

Henri Kenias

**TEKNISEN VELAN VIESTINNÄN MENETELMÄT
ORGANISAATIOSSA**



JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2022

TIIVISTELMÄ

Kenias, Henri

Teknisen velan viestinnän menetelmät organisaatiossa

Jyväskylä: Jyväskylän yliopisto, 2022, 52 s.

Tietojärjestelmätiede, kandidaatin tutkielma

Ohjaaja: Pekkala, Kaisa

Hyvien ohjelmistokehityksen käytänteiden yhtenä tarkoituksena voidaan pitää pyrkimystä minimoida ohjelmiston jatkotyöstämiseen käytetty vaiva. Vaikka hyvien käytänteiden noudattaminen tuottaakin monia hyötyjä ohjelmiston elinkaarren edetessä, eivät nämä pitkäaikaiset seuraukset aina ole ilmeisiä. Monet ohjelmistokehitystyön ominaispiirteet, kuten kiireiset aikataulut ja kehitettävän ohjelmiston jatkuvasti muuttuvat vaatimukset, vaikeuttavatkin hyvien ohjelmistokehityksen käytänteiden noudattamista. Kun käytänteitä ei pystytä noudattamaan, tuotetaan ohjelmistokehitysprosessissa ratkaisuja, jotka voivat lyhyellä aikavälillä tuottaa hyötyä esimerkiksi ohjelmistokehitystyön nopeutumisenä. Nämä hyödyt saavutetaan kuitenkin jättämällä jokin hyvien käytänteiden mukainen toimenpide toteuttamatta, jonka korvaava puutteellinen toteutus voi aiheuttaa pitkällä aikavälillä erilaisia ongelmia. Ennen kuin toteuttamatta jäänyt hyvien käytänteiden mukainen toimenpide on toteutettu, aiheuttaa puutteellinen ratkaisu ohjelmiston ikääntyessä jatkuvasti enemmän ongelmia. Lopputuloksena voivat ohjelmiston ylläpito ja jatkokehitys muuttua ongelmien takia mahdottomaksi. Edellä kuvattua ilmiötä, jossa puutteellisilla kehitysratkaisuilla voidaan tuottaa lyhyen aikavälin hyötyä, joka myöhemmin joudutaan korjaamaan isomilla kustannuksilla, kutsutaan tekniseksi velaksi. Sekä tieteellinen yhteisö että ohjelmistoteollisuus tiedostavat teknisen velan hallitsemattoman kertymisen mittavaksi haasteeksi. Oikein hallinnoituna teknistä velkaa voidaan kuitenkin kerryttää myös hallitusti, jolloin ohjelmistokehitystyötä pystytään resursoimaan ja aikataulutumaan tehokkaammin. Tämä voidaan toteuttaa valitsemalla lyhyen aikavälin hyötyä tuottavat kohteet harkitusti niiden vaikutukset huomioiden ja huolehtimalla otetun velan takaisinmaksusta ennen sen tuottamien ongelmien kohtuutonta laajenemista. Ohjelmistoteollisuudessa ei kuitenkaan osata soveltaa teknisen velan hallinnoinnin menetelmiä ja työkaluja, minkä yhtenä syynä on teknistä velkaa koskevan viestinnän puute. Tämä tutkielma on kirjallisuuskatsaus, jossa käsitellään teknisen velan hallinnointia ja erityisesti sen viestinnän osa-aluetta. Tutkielmassa teknisen velan viestintää tarkastellaan ohjelmistokehitystä harjoittavan organisaation näkökulmasta. Viestintää toteuttavat organisaation toimijat jaotellaan vastuutahoihin heidän ohjelmistokehityksen työnkuviansa mukaisesti. Tutkielman tarkoituksena on selvittää, miten vastuutahot voivat toteuttaa keskenään teknisen velan viestintää. Tutkielman tuloksena on kooste teknisen velan viestintään käytettävistä menetelmistä ja työkaluista.

Asiasanat: tekninen velka, viestintä, organisaatio, ohjelmistokehitys

ABSTRACT

Kenias, Henri

Technical Debt Communication in Organization

Jyväskylä: University of Jyväskylä, 2022, 52 pp.

