

Kasper Jakonen

# OHJELMISTOJEN OMINAISUUKSIEN ARVONMÄÄRITYS JA ARVON HYÖDYN- TÄMINEN PÄÄTÖKSENTEOSSA

Diplomityö  
Tekniikan ja luonnontieteiden tiedekunta  
Professori Teemu Laine  
Professori Kari Systä  
Syyskuu 2020

# TIIVISTELMÄ

Kaspero Jakonen: Ohjelmistojen ominaisuuksien arvonmääritys ja arvon hyödyntäminen päätöksenteossa

Diplomityö  
Tampereen yliopisto  
Tuotantotalous  
09/2020

---

Ohjelmistojen ominaisuuksien arvonmäärityksellä ja arvoperusteisella päätöksenteolla on aikaisemmissa tutkimuksissa tunnistettu olevan merkittäviä positiivisia vaikutuksia ohjelmistoprojektien menestyksen kannalta. Tästä huolimatta empiiristen tutkimusten mukaan ohjelmistoyritykset hyödyntävät ominaisuuksien arvoa usein vain epäsuorasti tai keskittymällä vain tiettyihin arvon osa-alueisiin. Tässä diplomityössä tutkittiin kirjallisuudessa yleisimmin esiintyviä ohjelmistominisuuksien arvotekijöitä sekä ominaisuuksien arvonmäärityksen prosessia. Työssä tutkittiin myös ominaisuuksien arvonmäärityksen pohjalta tehtävän päätöksenteon nykytilaa.

Työ toteutettiin systemaattisena kirjallisuuskatsauksena tarkastellen aikaisempia tutkimuksia ohjelmisto-ominisuuksien arvottamisesta ja arvoperusteisesta päätöksenteosta. Työssä yhdisteltiin eri tutkimusten tuloksia tavoitteena kuvata kattavasti ominaisuuksien arvon hyödyntämisen nykytilaa sekä tunnistaa mahdollisia puutteita nykyisessä tutkimuksessa. Työn systemaattiset kirjallisuushaut kohdistettiin laadukkaaseen 2000-luvun tieteelliseen kirjallisuuteen.

Empiirisissä tutkimuksissa yleisimmin mainituiksi ohjelmistojen ominaisuuksien arvotekijöiksi työssä tunnistettiin loppukäyttäjätyytyväisyys, tuotteen kilpailukyky, markkinoillesaantiaika sekä kehityksen kokonaiskustannukset. Tutkimuksissa vähäisimmälle huomiolle jäivät yrityksen strategia -dimension alle kuuluvat arvotekijät, kuten ominaisuuksien vaikutus tuotteen strategiaan, yrityksen portfolioon tai yrityksen brändiin. Työssä tunnistettiin ominaisuuksien arvotekijöiden olevan kuitenkin vahvasti konteksti- ja yritysriippuvaisia, minkä vuoksi arvotekijöiden painoarvot ovat aina kullekin tilanteelle yksilöllisiä.

Ominaisuuksien arvonmäärityksestä tunnistettiin löytyvän hyvin vähän tieteellistä kirjallisuutta. Arvonmääritystä ei usein nähty kirjallisuudessa erillisenä toimintona, vaan ominaisuuksien arvottaminen oli usein kiinteä osa arvoperusteisia ominaisuuksien priorisointimenetelmiä. Arvonmäärityksen tunnistettiin olevan hyvin haastavaa useiden syiden takia, kuten ympäristön epävarmuuden ja arvon moniulotteisen luonteen takia. Tutkimusten mukaan yritysten päätöksenteon kannalta olisi suotuisinta, jos ominaisuuksien arvonmääritys voitaisiin toteuttaa taloudellisin termein rahamääräisenä ilmaistuna. Useiden arvon komponenttien, kuten aineettomien strategisten hyötyjen arvioiminen rahamääräisenä on kuitenkin hyvin haastavaa, minkä vuoksi tutkimusten mukaan käytännön ohjelmistotuotannossa käytetään yleisemmin erilaisia pisteytysmenetelmiä. Pisteytysmenetelmien heikkoutena on, että ne eivät kykene esittämään ominaisuuksien todellista arvoa, vaan ilmaisevat arvon suhteellisena arvona muihin verrattuna.

Nykypäivän nopeasti muuttuvassa liiketoimintaympäristössä ohjelmisto-ominisuuksien toteutukseen liittyviä päätöksiä tehdään jatkuvasti. Ominaisuuksien valikointi- ja priorisointimenetelmiä on kirjallisuudesta tunnistettavissa useita. Työssä tunnistettiin, että ominaisuuksien arvo käsitellään usein eri menetelmissä eri tavalla. Esimerkiksi joissain priorisointimenetelmissä ominaisuuden toteutuksen kustannuksia ja kiireellisyyttä käsitellään arvosta erillisenä tekijänä, kun taas toisissa menetelmissä ne luetaan mukaan kokonaisvaltaisempaan käsitykseen arvosta. Arvoperusteisiksi ominaisuuksien priorisointimenetelmiksi, joissa ominaisuuksien arvo on kirjallisuuden mukaan keskiössä, työssä tunnistettiin cost-value -menetelmä, EVOLVE\*-menetelmä, incremental funding method (IFM) sekä value-oriented requirements prioritization (VOP). Myös eri priorisointimenetelmien soveltuvuuden tunnistettiin olevan konteksti- ja yritysriippuvaisia.

Avainsanat: Arvoperusteinen ohjelmistotuotanto, VBSE, ohjelmisto-ominaisuus, arvonmääritys, ominaisuuksien priorisointi, priorisointimenetelmä

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

# ABSTRACT

Kasper J. Jakonen: Software feature valuation and value-based decision-making  
Master's Thesis  
Tampere University  
Industrial Engineering and Management  
09/2020

---

Previous research has found software feature valuation and value-based decision-making as critical success factors for software projects. Despite the previous fact, empirical studies suggest that software companies often exploit the value of features only indirectly or by focusing only on certain aspects of value. In this master's thesis value factors of software features most mentioned in the literature were examined, as well as the process of feature valuation. The thesis also examined the current state of decision-making based on the value of features.

The research was carried out as a systematic literature review, examining previous studies which focused on the valuation of software features and value-based decision-making. The results of different studies were combined to achieve a comprehensive understanding of the current state on the field and to identify possible shortcomings in the current study. The systematic literature searches of the research were targeted on high-quality 21<sup>st</sup> century scientific literature.

The most mentioned value factors of software features identified from empirical studies were end-user satisfaction, product competitiveness, time-to-market and total development costs. In the studies, the least attention was paid to the value factors under the company strategy dimension, including factors such as feature's impact on product's strategy, company's portfolio or company's brand. However, it was identified in the research that the value factors of software features were strongly context and company dependent, which is why the importance of the value factors are specific for each situation.

Very little scientific literature was found on the valuation of features. Valuation of software features was often not seen as a separate function, but as an integral part of value-based feature prioritization methods. Valuation was identified as a very challenging activity for several reasons, including environmental uncertainty and the multidimensional nature of value. According to research, it would be most favorable for companies if the valuation of features could be carried out in monetary terms. However, in case of several value factors, such as intangible strategic benefits, this is very challenging, which is why studies show that different scoring methods are more commonly used in practical software engineering. The weakness of scoring methods is that they are not able to represent the real value of features but express the value as a relative value instead.

In today's rapidly changing business environment, decisions related to the implementation of software features are made constantly. Several methods for selecting and prioritizing features can be identified from the literature. The research recognized that the value of features is often understood differently in different methods. For example, in some prioritization methods, the cost and urgency of implementing a feature are treated as a separate factor from value, while in other methods they are included in a more holistic view of value. The research identified cost-value method, EVOLVE\*, incremental funding method (IFM) and value-oriented requirements prioritization (VOP) as value-based feature prioritization methods, in which the value of the features has a central role. The suitability of different prioritization methods was also identified as context and company dependent.

Keywords: value-based software engineering, VBSE, software feature, valuation, feature prioritization, prioritization method

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

# ALKUSANAT

Idean diplomityön aiheelle sain Tampereen yliopiston tietotekniikan yksiköstä. Tuotantotalouden opintosuunnastani huolimatta olen koko opintojen ajan ollut myös teknisesti orientoitunut, minkä vuoksi valitsinkin ohjelmistotekniikan ja ohjelmistotuotannon kandidaattidiplomivaiheeni sivuaineiksi. Tämän diplomityön tutkimuskysymykset muodostuivatkin näiden kiinnostuksen kohteiden ympärille. Vaikka aihe muodostuikin paikoin haastavaksi ja systemaattisen kirjallisuuskatsauksen tutkimusmenetelmä hyvin työlääksi, olen tyytyväinen siihen, kuinka paljon olen oppinut aiheesta tätä työtä tehdessäni.

Haluaisin kiittää diplomityön ohjaajia professori Teemu Lainetta ja professori Kari Systää työhön liittyvistä neuvoista ja ehdotuksista. Kiitän myös Kari Systää työn aiheesta ja työn suunnasta. Kiitokset kuuluvat myös työ- ja opiskelukavereilleni, jotka ovat koko opintojen ajan tukeneet minua matkan varrella. En tule koskaan unohtamaan tuotantotalouden kilttaa Indeksiä, killan jäsenien kanssa kokemia huikeita hetkiä sekä lukuisia kiltahuoneella vietettyjä iltoja. Haluan kiittää myös Tampereen teknillistä yliopistoa, jonne opiskelemaan hakeminen oli elämäni parhaita päätöksiä.

Lopuksi haluaisin antaa erityiskiitokset vielä vanhemmilleni sekä avopuolisolleni Sofialle, jotka ovat tukeneet ja kannustaneet minua koko opintojen ja tämän diplomityöprojektin ajan. Kiitän heitä myös työn oikolukuavusta.

Tampereella, 26.8.2020

Kasper Jakonen

# SISÄLLYSLUETTELO

1. JOHDANTO .....	1
1.1 Tutkimuksen motivaatio.....	2
1.2 Tavoite ja tutkimuskysymykset.....	3
1.3 Työn rakenne .....	4
2. TUTKIMUSMETODOLOGIA .....	6
2.1 Systemaattinen kirjallisuuskatsaus.....	6
2.2 Hakuprosessi ja hakutulokset.....	8
2.3 Tutkimusten valinta .....	11
3. ARVO OHJELMISTOTUOTANNOSSA .....	13
3.1 Arvon määritelmä ja arvoperusteisen ohjelmistotuotannon motiivit.....	13
3.2 Ohjelmiston ominaisuuksien määritelmä .....	16
3.3 Ohjelmiston ominaisuuksien arvotekijät.....	17
4. ARVON MÄÄRITTÄMINEN.....	23
4.1 Arvonmääritys ohjelmistotuotannossa .....	23
4.2 Arvonmäärityksen haasteet.....	27
4.2.1 Epävarmuus.....	28
4.2.2 Sidosryhmien eroavat näkemykset arvosta .....	28
4.2.3 Arvon moniulotteinen luonne.....	30
4.3 Ohjelmisto-ominaisuuksien arvonmääritys .....	31
4.4 Nykyiset ominaisuuksien arvonmääritysmenetelmät .....	33
4.4.1 Yleisiä ominaisuuksien arvonmääritysmenetelmiä .....	33
4.4.2 Ominaisuuden käyttöasteeseen perustuva arvonmääritys .....	36
5. OMINAISUUKSIEN ARVON HYÖDYNTÄMINEN PÄÄTÖKSENTEOSSA.....	39
5.1 Inkrementaalinen ohjelmistokehitys ja julkaisusuunnittelu .....	39
5.2 Ominaisuuksien valinta ja priorisointi arvoperusteisesti .....	42
5.3 Ominaisuuksien arvoperusteiset priorisointimenetelmät.....	45
5.3.1 Cost-value.....	46
5.3.2 EVOLVE* .....	47
5.3.3 Incremental Funding method (IFM) .....	48
5.3.4 Value-Oriented Requirements Prioritization (VOP).....	49
6. PÄÄTELMÄT.....	51
LÄHTEET .....	54
LIITE A: TAULUKON 2 ARVOTEKIJÖIDEN KUVAUKSET .....	61

# LYHENTEET JA MERKINNÄT

AHP	Analytical Hierarchy Process
BST	Binary Search Tree, binäärinen hakupuu
DOM	Document Object Model, dokumenttioliomalli
IEEE	Institute of Electrical and Electronics Engineers, kansainvälinen tekniikan alan järjestö
IFM	Incremental Funding Method
MoSCoW	Must have, Should have, Could have, Won't have
QFD	Quality Function Deployment
TK	Tutkimuskysymys
VBSE	Value-Based Software Engineering, arvoperusteinen ohjelmistotuotanto
VOP	Value-oriented prioritization

# 1. JOHDANTO

Nykyisessä nopeasti muuttuvassa ohjelmistotuotannon kilpailukentässä suurin osa ohjelmistotuotteista kehitetään ja toimitetaan asiakkaille jatkuvien iteratiivisten julkaisujen kautta (Greer & Ruhe 2004; Tourwé et al. 2009). Selvitäkseen kilpailuilla markkinoilla ja vastatakseen asiakastarpeisiin ohjelmistoyritykset lisäävät yhä enemmän ominaisuuksia (feature) ohjelmistoihinsa, mikä kasvattaa ohjelmistojen kokoa ja monimutkaisuutta (Elliott 2007). Samalla kun ohjelmistojen koko ja monimutkaisuus kasvavat, kasvavat myös jokaisen ohjelmistoa koskevan päätöksenteon vaikutus koko yrityksen tarjontaan (Khurum et al. 2013). Ominaisuuksien valintoihin ja priorisointeihin koskevaa päätöksentekoa toteutetaan etenkin julkaisusuunnittelun (release planning) yhteydessä. Julkaisusuunnittelulla on merkittävä vaikutus ohjelmistotuotteiden kehityksen onnistumisen tai epäonnistumisen kannalta (Mc Elroy & Ruhe 2010).

Lukuisat tutkimukset osoittavat, että ensisijainen menestystekijä, joka erottaa onnistuneet ohjelmistotuotteet ja -projektit epäonnistuneista löytyy arvoperusteisen ajattelun alueelta (Biffel et al. 2006, s. 3). Boehmin (2003b) mukaan perinteisesti harjoitetun arvoneutraalin ohjelmistotuotannon periaatteet eivät pysty käsittelemään useimpia ohjelmistoprojektien epäonnistumisen lähteitä, kuten puutteellisia tai muuttuvia asiakasvaatimuksia, resurssipuutteita tai epärealistisia odotuksia tai aikatauluja. Perinteisessä arvoneutraalissa ympäristössä jokaista vaatimusta, käyttötapausta, tavoitetta ja vikaa käsitellään arvoltaan yhtä tärkeinä (Boehm 2003b). Säilyttääkseen kilpailukykyä, innovatiivisuutensa ja kasvaakseen yritysten tulisi hyödyntää arvoperusteista päätöksentekoa, jossa päätökset optimoidaan yrityksen kokonaisarvonluonnin suhteen (Freitas et al. 2017). Vaikka ominaisuuksien valinnalla ja priorisoinnilla on todettu olevan merkittävä vaikutus ohjelmistotuotteiden menestyksen kannalta, hyvin harvat tutkimukset ovat keskittyneet arvoon kyseisessä päätöksenteon kontekstissa (Wohlin & Aurum 2005; Rodriguez et al. 2018). Koko ohjelmistotuotteen arvo koostuu julkaisuille valittujen ominaisuuksien joukosta, joten ymmärrys ominaisuuksien arvon merkityksestä päätöksenteon yhteydessä on yrityksille tärkeää (Rodriguez et al. 2018).

Ominaisuuksienhallintaprosessin kriittisimpiä osia on sopivien ominaisuuksien osajoukon valinta ja priorisointi eri sidosryhmien kilpailevien ja ristiriitaisten odotusten joukosta (Wieggers 1999). Ohjelmiston liiketoiminnallista arvoa tuovien ominaisuuksien lukumää-

rän on todettu olevan pienempi kuin ominaisuuksien yhteenlasketun lukumäärän. Bullockin (2000) mukaan 80 prosenttia ohjelmiston liiketoiminta-arvosta tulee vain 20 prosentista ohjelmisto-ominaisuuksia. Ohjelmistoyritysten tulisi uusien ominaisuuksien toteutuksen arvioinnissa ottaa huomioon kokonaisvaltainen näkemys ominaisuuksien arvosta, joka ottaa huomioon sekä lyhyen että pidemmän aikavälin tavoitteet. Kyseiset tavoitteet vaihtelevat sidosryhmien välillä, ja ominaisuuksien arvon arvioinnissa tulisi ottaa huomioon sidosryhmien eriävät näkemykset arvosta. (Mendes 2018) Ohjelmiston ominaisuuksien arvon arviointi on kuitenkin haastavaa, sillä asiakkaan käsitys ominaisuudesta ei ole staattinen, vaan vaihtelee ajan myötä. Kerran erittäin arvokkaaksi koettu ominaisuus voi ajan myötä menettää arvoaan tai arvo voi jopa mitätöityä kokonaan. Tarpeettomat tai arvoa liian vähän tuottavat toiminnot eivät pelkästään kuluta turhaan resursseja kuten työvoimaa ja aikaa, vaan voivat myös laskea suorituskykyä, vaatia suurempia laitteistoja ja lisätä ylläpitokustannuksia. (Marciuska et al. 2014)

## 1.1 Tutkimuksen motivaatio

Huolimatta arvon konseptin tärkeydestä ohjelmistotuotannossa, arvon käsitettä ei vielä ole yksityiskohtaisemmin ymmärretty yrityksissä (Rodríguez et al. 2018). Vaikka eri komponenttien arvon määrittäystä pidetään usein kiinteänä osana ohjelmistotuotannon päätöksentekoa, arvon määrittelmä on usein dokumentoitu huonosti ja sitä harvoin hyödynnetään selkeällä tavalla edes arvoa korostavissa kehitysprosesseissa (esim. ketterät menetelmät) (Yanzer Cabral et al. 2014). Tämän seurauksena päätöksenteko perustuu usein ainoastaan epäsuoriin ja päätöksentekijän intuitioon perustuviin arvon määrittäisiin, joihin liittyy paljon epäselvyyttä ja epämääräisyyttä. Näin ollen kehitysprosessin komponenttien arvo usein määritellään riittämättömin ja puutteellisin perustein, jolloin päätöksenteko pohjataan epätehokkaaseen tietoon, joka usein johtaa ristiriitaisiin näkymiin eri sidosryhmien keskuudessa. (Biffel et al. 2006; Mc Elroy & Ruhe 2010)

Rodríguez et al. (2018) mukaan lukuisat ohjelmistoalan yritykset ovat omaksumassa arvoperusteista päätöksentekoa uusien ominaisuuksien valitsemisessa. Arvoperusteiset päätöksentekomekanismit yhdistettynä ohjelmistotuotteiden kehitykseen tavoittelevat kasvun ja kilpailukyvyn ylläpitoa (Mohamed & Wahba 2008). Lisäksi toteutettavien ohjelmisto-ominaisuuksien arvoa arvioimalla voidaan vähentää ylläpitokustannuksia sekä lisätä järjestelmän suorituskykyä karsimalla toteutettavien ominaisuuksien listalta marginaalista arvoa tuottavia ominaisuuksia (Marciuska et al. 2013a). Mendes et al. (2017) ajankohtaisen tutkimuksen mukaan siitä huolimatta, että ohjelmistotuotannossa on kasvavissa määrin alettu omaksua arvoperusteista lähestymistapaa monissa päätöksente-



kotilanteissa, perustuvat päätökset kuitenkin hyvin usein vielä avainsidosryhmien hiljaiseen tietoon, eikä apuna käytetä työkaluja tai arvon arviointia. Ohjelmistojen ominaisuuksien arvonmäärittäminen on todettu monessa alan julkaisussa olevan haastavaa. Biffi et al. (2006, s. 39) mukaan suurin haaste ohjelmistokomponenttien arvon määrittämisessä on epävarmuus. Tämän lisäksi Marciuska et al. (2013b) mainitsevat kaksi muuta ensisijaisista haastetta arvonmäärittämisessä: arvon moniulotteisen luonteen sekä sidosryhmien eroavat näkemykset arvosta.

Jan & Ibrarin (2010) systemaattisen kartoitustutkimuksen mukaan nykyinen ohjelmistotuotannon tutkimus on usein arvonäkökohtien osalta toisistaan eristäytynyttä ja keskittyy usein vain yhteen tiettyyn asiaan, kuten kehityskustannuksiin tai tuotteen ominaispiirteisiin, kuten käytettävyyteen. Khurum et al. (2013) kehittivät tutkimuksessaan ohjelmiston arvokartan (software value map, SVP), jonka avulla tutkimuksessa tunnistettiin 62 eri ohjelmistojen arvoon vaikuttavaa tekijää ottaen huomioon myös eri sidosryhmien arvonäkemykset. Kyseinen tutkimus on kuitenkin toteutettu hyvin yleisellä tasolla, jonka vuoksi sitä on haastava soveltaa yksityiskohtaisemmassa päätöksentekotilanteessa, kuten ohjelmiston ominaisuuksien valinnassa.

## 1.2 Tavoite ja tutkimuskysymykset

Tässä tutkimuksessa tarkastellaan ohjelmistojen ominaisuuksien arvotekijöitä, arvonmäärittäystä sekä ominaisuuksien arvon hyödyntämistä päätöksenteossa. Tutkimus suoritetaan systemaattisena kirjallisuuskatsauksena, jonka avulla tavoitteena on tutkia nykyisiä arvoperusteisen ominaisuuksienhallinnan käytäntöjä ja menetelmiä. Tutkimuksen toisena tavoitteena on tunnistaa puutteita nykyisessä arvoperusteisen ominaisuuksienhallinnan tutkimuksessa. Diplomityön tutkimuskysymykset (TK) ovat seuraavat:

- TK1. Millaisia ominaisuuksien arvotekijät ovat?
- TK2. Miten ohjelmistojen ominaisuuksien arvo määritetään?
- TK3. Miten tietoa ominaisuuksien arvosta hyödynnetään päätöksenteossa?

Jokaiseen tutkimuskysymystä varten työssä on tavoitteena tunnistaa mahdollisimman monipuolisesti olemassa oleva tutkimuskysymysten kannalta oleellinen tieteellinen kirjallisuus. Tämän vuoksi tutkimusmenetelmäksi on valittu systemaattinen kirjallisuuskatsaus, jota käydään tarkemmin läpi työn toisessa luvussa. Ensimmäiseen tutkimuskysymykseen vastaamiseksi työssä etsitään kattavasti kirjallisuudesta empiirisiä tutkimuksia, jotka käsittelevät ohjelmistojen ominaisuuksien arvotekijöitä. Toisessa tutkimuskysymyk-

sessä käsitellään ohjelmistojen ominaisuuksien arvonmäärittystä ja sen haasteita. Arvonmäärittys on usein integroitu osa ominaisuuksien priorisointimenetelmiä, mikä tekee toisen ja kolmannen tutkimuskysymyksen erottelusta hieman haastavaa. Kolmannessa tutkimuskysymyksessä käsitellään ominaisuuksien arvoperusteisia valikointi- ja priorisointimenetelmiä. Kirjallisuudessa ominaisuuksien valikointi- ja priorisointimenetelmistä puhutaan usein samana asiana. Ominaisuusvaatimusten selvitysprosessi (feature elicitation) on edellisiä päätöksentekotilanteita edeltävä vaihe, joka on rajattu tämän työn tarkastelun ulkopuolelle. Toinen ja kolmas tutkimuskysymys ovat ensimmäistä tutkimuskysymystä yleisemmän tason kysymyksiä, ja työn tavoitteena onkin kuvata kyseisten käytäntöjen nykytilaa ja löytää mahdollisia puutteita kirjallisuudesta, eikä esimerkiksi tarjota tarkkaa ohjekirjaa eri priorisointimenetelmien käytöstä.

Työn tutkimustulokset ovat sekä tieteellisesti että käytännöllisesti katsoen tärkeitä niin suomalaisille kuin kansainvälisillekin ohjelmistoalan yrityksille, jotka etsivät tehokkaampia ja luotettavampia tapoja ominaisuuksien arvon arviointiin päätöksenteon tueksi. Tieteellisesti työ tarjoaa yleiskuvan nykypäivän arvoperusteisesta ohjelmisto-ominaisuuksien hallinnasta. Käytännölliseltä kannalta työn kontribuutio on hyödyllistä erityisesti ohjelmistojen ominaisuuksien toteutuksista ja priorisoinnista vastaaville henkilöille, jotka voivat hyödyntää työssä toteutettua arvoperusteista selvitystä päätöksenteossa. Tällaisia henkilöitä ovat tyypillisesti projektipäälliköt tai yhä yleistyvässä ketterässä ohjelmistokehityksessä eri nimikkeillä toimivat ominaisuuslistoja hallinnoivat työntekijäroolit (Esim. Scrum-kehityksessä Scrum Masterit).

### **1.3 Työn rakenne**

Tämä diplomityö on jaettu kuuteen pääluukuun. Johdannon jälkeen toisessa luvussa käsitellään työn tutkimusmenetelmäksi valittua systemaattisen kirjallisuuskatsauksen prosessia. Kolmannessa luvussa tarkastellaan arvon käsitettä ohjelmistotuotannossa sekä arvoperusteisen ohjelmistotuotannon motiiveja. Luvun päätteeksi käsitellään ohjelmisto-ominaisuuksien arvotekijöitä. Neljännessä luvussa tarkastellaan arvonmäärittystä sekä arvonmäärittymisen haasteita ensin ohjelmistotuotannon tasolta. Tämän jälkeen luvussa keskitytään ohjelmistojen ominaisuuksien arvonmäärittymiseen sekä nykyisiin ominaisuuksien arvonmäärittymismenetelmiin. Työn viidennessä luvussa keskitytään tarkastelemaan arvon merkitystä ohjelmisto-ominaisuuksiin liittyvässä päätöksenteossa. Luvussa esitellään ensin inkrementaalisen ohjelmistokehityksen sekä julkaisusuunnittelun käsitteet, joiden kontekstissa ominaisuuksiin liittyviä päätöksiä nykyisessä jatkuvasti muuttuvassa liiketoimintaympäristössä perinteisesti tehdään. Tämän jälkeen luvussa keskity-

tään ominaisuuksien arvoperusteiseen valintaan ja priorisointiin sekä nykyisiin priorisointimenetelmiin, joissa kirjallisuus on tunnistanut arvon olevan keskiössä. Kuudennessa luvussa esitetään työn päätelmät sekä aiheet jatkotutkimukselle.

