

Walteri Vuorimies

KETTERÄN OHJELMISTOPROJEKTIN KANNATTAVUUDEN ARVIOINTI MYYN- TIVAIHEESSA

Diplomityö
11/2019

TIIVISTELMÄ

Walteri Vuorimies: Ketterän ohjelmistoprojektin kannattavuuden arviointi myyntivaiheessa
Diplomityö, 68 sivua, 1 liitesivua
Tampereen yliopisto
Tietojohdamisen diplomi-insinöörin tutkinto-ohjelma
Marraskuu 2019
Tarkastaja: Professori Samuli Pekkola

Avainsanat: Ketterä ohjelmistoprojekti, kannattavuus, hinnoittelu, kustannukset, riskit
myyntivaihe, arviointimenetelmät

Kannattavuuden arviointi on yhtenä edellytyksenä kestäväälle ja kannattavalle liiketoiminnalle. Ohjelmistoala on jo vuosia toiminut projektiluontoisesti ja uusia projektimenetelmiä kehitetään jatkuvasti. Vuonna 2001 julkaistun Agile Manifest -ohjeistuksen myötä ketterät ohjelmistoprojektit ovat saavuttaneet suuren suosion. Aikataulujen sekä kustannusten venyminen ovat tiedettävästi olleet jo vuosikymmeniä osana ohjelmistoprojekteja, mutta asiaan ei olla keksitty tarpeeksi hyviä ratkaisuja. Ketterät ohjelmistoprojektimenetelmät ovat auttaneet aikataulu- sekä kustannusarvioihin liittyviin ongelmiin ja aiheesta on tehty paljon tutkimuksia. Kannattavuuden arviointi jo projektien myyntivaiheessa on kuitenkin haastavaa samasta syystä kuin aikataulujen sekä kustannusten arviointi. Ohjelmistoprojekteissa tulee usein vastaan odottamattomia haasteita, jotka venyttävät projektien kustannuksia sekä aikatauluja. Ongelman haastavuus ja laajuus eivät kuitenkaan tarkoita sitä, että ongelmaa ei voisi ratkaista.

Tämän tutkimuksen tavoitteena on selvittää syitä sille, miksi ohjelmistoprojektien kannattavuuden arviointi on haastavaa ja miten ongelma voitaisiin ratkaista. Tutkimuksessa keskitytään kannattavuuden arviointiprosessiin ketterien ohjelmistoprojektien osalta ja tutkimuksen tavoitteena on luoda uusi toimivampi menetelmä kannattavuuden arviointiprosessiin jo projektin myyntivaiheessa. Tutkimuksessa selvitetään, miten ketterän ohjelmistoprojektin kannattavuus tulee arvioida myyntivaiheessa, jotta sisältöön, aikatauluun, henkilöihin ja kustannuksiin liittyvät riskit pystytään huomioimaan. Vastauksia ongelmaan pyritään löytämään jaottelemalla aihe kolmeen kokonaisuuteen: riskeihin, kustannuksiin ja hinnoitteluun sekä kannattavuuden arviointimenetelmiin.

Tutkimuksen tuloksena voidaan todeta, että kannattavuuden arviointi ketterissä ohjelmistoprojekteissa on haastavaa, mutta kannattavuuden arviointiprosessia voidaan kehittää vielä paljon. Tutkimuksen tuloksena saatiin muodostettua uusi kannattavuuden arviointiin tarkoitettu menetelmä, joka hyödyntää asiantuntijoiden osaamista sekä systemaattisia menetelmiä. Aikaisemmat tutkimukset ovat osoittaneet, että kannattavuuden arvioinnin osalta tarkimpana menetelmänä voidaan pitää asiantuntijoiden arvioita. Tämä on suuri syy sille, miksi asiantuntijoiden arviointikykyyn tulee luottaa myös jatkossa. Asiantuntijoiden tueksi tulee kuitenkin löytää parhaat systemaattiset menetelmät ja työkalut, jotka voivat tukea heidän arvioitansa. Tähän tarkoitukseen löydettiin useita menetelmiä pohjautuen kirjallisuuteen ja tämän tutkimuksen tuloksiin.

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

ABSTRACT

Waltteri Vuorimies: The estimation of profitability in agile software projects during the sales phase

Master's Thesis

Tampere University

Master's degree programme in Information and knowledge management

November 2019

Examiner: Professor Samuli Pekkola

Keywords: Agile software project, profitability, pricing, expenses, risks, sales phase, estimation methods

The evaluation of profitability of an organization is a crucial part of a long-lasting business. The software industry has been using various of project tools in their workload for years and these tools are being developed continuously for better results. After the launch of Agile Manifest - directory in 2001, a certain project tool gained a huge popularity amongst the industry – the agile software development. It is an assumption within the industry that most of the time planned schedule will be delayed and budget will be surpassed. While this is frustrating for many, better ways of evaluating have not been developed. Agile software development has been used to ensure that the estimation of both the schedule and the budget plan for projects are exact. A lot of research has been done about the topic, however, the evaluation of profitability during the sales phase is challenging as there are too many factors to consider. During the software projects countless of unexpected challenges tend to occur that messes with the initial plan. This specific key element that affects the daily tasks, but also the whole organization, is an aspect worth developing to ensure better profitability and more efficiency.

The goal of this research is to find reasons to why the profitability of software projects is hard to evaluate and how this problem could be solved. The main emphasis of the research is the evaluation of profitability in the sales phase of an agile software project. Thus, this research is done to achieve a better functioning evaluation of profitability already in the early phase of a project, more specifically already when the sales is being negotiated. The research finds out how the profitability of an agile software project could be better evaluated in the sales phase so that the risks related to scheduling, staff and other expenses are considered thoroughly. This matter is solved by categorizing the topic into three areas: risks, costs/pricing and the estimation models for profitability.

Looking at the results of this research, one can establish that evaluation in agile software projects is challenging, however, the estimation process could be improved. Using the results of the research, a new model was created to help the estimation process. The new process takes the level of expertise of the experts in the field and systematic methods into account. Previous research around the topic has proved that expert estimation has been one of the most precise evaluations in software projects. Due to this, one must continue to trust the capabilities of experts and their estimations in the future. It may however be beneficial to find new systematic ways that are thorough and science-based to help experts do their estimations. This research found various of methods by looking through different literature.

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

ALKUSANAT

Tämä tutkimus toteutettiin yhteistyössä Digia Finland Oy -yrityksen kanssa ja suurin kiitos tutkimuksen onnistumisesta kuuluu heille. Tutkimusta tehtiin noin seitsemän kuukauden ajan. Tutkimuksen aiheen valinta oli mielenkiintoista ja yhteistyöyrityksen kanssa keskusteltua aiheeksi valikoitui ketterän ohjelmistoprojektin kannattavuuden arviointi myyntivaiheessa.

Aihe oli henkilökohtaisesti kiinnostava, sillä se yhdistää useita elementtejä, joita olen opinnoissani opiskellut. Aihe sisältää tietotekniisiin projekteihin liittyviä elementtejä sekä talouteen ja johtamiseen liittyviä elementtejä. Tämän takia aihe sopi mielestäni hyvin minulle sekä tutkinto-ohjelmaani. Mielenkiintoisen aiheen vuoksi tutkimusta oli erittäin miellyttävää tehdä alusta loppuun saakka.

Haluaisin kiittää myös Tampereen yliopistoa sekä erityisesti ohjaajaani Samuli Pekkola, joka ohjasi minua työssä eteenpäin. Sain apua aina tarvittaessa niin ohjaajaltani kuin yhteistyöyrityksen henkilöiltäkin. Tutkimuksessa haastateltavat ihmiset olivat mielenkiinnolla mukana vastaamassa kysymyksiini sekä testaamassa muodostettua mallia. Iso kiitos myös heille.

Tampereella, 21.11.2019

Waltteri Vuorimies

SISÄLLYSLUETTELO

1. JOHDANTO	1
1.1 Työn tausta	1
1.2 Työn rakenne	3
1.3 Työn rajaukset ja tavoitteet	5
2. KETTERÄN OHJELMISTOPROJEKTIN KANNATTAVUUS	8
2.1 Ketterä ohjelmistoprojekti	8
2.1.1 Scrum-menetelmä	10
2.2 Kannattavuuden arviointi	13
2.2.1 Kustannusten muodostuminen	15
2.2.2 Hinnoittelu	16
2.2.3 Riskienhallinta	20
3. KANNATTAVUUDEN ARVIOINTIKEINOT	24
3.1 Työmäärän arviointi ohjelmistoprojekteissa	24
3.1.1 WBS	27
3.1.2 Program Evaluation and Review Technique	28
3.1.3 Function Point Analysis -menetelmä	29
3.1.4 Delphi-menetelmä	31
3.1.5 Constructive Cost Model	32
3.2 Kannattavuuden arvioinnin systemaattinen malli	34
4. TUTKIMUKSEN TOTEUTUS	39
4.1 Tutkimuksen toteutus	39
4.1.1 Yrityksen esittely	40
4.1.2 Tutkimusotanta	40
4.1.3 Haastattelut	42
4.1.4 Mallin testaaminen kokeellisesti	44
4.2 Tutkimusmetodologia	45
4.3 Sekundääriaineisto	47
5. EMPIIRISET TULOKSET	48
5.1 Haastattelun tulokset	48
5.2 Mallin testaamisen tulokset	51
6. KONSTRUKTION ARVIONTI	53
6.1 Mallin arviointi	53
6.2 Mallin jatkokehitys mahdollisuudet	54
7. YHTEENVETO	58
7.1 Yhteenveto ja tutkimuskysymyksiin vastaaminen	58
7.2 Tutkimuksen arviointi	60
7.3 Aiheen jatkojalostus	62
LÄHTEET	64
LIITE A: KONSTRUKTIO	69

KUALUETTELO

Kuva 1.	<i>Projektinhallinta kolmio (Mukaillen Wysocki 2011; Atkinson, 1999)</i>	2
Kuva 1.	<i>Tutkimuksen rakenne</i>	4
Kuva 2.	<i>Scrum-prosessi (Ashraf & Aftab, 2017)</i>	12
Kuva 3.	<i>First Copy Cost -efekti (Anding & Hess, 2004)</i>	16
Kuva 4.	<i>Kustannus- ja arvoperusteisen hinnoittelun erot IT-palvelun hinnoittelu prosessissa (mukaillen Harmon et al. 2009)</i>	19
Kuva 5.	<i>OCTAVE Allegro -menetelmä (Caralli, 2007 s.57)</i>	22
Kuva 6.	<i>Työmäärän arviointitekniikat (Boehm et al. 2000)</i>	25
Kuva 7.	<i>FPA-menetelmän yhteenlaskutaulukko (Longstreet, 2002)</i>	30
Kuva 8.	<i>Haastattelukysymykset</i>	43
Kuva 9.	<i>Tutkimusmetodologiset valinnat</i>	46

TAULUKKOLUETTELO

<i>Taulukko 1.</i>	<i>Haastateltavat henkilöt.....</i>	<i>41</i>
<i>Taulukko 2.</i>	<i>Mallin testaamisen tulokset.....</i>	<i>51</i>

1. JOHDANTO

1.1 Työn tausta

Ohjelmistoalan johtajilla on monesti edessään kysymys siitä, tulisiko projekti jatkaa loppuun asti vai ei, kun projekti etenee ei toivottuun suuntaan. Tässä tilanteessa projekti-johtajat miettivät, onko projektia enää kannattavaa jatkaa taloudellisesta näkökulmasta katsoen. Tähän kannattavuuden arviointiin liittyvän kysymyksen ratkaisuun tulisi olla käytössä systemaattisia menetelmiä, mutta niitä ei ole. (Wagner et al. 2007) Ohjelmistoala on ollut olemassa jo vuosikymmeniä, ja haasteet projektien kannattavuuden arvioinnista ovat seuranneet mukana koko tämän ajan.

Ohjelmistoprojektien arvioinnista on tehty tuhansia tutkimuksia sekä raportteja ja niitä on julkaistu ohjelmistoalan oppikirjoissakin. Huolimatta siitä, että aihe on saanut paljon huomiota osakseen, pitkäikäisiä arviointimenetelmiä ei ole syntynyt ja ohjelmistoprojektien arviointi on ollut pitkälti asiantuntijoiden arviointia. (Keung, 2007) Ohjelmistoprojektien haasteina ovat esimerkiksi hallitsemattomat kustannukset, myöhästyneet aikataulut ja ennustamaton laatu. Ohjelmistoalan yritysten olisi toimitettava laadukkaita tuotteita, ajallaan ja budjetissa pysyen. (Slaughter et al. 1998)

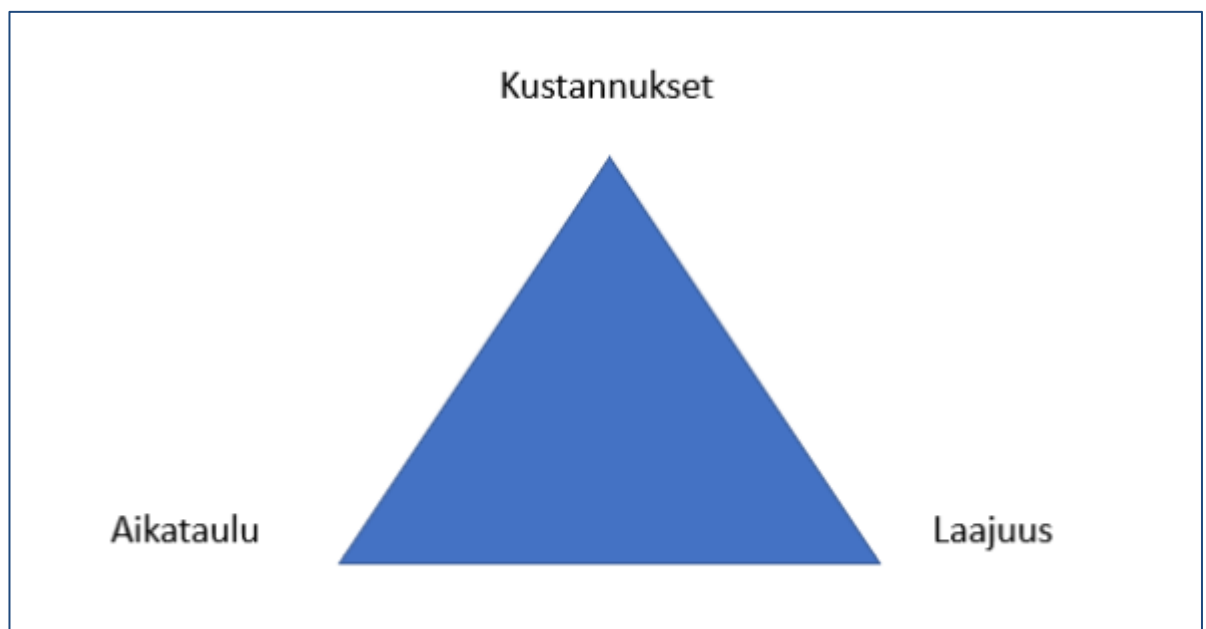
Aihetta on syytä tutkia, sillä ongelmat ovat olleet alalla pitkään ja ratkaisua ei olla keksitty. Jørgensenin (2007) mukaan jo 1980-luvulla on tullut esille ongelmia ohjelmistoalan ja sen projektien arvioinnissa. Näitä arviointiin liittyviä ongelmia ei olla vielääkään pystytty ratkaisemaan ja osa arviointiin liittyvistä ongelmista voivat olla mahdottomia ratkaista (Jørgensen, 2007). Ohjelmistoalalla työtä tehdään usein projekteissa ja projektien johtaminen on haastavaa, niin ohjelmistoalalla kuin perinteisemmilläkin aloilla. Ohjelmistoalalla kustannusten hallinta sekä johtaminen ovat perinteisempiä aloja haastavampaa. Budjetit voivat venyä moninkertaiseksi ja aikataulut voivat venyä jopa kaksinkertaisiksi. (Putnam 1978)

Ohjelmistoalan ongelmakohdat ovat tunnistettu jo kauan aikaa sitten ja ongelmaan on myös pyritty keksimään ratkaisuja. Yleisesti käytössä olevia systemaattisia kannattavuuden arviointi menetelmiä ei ole, mutta yhtenä projektien hallintaa parantavana tekijänä ohjelmistoalalle on otettu käyttöön ketterä ohjelmistokehitys -menetelmä. Ketterä ohjelmistokehitys yleistyi vuonna 2001, kun ryhmä ohjelmistokehittäjiä halusi kehittää edistyneempiä tapoja ohjelmistojenkehitykseen ja tämän seurauksena syntyi The Manifesto

for Agile Software Development eli ketterän ohjelmistokehityksen julistus (Chow & Cao, 2008).

Ketterät ohjelmistoprojekti -menetelmät ovat kehittyneet asiakasvaatimusten parempaan tunnistamiseen sekä asiakkaan tyytyväisyyden parantamiseen. Ketterän ohjelmistoprojektin avulla projektiin saadaan joustavuutta, joka parantaa asiakkaan tyytyväisyyttä. (Kotaiah & Khalil, 2017) Toimittajan näkökulmasta tämä poistaa painetta esimerkiksi ohjelmiston toiminnallisuuden toteuttamisesta tietyssä aikataulussa ja sen tulisi myös helpottaa toimittajan näkökulmasta kannattavuuden arviointia jo projektin myyntivaiheessa.

Ketterä ohjelmistoprojekti helpottaa ohjelmistoalan myös hinnoitteluongelmaa, sillä kiinteähintainen projekti ei pitäisi olla ketterässä ohjelmistoprojektissa sopimusteknisesti mahdollinen (Jaakonaho 2009). Tämä on yksi ketterän ohjelmistoprojektin etu ottaen huomioon ohjelmistoprojektien aikataulujen ja budjettien venymisen. Yleisesti ottaen jokaiseen projektiin kuuluu alasta riippumatta kolme keskeistä osa-alueetta: aika, kustannukset ja laajuus (Wysocki 2011). Kolme projektinhallinta kolmion (eng. Project Management Triangle) muodostavana osa-alueena voidaan nähdä myös aika, kustannukset sekä laatu, josta yhdenkin osa-alueen venyessä toisen osa-alueen on myös venyttävä (Atkinson, 1999). Esimerkiksi projektin aikataulun venyessä, joko kustannukset kasvavat tai projektin laajuutta on pienennettävä. Kuvassa 1. esitelty mukailtu projektinhallinta kolmio.



Kuva 1. Projektinhallinta kolmio (Mukaillen Wysocki 2011; Atkinson, 1999)

Ohjelmistoprojektien haasteisiin liittyvää metodologiaa on yritetty kehittää paljon, mutta haasteita esimerkiksi viivästymisien sekä epäonnistumisien suhteen ei olla pystytty ratkaisemaan (Chow & Cao, 2008). Kannattavuuden arvioinnissa täytyy ottaa huomioon useita osa-alueita. Tässä työssä ongelmaa lähestytään ketterän ohjelmistokehityksen näkökulmasta ja keskeisimpiä kannattavuuteen arviointiin vaikuttavia tekijöitä ovat kustannusten muodostuminen, riskienhallinta sekä hinnoittelu. Riskienhallinta ohjelmistoprojekteissa sisältää paljon asiaa ja riskien suuruutta on haastavaa mitata. Yleisesti ottaen riskin suuruus voidaan laskea kertomalla tapahtuman todennäköisyys toteutuneen tapahtuman menetyksen arvolla (Benaroch et al. 2010).

Ohjelmisto projektien osalta kustannusten muodostuminen ei ole yksinkertainen laskutoimitus vaan myös se sisältää haasteita. Kustannukset voivat venyä monista syistä ja riskien toteutuessa projekteille voi syntyä suuria summia odottamattomista kustannuksista. Kustannusten arvioiminen systemaattisesti vaatii laadukasta työkalua, jota ohjelmistotalalla ei ole yleisesti tiedossa (Wagner et al. 2007).

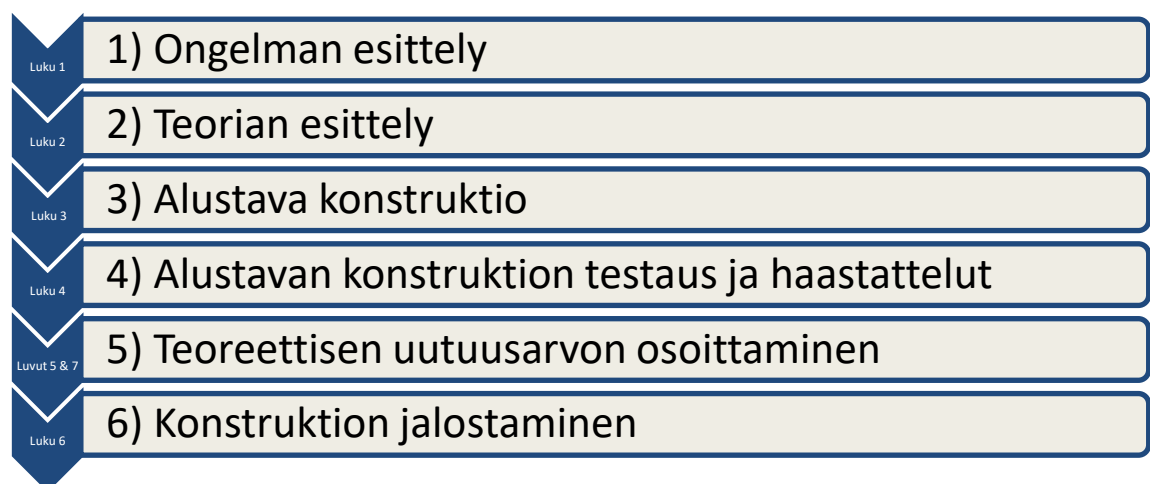
Tämä tutkimus toteutetaan yhteistyössä Digia Finland Oy- yrityksen kanssa. Kohdeyrityksessä on hyödynnetty jo vuosia ketterää ohjelmistokehittämistä, mutta yhtenä haasteena prosessien osalta voidaan vielä pitää kannattavuuden arvioinnin menetelmien puutetta myyntivaiheessa. Tutkimuksen on tarkoituksena ratkaista kohdeyrityksen ketteriin ohjelmistoprojekteihin liittyvä haaste ja saada luotua konstruktio, jonka avulla ketterän ohjelmistoprojektin kannattavuuden arviointiin myyntivaiheessa saadaan systemaattinen työkalu. Haasteet ketterien ohjelmistoprojektien kannattavuuden arvioinnissa johtuvat monista seikoista, mutta esimerkiksi kohdeyrityksen tapauksessa suuren haasteen luo uudenlaisien ohjelmistosisältöjen arvioinnin vaikeus. Tämä on haasteena projektien kannattavuuden arvioinnille, koska on vaikeaa arvioida kuinka paljon aikaa ja vaivaa uudenlaisten ominaisuuksien kehittäminen vie sekä mitä haasteita tulee vastaan, kun luodaan uutta.

1.2 Työn rakenne

Työn tavoitteena on muodostaa konstruktio ketterien ohjelmistoprojektien kannattavuuden arviointiin myyntivaiheessa ja työ on muodostettu konstruktivisella tutkimusperiaatteella. Konstruktivisessa tutkimuksessa neljä keskeistä osa-aluetta, joista lopputuloksena muodostuu konstruktio ovat teoreettinen uutuusarvo, teoriaan kytkentä, käytännön toimivuus sekä käytännön relevanssi (Oyegoke, 2011).

Teoreettinen uutuusarvo on mahdollista havaita vertaamalla tutkimusta aikaisempaan kirjallisuuteen. Konstruktivisen tutkimuksen yhtenä periaatteena on luoda uutta tutkimustietoa vanhan tiedon päälle (Kasanen, 1993). Teoriaan kytkennän tarkoituksena on luoda yhteys konstruktion ja teorian välille. Tässä tutkimuksessa teoria ja konstruktiot ovat kytkettyinä, sillä tutkimuksen lopputuloksena saatu malli on muodostettu teorian pohjalta empirian avulla jalostaen. Tutkimuksessa käytännön relevanssi on ilmeinen kohdeyrityksen tarpeisiin nähden. Kannattavuuden arviointimenetelmälle on suurta kysyntää ja konstruktion avulla tähän kysyntään voidaan vastata systemaattisesti. Käytännön toimivuuden osalta tutkimuksessa muodostettua konstruktiota testataan kohdeyrityksen dataa hyödyntäen sekä haastatteleamalla ketterien ohjelmistoprojektien kannattavuuden arviointia tekeviä ammattilaisia.

Tämän tutkimuksen rakenne on muodostettu konstruktivisella tutkimustavalla, joka muodostuu kuudesta vaiheesta. Konstruktivinen tutkimus etenee käytännön ongelmasta ratkaisun viimeistelyyn. Tässä tutkimuksessa lähtöongelmana on systemaattisuuden puute ketterien ohjelmistoprojektien kannattavuuden arvioinnissa ja työn tavoitteena on löytää tähän ongelmaan ratkaisu, teorioita ja empiriaa hyödyntäen. Konstruktivisen tutkimuksen kuusi keskeistä vaihetta ovat: 1) käytännön ongelman etsiminen, 2) yleisymmärryksen muodostaminen teorian pohjalta, 3) alustavan konstruktion muodostaminen, 4) alustavan konstruktion testaus, 5) teoreettisen uutuusarvon osoittaminen sekä 6) konstruktion jalostaminen ja sen käytännöllisyyden arvon osoittaminen (Oeygoke, 2011; Kasanen, 1993). Kuvassa 3. esiteltynä tutkimuksen rakenne.



Kuva 1. Tutkimuksen rakenne.

Kuvassa 3. esitellyssä rakenteessa konstruktivisen tutkimuksen rakenne esiteltynä numeroituna. Konstruktivisen tutkimuksen vaiheet kuvattu numeroina, joiden vieressä esitelty luku kertoo, missä tutkimuksen luvussa kyseinen aihe esiintyy.

1.3 Työn rajaukset ja tavoitteet

Tämän tutkimuksen tavoitteena on selvittää keinot, miten kannattavuuden arviointia voidaan parantaa sekä luoda systemaattinen malli kohdeyrityksen tarpeisiin arvioida kannattavuutta tarkemmin jo ketterän ohjelmistoprojektien myyntivaiheessa. Tavoitteisiin pääsyyn tarvitaan teoreettista tietoa ketteristä ohjelmistoprojekteista, kannattavuuden arvioinnista, riskienhallinnasta, kustannusten muodostumisesta sekä hinnoittelusta. Tutkimuksen myötä muodostuneen konstruktion on tarkoitus parantaa kohdeyrityksen ketterien ohjelmistoprojektien kannattavuuden arviointia myyntivaiheessa ja samalla konstruktion on tarkoituksena yhtenäistää sekä systematisoida menetelmiä, joita kannattavuuden arvioinnissa käytetään.