Information Systems, Bachelor's Thesis

Supervisor: Pekkala, Kaisa

Software development best practices can be understood as working methods of which one aim is to minimize the effort needed to further develop the software. Although these practices provide many benefits as the software ages, these long-term benefits are not always obvious. Software development industry is characterized by many development hampering aspects, such as tight schedules and ever-changing requirements for the software-in-development, which can make the following of best practices extremely hard. In these cases, it is common to use sub-optimal solutions to quickly get the work done. These solutions may provide short-term benefits for the development process, such as increased productivity, but at the cost of missing the best practices solution. Sub-optimal solutions may cause otherwise avoidable problems in the continued development of the software. These problems cascade as time goes on until either the sub-optimal solution is replaced, or the software has become unmaintainable. This software development phenomenon is called technical debt. Technical debt management is recognized as one of the most critical challenges faced in software development. When technical debt is managed correctly it transforms from an issue into an effective tool, which can be used to resource and schedule development work more efficiently. Current software industry is unable to apply technical debt management methods and tools in a productive way. A central contributor in the current state of technical debt management is the lack of communication about technical debt. This systematic literature review is a study into technical debt management with a focus on technical debt communication, a sub-category of technical debt management. In this study, technical debt communication is researched from a point-of-view of the developer organization. Organization's personnel practicing technical debt communication are categorized as stakeholders based on their professional roles. The goal of this study is to find out how different stakeholders practice technical debt communication. As a result, a summary of tools and methods used in technical debt communication is presented.

Keywords: technical debt, communication, organization, software development

KUVIOT

KUVIO 1	Teknisen velan perusmalli	13
KUVIO 2	Teknisen velan käsitelmä	16
KUVIO 3	Teknisen velan päätöksenteon elementit.....	19
KUVIO 4	Teknisen velan päätöksenteon elementit vastuutahottain.....	27
KUVIO 5	Teknisen velan vastuutahojen yhteiset intressit	28
KUVIO 6	Teknisen velan hallinnointi liiketoiminnan prosesseilla	36

TAULUKOT

TAULUKKO 1	Teknisen velan tyypit	14
TAULUKKO 2	Teknisen velan viestinnän menetelmien kooste.....	38
TAULUKKO 3	Teknisen velan viestinnän työkalujen kooste.....	43

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KUVIOT JA TAULUKOT

1	JOHDANTO.....	6
1.1	Tutkimusaiheen tausta.....	6
1.2	Tutkimuksen tavoite ja rakenne	8
1.3	Tutkimusmenetelmä	9
2	TEKNINEN VELKA JA SEN HALLINNOINTI	10
2.1	Teknisen velan määritelmä	10
2.2	Teknisen velan teoreettiset mallit.....	12
2.3	Teknisen velan hallinnointi.....	17
3	AIEMPI TUTKIMUS	21
3.1	Aiemmat tutkimukset teknisestä velasta	21
3.2	Aiemmat tutkimukset teknisen velan hallinnoinnista	23
4	TEKNISEN VELAN VASTUUTAHOT JA VIESTINTÄ	24
4.1	Vastuutahojen esittely	24
4.2	Vastuutahot ja päätöksenteko.....	26
4.3	Viestintä vastuutahojen välillä.....	28
4.4	Teknisen velan viestinnän lähestymistavat	30
5	TEKNISEN VELAN VIESTINNÄN MENETELMÄT JA TYÖKALUT	34
5.1	Teknisen velan viestinnän menetelmien esittely	34
5.2	Johtopäätökset teknisen velan viestinnän menetelmistä	38
5.3	Teknisen velan viestinnän työkalujen esittely.....	39
5.4	Johtopäätökset teknisen velan viestinnän työkaluista	42
6	YHTEENVETO JA POHDINTA	46
	LÄHTEET	49