## 2. TUTKIMUSMETODOLOGIA

Tässä luvussa käsitellään tutkimuksessa käytettyä tutkimusmetodologiaa ja tutkimusprosessia. Luvussa esitetään ensin työn tutkimusmenetelmäksi valitun systemaattisen kirjallisuuskatsauksen periaatteet ja motiivit. Tämän jälkeen luvussa kuvataan työn tutkimusprosessin vaiheet sekä niiden toteuttaminen.

### 2.1 Systemaattinen kirjallisuuskatsaus

Systemaattinen kirjallisuuskatsaus on kirjallisuuden tarkastelun prosessi, jossa hyödynnetään kattavaa ennakkoon suunniteltua strategiaa olemassa olevan kirjallisuuden löytämiseksi, arvioimiseksi, analysoimiseksi ja yhdistämiseksi sekä raportoimiseksi. Systemaattisen kirjallisuuskatsauksen tavoitteena on luoda mahdollisimman järjestelmällinen ja läpinäkyvä tutkimusprosessi. Systemaattinen kirjallisuuskatsaus on lähtöisin lääketieteen alalta, mutta viimeisten vuosikymmenien aikana sen merkitys on kasvanut myös muilla tieteenaloilla. (Denyer & Tranfield 2009)

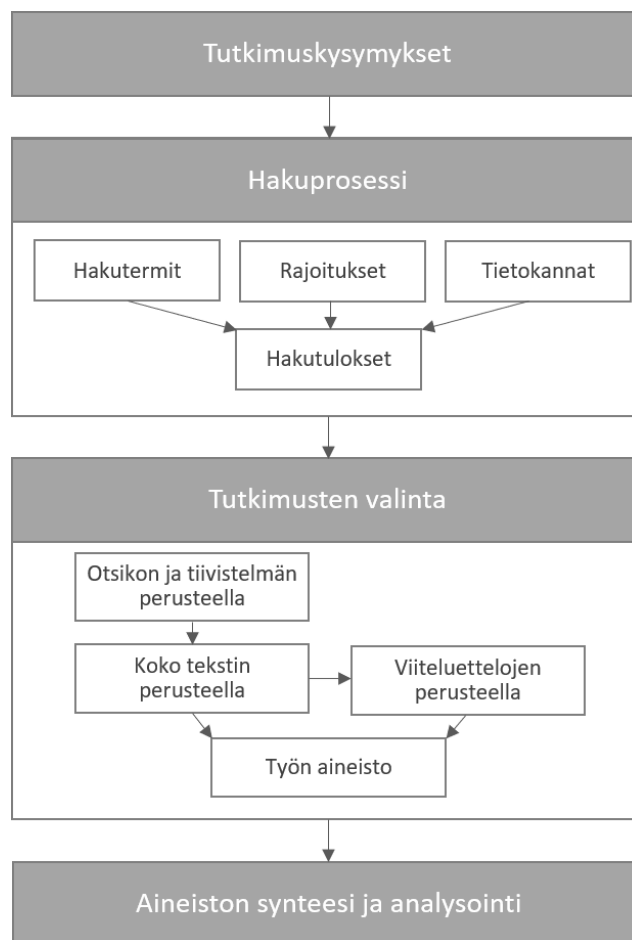
Systemaattinen kirjallisuuskatsaus ohjelmistotuotannon osa-alueella on keino tunnistaa, arvioida ja tulkita kaikki käytettävissä olevat tutkimukset, jotka liittyvät tiettyyn tutkimuskysymykseen, aihepiiriin tai ilmiöön (Kitchenham & Charters 2007). Ohjelmistotuotannon alalla systemaattisen kirjallisuuskatsauksen yleisen tason pääasiallisia hyötyjä ovat vääristymien väheneminen sekä katsauksen laadun ja luotettavuuden paraneminen (Stapić et al. 2012). Kuten muillakin tieteenaloilla, Budgenin ja Breretonin (2006) mukaan myös ohjelmistotuotannon alalla yleisiä syitä systemaattisen kirjallisuuskatsauksen toteuttamiseen ovat seuraavat tavoitteet:

- Tehdä yhteenveto nykyisistä käytännöistä tai menetelmistä
- Tunnistaa puutteet nykyisessä tutkimuksessa, jotta voidaan selvittää missä tarvitaan mahdollisesti lisätutkimusta
- Auttaa asettelemaan uutta tutkimustoimintaa
- Tutkia, kuinka pitkälle tiettyä hypoteesia tuetaan saatavilla olevien empiiristen todisteiden perusteella

Tämä tutkimus keskittyy edellisistä kohdista kahteen ensimmäiseen. Tutkimuksen pää-tavoitteena on tutkia asetettujen tutkimuskysymysten mukaisia nykyisiä arvoperusteisen ominaisuuksienhallinnan käytäntöjä ja menetelmiä. Tutkimuksen toisena tavoitteena on tunnistaa puutteita nykyisessä arvoperusteisen ominaisuuksienhallinnan tutkimuksessa.

Summaamalla nykyistä tutkimusta yhteen työssä tavoitellaan tietoa siitä, mitä aiheesta tällä hetkellä tiedetään ja mitä ei vielä tiedetä.

Tässä työssä systemaattisen kirjallisuuskatsauksen pohjana käytetään Stapić et al. (2012) esittelemää ohjelmistotuotannon alaan soveltuva kevyemmän systemaattisen kirjallisuuskatsauksen mallia. Kyseinen systemaattisen kirjallisuuskatsauksen malli soveltuu myös yksittäisen tutkijan toteutettavaksi. Työn tutkimusprosessi koostuu neljästä eri vaiheesta: tutkimuskysymysten muodostamisesta, hakuprosessin toteuttamisesta ennalta määritellyn hakustrategian mukaisesti, tutkimusten kriittisestä valinnasta sekä aineiston synteesisistä ja analysoinnista. Hakuprosessin ja tutkimusten valinnan jokainen vaihe dokumentoidaan systemaattisen kirjallisuuskatsauksen periaatteiden mukaisesti mahdollisimman läpinäkyvästi. Hakuprosessia ja tutkimusten valintaa kuvataan yksityiskohtaisemmin seuraavissa alaluvussa. Kuvassa 1 on esitelty työn tutkimusprosessin vaiheet.



**Kuva 1.** Tutkimusprosessin vaiheet

## 2.2 Hakuprosessi ja hakutulokset

Ennen varsinaisten hakusanojen muodostamista tutkimuskysymyksiä varten työssä tehtiin lukuisia iteraatioita käyttämällä muun muassa Google Scholarin keskeisten käsitteiden tunnistamiseen. Esimerkiksi ensimmäistä tutkimuskysymystä varten arvotekijöiden semantiikan tunnistamiseksi luetuista tieteellisistä artikkeleista muodostui kuva, että ohjelmistotuotannossa arvon tekijöistä käytettiin hyvin yleisesti englannin kielen termiä ”value proposition”. Lukemalla Biffel et al. (2006) arvoperusteisen ohjelmistotuotannon kirjan, huomattiin että arvon tekijöille käytettiin myös termejä ”value factors” ja ”value components”. Synonyymien ja hakukäsitteiden tunnistamiseksi työssä käytettiin myös Saunders et al. (2019, s. 92–94) suosittamana sanakirjoja sekä tietosanakirjoja.

Artikkeleita pidemmälle lukemalla ja seuraamalla ohjelmistojen ominaisuuksia käsittelevien tieteellisten kirjoitusten viittausketjuja huomattiin myös, että ominaisuuksista (feature) ja vaatimuksista (requirement) kirjoitettiin hyvin usein synonyymeina. Tämän vuoksi oleellista tutkimusta aiheesta olisi jäänyt pois ilman vaatimusten mukaan ottamista hakutermeihin kaikkien tutkimuskysymysten hakujen osalta. Tämän työn pääfokus on kuitenkin vaatimuksia hieman korkeamman tason ominaisuuksien käsitteessä, jota käytetään yleisesti päätöksenteon yhteydessä. Tämän vuoksi kirjoittajan toimesta on kunkin työhön mukaan otetun vaatimuksia painottavan tutkimuksen kohdalla arvioitu tarkkaan, että vaatimuksia joko käsitellään synonyymeina ominaisuuksille tai vaatimuksia käsitellään sen verran korkealla tasolla, että samat käytännöt ovat sovellettavissa myös ominaisuuksiin.

Hakutermien muodostuksessa käytettiin Boolean operaattoreita ’AND’ rajaamaan saatujen tulosten joukkoa sekä ’OR’ laajentamaan joukkoa ottamalla huomioon lähinnä synonyymit, jotta oleellisia julkaisuja ei jäisi hakutuloksista pois. Rinnakkaiskäsitteiden tai synonyymien haun kohdalla kyseisten termien ympärillä käytettiin sulkeita, jotta ’OR’-operaattoria hallitsevampi ’AND’-operaattori ei rajaisi hakua tarkoituksen vastaiseksi. Feature selection -hakutermien tapauksessa käytettiin myös Boolean operaattoria ’NOT’ rajaamaan pois tulokset koneoppimisen ja tilastotieteen aloilta, jotka käyttävät samaa termiä tämän työn kannalta epärelevantissa tarkoituksessa. Hauissa käytettiin myös katkaisumerkkiä ’\*’, jotta saatiin ominaisuuksista ja vaatimuksista myös monikkomuodot hakutuloksiin mukaan.

Tieteellisen tekstin hakupalveluiksi valittiin kattavasti useista eri tietokannoista julkaisuja hakevat Elsevierin Scopus sekä Tampereen yliopiston kirjaston hakupalvelu Andor. Hakutuloksia käsitellessä huomattiin, että hyvin suuri osa tutkimuskysymykselle relevantista kirjallisuudesta on julkaistu Institute of Electrical and Electronics Engineersin (IEEE)

kirjastossa, joten hakuja toteutettiin myös IEEE:n tietokantaan. Näin löydettiin esimerkiksi ensimmäiselle tutkimuskysymykselle oleellinen Rodriguez et al. (2020) julkaisu, joita aikaisemmat tietokannat eivät vielä indeksoineet hakutuloksissaan. Hakuja kokeiltiin myös ScienceDirectin sekä ACM Digital Libraryn tietokantoihin, mutta selaamalla hakutuloksena saatuja artikkeleita todettiin, että kyseiset tietokannat antoivat paljon sekä jo aikaisemmin löydettyjä tutkimuksia että kontekstiltaan tutkimuksen kannalta epärelevanttejä tuloksia, joten kyseiset tietokannat päätettiin jättää hakujen ulkopuolelle.

Tietoa haettiin ainoastaan englanninkielisestä kirjallisuudesta ja hakutulokset rajoitettiin ainoastaan 2000-luvun julkaisuihin. Hakujen rajaukseen vain 2000-luvun julkaisuihin on kaksi syytä. Arvoperusteinen ohjelmistotuotanto (value-based software engineering, VBSE), joka ottaa arvon kokonaisvaltaisemmin huomioon, syntyi Boehmin (2003b) aloitteesta vuonna 2003. Ottamalla hakuihin mukaan vain 2000-luvun julkaisut varmistetaan, että saadaan mahdollisimman ajankohtaista tietoa arvoperusteisesta ohjelmistotuotannon tutkimuksesta. Toisena pääsyyinä vuosilukurajoitteeseen on hakutulosten joukon rajaaminen kohtuulliseksi, jotta jokainen hakutuloksen julkaisu voidaan käydä läpi tarvittavalla tarkkuudella. Työhön on kuitenkin otettu mukaan myös muutamia ennen 2000-lukua julkaistuja alkuperäislähteitä, kun hakutulosten julkaisuja lukemalla on huomattu työn kannalta oleellisten näkökulmien alkuperän sijoittuvan aikaisempaan tutkimukseen.

Kaikki Scopusin ja Andorin hakutulokset rajoitettiin vertaisarvioituun tieteelliseen kirjallisuuteen, jonka avulla varmistettiin hakutuloksena saatujen artikkelien laatua sekä rajattiin hakutulosjoukkoa. Scopus tarjoaa lähtökohtaisesti ainoastaan vertaisarvioitua tieteellistä kirjallisuutta, mutta esimerkiksi Andorissa ensimmäisen tutkimuskysymyksen haun hakutulosten joukko rajautui 597:sta 17 hakutulokseen, kun mukaan otettiin ainoastaan vertaisarvioidut julkaisut. Andoria ei kuitenkaan käytetty esimerkiksi ominaisuuksien arvonmääritykseen liittyvän kirjallisuuden hauissa, koska hakujoukkoa ei saatu rajattua riittävän pieneksi. Hakutulosten joukko oli Andorissa usein muita tietokantoja suurempi, koska Andorissa hakuja ei voitu toteuttaa muiden tietokantojen tapaan ottamaan huomioon vain otsikon, tiivistelmän ja avainsanat. IEEE:n tietokannan kohdalla hakuja ei voitu rajoittaa vain vertaisarvioituihin artikkeleihin, joten hakutulosten joukkoon päätyi myös vertaisarvioimattomia artikkeleita ja konferenssijulkaisuja. Näiden, kuten muidenkin työhön otettujen julkaisujen laatua arvioitiin kriittisesti tarkastelemalla julkaisujen kirjoitusasua sekä kyseisiin julkaisuihin kohdistuvia viittausmääriä. Lisäksi hakutulosten rajaamiseksi työn kannalta oleelliseen kirjallisuuteen hauissa käytettiin myös aihealuerajauksia. Pääasiassa haut rajattiin aina tietojenkäsittelytieteen (computer science) alalle tai tarkemmin edellisen alle kuuluvaan ohjelmistotuotannon (software engineering) alalle.

Aihealuerajaukset on esitetty tarkemmin hakutermien ja hakutulosten yhteydessä taulukossa 1.

**Taulukko 1. Kirjallisuuskatsauksen hakutulokset**

Hakutermit	Tietokanta	Vuodet	Aihealue	Hakukentät	Hakutulokset
(feature* OR requirement*) AND ("value proposition" OR "value factors" OR "value components")	Scopus	2000-2020	Computer science	Otsikko, tiivistelmä, avainsanat	201
	Andor	2000-2020	Software Engineering	Kaikki kentät	98
		2000-2020	Computer Software Industry & Software	Kaikki kentät	17
	IEEE Xplore	2000-2020	Kaikki	Otsikko, tiivistelmä, avainsanat	80
software AND (feature* OR requirement*) AND ("valuation" OR "value estimation" OR "value assessment" OR "value evaluation" OR "value prediction")	Scopus	2000-2020	Computer science	Otsikko, tiivistelmä, avainsanat	109
	IEEE Xplore	2000-2020	Kaikki	Otsikko, tiivistelmä, avainsanat	124
software AND ("feature selection" OR "feature prioritization" OR "requirement prioritization") AND (value OR value-based) AND NOT ("data mining" OR "dataset*")	Scopus	2000-2020	Computer Science	Otsikko, tiivistelmä, avainsanat	197
(feature* OR requirement*) AND (value OR value-based) AND (selection OR prioritization)	Scopus	2000-2020	Computer Science	Otsikko	66
<b>Yhteensä</b>					<b>892</b> (duplikaatit mukana)



Artikkelien lukemisen jälkeen toteutettiin vielä muutamia iteraatioita, lisäämällä hakutermeihin tutkimuksissa havaittuja oleellisia synonyymeja sekä avainsanoja. Taulukossa 1 esitetyt hakutermit esittävät lopullisia, iteraatiokierrosten jälkeen tutkimuskysymyksille oleellisimmiksi osoittautuneita hakutermejä. Hakutulosten joukko rajattiin aina niin pieneksi, että jokaiseen hakutuloksen tutkimukseen pystyttiin perehtymään tiivistelmätasolla. Esimerkiksi taulukon 1 esittämällä tavalla viimeisiä hakuja rajattiin lisäämällä hakutermeihin sana 'software', jonka tuli esiintyä joko tutkimuksen otsikossa, tiivistelmässä tai avainsanoissa. Kyseisen haun aihealue oli jo kuitenkin rajattu tietojenkäsittelytieteen alle, eikä jokaisessa työn kannalta oleellisessa tutkimuksessa välttämättä mainittu edes tiivistelmässä sanaa 'software'. Tämän vuoksi suoritettiin vielä uusi haku ilman kyseistä avainsanaa, rajaamalla haku ainoastaan tutkimusten otsikkoon.

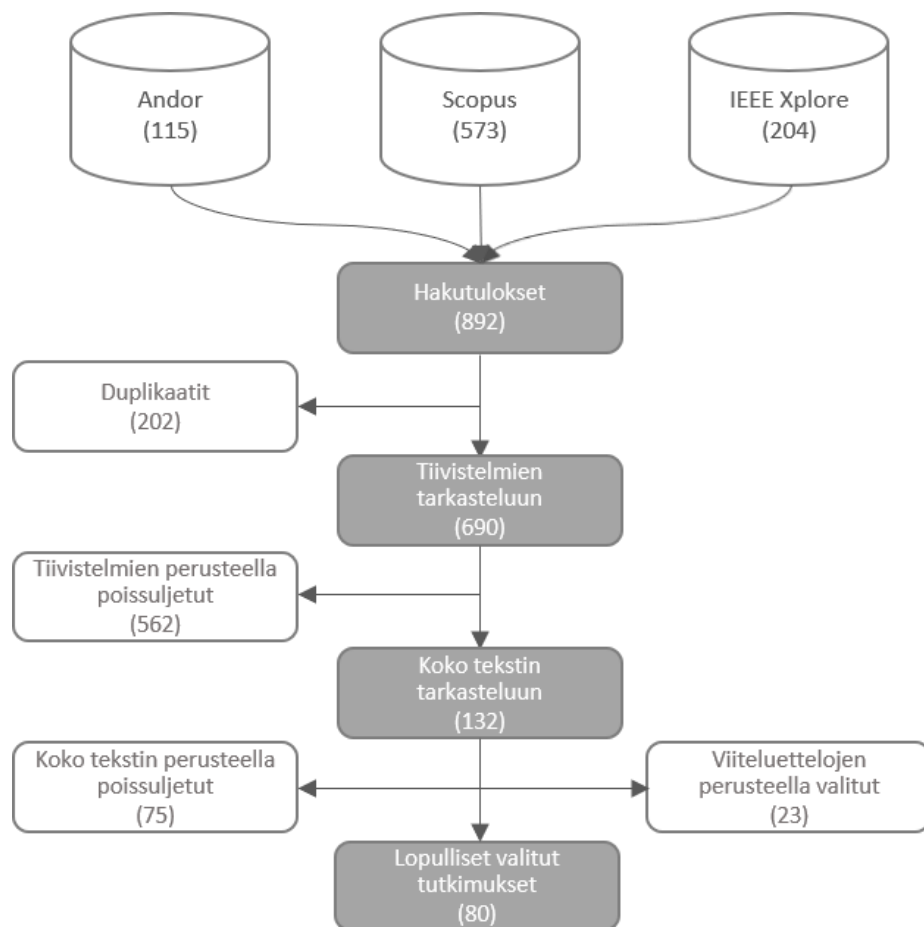
### **2.3 Tutkimusten valinta**

Suurimmasta osasta taulukon 1 esittämällä tavalla saaduista hakutulostuloksista luettiin aluksi otsikon lisäksi tiivistelmä. Joissain tapauksissa kuitenkin tutkimukset voitiin poisulkea työn kannalta epäolennaisina jo otsikon perusteella tai aikaisessa vaiheessa tiivistelmää luettaessa, kun huomattiin että artikkelit käsitelivät esimerkiksi ohjelmistojen sijaan laitteistoja (hardware). Yleinen syy julkaisujen poissulkemiseen oli arvonäkökoh- tien puuttuminen ominaisuuksiin liittyvän päätöksenteon kirjallisuudessa. Monet julkaisut myös liittyivät yleisellä tasolla arvoperusteiseen ohjelmistotuotantoon, ilman tarkempaa fokusta tämän työn painopisteenä oleviin ohjelmistojen ominaisuuksiin. Näissä tapauksissa oli kuitenkin hyvin haastavaa tiivistelmän perusteella tehdä tulkintoja tutkimusten sisällöstä, joten yleensä korkeammalla tasolla arvoperusteista ohjelmistotuotantoa käsittelevät julkaisut hyväksyttiin tässä vaiheessa mukaan seuraavaan vaiheeseen.

Työn hakutermit olivat toisiaan täydentäviä ja jokaista hakutulosjoukkoa peilattiin koko työn tavoitteisiin ja tutkimuskysymyksiin. Tämä tarkoittaa sitä, että jokaisesta hakutulosjoukosta löydettiin oleellisia tutkimuksia myös työn eri kohtiin, kuin mihin tutkimuskysymyksestä johdettu hakutermin viittasi. Moni tutkimus osoittautui hyödylliseksi myös useamman tämän työn tutkimuskysymyksen kannalta. Hakuprosessissa käytettyjen laaturajausten (esim. vain vertaisarvioidut artikkelit) lisäksi työhön mukaan otettujen julkaisujen laatua arvioitiin niiden tieteellisten tutkimusmenetelmien sekä viittaussäärien avulla.

Taulukon 1 mukaisia hakutermejä käyttämällä saatiin tuloksena 892 julkaisua. Duplikaattien poiston jälkeen jäljelle jäi 690 julkaisua. Tiivistelmien lukemisen jälkeen tutkimukselle oleelliseksi kirjallisuudeksi valikoitui 132 tutkimusta. Kyseisestä joukosta valikoitui

koko tekstin lukemisen jälkeen vielä 75 artikkelia pois kyseiselle tutkimukselle epärelevantteina, jättäen jäljelle 57 tutkimuskysymyksille oleellista julkaisua. Jotta saatujen hakutulosten ulkopuolelta jäisi mahdollisimman vähän työn kannalta oleellisia julkaisuja työn ulkopuolelle, käytiin viimeisessä tutkimusten valinnan vaiheessa läpi vielä otsikkotasolla jokaisen työhön valitun julkaisun viiteluettelot. Viiteluetteloiden perusteella löytyi vielä 23 työn kannalta oleellista tutkimusta. Lopulta työhön valikoitui systemaattisen prosessin jälkeen 80 tutkimusta. Kuva 2 esittää työn tutkimusten valintaprosessin.



**Kuva 2.** Tutkimusten valintaprosessi

### 3. ARVO OHJELMISTOTUOTANNOSSA

Arvon käsite on eri alojen tieteellisissä julkaisuissa määritelty ainakin viimeisen viidenkymmenen vuoden aikana hyvin eri tavoin (Rodriquez et al. 2018). Suositun Milesin (1972) arvon analysointiin ja suunnitteluun keskittyvä julkaisu näkee arvon yhtälönä, missä arvo lasketaan suorituskyvyn funktiona kustannuksiin nähden, jolloin arvoa voidaan lisätä joko parantamalla suorituskykyä tai vähentämällä kustannuksia. Monissa tieteenaloissa arvo määritellään puhtaasti taloudellisin käsittein. Esimerkiksi tunnettu yhdysvaltalainen sanakirjojen tuottaja Merriam-Webster määrittelee arvon tarkoittamaan tarkasteltavan kohteen ”rahallista arvoa” tai ”kaupallista arvoa” (Staff 2004).

Tässä luvussa keskitytään tarkastelemaan ensin arvoa ohjelmistotuotannon alalla, jättäen arvonmäärityksen ja arvoperusteisen päätöksenteon seuraaviin lukuihin. Luvussa tarkastellaan ensin yleisemmältä ohjelmistotuotannon tasolta arvon määritelmää ja arvoperusteisuuden motiiveja, minkä jälkeen siirrytään tarkastelemaan työn keskiössä olevien ohjelmistojen ominaisuuksien arvotekijöitä.

#### 3.1 Arvon määritelmä ja arvoperusteisen ohjelmistotuotannon motiivit

Kuten monilla muilla tieteenaloilla, myös ohjelmistotuotannon puolella ohjelmistojen taloudellinen puoli on ollut tärkeä osa tutkimusta jo vuosikymmeniä (Boehm 1984; Rodriquez et al. 2018). Sen lisäksi, että ohjelmistotuotannon tutkimus on vuosikymmenet keskittynyt ohjelmistoihin liittyvien päätösten taloudelliseen näkökulmaan, tutkimuksissa on historiallisesti arvon osalta otettu huomioon vain lyhyen aikavälin näkökulmat, kuten välittömät kustannukset ja aikataulu (Boehm 2003a; Browning 2003). Edellä mainitun tutkimusten painopisteen vuoksi viimeisen 30 vuoden aikana on ohjelmistotuotannon saralla panostettu huomattavasti erilaisiin kustannusten arviointimenetelmien kehitykseen (Jorgensen & Shepperd 2007). Biffi et al. (2006) mukaan ohjelmistotuotteiden arvo tulisi ottaa huomioon kokonaisvaltaisemmin, huomioiden lyhyen aikavälin arvokomponenttien lisäksi myös pidemmän aikavälin arvokomponentit. Myös Mendes et al. (2017) ajankohdaisempi tutkimus osoittaa, että säilyttääkseen kasvun ja kilpailuedun sekä innovatiivisuutensa ohjelmistoalan yritysten tulee muuttaa ajattelutapaansa arvoperusteisempaan suuntaan, jossa sekä lyhyen että pitkän aikavälin arvon näkökulmat otetaan huomioon.