Kannattavuuden arviointi ohjelmistoprojekteissa on ollut jo pitkään haasteena alalla ja sen merkitys niin kohdeyritykselle kuin koko alalle on suuri. Ohjelmistoalan räjähdysmäisen kasvun myötä haasteet ja ongelmakohdat projektien osalta ovat tulleet peremminkin esiin (Slaughter et al. 1998). Näitä haasteita ovat esimerkiksi hallitsemattomat kustannukset, aikataulujen venymiset sekä vaikeasti arvioitavissa oleva lopullisen tuotteen laatu (Slaughter et al. 1998). Nämä kaikki haasteet liittyvät ohjelmistoprojektin lopulliseen kannattavuuteen, jonka takia aihe on ajankohtainen.

Tutkimuksen teoriaosuuden tarkoituksena on avata yleisiä haasteita ketterän ohjelmistoprojektin kannattavuuden arviointiin liittyen sekä löytää näihin haasteisiin ratkaisuja. Aiheen haastavuuden takia on tärkeää, että teoriaosuudessa käydään läpi monia mahdollisia menetelmiä, joiden avulla ongelmaa on mahdollista ratkaista. Teorian pohjalta on tarkoituksena luoda alustava konstruktio/malli kannattavuuden arviointiin ketterissä ohjelmistoprojekteissa ja tätä mallia testataan tutkimuksen myöhemmässä vaiheessa.

Teoriaosuuden ja mallin luomisen jälkeen on tärkeää testata, kuinka hyvin malli toimii ja miten sitä voisi jatkokehittää. Mallia testataan kohdeyrityksen vanhojen projektien datalla ja tämän avulla voidaan nähdä miten hyvin malli olisi toiminut historiassa. On tärkeää, että malli toimii historiassa tapahtuneissa projekteissa, mutta tämänkaltaisen testaus osoittaa vain sen, miten hyvin malli olisi toiminut eikä sitä, miten hyvin se tulee toimimaan tulevaisuudessa projekteissa. Datan avulla tapahtuneen testauksen lisäksi mallia pyritään kehittämään haastattelujen avulla. Haastattelujen tarkoituksena on saada kvalitatiivista dataa tietoa siitä, mitä mieltä ketterien ohjelmistoprojektien kannattavuuden arviointia tekevät ovat mallista sekä siitä, mihin suuntaan mallia heidän mielestään tulisi kehittää.

Lopputuloksena teorian pohjalta luodun mallin ja sen testauksen sekä haastattelujen avulla on tarkoituksena luoda lopullinen konstruktio, joka vastaa kohdeyrityksen tarpeisiin ja tutkimuskysymyksiin. Tutkimuskysymyksiä tässä työssä on yhteensä kolme, yksi päätutkimuskysymys sekä kaksi alatutkimuskysymystä. Lopputulokseen päätymistä varten valittiin seuraavanlainen päätutkimuskysymys.

1. Miten ketterän ohjelmistoprojektin kannattavuus tulee arvioida myyntivaiheessa, jotta sisältöön, aikatauluun, henkilöihin ja kustannuksiin liittyvät riskit pystytään huomioimaan?

Päätutkimuskysymys on avattu pienempiin osiin apututkimuskysymysten avulla, joihin vastaamalla voidaan vastata päätutkimuskysymykseen. Apututkimuskysymykset ovat seuraavanlaiset.

1.1 Mitkä tekijät aiheuttavat riskejä ja miten riskit tulee ottaa huomioon ketterän ohjelmistoprojektin kannattavuuden arvioinnissa?

1.2 Mitkä tekijät vaikuttavat ketterän ohjelmistoprojektin kustannuksiin ja hinnoitteluun?

1.3 Millä keinoilla kannattavuudenarviointi prosessia voidaan parantaa?

Tutkimuskysymyksiin vastataan teorian avulla ja tätä teoriasta saatua tietoa täydennetään empirian avulla. Haastattelujen avulla saadaan tietoa siitä, mitä teorian avulla ei osattu ottaa huomioon mallia luodessa ja mallin testauksen avulla on myös mahdollista löytää ongelmakohtia, jotka vastaavat tutkimuskysymyksiin.

Pää- ja alatutkimukset pitävät sisällään olettamuksen siitä, että ketterän ohjelmistoprojektien kannattavuutta on mahdollista arvioida. Tutkimuksen kannalta on oleellista huomata, että alkuolettamuksena on myös se, että kannattavuuden arviointiin on mahdollista luoda malli, jota hyödynnetään systemaattisesti jo projektien myyntivaiheessa. Tutkimuksen lopputulos osoittaa nämä alkuolettamukset joko todeksi tai vääräksi.

Tutkimuksen laajuuteen liittyen sekä kohdeyrityksen kiinnostuksen kohteisiin liittyen työ sisältää rajoituksia. Tutkimuksessa puhuttava projekti on rajattu kokoluokan mukaan. Tutkimuksessa keskitytään projekteihin, jotka ovat laajuudeltaan 150-1000 henkilötyöpäivää, projektiin kuuluu toimittajan osalta 4-7 henkilön projektiryhmä ja aikataulullisesti projekti kestää minimissään kolme kuukautta ja maksimissaan 13 kuukautta.

Ketterien ohjelmistoprojektien osalta tutkimusta ei ole syytä rajata tiukasti, sillä valitusta ketterästä ohjelmistoprojekti -menetelmästä huolimatta kannattavuuden arvioinnissa pätee samat lainalaisuudet. Teoria osuudessa kerrotaan kuitenkin tarkemmin, minkälainen ketterä ohjelmistoprojekti -menetelmä kohdeyrityksellä on käytössään. Tutkittavana or-

ganisaationa on kohdeyritys Digia Finland Oy ja mallin testaus suoritetaan kohdeyrityksen datalla. Teknologioiden osalta konstruktiossa hyödynnetään Microsoft Excel -laskentaohjelmaa sekä Microsoft Power BI -analysointiohjelmaa.

2. KETTERÄN OHJELMISTOPROJEKTIN KANNATTAVUUS

2.1 Ketterä ohjelmistoprojekti

Ketterät ohjelmistokehitysmenetelmät ovat yleistyneet vuonna 2001 julkaistun The Agile Manifesto:n (Ketterän ohjelmistokehityksen julistus) jälkeen ja ovat nykyään yleisesti käytössä ohjelmistoalalla (Agile Alliance, 2001). Ohjelmistoala on tunnettu projektien viivästymisistä ja epäonnistumisista, joten uusien menetelmien löytäminen tähän ongelmaan oli jo vuosituhannen alussa tarpeellista. Ketterän ohjelmistokehityksen julistuksessa menetelmän arvoiksi kuvataan seuraavat asiat:

- ”- *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 Alliance, 2001)

Julistuksessa on lueteltu edellä mainittujen neljän arvon lisäksi myös kaksitoista ohjetta, jotka toimivat periaatteina ketterissä ohjelmistoprojekteissa:

1. ”Tärkein tavoitteemme on tyydyttää asiakas toimittamalla tämän tarpeet täyttäviä versioita ohjelmistosta aikaisessa vaiheessa ja säännöllisesti.
2. Otamme vastaan muuttuvat vaatimukset myös kehityksen myöhäisessä vaiheessa. Ketterät menetelmät hyödyntävät muutosta asiakkaan kilpailukyvyn edistämiseksi.
3. Toimitamme versioita toimivasta ohjelmistosta säännöllisesti, parin viikon tai kuukauden välein, ja suosimme lyhyempää aikaväliä.
4. Liiketoiminnan edustajien ja ohjelmistokehittäjien tulee työskennellä yhdessä päivittäin koko projektin ajan.
5. Rakennamme projektit motivoituneiden yksilöiden ympärille. Annamme heille puitteet ja tuen, jonka he tarvitsevat ja luotamme siihen, että he saavat työn tehtyä.
6. Tehokkain ja toimivin tapa tiedon välittämiseksi kehitystiimille ja tiimin jäsenten kesken on kasvokkain käytävä keskustelu.

7. Toimiva ohjelmisto on edistymisen ensisijainen mittari.
8. Ketterät menetelmät kannustavat kestävään toimintatapaan. Hankkeen omistajien, kehittäjien ja ohjelmiston käyttäjien tulisi pystyä ylläpitämään työtahtinsa hamaan tulevaisuuteen.
9. Teknisen laadun ja ohjelmiston hyvän rakenteen jatkuva huomiointi edesauttaa ketteryyttä.
10. Yksinkertaisuus - tekemättä jätettävän työn maksimointi - on oleellista.
11. Parhaat arkkitehtuurit, vaatimukset ja suunnitelmat syntyvät itseorganisoituvissa tiimeissä.
12. Tiimi tarkastelee säännöllisesti, kuinka parantaa tehokkuuttaan, ja mukauttaa toimintaansa sen mukaisesti.” (Agile Alliance, 2001)

Ketterä ohjelmistoprojekti on ohjelmistoalalla paljon käytetty projektimenetelmä, joka pyrkii vastaamaan paremmin asiakkaiden tarpeisiin. Ketterät ohjelmistoprojektit tuovat perinteisiin projekteihin nähden enemmän joustavuutta ja tämän avulla on mahdollista parantaa asiakkaan tyytyväisyyttä. (Kotaiyah & Khalil, 2017) Yksi tärkeimmistä ketterän projektin ominaisuuksista on se, että asiakas tietää jo projektivaiheessa tarkemmin, millälaisen tuotteen he ovat saamassa ja toimittajan on helpompaa sekä edullisempaa muokata toimitettavaa ohjelmistoprojektia tarpeen vaatiessa jo projektin aikana.

Tämä on mahdollista ketterissä ohjelmistoprojekteissa, sillä asiakas ja toimittaja ovat koko projektin ajan kanssakäymisessä keskenään, ja ohjelmistoa tuotetaan useiden sprinttien aikana, jolloin ohjelmistosta tehdään valmiiksi tietty, pieni, osa kerrallaan (Kotaiyah & Khalil, 2017). Ohjelmisto tuotetaan niin, että projektissa lähdetään liikkeelle tärkeimmistä ominaisuuksista, jotka tuotetaan yksi osa kerrallaan asiakkaan tyytyväisyysmielessä pitäen. Asiakkaalla on ketterässä ohjelmistoprojektissa mahdollista vaikuttaa tuotteeseen osien valmistuessa ja tämän avulla varmistetaan, ettei lopullinen tuote ole jotain muuta, mitä asiakas on ajatellut.

Ohjelmistokehityksen yhtenä trendinä on ollut käyttäjäkokemukseen panostaminen ja ketterän ohjelmistomenetelmän voidaan nähdä vastaavan juuri tähän trendiin. Ketterät ohjelmistoprojektit ovat ennen kaikkea luotu parempaa asiakaskokemusta varten, mutta sitä voidaan pitää myös kehittäjälähtöisenä menetelmänä. Menetelmän tarkoituksena on tuottaa siis lisäarvoa niin kehittäjille kuin asiakkaillekin. (Law & Lárusdóttir, 2015)

Ketteriä ohjelmistomenetelmiä on useita. Menetelmät ovat pääosin hyvin samantyyliisiä, mutta niissä on hieman eroavaisuuksia. Yleisimpiä ohjelmistoalalla käytössä olevia

metodeja ovat Scrum, Extreme Programming (XP), Dynamic System Development Method (DSDM), Adaptive Software Development (ASD), Lean Software Development (LD), Crystal tai Feature-Driven Development (FDD). Ketteristä ohjelmistomenetelmistä yleisin on Scrum. (Chow & Cao, 2008)

Kohdeyrityksen käytössä on projektimenetelmä, joka on Scrum-pohjainen, mutta siihen on lisätty Microsoftin Surestep -menetelmästä osia, jotka soveltuvat hyvin Microsoftin alustojen kanssa toimiville organisaatioille. AES toimii yhteistyössä Microsoftin kanssa ja ohjelmistoja ei projekteissa rakenneta tyhjästä, vaan kohdeyrityksen ketterän ohjelmistoprojektit rakentuvat Microsoft Dynamics 365 -ohjelmiston päälle ja tämän takia Scrum-menetelmää on muokattu soveltumaan paremmin kohdeyrityksen tarpeisiin.

2.1.1 Scrum-menetelmä

Scrum-menetelmä on yksi suosituimmista ketterän ohjelmistokehittämisen menetelmistä. Scrum-projektissa projektitiimi muodostuu 1-9 henkilön tiimistä, joka on jakautunut kolmeen rooliin. Scrum-projektissa rooleina ovat Scrum Master, Tuoteomistaja (eng. Product Owner) sekä tiimin jäsen. (Abdulhalim et al. 2018) Yleisesti ottaen Scrum-projektit ovat kestoaltaan noin kolme kuukautta, ja siihen sisältyy noin kolme kappaletta sprinttejä, joissa ohjelmistoa toteutetaan osissa (Leffingwell, 2011, s. 15). Scrum-projektin voidaan katsoa sisältävän kolme keskeistä vaihetta: käynnistys, kehitys sekä lopetus (Schwaber, 1997).

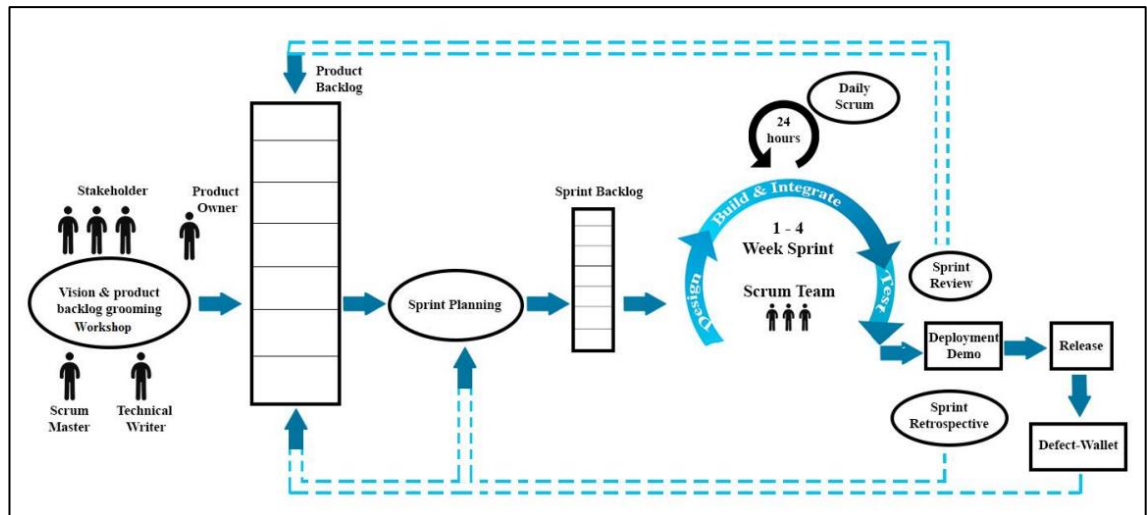
On tärkeää kuitenkin huomata, että ketterät ohjelmistoprojektit voivat kestää pidempään ja projekti ryhmän koko voi vaihdella. Esimerkiksi tämän tutkimuksen osalta tarkastelun kohteena ovat projektit, jotka ovat kestoaltaan 3-13 kuukautta pitkiä ja projekti ryhmään kuuluu 4-7 henkilöä. Tämä raja on valittu, sillä kohdeyrityksen ketterät projektit ovat luonteeltaan edellä mainitun kaltaisia ja kohdeyrityksen projektimenetelmä ei ole puhtaasti Scrum-menetelmä. Kohdeyrityksen ketterä ohjelmistomenetelmä on luotu vastamaan Microsoft-alustan tuomiin rajoituksiin ja suurimpana erona Scrum-menetelmään nähden, kohdeyrityksen omissa ketterissä ohjelmistoprojekteissa analyysivaihetta on muokattu sopimaan omiin tarpeisiin.

Projektien hyvä suunnittelu on edellytyksenä onnistuneelle projektille. Scrum-projektien käynnistysvaiheen keskeisimpinä tehtävinä on tehdä suunnitelma koko projektille sekä tuotteen implementoinnille. Ketterien ohjelmistoprojektien yhtenä tunnuspiirteenä voidaan pitää tuotteen kehitysjonoa (eng. Product Backlog), joka luodaan käynnistysvaiheessa. Kehitysjonolla tarkoitetaan listaa tuotteen ominaisuuksista, jotka lopulliselle tuot-

teelle halutaan sekä näiden ominaisuuksien priorisoimista järjestykseen, jossa niitä projektin edetessä toteutetaan. Käynnistysvaiheessa projektin suunnitellaan, miten halutut ominaisuudet toteutetaan sekä aikataulutetaan, milloin ja mitä ominaisuuksia sprinteissä tehdään. Käynnistysvaiheen aikana on myös tarkoituksena koota toimiva tiimi projektille sekä arvioida riskejä, joita toteutuksen aikana voi syntyä (Schwaber, 1997) Kannattavuuden arvioinnin osalta on tärkeää osata ottaa huomioon jo käynnistysvaiheessa, mitä riskejä projekti sisältää.

Käynnistysvaiheen jälkeen ketterissä ohjelmistoprojekteissa vuorossa on kehitysvaihe. Tämä kehitysvaihe sisältää useita sprinttejä, jotka ovat yleensä kestoaltaan noin 1-6 viikkoa. Sprintit ovat usein pilkottu pienempiin osiin, jopa päiväkohtaisiin sprintteihin. Kehitysvaiheen tarkoituksena on toteuttaa tuotteeseen halutut ominaisuudet sprintti kerrallaan kehitysjonon ominaisuuksia toteuttaen. Sprinttien aikana projektitiimi voidaan jakaa useisiin kehitystiimeihin, jotka kukin toteuttaa omia osuuksiaan lopullisesta toteutuksesta. Sprinttien alussa projektitiimi jaetaan kehitystiimeihin, suunnitellaan mitä ominaisuuksia kyseisessä sprintissä halutaan toteuttaa ja mikä kehitystiimi toteuttaa mitä. Kehitysvaiheen alun suunnittelukokouksen jälkeen kehitystiimit toteuttavat osansa, jonka jälkeen osat kootaan yhteen, arvioidaan sekä mukautetaan yhteneväisiksi. Sprintit suoritetaan kukin suunnitellussa aikataulussa, jonka jälkeen siirrytään seuraavaan sprinttiin ja kehitysjonon seuraavien toiminnallisuuksien toteutukseen. (Schwaber, 1997)

Ketterän ohjelmistoprojektin viimeisessä, lopetusvaiheessa on tarkoituksena ottaa toteutettu ohjelmisto käyttöön. Tähän vaiheeseen siirrytään, kun sprintit ovat suoritettu ja projektin johto on tyytyväinen toteutettujen toiminnallisuuksien määrään sekä laatuun. Testaus on tässä vaiheessa toteutettu ja ohjelmisto ottaa käyttöön tai joissain tapauksissa ohjelmiston osa voidaan siirtää tuotantoon. Lopetusvaiheen yhtenä tärkeänä kohtana on myös integraatioiden toimivuuden varmistuminen. (Schwaber, 1997) Kuvassa 4. on esiteltynä Scrum-ohjelmistoprojekti kokonaisuus lineaarisesti kuvattuna vasemmalta oikealle.



Kuva 2. Scrum-prosessi (Ashraf & Aftab, 2017)

Kuten kuvasta 4. voidaan nähdä, tuoteomistaja sekä Scrum Master ovat molemmat projektin alusta alkaen mukana. Myös tiimin jäsenet ovat mahdollisesti mukana alusta alkaen, mutta heidän roolinsa ainoastaan vaan suorittaa heille annetut tehtävät (Kotaiyah & Khalil, 2017). Käynnistysvaiheessa tärkeimmässä roolissa on tuoteomistaja. Tuoteomistajan tehtävänä on määritellä millainen tuotteesta loppujen lopuksi tulee sekä mitä projektin kehitysjono sisältää (Ashraf & Aftab, 2017). Tuotteen lopputulokseen sekä kehitysjonoon vaikuttavat myös muut projektissa toimivat, mutta vastuu suunnitelmien ja kehitysjonon synnystä on tuoteomistajalla.

Tuotteen suunnittelussa sekä projektin edetessä kommunikaatio asiakkaan ja toimittajan välillä tulee olla sujuvaa ja tuote omistajan yhtenä tehtävänä on varmistaa tämän toimivuus (Kotaih & Khalil, 2017). Lopullisen tuotteen on tärkeää vastata niin asiakkaan, kuin sen käyttäjienkin tarpeita ja toiveita. Tuoteomistajan yhtenä roolina on myös pitää huolta siitä, että tuote vastaa asiakasyrityksen liiketoimintatarpeisiin, mutta samaan aikaan hänen tulisi pitää mielessä loppu käyttäjät ja saada tuotteeseen heille keskeiset tekniset ominaisuudet (Kotaih & Khalil, 2017).

Tuoteomistajan rooli on ennen kaikkea olla asiakkaan sekä käyttäjien edustaja, Scrum Master pitää puolestaan huolen projektin etenemisestä. Scrum Masterin keskeisimpiä rooleja ovat projektin johtaminen sekä tuotteista, tapaamisista ja projektikäytännöistä vastaaminen. Kehitystiimin johtamisen lisäksi Scrum Master vastaa projektitiimin perehdytyksestä Scrumin toimintaperiaatteisiin sekä pitää huolen siitä, että tiimin jäsenet tietävät roolinsa ja tehtävänsä. (Ashraf & Aftab, 2017) Vaikka Scrum-projektit ovat luonteeltaan ketteriä, on kuitenkin pidettävä huolta siitä, että kaikki projektiin kuuluvat tietävät vastuunsa, jotta projekti tulee onnistumaan.

2.2 Kannattavuuden arviointi

Moni organisaatio määrittelee projektit riskien, liiketoiminta-arvon, laajuuden, kompleksisuuden sekä käytettyjen teknologioiden perusteella (Wysocki, 2011). Kannattavuuden arviointia tehdessä on siis tärkeä käydä läpi kaikkia osa-alueita. Arviointi määritellään epävarman tapahtuman informoiduksi arvioiksi (Nelson & Morris, 2014) Ohjelmiston kehityksessä on ollut käytössä jo yli 50 vuotta arviointimenetelmiä, mitkä ovat muodostettu tutkimusten ja raporttien pohjalta. Tästä huolimatta asiantuntija-arviointien hyödyntäminen ohjelmistoprojektien arvioinnissa ovat olleet alalla selvä valinta. (Keung, 2007) Tutkimukset ovat osoittaneet, että ohjelmistojen asiantuntijoiden arviointeihin vaikuttaa vahvasti ankkurointi, ylioptimistisuus sekä toiveikas ajattelu (Aranda and Easterbrook 2005; Jørgensen and Grimstad 2005; Jørgensen and Grimstad 2008).

Tutkimuksen mukaan 61 prosenttia ohjelmistoprojekteista voidaan katsoa epäonnistuneen jollain tapaa. Sama tutkimus osoittaa myös, että projektin arviointivaiheen käytetyin arviointimenetelmä on asiantuntijapohjainen samankaltaisiin projekteihin vertailu pohjautuen henkilön omaan muistiin. Kaksi vähiten hyödynnettyä arviointimenetelmää ovat matemaattisten kaavojen sekä arviointityökalujen hyödyntäminen, joiden edellä olivat muun muassa intuitio sekä arvailu. (Nelson & Morris, 2014) IT-projektien epäonnistumisprosentteja havainnoidessa sekä arviointimenetelmien käyttöasteita voidaan todeta, että nykyinen asiantuntijoiden arviointi kykyyn perustuva projektien arviointi ei ole toimiva keino ja asiaan tulisi keksiä parempia ratkaisuja.

On kuitenkin huomattava, että ketterät menetelmät onnistuvat perinteistä vesiputous-projektimallia paremmin pitämään arviointinsa ja 2010-luvun puolella on tapahtunut hieinan kehitystä projektien onnistumisprosentteissa. Vesiputous-projektit pysyvät 20% kustannusten osalta budjetissaan, ketterät-projekti puolestaan pysyvät budjetissaan 33% tarkkuudella. (Nelson & Morris, 2014) 2000-luvun vaihteessa yleistynyt ketterä ohjelmistomenetelmä voidaan katsoa olevan selvä parannus vesiputousmalliin, mutta parannettavaa jää silti. Ketterien ohjelmistoprojektien osalta menetelmät, joita käytetään eniten projektien koon ja kompleksisuuden arviointiin ovat järjestyksessä eniten käytetystä vähiten käytettyyn: Tarinapistet (eng. Story points), Työnoitus (WBS), Työajurit (eng. Work drivers), Funktiopisteet (eng. Function points) ja Koodirivien määrä (eng. Lines of Codes) (Nelson & Morris, 2014).

IT projektit kärsivät vieläkin huonoista projektien arvioinneista. Tutkimukset osoittivat jo 20-vuotta sitten tämän ongelman, mutta merkittäviä parannuksia ei ole siitä huolimatta näkynyt. Epärealistiset odotukset, jotka perustuvat epätarkkoihin arvioihin ovat suurin yksittäinen syy IT projektien epäonnistumiselle. (Nelson & Morris, 2014) Ongelma on siis

tiedostettu jo vuosikymmeniä, mutta toimivaa ratkaisua etsitään vieläkin. Yhtenä ongelmanratkaisun esteenä voidaan pitää sitä, että tarjouskilpailuissa yritykset, jotka ovat yli-optimistisia työmäärän suhteen voittavat monesti kilpailun (Jørgensen, 2007). Tämä on haasteena, koska yritykset, jotka arvioivat projektiaan kilpailijaa lyhyemmäksi voittavat vaikka tosiasiallisesti projekti on arvioitua pidempi. Asiakasyritysten tulisi vaatia tarkempia arvioita, jotta palvelun tarjoajat panostaisivat enemmän arvioinnin laatuun ja projektien onnistumisprosentti kasvaisi.

Suurimmat syyt epätarkalle arvioinnille epärealististen odotusten jälkeen ovat: jatkuva muutosten pyytäminen, käyttäjien omien vaatimusten ymmärryksen puute, tehtävien haastavuuden väheksyminen, heikko ongelmien määrittely sekä puutteellinen analyysi arviointivaiheessa (Nelson & Morris, 2014). Nämä haasteet tulee tiedostaa projektien arvioita tehdessä ja ne tulee ottaa huomioon mallia luodessa.

IT-alalla on useita malleja työmäärän arviointiin, yksinkertaisista monimutkaisiin. Haasteena näissä malleissa on kuitenkin toteutuksen kompleksisuuden arviointi analyysivaiheessa. (Jørgensen, 2007) Tiettyjen ominaisuuksien toteuttaminen saattaa analyysivaiheessa vaikuttaa yksinkertaisilta, joka loppujen lopuksi osoittautuu odotettua kompleksisemmaksi. Tätä kompleksisuuteen liittyvää haastetta voidaan pienentää jakamalla ohjelmisto, niin pieniin osiin ettei ongelmien vastaan tullessa projektin aikataulu tai laajuus kärsi. Tutkimukset osoittavatkin, että malleihin perustuvat arvioit ovat suoriutuneet asiantuntija arvioita paremmin tässä asiassa (Jørgensen, 2007). Asiantuntijoita tarvitaan arviointivaiheessa, mutta heidän roolinsa tulisi muuttua lopullisen arvion tekijöistä systemaattisten mallien hyödyntäjiksi.