1 JOHDANTO

Teknisen velan hallitsematon kertyminen tunnistetaan laajamuotoiseksi ongelmaksi niin tieteellisen yhteisön kuin ohjelmistokehitysteollisuudenkin toimesta. Teknisen velan hallinnointi on kuitenkin ohjelmistokehitystä harjoittavissa organisaatioissa havaittu tieteellisten tutkimusten perusteella usein joko puutteelliseksi tai olemattomaksi, mutta myös teknisen velan tutkimuksen voidaan katsoa olevan puutteellista tarjoamaan hallinnointiin soveltuvia menetelmiä ja työkaluja. Tutkimukset esittävät monesti puutteellisen hallinnoinnin osasyiksi teknisen velan viestinnän laiminlyönnin. Tämän tutkielman tutkimusaiheena onkin teknisen velan viestintä, jota tarkastellaan laajemman teknisen velan tutkimuksesta tehdyn kirjallisuuskatsauksen kautta. Tutkielma pyrkii edistämään tehokkaampaa teknisen velan viestintää esittelemällä siihen soveltuvia menetelmiä ja työkaluja, joita voidaan käyttää ohjelmistokehitystä harjoittavassa organisaatiossa sen eri vastuutahojen välillä. Tässä luvussa teknisen velan viestinnän tutkimusaihetta taustoitetaan esittelemällä kooste teknisen velan aiemmasta tutkimuksesta sekä esitellään tutkielman tutkimuskysymys, rakenne ja tutkimusmenetelmä.

1.1 Tutkimusaiheen tausta

Ohjelmistokehittäjien työtä ohjaavat ohjelmistokehitysorganisaation hallinnon määrittämät tavoitteet ja rajoitukset, kuten aikataulut ja käytettävät toimintamallit (Fernández-Sánchez, Garbajosa, Yagüe & Perez, 2017). Muun muassa Lim, Taksande ja Seaman (2012) sekä Kruchten, Nord ja Ozkaya (2012) kuvaavat esimerkein tilanteita, joissa hallinnon esittämät vaatimukset luovat kehittäjille painetta käyttää hyvien ohjelmistokehityskäytänteiden vastaisia oikopolkuja työn alla olevan iteraation tai ohjelmiston julkaisuversion ulossaattamiseksi vaaditun aikataulun mukaisesti. Oikopolut, kuten kehitettävän ohjelmiston dokumentaation päivittämättä jättäminen, voivat mm. nopeuttaa uuden version julkaisua tai lisätä hetkellisesti kehittäjien tuottavuutta. Oikopolut kuitenkin aiheuttavat

ohjelmiston iän karttuessa monenlaisia ongelmia, joiden korjaaminen vaatii ajan kuluessa usein lisääntyvässä määrin resursseja. Mikäli valittua oikopolkua ei missään vaiheessa toteuteta hyvien käytänteiden mukaisella ratkaisulla, ongelmat jatkavat kasvamistaan tehden ohjelmiston hallinnoinnin lopulta mahdottomaksi. Cunningham (1992) oli ensimmäisiä ilmiön kuvaajia ja kutsui tätä ohjelmistokehityksen ilmiötä tekniseksi velaksi.

Cunninghamin (1992) alkuperäinen määritelmä rajoittui käsittelemään vain ohjelmiston lähdekoodia ja sen aiheuttamia seurauksia, mutta myöhempi teknisen velan tutkimus on laajentanut teknisen velan määritelmän kattamaan myös muita ohjelmistokehityksen osa-alueita, joissa ohjelmistokehityksen työtehtävien siirtäminen tulevaisuuteen on vaihtokauppaa lyhyen aikavälin hyötyjen ja pitkän aikavälin kustannusten välillä (Lim ym., 2012). Nykyään yleisimmin käytetty teknisen velan määritelmä kattaakin monet ohjelmistokehityksen osa-alueet aina ohjelmiston lähdekoodista ja arkkitehtuurista sen vaatimusmäärittelyihin ja käytettävyyteen, kunhan edellä kuvattu ilmiö on havaittavissa niissä (Alves ym., 2016).