Boehmin (2003b) mukaan suurin osa ohjelmistotuotannon tutkimuksesta on toteutettu arvoneutraalissa ympäristössä, jossa

- Jokainen vaatimus, käyttötapaus, tavoite ja vika käsitellään arvoltaan yhtä tärkeinä
- ”Ansaitun arvon” (earned value) järjestelmät monitoroivat projektin kustannuksia ja aikataulua, eivät sidosryhmien kokemaa tai liiketoiminnallista arvoa
- Työnjako on hoidettu niin, että ohjelmistoinsinöörien vastuu rajoittuu ohjelmistovaatimusten muuttamiseen toimivaksi koodinpätkäksi

Boehm (2003a) mainitsee, että suurin osa ohjelmistotuotannon arvoperusteisesta tutkimuksesta ottavat huomioon vain arvon lyhyemmän aikavälin näkökulmat. Tarve tasapainottaa lyhyen ja pidemmän aikavälin näkökulmat arvon tarkastelussa ovat luoneet pohjan arvoperusteisen ohjelmistotuotannon tutkimusalueen syntymiselle (Value-Based Software Engineering, VBSE) (Boehm (2003a)). VBSE korostaa, että ohjelmistotuotannossa ei riitä, että noudatetaan vain aikataulu-, budjetti ja laatuvaatimuksia, vaan ohjelmiston on viime kädessä luotava arvoa (Biffel et al. 2006). Boehm (2003b) esitteli VBSE:n pohjaksi laajemman näkemyksen arvosta, joka ylittää perinteisen taloudellisen painopisteen ottamalla arvon käsitteeseen huomioon myös esimerkiksi suhteellisen arvon, hyödyllisyyden ja tärkeyden. Arvon käsite ottaa siis huomioon sekä lyhyen että pidemmän aikavälin näkökulmat ohjelmistoihin liittyvien päätösten tekemisissä (Rodríguez et al. 2018). Rodríguez et al. (2012) mukaan ketterien menetelmien viimeaikainen suosio ja niiden painotus tuottaa arvoa asiakkaille ovat osaltaan edistäneet VBSE:n suosiota sekä laajentaneet sen soveltamisalaa.

Biffel et al. (2006, s. 3) mukaan lukuisat tutkimukset osoittavat, että ensisijainen menestystekijä, joka erottaa onnistuneet ohjelmistotuotteet ja -projektit epäonnistuneista löytyy arvoperusteisen ajattelun alueelta. Ohjelmistotuotannossa hallitsevan jalansijan 2000-luvulla saanut ketterä kehitys (agile development) korostaa ohjelmistojen arvoa asiakkaalle. Ketterä ohjelmistokehitys pohjaa kuitenkin näkemyksensä arvosta pääasiassa vain asiakkaan näkökulmasta, eli esimerkiksi ominaisuudet valitaan sen perusteella, minkä ominaisuuden odotetaan tarjoavan suurimman asiakasarvon (Rodríguez et al. 2013).

Huolimatta arvon konseptin kiistattomasta tärkeydestä ohjelmistotuotannossa, arvon käsitettä ei vielä ole yksityiskohtaisemmin ymmärretty yrityksissä (Rodríguez et al. 2018). Biffel et al. (2006, s. 182) mukaan arvon käsitettä on vaikea määritellä ohjelmistotuotannon kontekstissa, sillä kehitysprosessiin osallistuu monia sidosryhmiä, joista kukin määrittelee arvon omasta näkökulmastaan. Esimerkiksi ohjelmistotuotteen arvo asiakkaalle ilmaistaan usein hyötyjen ja kustannusten suhteen, kun taas ohjelmistoyritykselle arvo määritellään tuotteen myynnistä saavutetulla tuotolla (Biffel et al. 2006, s. 182). Boehmin

(2003a) mukaan arvon tulisi ottaa huomioon kaikkien sidosryhmien kokema arvo sekä liiketoiminnallinen arvo. Biffi et al. (2006, s. IX – X) mukaan sidosryhmät toivovat sekä yksilöllisten että yhteisten tavoitteiden ohjaamana saavuttavansa aina joitain hyötyjä, olivat ne sitten aineellisia tai aineettomia, taloudellisia tai sosiaalisia, rahallisia tai käytännöllisiä tai jopa esteettisiä tai eettisiä. He määrittelevät arvon kokonaisvaltaisesti tarkoittamaan tätä perimmäistä hyötyä, joka on usein joissain määrin katsojan silmissä ja hyväksyy useita eri luonnehdintoja.

Khurum et al. (2013) kokonaisvaltainen yleisen tason kuvaus arvonäkökulmista jakaa arvon neljään eri perspektiiviin: taloudelliseen arvoon, asiakasarvoon, sisäiseen liiketoiminta-arvoon sekä innovoinnin ja oppimisen arvoon. Janin ja Ibrarin (2010) systemaattisen kartoitustutkimuksen mukaan arvonäkökulmien tutkimus ohjelmistotuotannon osalla on yleensä eristäytynyttä ja keskittyy usein vain tiettyihin rajattuihin arvonäkökulmiin. Esimerkiksi useassa alan tutkimuksessa otetaan arvon näkökulmista huomioon ainoastaan asiakkaalle tuotettu arvo. Woodruffin (1997) paljon viitatus määritelmän mukaan asiakasarvo sisältää asiakkaan mieltymyksen ja arvion niistä tuotteen piirteistä, piirteiden suorituskyvyistä ja seurauksista, jotka helpottavat asiakasta saavuttamaan tavoitteensa erilaisissa käyttötilanteissa. Biffi et al. (2006, s. 180) mukaan kuitenkin keskittymällä pelkästään tietyn asiakkaan kokemaan arvoon voi johtaa siihen, että arvon tekijöitä poissuljetaan muilta sidosryhmiltä mukaan lukien muut asiakkaat, eri markkinat, ohjelmistokehittäjät ja projektipäälliköt. Heidän mukaansa tämä voi vaarantaa ohjelmistoyrityksen pidemmän aikavälin elinkelpoisuuden. Koska eri sidosryhmillä on erilaisia tarpeita ja toiveita, jotka kaiken lisäksi vaihtelevat ajan mukaan, ohjelmistoyritykset ovat pakotettuja luomaan arvoa monissa ulottuvuuksissa mukaan lukien taloudellisissa, fyysisissä, tunnepohjaisissa, sosiaalisissa, kognitiivisissa ja poliittisissa ulottuvuuksissa (Nunamaker et al. 2001). Khurum et al. (2013) mukaan arvonluonti realisoituu vasta, kun yritys tekee tulosta menestymällä markkinoilla.

Barney et al. (2008) jakavat ohjelmistotuotteiden arvon kolmeen eri dimensioon: tuotteen arvo (product value), asiakkaan havaittu arvo (customer's perceived value) ja asiakassuhteiden arvo (relationship value). Tuotteen arvo linkittyy tuotteen hintaan ja se voi muuttua kilpailevien tuotteitten mukaan. Asiakkaan havaittu arvo on asiakkaan samaa hyötyä tuotteesta. Asiakkaan havaitsemaan arvoon voivat vaikuttaa muun muassa asiakasodotukset, aiemmat kokemukset sekä kulttuurillinen tausta. asiakassuhdearvo taas syntyy yrityksen ja asiakkaan välisistä sosiaalisista suhteista. (Barney et al. 2008)

Erdogmus et al. (2006) käyttävät arvosta määritelmää, jossa arvo ymmärretään tarkasteltavan kohteen nettovarallisuudeksi tai hyötyjen ja kustannuksien väliseksi erotukseksi,

jotka kaikki on oikaistu tietyn ajanhetken riskillä. Jos kustannukset jätetään huomiotta, ne ovat implisiittisiä tai ne ovat syntyneet ennen omaisuuserän arviointia, arvo voi viitata tulevaisuuden hyötyihin tai omaisuuserän jäljellä olevaan arvoon kyseisenä ajan-kohtana (Erdogmus et al. 2006).

### 3.2 Ohjelmiston ominaisuuksien määritelmä

Apel et al. (2016) mukaan ohjelmisto-ominaisuuden käsitettä on luonnostaan vaikea määritellä tarkasti, sillä toisaalta ominaisuudet kuvaavat sidosryhmien kuten loppukäyttäjien odotuksia tuotteelta, kun taas toisaalta ne kuvaavat suunnitteluun ja toteutustason liittyviä konsepteja, joita hyödynnetään ohjelmistokomponenttien luokittelussa. Tieteellisessä kirjallisuudessa viitataan usein Kang et al. (1990) määritelmään ohjelmisto-ominaisuuksista, joka määrittelee ominaisuuden merkittäväksi tai erottuvaksi käyttäjälle näkyväksi ohjelmistojärjestelmän näkökohdaksi, laatutekijäksi tai ominaispiirteeksi. IEEE:n (2008) (The Institute of Electrical and Electronics Engineers) standardin mukaan ominaisuus määritellään järjestelmän erottuvaksi ominaispiirteeksi, joka sisältää sekä toiminnalliset (functional) että ei-toiminnalliset (non-functional) piirteet, kuten esimerkiksi suorituskyvyn ja uudelleenkäytettävyyden. Salu & Ruhe (2005) määrittelevät nykypäivänä paljon yleistyneen inkrementaalisen ohjelmistokehityksen kontekstissa ominaisuudet tarkoittamaan asiakkaille toimitettuja myyntiyksiköitä.

Biffi et al. (2006) määrittelevät ohjelmistojen ominaisuudet yksiköiksi, jotka tuottavat lisäarvoa ohjelmistoille tarjoamalla lisätoimintoja, jotka ovat asiakkaalle kiinnostavia tai tuottavat tuloja ohjelmistotoimittajalle. Myös Mendes et al. (2017) lähestyvät ominaisuuksia arvon kautta, määritellen ominaisuudet tuotteen toiminnallisuuden palasiksi, jotka tuottavat liiketoiminta-arvoa. Edelliset määritelmät ovat kuitenkin ristiriidassa Marciuska et al. (2014) tutkimuksen kanssa, jonka mukaan ohjelmistotuotteisiin päätyy myös arvottomia tai arvoltaan marginaalisia ominaisuuksia, jotka muun muassa kuormittavat laitteistoa turhaan.

Rodriguez et al. (2018) ovat tieteellisen kirjallisuuden pohjalta valinneet ohjelmisto-ominaisuuksille kansanomaisen määritelmän, jonka mukaan ominaisuudet edustavat tarpeita, jotka kerätään tapaamisten kautta asiakkaiden tai muiden sidosryhmien kanssa, ja joiden valitsemisen jälkeen ominaisuuksia tarkennetaan vaatimusten selvitysprosessin (requirements elicitation process) aikana. Van Gorp et al. (2000) määrittävät ominaisuudet loogisiksi yksiköiksi, jotka määritellään joukolla vaatimuksia, jotka ovat sekä toiminnallisia että laadullisia. Tämä ominaisuuksien ja vaatimusten suhde on muutenkin mielenkiintoinen, koska useassa tieteellisessä artikkelissa niistä puhutaan synonyymeina. Myös Rodriguez et al. (2018) toteavat, että vaikka ominaisuudet voivat sisältää

useampia vaatimuksia ja näin ollen ovat hieman ylemmän tason käsite, monissa tieteellisissä artikkeleissa ei tehdä selvää eroa ominaisuuksien ja vaatimusten välille. Näiden artikkelien ja tutkimusten pohjalta muodostetun ymmärryksen avulla tässä työssä on otettu vaatimuksia painottavia tutkimuksia mukaan, jos vaatimuksia käsitellään riittävän korkealla tasolla ja samat käytännöt pätevät myös ominaisuuksien tapauksessa.

### 3.3 Ohjelmiston ominaisuuksien arvotekijät

Ohjelmistotuotteen arvo koostuu tuotteen julkaisulle valituista ominaisuuksien joukosta, joten arvon ymmärtäminen ominaisuuksien valinnan yhteydessä on hyvin tärkeää. Käytännön ohjelmistotuotannossa arvo määritellään usein joukkona tekijöitä, joita ei ole eksplisiittisesti todettu missään, vaan ne perustuvat päätöksentekijöiden subjektiivisiin näkemyksiin (Rodriguez et al. 2018). Ohjelmistotuotannon tieteenalan alkuvaiheilla ominaisuuksien arvon tekijät määriteltiin usein ainoastaan konkreettisesti mitattavina asioina, jotka loivat suoraa taloudellista hyötyä (esim. Boehm 1984). Sittemmin ohjelmistojen ominaisuuksien arvon tekijöiksi on tieteellisessä kirjallisuudessa alettu tunnistamaan myös useita aineettomia, vaikeammin mitattavissa olevia, niin sanottuja ”pehmeitä” arvotekijöitä (esim. Farbey et al. 1995; Biffel et al. 2006, s. 41). Favaren (2002) mukaan ohjelmistokehittäjät voivat lisätä arvoa ominaisuuksiin useilla eri tavoin, kuten erimerkiksi oppimalla luomaan uudelleenkäytettäviä ominaisuuksia tai oppimalla strategista joustavuutta ominaisuuksien toteutusprosessiin.

Vaikka tieteellistä kirjallisuutta löytyy jonkin verran ominaisuuksien arvottamisesta ja etenkin tähän pohjautuvista ominaisuuksien valinta- ja priorisointikriteereistä, niin toisaalta siitä, mistä ominaisuuksien arvo koostuu (ominaisuuksien arvotekijät), löytyy hyvin vähän kirjallisuutta. Myös useat ajankohtaiset viime vuosina tehdyt tutkimukset toteavat, että ominaisuuksien arvon tekijät eri sidosryhmille ovat tähän mennessä olleet tutkimattomia (Alahyari et al. 2017; Rodriguez et al. 2018; Rodriguez et al. 2020). Rodriguez et al. (2020) mukaan jotkut tutkimukset keskittyvät sidosryhmien arvotekijöiden keräämiseen ohjelmistotuotteiden hallinnan yhteydessä, mutta eivät spesifimmin keskittyen ohjelmisto-ominaisuuksiin.

Rodriguez et al. (2018) kattavassa soveltavassa tutkimuksessa tavoitteena oli tarjota yrityksille arvon eri tekijöistä konsolidoidun tason ymmärrys, joka sulkee pois subjektiivisuuden ja väärinymmärrykset. Ominaisuuksien arvolle löydettiin empiirisen tutkimuksen ja grounded theory -menetelmää yhdistämällä 36 eri tekijää, jotka luokiteltiin kuuden eri dimension alle: Asiakasarvo, kilpailukyky markkinoilla, taloudellinen arvo/kannattavuus, kustannustehokkuus, teknologia ja arkkitehtuuri sekä yrityksen strategia. Asiakasarvo

kattaa kaikki arvon komponentit, joita pidetään asiakkaan kannalta merkityksellisinä. Kilpailukyky markkinoilla koostuu arvon tekijöistä, jotka liittyvät tuotteeseen lisättyyn arvoon suhteutettuna vastaaviin kilpaileviin tuotteisiin markkinoilla. Taloudellisen arvon ja kannattavuuden arvodimensio taas pitää sisällään tekijät, jotka vaikuttavat yrityksen varallisuuteen, pääsääntöisesti rahamääräisesti arvioituna pidemmällä aikavälillä. Kustannustehokkuusdimensio sisältää kaikki suorat kustannukset, jotka ominaisuuden toteuttamisesta syntyisi. Teknologian ja arkkitehtuurin dimensio keskittyy teknisten näkökohtien ympärille kuten tuotearkkitehtuuriin sekä olemassa olevaan kykyyn toteuttaa tietty ominaisuus. Viimeinen arvon dimension kuudesta eli yrityksen strategia keskittyy näkökohtiin, jotka tarjoavat yleisen suunnan yrityksen organisaatiollisille tavoitteille sekä strategian saavuttaa nämä tavoitteet. Empiirisessä haastattelututkimuksessa haastateltavat ohjelmistoalan yritysten päätöksentekijät mainitsivat asiakasarvodimensioon kuuluvat arvotekijät ylivoimaisesti yleisimmin. Vähiten mainitut arvon tekijät kuuluivat taloudellisen arvon ja kannattavuuden sekä yrityksen strategia -dimensioiden alle. (Rodriguez et al. 2018)

Rodriguez et al. (2020) toistivat hieman eri kokoonpanolla edellistä vastaavan tutkimuksen ottamalla tutkimuksen haastatteluihin mukaan edellisen tutkimuksen kohdeyrityksen avainsidosryhmien lisäksi kahden muun ohjelmistoalan yrityksen avainsidosryhmät. Näin löydettiin yhteensä 47 arvotekijää, eli 11 uutta arvotekijää aikaisempaan tutkimukseen verrattuna. Myös nämä uudet arvotekijät pystyttiin luokittelemaan aiempien kuuden eri arvodimension alle. Uutena löydöksenä tutkimuksessa kohdeyrityksien lukumäärää kasvattamalla havaittiin, että arvotekijät voidaan luokitella kahteen eri ryhmään: arvon perustekijöihin (core value propositions) ja spesifisiin arvotekijöihin (specific value proposition). Arvon perustekijät olivat yhteisiä kaikissa tutkimuksen kolmessa kohde yrityksessä, kun taas spesifiset arvotekijät olivat konteksti- tai yritysriippuvaisia. Arvon perustekijät sisälsivät muun muassa asiakastyytyvyyden, markkinoillesaantiajan sekä sijoitetun pääoman tuottoasteen. Spesifisiin arvotekijöihin taas kuuluivat esimerkiksi ominaisuuksien näkyvyys sekä tuoteturvallisuus. (Rodriguez et al. 2020) Myös Barney et al. (2008) tutkimus tukee väitettä ominaisuuksien arvotekijöiden kontekstiriippuvuudesta, mainitsemalla ominaisuuksien arvoon vaikuttaviksi tekijöiksi muun muassa sopimuksen ja asiakkaan luonteen.

Lisäksi Rodriguez et al. (2020) tutkimuksessa huomattiin, että arvotekijät vaihtelivat kahden eri tekijän mukaan: 1) ominaisuuden tyyppin perusteella (esim. pakollinen ominaisuus vs. ei-pakollinen ominaisuus) ja 2) sidosryhmien perusteella (eri sidosryhmät keskittyivät eri arvotekijöihin). Pakollisilla ominaisuuksilla tarkoitetaan sellaisia ominaisuuksia, joita



ilman tuote ei toimisi, joten näiden osalta yleensä päätöksenteko esimerkiksi ominaisuuden toteuttamisesta on melko suoraviivainen ja toteutuu usein nojaten vain yhteen pääasialliseen arvotekijään. Eri sidosryhmäkategorioilta taas löydettiin lievästi toistuvia kaavoja eri arvotekijöiden tunnistamisessa, esimerkiksi liiketoimintasidosryhmät mainitsivat asiakasarvotekijöihin kuuluvat arvotekijät muita sidosryhmiä useammin. (Rodriguez et al. 2020)

Taulukossa 2 esitetään eri tutkimuksissa tunnistettuja ohjelmistojen ominaisuuksien arvotekijöitä. Taulukkoon on valittu luvussa 2 kuvatun systemaattisen kirjallisuuskatsausprosessin perusteella luetuista artikkeleista ne, joissa tunnistettiin ominaisuuksien arvoon vaikuttavia arvotekijöitä empiirisissä ohjelmistoalan yrityksiin toteutetuissa tutkimuksissa. Taulukon pohjana käytetään aiemmin kuvattua Rodriguez et al. (2020) tutkimuksesta kaikille kolmelle case-yritykselle yhteisesti tunnistettua 24 perusarvotekijää, sillä ne viittaavat muihin yrityksiin paremmin yleistettävissä oleviin arvotekijöihin. Tulee kuitenkin huomioda, että Rodriguez et al. (2020) tutkimuksessa löydettiin kolmen case-yrityksen avainsidosryhmien välillä myös 23 muuta arvotekijää. Tämä tukee väitettä, että on myös paljon yritys- ja kontekstiriippuvaisia arvotekijöitä. Kontekstisidonnaisten arvotekijöiden takia on mahdotonta määrittellä kaikille ohjelmistoalan yrityksille yhtä yleistä mallia, joka esittäisi jokaiseen tilanteeseen sopivat ominaisuuksien arvotekijät. Useissa taulukon 2 tutkimuksissa myös käytettiin eri termejä samankaltaisista arvotekijöistä, ja taulukkoa muodostaessa kirjoittaja onkin joutunut tekemään arvotekijöiden kuvausten perusteella useita tulkintoja siitä, tarkoitetaanko kyseisen tutkimuksen arvotekijällä samaa asiaa, kuin taulukkoon valituissa arvotekijöissä. Taulukon 2 arvotekijöiden lyhyet kuvaukset löytyvät työn liitteestä A.

**Taulukko 2.** Tutkimuksissa tunnistettuja ohjelmisto-ominaisuuksien arvotekijöitä (Rodríguez et al. 2020 vertailulogiikkaa mukailien)

Rodríguez et al. (2020) mukaiset 24 arvon perustekijää	Biffi et al. (2006, s. 187-188)	Barney et al. (2009)	Alahyari et al. (2017)	Mendes et al. (2017)
<b>Asiakasarvo</b>				
T1. Asiakastyytyväisyys	x	x		x
T2. Loppukäyttäjätyytyväisyys	x	x	x	x
T3. Markkinatyytyväisyys				x
T4. Asiakkaiden lukumäärä				
T5. Johtavien asiakkaiden tyytyväisyys				
<b>Kilpailukyky markkinoilla</b>				
T6. Tuotteen kilpailukyky	x	x	x	x
T7. Markkinoillesaantiaika	x	x	x	x
T8. Tuotteen laatu			x	
T9. Tuotteen suorituskyky			x	
T10. Käyttäjäkokeksellinen yksinkertaisuus			x	x
<b>Taloudellinen arvo / kannattavuus</b>				
T11. Sijoitetun pääoman tuotto	x		x	x
T12. Myyntimäärä				x
T13. Asiakaskunta/Markkinaosuus				x
<b>Kustannustehokkuus</b>				
T14. Kehityksen kokonaiskustannukset	x	x	x	x
T15. Uudelleenkäytettävyys				
T16. Kolmansien osapuolien kustannukset				
<b>Teknologia ja arkkitehtuuri</b>				
T17. Toteutuksen yksinkertaisuus	x	x		
T18. Tuotearkkitehtuuri	x	x		
T19. Kehitysvalmiudet	x	x		x
<b>Yrityksen strategia</b>				
T20. Yrityksen strategia		x		x
T21. Tuotteen strategia				
T22. Yrityksen portfolio				
T23. Liiketoiminnallinen edesauttaja				x
T24. Yrityksen brändi				

Biffi et al. (2006, s. 187-188) empiirinen tutkimus toteutettiin kyselytutkimuksena kohdistuen kahteen ohjelmistoalan yritykseen. Tutkimuksessa tunnistettiin 13 ominaisuuksien arvotekijää. Taulukkoon 2 valittujen arvon perustekijöiden ulkopuolelta tutkimuksessa

tunnistettiin arvotekijöiksi myös ominaisuuden muutosherkkyys, ominaisuuksien ylläpidolliset vaikutukset sekä tuen tarve koulutukselle. Tutkimuksen case-yritykset arvottivat ominaisuuksien arvotekijöistä merkityksellisimmiksi kehityksen kustannus-hyötysuhteen, toimitusajan, ominaisuusvaatimuksen liikkeellelaskijan sekä saatavilla olevat resurssit. Vähemmän arvokkaina arvon tekijöinä tutkimuksessa pidettiin kehitykseen ja ylläpitoon liittyviä tekijöitä, kuten monimutkaisuutta sekä ominaisuuden vaikutusta olemassa olevaan järjestelmään.

Barney et al. (2009) tutkimus jakaa ominaisuuksien arvoon vaikuttavat tekijät liiketoiminta-, projekti- ja tuotenäkökulmiin. Tutkimus suoritettiin empiirisenä tutkimuksena, johon osallistui 11 Kiinassa toimivaa yritystä. Tutkimuksessa arvon tekijöitä ei ole jaettu yhtä yksityiskohtaiselle tasolle kuin taulukossa 2 pohjana käytettyä Rodriguez et al. (2020) tutkimusta, mikä voi osaltaan selittää vähäisemmän arvotekijöiden määrän. Barney et al. (2009) tutkimus nimesi myös taulukkoon 2 mukaan otettujen arvon perustekijän ulkopuolelta arvotekijöiksi ominaisuuksien muutosherkkyden, ylimääräiset kustannukset sekä jälkimarkkinointituen. Tutkimuksen case-yritykset painottivat eniten liiketoimintänäkökulmaan jaettuja ominaisuuksien arvotekijöitä, johon kuuluivat liiketoimintastrategia, asiakastyytyväisyys, kilpailijat sekä vaatimuksen liikkeellelaskija.

Alahyari et al. (2017) empiirisen tutkimuksen tavoitteena oli selvittää, miten arvo havaittiin 14:sta ketterässä ohjelmistokehitysyrityksessä. Tutkimuksen mukaan yleisesti tärkeimpänä arvotekijänä pidettiin toimitusprosessia toimitusajan näkökulmasta. Roolikohdaisesti tutkimuksessa tuoteomistajat ja projektipäälliköt arvottivat havaitun laadun tärkeimmäksi arvotekijäksi ja toiminnallisuuden vähiten tärkeimmäksi arvotekijäksi. Kuten monet muutkin empiiriset tutkimukset ominaisuuksien arvotekijöistä, myös Alahyari et al. (2017) tutkimus tukee väitettä arvotekijöiden tärkeyden kontekstiriippuvuudesta. Alahyari et al. (2017) mukaan arvotekijöiden tärkeydet vaihtelevat paljon erityyppisillä ohjelmistotuotteilla.