IT-projektit käyttävät 30-40 prosenttia enemmän työtä projekteihin, kuin alun perin suunniteltiin. Tämä tilanne on ollut jo vuosia, mutta vain pientä kehitystä on ollut havaittavissa. (Jørgensen, 2007) Tämän takia on tärkeää löytää keinot parempaan työmäärän arviointiin, sillä suuri osa kustannuksista IT-projekteista muodostuu työntekijä kustannuksista. IT projekteissa huono arviointi aiheuttaa paineita aikataulullisesti, joka johtaa stressiin projektin sisällä ja tämä puolestaan altistaa projektin virheille. Huonon arvioinnin takia myös budjetin arviointi on haastavaa ja yrityksen kannattavuuden arviointi on epätarkkaa. (Nelson & Morris, 2014)

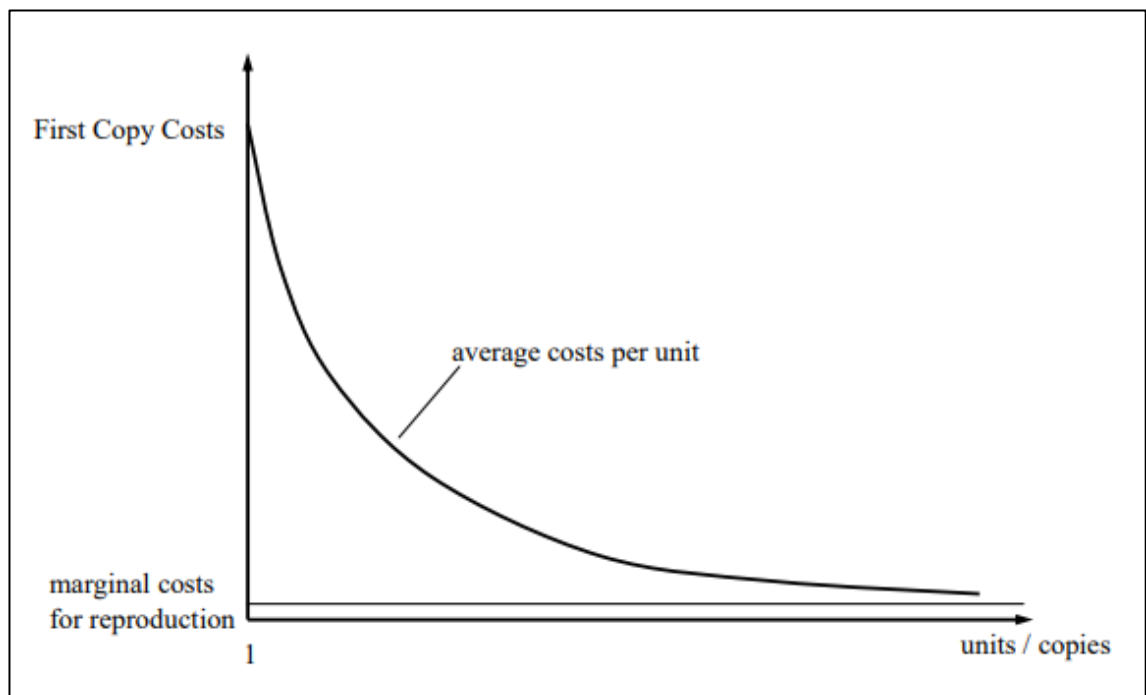
Projektin työmääräarviota voidaan siis tehdä joko asiantuntija-arviona tai malliin-pohjautuvalla arviointiprosessilla (Jørgensen, 2007). Näistä kahdesta vaihtoehdosta malliin pohjautuva arviointi on parempi vaihtoehto, sillä IT-alalla tällä hetkellä paljon käytetty asiantuntija-arviointi ei ole toiminut vuosikymmenten aikana (Nelson & Morris, 2014). Malleihin pohjautuvat projektien työmäärä arviot tarjoavat erityisesti kaksi ominaisuutta,

jotka ovat parempia verrattuna asiantuntija arvioihin: mallit ovat johdonmukaisia eli sama syöte tuottaa saman tuloksen ja mallit tekevät arvioita systemaattisesti ilman että joudutaan pohtimaan, mikä on validia dataa arviontiin liittyen ja mikä ei ole (Jørgensen, 2007).

Tutkimuksen (Nelson & Morris, 2014) johtopäätöksenä todettiin, että yhtenä tärkeimmistä keinoista parantaa projektin arviointia on ottaa käyttöön arviointi työkalut. Näiden työkalujen avulla on esimerkiksi mahdollista arvioida ja vertailla arvioita aikaisempien projektien datan kanssa sekä luoda aikatauluja, mitkä pitävät. Yrityksen on myös tärkeää päästä eroon intuition käyttämisestä sekä arvailusta. Jopa asiantuntija arvioinnit sisältävät paljon biaksia, mitkä ohjaavat projektin arviointia väärään suuntaan.

2.2.1 Kustannusten muodostuminen

Kustannuksiksi voidaan laskea kaikki muut markkinoiden elementit paitsi hinta (Woodside, 1994 s.140). Tuotteen tai palvelun tarjontaan liittyy paljon kustannuksia, jotka tulee ottaa huomioon. IT-alan kustannusrakenne poikkeaa huomattavasti muiden alojen kustannusrakenteesta. Yleisesti ottaen IT-alan kustannukset muodostuvat vahvasti First Copy cost -tyyppisestä kustannusten ilmiöstä. (Shapiro & Varian, 1999 s.21) Kuvassa 5. esitellyssä First Copy Cost -efektissä ensimmäisen tuotteen tekemiseen liittyy paljon kustannuksia, mutta uudelleen tuottamisen tai olemassa olevan ohjelmiston kopioimisen kustannukset ovat minimaaliset. Tämän takia tuotteen keskiarvo kustannukset pienenevät eksponentiaalisesti, kun tuotetta valmistetaan enemmän. (Anding & Hess, 2004)



Kuva 3. *First Copy Cost -efekti (Anding & Hess, 2004)*

First Copy Cost- efektissä suuri kustannuserä syntyy jo ennen kuin yhtään tuotetta on toimitettu ja käytännössä ottaen yhden tuotteen kustannukset ovat samat kuin miljoonan tuotteen kustannukset (Bontis & Chung, 2000 s.246). Kohdeyrityksen osalta liiketoiminta poikkeaa hieman. Kustannukset eivät synny ensimmäisen tuotteen valmistuksesta, vaan asiakasyrityksille projekti myydään työmääränä. Tässä tapauksessa suurin osa kustannuksista syntyy siis työvoimakuluista, joita mitataan henkilötyöpäivissä.

Ohjelmistoalan yritysten tuotteiden ja etenkin palvelujen myyntiin liittyy muitakin kustannuksia. Kustannuksia, joita ohjelmistotuotteiden sekä -palveluiden myyntiin liittyy ovat tuotanto (työvoima), jakelu, markkinointi, tekniset kustannukset sekä tukikustannukset. (Mayer, 1987). Yksi poikkeava tapa kustannusten arviointiin on määritellä kaikki kustannukset tuotantokustannuksiksi (Cooper & Kaplan, 1988). Tuotannon kustannuksiksi voidaan jakaa esimerkiksi skaalautuvuuteen liittyvät kustannukset sekä muutoksiin liittyvät kustannukset (Stalk, 1988). Ohjelmistotalalla kustannuksia syntyy myös jatkuvasta ylläpidosta ja korjauksista tai päivityksistä, joita ohjelmistoon julkaistaan (Chow & Cao, 2008)

Ohjelmistojen koon, kehitysajan ja kustannusten arviointi on ollut menneisyydessä pitkälti intuitiivisia ja etenkin suuremmissa projekteissa tämän perinteinen kustannusten arviointitapa ei ole toiminut. (Putnam, 1978) Suuriin projekteihin sisältyy paljon edellä mainittuja kustannuksia, jonka takia esimerkiksi kustannusten kohdistaminen tietylle tuotteelle tai palvelulle voi olla haastavaa.

Kohdeyrityksen osalta asiakkailta laskutetaan tehdyn työn mukaan, jonka takia kustannusten laskeminen on yksinkertaista. Kannattavuuden arvioinnin osalta kustannusten osuus on helppo laskea, mutta niiden arvioiminen projektien analyysivaiheessa on haastavaa, sillä ohjelmistojen tuottaminen tuo tullessaan odottamattomia haasteita projektin edetessä. Nämä odottamattomat haasteet saavat projektissa aikaan lisää työtunteja, joka kostautuu joko toimittajan kannattavuudelle tai asiakkaan lopulliseen hintaan.

2.2.2 Hinnoittelu

Hinnoittelu on merkittävä osa kannattavuutta ja sen suunnitteluun sekä hinnoittelu strategian muodostamiseen tulisi käyttää aikaa. Huolimatta hinnoittelun merkityksestä yritykset tekevät vain harvoin tutkimusta liittyen hinnoitteluun (Malhotra, 1996). Yrityksillä tulisi olla selvä strategia hinnoittelun suhteen ja strategia tulisi pohjautua tutkimustulok-

siin. Hinnoittelulla on suuri merkitys yrityksen kannattavuuden muodostumiselle. Kannattavuuden voidaan katsoa muodostuvan tuloista vähennetyistä kuluista. Tulot määräytyvät hinnan mukaan, joten projektin arviointivaiheessa on tärkeää olla perillä siitä, mihin hintaan projekti voidaan myydä, jotta yritys pääsee kannattavuustavoitteeseen. Lähtökohtana hinnoittelulle on yrityksen kannattavuuden parantaminen, sillä liiketoiminnan jatkuvuuden edellytyksenä on kannattavuus (Tähtinen & Parviainen, 2003)

Hinta voidaan jaotella kahteen eri tekijään, katteeseen ja yksikkökustannuksiin. Tuotteesta on saatava tarpeeksi hintaa kattaakseen valmistuskustannuksen sekä voittotavoite. Tämä on hinnoittelu taloudellisesta näkökulmasta. Psykologisella näkökulmalla voidaan katsoa, että hinta kertoo asiakkaalle tuotteen arvon. Hintaa vaikuttaa suuresti siihen, koetaanko tuote laadukkaaksi vai epälaadukkaaksi hinnan asettelun takia. Hinnoittelussa onnistumiseen vaaditaan kustannusten perusteellisen laskemisen lisäksi sitä, että yritys tuntee asiakaskuntansa ja tietää heidän odotuksensa. Tämän takia on tärkeää pitää mielessä se, että hinnoittelua tulee tarkastella sekä asiakkaan että yrityksen näkökulmasta niin, että tuotteen laadun ja hinnan välillä on kilpailukykyinen tasapaino. (Laitinen, E.K. 2007, 225)

Hinnoittelustrategioita on monia ja yrityksen toimiala vaikuttaa niihin paljon. Yleisesti ottaen hinnoittelustrategioita voidaan kuitenkin katsoa olevan kolme: kustannuspohjainen, kilpailijapohjainen- sekä arvopohjainen hinnoittelu. (Hinterhuber, 2008) On tärkeää, käydä kaikkia kolmea hinnoittelumenetelmää hieman läpi, sillä yrityksen hinnoittelustrategia on monesti yhdistelmä näistä kolmesta. Kustannuspohjainen hinnoittelustrategia perustuu kustannuksiin sekä tavoiteltuun voittoon, kilpailijapohjainen hinnoittelustrategia perustuu markkinoiden määräämään hintaan ja arvopohjainen hinnoittelustrategia perustuu asiakkaalle tuodun arvon määrään.

Kustannusperusteinen hinnoittelu on hinnoittelustrategioista yksinkertaisin ja sen avulla hinnoitellut tuotteet ovat helposti perusteltavissa asiakkaalle. Tässä hinnoittelustrategiassa hinta muodostuu tuotteen tuottamisesta, myymisestä sekä jakelusta aiheutuneista kustannuksista. Näihin kuluihin lisätään yrityksen tavoiteltu kate, joka pohjautuu muun muassa riskeihin. (Kotler & Armstrong, 2008) IT-alalla projektit sisältävät paljon riskejä ja projektin epäonnistumisen mahdollisuus on suuri. Tämä tulee ottaa huomioon hinnoittelussa, jotta kannattavuus tavoitteisiin päästään.

Kustannusperusteinen hinnoittelustrategia toimii parhaiten yksilöllisten tuotteiden tavoitteellista hintaa arvioidessa, projektien hinnoittelussa ja määrittäessä asennus- ja huoltopalveluiden hintoja (Neilimo ym. 2005,192) Kustannusperusteinen hinnoittelustrategia

on hinnoittelustrategioista toiseksi yleisin, sen käyttöprosentin ollessa 37%. Tätä hinnoittelustrategiaa edellä on ainoastaan kilpailuperusteinen hinnoittelustrategia, mitä käytetään 44% yrityksissä. (Hinterhuber, 2008)

Kustannusperusteinen hinnoittelu on IT-alalla suosittua, sillä pohjautuu olemassa olevaan informaatioon, mikä saadaan kirjanpito järjestelmistä. IT- ja finanssijohtajat hinnoittelevat IT-palvelut niin, että kustannukset huomioon ottaen vastataan tavoiteltuun voittoon. Tässä hinnoittelumenetelmässä on helppo ottaa huomioon ROI-luku, sillä kustannukset ja tavoiteltu voitto tiedetään jo tuotteen tai palvelun myyntivaiheessa. (Harmon et al. 2009)

Arvoperusteisen hinnoittelustrategian periaatteena on hinnoitella tuote asiakkaalle tuovan arvon perusteella. Tämän hinnoittelustrategian mukaan hyvän hinnoittelun lähtökohdana on ymmärtää, millaista arvoa tuote tuottaa asiakkaalle. (Kotler & Armstrong, 2008) Asiakkaalle voidaan tuottaa arvoa monella tapaa. Arvoa voi tuottaa esimerkiksi myyty tuote tai palvelu, projektin johtaminen tai yrityksen brändi.

Arvoperusteinen hinnoittelustrategia ei ole kaikista suosituin menetelmä, sillä vain 17% yrityksistä toteuttaa hinnoittelunsa arvoon pohjautuvalla menetelmällä. Useat teoreetikot ovat kuitenkin arvopohjaisen hinnoittelustrategian puolesta puhujia sen tuomien etujen takia. (Hinterhuber, 2008) Arvoperusteinen hinnoittelu on toimiva strategia silloin, kun asiakkaat kokevat tuotteen tai palvelun olevan paljon arvokkaampi mitä sen tuottamiskustannukset ovat. IT-alan osalta esimerkiksi yritykset, jotka myyvät lisenssejä tai valmiiksi tehtyjä tuotteita voisivat hyödyntää arvoperusteista hinnoittelustrategiaa. Arvoperusteisen hinnoittelun tavoitteena on mahdollistaa pitkän aikavälin tuotto saamalla lisää arvoa tuotteesta tai palvelusta (Harmon et al. 2009).

Arvoperusteisen hinnoittelumenetelmän kiinnostuksen lisääntymisestä IT-alalla kirjoitettiin kuitenkin vuonna 2007 (Ohsugi et al. 2007), mutta siitä huolimatta suosio on jäänyt vähäiseksi. Yhtenä ongelmana arvoperusteisessa hinnoittelustrategiassa IT-alalla on hinnan muodostumisen selittäminen asiakkaalle myyntivaiheessa. Asiakkaat haluavat monessa tapauksessa tietää tarkkaan, mistä hinta muodostuu ja tietotyön tapauksessa hinta muodostuu pääosin työntekokustannuksista. Hyvä maineiset yritykset voivat hyödyntää tätä hinnoittelustrategiaa onnistuneemmin, sillä heidän projektien arvonsa muodostuu osittain brändin perusteella.

Kilpailuperusteisen hinnoittelustrategian periaatteena on, että hinta muodostuu markkinan yleisen hintatason mukaan. Yrityksen tulee tässä hinnoittelustrategiassa lähteä liikkeelle, siitä mikä yleinen hintataso markkinoilla on, jonka jälkeen yrityksen tulee verrata oman tuotteen ominaisuuksia markkinoiden tuotteisiin ja tämän pohjalta muodostaa

hinta. (Richards et al. 2005) Kilpailuperusteinen hinnoittelustrategia on yleinen monesta syystä, mutta yhtenä syynä voi olla sen yksinkertaisuus. Yrityksen tuodessa uutta tuotetta markkinoille, sen on selvitettävä markkinoiden yleinen hintataso, jonka jälkeen yritys voi pohtia pystyykö tuottamaan kilpailevaa tuotetta markkinoiden hintaan niin, että se pääsee kannattavuus tavoitteisiinsa. Kuvassa 6. on esitelty kustannusperusteisen- ja arvoperusteisen hinnoittelun erot IT-palvelun hinnoitteluprosessin edetessä.



Kuva 4. Kustannus- ja arvoperusteisen hinnoittelun erot IT-palvelun hinnoittelu prosessissa (mukaillen Harmon et al. 2009)

Kuvasta 6. voidaan nähdä, että hinnoittelustrategian valinnalla on huomattava ero IT-palvelun hinnoittelussa. Kustannusperusteisessa hinnoittelussa lähdetään liikkeelle itse tuotteesta, johon liittyy kustannuksia. Kustannusten päälle lisätään haluttu voittomarginaali ja tästä muodostuu palvelun hinta. Tämän jälkeen palvelun arvo hahmotetaan ja palvelu myydään asiakkaalle. Arvoperusteisessa hinnoittelussa puolestaan lähetään liikkeelle asiakkaasta ja arvon hahmottamisesta. Tästä arvosta muodostuu palvelun hinta, jonka jälkeen palvelun kustannukset määritellään. Kustannuksien jälkeen jäljellä jää jäännös marginaali ja lopuksi IT-palvelu astuu kuvaan.

2.2.3 Riskienhallinta

Riskienhallinta on tärkeä osa kaikkea liiketoimintaa. Riskin suuruus määräytyy riskin toteutumisen todennäköisyyden ja riskin toteutuessa syntyneen vahingon tulona. Ohjelmistokehityksen osalta riskin suuruus määräytyy puolestaan projektin epävarmuuden sekä projektin epäonnistumisesta syntyvän potentiaalisen häviön suuruuden tulosta. (Barki et al. 1993). Ottaessa riskienhallintaan käyttöön projekteissa se voi näyttää lisäävän kompleksisuutta valmiiksi monimutkaisiin hankkeisiin. Ongelmien proaktiivinen käsittely on kuitenkin parempi ratkaisu kuin hoitaa niitä reaktiivisesti. (McManus, 2004 s.4)

Riskienhallinta tunnetaan tärkeänä osana projektien johtamista. Riskienhallinnan avulla voidaan käsitellä paremmin eteen tulevia ongelmia, minimoida uhkia ja maksimoida mahdollisuuksia keskittyen, silti projektin etenemiseen. (Hilson, 2003) Riskien suuruuden arvioinnissa on huomattu olevan haasteita ohjelmistoprojektien osalta jo vuonna 1993 (Barki et al. 1993).

Wallace & Keil (2004) toteavatkin, että projektin johtajat hyötyisivät paljon riskien merkityksen ymmärtämisestä projektien lopputuloksen kannalta. Paremman ymmärryksen avulla olisi mahdollista alentaa projektien epäonnistumisprosenttia (Wallace & Keil, 2004). Riskienhallinnan arvon voidaan katsoa muodostuvan säästetyistä projektin tuotannon kustannuksista. Arvo ei kuitenkaan muodostu itsestään vaan yrityksen on nähtävä vaivaa riskien hallinnan tehokkaaseen hyödyntämiseen. (McManus, 2004 s.14) Riskeihin tulee puuttua mahdollisimman ajoissa, tehokkaammin ja nopeammin, jotta taloudelliset sekä aikatauluihin liittyvät haasteet eivät pääse toteutumaan (Sommerville, 2007 s.104).

Riskienhallinta tekee ohjelmistoprojekteista todellisuudessa vähemmän monimutkaisia. Riskien tunnistus sekä priorisointi mahdollistaa projektin henkilöiden keskittymisen projektin tärkeisiin osa-alueisiin. Riskien minimointitoiminnot vähentävät projektin riskiä kokonaisuudessa ja nopeuttaa projektin valmistumista. Riskien toteutumisen ehkäisemisen takia projektit pysyvät aikataulussaan ja täten projektin kannattavuus pysyy toivotulla tasolla. Riskeihin varautuessa yllätysten todennäköisyys projektissa pienenee ja kannattavuutta sekä projektia kokonaisuudessaan on helpompi arvioida jo projektin alkuvaiheessa. Riskienhallinta edistää siis resurssien tehokkaampaa käyttöä, aikataulujen pitämistä sekä asiakastyytyväisyyttä. (McManus, 2004 s.4)

Ohjelmistoprojektit sisältävät paljon tekijöitä, jotka voivat muuttuvat projektien edetessä. Projektin laajuuden ja pituuden kasvaessa riskien mahdollisuus kasvaa (Coram & Bohner, 2005). Laajojen projektien hankaluus syntyy niiden kasvavasta kompleksisuudesta, ominaisuuksien määrästä sekä projektihenkilöstön vaihtumisesta (Coram & Bohner, 2005). Erinäiset riskitekijät, joista ohjelmistotuotannossa voi syntyä vahinkoa ovat: Ohjelmiston koko, tiimin koko, tekninen kompleksisuus, ulkopuoliset toimittajat, uudet teknologiat, uudet toiminnallisuudet, tiimin monimuotoisuus, tiimin ammattitaito, johtajien kokemus, käyttäjien asenteet, konfliktit projektin sisällä, ylimmän johdon tuki (Barki et al. 1993).

Sommerville (2007) mukaan riskienhallinta prosessi jaetaan neljään eri vaiheeseen:

1. Riskin tunnistaminen
2. Riskianalyysi
3. Riskisuunnittelu
4. Riskinvalvonta.

Riskien tunnistamisen vaiheessa tarkoituksena on havaita olemassa olevat riskitekijät. Tämän jälkeen riskianalysissä tehdään arvio siitä, kuinka merkittävä riski on kyseessä. Riskisuunnittelun ideana on suunnitella toimenpiteet, jokaiselle riskille. Osa riskeistä vaatii välittömiä toimenpiteitä, osa voidaan jättää huomioimatta kokonaan. Riskinvalvonnassa riskien hallinnan toimenpiteitä sekä riskejä arvioidaan jatkuvasti, jotta riskien hallinta olisi tehokkaampaa. (Sommerville, 2007 s.106)

Sommerville (2007) jakaa riskit kolmeen kategoriaan, jotka jakautuvat kuuteen tyyppiin. Kolme riskien kategoriaa ovat:

1. Projektiriskit
2. Tuoteriskit
3. Liiketoimintariskit. (Sommerville, 2007 s. 104-105)

Projektiriskeiksi kategorisoidaan sellaiset riskit, jotka vaikuttavat projektien aikatauluihin tai resursseihin. Esimerkiksi ohjelmistoprojektin avain- eli tärkeimmän henkilön menetys. Tuoteriskit puolestaan ovat riskejä, jotka ovat suoraan vaikutuksessa ohjelmistoprojektin tuotteen laatuun tai suorituskykyyn, esimerkiksi ohjelmiston koko kasvaa liian suureksi laskentatehon kannalta. Ohjelmistoa tuottavaan yritykseen suoraan liittyvät riskit kuuluvat liiketoimintariskit-kategoriaan, esimerkiksi kilpailijoiden ehtiminen aikaisemmin markkinoille samankaltaisen tuotteen kanssa. (Sommerville, 2007 s. 104-105) Kuusi riskityyppiä jaetaan seuraavalla tavalla:

1. Teknologiariskit
2. Henkilöriskit
3. Organisaatoriskit
4. Työkaluriskit
5. Vaatimusriskit
6. Arviointiriskit. (Sommerville, 2007 s.107)

Teknologiariskeilla tarkoitetaan riskejä, jotka liittyvät ohjelmistokehityksen ohjelmistoihin sekä laitteisiin. Henkilöriskit ovat riskejä, jotka liittyvät ohjelmistoprojektissa toimiviin henkilöihin. Organisaatio riskit puolestaan sisältävät riskejä, jotka syntyvät projekti organisaation ympäristön vaikutuksesta. Työkaluista sekä niiden toiminnasta liittyvät riskit jaotellaan työkaluriskeihin ja asiakkaan vaatimuksien muutoksiin liittyvät riskit jaotellaan vaatimusriskeihin. (Sommerville, 2007 s.107)

Riskienhallinnan yhtenä haasteena ohjelmistoprojekteissa voidaan pitää riskin suuruuden arvioimista. Monet riskeiksi tiedostetut tekijät on helppo tunnistaa, mutta niiden suuruus voi olla haastavaa arvioida. Ketterissä ohjelmisto projekteissa riskit tulee ottaa huomioon työmäärä arvioissa, joka on vaikeaa ilman systemaattista lähestymistapaa. Yksi mahdollinen riskin suuruuden arviointimenetelmä on OCTAVE Allegro, joka on esitelty kuvassa 7.

Impact Area	Ranking	Impact Value	Score
Reputation	4	Moderate (2)	8
Financial	5	Low (1)	5
Productivity	3	Low (1)	3
Safety and Health	1	Low (1)	1
Fines/Legal	2	High (3)	6
Total Score			23

Kuva 5. OCTAVE Allegro -menetelmä (Caralli, 2007 s.57)

OCTAVE Allegro -menetelmässä on viisi riskikategoriaa, jotka sijoitetaan tärkeys järjestykseen tärkeimmästä alaspäin. Tämän jälkeen kukin riskikategoria arvioidaan yhdestä-

kolmeen arvosanalla, joka määräytyy riskin suuruudesta. Tämän jälkeen jokainen riskikategoria arvostellaan tuloksella, joka muodostuu kategorian tärkeyden ja riskin suuruuden tulona. Tämä on yksi esimerkki systemaattisesta lähestymistavasta, jota riskien hallinnassa voidaan hyödyntää. Ketterissä ohjelmistoprojekteissa tulisi hyödyntää systemaattisia menetelmiä riskien hallinnan osalta, ja näissä menetelmissä tulisi ottaa huomioon riskin suuruus sekä sen toteutumisen todennäköisyys. Riskin suuruuden määräytyessä sen arvon muuttaminen asiakkaalle esitettävään muotoon on haastavaa.