Tekninen velka on läsnä kaikessa ohjelmistokehityksessä, joskin tietyt toimitamallit, kuten ketterät menetelmät, kerääntyvät sitä toisia enemmän (Behutiye, Rodríguez, Oivo & Tosun, 2017). Tekninen velka ei silti aina ole vain negatiivinen tai edes epätoivottu tuote, vaan sen kohtuullista kerryttämistä voidaan käyttää osana ohjelmistokehityksen työn tehostamista, kunhan velka pysyy hallittavissa (mm. Seaman & Guo, 2011; Tom, Aurum & Vidgen, 2013; Ampatzoglou ym., 2015). Teknisen velan hallitsematonta kasaantumista pidetään kuitenkin yhtenä keskeisimmistä ohjelmistokehityksen haasteista (Lim ym., 2012). Teknisen velan rahallisten kustannusten, eli hinta joka kaiken olemassa olevan teknisen velan korjaaminen vaatisi, on jo 2010-luvun alussa arvioitu olevan satoja miljardeja dollareita ylittäen muun muassa koko sen hetkisen globaalin tietoturvan markkinat (Tom ym., 2013). Ohjelmistoteollisuudessa kyllä tiedostetaan teknisen velan ilmiön olemassaolo, mutta sen hallinnoinnin menetelmiä ei joko tunnisteta tai osata soveltaa käytännössä (Ernst, Bellomo, Ozkaya, Nord & Gorton, 2015).

Teknisen velan tieteellisiä tutkimuksia on monesti kritisoitu kyvyttömyydestä tarjota käytännön ympäristöihin soveltuvia teknisen velan hallinnoinnin ratkaisuja (mm. Ernst ym., 2015; Rios, de Mendonça Neto & Spínola, 2018; Nielsen, Østergaard Madsen & Lungu, 2020). Etenkin ohjelmistokehittäjät kokevat, ettei teknisen velan hallinnoinnin menetelmiä ja työkaluja ole saatavilla (Ernst ym., 2015) ja teknisen velan hallintointiin liittyvät empiiriset tutkimukset ovat kaiken kaikkiaan harvinaisia (Yli-Huumo, Maglyas & Smolander, 2016). Teknisen velan hallinnoinnin tutkimuksen vajavaisuuden voidaankin esittää olevan yksi tekijä osana havaittua käytännön puutteellista hallintointia.

Teknisen velan hallintointi jaetaan monissa tutkimuksissa erilaisiin toimenpiteisiin, kuten velan tunnistamiseen, määrän arviointiin ja takaisinmaksuun (Li, Avgeriou & Liang, 2015). Yhtenä useissa tutkimuksissa toistuvana hallinnoinnin toimenpiteenä on teknisen velan viestintä (mm. Li ym. 2015; Alves ym. 2016; Rios ym., 2018). Teknisen velan viestintä voidaan puolestaan määritellä

toimenpiteiksi, joilla tekninen velka saatetaan näkyville kaikille sen hallinnointiin osallistuville tahoille heidän ymmärtämässään muodossa (Li ym. 2015). Teknisen velan viestintä koetaan erityisen tarpeelliseksi ohjelmistokehitysorganisaation sisällä ohjelmistokehittäjien ja liiketoiminnan hallinnon välillä (Avgeriou, Kruchten, Ozkaya & Seaman, 2016). Teknisen velan viestintä on kuitenkin havaittu haasteelliseksi toteuttaa käytännössä. Eri tahot eivät esimerkiksi aina ole tietoisia siitä, minkälaisia teknisen velan vaikutuksia he painottavat omassa päätöksenteossään tai mitä seurauksia päätöksistä on toiselle osapuolelle. Tällaisissa tilanteissa viestinnän onnistuminen on erityisen tärkeää, jotta kunkin tahon tarpeet tulevat kommunikoiduiksi muille osapuolille (Ampatzoglou, Ampatzoglou, Chatzigeorgiou & Avgeriou, 2015). Apan, Jeronimon, Nascimenton, Vallespirin ja Travassosen (2020) mukaan teknisen velan viestintä onkin yksi tärkeimmistä teknisen velan hallinnoinnin toimenpiteistä.