Mendes et al. (2017) empiirinen tutkimus ominaisuuksien arvotekijöistä toteutettiin yhden suuren ohjelmistoalan case-yrityksen kanssa. Tutkimuksessa haastateltiin yrityksen tietyn tuotteen kolmea avainsidosryhmää, mikä tuotekontekstisidonnaisuuden sekä otannan pienen määrän vuoksi voi rajoittaa arvotekijöiden yleistettävyyttä. Tutkimus toteutettiin kahdessa vaiheessa, jossa ensimmäisessä vaiheessa jokaisen kolmen sidosryhmän kanssa toteutettiin yksilöllinen puolistrukturoitu haastattelu, jonka pohjalta ominaisuuksien arvotekijät koostettiin. Toisessa vaiheessa kaikkien kolmen sidosryhmän kanssa järjestettiin yhteinen kohderyhmätapaaminen, jossa muun muassa yhdessä validoitiin kerättyjen ominaisuuksien arvotekijöiden todellinen merkitys sidosryhmille ja pudotettiin

pois vähemmän tärkeitä arvotekijöitä. Tutkimuksessa tunnistettiin lopulta 21 ominaisuuksien arvotekijää. Arvotekijöistä käytettiin jälleen usein eri termejä kuin taulukon 2 arvotekijöistä, jolloin arvotekijöiden taulukkoon sijoittamisessa on taas kirjoittajan toimesta tehty tulkintoja tekijöiden kuvausten pohjalta (esim. loppukäyttäjäkokemus rastitettu käyttäjäkokemuksellisen yksinkertaisuuden riville). Taulukkoon 2 valittujen arvotekijöiden ulkopuolelta tutkimuksessa tunnistettiin muun muassa tietoturvallisuuden vaikuttavia tekijöitä sekä tuki- ja ylläpitokustannustekijöitä.

Taulukkoon 2 valituista tutkimuksista jokainen mainitsi ominaisuuksien arvotekijöiksi loppukäyttäjätyytyväisyyden, tuotteen kilpailukykyyn, markkinoillesaantiajan sekä kehityksen kokonaiskustannukset. Seuraavaksi yleisimmin arvotekijöiksi mainittiin asiakastyytyväisyys, sijoitetun pääoman tuotto sekä kehitysvalmiudet/-resurssit. Empiirisissä tutkimuksissa vähiten huomioitiin yrityksen strategia -dimension alle kuuluvia arvotekijöitä. Huomionarvoista myös on, että taulukon pohjana käytetyistä Rodriguez et al. (2020) esittelemistä arvon perustekijöistä ei muissa tutkimuksissa mainittu lainkaan asiakkaiden lukumäärää, johtavien asiakkaiden tyytyväisyyttä, uudelleenkäytettävyyttä, kolmansien osapuolien kustannuksia, tuotteen strategiaa, yrityksen portfolioa sekä yrityksen brändiä. Edellinen tosin voi selittyä myös sillä, että taulukon sarakkeina esiintyvien tutkimusten arvotekijät olivat usein abstraktiotasoltaan hieman korkeammalta tasolta. Myös työssä tehdyt sijoitukset arvotekijöiden samankaltaisuuden perusteella vaikuttavat taulukon tuloksiin.

## 4. ARVON MÄÄRITTÄMINEN

Olisi yksinkertaista olettaa, että kaikki ohjelmistoprojektiin vaikuttavat asiat, kuten esimerkiksi eri sidosryhmät tai eri ominaisuudet, olisivat kaikki samanarvoisia. Myös päätöksenteko arvoneutraalissa ympäristössä olisi yksinkertaisempaa ja nopeampaa. Reaalimaailmassa kuitenkin edellä mainittu yksinkertaistaminen saavutetaan usein tulosten kustannuksella, jolloin järjestelmät eivät kykene tarjoamaan parasta mahdollista arvoa sidosryhmille. (Biffel et al. 2006, s. 155 – 156) Arvoperusteinen ohjelmistotuotanto edellyttää kykyä mitata ja arvioida arvoa, jotta arvon pohjalta voidaan tehdä tietoon perustuvia päätöksiä tulosten arvon maksimoimiseksi.

Tässä luvussa tarkastellaan ohjelmistojen arvonmäärityksen menetelmiä sekä haasteita. Luvussa luodaan ensin katsaus arvonmääritykseen ja sen haasteisiin yleisemmältä ohjelmistotuotannon tasolta. Tämän jälkeen luvussa siirrytään tarkastelemaan ohjelmist ominaisuuksien arvonmääritystä sekä nykyisiä ominaisuuksien arvonmääritysmenetelmiä.

### 4.1 Arvonmääritys ohjelmistotuotannossa

Perinteinen ohjelmistotuotannon taloudellisen puolen tutkimus keskittyy usein yksinomaan kustannusaspekteihin ja teknisiin näkökulmiin tuottaakseen tietoa päätöksenteon tueksi. Perinteiset kustannuspainotteiset tekniikat ottavat kuitenkin huomioon vain konkreettiset hyödyt, kuten suorat säästöt vähentyneen työn vuoksi. Arvo sisältää kuitenkin myös aineettomia komponentteja, kuten joustavuuden sekä oppimisen, jotka voivat tuottaa merkittävää arvoa ohjelmistojen kehittämiseksi pidemmällä aikavälillä. Arvoperusteinen ohjelmistotuotanto (VBSE) laajentaa ajattelumaailmaa kustannuspainotteisuudesta ottaen huomioon kattavammin hyödyt, mahdollisuudet ja riskit. (Biffel et al. 2006, s. 39 – 42). Keskeistä tässä yhteydessä on arvonmääritys (valuation), jonka Biffel et al. (2006, s. 39) määrittävät prosessiksi tuotteen, palvelun tai prosessin taloudellisen arvon määrittämiseksi. Arvonmäärityksen keskeinen tavoite on tunnistaa kokonaisuuksien ja komponenttien arvoa tuottavat tekijät, jotta arvoa voidaan hallita jollain formaalilla tavalla. Arvonmäärityksellä tavoitellaan myös yhtenäistä näkymää arvosta, joka mahdollistaa paremman kommunikoinnin arvonäkökulmiin ja päätöksentekoon liittyen eri sidosryhmien välillä. (Khurum et al. 2013)

Ohjelmistokomponenttien arvonmääritys riippuu yksityiskohtaisesta analyysistä taustalla olevista kustannuksista ja hyödyistä (Biffi et al. 2006, s. 41; Khurum et al. 2013). Kustannus-hyötyanalyysin ennakoedellytys on asiaankuuluvien arvo- ja kustannustekijöiden tunnistaminen. Vaikka ohjelmistojen kehittämiskustannusten mallit ovat vakiintuneita, puuttuvat ohjelmistotuotannon alalta kattavat määritelmät yksittäisistä arvoajureista (esim. suoritusmuuttujat, jotka tukevat päätöksentekoa ja priorisointia) sekä arvon luomisen viitekehykset. (Biffi et al. 2006, s. 41) Myös Imtiaz et al. (2009) mainitsevat yleisimmäksi arvonmäärityksen lähestymistapojen ensimmäiseksi vaiheeksi oleellisten arvoajurien tunnistamisen sekä kustannuksille että hyödyille. Khurum et al. (2013) systemaattisen kartoitustutkimuksen mukaan kuitenkin perinteiset kustannus-hyötyanalyysimenetelmät eivät subjektiivisen luonteensa takia riitä havaitsemaan kaikkia arvon tekijöitä (kuten esimerkiksi innovaatioarvoa). Myös Rodriguez et al. (2018) mukaan käytännön ohjelmistotuotannossa arvo määritetään usein joukkona arvotekijöitä, joita ei yleensä ole eksplisiittisesti todettu, vaan ovat päätöksentekijöiden palveluksessa subjektiivisesti. Tämän vuoksi eri tutkimuksissa on viime vuosina yritetty kehittää objektiivisempia mittareita arvon määrittämiselle, jotka eivät olisi yhtä alttiita päätöksentekijöiden subjektiivisen kokemuksen tuomille virheille.

Alan kirjallisuudessa taloudellisuutta on usein pidetty tärkeimpänä tekijänä liiketoimintapäätösten tekemisessä, jonka vuoksi se on muodostanut perustan arvon määrittämiselle. Khurumin ja Barney'n (2009) mukaan ohjelmistotuotannon vakiintunein investointien arvonmääritysmenetelmä on nettonykyarvon laskeminen diskonttaamalla arvioidut tulevaisuuden kassavirrat markkinoiden riskittömällä korolla. Farbey et al. (1995) kuitenkin argumentoivat, että vaikka toimintojen arvoa organisaatiossa voidaan monissa tapauksissa mitata suoraan taloudellisesti esimerkiksi lyhyen aikavälin helposti kvantifioitavien kustannussäästöjen perusteella, niin on myös olemassa paljon arvokomponentteja kuten pehmeät strategiset edut, joita ei voida ilmaista suoraan rahamääräisinä. Myös Auer et al. (2011) kyseenalaistavat taloudellisten arvonmääritysmenetelmien, kuten nettonykyarvon ja sijoitetun pääoman tuoton kyvyn ottaa huomioon arvoon kuuluvat aineettomat hyödyt ja pitkän aikavälin strategiset edut.

Khurum et al. (2013) jakavat arvonmääritysmenetelmät kolmeen eri pääkategoriaan niiden toimintalogiikan mukaan. Nämä kolme eri kategoriaa ovat arvonmääritys pisteytysmenetelmillä, indeksimenetelmillä sekä rahamääräisillä menetelmillä. Pisteytysmenetelmiä käytetään arvonmäärityksessä ohjelmistotuotannon alalla laajasti. Pisteytysmenetelmässä usein ensin tunnistetaan oleelliset arvotekijät ja niiden painoarvot, jonka jälkeen sidosryhmät arvottavat arvotekijät subjektiivisesti niiden tärkeyden perusteella. Khurum et al. (2013) mainitsevat esimerkkinä asiakasarvoanalyysin, jossa asiakkaan

edustajalle annetaan ohjelmistotuotteen arvotekijät (esim. suorituskyky, käytettävyys, hyödyllisyys) pisteytettäväksi niiden tärkeyden perusteella. Pisteytysmenetelmät ovat suosittuja, koska ne ovat yksinkertaisia ja vakaita (Park & Park 2004). Pistemääräisen arvon arvioinnin heikkoutena on, että se ei kuitenkaan koskaan välitä todellista absoluuttista arvoa, vaan ennemminkin esittää tarkasteltavan kohteen suhteellisen arvon eri vaihtoehtoihin nähden.

Indeksimenetelmiä voidaan hyödyntää arvonmäärityksen toiminnallisten muotojen kehittämiseen. Indeksimenetelmistä käytetään kirjallisuudessa myös nimeä suhdelukumene-  
telmät (ratio methods). Indeksimenetelmät ovat pisteytysmenetelmiä joustavampia, koska ne mahdollistavat monipuolisempia arvonmääritysmittoja, kuten suhdelukuja ja prosenttimääriä. (Khurum et al. 2013) Suhdeluvut kykenevät ottamaan huomioon myös muita kuin taloudellisia arvon komponentteja (Renkema & Berghout 1997). Suhde- ja prosenttiluvut ovat kuitenkin myös suhteellisia mittoja, ja indeksimenetelmät kärsivätkin samasta heikkoudesta kuin pisteytysmenetelmät. Kumpikaan edellisistä menetelmistä ei kykene ilmoittamaan arvioitavan kohteen todellista arvoa.

Rahamääräiset arvonmääritysmenetelmät tavoittelevat mitattavan kohteen arvon ilmaisemista rahamääräisenä. Yleisin rahamääräinen arvonmääritysmenetelmä on nettonykyarvon laskeminen, joka ottaa huomioon myös rahan aika-arvon diskonttaamalla tulevat kassavirrat tietyllä diskonttokorolla (Khurum et al. 2013). Käytännössä jos mahdollista, rahamääräinen arvonmääritysmalli on kaikkein toivottavin, koska se esittää arvon sen todellisen merkityksen mukaisesti (Park & Park 2004). Kuitenkin kuten aiemmin mainittu, etenkin aineettomien strategisten hyötyjen arvonmääritys rahamääräisenä on hyvin haastavaa. Myös Park ja Park (2004) mainitsevat rahamääräisen arvonmäärityksen olleen pettymyksellistä tähän asti, minkä vuoksi rahamääräiset menetelmät eivät ole levinneet yritysten hyödyntämiin käytäntöihin laajalti. Khurum et al. (2013) tutkimuksen empiirinen osio telekommunikaatioyhtiö Ericssonissa osoitti, että monia arvon komponentteja (esim. toiminnallista arvoa) oli käytännössä hyvin vaikea arvioida rahamääräisenä.

Auer et al. (2011) jakavat ohjelmistojen arvonmääritysmenetelmät hieman edellisistä poikkeavalla tavalla viiteen eri kategoriaan: taloudellisiin menetelmiin, monikriteerisiin menetelmiin, suhdelukumene-  
menetelmiin, investointiportfoliomenetelmiin sekä muihin menetelmiin. Taulukossa 3 esitetään Auer et al. (2011) koostamat yleiset arvon arviointimenetelmät jaoteltuina edellisiin kategorioihin.

**Taulukko 3.** Yleisiä arvonmäärittämenetelmiä (mukailtu lähteestä Auer et al. 2011)

Kategoriat	Menetelmät
Taloudelliset menetelmät	Kannattavuusraja-analyysi Kustannus-hyötyanalyysi Nettonykyarvo Sisäinen korkokanta Takaisinmaksuaika Reaaliopioiteoria Sijoitetun pääoman tuotto
Monikriteeriset menetelmät	Analytical Hierarchy Process (AHP) Kriittiset menestystekijät Delfoi-metodi IT tulokortit Informaatiotaloustiede Pisteytysmenetelmät
Suhdelukumenetelmät	Return on Management (ROM) -menetelmä
Investointiportfolio menetelmät	Bedellin menetelmä Investointikartoitus Wardin menetelmä
Muut menetelmät	Benchmarking Peliteoria

Päätöksenteon kannalta olisi tärkeää, että arvonmäärittäksessä eri arvotekijät voitaisiin yhteismitallistaa jonkin tietyn mitattavan suureen alle. Myös erilaisia ohjelmistotuotannon kokonaisuuksia, kuten esimerkiksi kokonaisia ohjelmistoprojekteja, tulisi voida mitata saman suureen avulla vertailun mahdollistamiseksi. Liiketoimintapäätöksiä tehdessä (esim. investoinnit) päätöksentekijät usein suosivat arvonmäärittästä taloudellisiin mitoin, kuten esimerkiksi ohjelmistoprojektin tai -tuotteen arvonmäärittästä euromääräisenä. Khurum et al. (2013) mukaan arvonmäärittäksen keskeisenä haasteena on, että käytännön ohjelmistotuotannossa eri malleja käytetään eri arvorakenteiden arvonmäärittäseen, mikä johtaa eri arvomittojen yhteensopimattomuuteen.

Kun sidosryhmät ottavat päätöksentekoon rajallisten resurssien allokoinnin huomioon, heidän päätöksensä perustuvat jossain määrin heidän asenteisiinsa riskiä kohtaan sekä



miten heillä on tapana vastata epävarmuuteen. Nämä asenteet heijastuvat arvonmäärittämisessä päätöksentekijän arvioon arvosta. (Biffi et al. 2006, s. 43) Biffi et al. (2006, s. 43) mukaan arvonmäärittämisessä tärkein mallinnuskonsepti on hyödyllisyys. Taloustieteistä poiketen ohjelmistotuotannossa hyödyllisyysfunktiot otetaan usein käyttöön välttääkseen rahallisen arvon asettamista eduille ja kustannuksille. Useiden projektien vuorovaikutukset vaikuttavat usein myös arvoon. Arvon tarkan määrittelyn saavuttamiseksi nämä vuorovaikutukset on otettava huomioon. Eri ohjelmistoprojektien vuorovaikutusten huomioonottamiseksi suositellaan portfoliooperusteista lähestymistapaa. Portfoliooperusteisessa lähestymistavassa varoja ei arvoteta erikseen, vaan huomioon otetaan varojen yhteisvaikutus. (Biffi et al. 2006, s. 44) Tämä tarkoittaa sitä, että omaisuuserän kokonaisarvo ei siis ole vain yksinkertaisesti sen osien arvon summa.

Biffi et al. (2006, s. 160) mukaan ohjelmistokomponenttien arvonmäärittäminen alkaa määrittelyistä arvotekijöistä. Tämän työn aluvussa 3.3 käsiteltiin eri julkaisuissa tunnistettuja ohjelmisto-ominaisuuksien arvotekijöitä. Jokaista näistä arvotekijöistä tulee mitata (tai arvioida) sopivalla asteikolla. Arvotekijöiden mittaus- ja analysointiprosessi alkaa mittausuunnitelmalla, jossa määritetään, kuinka arvotekijöiden arvot saadaan kerättyä. On olemassa standardeja, jotka määrittävät mittausmalleja tietyille ominaisuuksille (esim. aika, hinta, toiminnallinen koko), mutta on myös monia arvotekijöitä (esim. teknologian hyväksyntä, strategiset vaikutukset), joiden arvonmäärittämiseen tarvitaan erilaisia lähestymistapoja, kuten empiirisiä haastatteluja sidosryhmien kesken. Arvotekijöiden arvottamisen jälkeen jokaiselle arvotekijälle määritetään suhteellinen vaikutusaste, jota käytetään kertoimena kunkin arvotekijän tärkeydelle (Biffi et al. 2006, s. 160).

## 4.2 Arvonmäärittäminen haasteet

Kun tietoa ohjelmistokomponenttien arvonmäärittämisestä haettiin systemaattisilla hauilla, saatiin tuloksena hyvin paljon julkaisuja arvonmäärittäminen haasteista. Edellä on erilaisien menetelmien yhteydessä huomioitu omaisuusvarojen, varsinkin aineettomien, arvonmäärittäminen haastavuus. Useimmissa arvonmäärittämisestä käsittelevissä alan julkaisuissa mainitaan epävarmuus pääasiallisesti arvonmäärittäminen haasteeksi (esim. Biffi et al. 2006, s. 39; Khurum & Barney 2009). Epävarmuuden lisäksi Marcuska et al. (2013b) mainitsevat kaksi muuta ensisijaista haastetta arvonmäärittämisessä: sidosryhmien eroavat näkemykset arvosta sekä arvon moniulotteisen luonteen. Tässä aluvussa tarkastellaan lyhyesti kaikkia kolmea edellistä haastetta tieteelliseen kirjallisuuteen nojautuen.

### 4.2.1 Epävarmuus

Kaikkia arvon määrittäviä tekijöitä ei usein tunneta arvon arviointihetkellä ympäristölle ominaisen epävarmuuden takia. Favaren (2002) mukaan ominaisuuksien tarkka arvonmäärittäminen on hyvin haastavaa tai jopa mahdotonta ohjelmistoprojektin elinkaaren alkuvaiheessa, jolloin tuntemattomia tekijöitä on enemmän. Ohjelmiston arvo realisoituu vasta, kun se on tuotu markkinoille (Park & Park 2004). Myös Lindgren et al. (2008) tukevat väitettä, että on hyvin haastavaa arvioida asiakasarvoa tai taloudellista arvoa, ennen kuin loppukäyttäjät ovat päässeet käyttämään projektissa kehitettyjä ominaisuuksia. Epävarmuuden mallintamisen mainitaankin kirjallisuudessa usein olevan enemmän taidetta kuin tiedettä (Park & Park 2004; Biffi et al. 2006, s. 40).

Biffi et al. (2006, s. 42) jakavat epävarmuuden lähteet kolmeen eri kategoriaan: luonnolliseen epävarmuuteen, parametrien epävarmuuteen sekä käytettyjen mallien epävarmuuteen. Luonnollinen epävarmuus johtuu suoraan ympäristömuuttujien vaihtelusta, kuten esimerkiksi ohjelmistotuotteen virheiden lukumäärien vaihtelusta. Parametrien epävarmuus liittyy parametrien arviointien luotettavuuteen, kuten esimerkiksi keskimääräisten vika-arvojen luotettavuuteen. Mallien epävarmuus taas liittyy käytettyjen mallien soveltuvuuteen, kuten esimerkiksi tietyn virheidenmallinnusjakauman soveltuvuuteen. (Biffi et al. 2006, s. 42)

Perinteinen lähestymistapa epävarmuuden käsittelemiseen on määrittellä todennäköisyysjakaumat taustalla oleville suureille, mikä mahdollistaa normaalilaskennan (standard calculus) periaatteiden hyödyntämisen. Muut lähestymistavat, kuten sumeat mittarit (fuzzy measure) tai Bayesilainen verkosto (Bayesian network) vaativat hieman monimutkaisempia ennakkotietoja. Joskus aineettomat hyödyt kuten oppiminen, kasvumahdollisuudet ja joustavuus ovat vallitsevia arvon lähteitä epävarmassa ympäristössä. Näitä etuja on vaikea määrittää ja ne edellyttävät edistyneempiä tekniikoita, kuten päätökstekopuita (decision tree) ja reaalioptioanalyysiä (real options analysis), jotka ovat suunniteltu käsittelemään epävarmuutta. (Biffi et al. 2006, s. 41–42) Huomionarvoista on, että kyseisetkään lähestymistavat eivät voi poistaa arvonmäärittämiseen liittyvää epävarmuutta, mutta niiden avulla voidaan joko vähentää epävarmuutta tai ainakin tunnistaa mahdollisia epävarmuustekijöitä.

### 4.2.2 Sidosryhmien eroavat näkemykset arvosta

Biffi et al. (2006, s. 157) mukaan arvo on aina tietyssä määrin katsojan silmissä, mikä vaikeuttaa arvonmäärittämisen prosessia. Sidosryhmien eroavat näkemykset arvosta sekä eri riskipreferenssit vaikuttavat arvonmäärittämisen prosessiin. Farbey & Finkelstein

(1999) määrittelevät sidosryhmiksi kaikki osapuolet, jotka vaikuttavat projektisuunnitelmaan tai joihin projektisuunnitelma vaikuttaa. Tämän työn kontekstiin sopii hyvin myös Rodriguez et al. (2020) vastaavanlainen määritelmä, joka määrittelee ohjelmisto-ominaisuuksien kontekstissa sidosryhmät tarkoittamaan ryhmää henkilöitä, jotka voivat vaikuttaa ominaisuuksien valintaan julkaisusuunnittelussa tai joihin kyseinen ominaisuuksien valinta vaikuttaa. Eri sidosryhmiä ovat muun muassa asiakkaat, loppukäyttäjät, kehittäjät, testaajat, projektipäälliköt ja tuoteomistajat.

Arvonmäärittämisessä tulisi ensin selvittää, mitä arvo tarkoittaa kullekin mukana olevalle sidosryhmälle. Khurum et al. (2013) tutkimuksen mukaan arvon määritelmä vaihtelee eri sidosryhmien välillä ja useissa tapauksissa arvo ymmärretään eri tavalla. Biffi et al. (2006, s. 157) tukevat väitettä huomauttamalla, että jokaisella henkilöllä on oma käsityksensä arvosta tietyssä tilanteessa. Marciuska et al. (2013b) mukaan perinteisesti ohjelmistoyritykset yrittävät arvioida ominaisuuksien eri sidosryhmille tuomaa arvoa haastatteleamalla useita keskeisiä sidosryhmiä. Myös useita formaalimpia menetelmiä on kehitetty tunnistamaan eri sidosryhmien kokemaa arvoa. Valitettavasti kuitenkin usein kyseiset menetelmät ovat helposti manipuloitavissa omaa etuaan tavoittelevien sidosryhmien toimesta (Duan et al. 2009). Esimerkiksi valittaessa ohjelmistojen julkaisuihin toteutettavia asioita eri sidosryhmiä motivoivat usein hyvin eri tekijät (Barney et al. 2009).

Eri sidosryhmillä on usein ristiriitaisia tavoitteita sekä eriävää tietoa, jolloin arvosta voi tulla usein vaikeaa arvioida. Sidoryhmien eriävien kannustimien vuoksi arvon arvioiminen varsinkin konsolidoidulla yritystasolla voi olla hyvin haastavaa. Vaikka nämä vaikutukset ovat jossain määrin väistämättömiä, niiden sisällyttäminen arvonmäärittämiseen voi johtaa objektiivisempiin tuloksiin. (Biffi et al. 2006, s. 41 - 45) Arvonmäärittäminen onkin muuttumassa henkilökohtaisista näkemyksistä enemmänkin ryhmäpohjaiseksi, jota usein kutsutaan sidoryhmien arvolupaukseksi (stakeholder value proposition). Sidoryhmien yhteinen konsensus muodostetaan usein lukuisten eri keskusteluiden pohjalta. Kun eri sidoryhmien välillä on saatu muodostettua yhteinen konsensusnäkemys arvosta, arvon komponentteja voidaan alkaa mittaamaan käyttämällä sovitteja menetelmiä ja asteikkoja. (Biffi et al. 2006, s. 157) Tourwé et al. (2009) tutkimuksen havainnot tukevat väitettä ajattelutavan muutoksesta, jossa arvonmäärittäminen on enenevässä määrin sosiaalista vuorovaikutusta yhteistyössä kaikkien eri sidoryhmien kesken.