3. KANNATTAVUUDEN ARVIOINTIKEINOT

3.1 Työmäärän arviointi ohjelmistoprojekteissa

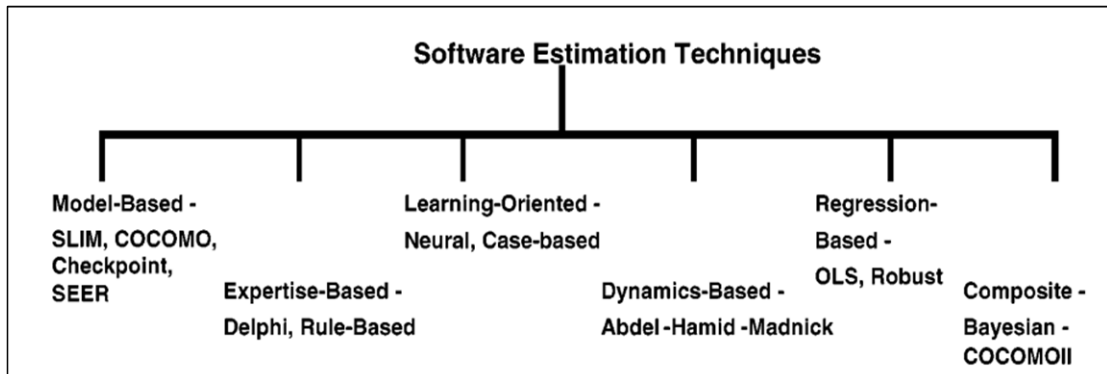
Työmäärän arvioinnissa hyödynnettävän mallien luomisen suhteen on tärkeää ottaa huomioon, että toimivassa mallissa ei saa olla liikaa muuttujia, sen on oltava tarpeeksi yksinkertainen, sen tekeminen ei voi olla liian monimutkaista ja malliin tulee löytää oikeat muuttujat (Jorgensen, 2007). Tämä on haastavaa, sillä ohjelmistoprojekteissa on paljon muuttujia, joista valita ja kunkin muuttujan painoarvoa on vaikeaa arvioida. Esimerkiksi riskien suuruuden määrittäminen arviointivaiheessa voi olla suuri haaste, sillä ohjelmistoprojekteissa tulee jatkuvasti vastaan haasteita, joita ei osattu odottaa.

Asiantuntija-arviot ovat parempi tilanteissa, joissa lähtökohtana on paljon kontekstuaalista informaatiota sekä tilanteissa, joissa projektin olosuhteet muuttuvat. Tutkimukset osoittavat myös, että mallien hyödyntäminen asiantuntija arvioiden kanssa on ollut tois-tuvasti parempi ratkaisu, kun kummankaan hyödyntäminen yksittäin. (Jorgensen, 2007) Osa arviointimenetelmistä voidaan jaotella sekä systemaattisiksi menetelmiksi että asi-antuntija-arvio menetelmiksi (Jørgensen et al. 2009).

Asiantuntijoiden arvioiden hyödyntäminen on yleisin menetelmä työmäärän arvioinnissa. Syynä tähän on todennäköisesti se, että systemaattisten mallien hyödyntäminen ei ole johtanut tarkempiin arvioihin. Tutkimuksen mukaan systemaattisia malleja hyödynnettiin vain 17% projekteista. Hyödynnetyt malleja oli muun muassa COCOMO, Use-Case-based -menetelmä sekä Function point -menetelmä. (Moløkken & Jørgensen, 2003) Tä-män tutkimuksen tulokset osoittavat myös systemaattisen mallin hyödyntämisen olevan haastavaa. IT-projekteja on ollut jo vuosikymmeniä, mutta niiden työ-määräarviointiin tu-lisi näidenkin tutkimustulosten pohjalta keksiä ratkaisuja.

Ohjelmisto projektien arvioon on useita tekniikoita, mutta mikään yksittäinen tekniikka ei ole osoittanut olevansa paras kaikissa tilanteissa. Eri tekniikat sopivat eri tilanteisiin ja tekniikoiden yhdistäminen voi olla paras keino parhaisiin arviointi tuloksiin. (Boehm et al. 2000) Kuvassa 8. on esiteltyä yleisimmät tekniikat, joita työmäärän arviointiin ohjelmis-toprojekteissa voidaan hyödyntää. Näitä hyödynnettäviä tekniikoita ovat malliin pohjau-tuvat tekniikat (eng. Model-Based), asiantuntija-arviot (eng. Expertise-Based), kokemuk-seen pohjautuvat tekniikat (eng. Learning-Oriented), dynamiikka perusteiset tekniikat

(eng. Dynamics-Based), regressio perusteiset (eng. Regression-Based) sekä hybridimallit (Composite-Based).



Kuva 6. Työmäärän arviointitekniikat (Boehm et al. 2000)

Kuvassa 8. on esitelty myös tekniikoiden lisäksi esimerkkejä spesifeistä menetelmistä, joita tekniikoihin kuuluu. Tässä tutkimuksessa käsitellään tarkemmin kuvassa esiteltyjä COCOMO- ja Delphi-malleja sekä muutamaa menetelmää, jotka puuttuvat kuvasta. Näiden lisäksi tutkimuksessa käsitellään työn osituksessa yleisesti käytettyä WBS-menetelmää (Work Breakdown Structure), joka on osana useaa työmäärän arviointi menetelmää.

Ensimmäinen systemaattinen ohjelmiston arvioimistyökalu luotiin 1970-luvun alkupuolella. Ensimmäinen tällainen malli oli prototyyppi COCOMO-mallista, jonka loi Barry Boehm kirjaansa *Software Engineering Economics*. (Suri & Ranjan, 2012) Vuoteen 2012 mennessä oli syntynyt jo useita metodeja mitata ohjelmistojen työmääriä, mutta niitä ei oltu otettu laajasti vielä käyttöön. (Suri & Ranjan, 2012) Myös vuoden 2012 jälkeen on julkaistu useita erilaisia menetelmiä, mutta mikään ei ole osoittanut olevansa ylivoimaisesti toisia parempia.

Yksi keino arvioida projektien työmääriä ja kannattavuutta on vertailla niitä aikaisempiin projekteihin, jotka ovat olleet ominaisuuksiltaan vastaavanlaisia. Tätä metodia on käytetty pitkään ja systemaattisten mallien heikkouksien vuoksi se on vieläkin laajasti käytössä. (Suri & Ranjan, 2012) Keino on yleinen, sillä parempia keinoja ei ole vielä löydetty. Asiantuntijoiden arviointi kyvyyn on koettu olevan tarkin ja asiantuntijoiden arvioit pohjautuvat pitkälti heidän kokemuksiinsa aikaisemmista projekteista.

Asiantuntija-arvioita tarvitaan erityisesti, kun asian mukaista dataa ei ole saatavilla (Suri & Ranjan, 2012). Tällaista dataa voi olla esimerkiksi historiatieto vastaavanlaisista projekteista tai systemaattisen mallin täyttämiseen tarvittava pohjatieto. Käyttökelpoisena asiantuntijoiden apuvälineenä pidetään esimerkiksi Delphi-menetelmää sekä WBS-menetelmää.

Suurin eroavaisuus systemaattisten mallien ja asiantuntija-arvioiden välillä syntyy niiden määrittämisvaiheessa. Tässä vaiheessa Inputista syntyy työmääräarvio. Systemaattisissa malleissa määrittämisvaiheessa arviot perustuvat mekaaniseen määrittämiseen kuten erilaisiin kaavoihin. Asiantuntija arvioissa määrittäminen puolestaan muodostuu asiantuntijaa tukevien menetelmien hyödyntämiseen, kuten WBS-menetelmään. Asiantuntija-arviot voivat syntyä puhtaasti tunteen perusteella ja pohjautuen WBS-menetelmiin tai aikaisempiin projekteihin. Systemaattisia menetelmiä hyödyntäessä arvioiden tausta ei vaihtelee vaan ne pohjautuvat aina kaavoihin ja alkuarvoihin. (Jørgensen et al. 2009)

Hyvässä työmäärän arvioinnissa tulisi olla jonkin verran rakeisuutta, jotta se on selitettävissä, koska työmäärän arvio on yksi eniten analysoitu ja tärkein osa-alue, jota analysoidaan projektin alussa (Suri & Sanjan, 2012) Monien syiden takia voidaan nähdä, että tulevaisuudessa arviointi prosessi tulisi pohjautua systemaattisen mallien hyödyntämiseen ja tutkimuksia tulisi kohdistaa niiden kehittämiseen. Asiantuntija-arvioissakin on paljon hyvää, mutta esimerkiksi ennakkoluulot ja toiveikas ajattelutapa heikentää niiden tarkkuutta. Kaikkiin tapauksiin hyödynnettävää mallia ei kuitenkaan kannata tehdä eikä sellaista ole mahdollista tehdä, sillä ohjelmistoprojektit vaihtelevat paljon ja asiantuntijoita tai systemaattisia malleja ei kumpaakaan tulisi hylätä täysin. (Jørgensen et al. 2009) Tämän takia työmäärän arvioinnissa voisi yhdistellä useampaa systemaattista mallia tai osaa niistä sekä asiantuntijoiden tuomaa näkemystä.

Vuonna 1987 tehdyssä tutkimuksessa Keremer testasi neljää (SLIM, COCOMO, Estimations, FPA) systemaattista mallia työmäärän arvioinnissa projekteissa, jotka eivät olleet ympäristöltään täysin samanlaisia kuin mallien luomisvaiheessa ollut ympäristö. Dataa ei tutkimuksessa muokattu lainkaan sopimaan malliin paremmin, vaan projektien data syötettiin malliin alkuperäisessä kunnossa. Tutkimus osoitti, että mallit arvioivat työmäärä arviot vahvasti yläkanttiin. (Finnie et al. 1997) Tämä osoittaa sen, että ainakaan nämä systemaattiset mallit eivät toimi universaalisti projektista ja sen ympäristöstä riippumatta. Tämä on yksi syy, miksi tutkimuksen konstruktiossa yhdistetään useaa mallia eikä luoteta vain yhden mallin toimivuuteen.

Työmääräarvioon on ehdotettu vuosien varrella useita eri menetelmiä kuten COCOMO, SLIM, Estimacs, Function Point Analysis (FPA), SPANS, Checkpoint ja COSTAR (Finnie

et al. 1997). Kaikkia menetelmiä ei ole järkeä käsitellä tässä tutkimuksessa, mutta kohdeyrityksen toiminnan kannalta tärkeimmät menetelmät on otettu tarkempaan käsitteelyyn. Seuraavissa alaluvuissa on käsitelty tämän tutkimuksen osalta tärkeimmät työmäärään tai kannattavuuden arviointiin käytettävät menetelmät. Käsiteltäviä menetelmiä ovat: Work Breakdown Structure, Program Evaluation and Review Technique, Function Point Analysis, COCOMO, Wisdom of Crowds sekä Delphi-menetelmä. Nämä menetelmät valittiin käsiteltäväksi sillä ne tukevat tutkimuksen tuloksena syntyvää konstruktiota tai ne ovat muuten merkittäviä menetelmiä ohjelmistoprojektien kannattavuuden arviointiin liittyen.

3.1.1 WBS

Work Breakdown Structure (WBS) -menetelmä on työnositus-menetelmä, joka luotu jakamaan toimitettava projekti hierarkkisesti osiin, jotta projektiryhmä voi seurata työn etenemistä ja saavuttaa työn etenemiseen liittyvät tavoitteet. Menetelmän avulla voidaan organisoida projektia ja se määrittelee projektin alussa koko projektin laajuuden. Menetelmän avulla kokonaistyö jaetaan osiin, määrittellen kauan osan toteuttaminen vie aikaa ja mahdollisesti määrittellen kuka projektin jäsen työosan toteuttaa. (Brotherton, 2008) WBS-menetelmää käytetään siis projektin alkuvaiheessa määrittelemään projektin laajuus jakamalla koko projekti halutun kokoiisiin osiin. Ohjelmistoprojektien osat ovat jaettu usein ohjelmiston tiettyjen ominaisuuksien mukaan, jotta projektin laajuus voidaan määrittää. Ohjelmistoprojekteissa on tärkeää tehdä työnositus huolellisesti, jotta aikataulujen liialliselta venymiseltä vältytään.

WBS on keino jakaa toimitettava projekti pienempiin projekteihin tehtäviin, sivutehtäviin, ominaisuuksiin ja niin edelleen yhä pienempiin ja pienempiin kokonaisuuksiin. WBS-menetelmä on tärkeä suunnittelutyökalu, joka yhdistää tavoitteet resursseihin ja toimintoihin. Jo vuonna 1979 WBS-menetelmä oli laajasti käytössä useilla aloilla, mutta ohjelmistoprojektien osalta se ei ollut vielä suosittu. Tausworthe kuitenkin osoittaa, että WBS-menetelmästä olisi ollut hyötyä jo vuonna 1979. (Tausworthe, 1979) Nykyaikana WBS-menetelmä on laajasti käytössä myös ohjelmistoalalla ja työnosittelu suunnitteluvaiheessa on edellytyksenä onnistuneella projektille. WBS-menetelmän voidaan katsoa olevan jopa edellytyksenä onnistuneelle projektienhallinnalle (Globerson, 1994). Haasteena ohjelmistoprojektien osalta on työnosittelu tarpeeksi pieniin osiin. Ominaisuuksiin jakaminen on hyvä lähtökohta, mutta työmääräarviointi kuhunkin ominaisuuteen voi olla monessa tapauksessa haastavaa.

WBS-menetelmä on perusta projektienhallinnassa käytettyjen prosessien aloittamiselle, suunnittelulle, toteuttamiselle sekä seurannalle ja hallitsemiselle. Huonosti rakennettu WBS voi johtaa haitallisiin tuloksiin, se voi haitata projektisuunnitelman tekoa, työtehtävien jakoa, laajuuden määrittelyä sekä loppujen lopuksi budjetin venymistä, joka johtaa kannattavuuden laskuun. (Brotherton, 2008) WBS-menetelmän hyödyntämiseen tulee siis käyttää aikaa ja vaivaa projektin suunnitteluvaiheessa, jotta haittavaikutuksilta vältytään. Onnistunut WBS-menetelmä tuo kuitenkin paljon lisäarvoa projektille, joten sitä on syytä hyödyntää myös ohjelmistoprojekteissa.

WBS-menetelmän avulla projektin kustannukset ja aikataulut on mahdollista integroida yhteen menetelmään. WBS tarjoaa mahdollisuuden yhdistää projektin avaintoiminnot yhteen ja luoda niiden välille yhteys. WBS-menetelmän hierarkkisten kerrosten määrä vaikuttaa paljon siihen, kuinka tarkka projektin työnositus on ja kuinka kompleksinen kokonaisuus menetelmästä syntyy (Jung & Woo, 2004) Hierarkkisten kerrosten määrä sekä työnosien tarkkuus riippuu projektista. Esimerkiksi kolme kuukautta kestävässä ohjelmistoprojektissa on syytä eritellä ominaisuudet pieniksi kokonaisuuksiksi, jotta aikataulu ei veny, mutta kaksi vuotta kestävässä rautatieprojektissa työnosittelu pieniin yksityiskohtiin voi olla liian työlästä ja se voi laskea menetelmän hyödyntämisestä syntyvää arvoa.

WBS integroi kustannukset ja aikataulutuksen yhteen, sillä tutkimukset ovat osoittaneet tämän olevan paras keino hallita projektien kannattavuutta. Kustannukset ja aikataulut ovat läheisessä yhteydessä toisiinsa, sillä ne jakavat paljon yhteistä informaatiota. Tämän takia kustannus- ja aikataulunhallintatoimintojen integrointi tarjoaa tehokkaan työkalun projektin kannattavuuden seuraamiseen. (Jung & Woo, 2004) WBS-menetelmä tarjoaa siis useita hyötyjä kannattavuuden hallintaan ja arviointiin. WBS-menetelmän avulla jaettuja kokonaisuuksia on helpompaa hallita ja aikatauluttaa. Ohjelmistoprojekteissa työnosituksen jälkeen ominaisuuksien jakaminen tiimeille tai projektin henkilöille onnistuu jo projektin alkuvaiheessa, mikä vähentää resurssiongelmia projektiin edetessä.

3.1.2 Program Evaluation and Review Technique

Program Evaluation and Review Technique (PERT) -menetelmä lähtee liikkeelle WBS-menetelmässä luodusta työnsoittelusta. Projekti jaetaan halutun kokoiisiin osiin, esimerkiksi ohjelmistoprojektien osalta projekti voidaan jakaa WBS-menetelmää hyödyntäen ohjelmiston ominaisuuksiin. Tämän työn osittelun jälkeen projektista luodaan toimintaverkko, joka muodostuu jaettujen tehtävien pohjalta. Toimintaverkon tarkoituksena on esittää projektin kulku tehtävä kerrallaan. Toimintaverkossa jokainen tehtävä jaotellaan

verkostoon, josta muodostuu eri reittejä projektin lopputulokseen eli valmiiseen tuotteeseen. (Cook, 1966)

Työnosittelun sekä toimintaverkon havainnollistamisen jälkeen projektista on muodostunut kokonaiskuva, jota on tarkoitus arvioida. PERT-menetelmä on syntynyt siitä lähtökohdasta, että tarkkoja aikoja projektin valmistumiselle ei olla tiedetty, mutta arviosta on joka tapauksessa tehtävä mahdollisimman tarkka. PERT-menetelmän avulla projektin eri työtehtävät ja eri reitti vaihtoehdot arvioidaan pessimistisellä sekä optimistisellä lähestymistavalla. PERT-menetelmän ideana on siis luoda koko projektin kestolle pessimistinen sekä optimistinen arvio riippuen siitä mitä reittiä toimintaverkkoa kuljetaan. PERT-menetelmässä erireitit antavat erilaisia työmääräarvioita, joista voidaan laskea optimistisin sekä pessimistisin arvio. Menetelmässä lasketaan myös keskihajontaa, jotta arvioiden paikkaansa pitävyyksiä voidaan arvioida. (Cook, 1996) Tämän menetelmän voidaan nähdä arvioivan siis riskien toteutumista, optimistisillä arvioilla sekä pessimistisillä arvioilla.

PERT-menetelmää voidaan siis hyödyntää niin, että projektia arvioidaan eri skenaarioiden kautta. Näitä näkökulmia on pessimistinen skenaario, jossa kaikki epäonnisuus ja riskit toteutuvat sekä optimistinen skenaario, jossa kaikki onnistuu eikä yksikään riski toteudu. (Juvonen, 2018) Ohjelmistoprojekteja on arvioitu myös menetelmällä nimeltä Three Point Estimation, jossa pessimistisen ja optimistisen skenaarion lisäksi arvioidaan todennäköisin skenaario. Tämän tutkimuksen lopputuloksen kannalta PERT-menetelmästä sekä Three Point Estimation-menetelmästä käyttökelpoisimpana osana on hyödyntää optimistista, todennäköisintä sekä pessimististä arvioita konstruktiossa.

3.1.3 Function Point Analysis -menetelmä

Function Point Analysis (FPA) -menetelmä on strukturoitumenetelmä ongelmanratkaisuun. Tämän menetelmän avulla ongelma jaetaan pieniksi komponenteiksi, jotta ne voidaan ymmärtää ja analysoida paremmin. Menetelmän periaate on ihmiselle tuttu, sillä ihmiset ovat tottuneet ratkaisemaan ongelmat jakamalla ne pienemmiksi, helpommin ymmärrettäviksi osiksi. Ongelmat, jotka saattavat aluksi vaikuttaa vaikeilta ja kompleksisilta ovat yksinkertaisia, kun ne on jaettu pienempiin osiin. (Longstreet, 2002)

FPA-menetelmä syntyi jo vuonna 1975. FPA -menetelmän tarkoituksena on arvioida ohjelmistoprojektien kokoja sekä kehittämiseen kuluva työmäärää. Kokoa ja työmäärää mitataan viidellä Function Point -ominaisuudella. Näitä toiminnallisia ominaisuuksia ovat: Inputs, Outputs, Inquires, Logical Files ja Interfaces. (Suri & Ranjan, 2012) Allen Alhrecht kehitti FPA-menetelmä alun perin mittaamaan liiketoimintatietojärjestelmiä (Symons,

1988). Ensimmäisen kerran FPA-menetelmä otettiin käyttöön jo 1970-luvun loppu puolella (Longstreet, 2002).

FPA-menetelmässä järjestelmän toiminnallisuudet jaetaan viiteen luokkaan tai järjestelmän ominaisuuteen. Kolme ensimmäistä ominaisuutta liittyvät ulkoisiin toimintoihin, Outputs, Inputs ja Inquire. Näissä kolmessa ominaisuudessa siirtyy tiedostoja, joten niitä kutsutaan tapahtumiksi. Kaksi seuraavaa, Logical Files ja Interfaces, ovat ominaisuuksia, joissa tietoa tallennetaan. Ohjelmistoprojektin kokoa arvioidessa kokonaistyömäärä jaetaan alkuvaiheessa osiin, esimerkiksi WBS-menetelmän avulla. Tämän jälkeen osia arvioidaan niiden kompleksisuuden perusteella, kompleksisuuden asteikot voivat olla esimerkiksi matala, keskiverto ja korkea. Kullekin asteikolle annetaan jokin painoarvo, esimerkiksi matalan kompleksisuuden painoarvo voi olla yksi, keskiverron 3 ja korkean 5. (Longstreet, 2002)

Tämän jälkeen ohjelmistoprojektin osat jaetaan johonkin viidestä FPA-menetelmän osasta, sen kompleksisuuden mukaan. Kun projektin kaikki toiminnalliset osat on jaettu johonkin viidestä FPA-menetelmän osasta, on aika laskea kokonaismäärä työlle käyttämällä aluksi määriteltyjä painoarvoja sekä toiminnallisten osien määriä. (Longstreet, 2002) Kuvassa 9. on havainnollistettu taulukon avulla, miten FPA-menetelmää hyödynnetään.

Type of Component	Complexity of Components			Total
	Low	Average	High	
External Inputs	x 3 =	x 4 =	x 6 =	
External Outputs	x 4 =	x 5 =	x 7 =	
External Inquiries	x 3 =	x 4 =	x 6 =	
Internal Logical Files	x 7 =	x 10 =	x 15 =	
External Interface Files	x 5 =	x 7 =	x 10 =	
Total Number of Unadjusted Function Points				
Multiplied Value Adjustment Factor				
Total Adjusted Function Points				

Kuva 7. FPA-menetelmän yhteenlaskutaulukko (Longstreet, 2002)

Kuten kuvassa 9. näkyy, kunkin FPA-menetelmän ominaisuuden riville lasketaan painotettu tulos. Lopuksi rivit lasketaan yhteen, jolloin saadaan ohjelmiston koon arvio työmääränä. Tämän jälkeen menetelmässä voidaan käyttää vielä lasku kerrointa työmäärän arvioinnin säätämiseen. FPA-menetelmä on käyttökelpoinen työkalu työmäärän arviointiin jo projektin alussa ja FPA-menetelmän lopputuloksena saatu luku on otettu käyttöön

yleisenä ohjelmiston koon mittarina (Longstreet, 2002). Suurimpana heikkoutena mallissa voidaan nähdä olevan sen riippuvuus painoarvoon. FPA-menetelmää hyödyntävä henkilö arvioi itse sitä, millaisia painoarvoja hän käyttää ja tämä subjektiivisuus voi johtaa epätarkkoihin työmäärän arvioihin. (Symons, 1988)

Tämän tutkimuksen ja kohdeyrityksen tarpeisiin nähden FPA-menetelmää ei voida hyödyntää sellaisenaan vaan sitä tulee muokata sopimaan paremmin kohdeyrityksen toimintaympäristöön. FPA-menetelmästä voi olla mahdollista saada parempi työkalu yhdistämällä sitä muihin työmäärän arviointimenetelmiin.

3.1.4 Delphi-menetelmä

Delphi-menetelmä on strukturoitu kommunikaatiomenetelmä tai -tekniikka, joka on alun perin luotu systemaattiseksi, interaktiiviseksi, menetelmäksi hyödyntää asiantuntijoita ennustamisessa (Brown, 1968). Delphi-menetelmän avulla usealta asiantuntijalta kerätään anonyymisti arvio halutusta asiasta, jonka avulla voidaan hyödyntää jokaisen menetelmään osallistuvan mielipidettä ilman, että muiden ajatukset muokkaavat heidän vastauksiaan. Menetelmään voidaan hyödyntää moneen tarkoitukseen, kuten liiketoiminnan ennustamiseen tai ohjelmistoprojektien koon arvioimiseen. (Linstone & Turoff, 1975)

Lähtökohtana Delphi-menetelmälle on se, että halutaan vastaus johonkin tiettyyn ongelmaan, kuten ohjelmiston koon arviointiin. Delphi-menetelmää hyödyntäessä koordinaattori organisoii menetelmän eri vaiheet ja asiantuntijat tekevät asiat koordinaattorin ohjeiden mukaan. Delphi-menetelmä sisältää kuusi vaihetta:

1. Koordinaattori esittää jokaiselle asiantuntijalle ongelman yksityiskohdat ja arviointi lomakkeen
2. Koordinaattori kutsuu menetelmään osallistuvat asiantuntijat tapaamiseen, jossa ongelmasta keskustellaan asiantuntijoiden sekä koordinaattorin välillä
3. Asiantuntijat täyttävät lomakkeet anonyymisti
4. Koordinaattori valmistelee ja analysoi yhteenvedon arvioista
5. Koordinaattori kutsuu ryhmän tapaamiseen, jossa käydään läpi arviointien yhteenvedo ja keskustellaan suurimmista poikkeavuuksista
6. Asiantuntijat täyttävät lomakkeen uudestaan anonyymisti. Kohdat 4-6 toistetaan niin monesti kuin on tarpeen. (Boehm, 1981)

Menetelmän on siis tarkoituksena olla iteratiivinen ja lopputuloksena tulisi olla yhtenevä arvio käsitellystä ongelmasta. Esimerkiksi ohjelmistoprojektin kokoa arvioitaessa arviot voivat ensimmäiselle kierroksella vaihdella paljon keskenään. Ohjelmistoprojektin osalta asiantuntijat voivat arvioida esimerkiksi projektin laajuutta sekä työmäärää (Suri & Ranjan, 2012). Yhteenvedosta keskustellessa vastauksia käydään läpi ja syitä eroavaisuuksille etsitään. Tämän keskustelun jälkeen asiantuntijat ovat saaneet uusia ajatuksia oman arvion tueksi, jonka jälkeen oma arvio voi muuttua. Vastauksia ja yhteenveto keskusteluja pidetään niin monesti kuin on tarve. Useiden kierrosten jälkeen arviot alkavat lähenemään toisiaan, jonka jälkeen voidaan todeta arvion olevan halutulla tasolla.

Delphi-menetelmä soveltuu luonteensa takia moneen ongelmaan ja tarkoitukseen. Systemaattista mallia muodostaessa Delphi-menetelmästä voidaan hyödyntää sen ideologiaa kollektiivisesta arviosta. Asiantuntijoiden arviot voivat poiketa lähtötilanteessa paljonkin toisistaan, mutta keskustelujen kautta eri asiantuntijoiden mielipiteet tulevat esille ja perusteltuja näkökulmia arvioihin syntyy. Tämä johtaa tulokseen, johon kukaan yksittäinen asiantuntija ei olisi päässyt, mutta tiedon jakamisen kautta syntyvät arviot voivat tuoda paljon arvoa arviointi prosessiin.

