallita yrityksen riskejä, mutta sopimuksen lisäksi yritysten välille tarvitaan luottamusta, sillä täysin aukotonta sopimusta on mahdoton tai kannattamatonta laatia. Luottamuksen luomisella voidaan parantaa asiakasyhteistyötä, jonka avulla projektit todennäköisesti onnistuvat paremmin.

Toinen tutkimuskysymys oli: Minkälaisia sopimusmalleja perinteisesti käytetään ohjelmistokehityksessä? Perinteisiä sopimusmalleja ovat kiinteä hinta, aika-ja-materiaali ja monivaiheinen sopimus. Kiinteähintaisessa ja monivaiheisessa sopimusmallissa toimittaja pääsääntöisesti kantaa riskit, kun taas aika-ja-materiaali - sopimusmallissa asiakas kantaa riskit. Riskien

epätasainen jakaantuminen voi häiritä asiakkaan ja toimittajan välistä yhteistyötä ja sitä kautta luottamuksen luomista.

Kolmas tutkimuskysymys oli: Mitä sopimusmalleja on esitetty käytettäväksi tai on käytössä ketterän lähestymistavan mukaisissa projekteissa ja onko niistä ollut apua luottamuksen luomisessa? Ketterän lähestymistavan mukaisiin projekteihin on esitetty käytettäväksi muun muassa tavoitehinta-, tavoiteaika- ja yhteistyösopimusmalleja. Kokemukset näistä ovat olleet pääasiassa positiivisia, ja yritykset ovat parantaneet niiden avulla asiakastyytyvyyttä. Luottamuksen luominen on monimutkainen prosessi, mutta jos sopimusmalli parantaa asiakasyhteistyötä, voidaan sen olettaa myös lisäävän luottamusta.

Kaikkiin ketterän lähestymistavan mukaisiin projekteihin soveltuvaa yleispätevää sopimusmallia ei tässä tutkielmassa löydetty. Jokaisen projektin yhteydessä on etsittävä tilanteeseen parhaiten sopiva sopimusmalli. Kaikkia tässä tutkielmassa esiteltyjä sopimusmalleja voi käyttää tietyissä tilanteissa myös ketterän lähestymistavan mukaisissa projekteissa. Ketterää menetelmää käyttävä toimittajayritys, joka on kokenut sopimusneuvottelut hankalaksi asiakkaiden kanssa, voi tämän tutkielman avulla kuitenkin löytää uusia vaihtoehtoja käyttämälleen sopimusmallille. Tutkielmasta voi löytää oman projektin ominaispiirteisiin parhaiten soveltuvan sopimusmallin. Näitä ominaispiirteitä, ovat esimerkiksi vaatimusmäärittelyn tarkkuus, projektin koko, julkaisu tiheys ja asiakkaan tuttuus. Tutkielmaa voi käyttää myös ohjelmistoa hankkiva asiakasyritys, jonka on myös kartoitettava sopimusmallista aiheutuvat riskit, kun projekti toteutetaan käyttämällä ketterää lähestymistapaa.

Tämä tutkielma on joiltakin osin rajoittunut, sillä tässä tutkielmassa ei esitelty kaikkia mahdollisia sopimusmalleja ja näiden variaatioita. Jokaista projektia varten täytyy usein laatia räätälöity sopimus ja muodostaa eri sopimusmallien

yhdistelmiä. Tässä tutkielmassa ei myöskään otettu kantaa siihen, kuinka projektin kustannukset lasketaan ja mikä vaikutus sillä olisi sopimusmallin valintaan. Julkiset, tietyn summan ylittävät hankinnat on aina kilpailutettava ja julkisella sektorilla on totuttu tekemään kiinteähintaisia sopimuksia. Julkisia hankintoja ei myöskään käsitelty tässä tutkielmassa. Tutkielman päätelmissä voi olla hieman epämääräisyyttä, sillä kirjallisuuslähteet tulkitsivat sopimusmalleja eri tavalla riippuen sopimuksen yksityiskohdista. Tapaustutkimuksista saatuihin kokemuksiin ei myöskään voida täysin luottaa. Sopimusmallit olivat olleet yritysten käytössä vasta vähän aikaa tai ne olivat yritysten itse kehittämiä sopimusmalleja, jolloin arviointi on voinut olla hieman subjektiivista. On myös syytä huomioda, että tutkielmassa tarkastellut tapaustutkimukset on tehty vuosina 2005–2009 ja niissä tutkimus on vielä alkutaipaleella. Jatkotutkimuksena voisikin tutkia tarkemmin jotakin ketterän lähestymistavan mukaiseen projektiin käytettäväksi esitettyä sopimusmallia ja sen soveltuvuutta erityyppisiin ohjelmistoprojekteihin.

LÄHTEET

- Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. (2002). Agile software development methods. Review and analysis [online]. Helsinki. VTT, [viitattu 17.12.2008]. VTT:n julkaisuja 478. Saatavilla pdf-muodossa: <URL: <http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>>
- Agile Alliance. (2001). Agile Manifesto. Agile Alliance. Saatavilla osoitteessa <<http://agilemanifesto.org/>>[viitattu 26.10.2009]
- Baskerville R., Ramesh B., Levine L., Pries-Heje J.& Slaughter S. (2003). Is internet speed software development different? IEEE Software, 20(6), 70-77.
- Beck, K. (1999). Embracing change with extreme programming. Computer, 32(10), 70-77.
- Cadle J., Yeates D. (2008). Project management for information systems. 5.painos, Prentice-Hall.
- Davis, A.M. (1994). Fifteen principles of software engineering. IEEE Software 11(6), 94 - 96.
- Eckfeldt, B., Madden, R. & Horowitz, J. (2005). Selling Agile: target-cost contracts. Teoksessa: Proceedings of the Agile Development Conference, New York, NY, USA, July 24-29, 160-166.
- Franklin, T. (2008). Adventures in Agile contracting: evolving from time and materials to fixed price, fixed scope contracts. Teoksessa: Proceedings of the Agile, Washington, DC, USA, Aug. 4-8, 269 - 273.
- Fukuyama, F. (1996). Trust. New York, NY, USA: The Free Press Paperbacks.