Rodriguez et al. (2018) soveltavan tutkimuksen tulokset tuovat hieman edeltävistä poikkeavan näkökulman, koska tuloksissa ei huomattu suurta eroa eri sidoryhmien näkemysarvoista. Tutkimuksen yhtenä tavoitteena oli tunnistaa sekä yhteiset että eroavat arvon tekijät eri sidoryhmäjoukoilta, jotka osallistuvat arvoperusteiseen ominaisuuk-

sien valintaan. Dataa kerättiin empiirisissä haastatteluissa suuren telekommunikaatioalan case-yrityksen kymmenen eri avainsidosryhmän välillä, johon kuuluivat muun muassa strategiset tuotepäälliköt, tuoteomistajat, projektipäälliköt ja tekniset johtajat. Tutkimuksen tuloksena saatiin ymmärrys, että ainakaan valitun case-yrityksen kohdalla eri sidosryhmien arvoperspektiivit eivät olleet toisistaan poikkeavia, vaan toisiaan täydentäviä ja edustivat erilaisia päätöksentekoperspektiivejä. Kuitenkin myös joitain pieniä konfliktilähteitä eri sidosryhmien välillä löydettiin tuotteen laatuun liittyvistä näkökohdista (ts. mikä laatutaso on riittävä). (Rodriguez et al. 2018) Vielä ajankohtaisemmassa Rodriguez et al. (2020) tutkimuksessa kuitenkin tutkimuksen otantakokoa kasvattamalla pystyttiin tunnistamaan kaavoja eri sidosryhmien arvofokusten välillä (esim. liiketoimintasidosryhmät painottivat enemmän asiakasarvodimension alaisia arvon tekijöitä kuin muut sidosryhmät).

### **4.2.3 Arvon moniulotteinen luonne**

Yksi yleisimmin mainituista arvonmäärityksen haasteista on arvon moniulotteinen luonne. Arvoa ei voida suoraan mitata samalla tavalla kuin esimerkiksi massaa, tilavuutta ja aikaa. Myös monet arvokäsitteen sisältämät asiat, kuten riski ja laatu, ovat samalla tavalla moniulotteisia asioita. (Biffi et al. 157 – 159) Marciuska et al. (2014) mainitsevat myös, että eri sidosryhmien kuten asiakkaiden on vaikea arvioida havaitsemaansa arvoa sen moniulotteisen luonteen takia. Arvon moniulotteinen luonne ei kuitenkaan ole vain ohjelmistotuotannon arvonmääritykselle ominainen haaste, vaan esimerkiksi Ulaga ja Chacour (2001) käsittelevät arvon moniulotteisuutta liiketoiminnan alalla. Ulagan ja Chacourin (2001) tutkimus osoittaa, että arvoa tulisi mitata ja arvioida moniulotteisena rakenteena.

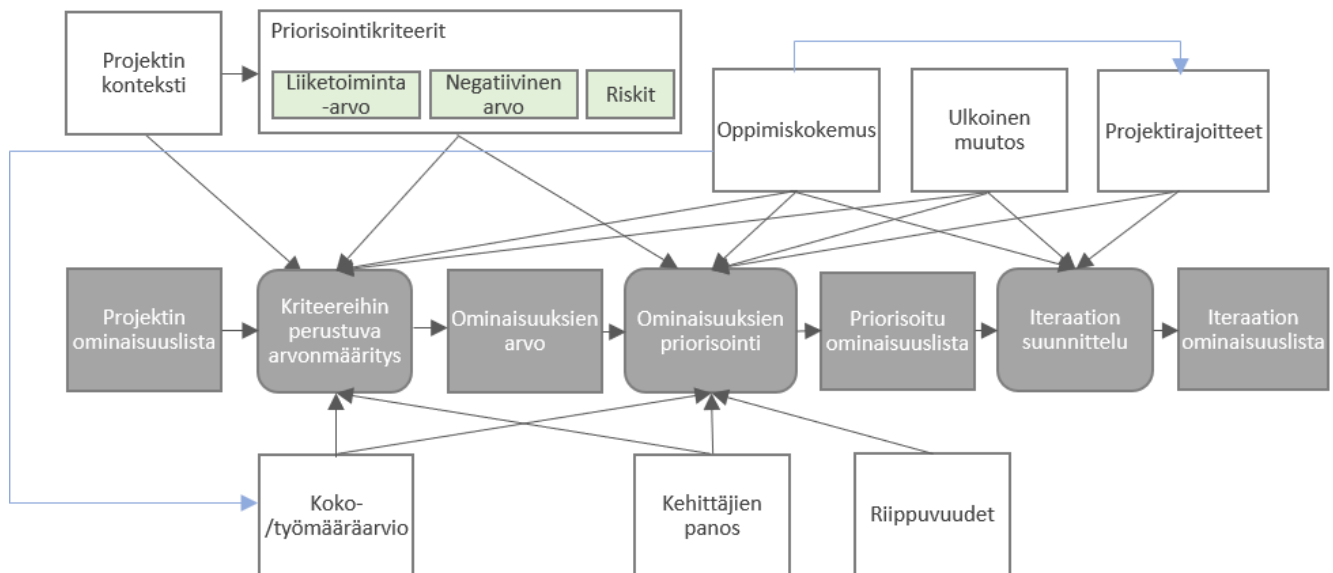
Useissa ohjelmistotuotannon ominaisuuksien arvottamista käsittelevissä artikkeleissa kerrotaan arvonmäärityksen olevan haastavaa arvon moniulotteisen luonteen takia, mutta harvassa tutkimuksessa käsitellään vaihtoehtoisia ratkaisuja tämän huomioon ottamiseksi. Biffi et al. (2006, s. 159) mukaan moniulotteisen luonteensa takia arvon arviointi vaatii useampia näkymiä tarkasteltavaan kohteeseen. Nämä näkymät myöhemmin summataan arvioitavana olevan kohteen kokonaisvaltaiseksi näkemykseksi. Esimerkiksi ohjelmiston laatua määrittäessä laatu voidaan jakaa eri laatutekijöihin, joille annetaan painoarvot niiden suhteellisen merkityksen mukaisesti (Biffi et al. 2006, s. 159). Täten voidaan päätellä, että esimerkiksi ohjelmistojen ominaisuuksien arvonmäärityksessä arvon jakaminen alaluvussa 3.3 käsiteltyihin arvotekijöihin tavoittelee muun muassa juuri arvon moniulotteisuuden huomioon ottamista.

### 4.3 Ohjelmisto-ominaisuuksien arvonmääritys

Ohjelmiston ominaisuuksien arvonmääritys on hyvin tärkeä päätöksentekoa tukeva osatehtävä. Lehtolan ja Kauppisen (2006) mukaan monet ominaisuuksien valinnan ja priorisoinnin haasteet johtuvat siitä, että yksittäisten ominaisuuksien arvoa ei pystytä arvioimaan luotettavasti. Ilman ominaisuuden todellisen arvon huolellista määrittämistä ominaisuuksien priorisointijärjestykset perustuvat vain karkeisiin arvauksiin (Lehtola & Kauppinen 2006). Mendes et al. (2017) tukevat väitettä, että arvoa pitää arvioida mahdollisimman luotettavasti ennen kuin siitä voi olla hyötyä osana päätöksentekoprosessia. He ehdottavat, että arvonmäärityksen prosessin tulisi alkaa sidosryhmien arvottamien arvo-tekijöiden keräämisellä. Edellinen toteutettaisiin käyttämällä laadullisen tutkimuksen menetelmiä, kuten avainsidosryhmien yksilöllisiä haastatteluja tai kohderyhmätapaamisia, jolloin sidosryhmät pyydetään määrittelemään myös, miten valittavien ominaisuuksien odotetaan vaikuttavat kuhunkin arvon tekijään (Mendes et al. 2017).

Ohjelmiston ominaisuuksien arvonmääritys on monesta syystä hyvin haastavaa. Edellisessä alaluvussa esitettyjen yleisemmän tason ohjelmistotuotannon komponenttien arvonmäärityksen haasteiden lisäksi ominaisuuksien arvonmääritystä vaikeuttaa esimerkiksi se, että asiakkaan käsitys ominaisuudesta ei ole staattinen, vaan vaihtelee ajan myötä. Marciuska et al. (2014) mukaan joskus erittäin arvokkaana pidetty ominaisuus voi menettää arvoaan tai ominaisuuden arvo voi jopa mitätöityä kokonaan ajan myötä. Ominaisuuksien arvon dynaamisen luonteen takia arvoa tulisi jatkuvasti validoida asiakkaan mieltymysten perusteella (Wittern & Zirpins 2011). Favare (2002) myös huomauttaa, että ominaisuuksien täydellisen arvoanalyysin toteuttaminen vaatii aika- ja resurssi-investointeja ja on usein vaikeampaa kuin itse suunnittelu- tai toteutustyö. Favaren (2002) mukaan myös ominaisuuksien täydellisiä taloudellisia vaikutuksia ei voida tietää ennen koko kehitysjakson elinkaaren päättymistä. Johansson et al. (2015) ominaisuuksien kehitysvaihetta aikaisempiin aspekteihin keskittyvä tutkimus tukee väitettä, että ominaisuuksien arvoa voidaan ainoastaan arvioida eri analyysitekniikoilla ennen ominaisuuksien toteuttamista, ja odotettu arvo voidaan vahvistaa vasta pidemmän ajan käytön seurauksena.

Tämän työn systemaattisilla hauilla alan kirjallisuuteen saatiin hyvin vähän tietoa ominaisuuksien ja vaatimusten arvottamisesta erikseen. Yleisesti ominaisuuksien arvonmäärittämisestä ei puhuttu kirjallisuudessa lainkaan, vaan arvonmäärittäystä pidettiin kiinteänä osana ominaisuuksien priorisointimenetelmiä. Jokainen ominaisuuksien priorisointimenetelmä tavoittelee jollain tekniikalla ominaisuuksien asettamista arvojärjestykseen, eli niissä jokaisessa toteutetaan myös jollain tasolla ominaisuuksien arvonmäärittäystä. Arvonmäärittäminen tapahtuu kyseisissä tapauksissa ennen arvoperusteista päätöksentekoa, minkä vuoksi tässä työssä on katsottu tarkoituksenmukaiseksi erotella mahdollisuuksien mukaan kyseiset vaiheet toisistaan. Myös esimerkiksi Bakalova et al. (2011) ketterän kehityksen priorisointiprosessin käsitteellisessä mallissa ominaisuuksien arvonmäärittäminen on kuvattu priorisointia edeltävänä vaiheena. Bakalova et al. (2011) priorisoinnin prosessimalli on esitetty kuvassa 3.



**Kuva 3.** Priorisoinnin prosessimalli ketterässä kehityksessä (mukailtu lähteestä Bakalova et al. 2011)

Kirjallisuudessa mainitut yleisimmät ominaisuuksien arvonmäärittämisprosessit alkavat usein ominaisuuksien oleellisten arvotekijöiden tunnistamisesta (Biffel et al. 2006, s. 159; Bakalova et al. 2011). Arvotekijöistä käytetään arvonmäärittämisessä ja päätöksenteon kontekstissa myös nimityksiä arvokomponentit, arvokriteerit ja priorisointikriteerit. Tässä työssä alaluvussa 3.3 käsitellään ominaisuuksien yleisiä arvotekijöitä ja kyseisen alaluvun taulukkoon 2 on koostettu empiirisissä tutkimuksissa esiintyviä ominaisuuksien arvotekijöitä. Yleisessä arvonmäärittämisprosessissa arvotekijöiden tunnistamisen jälkeen arvotekijöille annetaan suhteelliset painoarvot niiden tärkeyden mukaan (Biffel et al. 2006, s. 159). Tämän jälkeen kaikkia päätöksenteon kannalta oleellisia sidosryhmiä pyydetään arvottamaan kunkin ominaisuuden kohdalla jokainen arvotekijä tiettyä pisteytysasteikkoa

käyttäen. Lopuksi kunkin ominaisuuden kohdalla jokaisen arvotekijän saama pistearvo kerrotaan kyseisen arvotekijän painoarvolla. Edellinen pisteytysmenetelmä ei kuitenkaan kykene tarjoamaan ominaisuuksien absoluuttista arvoa, vaan ilmaisee arvon suhteellisen arvona, jolloin sitä voidaan vertailla muiden ominaisuuksien saamaan arvoon.

Saliu ja Ruhe (2005) ehdottavat ominaisuuksien arvonmäärittämiseen perinteisten sidosryhmiltä kerätyn subjektiivisen ominaisuuden arvopisteytyksen lisäksi myös ohjelmistokehitysyhtiön toteuttamaa sidosryhmien tärkeyden arvottamista. Tässä tapauksessa esimerkiksi sekä sidosryhmät voidaan arvottaa pisteskaalalla yhdestä yhdeksään, että sidosryhmät voidaan pyytää arvottamaan ominaisuudet samalla skaalalla. Tällöin ominaisuuden arvonmäärittäminen ottaa myös sidosryhmien suhteellisen merkityksen huomioon. Ehdotettu yhdeksän pisteen skaala ei suinkaan ole ainoa mahdollinen arvottamisessa käytettävä mittaus tapa, vaan arvoa voidaan arvioida myös taloudellisesti tai muilla arvon mittareilla. (Saliu & Ruhe 2005)

Tourwé et al. (2009) mukaan yrityksen kyky määrittää ominaisuuksien arvoa riippuu siitä, kuinka helposti yritys osaa ilmaista, mitä arvo tarkoittaa heidän kontekstissaan. Tietylle asiakkaalle kustomoituja ohjelmistoja tuottavalle yritykselle arvo voi tarkoittaa eri asiaa kuin kaupallisesti saatavana olevia ohjelmistotuotteita tarjoavalle yritykselle. Ensimmäisessä tapauksessa asiakas voi olla käytettävissä arvioimaan ominaisuuksien arvoa ja kommentoimaan seuraavaa tuotejulkaisua. Jälkimmäisessä tapauksessa eri asiakkailla voi olla erilaisia toiveita ja mielipiteitä, jolloin on yleensä vaikeampaa ennustaa, kuinka seuraava julkaisu otetaan markkinoilla vastaan. (Tourwé et al. 2009)

## **4.4 Nykyiset ominaisuuksien arvonmäärittäminen menetelmät**

Kuten ominaisuuksien arvonmäärittämisestä erillisenä toimintoja, ei myöskään ominaisuuksien arvonmäärittäminen menetelmistä löytynyt paljoakaan tietoa kirjallisuuskatsauksen hauilla. Kirjallisuudessa puhutaan ominaisuuksien priorisointimenetelmistä, joita voidaan käytännössä pitää myös arvonmäärittäminen menetelminä, sillä ne usein sisältävät ominaisuuden arvonmäärittäminen. Tässä alaluvussa käsitellään lyhyesti priorisointimenetelmistä eroavia vain arvonmäärittäminen keskittyviä menetelmiä. Alaluvussa 4.4.2 käsitellään hieman tarkemmin objektiivisuutta tavoittelevan ominaisuuksien käyttöasteeseen perustuvaa arvonmäärittäminen.

### **4.4.1 Yleisiä ominaisuuksien arvonmäärittäminen menetelmiä**

Yksinkertaiset ominaisuuksien arvonmäärittäminen (myös priorisoinnin) menetelmät, kuten sidosryhmien toteuttamat 1-10 pisteytys tai MoSCoW-metodi (täytyy olla, pitäisi olla,

voisi olla, ei tarvitse olla) eivät ole riittäviä menetelmiä havaitsemaan ominaisuuksien todellista arvoa. Kyseiset subjektiiviset menetelmät usein virheellisesti olettavat, että sidosryhmät ovat taitavia priorisoimaan ominaisuudet oikein todellisen liiketoiminnallisen arvon mukaisesti. Nämä tekniikat eivät yleensä yksinään ole sopivia erilaisille herkkyysanalyysille (esim. liiketoimintatavoitteiden muutosten vaikutus ominaisuuksien arvoon) tai vertaileville analyysille, joissa otetaan huomioon jo toteutetut ominaisuudet ja miten uudet tai muuttuneet ominaisuudet vaikuttavat niiden arvoon. (Kukreja 2013) Edelliset ominaisuuksien arvon arviointimenetelmät myös antavat hyvin usein tasapisteitä ominaisuuksien välille (esim. täytyy olla -ominaisuudet MoSCoW-metodin tapauksessa), jolloin vaaditaan hyvin usein useita arvon arvioinnin kierroksia (Kukreja et al. 2012).

Finanssipuolen investointilaskelmista tuttu nettonykyarvon laskenta on hyvin yleinen menetelmä myös ohjelmistojen ominaisuuksien arvonmäärityksessä, varsinkin jos ominaisuuksien hyödyt ovat helposti taloudellisesti mitattavia. Kustannukset ja hyödyt esiintyvät usein eri ajankohtina ja ne onkin diskontattava ottamaan huomioon rahan aika-arvon peruseriaate, jonka mukaan raha on nykypäivänä arvokkaampaa kuin tulevaisuudessa tavanomaisissa taloudellisissa olosuhteissa. Käytettävä sopiva diskonttokorko riippuu hankkeen riskistä sekä vaihtoehtoisten aloitteiden odotetusta tuotosta. Nämä tekijät on analysoitava diskonttokoron määrittämiseksi. (Biffel et al. 2006, s. 40) Edellä kuvatussa nettonykyarvon laskentaan perustuvassa ominaisuuksien arvottamisessa on kuitenkin lukuisia ongelmia varsinkin nykyisessä nopeasti muuttuvassa epävarmassa ympäristössä. Khurumin ja Barney'n (2009) mukaan ehkä ongelmallisinta kyseisessä menetelmässä on, että se virheellisesti olettaa riskitasojen ja tuottojen olevan staattisia. Nettonykyarvo myös joustamattomasti olettaa, että olemassa on vain kaksi skenaariota: sijoita ominaisuuteen tässä hetkessä tai älä sijoita koskaan (Khurum & Barney 2009).

Khurum ja Barney (2009) ehdottavat nettonykyarvoon perustuvan ominaisuuksien arvottamisen tilalle reaaliopitoteoriaan (Real Options Theory) perustuvaa arvonmääritystä. Reaaliopitot perustuvat taloustieteestä tuttuihin optioihin. Optiot antavat osapuolelle oikeuden, mutta ei velvollisuutta ostaa tai myydä omaisuusvaraa tietyssä tulevana päivänä tai sitä aikaisemmin toisen osapuolen asettamaan hintaan. Reaaliopitoteoria tarjoaa ominaisuuksien arvonmääritysmenetelmän, jonka avulla voidaan analysoida erilaisia tulevaisuuden mahdollisuuksia ja tehdä joustavampia päätöksiä kuin ominaisuuteen sijoittaminen nyt tai ei koskaan. (Khurum ja Barney 2009) Reaaliopitot myös ottavat epävarmuuden eksplisiittisesti huomioon sekä antavat mahdollisuuden arvottaa ominaisuudet uudelleen markkina- tai teknologiamuutosten tapauksissa (Biffel et al. 2006, s. 61; Khurum ja Barney 2009).



Mohamedin ja Wahban (2008) esittelemä Value Point -menetelmä on harvoja kirjallisuudessa esiintyneitä priorisointimenetelmien ulkopuolisia ainoastaan ohjelmisto-ominaisuuksiin keskittyviä arvonmäärittämenetelmiä. Myös Mendes et al. (2017) nostavat Value Point -menetelmän esiin harvojen ominaisuuksien arvonmäärittämenetelmien joukosta. Monet nykyisistä ominaisuuksien arvonmäärittäykseen sovellettavista menetelmistä, kuten edellä mainitut nettohyötyarvo ja reaaliopioiteoria mittaavat ainoastaan taloudellista arvoa, kuten kustannussäästöjä ja liikevaihdon kasvua. Value Point -menetelmä ottaa huomioon taloudellisen arvon lisäksi myös muut kuin taloudelliset arvon komponentit, kuten markkinaosuuden kasvun, laadun ja luotettavuuden kasvun, riskin pienemisen, terveyden tai turvallisuuden parantumisen sekä asiakastoiminnan paranemisen. (Mohamed & Wahba 2008)

Value Point -menetelmä muistuttaa jonkin verran edellä kuvattuja yleisiä ominaisuuksien arvotusmenetelmiä, jossa ensin tunnistetaan ominaisuuksien arvotekijät, minkä jälkeen sidosryhmät subjektiivisesti arvotavat ominaisuudet arvotekijöiden pohjalta pisteytysmenetelmiä hyödyntäen. Value Point -menetelmä käyttää syötteenä toteutettavien ominaisuuksien listaa. Arviointiprosessissa jokaiselle päätöksenteossa mukana olevalle avainsidosryhmälle annetaan painoarvo sen mukaan, kuinka tärkeitä he ovat organisaation ja kyseisen ohjelmistotuotteen kannalta. Sidosryhmät arvioivat tämän jälkeen annettujen ominaisuuksien arvoa käyttämällä 14 ominaisuuksien yleispiirrettä, joille jokaiselle annetaan arvo neljän pisteen skaalaa käyttämällä. Kyseiset 14 ominaisuuksien yleispiirrettä muistuttavat hyvin paljon yleisiä laatutekijöitä, sisältäen esimerkiksi kiireellisyyden, soveltuvuuden, johdonmukaisuuden ja ymmärrettävyyden. Edellisten vaiheiden jälkeen arvopisteiden laskun kohteelle olevan projektin tyyppi otetaan huomioon. Projektit jaotellaan tyypeiltään uusiin projekteihin ja vanhojen projektien paranteluihin. Nämä eri syötet otetaan huomioon arvopisteiden laskemisessa, jonka tuloksena lopulta saadaan ominaisuuksille laskettua vertailtavat ranking-arvot. (Mohamed & Wahba 2008)

Value Point -menetelmä ei kuitenkaan esitä ominaisuuksien arvoa absoluuttisena arvona, vaan määrittää arvon suhteellisena arvona verrattuna muihin ominaisuuksiin. Mendes et al. (2017) argumentoivat myös, että Value Point -menetelmää ei voida pitää kokonaisvaltaisena arvonmäärittäksen mallina, vaan menetelmä tarjoaisi ainoastaan mitaustavan kunkin ominaisuuden arvonmäärittäykseen. Kuten suurin osa muistakin arvonmäärittämenetelmästä, Value Point -menetelmä perustuu myös sidosryhmien toteuttamiin subjektiivisiin arvioihin. Seuraavassa toisen tason alaluvussa käydään läpi ominaisuuksien käyttöön perustuvaa objektiivisempaa arvonmäärittämenetelmää.

#### 4.4.2 Ominaisuuden käyttöasteeseen perustuva arvonmääritys

Ominaisuuksien käyttöön perustuvan arvonmäärityksen voidaan argumentoida olevan päätöksenteon kannalta tarpeetonta, sillä kyseisessä menetelmässä ominaisuuksien arvoa voidaan mitata vasta ominaisuuksien toteuttamisen jälkeen. Täten esimerkiksi ominaisuuksia priorisoidessa seuraaviin julkaisuihin on mahdotonta hyödyntää tietoa näiden toteuttamattomien ominaisuuksien käyttöasteista. Jo toteutettujen ominaisuuksien käytön mittauksesta voidaan kuitenkin saada tärkeää tietoa ominaisuuksien arvosta myös päätöksenteon tueksi. Ensinnäkin ominaisuuksien käyttöasteista voidaan saada tärkeää dataa samankaltaisten ominaisuuksien arvottamisessa tulevien ominaisuuksien priorisoinnin kontekstissa. Toiseksi, ominaisuuksien arvoon perustuva päätöksenteko ei rajoitu pelkästään tulevien ominaisuuksien priorisointiin, vaan ominaisuustulvan välttämiseksi myös jo toteutettujen ominaisuuksien arvoa tulee monitoroida. Rein ja Münch (2013) ehdottavat case-tutkimuksensa pohjalta ominaisuuksien ennakoituun käyttöön perustuvaa arvon määritystä toteuttamalla ohjelmistoon valeostoksia (mock purchase). Valeostoksissa käyttäjä sitoutuu lataamaan tai ostamaan tietyn ominaisuuden sovelluksen sisällä, mutta ominaisuutta ei ole oikeasti toteutettu eikä sitä toimiteta käyttäjälle. Valeostosten perusteella saadun arvonmäärityksen myötä saataisiin tietoa myös ominaisuuksien priorisointia varten jo ennen ominaisuuksien toteuttamista. Valeostosten käytön hyödyt tulevat kuitenkin asiakastyytyvyyden kustannuksella, minkä vuoksi myös Rein ja Münch (2013) suosittelevat käyttämään menetelmää vain hyvin perusteltujen ja päätöksenteon kannalta tärkeiden ominaisuuksien kohdalla. Lisäksi valeostokset soveltuvat vain hyvin rajatulle joukolle ohjelmistotuotteita, kuten mobiilisovelluksille.

Marciuska et al. (2013b) exploratiivisessa case-tutkimuksessa sekä samojen kirjoittajien jatkotutkimuksessa (Marciuska et al. 2014) tutkitaan, kuinka ominaisuuksien käyttö asiakkailla liittyy asiakkaiden kokemaan arvoon ominaisuudesta ja valaistaan tähän suhteeseen vaikuttavia tekijöitä. Tutkimuksen pohjalla on Woodruffin (1997) tutkimuksen toteamus siitä, että asiakkaan kokema arvo ominaisuudesta liittyy läheisesti ominaisuuden käyttöasteeseen. Kummassakin Marciuska et al. (2013b; 2014) edellä mainitussa tutkimuksessa pääfokus on asiakkaan kokemassa arvossa, eikä siis ota huomioon kokonaisvaltaisemmin muiden sidosryhmien kokemaa arvoa. On myös syytä huomioida, että molemmissa case-tutkimuksissa käytetty empiirisessä tutkimuksessa samaa web-pohjaista sovellusta, joten tutkimuksen tulokset eivät sellaisinaan ole yleistettävissä kaikille erilaisille ohjelmistotuotteille. Case-tutkimusten tavoitteena on luoda objektiivisempaa mitaustapaa ominaisuuksien arvolle, kuin mitä perinteisillä haastattelututkimuksilla on tähän mennessä saatu aikaan.