3.1.5 Constructive Cost Model

Tarve arvioida ohjelmistoprojektien kokoa on kasvanut valtavasti 2000-luvun puolella ja niiden arvioinnista on tullut yhä haastavampaa. Yksi ensimmäisiä malleja, joita ohjelmistojen koon arvioinnissa käytettiin, on Constructive Cost Model (COCOMO). Tämä malli perustettiin alun perin 63 ohjelmistoprojektin pohjalta ja se on yksi eniten käytetyistä systemaattisista ohjelmiston koon arviointimalleista. (Sheta, 2006) COCOMO-malli kehitettiin vuonna 1981 sen ajan projekteista kertyneen datan pohjalta. Ohjelmistoala on kuitenkin kehittynyt valtavasti noista ajoista, joten voidaan olettaa, että käyttäessä mallia sellaisenaan, ei saada tarkkoja arvioita ohjelmistoprojektien koosta. (Chalotra et al. 2015)

COCOMO-mallin tarkoituksena on arvioida ohjelmistoprojektin työmäärää. Arvio tehdään hyödyntämällä projektin koodirivien määrää sekä matemaattisesti määritettyjä vakio kertoimia. Nämä kertoimet määräytyvät ohjelmistoprojektin kompleksisuuden mukaan, vaihtoehtoina ovat helppo, puolivaikea ja vaikea projekti. Näiden kertoimien avulla projektin kompleksisuus otetaan huomioon ja tämä yhdistettynä koodirivien määrään, saadaan arvio kokonaistyömäärästä. (Boehm, 1981)

COCOMO-mallin kaava on seuraava:

Työmäärä = $a(KLOC)^b$,

Tässä kaavassa a ja b ovat 63 projektin perusteella määritettyjä vakio arvoja, jotka vaihtuvat projektin kompleksisuuden mukaan. KLOC-arvo on koodi rivein määrä tuhansissa riveissä mitattuna.

COCOMO-mallia on käytetty pääosin arvioimaan ohjelmistoprojektin toteutukseen kuluva aikaa ja kustannuksia. Tämän lisäksi malli tarjoaa kuitenkin selityksiä mistä nämä arviot muodostuvat sekä, miten mallissa käytetyt kertoimet muodostuvat. Se tarjoaa mahdollisuuden analysoida saatua arviota ja sen avulla voidaan tarkastella myös projektin työmäärä arvioin herkkyyttä muuttamalla kaavassa käytettyjä arvoja hieman. (Boehm, 1981) COCOMO-mallin avulla saadaan luotua matemaattinen yhteys ohjelmiston kehityksen kuluvan ajan sekä ylläpitoon kuluvan vaivan välille (Sheta, 2006).

Alkuperäisen COCOMO-mallin oletuksena on, että ohjelmistoprojektin työmäärä voidaan arvioida hyödyntämällä vuonna 1981 kehitettyä kaavaa ja ohjelmiston arvioitua koodirivein määrää. Moderni ohjelmistoala on kuitenkin muuttunut paljon ja myöhemmin tehdyt jalostukset alkuperäisestä mallista ovat joustavampi ja näin ollen sopivat paremmin nykyajan tarpeisiin. (Suri & Ranjan, 2012) Alkuperäisen COCOMO-mallin pohjalta on jatkojalostettu useita eri malleja, kuten COCOMO II- sekä Fuzzy COCOMO -malli (Sheta, 2006). Myöhemmin muodostetuissa malleissa kertoimia on vain muokattu omien tarpeiden tai sen hetkisen datan pohjalta. Kertoimet riippuvat projektin luonteesta ja COCOMO-mallin jatkojalostaminen onnistuu muokkaamalla vakio kertoimia oman datan pohjalta. (Chalotra et al. 2015)

Suurimpana ongelmana COCOMO-mallissa on se, että ohjelmistoprojektin alussa on mahdotonta arvioida koodirivien määrää tarkasti ja projektin mennessä eteenpäin, vastaan tulee odottamattomia ongelmia, joiden takia COCOMO-mallin vakio arvot eivät toimi projektista toiseen. Malli sisältää paljon heikkouksia, joihin tulisi vastata, mutta se on niistä huolimatta ollut laajasti käytössä jo vuosia. Asiantuntijoiden arvioiden on kuitenkin huomattu pitävän paremmin paikkaansa ja tämän takia COCOMO-malli tai mikään muu malli ei ole jäänyt alalle käyttöön standardin omaisesti.

Kohdeyrityksen osalta mallia ei voida hyödyntää sellaisenaan, sillä ohjelmiston kokoa pitää arvioida kaavassa koodi riveinä. Tämä ei kohdeyrityksen toimintaympäristössä toimi, sillä projektin alussa toteutettavaa ohjelmistoa arvioidaan vain sen ominaisuuksien perusteella, ei koodi riveinä. COCOMO-mallia on kuitenkin tärkeä käsitellä tässä tutkimuksessa, sillä se on yksi merkittävimmistä systemaattisista työmäärän arviointi malleja, joita ohjelmistoalalla on käytetty.

3.2 Kannattavuuden arvioinnin systemaattinen malli

Edellisessä luvussa esiteltiin kuusi työmäärän arviointiin tai kannattavuuden arvioinnissa käytettävää menetelmää. Nämä kuusi menetelmää valittiin työhön, sillä kirjallisuuskatsauksen perusteella sekä alan asiantuntijoilta kuultuna ne ovat keskeisimpiä keinoja arvioida kannattavuutta. Osa menetelmistä oli ainoastaan kannattavuuden arvioinnin tai työmäärän arvioinnin osia, mutta ovat menetelmissä keskeisessä roolissa.

Tutkimuksessa muodostetussa mallissa hyödynnettiin tiettyjä periaatteita kuudesta esitellystä menetelmästä ja malli on pyritty muodostamaan niin, että se tukee mahdollisimman hyvin ketterää ohjelmistokehitystä. Kaikista esitellyistä menetelmistä ei otettu malliin ominaisuuksia. Mallissa hyödynnettiin WBS-menetelmän työn osittelu, FPA-menetelmässä esiintynyttä kompleksisuuden arviointia, PERT-menetelmästä tuttua optimistista ja pessimististä työmäärän arviota sekä niiden avulla laskettavaa matemaattista arviota sekä Delphi-menetelmän periaatetta asiantuntijoiden älykkyyksien yhdistämistä. Mallissa ei hyödynnetty mitään osaa COCOMO-mallista, jossa arvio muodostuvat ohjelmiston koodimäärään sekä matemaattisiin kaavoihin.

Yksi jokaisessa menetelmässä esiintynyt osa-alue oli työnosittelu. Työnosittaminen paikoiksi on yksinkertainen ja tehokas keino arvioida projekteja ja ohjelmistoalalla, sitä on käytetty jo pitkään. WBS-menetelmän avulla ohjelmisto voidaan jakaa toiminnallisuuksien mukaan osiin, jonka jälkeen työmäärän ja täten kannattavuuden arviointi on mahdollista tehdä. Tämän tutkimuksen keskeisenä tavoitteena on luoda malli kannattavuuden arviointiin ja WBS-menetelmän mukainen työnosittelu sisältyy muodostettavaan malliin.

Kannattavuudenarviointiprosessi vaiheet:

1. Työnosittelu
2. Toteutettavan työnosan sisällön kuvaaminen
3. Työmäärien arviointi roolipohjaisesti
4. Työnosan kompleksisuuden arvio
5. Malli toteuttaa työmääränarvioinnin hyödyntäen edellä mainittuja tietoja
6. Kannattavuudenarviointi työmäärän arvioon, haluttuun katteeseen sekä hinnoittelumenetelmään pohjautuen

Kannattavuudenarviointi prosessissa hyödynnetään vaiheeseen 4. saakka asiantuntijan osaamista, jonka jälkeen muodostettu malli tekee lopullisen työmääränarvioinnin. Mallin

arvioitua lopullisen työmääräarvion, asiantuntijaa tarvitaan prosessissa arvioimaan kannattavuutta halutun katteen, hinnoittelun sekä kustannusten mukaan. Asiantuntijan osaamisen lisäksi kannattavuuden arviointiprosessissa hyödynnetään seuraavia systemaattisia malleja:

1. WBS, käytetään osittelemaan koko projekti pieniin työosiin
2. FPA-menetelmä, työosien kompleksisuutta arvioidaan asteikolla 1-3
3. PERT-menetelmä, optimistisin ja pessimistin työmäärä arvio lasketaan kompleksisuuden avulla
4. Delphi-menetelmä, yhden työmäärä arvion sijaan kannattavuuden arviota tulisi tehdä iteratiivisena prosessina useaa asiantuntijaa hyödyntäen.

Mallissa on pyritty yhdistelemään eri systemaattisten menetelmien osia, jotka on koettu tuovan tukea asiantuntijalle. Kannattavuuden arviointi prosessin tärkeimpänä yksittäisenä tekijänä on kuitenkin arvioita tekevä asiantuntija, mutta hänen arviointikykyänsä tuetaan eri menetelmien avulla. Menetelmien hyödyntämisellä on tarkoituksena myös lisätä systemaattisuutta projektien välille.

Mallissa lähdetään tilanteesta, jossa tiedetään, minkälainen projekti on kyseessä, mutta työn tarkempaa osittelua ei olla vielä tehty. Mallin ensimmäisenä vaiheena on jakaa projektikokonaisuus pieniin toiminnallisiin osiin ohjelmiston toteutuksen mukaan. Tässä työn osittelussa tulee ottaa huomioon kaikki projektiin liittyen aikaa vievät asiat. Mallissa tulee siis ottaa huomioon ohjelmiston suunnittelu, toteutus sekä testaus. Myös projektin hallintaan sekä toimittajan ja asiakkaan väliseen kommunikaatioon on varattava aikaa jo työn ositteluvaiheessa. Projektikokonaisuus jaetaan pieniin osiin, tavoitteena saada niistä mahdollisimman tarkkoja pieniä aikaa vieviä kokonaisuuksia. Malliin tulee sisällyttää tieto siitä, missä sprintissä kyseinen työnosa toteutetaan, joka on tärkeä tieto käytettäessä ketterää ohjelmistokehittämistä.

Työnosittelun jälkeen työ voidaan jakaa eri työntekijöille eli työ ositellaan tarvittavien resurssien/roolien mukaan. Esimerkiksi tehtävä X voidaan arvioida niin, että siihen tarvitaan tekniseltä arkkitehdiltä yksi henkilötyöpäivä (htp), junior konsultilta kaksi htp ja senior konsultilta yksi htp. Näin ollen työn osalle arvioitu henkilötyöpäivämääräarvio on neljä päivää ja lopullisessa kannattavuuden arvioinnissa voidaan huomioida kuinka paljon kustannuksia työn osa aiheuttaa pohjautuen erihintaisiin rooleihin. Roolipohjainen työmäärän jaottelu on edellytyksenä hyvälle kustannusten seuraamiselle ja kustannusperusteiselle hinnoittelulle.

Tässä vaiheessa mallissa on hyödynnetty WBS-menetelmää, jonka avulla työ on ositeltu ja työn osille on tehty työmääräarvio henkilötyöpäivissä mitaten. Kaikki työn osat arvioidaan siis yksitellen, jotta arvio on mahdollista toteuttaa. Seuraavana vaiheena ohjelmistojen työmäärän arvioinnissa on määritellä tarkkuus, joilla arviot on tehty. Ohjelmistoprojekteissa tulee aina vastaan odottamattomia haasteita ja arviot venyvät. Tämä on tiedetty jo vuosikymmeniä, mutta järkevää ratkaisua ei olla keksitty. Edellisessä luvussa esiteltyjen mallien avulla tätä ongelmaa on yritetty ratkaista eri keinoin, mutta mikään menetelmä ei ole ollut ylivoimaisesti toista parempi. Parhaan arvion on tähän mennessä antanut asiantuntijat, jotka arvioivat työmäärää oman osaamisen pohjalta.

Asiantuntijoiden mukaan ottaminen malliin on siis välttämätöntä, sillä tutkimukset ovat osoittaneet, että asiantuntija-arviot ovat olleet kaikista tarkimpia. Kirjallisuuden pohjalta voidaan myös todeta, että asiantuntijoiden sekä systemaattisten mallien yhdistäminen voisi olla keino tarkempaan kannattavuuden arviointiin. Tästä syystä tutkimuksessa muodostettuun malliin sisältyy asiantuntijoiden arvioita sekä osia erilaisista systemaattisista malleista tai menetelmistä.

Työmääräarvion tarkkuus tulee arvioida jotenkin, jotta toimittaja voi laskea projektille kannattavuutta ja täten esittää tarjouksen asiakkaalle. Kirjallisuuden sekä asiantuntijoiden tietämyksen perusteella suurimmat tekijät, jotka vaikuttavat arvioinnin tarkkuuteen ovat riskit, työn osan koko sekä työn osittelussa syntyneen työn osan monimutkaisuus. Yksi keino ratkaista tätä arvion tarkkuuteen liittyvää ongelmaa on hyödyntää PERT-menetelmästä tuttua tapaa, jossa asiantuntija arvioi hänen mielestään todennäköisimmän, pessimistisimmän sekä optimistisimmän arvion työnosalle. Näiden kolmen luvun avulla voidaan laskea beta-jakaumaa käyttäen osuvin työmääräarvio työnosalle, joka pohjautuu asiantuntijan taitoihin sekä matemaattiseen laskentaan. Todennäköisimmän, optimistisimmän ja pessimistisimmän arvion avulla voidaan myös laskea tarkkuus matemaattisesti muodostetulle työmääräarviolle.

Hyödyntäen PERT-mallista tuttua menetelmää sekä asiantuntijan arviointikykyä, mallissa on hyödynnetty tässä vaiheessa sekä asiantuntijan arviota, että systemaattista mallia. Tällä keinolla voidaan parantaa asiantuntijan arviota, ja jopa tehdä arviolle herkkyysanalyysiä. Ongelmana tässä todennäköisimmän, pessimistisimmän ja optimistisimmän työmäärän arvioinnissa on siihen kuluva aika. Arvioita tekevät asiantuntijat tiedostavat sen, että arviot eivät pääosin toteudu vaan projektit venyvät. Tämän takia lisäajan kuluttaminen työn osan arviointiin ei välttämättä ole heidän mielestään kannattavaa. Aikaisemman yhden työmäärä arvion sijaan (todennäköisin työmääräarvio) he joutuisivat arvioimaan saman työnosan kolmella tavalla (todennäköisin-, optimistisin-, pessimistisin arvio).

Työmäärän arviointiin ei saisi kulua liikaa aikaa eikä malli saisi tulla liian monimutkaiseksi käyttää. Yksi keino tähän ongelmaan voisi olla se, että asiantuntijan ei tarvitsisi arvioida kaikkia kolmea skenaariota vaan hänen tulisi arvioida työn osalle työmääräarvio sekä arvioida työn osan kompleksisuus. Kompleksisuuden arviointi on skenaarioita helpompaa ja kompleksisuuden arvion avulla on mahdollista muodostaa skenaariot matemaattisesti. FPA-menetelmän tapaan työn osa voidaan arvioida sen kompleksisuuden mukaan joko tasolle matala (1), keskiverto (2) ja korkea (3). Mallin käyttäjän tulisi arvioida työn osan kompleksisuutta kolmen keskeisimmän muuttujan kautta: riskien, työmäärän arvion koon sekä toteutettavan toiminnallisuuden monimutkaisuuden osalta.

Kannattavuuden arviointia tekevän asiantuntijan tulisi siis ositella projekti pieniin työnosiin, resursoida työn osat, arvioida kokonaistyömäärä sekä arvioida kuinka kompleksinen työn osa on asteikolla matala, keskiverto ja korkea. Tämän jälkeen asiantuntijan arviota tulisi tukea systemaattisin keinoin. Kompleksisuuden arvioinnin jälkeen malli laskee siis optimistisimman ja pessimistisimmän arvion työn osalle. Kun malli on laskenut nämä arvot, se voi laskea lopullisen työmäärä arvion beta-jakauman avulla.

Asiantuntijan arvioidessa työn kompleksisuutta riskien, työmäärän arvion koon sekä toteutuksen monimutkaisuuden perusteella, voidaan siis kompleksisuuden tason mukaan muodostaa matemaattisesti optimistisin sekä pessimistisin arvio. Esimerkiksi asiantuntija arvioi tehtävä X:n olevan neljän henkilötyöpäivän työ ja hän arvio sen olevan keskiverto kompleksinen. Näiden tietojen avulla malli muodostaa systemaattisia kertoimia käyttäen pessimistisimmän sekä optimistisimman arvion. Jokaiselle kompleksisuuden tasolle on omat kertoimet, joilla nämä arviot muodostetaan.

Esimerkiksi kompleksisuuden ollessa keskiverto optimiselle arviolle voidaan käyttää 0.8 kerrointa ja pessimistiselle 2.0 kerrointa, jolloin esimerkki tapauksessa (neljän htp:n arviossa) optimistisin arvio työnosan toteutukselle olisi 3.2 htp (0.8×4 htp), todennäköisin olisi 4 htp ja pessimistisin olisi 8 htp (2.0×4 htp). Näiden lukujen avulla malli arvioi lopullisen työmäärä arvion ja tarvittaessa tekee myös herkkyysanalyysin. Malli siis hyödyntää asiantuntijan kokemusta ja osaamista, jonka avulla hän arvioi työmäärän sekä kompleksisuuden, ja tämän jälkeen mallissa hyödynnetään systemaattista lähestymistapaa laskemalla matemaattisesti optimistisimman ja pessimistisimmän arvion sekä laskemalla näiden avulla todellisen työmäärän arvion.

Asiantuntija tekee jokaiselle työnosalle samat arviot ja jaottelut, jonka jälkeen malli laskee projektille kokonaiskustannuksen hyödyntäen mallin tietoa työmääristä sekä tarvi-

tuista resursseista. Näiden tietojen avulla voidaan laskea lopullinen hinta projektin toimittamiselle sekä projektin kannattavuus. Liitteessä A on esitelty malli sekä siinä käytetyt tarkat luvut ja kertoimet.

Delphi-menetelmään pohjautuen kannattavuuden arvioinnin tarkkuutta voidaan lisätä ottamalla arviointiprosessiin mukaan useampia asiantuntijoita. Delphi-menetelmän peruseriaatteena on, että useampi asiantuntija tekee arvion esimerkiksi työmäärästä, jonka jälkeen näistä arviosta keskustellaan ja keskustelun jälkeen asiantuntijat tekevät uuden arvion. Uuden arvion jälkeen tuloksista keskustellaan ja tämä sykli toistetaan niin useasti kuin on tarpeen, jotta asiantuntijat pääsevät yhteisymmärrykseen arvioitavasta asiasta.

Tämän työn osalta kannattavuuden arviointiprosessiin voisi ottaa usean asiantuntijan mukaan Delphi-menetelmän tavoin. Asiantuntijat tekisivät itsenäisesti projektin alussa työmäärän arviot hyödyntäen työssä muodostettua mallia. Heidän arvionsa todennäköisesti poikkeavat toisistaan ja näistä poikkeamista he voivat keskustella palaverin muodossa. Kun mallia on ensimmäisen kerran käytetty ja ensimmäinen tulosten purkamis-palaveri on käyty, he voivat päättää suorittavatko uusia kierroksia vai pääsivätkö he yhteisymmärrykseen työmäärän arviosta. Tällaisen kannattavuuden arvioinnin prosessin avulla voidaan hyödyntää useampaa asiantuntijaa sekä useampaa systemaattista mallia. Tutkimuksen hypoteesina on, että tätä suunniteltua prosessia hyödyntäen työmäärän arvioiden tarkkuus paranee ja projektien kannattavuuksien arviointi on jatkossa paremmalla tasolla.

Kannattavuuden arviointiin kehiteltävässä prosessissa ja mallissa lähdetään siis liikkeelle projektin kokonaisuuden jaottelusta osiin WBS-menetelmää hyödyntäen. Tämän jälkeen asiantuntija arvio minkä tyyppisiä työntekijöitä työnosiin tarvitaan ja arvio jokaiselle työnosalle kokonaistyömäärän sekä sen kompleksisuuden. Näiden asiantuntijan syöttämien tietojen avulla malli laskee pessimistisimmän sekä optimistisimman työmäärä arvion kullekin työnosalle hyödyntäen kompleksisuuden tasoa, ja laskee niiden avulla matemaattisesti tarkemman työmääräarvion. Työmäärien arviontien avulla sekä tarvittujen resurssien pohjalta malli laskee lopulliset kustannukset kokonaisprojektille ja näin ollen voidaan arvioida projektin kannattavuutta.

Lopullinen kannattavuus muodostuu kustannuksista, halutusta katteesta sekä käytetystä hinnoittelumenetelmästä. Tässä tutkimuksessa ehdotetaan, että hinnoittelussa käytetään kustannusperusteista hinnoittelua, sillä sen avulla hinnan perustelu asiakkaalle on yksinkertaista ja kannattavuuden seuraaminen on helppo, sillä projektien toteutumisen jälkeen tulee seurata vain menoja ja tuloja. Hypoteesina on, että tällä kannattavuuden arvioinnin prosessilla systemaattisuus projektien arvioinnista kasvaa projektien välillä.

4. TUTKIMUKSEN TOTEUTUS

4.1 Tutkimuksen toteutus

Tämän työn tarkoituksena on selvittää keinoja, kuinka kannattavuuden arviointia voidaan parantaa ketterissä ohjelmistoprojekteissa. Ohjelmistoprojektien aikataulujen venyminen ja kannattavuuden arviointi projektien alkuvaiheessa on ollut suuri haaste vuosikymmeniä. Arviointitapoja on ollut useita, asiantuntija-arvioista systemaattisiin malleihin. Ongelmaa ei olla kuitenkaan pystytty ratkaisemaan tarpeeksi hyvin. Osaltaan tämän ongelmanratkaisun uupumiseen vaikuttaa myös se, että ohjelmistoalan toimijat eivät ole tarpeeksi motivoituneita löytämään ratkaisua tähän. Asiakkaat kilpailuttavat toteutettavia ohjelmistoprojekteja ja usein se toimittaja voittaa kilpailun, joka tarjoaa alhaisinta työmääräarviota. Todellisuudessa nämä halvimmat projektit venyvät aikataulun ja kustannusten suhteen. Asiakas ei ole tähän tyytyväinen ja toimittajan osalta kannattavuuden arviointi projektin alussa oli turhaa, sillä se ei myöhästyneissä projekteissa pidä paikkaansa.

Tämän työn lopputuloksena on tarkoitus muodostaa toimiva työkalu kannattavuuden arviointiin ketterissä ohjelmistoprojekteissa. Tutkimukset ja kirjallisuus on osoittanut, että asiantuntijoiden arviot ovat tällä hetkellä paras keino arvioida projektien työmääriä, mutta nekin eivät ole vaaditulla tasolla tarkkuuden suhteen. Systemaattiset mallit ovat vaikeita toteuttaa ja ohjelmistoalan kehittyessä nämä mallit vanhenevat nopeasti. Teoria osoittaa, että paras kannattavuuden arviointi menetelmä voisi syntyä yhdistelemällä asiantuntijoiden tietämystä, poistaen kuitenkin menetelmän subjektiivisuutta systemaattisten mallien keinoin.

Teorian pohjalta muodostettiin malli, joka on yhdistelmä asiantuntijoiden hyödyntämistä sekä systemaattisia menetelmiä. Tutkimuksen tarkoituksena on testata, kuinka hyvin ensimmäinen versio muodostetusta mallista toimii ja mitä kehitys ideoita haastateltavilta asiantuntijoilta syntyy. Tutkimuksen tarkoituksena on siis tutkia kannattavuuden arviointi mallin toimivuutta hyödyntäen kohdeyrityksen dataa vanhoista ohjelmistoprojekteista sekä suorittamalla haastatteluja kannattavuuden arvioita tekevien ihmisten kanssa.

Datan avulla mallin toimivuutta voidaan testata syöttämällä dataa uuteen malliin ja tätä kautta nähdä millaisen arvion uusi malli antaa verrattuna vanhaan menetelmään. Vanhan ja uuden menetelmän lukuja voidaan verrata ja, koska projektit ovat jo toteutuneet voidaan myös verrata sitä, kuinka hyvin tämän työn pohjalta toteutettu malli olisi toiminut verrattuna toteutuneeseen työmäärään. Datan hyödyntämisen avulla nähdään, toimiiko

uusi malli vanhaa menetelmää paremmin työmäärän sekä kannattavuuden arvioinnissa, ja onko uuden mallin arvioinnin tarkkuus, kuinka lähellä projektien toteutuneita työmääriä sekä kannattavuuksia. Datan pohjalta tehtyjen arviointien avulla voidaan kvantitatiivisesti osoittaa eroavaisuuksia uuden mallin, vanhan mallin sekä toteutuneiden arvioiden välille.

Haastattelujen avulla puolestaan esitellään arvioita tekeville asiantuntijoille mallia ja haastatellaan heitä uuden mallin toimivuudesta. Näistä haastatteluista saadaan tietoa asioista, joita teorian kautta ei olla tajuttu ottaa huomioon. Asiantuntijoilla on paljon tietoa pohjautuen heidän kokemuksiinsa ja aikaisempiin projekteihin. Tätä haastattelusta saatua kvalitatiivista tietoa voidaan hyödyntää mallin arvioinnissa ja sen muokkaamisessa. Yhdistelemällä mallin testausta datan avulla ja haastatteleamalla asiantuntijoita, saadaan kattavasti sekä kvantitatiivista, että kvalitatiivista tietoa.

4.1.1 Yrityksen esittely

Tämä tutkimus toteutetaan yhteistyössä Digia Finland Oy -yrityksen kanssa. Digia on suomalainen ohjelmisto- ja palveluyritys, joka toteuttaa asiakkaille erilaisia alustaratkaisuja ja digitaalisia palveluja asiakkaiden tarpeisiin. Digia:n liikevaihto vuodelta 2018 oli noin 112,1 miljoonaa euroa, josta 5,8% eli 6,5 miljoonaa euroa oli liikevoittoa. Digialla on pitkä historia ja yhtiön rakenne on muuttunut paljon vuosien saatossa. (Digia, 2019)

Yritys aloitti liiketoimintaansa vuonna 1990 ja on sen jälkeen kasvanut paljon ja yhtiö on laajentunut sekä orgaanisesti, että epäorgaanisesti yrityskauppojen myötä. Digia:lla on asiakkaita monista eri toimialoista, mutta keskeisimpiä asiakkaita on finanssipalvelut, julkishallinto, kaupallinen ala, teollisuuden ala sekä energia-ala. Digia tekee yhteistyötä alasta riippumatta kaikkien halukkaiden yritysten kanssa ja yhteistyö on mahdollinen, sillä kaikilla aloilla tarvitaan nykyään digitaalisia palveluita. (Digia, 2019) Projekteissaan käytössä on sekä ketteriä menetelmiä, että vesiputous malleja, joista tässä työssä tarkastellaan ainoastaan ketteriä menetelmiä.