1.2 Tutkimuksen tavoite ja rakenne

Teknisen velan viestintä on harvoin tieteellisten tutkimusten päätutkimusaiheena. Teknisen velan hallinnoinnin tutkimus, joka jo itsessään on vähemmän tutkittu aihealue kuin teknisen velan ilmiön kuvaamiseen liittyvä tutkimus, painottuu tiettyihin hallinnoinnin toimenpiteisiin ja kyseistä toimenpidettä avustaviin menetelmiin ja työkaluihin. Teknisen velan viestintä ei kuulu näihin painopisteinä oleviin tutkimusaiheisiin. Koska teknisen velan viestinnän menetelmiä ja työkaluja päätutkimusaiheenaan käsittelevät tutkimukset ovat harvinaisia, pyritäänkin tässä tutkielmassa vastaamaan seuraavaan kysymykseen:

Millaisilla keinoilla teknisestä velasta voidaan viestiä organisaation vastuutahojen välillä?

Lisäksi tutkielmassa esitellään käytännön työskentelymenetelmiä ja työkaluja teknisen velan tehokkaan viestinnän edistämiseksi. Tutkielman tarkoituksena on koostaa teknisen velan viestinnässä käytettäviä menetelmiä ja työkaluja, joiden avulla teknisen velan hallinnointia voidaan toteuttaa tehokkaasti myös ohjelmistoteollisuuden käytännön ympäristöissä.

Tutkielma koostuu johdannosta, yhteenvedosta ja pohdinnasta sekä sisälöluvuista, jotka käsittelevät seuraavia aihealueita:

- Teknisen velan ja sen hallinnoinnin esittely
- Aiemman tutkimuksen esittely
- Organisaation vastuutahojen ja teknisen velan viestinnän esittely
- Teknisen velan viestinnän menetelmien ja työkalujen koostaminen

Kussakin luvussa esitellään käsiteltävä aihealue, määritetään käsitteet ja havainnoidaan käsiteltävän aihealueen nykytilaa aiempien tutkimusten kautta.

1.3 Tutkimusmenetelmä

Tutkielma on toteutettu kirjallisuuskatsauksena. Tutkimusaineiston etsimiseen on käytetty ensisijaisesti tieteellisten julkaisujen hakupalvelu Google Scholaria. Aineistoa on haettu myös Jyväskylän yliopiston JYKDOK-tietueesta, joka on linkitetty useisiin akateemisiin tietokantoihin, kuten IEEE Xplore ja Association for Computing Machinery (lyh. ACM). Lisäksi Semantic Scholar -hakupalvelua on hyödynnetty tarkistamaan hakusanojen soveltuvuus tilanteissa, joissa edellä mainitut palvelut eivät ole tuottaneet hyödynnettävää hakutulosta. Aineiston kokoaminen on aloitettu teknisen velan perusteoksien etsinnällä käyttäen mm. hakusanojen ”tekninen velka”, ”teknisen velan hallinnointi” ja ”teknisen velan viestintä” englanninkielisiä vastineita. Hakusanoja on lisätty ja tarkennettu löydetyn aineiston perusteella. Myös Jyväskylän yliopiston julkaisuarkisto JYX:in sisältämiä teknistä velkaa käsitteleviä opinnäytetöitä käytettiin teknisen velan tutkimuksen perusteosten kartoittamiseen. Lisäksi Jyväskylän yliopiston kurssimateriaaleja on hyödynnetty tarkentamaan sekä teknisen velan että tekniseen velkaan epäsuorasti liittyvän muun tutkimuksen hakusanoja.

Hakutulosten suodatus on aloitettu tutkimuksen otsikon ja esikatselutekstin perusteella. Mikäli otsikko ja esikatseluteksti vastaavat haettua aihealuetta, on tutkimuksesta luettu vähintään abstrakti ja yhteenveto, jotta tutkimuksen hyödyllisyys on voitu arvioida. Lopuksi tutkimuksen laadukkuutta on arvioitu sen julkaisijalle myönnetyn Julkaisufoorumi-laatuluokituksen, tutkijoiden tunnettavuuden ja tutkimuksen saamien viittausten perusteella. Laadukkaaksi arvioidut tutkimukset on valittu osaksi lähdeaineistoa. Lähdeaineiston tutkimuksien viittaamia ja tutkimuksen julkaisualustan ehdottamia tutkimuksia on valittu osaksi lähdeaineistoa edellä kuvatuin kriteerein.