Marciuska et al. (2013b; 2014) tutkimuksissa ominaisuuden käyttöä seurattiin automaattisesti ohjelmistosovellukseen asetetun koodin avulla. Jokaiselle dokumenttioliomallille (Document Object Model, DOM) annettiin uniikki nimike käyttäjien hyödyntämien ominaisuuksien seurantaan varten. Ensimmäiseen Marciuska et al. (2013b) tutkimukseen osallistui 19 web-sovelluksen aktiivista käyttäjää. Jokainen tutkimukseen osallistunut käyttäjä identifioitiin erikseen ja heidän käyttämiään ominaisuuksia seurattiin kahden kuukauden ajan. Tämän jälkeen heitä pyydettiin myös arvottamaan sovelluksen eri ominaisuudet käyttäen vaatimusten priorisoinnissa yleistä sadan pisteen menetelmää (100-point method), jossa käyttäjälle annetaan käyttöönsä sata pistettä, jotka heidän tulee jakaa ominaisuuksien kesken heidän kokeman arvon perusteella. Lopulta tutkimuksessa verrattiin käyttäjien sadan pisteen menetelmällä kerättyä ominaisuuksien koettua arvoa automaattisesti kerättyyn tietoon ominaisuuksien käyttökerroista kyseisillä käyttäjillä.

Marciuska et al. (2013b) tutkimuksen tulokset vahvistivat, että on hyvin vaikeaa mitata asiakkaiden havaitsemaa arvoa pelkkien kyselyjen avulla, koska heillä saattaa olla joitain implisiittisiä oletuksia siitä mitä arvo tarkoittaa. Tutkimuksen tulosten perusteella suurimmassa osassa tapauksia, mitä suurempi käyttöaste ominaisuudella oli, sitä arvokkaampi se käyttäjälle oli myös sadan pisteen metodilla saatujen tulosten perusteella. Toisaalta tutkimuksessa taas ei löydetty suurta korrelaatiota pienen käyttöasteen ja vähäisemmän arvon välillä. Haastattelut osoittivat, että jotkut käyttäjät pitivät arvokkaina myös ominaisuuksia, joita he käyttivät hyvin harvoin. Myös, vaikkakin harvemmin, joissain tapauksissa käyttäjät käyttivät ominaisuuksia useasti, mutta eivät antaneet niille suurta arvoa. Näille molemmille tapauksille jatkohaastatteluissa käyttäjät ilmoittivat viisi eri syytä, miksi tietyn ominaisuuden käyttöaste ja koettu arvo eivät olleet linjassa keskenään: (Marciuska et al. 2013b)

1. Jotkut ominaisuudet ovat hyvin arvokkaita, vaikka niitä ei käytetäkään usein (esim. case-tutkimuksen elokuvien ehdotus -sovelluksessa kohdalla rikkinäisen trailerin ilmoittaminen koettiin arvokkaaksi, vaikka sitä käytettiin harvoin)
2. Käyttäjät ilmoittivat tehneensä virheen ominaisuuksien arvottamisessa (antoivat liian suuren arvon)
3. Jotkut ominaisuudet olivat osa prosessia eikä niitä ollut mahdollista ohittaa (esim. jos merkkasi elokuvan nähdyn, aukesi selaimelle pop-up -ikkuna, jonka avulla elokuva piti arvostella)
4. Ominaisuus merkattiin pienelle arvolle, vaikka sitä oli käytetty, koska ajateltiin että tulevaisuudessa sitä tulotaisiin käyttämään vähemmän

5. Jotkut ominaisuudet olivat käyttäjille tärkeitä eri syistä kuin mihin ominaisuus oli alun perin tarkoitettu.

Tämän Marciuska et al. (2013b) toteuttaman tutkimuksen tulokset osoittivat kuitenkin, että pelkästään ominaisuuksien käyttö yksinään ei riitä arvottomien ominaisuuksien tunnistuksessa. Tästä syystä Marciuska et al. (2014) toteuttivat uuden case-tutkimuksen kohdistuen samaan web-sovellukseen, mutta tällä kertaa ottivat mukaan 12 aktiivisen loppukäyttäjän lisäksi kolme sovelluksen kehittäjää. Sovelluksen kehittäjien tehtävänä oli arvioida suunniteltua ominaisuuksien käyttöastetta ominaisuuksien kehittämisvaiheessa. Tämän lisäksi tutkimukseen otettiin mukaan ominaisuuksien käytön toteutuma suhteellisenä mittarina verrattuna kaikkien ominaisuuksien käyttökertoihin. Näiden kahden tiedon perusteella tutkimuksen tavoitteena oli luoda ominaisuuksien käytölle raja-arvo suhteessa niiden suunniteltuun käyttöön, jonka avulla voitaisiin tunnistaa marginaalista arvoa tuottavat ominaisuudet, jotka tulisi poistaa sovelluksesta.

Marciuska et al. (2014) suorittama case-tutkimuksen tulokset olivat linjassa aikaisemman tutkimuksen kanssa siinä, että arvottomampia ominaisuuksia käytettiin useimmissa tapauksissa vähemmän kuin arvokkaita ominaisuuksia. Siksi yritykset, jotka ovat kiinnostuneita marginaalista arvoa tuottavien ominaisuuksien löytämisestä tulisi ensisijaisesti keskittyä niihin ominaisuuksiin, joita käytetään harvoin. Tutkimuksen mukaan näiden ominaisuuksien tunnistamisen lisäksi on tarkoituksenmukaista kerätä kehittäjiltä tietoa ominaisuuksien suunnitellusta käytön yleisyydestä. Tutkimuksen mukaan ne ominaisuudet, joita käytetään todellisuudessa vähemmän kuin kehittäjät ovat suunnitelleet, ovat kandidaatteja järjestelmästä poistolle. Ominaisuuksia tulisi kuitenkin monitoroida riittävän pitkän ajan ennen kyseisiä ominaisuuksien poistamispäätöksiä. (Marciuska et al. 2014)

## 5. OMINAISUUKSIEN ARVON HYÖDYNTÄMINEN PÄÄTÖKSENTEOSSA

Ohjelmistoyritykset ovat siirtymässä asteittain arvoperusteiseen päätöksentekoon (Alahyari et al. 2017; Rodriguez et al. 2018). Ohjelmistotuotteiden koon ja monimutkaisuuden kasvaessa samalla on kasvanut myös jokaisen ohjelmistoon liittyvän päätöksenteon vaikutus koko yrityksen kannattavuuteen (Gorschek et al. 2010; Khurum et al. 2013). Freitas et al. (2017) mukaan säilyttääkseen kilpailukykyä, innovatiivisuutensa ja kasvaakseen yritysten tulisi hyödyntää arvoperusteista päätöksentekoa, jossa päätökset optimoidaan yrityksen kokonaisarvonluonnin suhteen.

Tässä luvussa tarkastellaan ensin inkrementaalista ohjelmistokehitystä sekä julkaisu-suunnittelua, joiden kontekstissa ohjelmisto-ominaisuuksiin liittyviä päätöksiä nykyisessä ohjelmistotuotannon ympäristössä perinteisesti tehdään. Tämän jälkeen luvussa keskitytään käsittelemään ominaisuuksien arvoperusteista valintaa ja priorisointia sekä nykyisiä priorisointimenetelmiä, joissa kirjallisuus on tunnistanut arvon olevan keskiössä.

### 5.1 Inkrementaalinen ohjelmistokehitys ja julkaisusuunnittelu

Asiakkaiden todellisten tarpeiden tyydyttämiseksi sekä suuremman joustavuuden saavuttamiseksi yhä useampi ohjelmistoalan yritys on omaksumassa inkrementaalisen ohjelmistokehityksen ja -toimittamisen periaatteita. Inkrementaalisen kehittämisen mallissa pitkien kehitysjaksojen ja suurikokoisten julkaisujen sijasta järjestelmään toteutetaan pienempiä peräkkäisiä julkaisuja. Kyseisessä mallissa ohjelmistoon toimitetaan ominaisuuksia jatkuvasti uusien julkaisujen kautta. (Greer & Ruhe 2004)

Inkrementaalisen kehityksen ja toimittamisen mallin avulla voidaan ohjelmistotuotannon saralla saavuttaa lukuisia etuja perinteiseen vesiputousmalliin verrattuna. Inkrementaalisisessa toimituksessa ominaisuuksien toteutukset voidaan priorisoida siten, että suurinta arvoa tuovat ominaisuudet voidaan toimittaa järjestelmään ensin, jolloin myös niiden edut saadaan aikaisemmin käyttöön. Tämän myötä myös budjetin tai ajan loppuessa kesken on todennäköisempää, että vähemmän arvoa tuovat ominaisuudet jäävät toteuttamatta. Jatkuvasa inkrementaalisisessa järjestelmän toimittamisessa asiakkaat myös saavat osan järjestelmästä jo aikaisemmin käyttöönsä, jolloin ominaisuuksista on mahdollista saada palautetta jo aikaisemmassa vaiheessa. Myös kustannuksia sekä aikataulua on helpompi arvioida etukäteen, kun julkaisukoot ovat pienempiä. (Greer & Ruhe

2004; Ruhe & Saliu 2005) Greerin ja Ruhen (2004) mukaan tärkeimpänä jatkuvan toimittamisen mallin etuna kuitenkin on, että kyseinen inkrementaalinen lähestymistapa mahdollistaa paljon paremman reagoinnin ominaisuuksien muutoksiin ja tulevaisuuden lisäyksiin.

Inkrementaalisen ohjelmistokehityksen suunnittelua kutsutaan julkaisusuunnitteluksi (release planning) (Mc Elroy & Ruhe 2010). Julkaisusuunnittelu on tärkeä vaihe inkrementaalista ohjelmistokehitystä. Julkaisusuunnittelu koostuu päätöksenteosta, jossa määritellään mitkä ominaisuudet on toteutettava minkäkin julkaisun aikana (Ruhe 2003). Julkaisusuunnittelulla on merkittävä vaikutus ohjelmistotuotteiden kehityksen onnistumisen tai epäonnistumisen kannalta (Mc Elroy & Ruhe 2010). Ilman hyvää julkaisusuunnittelua kriittiset ominaisuudet saattavat lykkääntyä kehitysjakson loppupuolelle, ilman että muita ominaisuuksia tai julkaisuajankohtaa ehditään enää korjaamaan. Tämä tilanne voi johtaa tyytymättömiin asiakkaisiin, aikataulujen ja budjetin ylityksiin sekä markkinaosuuksien menetykseen. (Penny 2002) Jokainen julkaisu koostuu kokoelmasta ominaisuuksia, jotka muodostavat kokonaisen ja toimivan järjestelmän, joka tuottaa arvoa asiakkaalle. Julkaisusuunnittelu siis koostuu pääasiassa ominaisuuksien valinnasta ja priorisoinnista. Julkaisusuunnittelu on monimutkainen prosessi, joka sisältää eri sidosryhmien näkökulmat, kilpailevat tavoitteet sekä erityyppisiä rajoitteita. (Saliu & Ruhe 2005) Resurssi- ja aikarajoitteiden sekä sidosryhmien ristiriitaisten toiveiden vuoksi kaikkia ominaisuuksia ei voida toimittaa yhdessä julkaisussa (Tourwé et al. 2009). Täten arvokkaimmat ominaisuudet toimitetaan ensin, kun taas vähemmän arvoa tuottavien ominaisuuksien kehitys siirretään tuleviin julkaisuihin (Saliu & Ruhe 2005; Tourwé et al. 2009).

Julkaisusuunnittelun tavoitteena on tarjota optimaalinen osajoukko ominaisuuksia tietyssä julkaisussa (asiakkaalle toimitettavassa tuotteen versiossa) (Carlshamre 2002; Saliu & Ruhe 2005; Biffi et al. 2006, s. 185). Julkaisusuunnitteluprosessin tavoitteena on siis maksimoida saavutettava arvo tasapainottelemalla eri sidosryhmien näkökulmien sekä resurssi-, riski-, budjetti ja aikarajoitusten kanssa (Carlshamre 2002). Edellisten lisäksi Saliun ja Ruhen (2005) mukaan tulee optimaalista ominaisuuksien osajoukkoa määriteltäessä ottaa huomioon myös tärkeimmät tekniset rajoitteet. Näitä haasteita on vaikea ratkaista jopa keskikokoisissa ohjelmistojärjestelmissä. Julkaisusuunnittelu muuttuu yhä haastavammaksi kehittyvissä järjestelmissä, joissa on otettava huomioon myös jo olemassa olevat järjestelmän ominaispiirteet, kuten jo toteutettujen ominaisuuksien monimutkaisuus, koko, kriittisyys ja ymmärrettävyys. Suunnittelun laajuus ei saisi myöskään rajoittua vain seuraavaan julkaisuun. Organisaation tavoitteiden kannalta on tärkeää antaa vastaus myös kysymykseen, milloin tietyn ominaisuuden pitäisi olla julkaistu, jos sitä ei julkaista seuraavassa julkaisussa. (Saliu & Ruhe 2005)

Rodriquez et al. (2020) huomauttavat, että julkaisusuunnittelussa ei ole kyse vain seuraavan julkaisun lyhyen aikavälin arvon tarkastelusta, vaan ohjelmistotuotteen pitkäaikainen kehitys on otettava huomioon. Julkaisusuunnittelussa joudutaan tekemään jatkuvia kompromisseja asiakkaiden ja markkinoiden lyhyen aikavälin liiketoimintatavoitteiden sekä ohjelmiston pitkän aikavälin kehityksen välillä, jotta varmistettaisiin ohjelmistotuotteen kilpailukyky sekä lyhyellä että pidemmällä aikavälillä. Lindgren et al. (2008) mukaan valitettavan usein nykyään projektipäällikön katsotaan onnistuneen tehtävissään, jos hän onnistuu toteuttamaan julkaisun tai koko projektin ajallaan. Tämän vuoksi projektipäälliköt pyrkivät lyhyisiin markkinoillesaantiaikoihin, joka usein johtaa teknisen velan kasvuun sekä laadun heikkenemiseen (Lindgren et al. 2008). Myös tulevaisuuden epävarmuus ajaa ohjelmistoyrityksiä ottamaan vain lyhyen aikavälin arvon huomioon, joka osaltaan kasvattaa teknistä velkaa (Lindgren et al. 2008; Abad & Ruhe 2015). Abad ja Ruhe (2015) ehdottavat teknisen velan vähentämiseen epävarmoissa olosuhteissa reaaliop-tioteorian hyödyntämistä. Reaaliop-tioteoriaa on lyhyesti kuvailtu tässä työssä yleisten arvonmäärittämenetelmien joukossa alaluvussa 4.4.1.

Päätöksenteon kannalta julkaisusuunnittelu siis tarkoittaa sitä, että on määriteltävä ja tehtävä päätöksiä, mitkä ominaisuudet toimitetaan missäkin tulevassa julkaisussa. Greerin ja Ruhen (2004) mukaan kolme pääasiallista näkökohtaa tulee ottaa huomioon tehtäessä päätöksiä ominaisuuksien jaottelusta eri tulevien julkaisujen välille: ominaisuuksien tekniset ensisijaisuusrajoitukset, eri sidosryhmien tyypillisesti ristiriitaiset ominaisuuksien prioriteettijärjestykset sekä tasapaino vaadittujen ja käytettävissä olevien resurssien välillä. Tekniset ensisijaisuusrajoitukset liittyvät tilanteisiin, joissa tiettyä ominaisuutta ei voida toteuttaa ennen kuin jokin toinen tietty ominaisuus on toteutettu järjestelmään. Sidoryhmien eroavat prioriteettijärjestykset ominaisuuksille perustuvat usein ominaisuuksien havaittuun arvoon tai kiireellisyysasteisiin tietylle sidoryhmälle. Tekniset ensisijaisuusrajoitukset sekä yksittäisten sidoryhmien prioriteetit ominaisuuksien toteutusjärjestyksestä voivat olla myös keskenään ristiriidassa ja vaikeasti keskenään sovittavissa. (Greer & Ruhe 2004) Saliu ja Ruhe (2005) lisäävät edellisiin myös käyttöön liittyvät rajoitukset, jolloin ominaisuudet voitaisiin teknisesti toteuttaa eri julkaisuissa, mutta ominaisuudet eivät olisi käyttökelpoisia loppukäyttäjille ilman toisiaan.

Saliu ja Ruhe jakavat julkaisusuunnittelun kahteen eri lähestymisnäkökulmaan: 1) julkaisusuunnittelun taidenäkökulmaan, joka pohjautuu pääasiassa päätöksentekijöiden intuition, kommunikointiin sekä kykyyn tasapainotella ristiriitaisten tavoitteiden ja rajoitteiden välillä, sekä 2) julkaisusuunnittelun tiedenäkökulmaan, joka korostaa ongelman formalisointia sekä laskennallisten algoritmien hyödyntämistä parhaiden mahdollisten rat-

kaisujen löytämiseksi. Molemmat näkökulmat ovat tärkeitä julkaisusuunnittelujen tulosten onnistumisen kannalta. (Saliu & Ruhe 2005) Päätöksentekijöiden kokemukseen ja intuitioon perustuvat julkaisusuunnitelmat voivat pienemmissä ohjelmistokokonaisuuksissa toimia, mutta ominaisuuksien lukumäärän kasvaessa päätöksentekijöiden on hyvin vaikeaa pelkästään subjektiiviseen ajatteluun perustuvalla lähestymistavalla ottaa huomioon esimerkiksi sidosryhmien ristiriitaiset tavoitteet. Järjestelmien monimutkaisuus onkin johtanut julkaisusuunnitelmaongelmien formaalimpaan mallintamiseen objektiivisuuden saavuttamiseksi. (Ruhe & Ngo The 2005; Mc Elroy & Ruhe 2010) Mc Elroy ja Ruhen (2010) mukaan järjestelmälliset lähestymistavat tuottavat todennäköisemmin suunnitelmia, jonka avulla saavutetaan korkeampi liiketoiminnallinen kokonaisarvo.

Saliun ja Ruhen (2005) mukaan julkaisusuunnittelussa heti ominaisuusvaatimusten selvityksen ja ongelmanmäärittämisen jälkeen tulisi keskittyä saatavilla olevien resurssien arviointiin. Resursseja on aina rajallinen määrä ja resurssien arvioinnilla julkaisusuunnittelun kontekstissa pyritään määrittämään todennäköinen työn määrä, kustannukset ja ominaisuuden toteuttamiseen vaadittava kalenteriaika. Resurssien arviointi tuottaa pohjatietoa, jonka päälle lopullinen ominaisuuksien valikointi ja priorisointi voidaan rakentaa. Julkaisusuunnittelun kontekstissa tulee myös ottaa huomioon, että useimmissa tapauksissa ominaisuuden toteutus vaatii useiden jo toteutettujen komponenttien modifiointia. (Saliu & Ruhe 2005)

## **5.2 Ominaisuuksien valinta ja priorisointi arvoperusteisesti**

Hyvin harvat tutkimukset ovat keskittyneet arvoon ominaisuuksien valinnassa, vaikka ominaisuuksien valinta julkaisujen suunnittelussa on olennainen toiminto ohjelmistotuotteiden menestyksen kannalta (Wohlin & Aurum 2005; Rodriguez et al. 2018). Koko ohjelmistotuotteen arvo koostuu julkaisuille valittujen ominaisuuksien joukosta, joten ymmärrys arvon merkityksestä ominaisuuksien valinnan yhteydessä on tärkeää (Rodriguez et al. 2018). Bullockin (2000) mukaan 80 prosenttia ohjelmiston liiketoiminta-arvosta tulee vain 20 prosentista ohjelmisto-ominaisuuksia. Ominaisuusvaatimuksia on myös yleensä enemmän kuin mitä on mahdollista toteuttaa annetussa aikataulussa (Kukreja 2013; Cleland-Huang 2015). Ominaisuuksien hallintaprosessin kriittinen osa on sopivien ominaisuuksien osajoukon valinta eri sidosryhmien kilpailevien ja ristiriitaisten odotusten joukosta (Wieggers 1999). Tiettyjen ominaisuuksien sisällyttäminen tai poissulkeminen tuotteen etenemissuunnitelman tai julkaisujen suunnittelun kontekstissa vaikuttavat tuotteen arvoon ja tätä kautta usein vaikuttavat myös yrityksen elintärkeisiin prosesseihin, kuten tuotteiden hinnoitteluun ja markkinointistrategiaan (Rodriguez et al. 2020) Azar et al. (2007) mukaan varsinkin pienemmissä organisaatioissa ominaisuuksien valinta ja



priorisointi eivät vaikuta pelkästään ohjelmistotuotteen menestykseen, vaan voi olla kriittinen tekijä myös koko yrityksen selviytymiskyvyn kannalta.

Kirjallisuudessa tunnutaan olevan yksimielisiä siitä, että ominaisuuksien priorisoinnissa tulisi aina valita kokonaisarvoltaan arvokkaimmat ominaisuudet toteutettavaksi ja toimitettavaksi ensin (Lehtola et al. 2009; Kukreja et al. 2012). Varsinkin ohjelmistotuote-liiketoiminnassa arvokkaimpien ominaisuuksien onnistunut valinta seuraaviin tuotejulkaisuihin on yrityksen liiketoiminnan kannalta erittäin tärkeää (Lehtola et al. 2009). Kukreja et al. (2012) tutkimuksen mukaan ohjelmisto-ominaisuuksien priorisoinnissa on tarve uusille arvoperusteisille priorisointimenetelmille, joissa kokonaisvaltainen näkemys tuotetusta arvosta on keskiössä. Arvon kokonaisvaltainen huomioiminen ominaisuuksien priorisoinnissa on kuitenkin haastavaa. Barney et al. (2008) toteavat, että ohjelmistotuotteen arvonluominen riippuu kontekstista ja siksi sitä on vaikea ymmärtää. He löysivät tutkimuksessaan monta kontekstisidonnaista tekijää, kuten tuotteen kypsyyssasteen, markkinatilanteen sekä yrityksen kehityskäytännöt, jotka vaikuttavat ominaisuuksien valinnan kautta arvonluontiin.

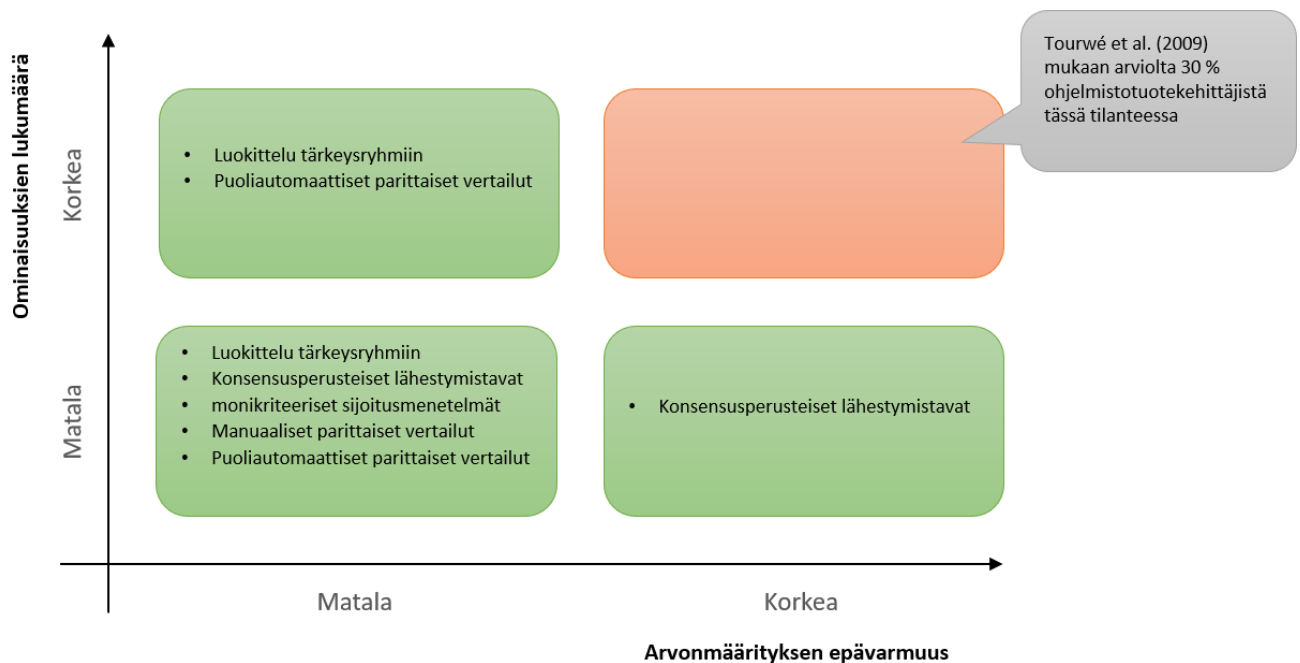
Arvoperusteisen ominaisuuksienhallinnan tutkimus ehdottaa menestyksekkään ominaisuuksien valinnan ja priorisoinnin pohjaksi kolmivaiheista prosessia: 1) tunnistetaan menestyskriittiset sidosryhmät, 2) kerätään menestyskriittisten sidosryhmien arvottamat arvotekijät ominaisuuksille ja 3) muodostetaan näistä arvotekijöistä yhteiset tavoitteet ohjelmistojärjestelmälle (Boehm 2003b). Ominaisuuksien priorisoinnin prosessista tulee kuitenkin hyvin monimutkaista ja haastavaa sidosryhmien ja ominaisuuksien lukumäärän kasvaessa (Tourwé et al. 2009).

Yleisimmät ominaisuuksien valinta- ja priorisointimenetelmät perustuvat eri järjestelmiin, jotka luokittelevat ominaisuudet prioriteettiltaan eri kategorioihin (esim. korkea, kohtalainen, matala) tai joissa ominaisuuksille jaetaan priorisoinnissa myöhemmin käytettäviä pisteitä. Nämä tekniikat ovat kuitenkin Cleland-Huangin (2015) mukaan naiiveja ja pisteytyksiin osallistuvat sidosryhmät pystyvät helposti vaikuttamaan näihin mielensä mukaisesti. Cleland-Huangin (2015) mukaan parempi lähestymistapa pyrkii konsensukseen kokoamalla keskeiset sidosryhmät yhteen, identifioimalla kompromissit määrittelemällä arvon tekijät ja lopulta saavuttamalla yhteisymmärryksen eri sidosryhmien välillä.