- Haikala, I. & Märijärvi, J. (2000). Ohjelmistotuotanto. 7. uudistettu painos. Helsinki, Suomi: Satku - Kauppakaari Oyj.
- Heikkala, M.-S. (2009). Riskienhallinta: Riskiraportoinnin ja riskienhallinnan kehittyminen - Case Kemira Oyj, Neste Oil Oyj, Aspoo Oyj ja Finnlines Oyj. Pro gradu-tutkielma, Markkinoinnin ja johtamisen laitos, Helsingin kauppakorkeakoulu.
- Heiskanen, A., Newman, M. & Eklin, M. (2008). Control, trust, power and the dynamics of information system outsourcing relationships: A process study of contractual software development. *Journal of strategic information system* 17(4), 268-286.
- Hemmo, M. (2003). Sopimusoikeus I. Helsinki, Suomi: Talentum.
- Hemmo, M. (2005). Sopimusoikeus III. Helsinki, Suomi: Talentum.
- Highsmith, J. (2002). Agile software development ecosystems. Agile Software Development Series, Boston, MA, USA: Pearson Education, Inc.
- IEEE Guide Adoption of PMI Standard. (2004). A Guide to the Project Management Body of Knowledge. The institute of Electrical and Electronics Engineers, Inc. May 2004.
- Kanninen, V. & Määttä, K. (1996). Näkökulmia oikeustaloustieteeseen. Tampere, Suomi: Gaudemus.
- Leffingwell, D. (2007). Scaling software agility – Best practices for large enterprises, Addison-Wesley.
- Leung H. & Fan Z. (2002). Software cost estimation, Handbook of software engineering, Hong Kong Polytechnic University.

- Möllering, G. (2005). The trust/control duality: an integrative perspective on positive expectations of others. *International Sociology* 20 (3), 283–305.
- Nelson C., Taran G. & Hinojosa L. (2008). Explicit risk management in agile processes. In Abrahamsson P., Baskerville R., Conboy K., Fitzgerald B., Morgan L., Wang X. (toim.) *Agile processes in software engineering and extreme programming*, LNBIP 9, Springer, 2008.
- Nysten-Haarala, S. (1998). The long-term contract. *Contract law and contracting*, in Faculty of Law, Rovaniemi, Finland: University of Lapland. 315.
- O'Sullivan, A. & Sheffrin, S.M. (2003). *Economics: Principles in action*. Upper Saddle River, New Jersey, USA: Pearson Prentice Hall.
- Pitette, S. (2001). Progressive Acquisition and the RUP: comparing and comparing iterative processes for acquisition and software development. *The Rational Edge*, November 2001.
- Poppendieck, M. & Poppendieck T. (2003). *Lean software development: an agile toolkit*. Boston, MA., USA: Addison-Wesley.
- Reel, J. (1999). Critical success factors in software projects, *IEEE Software* 16(3), 18-23.
- Sabherwal, R. (1999). The role of trust in outsourced IS development projects. *Communications of the ACM* , 42(2), 80-86.
- Schwaber, K & Beeble, M. (2002). *Agile software development with Scrum*. Redmont, Washington, USA: Microsoft Press.
- Schwaber, K (2004). *Agile project management with Scrum*. Redmont, Washington, USA: Microsoft Press.

- Sheta, A., Rine, D. & Ayesha, A. (2008). Development of software effort and schedule estimation models using Soft Computing Techniques. Evolutionary Computation, IEEE World Congress on Computational Intelligence. June 1-6 . 1283 - 1289.
- Sliger M., Broderick, S. (2008). The software project manager's bridge to agility, Addison-Wesley.
- Stevens, P. (2009a). Contracting for agile software project, part 1, saatavilla osoitesta
<<http://agilesoftwaredevelopment.com/blog/peterstev/contracting-agile-software-projects>> [viitattu 15.11.2009].
- Stevens, P. (2009b). 10 Contracts for your next agile software project, saatavilla osoitesta <<http://agilesoftwaredevelopment.com/blog/peterstev/10-agile-contracts>> [viitattu 15.11.2009].
- Thorup, L. & Jensen, B. (2009). Collaborative agile contract. Teoksessa: Proceedings of the Agile Development Conference, Chicago, USA, Aug. 24-28,195 - 200.
- Tiainen T., Luomala H., Kurki S. & Mäkelä Kari (Toim.). (2004). Luottamus sähköisissä palveluissa: kuluttajan ja palvelun tarjoajan vuorovaikutus. Tampereen yliopisto, Tietojenkäsittelytieteiden laitos. Raportti B-2004-11. Saatavilla osoitteesta: <<http://www.cs.uta.fi/reports/bsarja/B-2004-11.pdf>>. [viitattu 8.11.2009]
- Tietotekniikan liitto (2002). Tietojärjestelmän hankinta: Ohjelmistotoimittajan ja -ratkaisun valinta. TTL:n julkaisusarja, Helsinki, Suomi: Talentum Media Oy.

Turner, J.R. & Simister, S.J. (2001). Project contract management and a theory of organization. Erasmus Research Institute of Management.

Warsta, J. & Seppänen, V. (2000). Process view on the software contracting in the internet business. Euromicro Conference, 2000. 240 - 245.