Teknisen velan perusteoksiksi tunnistetut tutkimukset on julkaistu pääosin 2010-luvulla, kuten suurin osa muusta teknisen velan tutkimuksesta. Perusteokseksi on mielletty tutkimus, jonka tälle tutkielmalle keskeisen käsitteen, mallin tai muun konseptin määritelmä on laajasti käytössä myös muussa tutkielman lähdeaineistossa. Esimerkkejä perusteoksista ja niiden valintaperusteista ovat Li ym. (2015) ja Alves ym. (2016) teknisen velan tyyppijaottelun esittelyn ansiosta sekä Avgeriou ym. (2016) teknisen velan käsittemallin ansiosta. Tutkimusten julkaisuajankohtaa ei ole käytetty rajoittavana hakutekijänä, vaan lähdeaineiston tutkimukset on valittu niiden sisällön ajankohtaisuuden perusteella. Vain teknisen velan viestintään keskittyviä, viestinnän menetelmiä ja työkaluja yhteen koavia koosteita ei tähän tutkimukseen tehdyn aineiston keruun yhteydessä löytynyt.

Lähdeaineiston käsitteistöä on suomennettu käyttäen Redfox- ja Sana-kirja.fi-palveluita. Kaikille lähdeaineiston käyttämille käsitteille ei kuitenkaan löytynyt suoraa tai vakiintunutta suomennosta. Näille käsitteille on Google-hakukoneella etsitty toista, samaa määritelmää kuvaavaa käsitettä. Mikäli soveltuva käsitettä ei löytynyt, on sen suomentanut tämän tutkielman tekijä parhaan ymmärryksensä mukaisesti.

2 TEKNINEN VELKA JA SEN HALLINNOINTI

Tekninen velka voidaan mieltää ohjelmistokehitysprosessissa tehdyiksi päätöksiksi, joista voi seurata lyhyen aikavälin hyötyä mutta joista seuraa aina pitkän aikavälin kustannuksia. Tekninen velka voidaan jaotella teoreettisten mallien avulla erilaisiin osakokonaisuuksiin mm. sen aiheuttajan tai seurausten perusteella. Jaotteluiden perusteella pystytään toteuttamaan teknisen velan hallinnointia, eli toimenpiteitä joilla teknisen velan hallitsematonta kertymistä pyritään estämään ja pitämään jo kerrytetyn teknisen velan määrä kohtuullisena. Tämä luku esittelee teknisen velan sekä teknisen velan hallinnoinnin. Kahdessa ensimmäisessä alaluvussa määritellään teknisen velan käsite ja peruskäsitteistö sekä esitellään velan teoreettisia malleja. Viimeisessä alaluvussa määritellään teknisen velan hallinnoinnin käsite ja käsitteistöä sekä esitellään vaatimukset teknisen velan hallinnoinnin käytännön toteuttamiselle.

2.1 Teknisen velan määritelmä

Ohjelmistokehitysteollisuudessa ei aina huomioida hyvien ohjelmistokehityskäytänteiden merkitystä ohjelmistoa kehittävän tai ylläpitävän organisaation liiketoiminnalle. Hyvät ohjelmistokehityksen käytänteet voidaan yleisellä tasolla määritellä sisältämään kaikki toimenpiteet, joilla pyritään minimoimaan ohjelmiston jatkotyöstämiseen käytettävä vaiva. Näitä hyviä käytänteitä ovat mm. ohjelmiston rakenteen ja lähdekoodin tarpeettoman monimutkaisuuden karsiminen sekä dokumentaation ajantasaisuuden varmistaminen (Li ym., 2015). Hyvät käytänteet eivät rajoitu vain kehitettävään ohjelmistoon suoraan liittyviin toimenpiteisiin, vaan sisältävät kaikki kehitysprosessiin vaikuttavat tekijät, kuten ohjelmistokehitykseen soveltuvan it-infrastruktuurin ylläpidon sekä kehitystii- melle soveltuvien toimintamallien noudattamisen (Alves ym., 2016). Vaikka edellä kuvattujen hyvien käytänteiden noudattaminen tuottaakin monia hyötyjä ohjelmiston elinkaaren edetessä, eivät niiden lyhytaikaiset vaikutukset ole ilmeisiä kaikille ohjelmistokehitykseen osallistuville toimijoille. Etenkin