Ruhen ja Saliun (2005) mukaan ominaisuuksia voidaan priorisoida niiden arvon sekä kiireellisyyden perusteella. Kiireellisyydellä kirjoittajat viittaavat ominaisuuden markkinoillesaantiaikaan liittyviin asioihin. Tämä lähestymistapa on kuitenkin hieman ristiriidassa monen alan tutkimuksen kanssa, jossa ominaisuuden arvo kokonaisvaltaisena käsitteenä sisältää myös markkinoillesaantiajan. Ruhe ja Saliu (2005) ovat irrottaneet

arvokäsitteestä erillisiksi tekijöiksi myös esimerkiksi riskin sekä sijoitetun pääoman tuottoasteen, jotka tässä työssä kirjallisuuteen nojaten luetaan arvon osatekijöiksi.

Tourwé et al. (2009) jaottelevat ominaisuuksien valinta- ja priorisointimenetelmät kuuteen eri kategoriaan niiden toimintaperiaatteiden mukaisesti: 1) luokittelu tärkeysryhmiin, 2) konsensuskseen perustuvat lähestymistavat, 3) monikriteeriset sijoitusmenetelmät, 4) parittaiset vertailut, 5) äänestysjärjestelmät sekä 6) taloudelliset lähestymistavat. Heidän mukaansa sopivan priorisointimenetelmän valinta julkaisusuunnittelun kontekstissa riippuu pääosin kahdesta tekijästä: ominaisuuksien lukumäärästä sekä epävarmuudesta ominaisuuksien arvonnäilyksessä. Tutkimuksen mukaan kaikki edellä mainitut priorisointimenetelmien kategoriat soveltuvat tilanteeseen, jossa sekä ominaisuuksien lukumäärä että arvonnäilyksen epävarmuus ovat alhaisia. Jotkut menetelmät myös toimivat tilanteissa, joissa toinen edellisistä muuttujista on korkea. Huomionarvoista kuitenkin on, että tutkimuksen mukaan mitkään priorisointimenetelmät eivät sovellu tilanteisiin, joissa sekä ominaisuuksien lukumäärä että arvonnäilyksen epävarmuus ovat korkeita. Tutkimuksen mukaan arviolta noin 30 % ohjelmistointensiivisten tuotteiden kehittäjäyrityksistä on tässä kyseisessä tilanteessa. (Tourwé et al. 2009) Kuva 4 esittelee edellä esiteltyjen menetelmäkategorioiden soveltuvuuden ominaisuuksien lukumäärän sekä arvonnäilyksen epävarmuuden suhteen.



**Kuva 4.** Eri priorisointimenetelmäryhmien soveltuvuus ominaisuuksien lukumäärän ja arvonnäilyksen epävarmuuden suhteen (mukailtu lähteestä Tourwé et al. 2009)

Morales-Ramirez et al. (2017) mukaan viime aikoina on kehitelty monenlaisia automatisoituja päättelytekniikoita tukemaan monikriteeristä ominaisuuksien priorisointia sidosryhmäkokojen kasvaessa. Näiden automatisoitujen päättelytekniikoiden tarkoituksena on vähentää ihmisten tekemää manuaalista työtä sekä parantaa ominaisuuksien priorisointien tulosten laatua. Esimerkiksi hakupohjaisia optimointimenetelmiä (search-based optimization techniques) on kehitetty ratkaisemaan priorisoinnin monitavoitteellisia optimointiongelmia. Myös useita ohjelmistotyökaluja on kehitetty tukemaan ominaisuuksien priorisointia, perustuen usein pareittaisiin ominaisuuksien vertailuihin. (Morales-Ramirez et al. 2017)

Vaikka lähes kaikissa tähän kirjallisuuskatsaukseen luetuissa tutkimuksissa painotettiin ominaisuuksien valinnan tärkeyttä liiketoiminnan menestyksen kannalta, niin Komssi et al. (2015) tutkimuksen mukaan liiallinen ominaisuuksiin keskittyminen voi huonoimmassa tapauksessa jopa laskea asiakasarvoa. Tutkimuksen mukaan liiallinen ominaisuuksiin keskittyvä ohjelmistokehitys voi johtaa kolmeen sudenkuoppaan, joita tulisi välttää: 1) liian monen ominaisuuden lisääminen, 2) tiettyjen ominaisuuksien liiallinen parantaminen sekä 3) ominaisuuksien riisuttujen versioiden liian nopea julkaiseminen. Komssi et al. (2015) ehdottavat edellisten ongelmien välttämiseksi siirtymistä ominaisuuksien priorisoinnista ominaisuuksia korkeamman tason asiakastoimintojen priorisointiin.

### **5.3 Ominaisuuksien arvoperusteiset priorisointimenetelmät**

Ominaisuuksien ja vaatimusten priorisointimenetelmiä on kirjallisuudesta tunnistettavissa useita. Esimerkiksi Achimugu et al. (2014) tunnistivat systemaattisessa kirjallisuuskatsauksessaan 49 eri priorisointimenetelmää. Kirjallisuuskatsauksen mukaan viisi kirjallisuudessa eniten viitattua ominaisuuksien priorisointimenetelmää olivat Analytic hierarchy process (AHP), Quality functional deployment (QFD), Planning game, Binary search tree (BST) ja \$100 allocation -metodi. Kaikki edelliset priorisointimenetelmät luokituvat myös Kukreja et al. (2012) koostamaan 17 yleisimmin käytetyn priorisointimenetelmien listalle. Myös esimerkiksi Ramzan et al. (2009) mainitsevat merkittävimpinä priorisointitekniikoina kaikki edelliset QFD:a lukuun ottamatta. Achimugu et al. (2014) systemaattisen kirjallisuuskatsauksen mukaan parittaiseen vertailuun perustuva AHP on ylivoimaisesti yleisimmin kirjallisuudessa viitattu menetelmä, ja myös monet muut menetelmät hyödyntävät usein AHP:a pohjanaan.

Tässä työssä kuitenkin keskitytään priorisointimenetelmiin, joilla on selkeä sidos arvoperusteiseen priorisointiin, jossa ominaisuuksien arvo on päätösten keskiössä. Priorisointi-

menetelmien arvokeskeisyyttä on kuitenkin vaikea vahvistaa ja jokaisen priorisointimenetelmän taustalta on tunnistettavissa arvoon perustuvia komponentteja. Kukreja et al. (2012) tutkimus tukee väitettä priorisointimenetelmien arvoperusteisuuden tunnistamisen vaikeudesta. He luokittelevat esimerkiksi perinteiset ominaisuuksien priorisointiin käytetyt 1-10 pisteytys tai MoSCoW-metodit arvoneutraaleiksi menetelmiksi, sillä kyseisissä menetelmissä oletetaan usein virheellisesti, että sidosryhmät olisivat kykeneviä priorisoimaan ominaisuuksia niiden todellisen arvon perusteella (Kukreja et al. 2012; Kukreja et al. 2013).

Biffi et al. (2006, s. 250) tunnistavat arvoperusteisen ohjelmistotuotannon kirjassaan kirjallisuuden pohjalta neljä arvoon perustuvaa ominaisuuksien priorisointimenetelmää: ad hoc -suunnittelun, cost-value -priorisoinnin, Incremental Funding -menetelmän sekä EVOLVE\*-menetelmän. Tässä alaluvussa käsitellään kaikkia muita edellisiä menetelmiä, paitsi ad hoc -suunnittelua, koska siinä arvoa käsitellään vain epäsuorasti perustuen päätöksentekijän yleiseen ja intuitiiviseen arviointiin. Edellä mainitun kirjan julkaisun jälkeen on myös kehitetty uusia arvoperusteisia priorisointimenetelmiä. Yksi suosituimmista on Azar et al. (2007) esittelemä Value-Oriented Requirements Prioritization -menetelmä (VOP). VOP on valittu esiteltäväksi tässä alaluvussa menetelmän arvokeskeisyyden sekä suosion perusteella. Kyseinen menetelmä oli myös Achimugu et al. (2014) systemaattisen kirjallisuuskatsauksen kahdenkymmenen eniten viitatuimman menetelmän joukossa.

### **5.3.1 Cost-value**

Karlssonin ja Ryanin (1997) esittelemä cost-value -menetelmä on yksi yleisimmistä kirjallisuudessa mainituista ominaisuuksien priorisointimenetelmistä. Achimugu et al. (2014) systemaattisen kirjallisuuskatsauksen mukaan cost-value -menetelmä oli kuudenneksi yleisimmän kirjallisuudessa siteerattu priorisointimenetelmä. Kuten moni muukin ominaisuuksien priorisointimenetelmä, myös cost-value -menetelmä käyttää pohjanaan AHP:a, jonka avulla ominaisuuksia vertaillaan pareittain suhteellisen arvon ja kustannusten mukaisesti. Huomioitavaa menetelmässä on, että se käsittelee kustannuksia ja arvoa erikseen, eli kokonaisarvoon ei huomioida ominaisuuksien kehityskustannuksia. Kuten tämän työn ominaisuuksien arvotekijöitä käsittelevästä taulukosta 2 huomataan, jokainen taulukon empiirisistä tutkimuksista sisällytti myös kehityskustannukset arvon kokonaisvaltaiseen määritelmään.

Cost-value -menetelmä sijoittaa ominaisuudet kahden dimension perusteella: ominaisuuksien suhteellisen asiakasarvon sekä arvioitujen kehityskustannusten perusteella. Sekä suhteellisen arvon että kehityskustannusten määrittämisessä hyödynnetään

AHP:n parittaisen vertailun menetelmää. Kun edelliset suhteelliset arvot on laskettu, menetelmä mahdollistaa ominaisuuksien arvon visualisoinnin kehityskustannusten funktiona. Tuloksena saadun graafisen visualisoinnin perusteella projektipäälliköt voivat tarkastella arvoja ominaisuuksien valinnan ja priorisoinnin perustaksi. (Karlsson & Ryan 1997; Biffi et al. 2006, s. 250)

Cost-value -menetelmän suurimpina rajoitteina on sen kykenemättömyys skaalautua ominaisuuksien lukumäärän kasvaessa. Achimugu et al. (2014) mainitsevat, että menetelmä on aikaa vievä ja skaalautumaton. Achimugun ja Selamatin (2015) mukaan menetelmän haittapuolena on myös, että se on kyvytön päivittämään arvojärjestettyjen ominaisuuksien sijoituksia aina kun ominaisuuslistaan tulee muutoksia, eli ominaisuuksia lisätään tai poistetaan. Thakurtan (2013) mukaan cost-value -menetelmä ei myöskään kykene ottamaan huomioon ominaisuuksien keskinäisiä riippuvuuksia. Cost-value menetelmässä arvolla myös viitataan pääsääntöisesti asiakkaan havaitsemaan arvoon, eikä se eksplisiittisesti ota huomioon muita sidosryhmiä.

### 5.3.2 EVOLVE\*

Ruhe ja Ngo The (2004) esittelivät EVOLVE\*-menetelmän, jota pidetään yleisesti arvo-keskeisenä priorisointimenetelmänä. EVOLVE\* on jatkokehitetty versio EVOLVE-menetelmästä, jonka Greer ja Ruhe (2004) esittelivät menetelmäksi suoriutumaan julkaisusuunnittelun yleisistä ongelmista. EVOLVE-menetelmässä annetaan syötteenä joukko ominaisuuksia sekä niiden työmääräarviot ja ominaisuuksia edustavien sidosryhmien toimesta toteutetut prioriteettiluokitukset. Näiden syötteiden pohjalta menetelmä käyttää geneettisiä algoritmeja mahdollisten julkaisusuunnitelmien selvittämiseksi ennalta määritettyjen rajoitteiden rajoissa. Geneettisiä algoritmeja hyödynnetään useita kertoja iteratiivisesti painotetun arvon maksimoimiseksi ottaen huomioon kaikki eri sidosryhmät. (Greer & Ruhe 2004)

EVOLVE\* on hybridi menetelmä, jonka taustalla on usko siitä, että laskennallinen älykkyys ei voi korvata ihmistä päätöksentekijänä ja päinvastoin. Menetelmässä ”hybridiäly” selvittää laskennallisen älykkyyden ja päätöksentekijöiden välisiä synergioita. EVOLVE\* on suunniteltu iteratiiviseksi jatkuvasti suoritettavaksi menettelytavaksi, joka helpottaa ohjelmistojen julkaisusuunnitteluun liittyvien ongelmien selvittämistä. EVOLVE\* käyttää hyväksi sekä laskennallisen älykkyyden työkaluja eksplisiittisen ja tarkan datan käsittelyyn, että ihmismielen kykyä selviytyä paremmin hiljaisen tiedon ja sumean datan käsittelystä. (Biffi et al. 2006, s. 250 – 251) EVOLVE\*:*n* pääasiallisena tavoitteena on evolu-

tiivisesti arvioida todellisuutta ja vähentää mahdollisten ratkaisujen joukkoa mahdollisimman pieneksi, josta päätöksentekijä voi valita tilanteeseen sopivimman ratkaisun (Ruhe & Ngo The 2004).

EVOLVE-menetelmät mahdollistavat iteratiivisen luonteensa takia myös ominaisuusvaatimusten myöhäiset muutokset, sidosryhmien prioriteettijärjestysten muutokset sekä työmääräarvioiden muutokset (Thakurta 2013). Thakurtan (2013) mukaan EVOLVE-menetelmien heikkoutena on niiden laskennallinen monimutkaisuus sekä kustannustekijöiden huomioimattomuus. Biffi et al. (2006, s. 251) ovat myös kehittäneet menetelmästä modifioidun version, F-EVOLVE\*:n, joka ilmaisee arvon taloudellisin termein. F-EVOLVE\*-menetelmässä sidosryhmiä pyydetään äänestämisen sijasta antamaan taloudelliset arviot ominaisuuksille (Biffi et al. 2006, s. 251). F-EVOLVE\*:a ei kuitenkaan esiintynyt tämän systemaattisen kirjallisuuskatsauksen muissa hakutuloksissa.

### 5.3.3 Incremental Funding method (IFM)

Dennen ja Cleland-Huangin (2004) esittelemä incremental funding -metodi (IFM) on dataohjautuva ominaisuuksienhallinnan menetelmä, joka käsittelee arvoa taloudellisin termein. Kyseinen menetelmä tavoittelee ohjelmistoinvestointien nettonykyarvon maksimointia analysoimalla ja sekvensoimalla ominaisuuksien toimitukset tarkkaan. Tuottojen ja nettonykyarvon maksimoimiseksi ominaisuudet jaetaan tarkkaan sekvensoituihin asiakasarvoa luoviin 'paloihin' (chunks). (Biffi et al. 2006, s. 250 – 251) IFM:n motivaationa Dennen ja Cleland-Huangin (2004) mukaan on, että ohjelmistokehitysyrietykset voivat menestyä ainoastaan tunnistamalla ohjelmistokehityspäätösten vaikutukset projektien taloudellisiin tunnuslukuihin.

IFM jakaa järjestelmän toiminnallisiksi yksiköiksi, jota kutsutaan vähimmäismarkkinakelpoiseksi ominaisuuksiksi (minimum marketable features). Vähimmäismarkkinakelpoiset ominaisuudet ovat pieniä itsenäisiä ominaisuuksia, jotka luovat arvoa asiakkaalle ja jotka voidaan nopeasti toimittaa asiakkaalle. Vähimmäismarkkinakelpoisten ominaisuuksien tunnistamisen avulla menetelmällä tavoitellaan optimaalisen ominaisuuksien toimitusjärjestyksen tunnistamista, joka maksimoi projektin nettonykyarvon. (Denne & Cleland-Huang 2004)

Huomionarvoista on, että IFM:a ei ole listattu Achimugu et al. (2014) systemaattisessa kirjallisuuskatsauksessa, joka tunnisti 49 eri priorisointimenetelmää. IFM kuitenkin esiintyy alan kirjallisuudessa usein, ja Biffi et al. (2006) lisäksi myös esimerkiksi Kukreja et al.

(2012) ovat valinneet IFM:in yleisimpien arvoperusteisten priorisointimenetelmien joukkoon. IFM:n heikkoutena voidaan pitää sen taloudellista painopistettä, jolloin arvon muut keskeiset komponentit voivat jäädä projekteissa huomiotta.

### 5.3.4 Value-Oriented Requirements Prioritization (VOP)

Value-Oriented Requirements Prioritization (VOP) on kvantitatiivinen menetelmä, joka tarjoaa viitekehysten ominaisuuksiin liittyvien priorisointien ja päätöksenteon perustaksi. Se tarjoaa näkyvyyden kaikille sidosryhmille päätöksenteon aikana, minimoiden eri sidosryhmien väliset pitkät keskustelut ja väittelyt liittyen yksilöiden eroaviin näkemyksiin vaatimuksista. VOP korostaa organisaation tunnistamia ydinliiketoiminta-arvoja, joita vasten viitekehys arvioi yksittäisten ominaisuuksien arvoa. (Azar et al. 2007)

VOP:in soveltamisen ensimmäisessä vaiheessa yrityksen johtajat tunnistavat ydinliiketoiminta-arvot, joille he antavat painoarvot hyödyntämällä yksinkertaista järjestysasteikkoa arvojen tärkeyden perusteella. Viitekehys tukee myös riskien tunnistamista, jotka luokitellaan liiketoimintariskeihin ja teknisiin riskeihin. Näille kahdelle riskikategorialle annetaan myös riskitoleranssin mukaan painoarvot, jonka jälkeen riskiä mitataan negatiivisella asteikolla. Edellisten tietojen perusteella VOP:issa rakennetaan priorisointimatriisi. Matriisissa sidosryhmät antavat jokaiselle ominaisuudelle numeroarvot asteikolla 1-10 ydinliiketoiminta-arvojen ja riskien suhteen. Tämän jälkeen kertomalla annetut arvot liiketoiminta-arvon tai riskikategorian painoarvolla jokaiselle ominaisuudelle voidaan laskea kokonaispisteet. (Azar et al. 2007) Taulukossa 4 esitetään esimerkkimatriisi, jossa kahdelle eri ominaisuudelle on annettu arvot ja laskettu kokonaispisteet.

**Taulukko 4.** VOP -priorisointimenetelmän esimerkkimatriisi (mukailtu lähteestä Azar et al. 2007)

Ominaisuus	Ydinliiketoiminta-arvot				Riskit		Pisteet
	Myynti	Markkinointi	Kilpailukyky	Strategia	Tekninen	Liiketoiminta	
	7	6	8	10	-8	-5	
O1	5	4	10	9	8	5	140
O2	7	8	4	5	2	7	128

Edellisessä taulukon 4 esimerkissä ominaisuus O1 saa enemmän kokonaispisteitä, joten sillä on ominaisuutta O2 korkeampi prioriteetti. Ominaisuus O1 on täten loogisempi valinta sisällytettäväksi esimerkiksi julkaisusuunnittelun tapauksessa seuraavaan julkaisuun.

Azar et al. (2007) mukaan VOP:in arvo perustuu sen ymmärrettävyyteen ja selkeyteen. Heidän mukaansa sidosryhmät ymmärtävät helposti VOP:in perustoiminnot ja prosessi ohjaa keskustelua yksittäisten tuotevaatimusten sijasta strategiseen liiketoimintakeskusteluun. Azar et al. (2007) tutkimuksen case-yritys myös raportoi menetelmän vähentäneen sisäistä väittelyä yksittäisistä tuotevaatimuksista ja lisäävän sidosryhmien ymmärrystä strategisista liiketoiminta-arvoista. Thakurta (2013) kuitenkin huomauttaa, että viitekehys jättää laskentamenettelyssään kokonaan huomioimatta ominaisuuksien väliset riippuvuudet. Duan et al. (2009) myös argumentoivat, että VOP soveltuisi vain pieniin ja keskisuuriin projekteihin eikä tarjoaisi soveltuvaa ratkaisua suurten projektien hallitsemiseen. Tourwé et al. (2009) huomauttavat, että kyseisenlainen pisteytysjärjestelmä on myös hyvin aikaa vievä ominaisuuslistan kasvaessa suureksi. VOP kehitettiin alun perin pienen ohjelmistokehitysyrityksen tarpeisiin (Azar et al. 2007).



## 6. PÄÄTELMÄT

Tässä systemaattisessa kirjallisuuskatsauksessa tarkasteltiin tieteellisten tutkimusten avulla ohjelmistojen ominaisuuksien arvotekijöitä, arvonmäärittystä sekä ominaisuuksien arvon hyödyntämistä päätöksenteossa. Ominaisuuksien arvonmäärittelyllä ja arvopeusteisella päätöksenteolla tunnistettiin olevan merkittäviä positiivisia vaikutuksia ohjelmistoprojektien menestyksen kannalta. Kirjallisuudesta kuitenkin tunnistettiin, että ohjelmistoyritykset hyödynsivät arvoa usein vain epäsuorasti tai keskittyivät vain tiettyihin arvon osa-alueisiin arvon kokonaisvaltaisemman hyödyntämisen sijasta. Esimerkiksi tutkimuksissa otettiin usein huomioon vain asiakasarvo, jättäen muut sidosryhmät ja arvon ulottuvuudet tarkastelun ulkopuolelle.

Kuten arvo yleisestikin, myös ohjelmisto-ominaisuuksien arvo on moniulotteinen käsite sisältäen useita eri arvotekijöitä. Ohjelmistojen ominaisuuksien arvon koostumisesta löytyi tieteellisestä kirjallisuudesta suhteellisen vähän tietoa. Yleisimmiksi empiirisissä tutkimuksissa mainituiksi ominaisuuksien arvotekijöiksi työssä tunnistettiin loppukäyttäjätyytyväisyys, tuotteen kilpailukyky, markkinoillesaantiaika sekä kehityksen kokonais kustannukset. Yleisiä tutkimuksissa tunnistettuja ominaisuuksien arvotekijöitä olivat myös asiakastyytyväisyys, sijoitetun pääoman tuotto sekä kehitysvalmiudet. Arvotekijöistä vähäisimmälle huomiolle tutkimuksissa jäivät yrityksen strategia -dimension alle kuuluvat arvotekijät, kuten ominaisuuksien vaikutus tuotteen strategiaan, yrityksen portfolioon tai yrityksen brändiin. Työssä tunnistettiin kuitenkin arvotekijöiden vahva konteksti- ja yritys-sidonnaisuus, minkä vuoksi arvotekijöiden painoarvot ovat aina kullekin tilanteelle yksilöllisiä. Tämän vuoksi yleistä jokaiselle ohjelmistoyritykselle tai -projektille sopivaa listaa tärkeimmistä ominaisuuksien arvotekijöistä ei ole mahdollista määrittää.

Ohjelmistojen ominaisuuksien arvonmäärittelystä löytyi hyvin vähän tieteellistä kirjallisuutta. Tämä selittyy osin sillä, että ominaisuuksien arvonmäärittystä ei kirjallisuudessa nähty erillisenä toimintona, vaan arvonmäärittely oli usein kiinteä osa arvoperusteisia ominaisuuksien priorisointimenetelmiä. Arvonmäärittelyn tunnistettiin olevan hyvin haastavaa useiden ulkoisten syiden takia, kuten ympäristöön liittyvän epävarmuuden, sidosryhmien eroavien näkemysten sekä arvon moniulotteisen luonteen takia. Yleisimmin ohjelmistoyrityksissä ominaisuuksien arvoa arvioidaan joko pisteytysmenetelmillä tai rahamääräisillä menetelmillä (Park & Park 2004; Khurum et al. 2013). Pisteytysmenetelmissä voidaan käyttää pohjana ominaisuuksien arvotekijöitä, joille avainsidosryhmät antavat tilanteeseen sopivaa skaalaa käyttäen pistearvot niiden tärkeyden perusteella kunkin arvioitavan ominaisuuden kohdalla. Pisteytysmenetelmien heikkoutena on, että ne eivät

kykene määrittämään ominaisuuksien absoluuttista arvoa, vaan ilmaisevat arvon suhteellisena arvona (Khurum et al. 2013). Rahamääräisissä arvonmäärittämissä, kuten nettonykyarvon laskemisessa, ominaisuuksien arvoa arvioidaan ominaisuuksien taloudellisen arvon perusteella. Tutkimusten mukaan kuitenkin monien arvon komponenttien kuten aineettomien strategisten hyötyjen arvioiminen rahamääräisenä on hyvin haastavaa, minkä vuoksi rahamääräisiä menetelmiä hyödynnetään käytännön ohjelmistoprojekteissa varsin vähän (Park & Park 2004; Auer et al. 2011).

Suurin osa ominaisuuksien arvonmäärittämisessä käytettävissä menetelmistä pohjautuvat hyvin vahvasti sidosryhmien subjektiiviseen arvon arviointiin (Khurum et al. 2013; Rodriguez et al. 2018). Subjektiivinen arvonmäärittäminen olettaa usein virheellisesti, että sidosryhmät ovat kykeneviä arvioimaan ominaisuuksia niiden todellisen arvon perusteella (Kukreja 2013). Viime aikoina alan tutkimuksessa onkin tunnistettu tarvetta objektiivisemmille arvonmäärittämis menetelmille. Ominaisuuksien käyttöasteeseen perustuva arvonmäärittäminen on menetelmä, joka minimoi subjektiivisen arvioinnin määrän. Ominaisuuksien käyttöasteeseen perustuvan arvonmäärittäksen heikkoutena kuitenkin on, että päätöksenteossa tarvittavaa tietoa ominaisuuksien arvosta saadaan usein liian myöhäisessä vaiheessa vasta ominaisuuksien toteutuksen jälkeen.