4.1.2 Tutkimusotanta

Tutkimusotanta on tähän tutkimukseen muodostettu osittain harkintaan pohjautuvalla otantamenetelmällä ja osittain hyödyntäen lumipallo-otantamenetelmää. Harkintaan pohjautuvaa menetelmää hyödynnettiin tutkimuksen alussa, kun luotiin tietyt kriteerit, joihin haastateltavan tulee vastata. Tämän menetelmän avulla on mahdollista saada parhaiten soveltuvat haastateltavat tutkimukseen mukaan. Tämän tutkimuksen osalta haastateltavat valittiin niin, että he vastaavat mahdollisimman hyvin niin sanottua tyyppillistä tapausa. (Saunders et al. 2012, ss. 237–240)

Tyypillisellä tapauksella tarkoitetaan tämän tutkimuksen osalta ihmistä, joka tekee kannattavuuden arviointia tai työmäärän arviointia projekteissa, jotka sisältyvät tutkimuksen rajoitteisiin. Näiltä harkintaan pohjautuvan otantamenetelmän avulla valituilta haastateltavilta tiedusteltiin henkilöitä, joita heidän mielestään olisi tämän tutkimuksen kannalta oleellista haastatella. Tällaista lähestymistapaa sanotaan lumipallo-otannaksi. Lumipallo-otannassa haastateltavan henkilön avulla saadaan seuraava haastateltava ja tätä kautta muodostuu tutkimuksen otanta joukko. (Saunders et al. 2012, ss. 237–240)

Tutkimusotantaan sisältyi ihmisiä, jotka ovat vastuussa projektien kannattavuuden tai työmäärien arvioinnista. Haastateltavat henkilöt toimivat kohdeyrityksessä projektipäällikköinä, teknisinä arkkitehteinä, tiimipäällikköinä, asiantuntijoina, myyjinä tai muissa johdon asemissa. Haastatteluun otettiin jokaisesta haastatellusta työryhmästä 2-3 henkilöä. Haastateltavia henkilöitä oli yhteensä 7. Osa haastateltavista tekee paljon työmäärän sekä kannattavuuksien arviointeja, osa puolestaan ovat enemmän niiden prosessien kehittämässä tai ovat vastuussa työmäärän arvioinnin hyväksymisessä.

Kohdeyrityksessä useat henkilöt ovat tekemässä näitä arvioita, sillä ne ovat keskeisessä roolissa projektien myynnin kannalta. Kannattavuuksien ja työmäärien arvioinnit ovat tärkeitä asioita projektien alussa ja useat eri työtehtäviä tekevät ovat siinä mukana. Haastatteluun on tärkeää saada laajalti eri työtehtävistä henkilöitä, sillä heillä on työtehtäviensä mukaan hieman erilainen näkemys kannattavuuden sekä työmäärien arvioinnista. Alla olevassa taulukossa 1. on esitelty haastateltavat henkilöt.

Haastateltava	Koodi	Kokemus alalta	Kokemus kannattavuuden arvioinnista	Työnimike
Haastateltava 1	H1	8	3	Projektipäällikkö/Ratkaisu konsultti
Haastateltava 2	H2	12	7	Myynnin ratkaisuarkkitehti/tiimin vetäjä
Haastateltava 3	H3	7	4	Tekninen arkkitehti/kehittäjä
Haastateltava 4	H4	7	6	Myynnin ratkaisuarkkitehti/tiimin vetäjä
Haastateltava 5	H5	8	7	Projektipäällikkö/Ratkaisu konsultti
Haastateltava 6	H6	9	7	Projektipäällikkö/Ratkaisu konsultti
Haastateltava 7	H7	10	6	Tekninen arkkitehti/kehittäjä

Taulukko 1. *Haastateltavat henkilöt*

Taulukossa 1. haastateltavat henkilöt ovat anonymisoitu nimeämällä heidät Haastateltava-nimellä. Taulukossa on tämän lisäksi esitelty koodi jokaiselle haastateltavalle, jota

käytetään tulosten avaamisessa. Taulukossa on esitelty myös haastateltavien kokemus alalta, kokemus kannattavuuden arvioinnista sekä heidän työnimikkeensä. Haastateltavien kokemus alalta oli vähintään 7 vuotta ja kokenein henkilö oli työskennellyt alalla jo 12 vuotta. Keskiarvoltaan heidän työkokemuksensa alalta oli 8,7 vuotta. Haastateltavien henkilöiden kokemus kannattavuuden arvioinnista oli 3-7 vuotta. Keskiarvoltaan kokemus kannattavuuden arvioinnista oli 5,7 vuotta. Haastatteluun osallistuneet henkilöt ovat siis kokeneita henkilöitä alallaan.

Työnimike-sarakkeessa on esitelty haastateltavien henkilöiden työrooli. Jokaisesta roolista haastatteluun otettiin vähintään kaksi henkilöä. Projekti päällikkö/Ratkaisu konsultti-roolista haastatteluun osallistui kolme henkilöä, Myynnin ratkaisuarkkitehti/tiimin vetäjä-roolista sekä Tekninen arkkitehti/kehittäjä-roolista haastateltiin kahta henkilöä kustakin. Haastattelutulosten kannalta on tärkeää huomata, että roolista huolimatta haastateltavat tekevät erityyppisiä tehtäviä ja heidän näkökulmansa poikkeaa toisistaan merkittävästi.

4.1.3 Haastattelut

Tutkimus toteutettiin kahdessa vaiheessa, haastattelujen avulla sekä testaamalla mallia kokeellisesti. Haastattelujen tavoitteena oli saada tietoa mallin toimivuudesta sekä huomioita sen hyvistä tai huonoista puolista. Haastattelussa selvitettiin myös tämän hetkisen arviointi prosessin toimivuutta. Haastateltavat henkilöt olivat kohdeyrityksessä eri tasoilla ja eri työtehtävissä työskenteleviä henkilöitä, jotka ovat tehneet paljon kannattavuuden sekä työmäärän arviointeja. Haastateltaviin henkilöihin otettiin yhteyttä sähköpostitse tai puhelimitse. Haastateltavilta tiedusteltiin myös heidän näkökulmaansa siitä voisiko tutkimuksessa toteutettu malli ja menettelytapa toimia heidän mielestään. Heiltä selvitettiin mitkä asiat tällä hetkellä kannattavuuden arvioinnissa toimivat ja asiat eivät toimi.

Haastattelu toteutettiin semi-strukturoituna yksilöhaastatteluna. Haastattelut nauhoitettiin tulosten keräämistä varten ja kaikilta haastateltavilta pyydettiin suostumus nauhoitukseen. Haastattelu sisälsi kysymyksiä nykyisestä kannattavuuden arvioinnista sekä mallin toimivuuteen ja kehittämiseen liittyvistä kysymyksistä. Nykyiseen kannattavuuden arviointi prosessiin liittyvät kysymykset lähetettiin etukäteen haastateltaville, jotta he pystyivät varautumaan niihin etukäteen ja täten haastattelun tuloksiin saatiin enemmän pohdittuja vastauksia nykyisiin arviointi prosesseihin liittyen. Haastattelu kysymykset löytyvät kuvasta 11.

Haastateltavan työkokemus:

1. *Minkälainen on työtaustasi?*
 - a. *Monta vuotta olet työskennellyt alalla?*
2. *Mikä on tämän hetkinen työtehtäväsi?*
3. *Kuinka paljon sinulla on kokemusta kannattavuuden/työmäärien arvioinnista?*

Tämän hetkinen kannattavuuden arviointi prosessi:

4. *Kuinka hyvin kannattavuuden/työmäärän arviointi toimii tällä hetkellä?*
 - a. *Miksi?*
5. *Mitkä asiat ovat hyvin nykyisessä menetelmässä?*
 - a. *Miksi?*
6. *Mitkä asiat ovat huonosti nykyisessä menetelmässä?*
 - a. *Miksi?*
 - b. *Miten kehittäisit niitä?*

Mallin toimivuus:

7. *Mitkä ovat ensivaikutelmat uudesta mallista?*
8. *Mitä parannuksia näet siinä verrattuna vanhaan malliin?*
9. *Mikä mallissa ei todennäköisesti toimi?*
 - a. *Miksi?*

Loppukysymykset:

10. *Ottaisitko uuden mallin käyttöön kannattavuuden arvioinnin työkaluksi?*
 - a. *Miksi?*
 - b. *Miksi ei?*

Kuva 8. Haastattelukysymykset

Kuvasta 11. nähdään, että, ensimmäiset, haastattelun lämmittelykysymykset ovat tarkoitettu selvittämään haastateltavan henkilön työtaustaa ja kokemusta, jotta haastateltavia henkilöitä voidaan jaotella tulosten analysointivaiheessa. Tämän jälkeen haastateltavat henkilöt vastasivat heille etukäteen lähetettyihin kysymyksiin, jotka liittyivät nykyiseen kannattavuuden arviointi prosessiin. Lopuksi haastateltaville henkilöille esiteltiin tutkimuksessa kehitetty malli ja selitettiin, miten ehdotettu kannattavuuden arviointi prosessi toimii. Mallin esittelyn jälkeen heiltä kysyttiin kysymyksiä liittyen tutkimuksessa toteutettuun malliin. Viimeisenä heiltä kysyttiin haastattelun päättävänä kysymyksenä pitääkö he mahdollisena, että ottaisivat mallin käyttöön kannattavuuden arviointi prosessissa.

4.1.4 Mallin testaaminen kokeellisesti

Tutkimuksen merkittävimpana osuutena voidaan pitää kannattavuuden arviointiin kehitetyn mallin muodostamista. Malli luotiin pohjautuen ohjelmistoalan arviointimenetelmiin. Tutkimusta lähdettiin toteuttamaan, sillä kohdeyrityksessä tiedostettiin puutteita kannattavuuden arvioinnissa, erityisesti kannattavuuden arvioinnin systemaattisuudessa. Teorian pohjalta luotiin alustava malli, jota testataan, haastattelujen lisäksi, kohdeyrityksen menneiden sekä tämän hetkisten projektien osalta.

Mallia testattiin kokeellisesti neljän projektin osalta. Mallin testaamiseen tarvittavaa dataa on tieto projektin työnosista sekä työmääräarvio kullekin työnosalle. Tämä data on helposti saatavilla kohdeyrityksen tietovarastosta. Tutkimukseen otettiin käynnissä olevia projekteja, joiden työmäärä on jo arvioitu sekä jo valmistuneita projekteja, jossa työmäärä on arvioitu ja projektin toteutunut työmäärä on selvillä.

Projektien dataa ei voida sellaisenaan kuitenkaan hyödyntää mallin testaamiseen. Mallissa tulee arvioida työnosien ja niiden työmäärien lisäksi työn kompleksisuutta, mitä ei ole kohdeyrityksen projekteissa aiemmin arvioitu. Tämä ongelmakohta ratkaistiin niin, että alkuperäisen työmäärän tehneeseen henkilöön otettiin yhteyttä ja hän teki arvion kompleksisuudesta jokaiselle työn osalle.

Mallia testattiin kokeellisesti siis hyödyntämällä kohdeyrityksen dataa työnosien ja työmäärän suhteen. Tämän jälkeen mallissa täytyi arvioida kompleksisuus. Alkuperäisen työmäärä arvion tehnyt henkilö otettiin mallin testaukseen mukaan ja hän teki arvion työnosan kompleksisuudesta projekteissa, jotka hän oli itse arvioinut. Datan keräämisen ja yhdistämisen jälkeen mallia käytettiin työmäärän/kannattavuuden arvioinnissa ja tätä arvioita verrattiin tehtyihin tai toteutuneisiin arvioihin. Testin avulla nähdään, kuinka paljon mallin avulla arvioitu työmäärä eroaa asiantuntijan arviosta ja mitkä ovat arvioiden eroavaisuudet. Hypoteesina mallin testaamiselle oli se, että projektien välinen systemaattisuus kasvaa, joka mahdollistaa jatkossa yhtenäisempiä arvioita.

Käytetty data oli erityyppisistä projekteista, jotka kukin osuivat tutkimuksen rajauksiin. Kompleksisuuden arviointiin otettiin jokaisen projektin osalta vähintään kaksi henkilöä, jotta subjektiivisuutta sen arvioinnista saatiin pienemmäksi. Mallin testaamisen avulla selvitettiin, kuinka tarkkoja arvioita malli antaa ja kuinka sen antamat arviot poikkeavat kohdeyrityksen aikaisemmista arvioista. Kannattavuuden arviointiin ehdotettua asiantuntijoiden kollektiivisen älykkyyden hyödyntämistä ei testattu tässä vaiheessa.

4.2 Tutkimusmetodologia

Tutkimuksen tarkoituksena on luoda uutta tietämystä eri keinoin ja tutkimusta tulee lähestyä tutkimusfilosofian kautta. Tutkimusmetodologiaa tulee pohtia tutkimusfilosofian, tutkimuksen lähestymistavan, tutkimusstrategian sekä tutkimus valintojen kautta. (Saunders et al. 2012) Tutkimusmetodologian suhteen tutkimusta on mahdollista suorittaa lukuisilla eri tavoilla.

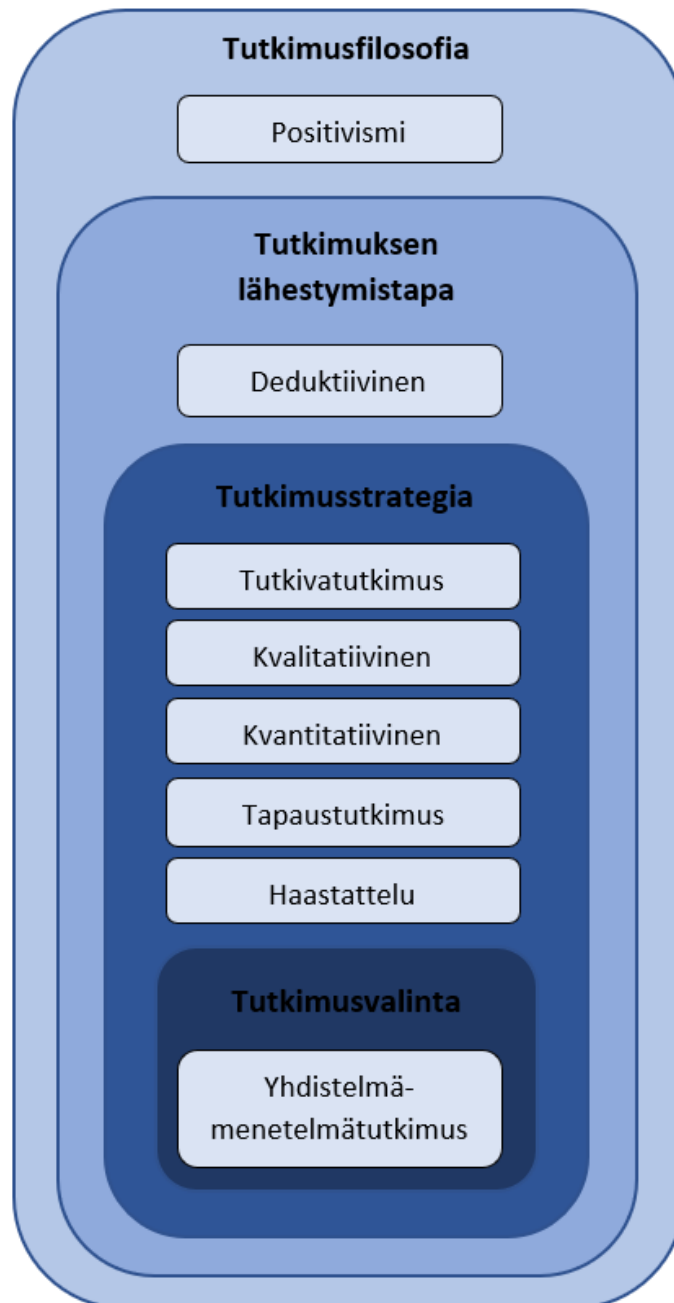
Tutkimusfilosofia kuvailee tietämyksen luonnetta, sen syntymistä ja kehittymistä (Saunders et al. 2012). Tässä tutkimuksessa tutkimusfilosofiaksi on valittu positivismi. Positivismi sopii tutkimuksen luonteeseen hyvin, sillä sen tarkoituksena on hankkia faktatietoa, luoda teoreettinen malli ja testata mallia empiirisen aineiston kautta (Saunders et al. 2012). Positivistinen filosofia uskoo siihen, että uutta faktatietoa on hankittavassa ja uusia, merkittäviä, tutkimustuloksia on mahdollista luoda.

Tutkimuksen lähestymistavat ovat keinoja, joiden avulla tutkimuksessa päädytään tutkimuskysymyksistä lopputuloksiin. Tutkimuksen lähestymistavat eivät ole yhtä abstraktilla tasolla kuin tutkimusfilosofiat, mutta nekin ovat vielä abstraktilla tasolla. (Saunders et al. 2012) Tämän tutkimuksen lähestymistavaksi on valittu deduktiivinen lähestymistapa, sillä deduktiivinen lähestymistapa lähtee liikkeelle teoriasta ja sen tavoitteena on teorian testaaminen ja hypoteesien muodostaminen. Deduktiivisessa lähestymistavassa on oltava riittävän suuret otoskoot ja käytössä tulee olla tieteelliset protokollat (Saunders et al. 2012).

Tutkimusstrategiset valinnat kertovat sen, minkälaista dataa tarvitaan sekä minkälaista dataa kerätään ja miten sitä kerätään (Saunders et al. 2012). Tämän tutkimuksen on tarkoituksena olla tutkivatutkimus, sillä työn tavoitteena on tiedon lisääminen sekä asioiden ymmärtäminen. Datan osalta tutkimuksessa tarvitaan sekä kvalitatiivista, että kvantitatiivista tietoa. Konstruktiota arvioidaan laadullisesti sekä määrällisesti, jotta sen voidaan todeta olevan käyttökelpoinen oikeassa yritysmaailmassa. Dataa kerätään haastattelujen avulla ja konstruktiota testataan myös case tutkimus -tyyppisesti, olemassa olevalla datalla.

Tutkimusvalinnat kertovat sen, miten tutkimusstrategioita hyödynnetään. Tässä tutkimuksessa parhaaksi vaihtoehdoksi osoittautui yhdistelmämenetelmätutkimus, jonka tarkoituksena on saada eri tiedonkeruu menetelmien avulla erilaista tietoa. (Saunders et al. 2012) Tämän tutkimuksen osalta haastattelujen avulla saadaan kvalitatiivista dataa ja case tutkimuksen avulla saadaan testattua mallia kvalitatiivisesti. Ajallisen fokuksen osalta työ suoritetaan poikittaistutkimuksena, mutta konstruktion testaamisessa käytetään myös historiallista dataa.

Kuvassa 2. esiteltynä tutkimusmetodologiset valinnat. Kuvassa tutkimusmetodologia on luotu kerroksittain, jonka laajimpana/alimpana kerroksena on tutkimusfilosofia ja suppeimpana/ylimpänä kerroksena on tutkimusvalinta. Kuvaa voidaan tulkita niin, että tiettyyn tutkimusmetodologian kerrokseen vaikuttaa kaikki sen kerroksen alapuolella olevat kerrokset.



Kuva 9. *Tutkimusmetodologiset valinnat*

Kuvasta 2. on jätetty pois tutkimuksen aikaan liittyvät seikat sekä tutkimuksen toteuttamiseen liittyvät tekniikat ja proseduurit, mitkä kuuluvat osaltaan tutkimusmetodologisiin valintoihin. Nämä kohdat ovat jätetty kuvasta pois, sillä kuvan on tarkoituksena olla laajempi kuvaus valituista tutkimusmetodologisista asioista.

4.3 Sekundääriaineisto

Tämän tutkimuksen tekemiseen käytettiin dataa, joka syntyi haastatteluista ja sen lisäksi tutkimuksessa hyödynnettiin kohdeyritykseltä saatua dataa. Kohdeyritykseltä hyödynnetty data liittyi mallin testaamiseen. Jotta mallia voitiin testata, täytyi hyödyntää dataa oikeista projekteista. Ilman tätä sekundääriaineistoa mallin testaaminen olisi jäänyt vajaan ja luotettavia tuloksia ei olisi saatu. Kohdeyritykseltä saadun sekundääriaineiston avulla voitiin testata, kuinka relevantti muodostettu malli olisi todellisessa liiketoimintaympäristössä.

Sekundääriaineistona hyödynnettiin pääosin dataa kohdeyrityksen vanhoista projekteista. Näitä vanhoihin projekteihin liittyviä dokumentteja olivat määrittelydokumentit, ja projekteihin liittyvät muistio- sekä suunnitteludokumentit. Tutkimuksen onnistumisen osalta määrittelydokumentit olivat tärkeimmässä roolissa, sillä niiden avulla saatiin tietoa projektien työmääräarvioista. Suunnitelma- ja muistiodokumenttien avulla pystyttiin määrittämään tarkemmin tietoja, joita mallin testaamiseen tarvittiin.

5. EMPIIRISET TULOKSET

5.1 Haastattelun tulokset

Haastattelut sujuivat odotetusti ja kaikki haastatteluun osallistuvat henkilöt jakoivat mielellään tietämystään. Haastattelu toteutettiin suunnitelmien mukaan eli haastatteluun saapuneet henkilöt esittelivät hieman itseään ja tämän jälkeen he kertoivat näkemyksiään nykyisestä kannattavuuden arviointi prosessista. Lopuksi heille esiteltiin tutkimuksen alustava kannattavuuden arviointimalli/prosessi ja heitä haastateltiin sen toimivuudesta. Haastattelu aloitettiin siis helpoilla kysymyksillä haastateltavien työtaustasta ja kokemuksista. Haastateltavilla oli heterogeeninen työtausta ja heidän nykyiset työtehtävänsä poikkesivat paljon toisistaan. Tämä mahdollisti sen, että jokainen haastattelu antoi paljon uusia näkemyksiä aiheesta ja jokainen heistä osasia katsoa aihetta omasta näkökulmastaan.

Ennen tutkimuksessa muodostetun mallin esittelyä, haastateltavilta selvitettiin miten he tällä hetkellä tekevät kannattavuuden ja työmäärän arviointia sekä mitkä ovat sen prosessin hyvät ja huonot puolet. Tutkimuksen kannalta oli tärkeää selvittää selviä kehityskohteita, joita asiantuntijat ovat havainneet sekä prosessissa toimivia osia, jotka he haluaisivat jatkossakin pitää prosessissa.

Kohdeyritys hyödyntää vahvasti asiantuntijoiden osaamista kannattavuuden arvioinnissa, mutta prosessissa on silti säännönmukaisuuksia. Jokainen haastatteluun osallistunut henkilö kertoi tämän hetkisen prosessin toimivan hyvin tai kohtaisesti. Haastateltavien mielestä kohdeyrityksessä oli tapahtunut suurta kehitystä kannattavuuden arvioinnissa viime vuosina. Tiivistettynä kannattavuuden arviointiprosessi etenee niin, että asiakkaalta tulee aluksi tarjouspyyntö halutusta tietojärjestelmästä, tämän jälkeen projekti ositellaan pieniin työnosiin ja asiantuntija arvioi työnosien työmäärät. Lopuksi, kun työ on ositeltu ja työmäärä on arvioutu, voidaan laskea mihin hintaan projekti tulee myydä, jotta se on kannattavaa.

H1, H2, H6 ja H7 mainitsivat tämän hetkisen joustavuuden toimivaksi piirteeksi kannattavuuden arvioinnissa. Nykyinen prosessi ei siis tukeudu mihinkään systemaattiseen malliin, vaan asiantuntijat tekevät työmäärien ja kannattavuuden arvioinnit. Haastateltavat kertoivat, että kukin tarvitsee oman tietämyksen lisäksi muita asiantuntijoita arvioinnin tekemisessä. Ainoastaan H5 mainitsi, että hän pyrkii kutsumaan asiantuntijat yhteen

tekemään arvioita ja hänen mielestään on ongelmallista, että arviointiprosessissa turvaututaan liikaa yhteen asiantuntijaan. Muut haastateltavat hyödynsivät muiden apua ainoastaan kysymällä arvioita tietyistä työnosista. Nykyiseen prosessiin ei siis kuulu arvioiden tekeminen ryhmissä, vaan ne tehdään pääosin yksin.

Kaikki muut haastateltavat paitsi H6 ja H7 mainitsivat, että kannattavuuden arviointiin tulisi panostaa enemmän aikaa ja he ovat huomanneet ajankäytön hyödyt työmääräarvioiden pitävyyden osalta. Yleisesti ottaen hyvinä ominaisuuksina nykyisessä prosessissa pidettiin joustavuutta, asiantuntijoiden hyödyntämistä sekä työnosittelua. Nämä kolme ominaisuutta ovat nykyisessä prosessissa keskeisessä roolissa, sillä arvioinnin voidaan katsoa olevan asiantuntija arviointia.

Kehitettäviä kohtia löytyi myös paljon. H2 ja H5 mainitsivat, että kannattavuuden arvioinnin toimivuutta ei mitata tarpeeksi. Myös H3 koki työmäärien arvioinnin ja toteutumisen seuraamisen olevan puutteellista. Haastatteluista kävi ilmi, että ajan- sekä tuenpuute ovat yhtenä haasteena kannattavuuden arvioinnissa. Arvioita tekevät ovat kiireisiä ja usein kannattavuuden arvioinnille tai työmäärän arvioinnille ei varata tarpeeksi aikaa. Suurena kehityskohtana nousi lähes kaikkien haastateltavien osalta systemaattisuuden puute. Arviot ovat tällä hetkellä liian henkilösidonnaisia ja arviot vaihtelevat paljon projekteittain.

Työnosittelu on toimivaa tällä hetkellä ja sen avulla on helppo kommunikoida asiakkaan kanssa. H4 ja H6 mainitsivat kuitenkin, että työnosat tulisi olla jaettu vielä pienempiin osiin, jotta tarkkuus paranisi. Tämä vaatii kuitenkin lisää ajan käyttöä prosessiin. Huomiona voidaan mainita, että ketterä ohjelmistoprojekti mahdollistaa kuitenkin sen, että työmäärien arvioiden ei tarvitse olla täysin pitäviä. Ketterissä ohjelmistoprojekteissa tietyt ominaisuudet tulee toteuttaa, mutta niiden toteuduttua voidaan edetä joustavammin ominaisuuksien toteutuksien kannalta.

H2 ja H4 mainitsivat kannattavuuden arvioinnin kannalta suurena ongelmana sen, että projektien kustannuksia ei seurata tarpeeksi tarkasti. Pieniä kustannuksia ei huomioida tarvittavalla tasolla ja työmääriä ei osata tällä hetkellä kohdistaa tarpeeksi tarkasti työnosille. Työmäärien osalta tulisi olla paremmin selvillä niiden toteutukset sekä syyt sille, miksi työn osa esimerkiksi venyi. Olisi tärkeää nähdä, kuinka kauan mikäkin työnosan toteutus on kestänyt, jotta tulevaisuudessa työmäärän arvioinnit olisivat tarkempia. H6 ja H7 mainitsivat, että nykyisessä prosessissa ei oteta tarpeeksi huomioon riskien merkitystä. Asiantuntijat arvioivat riskit intuitiivisesti työmäärä arvioihin, mutta systemaattisuus puuttuu tästä menetelmästä.

Tutkimuksessa kannattavuuden arviontiin kehitetty malli synnytti paljon mielenkiintoa ja mielipiteitä. Ensivaikutelmana haastateltavat olivat yhtä mieltä siitä, että malli näyttäisi parantavan nykyistä prosessia ja se paneutuu hyvin haastatteluissakin esiintyneisiin ongelmiin. Haastateltavat kertoivat ensivaikutelman perusteella mallin lisäävän systemaattisuutta, ja työnosien jako rooleihin vaikuttaisi helpottavan kustannuksiin ja kannattavuuteen liittyvien pulmien ratkaisuun.