organisaatioiden liiketoiminnasta vastaavat toimijat, jotka liiketoiminnan tavoitteiden määrittämisen lisäksi myös asettavat mm. ohjelmistokehitystyön resurssit, budjetin, aikataulut ja muut vaatimukset, painottavat vaatimuksissaan enemminkin uusien toiminnallisuuden ja muiden loppukäyttäjälle välittömästi näkyvien ohjelmiston osa-alueiden tuottamista kuin pitkän aikavälin kehityksen hyötyjen tavoittelemista. Keskeiseksi syyksi tähän painotukseen on esitetty, ettei organisaatioiden päätöksenteossa tiedosteta kehitystyön ja liiketoiminnan tavoitteiden vuorovaikutussuhteita (Lim, Taksande & Seaman, 2012). Kuten myöhemmin tässä tutkielmassa esitellään, voivat lyhyen aikavälin hyötyjä tavoittelevat ohjelmistokehityksen ratkaisut tuottaa mittavia, koko organisaation toimintaan vaikuttavia pitkäaikaisia seurauksia.

Edellisessä kappaleessa esitetty ilmiö voidaan mieltää kaupankäynniksi, jossa ohjelmistokehityksen työtehtävien siirtäminen tulevaisuuteen on vaihtokauppaa lyhyen aikavälin hyötyjen ja pitkän aikavälin kustannusten välillä (Lim ym., 2012). Mitä isompi työtehtävä siirretään tulevaisuuteen, eli toisin sanoen myöhäistetään, sen isommat sen aiheuttamat ongelmat todennäköisesti tulevat olemaan. Pelkkä ongelmien korjaaminen niiden ilmetessä ei yleensä ole pitkäikäinen ratkaisu, vaan ongelmien lopullinen korjaaminen vaatii lähes aina myöhäistetyn työtehtävän toteuttamista (Tom ym., 2013). Tomin ym. (2013) mukaan kaikki työtehtävän tuottamat ongelmat aina sen myöhäistämispäätöksen hetkestä työtehtävän toteuttamiseen asti voidaan mieltää ylimääräiseksi työksi, joka olisi voitu välttää hyviä käytänteitä noudattavalla toteutuksella. Mikäli työtehtävää ei missään vaiheessa suoriteta, riskeerataan ohjelmiston jatkokehityksen ja ylläpidon muuttuminen mahdottomaksi. Näitä potentiaalisia lyhyen aikavälin hyötyjä tavoittelevia, mutta pitkän aikavälin ongelmia tuottavia ratkaisuja, voidaan kuvata teknisen velan vertauskuvalla.