Varsinkin nykypäivänä yleisen inkrementaalisen ohjelmistokehityksen mallissa tehdään ohjelmisto-ominaisuuksien toteutukseen liittyviä päätöksiä jatkuvasti. Inkrementaalissa ohjelmistokehityksessä ominaisuuksien toteutukset voidaan priorisoida siten, että suurinta arvoa tuottavat ominaisuudet voidaan toimittaa järjestelmään ensimmäisinä. Päätöksentekoa ominaisuuksien valitsemisesta ja priorisoinnista seuraaviin julkaisuihin kutsutaan julkaisusuunnitteluksi. (Greer & Ruhe 2004; Mc Elroy & Ruhe 2010) Ominaisuuksien priorisoinnin tunnistettiin olevan haastava toiminto jopa yksinkertaisessa kehitysympäristössä, jossa ominaisuuksien ja arvonmäärittäksen epävarmuuden määrä ovat vähäisiä. Tutkimusten mukaan hyvin harvat nykyisistä ominaisuuksien priorisointimenetelmistä soveltuivat tilanteeseen, jossa sekä ominaisuuksien lukumäärä että arvonmäärittäksen epävarmuus kasvoivat korkeiksi. Kirjallisuuden mukaan keskeinen menestystekijä ominaisuuksien toteutukseen liittyvän päätöksenteon kannalta löytyy arvoperusteisista menetelmistä, jotka ottavat arvon ja arvonmäärittäykseen liittyvän epävarmuuden kokonaisvaltaisesti huomioon (esim. Boehm 2003b; Biffi et al. 2006).

Ominaisuuksien valikointi- ja priorisointimenetelmiä on kirjallisuudesta tunnistettavissa useita. Ominaisuuksien priorisointimenetelmät pohjautuvat usein ainakin jossain määrin ominaisuuksien arvoon, mutta arvo käsitetään eri menetelmissä eri tavalla. Esimerkiksi joissain priorisointimenetelmissä ominaisuuden kiireellisyyttä tai toteutuksen kustannuk-

sia käsitellään arvosta erillisenä tekijänä, kun taas toisissa menetelmissä ne luetaan mukaan kokonaisarvoon. Arvoperusteisiksi priorisointimenetelmiksi, joissa kirjallisuuden mukaan ominaisuuksien arvo on keskiössä, työssä tunnistettiin cost-value -menetelmä, EVOLVE\*-menetelmä, incremental funding method (IFM) sekä value-oriented requirements prioritization -menetelmä (VOP). Myös eri priorisointimenetelmien soveltuvuuden tunnistettiin olevan kontekstiriippuvaista ja vaihtelevan esimerkiksi ohjelmistoprojektin koon ja ominaisuuksien lukumäärän mukaan.

Käytännön ohjelmistoprojektien ominaisuuksien arvotekijöihin keskittyviä empiirisiä tutkimuksia löytyi kirjallisuudesta huonosti, ja niitä ehdotetaankin tässä työssä yhdeksi jatkotutkimuskohteeksi. Myös ohjelmisto-ominaisuuksien arvonmääritykseen keskittyvää tieteellistä kirjallisuutta löytyi hyvin vähän, mikä selittyy osittain sillä, että arvonmääritys on usein kiinteä osa ominaisuuksien priorisointimenetelmiä. Yleisemmältä ohjelmistotuotannon tasolta arvonmäärityksestä taas löytyi enemmän kirjallisuutta, mutta huomionarvoista on, että usein kyseinen kirjallisuus keskittyi käsittelemään pääasiassa arvonmäärityksen haasteita. Monissa arvonmääritystä käsittelevissä tutkimuksissa mainitaan, että päätöksenteon kannalta olisi paras, jos arvo voitaisiin määrittää rahamääräisenä. Rahamääräistä arvonmääritystä harjoitetaan kuitenkin käytännön ohjelmistotuotannossa kirjallisuuden mukaan hyvin vähän, sillä useita arvotekijöitä on hyvin vaikea arvioida taloudellisin suurein. Kirjallisuudessa ei kuitenkaan ollut juurikaan tunnistettavissa tutkimuksia erilaisten mahdollisten taloudellisten arvonmääritysmenetelmien kehityksestä, joka ottaisi kyseiset haasteet huomioon.

Sekä ominaisuuksien arvonmäärityksen että priorisointimenetelmien mainittiin yleisesti olevan konteksti- ja yritysriippuvaisia. Hyvin harvassa tutkimuksessa kuitenkin keskityttiin tarkemmin selvittämään, minkälaisiin tilanteisiin mitkäkin menetelmät soveltuvat parhaiten. Tässä työssä ehdotetaan tulevaisuudessa tutkittavaksi ominaisuuksien priorisointimenetelmien soveltamista erikokoisissa yrityksissä, jaoteltuina esimerkiksi pk- ja suuryrityksiin. Arvonmääritys- ja priorisointimenetelmien soveltuvuuksien eroja ei myöskään kirjallisuudessa eroteltu erityyppisiä tuotteita tai palveluja tarjoavien yritysten välillä. Kuten myös Tourwé et al. (2009) tutkimuksensa pohjalta toteavat, käytännön ohjelmistoliiketoiminnassa arvo voi koostua eri tekijöistä kaupallisia ohjelmistotuotteita tarjoavan yrityksen ja kustomoituja ohjelmistoja tarjoavan yrityksen välillä. Tulevissa jatkotutkimuksissa ehdotetaankin tutkittavaksi, miten nämä arvotekijät käytännössä eroavat eri liiketoimintamalleja hyödyntävien yritysten välillä. Mielenkiintoista olisi myös tietää, vaikuttavatko kyseiset arvotekijöiden erot arvonmäärityksen ja ominaisuuksien priorisointien prosesseihin.

## LÄHTEET

- Abad, Z. S. H., & Ruhe, G. (2015, August). Using real options to manage technical debt in requirements engineering. In 2015 IEEE 23rd International Requirements Engineering Conference (RE) (pp. 230-235). IEEE.
- Achimugu, P., Selamat, A., Ibrahim, R., & Mahrin, M. N. R. (2014). A systematic literature review of software requirements prioritization research. *Information and software technology*, 56(6), 568-585.
- Achimugu, P., & Selamat, A. (2015). A hybridized approach for prioritizing software requirements based on K-means and evolutionary algorithms. In *Computational Intelligence Applications in Modeling and Control* (pp. 73-93). Springer, Cham.
- Alahyari, H., Svensson, R. B., & Gorschek, T. (2017). A study of value in agile software development organizations. *Journal of Systems and Software*, 125, 271-288.
- Alwis, D., Hlupic, V., & Fitzgerald, G. (2003, June). Intellectual capital factors that impact of value creation. In *Proceedings of the 25th International Conference on Information Technology Interfaces, 2003. ITI 2003.* (pp. 411-416). IEEE.
- Apel, S., Batory, D., Kästner, C., & Saake, G. (2016). *Feature-oriented software product lines*. Springer-Verlag Berlin An.
- Auer, L., Kryvinska, N., Strauss, C., & Belov, E. (2011, June). Software-based business applications/tools to assess complex SOA investments-The cross-vendor comparative analysis. In *2011 International Conference on Complex, Intelligent, and Software Intensive Systems* (pp. 397-401). IEEE.
- Azar, J., Smith, R. K., & Cordes, D. (2007). Value-oriented requirements prioritization in a small development organization. *IEEE software*, 24(1), 32-37.
- Bakalova, Z., Daneva, M., Hermann, A., & Wieringa, R. (2011, March). Agile requirements prioritization: What happens in practice and what is described in literature. In *International Working Conference on Requirements Engineering: Foundation for Software Quality* (pp. 181-195). Springer, Berlin, Heidelberg.
- Barney, S., Aurum, A., & Wohlin, C. (2008). A product management challenge: Creating software product value through requirements selection. *Journal of Systems Architecture*, 54(6), 576-593.

- Barney, S., Hu, G., Aurum, A., & Wohlin, C. (2009). Creating software product value in China. *IEEE software*, 26(4), 84-90.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). Manifesto for agile software development.
- Biffi, S., Aurum, A., Boehm, B., Erdogmus, H., & Grünbacher, P. (Eds.). (2006). Value-based software engineering. Springer Science & Business Media.
- Boehm, B. W. (1984). Software engineering economics. *IEEE transactions on Software Engineering*, (1), 4-21.
- Boehm, B. (2003a). Value-Based Software Engineering: Reinventing "Earned Value" Monitoring and Control.
- Boehm, B. (2003b). Value-based software engineering. *ACM SIGSOFT Software Engineering Notes*, 28(2), 4.
- Browning, T. R. (2003). On customer value and improvement in product development processes. *Systems engineering*, 6(1), 49-61.
- Budgen, D., & Brereton, P. (2006, May). Performing systematic literature reviews in software engineering. In *Proceedings of the 28th international conference on Software engineering* (pp. 1051-1052).
- Bullock, J. (2000). Calculating the Value of Testing From an executive's perspective, software testing is not a capital investment in the physical plant, an acquisition, or another readily accepted business expense. A Quality Assurance Manager describes how to present testing as a business-process investment. *Software Testing and Quality Engineering*, 2, 56-63.
- Carlshamre, P. (2002). Release planning in market-driven software product development: Provoking an understanding. *Requirements engineering*, 7(3), 139-151.
- Cleland-Huang, J. (2015). Injecting value-thinking into prioritization decisions. *IEEE Software*, 32(2), 14-18.
- Denne, M., & Cleland-Huang, J. (2004). The incremental funding method: Data-driven software development. *IEEE software*, 21(3), 39-47.
- Denyer, D., & Tranfield, D. (2009). Producing a systematic review.
- Duan, C., Laurent, P., Cleland-Huang, J., & Kwiatkowski, C. (2009). Towards automated requirements prioritization and triage. *Requirements engineering*, 14(2), 73-89.

- Elliott, B. (2007, July). Anything is possible: Managing feature creep in an innovation rich environment. In 2007 IEEE International Engineering Management Conference (pp. 304-307). IEEE.
- Erdogmus, H., Favaro, J., & Halling, M. (2006). Valuation of software initiatives under uncertainty: concepts, issues, and techniques. In *Value-based software engineering* (pp. 39-66). Springer, Berlin, Heidelberg.
- Farbey, B., Land, F. F., & Targett, D. (1995). A taxonomy of information systems applications: the benefits' evaluation ladder. *European journal of information systems*, 4(1), 41-50.
- Farbey, B., & Finkelstein, A. (1999). Exploiting software supply chain business architecture: a research agenda.
- Favare, J. (2002). Managing requirements for business value. *IEEE software*, 19(2), 15-17.
- Freitas, V., Perkusich, M., Mendes, E., Rodríguez, P., & Oivo, M. (2017, December). Value-Based decision-making using a web-based tool: A multiple case study. In 2017 24th Asia-Pacific Software Engineering Conference (APSEC) (pp. 279-288). IEEE.
- Gorschek, T., Fricker, S., Palm, K., & Kunsman, S. (2010). A lightweight innovation process for software-intensive product development. *IEEE software*, 27(1), 37-45.
- Greer, D., & Ruhe, G. (2004). Software release planning: an evolutionary and iterative approach. *Information and software technology*, 46(4), 243-253.
- IEEE. (2008). *IEEE Standard for Software and System Test Documentation (829-2008)*, IEEE, USA, 1-150 p.
- Imtiaz, A., Bürger, T., Popov, I. O., & Simperl, E. (2009, April). Framework for value prediction of knowledge-based applications. In *International Conference on Business Information Systems* (pp. 153-158). Springer, Berlin, Heidelberg.
- Jan, N., & Ibrar, M. (2010). Systematic mapping of value-based software engineering: A systematic review of value-based requirements engineering.
- Johansson, E., Bergdahl, D., Bosch, J., & Olsson, H. H. (2015, June). Quantitative requirements prioritization from a pre-development perspective. In *International Conference on Software Process Improvement and Capability Determination* (pp. 58-71). Springer, Cham.
- Jorgensen, M., & Shepperd, M. (2006). A systematic review of software development cost estimation studies. *IEEE Transactions on software engineering*, 33(1), 33-53.

- Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., & Peterson, A. S. (1990). Feature-oriented domain analysis (FODA) feasibility study (No. CMU/SEI-90-TR-21). Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst.
- Karlsson, J., & Ryan, K. (1997). A cost-value approach for prioritizing requirements. *IEEE software*, 14(5), 67-74.
- Khurum, M., & Barney, S. (2009, September). Innovative features selection using real options theory. In 2009 Third International Workshop on Software Product Management (pp. 11-14). IEEE.
- Khurum, M., Gorschek, T., & Wilson, M. (2013). The software value map—an exhaustive collection of value aspects for the development of software intensive products. *Journal of software: Evolution and Process*, 25(7), 711-741.
- Kitchenham, B., Charters, S. (2007). Guidelines for Performing Systematic Literature Reviews in Software Engineering, Tech. Rep. EBSE-2007-01, Keele University and University of Durham.
- Komssi, M., Kauppinen, M., Töhönen, H., Lehtola, L., & Davis, A. M. (2015). Roadmapping problems in practice: value creation from the perspective of the customers. *Requirements Engineering*, 20(1), 45-69.
- Kukreja, N., Boehm, B., Payyavula, S. S., & Padmanabhuni, S. (2012, September). Selecting an appropriate framework for value-based requirements prioritization. In 2012 20th IEEE International Requirements Engineering Conference (RE) (pp. 303-308). IEEE.
- Kukreja, N. (2013, May). Decision theoretic requirements prioritization A two-step approach for sliding towards value realization. In 2013 35th International Conference on Software Engineering (ICSE) (pp. 1465-1467). IEEE.
- Kukreja, N., Payyavula, S. S., Boehm, B., & Padmanabhuni, S. (2013). Value-based requirements prioritization: usage experiences. *Procedia Computer Science*, 16, 806-813.
- Lehtola, L., & Kauppinen, M. (2006). Suitability of requirements prioritization methods for market-driven software product development. *Software Process: Improvement and Practice*, 11(1), 7-19.
- Lehtola, L., Kauppinen, M., Vähäniitty, J., & Komssi, M. (2009). Linking business and requirements engineering: is solution planning a missing activity in software product companies?. *Requirements engineering*, 14(2), 113-128.

- Lindgren, M., Wall, A., Land, R., & Norström, C. (2008, September). A method for balancing short-and long-term investments: Quality vs. Features. In 2008 34th Euromicro Conference Software Engineering and Advanced Applications (pp. 175-182). IEEE.
- Marcuska, S., Gencel, C., & Abrahamsson, P. (2014, April). Feature usage as a value indicator for decision making. In 2014 23rd Australian Software Engineering Conference (pp. 124-131). IEEE.
- Marcuska, S., Gencel, C., Wang, X., & Abrahamsson, P. (2013a, June). Feature usage diagram for feature reduction. In International Conference on Agile Software Development (pp. 223-237). Springer, Berlin, Heidelberg.
- Marcuska, S., Gencel, C., & Abrahamsson, P. (2013b, June). Exploring how feature usage relates to customer perceived value: A case study in a startup company. In International Conference of Software Business (pp. 166-177). Springer, Berlin, Heidelberg.
- Mc Elroy, J., & Ruhe, G. (2010). When-to-release decisions for features with time-dependent value functions. *Requirements engineering*, 15(3), 337-358.
- Mendes, F. F. (2018, May). The relationship between personality and value-based decision-making. In Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings (pp. 460-461).
- Mendes, E., Rodriguez, P., Freitas, V., Baker, S., & Atoui, M. A. (2017). Towards improving decision making and estimating the value of decisions in value-based software engineering: the VALUE framework. *Software Quality Journal*, 26(2), 607-656.
- Morales-Ramirez, I., Munante, D., Kifetew, F., Perini, A., Susi, A., & Siena, A. (2017, September). Exploiting user feedback in tool-supported multi-criteria requirements prioritization. In 2017 IEEE 25th International Requirements Engineering Conference (RE) (pp. 424-429). IEEE.
- Mohamed, S. I., & Wahba, A. M. (2008, December). Value estimation for software product management. In 2008 IEEE International Conference on Industrial Engineering and Engineering Management (pp. 2196-2200). IEEE.
- Nunamaker Jr, J. F., Briggs, R. O., De Vreede, G. J., & Sprague Jr, R. H. (2000). Enhancing organizations' intellectual bandwidth: The quest for fast and effective value creation. *Journal of management information systems*, 17(3), 3-8.
- Park, Y., & Park, G. (2004). A new method for technology valuation in monetary value: procedure and application. *Technovation*, 24(5), 387-394.



- Penny, D. A. (2002, October). An estimation-based management framework for enhance maintenance in commercial software products. In *International Conference on Software Maintenance*, 2002. Proceedings. (pp. 122-130). IEEE.
- Ramzan, M., Jaffar, M. A., Iqbal, M. A., Anwar, S., & Shahid, A. A. (2009, December). Value based fuzzy requirement prioritization and its evaluation framework. In *2009 Fourth International Conference on Innovative Computing, Information and Control (ICI-CIC)* (pp. 1464-1468). IEEE.
- Rein, A. D., & Münch, J. (2013, December). Feature prioritization based on mock-purchase: A mobile case study. In *International Conference on Lean Enterprise Software and Systems* (pp. 165-179). Springer, Berlin, Heidelberg.
- Renkema, T. J., & Berghout, E. W. (1997). Methodologies for information systems investment evaluation at the proposal stage: a comparative review. *Information and Software Technology*, 39(1), 1-13.
- Rodríguez, P., Markkula, J., Oivo, M., & Turula, K. (2012, September). Survey on agile and lean usage in finnish software industry. In *Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement* (pp. 139-148). IEEE.
- Rodríguez, P., Mendes, E., & Turhan, B. (2018). Key Stakeholders' Value Propositions for Feature Selection in Software-intensive Products: An Industrial Case Study. *IEEE Transactions on Software Engineering*.
- Rodríguez, P., Mikkonen, K., Kuvaja, P., Oivo, M., & Garbajosa, J. (2013, May). Building lean thinking in a telecom software development organization: strengths and challenges. In *Proceedings of the 2013 international conference on software and system process* (pp. 98-107).
- Rodríguez, P., Urquhart, C., & Mendes, E. (2020). A Theory of Value for Value-based Feature Selection in Software Engineering. *IEEE Transactions on Software Engineering*.
- Ruhe, G. (2003). Software engineering decision support: methodology and applications. *Innovations in decision support systems*, 3, 143-174.
- Ruhe, G. & Ngo The, A. (2004). Hybrid Intelligence in Software Release Planning, *International journal of hybrid intelligent systems*, Vol. 1(1-2), pp. 99-110. Available (accessed ID: cdi\_crossref\_primary\_10\_3233\_HIS\_2004\_11\_212)
- Ruhe, G., & Saliu, M. O. (2005). The art and science of software release planning. *IEEE software*, 22(6), 47-53.

- Saliu, O., & Ruhe, G. (2005). Software release planning for evolving systems. *Innovations in Systems and Software Engineering*, 1(2), 189-204
- Saunders, M.N.K., Lewis, P. & Thornhill, A. (2019). *Research methods for business students*, Pearson Education, Harlow, England
- Staff, M. W. (2004). *Merriam-Webster's collegiate dictionary*. Merriam-Webster.
- Stapić, Z., López, E. G., Cabot, A. G., de Marcos Ortega, L., & Strahonja, V. (2012, January). Performing systematic literature review in software engineering. In *CECIIS 2012-23rd International Conference*.
- Thakurta, R. (2013). A framework for prioritization of quality requirements for inclusion in a software project. *Software Quality Journal*, 21(4), 573-597.
- Tourwé, T., Codenie, W., Boucart, N., & Blagojević, V. (2009, June). Demystifying release definition: from requirements prioritization to collaborative value quantification. In *International Working Conference on Requirements Engineering: Foundation for Software Quality* (pp. 37-44). Springer, Berlin, Heidelberg.
- Uлага, W., & Chacour, S. (2001). Measuring customer-perceived value in business markets: a prerequisite for marketing strategy development and implementation. *Industrial marketing management*, 30(6), 525-540.
- Van Gorp, J., Bosch, J., & Svahnberg, M. (2000). Managing variability in software product lines. In *Proceedings of IEEE/IFIP Conference on Software Architecture* (pp. 45-54).
- Wieggers, K. (1999). First things first: prioritizing requirements. *Software Development*, 7(9), 48-53.
- Wittern, E., & Zirpins, C. (2011, March). Validating service value propositions regarding stakeholder preferences. In *2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops* (pp. 294-297). IEEE.
- Wohlin, C., & Aurum, A. (2005, November). What is important when deciding to include a software requirement in a project or release?. In *2005 International Symposium on Empirical Software Engineering*, 2005. (pp. 10-pp). IEEE.
- Woodruff, R. B. (1997). Customer value: the next source for competitive advantage. *Journal of the academy of marketing science*, 25(2), 139.
- Yanzer Cabral, A. R., Ribeiro, M. B., & Noll, R. P. (2014). Knowledge management in agile software projects: A systematic review. *Journal of Information & Knowledge Management*, 13(01), 1450010.

# LIITE A: TAULUKON 2 ARVOTEKIJÖIDEN KUVAUKSET

Rodriguez et al. (2020) mukaiset 24 arvon perustekijää	Kuvaus (Rodriquez et al. 2020)
<b>Asiakasarvo</b>	
T1. Asiakastyytyväisyys	Ominaisuuden toteutuksen vaikutus yrityksen asiakkaiden tyytyväisyyteen.
T2. Loppukäyttäjätyytyväisyys	Ominaisuuden toteutuksen vaikutus tuotteen loppukäyttäjien tyytyväisyyteen.
T3. Markkinatyytyväisyys	Ominaisuuden toteutuksen vaikutus markkinoihin, joilla yritys toimii.
T4. Asiakkaiden lukumäärä	Ominaisuuden toteutuksen vaikutus yrityksen tyytyväisten asiakkaiden lukumäärään (esim. niiden asiakkaiden lukumäärä, jotka ovat pyytäneet ominaisuutta).
T5. Johtavien asiakkaan tyytyväisyys	Ominaisuuden toteutuksen vaikutus yrityksen tärkeiden asiakkaiden tyytyväisyyteen.
<b>Kilpailukyky markkinoilla</b>	
T6. Tuotteen kilpailukyky	Ominaisuuden toteutuksen vaikutus tuotteen tilanteeseen kilpailijoihin nähden.
T7. Markkinoillesaantiaika	Ominaisuuden toteutuksen vaikutus julkaisun markkinoillesaantiaikaan (ts. aikaan, jolloin ominaisuus tulee olemaan saatavilla).
T8. Tuotteen laatu	Ominaisuuden toteutuksen vaikutus tuotteen laatuun yleisesti.
T9. Tuotteen suorituskyky	Ominaisuuden toteutuksen vaikutus tuotteen suorituskykyyn teknisestä näkökulmasta (esim. vikaantumisaste tai nopeus).
T10. Käyttäjäkokeuksellinen yksinkertaisuus	Ominaisuuden toteutuksen vaikutus tuotteen käytön yksinkertaisuuteen loppukäyttäjän näkökulmasta
<b>Taloudellinen arvo / kannattavuus</b>	
T11. Sijoitetun pääoman tuotto	Ominaisuuden sijoitetut pääoman tuotto rahamääräisenä ilmaistuna.
T12. Myyntimäärä	Ominaisuuden toteutuksen vaikutus tuoteyksiköiden myyntimäärään.
T13. Asiakaskunta/Markkinaosuus	Ominaisuuden toteutuksen vaikutus markkinaosuuteen (esim. auttaako ominaisuuden toteutus sitouttamaan uusia asiakkaita?)
<b>Kustannustehokkuus</b>	
T14. Kehityksen kokonaiskustannukset	Ominaisuuden toteuttamisen kokonaiskustannukset (suunnittelu, toteutus, testaus, varmennus ja integrointi jne.)
T15. Uudelleenkäytettävyys	Ominaisuuden toteutuksen vaikutus ohjelmistosisällön uudelleenkäytettävyteen (esim. missä määrin ominaisuutta voidaan käyttää uudelleen?).
T16. Kolmansien osapuolien kustannukset	Ominaisuuden toteutuksen vaikutus kolmansien osapuolien tuotteiden tai palvelujen hankintaan.
<b>Teknologia ja arkkitehtuuri</b>	
T17. Toteutuksen yksinkertaisuus	Ominaisuuden toteutuksen vaikutus tuotteen toteuttamisen yksinkertaisuuteen.
T18. Tuotearkkitehtuuri	Ominaisuuden toteutuksen vaikutus tuotteen arkkitehtuuriin (esim. missä määrin ominaisuuden toteutus parantaa sisäistä laatua?)
T19. Kehitysvalmiudet	Ominaisuuden toteutuksen vaikutus kykyyn kehittää kyseinen tai jokin muu ominaisuus ottaen huomioon sekä tietotaito että resurssit.
<b>Yrityksen strategia</b>	
T20. Yrityksen strategia	Ominaisuuden toteutuksen vaikutus yrityksen strategian tukemiseen (ts. missä määrin ominaisuus noudattaa yrityksen markkinastrategiaa?).
T21. Tuotteen strategia	Ominaisuuden toteutuksen vaikutus yrityksen tuotekonseptin rakentamiseen.

T22. Yrityksen portfolio	Ominaisuuden toteutuksen vaikutus yrityksen tarjontaportfolioon.
T23. Liiketoiminnallinen edesauttaja	Ominaisuuden toteutuksen vaikutus muun liiketoiminnan tai muiden tuotteiden vauhdittamiseen ja uusien arvovirtojen (value streams) mahdollistamiseen (esim. ristiinmyynnit yrityksen muiden tuotteiden kanssa).
T24. Yrityksen brändi	Ominaisuuden toteutuksen vaikutus yrityksen brändiin (esim. yrityksen maineeseen).