H2 ja H4 nostivat esiin roolien jaottelun vaikutuksen kannattavuuden arviontiin. Heidän mielestään se auttaa kannattavuuden arvioinnissa, sillä sen avulla hinnoittelu on helpompaa tehdä pohjautuen roolien erihintaisuuteen. Yleinen mielipide kompleksisuuden mukaan ottamisesta oli positiivinen. Kompleksisuuden systemaattinen arviointi ja sen avulla työmäärän arviointi koettiin hyväksi ominaisuudeksi ja haastateltavien se tuo malliin tarvittua systemaattisuutta. H5, H6 ja H7 nostivat esille sen, että malli yhdistää järkevällä tavalla asiantuntijoiden osaamisen ja mallin tuovan systemaattisuuden. Yleinen mielipide oli, että täysin systemaattinen malli ei toimisi, sillä projektit poikkeavat paljon toisistaan.

H4 ja H5 mainitsi haastattelussa, että malli vaikuttaa yksinkertaiselta käyttää. Haastatteluissa kävi ilmi, että kohdeyrityksessä on aiemmin kokeiltu muitakin kannattavuuden arviointimenetelmiä, mutta osa niistä on hylätty liiallisen monimutkaisuuden tai työläisyyden takia. Kompleksisuudessa koettiin hyvänä ominaisuutena, että asteikko arvioidaan arvoilla yksi, kaksi tai kolme, ja jokaiselle arvolle on oma selityksensä. Haastateltavat kokivat, että mikäli kompleksisuus arvioitaisi subjektiivisesti, se ei toisi tarpeeksi systemaattisuutta.

Lähes kaikki haastateltavat mainitsivat, että työn jako roolien mukaan on hyvä idea, mutta he kaipaivat malliin myös työtehtäväkohtaista jaottelua, joka on ollut aikaisemmin heidän prosessissaan. Osa haastateltavista kuitenkin mainitsi, että tämä mielipide voi kantautua siitä, että roolipohjainen työnosittelu on uusi tapa ja tämän takia aiheuttaa muutosvastarintaa. Mallin käyttöön toivottiin tarkkaa ohjeistusta, jotta asiantuntijat eivät tee arviointia vanhalla tavalla. Kompleksisuuden asteikkojen selitykset koettiin selkeinä, kaikki muut haastateltavat paitsi H1 olivat sitä mieltä, että kompleksisuus on helppo arvioida mallin asteikon mukaan.

Kaikki haastateltavat kertoivat ottavansa uuden mallin mielellään testiin eikä kukaan maininnut syytä, miksi ei ottaisi käyttöön. H2 mainitsi, että mallin muotoilun tulee olla yhteensopiva muiden yrityksessä käytettävien järjestelmien kanssa, jotta se otetaan laajasti käyttöön. H3 mainitsi myös, että mallin muotoiluun tulee panostaa, jotta siitä tulee mahdollisimman käyttäjäystävällinen.

5.2 Mallin testaamisen tulokset

Teorian pohjalta muodostetusta mallista haluttiin asiantuntijoiden mielipiteiden lisäksi saada kvantitatiivista dataa sen toimivuudesta. Mallia haluttiin testata kohdeyrityksen toimintaympäristössä oikealla projektidatalla. Mallin testaamiseen käytetty data oli muodostunut jokaisen testatun projektin alussa kannattavuuden arviointivaiheessa, ja osa projekteista oli jo päättyneet eli niissä pystyttiin hyödyntämään myös toteutuneita lukuja.

Testaamiseen tarvittiin asiantuntijoiden arviota työn osien kompleksisuudesta ja niiden arviointi tapahtui haastattelujen yhteydessä. Mallin testaamisessa käytettiin neljää projektia, jotka lueteltu taulukossa 2. Projektit ovat salassa pidettäviä, jonka takia ne ovat nimetty nimillä P1, P2, P3 ja P4. Projektien nimien lisäksi taulukosta löytyy alkuperäiset arviot kokonaistyömäärästä ja alkuperäisen arvion mukainen joustovara, jota kohdeyrityksessä on käytetty projektin venymisien huomioimiseksi. Tämän lisäksi taulukossa 2 on esitetty jokaisen päättyneen projektin toteutunut työmääräarvio, mallin avulla tehty kokonaistyömäärä arvio sekä systemaattisin menetelmin laskettu joustovara ja prosentuaalinen ero projektin toteutuneeseen työmäärään tai alkuperäiseen arvioon verrattuna.

Case	Alkuperäinen työmääräarvio	Joustovara alkuperäisen menetelmän mukaan	Mallin avulla tehty työmääräarvio	Joustovara mallin perusteella	Prosentuaalinen ero toteutukseen
P1	405	21%	419	25%	3%
P2	120	20%	127	26%	6%
P3	642	24%	666	29%	4%
P4	270	24%	270	24%	0%

Taulukko 2. *Mallin testaamisen tulokset*

Tuloksista voidaan nähdä, että tutkimuksessa muodostunut malli antaa hieman suurempia arviota, kuin kohdeyrityksessä on aikaisemmin arvioitu. Erot ovat pieniä sillä P1:ssa ero alkuperäisen arvion ja uuden mallin avulla tehdyn arvion välillä oli 3 prosenttia, P3:ssa sama ero oli 4 prosenttia ja P4:ssa ero oli 0 prosenttia. Projekti 2 oli saatu päätökseen, joten sen osalta mallin toimivuutta voitiin arvioida jo päättyneeseen projektiin.

P2:den toteutunut työmäärä oli 120 htp ja mallin avulla tehty työmäärä arvio puolestaan oli 127 htp, mikä tarkoittaa 6 prosentin eroa.

Kohdeyrityksen joustovarot projektien osalta vaihtelivat 20%-24% välillä eli vaihteluväli oli 4. Mallin avulla joustovara vaihtelu puolestaan oli 24%-29% välillä eli vaihteluväli oli 5. Mallin avulla arvioidessa joustovarot ja tätä kautta kokonaistyömäärä arviot ovat hieman kohdeyrityksen arviota suurempia ja vaihteluväli projektien joustovarojen välillä on molemmilla tavoilla arvioidessa lähes sama. Voidaan siis todeta, että mallin avulla tehdyt työmääräarviot ovat hieman alkuperäisiä arviota suurempia, mutta projektien välillä joustovara arvioiden vaihtelu on lähes samalla tasolla.

Testaamisen tarkoitus täyttyi hyvin, sillä tämän avulla saatiin verrattua kohdeyrityksen nykyistä arviointimenetelmää tutkimuksessa muodostettuun malliin. Erot olivat pieniä, mikä on positiivinen asia, sillä kohdeyrityksen työmäärien ja kannattavuuden arviointi on ollut hyvällä tasolla. Kohdeyritys on viime vuosina panostanut työmäärien ja kannattavuuden arviointiprosessiin mikä näkyy heidän arvioissaan ja mikä on poikkeus alalla. Tutkimukseen sisältyi vain neljä projektia, sillä tätä aikaisemmat projektit ovat hieman erilaisella arviointimenetelmällä arvioituja, jonka takia ne eivät olisi hyvää dataa vertailuun tutkimuksessa muodostetun mallin toimivuudesta. Tutkimuksen tarkoituksena oli parantaa kannattavuuden arviointia myyntivaiheessa ja testatun datan sekä haastattelujen pohjalta mallia voidaan jatkojalostaa sekä säätää mallin matemaattisia arvio tulosten mukaan.

6. KONSTRUKTION ARVIONTI

6.1 Mallin arviointi

Haastattelut sekä mallin testaaminen antoi paljon uutta tietoa mallin toimivuudesta ja kannattavuuden arviointiprosessin kehittämisestä. Asiantuntijoiden haastattelut antoivat paljon tietoa, jota teorian pohjalta ei tajunnut ottaa huomioon. Teorian pohjalta voitiin todeta, että kannattavuudenarviointi prosessissa olisi ideaalista hyödyntää sekä asiantuntijoiden osaamista, että mallien tuomaa systemaattisuutta. Tämä oli yksi lähtökohta, jota tutkimuksen tuloksena pyrittiin ratkaisemaan.

Haastattelut osoittivat, että moni hypoteesi osui oikeaan. Haastattelujen pohjalta voidaan todeta, että hypoteesi asiantuntijoiden sekä mallien yhdistämisestä osui oikeaan. Asiantuntijat olivat yhtä mieltä siitä, että systemaattisuuden lisääminen projektien välille on hyödyllistä. Kohdeyrityksen osalta työmääränarviointiprosessi on hyvällä tasolla alan keskiarvoon nähden, mutta projektien välillä huomattiin olevan liikaa vaihtelua subjektiivisuuden takia.

Kannattavuuden arviointia lähdettiin parantamaan roolikohtaisen hinnoittelun avulla, mutta se sai ristiriitaisia mielipiteitä asiantuntijoilta. Moni haastateltava oli sitä mieltä, että roolipohjaisuus työmääräarvioinnissa on hyvä asia, sillä sen avulla todellisten kustannusten sekä kannattavuuden arviointia on helpompi tehdä. Työmäärän jakaminen roolien välille oli kuitenkin haastateltaville uusi asia, jonka takia myös vastarintaa löytyi. Kaikki haastateltavat piti roolipohjaisesta ajattelusta, mutta se muuttaa tämän hetkistä työtehtäväpohjaista ajattelutapaa, joten sen toimivuudesta ei oltu varmoja. Osa haastateltavista ehdotti roolipohjaisuuden sekä tehtäväpohjaisuuden yhdistämistä, jotta työnosien arvioinneista saadaan tarkkoja, mutta samalla kannattavuuden arviointia voidaan kehittää.

Alustavan mallin mukaan asiantuntijoiden tuli arvioida kompleksisuutta riskien, työn koon sekä työn monimutkaisuuden mukaan. Tämä ei ollut testien sekä haastattelujen mukaan tarpeeksi systemaattinen tai yksinkertainen keino arvioida kompleksisuutta. Riskit ovat hankalaa arvioida, joka johtaa siihen, että asiantuntijat joutuvat käyttämään liikaa aikaa arviointiprosessiin. Liiallisen ajan tuhlaaminen arviointivaiheessa lisää työvoima kustannuksia, joka taas heikentää projektin kokonaisuutta.

Yleisesti ottaen haastattelujen osalta malli toimi työnosittelun, roolien sekä kompleksisuuden osalta. Työnjakaminen rooleihin ei kuitenkaan yksittäin riitä vaan tähän tulee keksiä parempi ratkaisu jatkoa ajatellen. Kompleksisuus oli asiantuntijoiden mukaan erittäin järkevää ajattelua ja sen avulla optimistisimman, todennäköisimmän ja pessimistisimmän arvion laskeminen oli kaikkien haastateltavien mielestä hyvä osa lisäämään systemaattisuutta.

Näiden arvioiden avulla malli laskee lopullisen työmääräarvion beta-jakaumaa hyödyntäen. Malli laskee lopullisen työmääräarvion beta-jakauman avulla laskemalla sen 80% tarkkuudella. Tämä toi testien mukaan kaivattua joustovaraa projekteille. Kohdeyrityksen projektit ovat yleensä ajallaan tai vain hieman viivästyneitä alkuperäisistä työmääräarvioista. Mallia testatessa saatiin alkuperäisiä arvioita hieman suurempia arvioita, joka vaikuttaa olevan hyvä taso joustovaralle.

Haastateltavat antoivat paljon lisätietoa kannattavuudenarvioinnista ja haastatteluotanta oli hyvän kokoinen tutkimukseen nähden. Tutkimuksessa haastateltiin seitsemää arvioinnin osaa, joista jokainen tuli hieman erilaisesta taustasta. Haastateltavien heterogeenisyys mahdollisti kattavat tutkimustulokset jo seitsemällä haastateltavalla.

Mallin testaamisen osalta saatiin myös määrällistä dataa siitä, miten malli arvioi työmääriä alkuperäisiin arvioihin nähden. Testaamisen osalta olisi saatu vielä paremmat tulokset, mikäli testattavia projekteja olisi ollut enemmän ja projektit olisivat olleet jo valmistuneita projekteja. Suuremman otoskoon avulla mallin systemaattista osuutta olisi saatu optimoitua haluttuun tasoon. Mallin taustalla pyörivää matematiikkaa on kuitenkin helppo säätää jatkokehityksenä projekti määrien kasvaessa.

Yleisesti ottaen malli näyttäisi olevan toimiva työkalu kannattavuudenarviointi prosessissa. Haastatteluissa esiintyi tietenkin pieniä kehitystarpeita, jotka jatkokehityksessä kannattaa ottaa huomioon. Mallin systemaattinen osuus tulee myös käytön mukana paranemaan, kun saadaan lisää dataa matemaattisten kaavojen toimivuudesta. Mallin pääkohtina olleet työnosittelu ja kompleksisuuden arviointi, jonka avulla lopullinen työmääräarvio tehdään, vaikuttaisi toimivan hyvin haastattelujen sekä testaamisen pohjalta.

6.2 Mallin jatkokehitys mahdollisuudet

Tutkimuksen alustavana konstruktiona muodostetussa kannattavuuden arviointi mallissa oli paljon onnistuneita elementtejä, mutta myös puutteita. Malli muodostettiin hyödyntäen pieniä osia eri ohjelmistojen arviointi työkaluista. Empirian avulla voitiin osoittaa,

että näiden työkalujen avulla muodostunut malli oli toimiva ja siitä on mahdollista saada erinomainen työkalu jatkokehityksen avulla.

Jatkokehityksen osalta suurimpia kehityskohteita ovat työtehtävien mukaan otto roolipohjaisen työmääräarvioinnin lisäksi, mallin matemaattisten osuuksien optimointi käytön myötä, kompleksisuuden arviointiin systemaattisten kriteerien lisääminen, riskien ottaminen huomioon paremmin, tarkempi kustannusten seuraaminen sekä hinnoittelumenetelmien pohtiminen. Asiantuntijoiden haastattelujen mukaan kehityskohteiksi nousi erityisesti työtehtäväkohtainen työmääräarviointi sekä kustannusten seuraaminen. Teorian pohjalta kannattavuuden kannalta tärkeimpiä ominaisuuksia ovat kustannukset, riskit sekä hinnoittelu, jotka jäivät pienemmälle huomiolle alustavassa konstruktiossa. Mallin testaaminen osoitti matemaattisten arvojen olevan hyvällä mallilla, mutta mallin käyttäminen osoittaa paremmin jatkossa sen, mihin suuntaa sitä tulee kehittää.

Asiantuntijat kokivat mallin hyvänä ominaisuutena roolipohjaisen työmääräjaottelun, jonka idea syntyi kustannuspohjaisen hinnoittelun tuloksena. Tämä oli kaikille haastateltaville uusi ominaisuus kannattavuuden arviointi prosessiin, sillä moni on aikaisemmin arvioinut työmääriä tehtäväkohtaisesti. Uudet ideat luovat monesti muutosvastarintaa, josta tässäkin oli osittain kyse. Tehtäväkohtaisen työmäärän arvioinnin mukaan ottaminen malliin on kuitenkin hyvä idea jatkokehitystä varten.

Roolikohtainen työmääräarviointi auttaa paljon kannattavuuden arvioinnissa, mutta työmäärästä saadaan vielä tarkempia arvioita, kun asiantuntijat pystyvät roolien lisäksi ottamaan huomioon kuhunkin työtehtävään kuluvan ajan. Näin ollen mallin jatkokehityksen jälkeen malli ottaa huomioon kannattavuuden arviointia parantavan roolipohjaisuuden sekä työmääräarviointia tarkentavan ja helpottavan tehtäväkohtaisen ajattelun.

Empiiriset tutkimuksen osoittivat, että verrattuna aikaisempiin arvioihin tai toteumiin, tutkimuksessa muodostettu malli antoi hieman aikaisempaa suurempia arvioita. Tämä voi olla hyvä asia, ottaen huomioon alan yleisen trendin työmäärien arvioiden ollessa yleensä liian pieniä. Kohdeyrityksessä kuitenkin työmäärän arvioit ovat olleet alan yleiseen tasoon verraten hyvällä tasolla, joten malli saattaa tuottaa tällä hetkellä myös liian suuria arvioita.

Kannattavuuden arviointiin muodostettua mallia tulee käyttää projekteissa ja sen tuottamia työmäärä arvioita tulee seurata projektien päätyttyä. Tämän avulla mallin toimivuudesta saadaan lisää dataa ja mallin taustalla toimivia matemaattisia arvoja sekä kaavoja voidaan sen mukaan optimoida. Tällä hetkellä malli arvioi kompleksisuuteen pohjautuen optimistisemmän sekä pessimistisimmän työmäärä arvion sekä laskee näiden avulla

beta-jakaumaa hyödyntäen lopullisen työmääräarvion. Optimistisimman sekä pessimistisimmän työmääräarvion lukuja, ja beta-jakauman todennäköisyyden arvoja voidaan siis jatkokehityksenä optimoida.

Kompleksisuuden avulla riskien sekä työn monimukaisuuden arviointi sai asiantuntijoilta hyvää palautetta. Kompleksisuus tulee arvioida mallissa asteikolla yhdestä kolmeen ja luku määräytyy sen mukaan kuinka suurena, asiantuntija kokee työn riskit sekä kuinka monimutkaisena hän pitää työtä. Tämä menetelmä toimi kohtalaisen hyvin, mutta osa haastateltavista nosti kehitysideana esille sen, että kompleksisuuden arviointiin voisi ottaa systemaattiset kriteerit käyttöön, subjektiivisen riskien ja monimutkaisuuden arvioinnin tilalle.

Jatkokehityksenä mallin kompleksisuus tulisi määräytyä selviin kriteereihin pohjautuen, jotta mallin systemaattisuus lisääntyä entisestään ja kannattavuuden arviointi prosessista saadaan pois henkilösidonaisuutta. Alustavaa konstruktiota muodostaessa kompleksisuuden perimmäinen tarkoitus oli ottaa riskit huomioon. Riskien suuruus määräytyy sen toteutumisen todennäköisyyden ja vaikutuksen mukaan. Nämä voidaan ottaa ohjelmointi projektissa huomioon esimerkiksi luomalla kriteerit työn osan menetelmien sekä prosessin tuntemisen ympärille. Mikäli projektissa tunnetaan työnosan tekoon vaadittavat prosessit ja menetelmät, kompleksisuus on pieni. Puolestaan, kun työnosan tekoon vaadittavat prosessit ja menetelmät eivät ole toimittajalle tuttuja on todennäköisempää, että riskit toteutuvat, jolloin kompleksisuuden tulee olla suuri.

Riskien arvioinnin sisällyttäminen kustannusten arviointiprosessiin on haastavaa. Sisällyttäminen on haastavaa, sillä ohjelmistoalan yritysten resurssit ovat rajalliset ja asiantuntijoiden sekä kirjallisuuskatsauksen mukaan kannattavuuden arviointiin ei monissa projekteissa ole käytössä tarpeeksi aikaa. Ideaali tapauksessa jokaiselle työnosalle tehtäisiin erikseen riskianalyysi. Työn osan mahdollisia riskejä listattaisi ja sen jälkeen jokaisen riskin toteutumisen todennäköisyyttä sekä sen vaikutusta toteutuessaan arvioidasi. Tästä syntyy kuitenkin erittäin paljon lisää työtä, joka ei ainakaan tutkimuksessa tehtyjen haastattelujen pohjalta ole realistista toteuttaa. Jatkokehityksenä on siis pohdittava sitä, miten riskejä saadaan paremmin mukaan kannattavuuden arviointiprosessiin vai onko se tarpeeksi tarkalla tasolla, mikäli sen sisällyttää kompleksisuuden arvioinnin kriteereihin.

Haastatteluissa esittele nousi myös kannattavuuden seuraamisen sekä kustannusten seuraamisen puute. Asiantuntijoiden mukaan kannattavuutta ei seurata tarpeeksi tarkasti eikä työntekijän kustannusten lisäksi mietitä tarpeeksi muita kustannuksia. Jotta

kannattavuuden arviointiprosessia on mahdollista kehittää, sen toteutumista tulisi seurata tarkasti. Yrityksen eivät saa tyytyä siihen, että projektin päätyttyä toimittajayritys on saanut tarpeeksi voittoa projektista vaan toteumaa tulisi verrata myyntivaiheessa arvioituun kannattavuuteen.

Vertailun avulla voitaisi havaita, kuinka paikkaansa pitäviä myyntivaiheessa tehdyt arvioit ovat ja tätä kautta voitaisi etsiä syitä sille miksi ne eivät ole pitäneet paikkaansa tai miksi ne ovat. Kannattavuuden arvioinnin kannalta on kriittistä olla tietoinen siitä, kuinka paljon kustannuksia projektissa syntyy ja mistä kustannukset syntyvät. Roolipohjaisuuden mukaan ottaminen kustannusten arvioinnissa on askel oikeaan suuntaan, sillä sen avulla työlle ei lasketa keskihintaa, vaan työn hinta muodostuu sen tekijän roolin kustannusten mukaan.

Kustannusten seuraamisessa on kuitenkin havaittu puutteita niin arvioiden ja toteutumien seuraamisen välillä, kuin sen osalta, mistä kustannukset syntyvät. Ohjelmistoalalla kustannukset syntyvät suurelta osin työntekijän kustannuksista, mutta on olemassa myös muita kustannuksia, jotka tulisi ottaa jatkokehitys ideana mukaan kannattavuuden arviointi prosessiin. Osa työstä saattaa ketterissäkin projekteissa olla syystä tai toisesta sellaista, josta ei voida laskuttaa ja se laskee projektin kannattavuutta. Esimerkiksi tällaisia kustannuksia tulee seurata, jotta yrityksen on mahdollista kehittää kannattavuuden arviointi prosessiaan.

Kannattavuudenarviointia tekevän yrityksen tulisi pohtia myös heidän hinnoittelustrategiaansa. Haastatteluissa nousi esille se, että kustannuspohjainen hinnoittelu on ohjelmistoalalla järkevää, sillä asiakkaat vaativat tietää mistä hinta muodostuu. Kustannuspohjainen hinnoittelu on tutkimuksessa muodostettuun konstruktion sisällytetty hinnoittelumenetelmä, mutta jatkokehityksenä voitaisi tutkia myös esimerkiksi arvopohjaisesta hinnoittelu menetelmää, jolloin hinta muodostuu projektin tuomasta arvosta.

Viimeisenä jatkokehitysideana on asiantuntijoiden kollektiivisen älykkyyden hyödyntämistä. Osa haastateltavista sanoi, että he hyödyntävät muiden kollegojen älykkyyttä kannattavuuden arviointi prosessissa, mutta moni mainitsi tämän olevan haastavaa resursipulan vuoksi. Delphi-menetelmän kaltainen yhteisen mielipiteen muodostaminen toisi varmasti lisää tarkkuutta työmäärien arviointiin, mutta se vaatii myös paljon resursseja. Jatkokehityksenä voidaan todeta, että tällainen kollektiivista älykkyyttä hyödyntävä kannattavuuden arviointi prosessi parantaisi työmäärien arvioinnin tarkkuutta, mutta resursien allokointia sen mahdollistamiseksi tulisi kehittää.

7. YHTEENVETO

7.1 Yhteenveto ja tutkimuskysymyksiin vastaaminen

Tutkimuksen tavoitteena oli selvittää vastaukset tutkimuskysymyksiin sekä muodostaa kannattavuuden arvioinnissa käytettävä malli. Tutkimuskysymyksiin pyrittiin pääosin vastaamaan teorian pohjalta, mutta niihin vastaamiseen hyödynnettiin myös empiriaa. Kirjallisuuskatsauksen avulla löydettiin mahdollisia vastauksia tutkimuskysymyksiin ja empirian oli tarkoituksena tukea näitä vastauksia sekä auttaa mallin kehityksessä. Tutkimuskysymykset luotiin sen pohjalta, mikä nähtiin teorian mukaan olevan tärkeimmät kannattavuuden tekijät.

Päätutkimuskysymys tutkimuksessa oli seuraavanlainen.

1. *Miten ketterän ohjelmistoprojektin kannattavuus tulee arvioida myyntivaiheessa, jotta sisältöön, aikatauluun, henkilöihin ja kustannuksiin liittyvät riskit pystytään huomioimaan?*

Apututkimuskysymykset, joiden avulla vastattiin päätutkimuskysymykseen, ovat seuraavanlaiset.

- 1.1. *Mitkä tekijät aiheuttavat riskejä ja miten riskit tulee ottaa huomioon ketterän ohjelmistoprojektin kannattavuuden arvioinnissa?*
- 1.2. *Mitkä tekijät vaikuttavat ketterän ohjelmistoprojektin kustannuksiin ja hinnoitteluun?*
- 1.3. *Millä keinoilla kannattavuudenarviointi prosessia voidaan parantaa?*

Apukysymyksiin pyrittiin vastaamaan tutkimalla kirjallisuutta riskien, kustannusten, hinnoittelun sekä kannattavuuden tai työmäärän arviointi mallien avulla. Riskien osalta kirjallisuus antoi paljon tietoa siitä, miten ja mitä tulisi ottaa huomioon riskejä analysoidessa. Riskien hallinnan osalta prosessi voidaan jakaa neljään vaiheeseen: riskien tunnistamiseen, riskianalyysiin, riskisuunnitteluun ja riskinvalvontaan. Lähteiden mukaan riskit voidaan jakaa kolmeen kategoriaan, jotka jakautuvat kuuteen tyyppiin. Kolme riski kategoriaa ovat projektiriskit, tuoteriskit sekä liiketoimintariskit.

Kuusi riskityyppiä puolestaan ovat teknologiariskit, henkilöriskit, organisaatoriskit, työkaluriskit, vaatimusriskit sekä arviointiriskit. Kirjallisuus osoittaa selvästi, miten riskit tulee ottaa huomioon ja mitä niiden osalta tulee ottaa huomioon. Kannattavuuden arvioinnin

osalta on tärkeää ottaa riskit huomioon, sillä riskien toteutuminen vaikuttaa projektin kannattavuuteen oleellisesti. Ketterissä ohjelmistoprojekteissa suuresti vaikuttavia riskejä ovat esimerkiksi teknologia- sekä työkaluriskit, sillä uusia ohjelistoja luodessa teko prosessit ja työkalut saattavat olla täysin uusia yritykselle.

Tutkimuksen tuotoksena syntynyt kannattavuuden arviointi malli ottaa jossain määrin huomioon ketteriin ohjelmistoprojekteihin sisältyvät riskit, mutta malliin ei saatu sisällytettyä laajaa riskianalyysia. Kannattavuuden arvioinnin osalta on oleellista tutkia myös syntyviä kustannuksia sekä hinnoittelumenetelmiä. Ohjelmistoala on kustannusrakenteeltaan selkeä, sillä ohjelmistot luodaan projektiluontaisesti ja suurin osa kustannuksista syntyy työnteko kustannuksista.

Empiriasta kävi kuitenkin ilmi, että työnteon kustannukset osataan ottaa hyvin mukaan kustannusarvionteihin, mutta muut kustannukset jäävät monesti ketterissä projekteissa taka-alalle. Asiakas maksaa toimittaja yritykselle tehdyn työn mukaan, jolloin esimerkiksi ihmisten liikkumiseen tai teknologioihin liittyviä kustannuksia ei laskuteta. Tämä johtaa siihen, että ketterissä ohjelmistoprojekteissa otetaan huomioon vain työnteko kustannukset, vaikka todellisia kustannuksia syntyy työnteon kustannusten lisäksi.

Tuotteen tai palvelun hinta voidaan katsoa muodostuvan katteesta sekä yksikkökustannuksista. Tähän liittyen yleisesti ottaen on olemassa kolme hinnoittelustrategiaa: kustannuspohjainen hinnoittelu, kilpailijapohjainen hinnoittelu sekä arvopohjainen hinnoittelu. Ketterissä ohjelmistoprojekteissa on pääosin käytössä kustannuspohjainen hinnoittelu, jossa asiakkaalle tarjottu hinta muodostuu kustannusten eli tehdyn työn mukaan. On kuitenkin oleellista tutkia jatkossa olisiko muun tyyppinen hinnoittelumenetelmä mahdollinen myös ohjelmistoalalla. Esimerkiksi arvopohjainen hinnoittelu voisi olla toimiva ratkaisu, mutta arvon ollessa subjektiivista, arvon määrittäminen ohjelmistoille on hankalaa.

Keinoja, joilla kannattavuudenarviointiprosessia voidaan parantaa, selvitettiin kirjallisuuskatsauksen sekä asiantuntijoiden haastattelujen avulla. Kirjallisuuden avulla tutkittiin sitä, mitkä ovat parhaat keinot kannattavuuden arviointiin ja ovatko systemaattiset menetelmät vai asiantuntija-arviot tarkempia arvioinnissa. Kirjallisuus osoittaa, että asiantuntijoiden merkitys kannattavuuden arviointi prosessissa on tärkeä, mutta systemaattisten menetelmien mukaan ottaminen voisi olla paras keino kannattavuuden arviointiin. Kumpikaan arviointitapa ei ole hyvä yksittäisenä, jonka takia tutkimuksessa pyrittiin löytämään keinot, joilla yhdistää asiantuntijoiden osaaminen sekä systemaattiset mallit.

Tutkimuksessa käsiteltiin kuusi erimenetelmää, joita kannattavuuden tai työmääränarvioinnissa voidaan käyttää ketterien ohjelmistoprojektien osalta. Näistä kuudesta menetelmästä pyrittiin ottamaan parhaat ominaisuudet ja yhdistämään ne yhteen tutkimuksessa muodostettuun malliin.

Tutkimuksessa muodostettu malli hyödyntää asiantuntijan osaamista työmäärien arvioinnissa, työn jaotteluissa eri työrooleille sekä työn kompleksisuuden arvioinnissa. Tämän lisäksi mallissa hyödynnettiin systemaattisista menetelmistä WBS-menetelmän työnosittelua, PERT-mallista tuttua optimistista, pessimististä sekä todennäköisintä työmäärä arvioita, FPA-menetelmästä tuttua kompleksisuuden arviointi tapaa sekä matemaattista beta-jakaumaa. Asiantuntija tekee parhaan arvionsa kunkin työn osan työmäärän arvioinnista ja hän arvio työosan kompleksisuuden. Tämän avulla malli laskee optimistisimmän sekä pessimistisimmän työmäärä arvion pohjautuen asiantuntijan rooleille määrättyihin työmääriin sekä kompleksisuuteen. Lopuksi malli ehdottaa lopullista työmäärä arviota, joka on laskettu beta-jakaumaa hyödyntäen.

Tämän työmääräarvioinnin jälkeen asiantuntijan tulee arvioida sitä, mihin hintaan projekti tulisi myydä ja minkälaisen kannattavuuden projekti saisi kyseisellä hinnalla. Haastattelujen pohjalta tutkimuksessa muodostettu malli vaikutti käyttökelpoiselta. Jatkokehitysideoita nousi ja aihetta olisi hyvä tutkia lisää. Mallin testaaminen osoitti, että mallin arviot vaikuttivat osuvilta, mutta mallia tulee testata lisää jatkossa. Ketterien ohjelmistoprojektien kannattavuus myyntivaiheessa tulee siis arvioida hyödyntämällä asiantuntijoiden osaamista sekä systemaattisia malleja. Kannattavuuden arvioinnissa tulee ottaa huomioon etenkin riskit, hinnoittelu sekä kustannukset. Projektien riskit tulee arvioida jokaiselle työnosalle erikseen ja riskien arvioinnissa on hyvä käyttää systemaattista menetelmää, jotta projektit ovat vertailukelpoisia ja niiden onnistumista kannattavuuden osalta voidaan arvioida.

7.2 Tutkimuksen arviointi

Tutkimusta voidaan arvioida tieteellisesti katsottuna neljästä näkökulmasta. Nämä neljä näkökulmaa tutkimuksen arviontiin on: 1.) uskottavuus ja luotettavuus, 2.) sovellettavuus ja yleistettävyyys, 3.) johdonmukaisuus ja varmuus sekä 4.) vahvistettavuus ja neutraaliuus. (Guba, 1981) Tutkimuksen onnistumista voidaan tarkastella myös käytännön onnistumisien kannalta vertaillen tavoitteita ja lopputuloksia.

Tutkimuksen uskottavuutta ja luotettavuutta puoltaa teoriaan pohjautuvan konstruktion muodostaminen. Konstruktion muodostaminen on selitetty selvästi teoriaan pohjautuen ja koko prosessi on ollut läpinäkyvä. Tutkimuksessa väitteet on perusteltu kirjallisuus

lähteiden avulla ja mallia muodostaessa selitettiin miksi mitäkin osia teoreettisista malleista otettiin mukaan. Lähteitä tutkiessa on kuitenkin huomattava, että osa lähteistä on vanhoja, sillä moni alalla käytettävistä malleista on luotu vuosikymmeniä sitten. Tieteellisesti muodostetut mallit ovat kuitenkin vieläkin käyttökelpoisia, joten lähteiden ikä ei heikennä tutkimuksen luotettavuutta tai uskottavuutta merkittävästi.

Empirian osalta on tärkeää huomioida tutkijan ja kohdeyrityksen välinen suhde. Tutkija työskenteli samassa yrityksessä haastateltavien kanssa, mikä voi vaikuttaa haastateltavien vastauksiin. Oletuksena on, että haastateltavat vastasivat rehellisesti kysymyksiin, ja haastattelija nähtiin objektiivisena tutkijana. Myös data, jota käytettiin mallin testaamiseen tuli kohdeyritykseltä. Tämän takia ei voida täysin objektiivisesti todeta mallin toimivuutta, sillä kohdeyrityksen projektit olivat subjektiivisesti valittu tutkimukseen.

Tutkimuksen sovelluttamisen ja yleistettävyyden osalta voidaan todeta, että tutkimus on toteutettu selittäen kaikki vaiheet mahdollisimman tarkasti. Tämän takia tutkimus on helposti toistettavasti. Haastattelut vaikuttivat paljon tutkimuksen jatkokehitys suuntaan. Mikäli tutkimus toistettaisiin, on todennäköistä, että haastattelut tuovat hyvin erilaisia tuloksia, jolloin jatkokehitys mahdollisuudet poikkeavat paljon tämän tutkimuksen jatkokehitys mahdollisuuksista.

Tutkimuksen varmuuden osalta tulisi jälleen pohtia empiriaa. Tutkimuksen empiria on tieteellisellä tasolla kohtalaisen suppea käytettävien resurssien vuoksi. Suurempi otanta haastateltavien sekä käytettävien projektien osalta olisi tuoneet varmasti erilaiset tulokset. Mikäli aihetta halutaan tutkia lisää, suurempi empiirinen otanta on hyvä lähtökohta.

Tutkimuksen vahvistettavuuden ja neutraalisuuden osalta voidaan nähdä, että tutkimus on onnistunut. Tutkimus on tuottanut uutta tietoa tutkimus kentälle etenkin empiiristen haastattelujen avulla ja tutkimuksen rungon tarkoituksena oli luoda mahdollisimman hyvät keinot tutkimuskysymyksiin vastaamiseen.

Tutkimuksen voidaan katsoa onnistuneen hyvin tavoitteisiin nähden. Tutkimuksen pää-tavoitteena oli keksiä keinot, joilla parantaa ketterien ohjelmistoprojektien kannattavuuden arviointia myyntivaiheessa ja empiiristen haastattelujen perusteella kehitystä oli tapahtunut. Kohdeyrityksen kannalta tutkimuksessa pyrittiin ratkaisemaan kannattavuuden arvioinnin prosessin kannattavuuteen liittyvät haasteet. Tämä sopi hyvin tutkimuskysymyksiin, sillä systemaattisuutta pystyttiin lisäämään teoriaan pohjautuvia menetelmiä hyödyntämällä.

Tutkimuksen voidaan katsoa onnistuneen eri menetelmien yhdistämisessä käyttökelpoiseksi malliksi. Teoriaan pohjautuneet mallit eivät ole olleet ohjelmistoalalla toimiva keino arvioida työmääriä tai kannattavuutta, mutta niiden yhdistäminen asiantuntijoiden

osaamiseen on huomattu olevan paras keino arviointiin. Käytännönläheisestä näkökulmasta katsottuna haasteena on kannattavuuden arvioinnissa ollut erityisesti resurssien puute. Tämän takia mallia muodostaessa oli tärkeää miettiä myös, kuinka raskas se on käyttää. Haastatteluista ilmeni, että malli ei tulisi saamaan laajaa käyttöönottoa, mikäli sen käyttö vie liikaa aikaa ja resursseja.

Teorian pohjalta saatiin muodostettu tehokkaasti käytössä oleva menetelmä, joka hyödyntää osia useista systemaattisista menetelmistä sekä tärkeimpänä asiantuntijoiden arviointikykyä. Haasteeksi tutkimuksessa muodostui riskien sisällyttäminen malliin laajassa skaalassa. Jatkokehityksessä noussut kompleksisuuden arviointi kriteereittäin sisällyttää riskejä mukaan, mutta sekään ei sisällä riskianalyysiä. Tutkimuksen empiirisestä osuudesta mallin testaaminen datan avulla olisi voinut olla laaja mittaisempaa, jotta mallin toimivuutta olisi voitu tarkastella paremmin. Tämä on kuitenkin helppo korjata jatkokehityksessä sekä käyttämällä mallia ja seuraamalla toteumia.

Teorian osalta kannattavuuden arviointimenetelmiä olisi voitu sisällyttää vielä enemmän ja eri osia toimivaksi huomattavista menetelmistä olisi voitu sisällyttää tutkimuksessa muodostettuun malliin. Teoriaa kuitenkin tutkittiin tämän työn resurssien puitteissa ja teorian määrä vastaa tutkimuksen tavoitteisiin.

Empiirisestä näkökulmasta haastattelut antoivat paljon arvoa mallin kehitykselle. Tutkimuksessa haastateltiin yhteensä seitsemää henkilöä, jotka olivat erilaisista taustoista tulleet alalle ja tekivät kannattavuuden arviointia erinäkökulmista. Tutkimukseen olisi voitu sisällyttää lisää haastatteluja, sillä niistä saatu arvo oli todella suurta. Hypoteesina oli kuitenkin, että mikäli haastattelujen määrä olisi kasvanut, haastattelujen vastaukset olisi toistaneet enemmän toisiaan ja uusia näkökulmia ei olisi saatu.

7.3 Aiheen jatkojalostus

Edellisessä luvussa puhuttiin mallin jatkojalostamisesta ja samassa luvussa sivuttiin myös aiheen jatkojalostamiseen liittyviä asioita. Tämän tutkimuksen tarkoituksena oli selvittää, miten ketterän ohjelmistoprojektin kannattavuus tulee arvioida myyntivaiheessa, jotta sisältöön, aikatauluun, henkilöihin ja kustannuksiin liittyvät riskit pystytään huomioimaan. Tutkimuksen tarkoituksena oli myös luoda malli, jota kannattavuuden arviointi prosessissa voidaan hyödyntää.

Aiheen osalta tutkimuskysymykseen pyrittiin vastaamaan selvittämällä yleisesti, mikä ketterä ohjelmistoprojekti on, mitä hinnoittelumenetelmiä on, miten riskejä voidaan arvioida sekä mitä yleisesti käytettyjä malleja ohjelmistoprojekteissa hyödynnetään. Tämä

tutkimuskysymyksiin vastaaminen tehtiin kirjallisuuskatsauksen avulla, ja tutkimuksen myöhemmässä vaiheessa aiheesta haastateltiin alan ammattilaisia semi-strukturoidun haastattelun muodossa.

Aiheen jatkojalostuksen osalta tutkimusta voitaisi kehittää soveltumaan johonkin tiettyyn ketterän kehityksen menetelmään tai siirtämällä kannattavuuden arvioinnin myös vesiputousmallilla tehtäviin ohjelmistoprojekteihin. Jatkokehityksen kannalta tutkimuksen rajauksia voidaan muuttaa ja tätä kautta saada erilaisia tuloksia. Tämän tutkimuksen osalta yhtenä tärkeimpänä rajauksena voidaan pitää projektin kokoa, jota voidaan jatkokehityksessä pidentää tai lyhentää.

Hinnoittelumenetelmien osalta jatkokehityksenä voitaisi pohtia miten eri hinnoittelumenetelmät vaikuttavat projektin kannattavuuteen. Tässä tutkimuksessa käsiteltiin yleisimpiä hinnoittelumenetelmiä, mutta eri hinnoittelumenetelmien testaaminen mallissa ei ollut mahdollista resurssien puitteissa.

Riskien osalta on todettu, että tutkimuksessa luotuun malliin ei saatu sisällytettyä riskianalyysiä halutulla tasolla. Riskit sekä niiden toteutuminen on oleellisena osana kannattavuuden arviointia ja jatkokehityksessä niiden vaikutusta sekä niiden huomioimisen keinoja tulisi tarkastella paremmin. On selvää, että mikä tahansa projekti sisältää riskejä, mutta ohjelmistoalalla riskien tunnistaminen on hankalaa, sillä projekteissa luodaan monesti jotain mitä aikaisemmin ei olla luotu ja tekotavat voivat olla täysin uusia.

Ohjelmistoalalla kannattavuuden arvioinnissa sekä työmäärien arvioinnissa käytetään monenlaisia malleja, mutta parhaimmaksi kannattavuuden arviointitavaksi on silti todettu asiantuntija-arvioinnit. Useat lähteet toteavat kuitenkin, että asiantuntijoiden tueksi olisi hyvä saada systemaattisia menetelmiä. Jatkokehityksenä systemaattisten menetelmien merkitystä voitaisiin tutkia tarkemmin ja löytää parhaat menetelmät, joita ohjelmistoprojektien kannattavuuden arvioinnissa voitaisi käyttää. Systemaattisten menetelmien hyödyntäminen kannattavuuden arvioinnissa on haastavaa ohjelmistoalan luonteen takia, mutta tämä ei tarkoita sitä, että niitä ei tulisi hyödyntää. Jatkokehityksen tavoitteena voisi olla myös luoda uusia systemaattisia menetelmiä, joita projektien arviointi prosessissa voidaan hyödyntää.

LÄHTEET

- Abdulhalim, H., Lurie, Y., & Mark, S. 2018. *Ethics as a Quality Driver in Agile Software Projects*. *Journal of Service Science & Management*, 11, 13.
- Anding, M., & Hess, T. (2004). *Modularization, individualization and the first-copy-cost-effect: Shedding new light on the production and distribution of media content (No. 1/2004)*. Research paper, Institute for Information Systems and New Media (WIM), University of Munich.
- Aranda, J., and Easterbrook, S., (2005). *Anchoring and Adjustment in Software Estimation*. *Software Engineering Notes*, (30:5): 346-355.
- Ashraf, S., & Aftab, S. (2017). *IScrum: An Improved Scrum Process Model*. *International Journal of Modern Education & Computer Science; Hong Kong*, 9(8), 16.
- Atkinson, R. 1999. *Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria*. *International journal of project management*, 17(6), 337-342.
- Barki, H., Rivard, S. and Talbot, J. (1993) 'Toward an Assessment of Barki, H., Rivard, S. and Talbot, J. (1993) 'Toward an Assessment of Software Development Risk', *Journal of Management Information Systems*, 10(2), pp. 203–225. *Software Development Risk*, *Journal of Management Information Systems*, 10(2), pp. 203–225.
- Barry, B. (1981). *Software engineering economics*. New York, 197.
- Benaroch, M. & Appari, A. 2010, "Financial Pricing of Software Development Risk Factors", *IEEE Software*, vol. 27, no. 5, pp. 65-73.
- Boehm, B., Abts, C., & Chulani, S. (2000). *Software development cost estimation approaches—A survey*. *Annals of software engineering*, 10(1-4), 177-205.
- Bontis, N., & Chung, H. (2000). *The evolution of software pricing: from box licenses to application service provider models*. *Internet Research*, 10(3), 246-255.
- Brauer, H. (1999). *Price setting in international markets (No. 915)*. Kiel working paper.
- Braun, V., & Clarke, V. (2006). *Using thematic analysis in psychology*. *Qualitative Research in Psychology*, 3(2), 77–101.
- Brotherton, S. A., Fried, R. T., & Norman, E. S. (2008). *Applying the work breakdown structure to the project management lifecycle*. In *PMI Global Congress Proceedings (pp. 1-15)*.
- Brown, B. B. (1968). *Delphi process: A methodology used for the elicitation of opinions of experts (No. RAND-P-3925)*. RAND Corp Santa Monica CA.

- Børte, K., & Nerland, M. (2010). *Software Effort Estimation as Collective Accomplishment: An analysis of estimation practice in a multi-specialist team*. *Scandinavian J. Inf. Systems*, 22(2), 4.
- Caralli, R., Stevens, J. F., Young, L. R., & Wilson, W. R. (2007). *Introducing octave allegro: Improving the information security risk assessment process*.
- Chalotra, S., Sehra, S. K., Brar, Y. S., & Kaur, N. (2015). Tuning of cocomo model parameters by using bee colony optimization. *Indian Journal of Science and Technology*, 8(14), 1.
- Chow, T., & Cao, D. B. 2008. *A survey study of critical success factors in agile software projects*. *Journal of systems and software*, 81(6), 961-971.
- Cohn, M. (2005). *Agile estimating and planning*. Pearson Education.
- Cook, D. L. (1966). PROGRAM EVALUATION AND REVIEW TECHNIQUE--APPLICATIONS IN EDUCATION.
- Cooper, R., & Kaplan, R. S. (1988). *Measure costs right: make the right decisions*. *Harvard business review*, 66(5), 96-103.
- David Hillson, (2003) "Using a Risk Breakdown Structure in project management", *Journal of Facilities Management*, Vol. 2 Issue: 1, pp.85-97
- David Hillson, (2003) "Using a Risk Breakdown Structure in project management", *Journal of Facilities Management*, Vol. 2 Issue: 1, pp.85-97
- Evaluating the Cost of Software Quality*, Sandra A. Slaughter, Donald E. Harter, and Mayuram S. Krishnan, 98
- Finnie, G. R., Wittig, G. E., & Desharnais, J. M. (1997). A comparison of software effort estimation techniques: using function points with neural networks, case-based reasoning and regression models. *Journal of systems and software*, 39(3), 281-289.
- Globerson, S. (1994). Impact of various work-breakdown structures on project conceptualization. *International Journal of Project Management*, 12(3), 165-171.
- Guba, E. (1981) *Criteria for Assessing the Trustworthiness of Naturalistic Inquiries*. *Educational Communication and Technology Journal* 29. 1981. pp. 75-91.
- Hanna, N., & Dodge, H. R. (2017). *Pricing: policies and procedures*. Macmillan International Higher Education.
- Harmon, R., Demirkan, H., Hefley, B., & Auseklis, N. (2009, January). *Pricing strategies for information technology services: A value-based approach*. In *2009 42nd Hawaii International Conference on System Sciences* (pp. 1-10). IEEE.
- Hinterhuber, A. (2008). *Customer value-based pricing strategies: why companies resist*. *Journal of Business Strategy*. 4(29), 41–50.
- Jaakonaho, Kaisa. 2009. *Sopimusmallit ketterän lähestymistavan mukaisissa projekteissa*. Jyväskylä: Jyväskylän yliopisto.

- Jung, Y., & Woo, S. (2004). Flexible work breakdown structure for integrated cost and schedule control. *Journal of construction engineering and management*, 130(5), 616-625.
- Juvonen, R. (2018). Ohjelmistoprojektin sudenkuopat ja miten ne vältetään. BoD-Books on Demand.
- Jørgensen, M. (2007). *Forecasting of software development work effort: Evidence on expert judgement and formal models*. *International Journal of Forecasting*, 23(3), 449-462.
- Jørgensen, M. 2007. *A critique of how we measure and interpret the accuracy of software development effort estimation*. In *First international workshop on software productivity analysis and cost estimation*. Information Processing Society of Japan Nagoya.
- Jørgensen, M., and Grimstad, S. (2005). *Over-optimism in Software Development Projects: "The winner's curse"*. In: *Proceedings of IEEE CONIELECOMP, IEEE Computer Society, Puebla, Mexico*, pp. 280–285.
- Jørgensen, M., and Grimstad, S., (2008). *Avoiding Irrelevant and Misleading Information When Estimating Software Development Effort*. *IEEE Software*, (May/June): 78-83.
- Jørgensen, M., Boehm, B., & Rifkin, S. (2009). Software development effort estimation: Formal models or expert judgment?. *IEEE software*, 26(2), 14-19.
- Kasanen, E. (1993). *The Constructive Approach in Management Accounting Research*, 23
- Kotler, P. & Armstrong, G. (2008). *Principles of marketing*. New Jersey: Pearson Prentice Hall.
- Laitinen, E.K. 2007. *Kilpailukykyä hinnoittelulla*. Helsinki: Talentum.
- Law, E. L.-C., & Lárusdóttir, M. K. 2015. *Whose Experience Do We Care About? Analysis of the Fitness of Scrum & Kanban to User Experience*. *International Journal of Human-Computer Interaction*, 31(9), 584–602.
- Leffingwell, D. 2011. *Agile software requirements: lean requirements practices for teams, programs, & the enterprise*. Agile software development series. Addison-Wesley.
- Linstone, H. A., & Turoff, M. (Eds.). (1975). *The delphi method* (pp. 3-12). Reading, MA: Addison-Wesley.
- Longstreet, D. (2002). *Fundamentals of function point analysis*. Longstreet Consulting, Inc.
- M. Coram and S. Bohner, "The impact of agile methods on software project management," *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)*, Greenbelt, MD, USA, 2005, pp. 363-370.

- Malhotra, N. (1996). The impact of the academy of marketing science on marketing scholarship: An analysis of the research published in JAMS. Academy of Marketing Science Journal, 4(24), 291-298.*
- Mayer, R. J. (1987). Winning Strategies for Manufacturers in Mature Industries. Journal of Business Strategy, 8(2), 23-30.*
- Moløkken, K., & Jørgensen, M. (2003). A review of surveys on software effort estimation.*
- Nagle, T. T., & Holden, R. K. (2002). The strategy and tactics of pricing: A guide to profitable decision making.*
- Neilimo K & Uusi-Rauva E. 2005. Johdon laskentatoimi. 6-7. painos. Helsinki: Edita.*
- Nelson, R. R., & Morris, M. G. 2014. IT Project Estimation: Contemporary Practices and Management Guidelines. MIS Quarterly Executive, 13(1).*
- Ohsugi, N., Inoguchi, N., Yamamoto, H., Shimizu, S., Hattori, N., Yoshino, J., ... & Kitani, T. (2007). Fair Software Value Quantification by Productivity Data Analysis. Software Productivity Analysis and Cost Estimation.*
- Oyegoke, A. (2011). The constructive research approach in project management research. International Journal of Managing Projects in Business, 4(4), 573-595*
- Putnam, L. H. (1978). A general empirical solution to the macro software sizing and estimating problem. IEEE transactions on Software Engineering, (4), 345-361.*
- Richards, J., Reynolds, J. & Hammerstein, M. (2005). Neglected art of strategic pricing. Financial Executive, 5(21), 26-29.*
- Saunders, M., Lewis, P., Thornhill, A. 2012. Research methods for Business Students, 6th edition. Prentice Hall.*
- Schwaber, K. 1997. SCRUM Development Process. Teoksessa J. Sutherland, C. Casanave, J. Miller, s. Patel, & G. Hollowell (Toim.), Business Object Design & Implementation (ss. 117-134). London: Springer London.*
- Shapiro, C., Carl, S., & Varian, H. R. (1998). Information rules: a strategic guide to the network economy. Harvard Business Press.*
- Sheta, A. F. (2006). Estimation of the COCOMO model parameters using genetic algorithms for NASA software projects. Journal of Computer Science, 2(2), 118-123.*
- Slaughter, S. A., Harter, D. E., & Krishnan, M. S. 1998. Evaluating the cost of software quality. Communications of the ACM, 41(8), 67-73.*
- Software Productivity Analysis and Cost Estimation, Jacky Keung, 2007*
- Software Productivity Analysis Cost Estimation, Jacky Keung, 2007*
- Sommerville, I. 2007. Software Engineering, 8th Edition. Englanti. Addison-Wesley Publishers Ltd. 840 s.*

Stalk, G. (1988). Time--the next source of competitive advantage.

Suri, P. K., & Ranjan, P. (2012). Comparative analysis of software effort estimation techniques. *International Journal of Computer Applications*, 48(21), 12-19.

Symons, C. R. (1988). Function point analysis: difficulties and improvements. *IEEE transactions on software engineering*, 14(1), 2-11.

Tausworthe, R. C. (1979). The work breakdown structure in software project management. *Journal of Systems and Software*, 1, 181-186.

The Agile Manifesto. Viitattu 27.6.2019, osoitteesta <http://agilemanifesto.org/>

Tähtinen, J. & Parviainen, P. (2003). *Ohjelmistojen markkinointi. Teoksessa Hyvönen, E. (toim.), Ohjelmistoliiketoiminta (s. 41–76). Vantaa: WSOY.*

Wagner, S., Xie, S., Rübél-Otterbach, M., & Sell, B. 2007. *Profitability estimation of software projects: A combined framework. In The First International Workshop on Software Productivity Analysis and Cost Estimation (SPACE'07).*

Wallace, L. and Keil, M. (2004) 'Software Project Risks and their Effect on Outcomes', *Communications of the ACM*, 47(4), pp. 68–73

Woodside, A. G. 1994. *Making Better Pricing Decisions in Business Marketing. Advances in Business Marketing and Purchasing. Handbook of Business-To-Business Marketing Management 139-183. Greenwich, CN. JAI Press.*

Wysocki, R. K. 2011. *Effective project management: traditional, agile, extreme. John Wiley & Sons.*

LIITE A: KONSTRUKTIO (SALATTU)