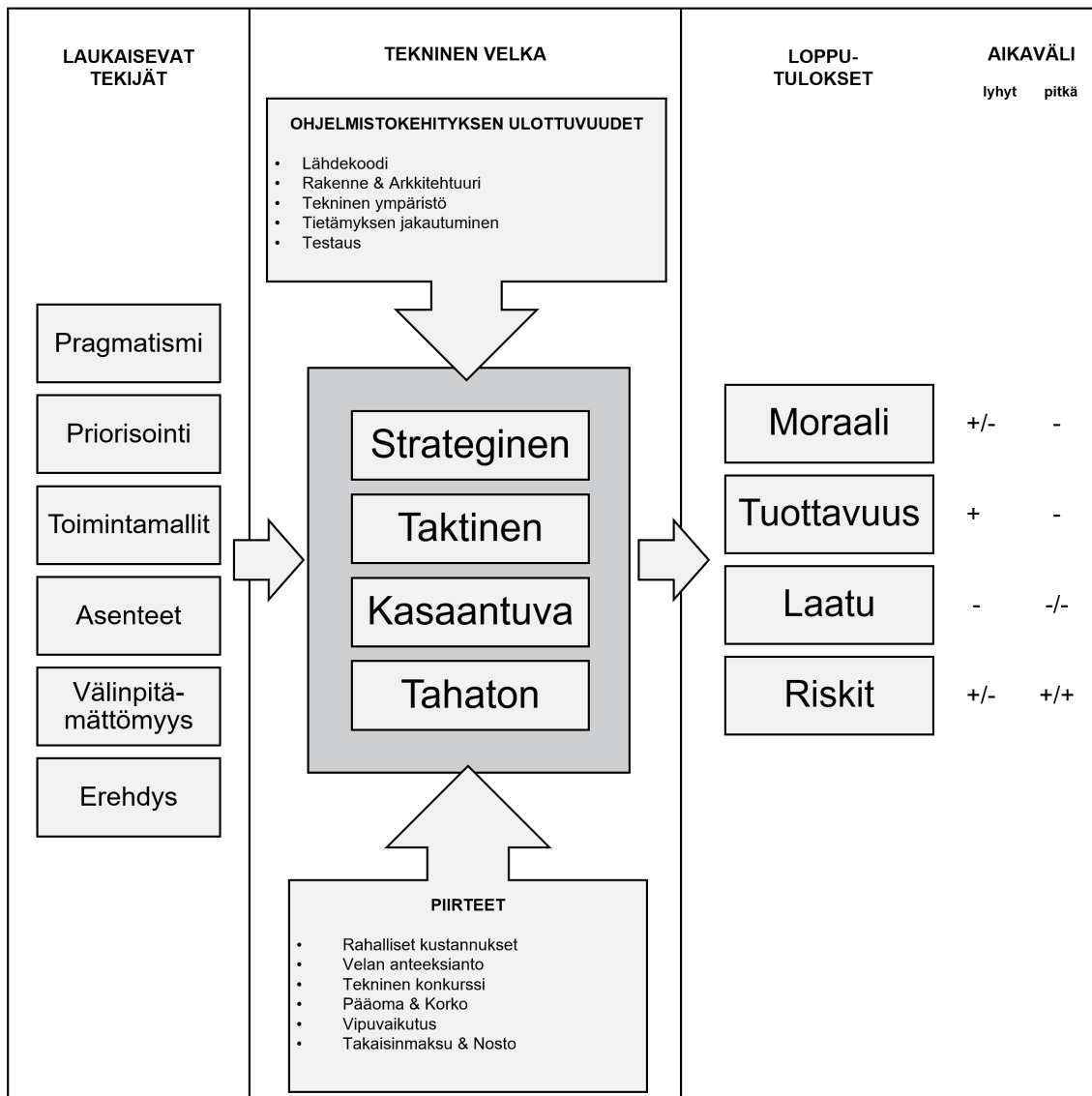
Cunningham (1992) oli ensimmäinen, joka kutsui edellä kuvattua ilmiötä vertauskuvallisesti tekniseksi velaksi. Teknisen velan vertauskuva soveltaa taloustieteen käsitteistöä tämän ohjelmistokehityksen ilmiön havainnollistamiseen. Ohjelmistokehityksessä pyritään vertauskuvan mukaan saavuttamaan lyhytaikaista hyötyä mahdollistamalla eniten lisäarvoa tuottavien toimenpiteiden suorittaminen välittömästi, kuten taloudellinen velan ottaminen mahdollistaa investointien toteuttamisen välittömästi. Taloudellisen velan tavoin myös tekninen velka täytyy maksaa takaisin korkoineen. Teknisen velan korko voidaan mieltää ylimääräiseksi työpanokseksi tai muuksi kustannukseksi, jota teknisen velan keräily on tuottanut. Tämä tutkielma käyttää Avgerioun ym. (2016) määritelmiä, pois lukien koron takaisinmaksun todennäköisyys, teknisen velan vertauskuvan keskeisimmistä taloustieteen käsitteistä. Avgerioun ym. (2016) tutkimus ei sisällä koron takaisinmaksun määritelmää, mutta sen määritelmät muille käsitteille on tutkielman kirjoittaja arvioinut laadukkaimmiksi. Takaisinmaksun todennäköisyyttä määritelmäksi on valittu varhaisempi, mutta myös tunnettu Seamanin ja Guon (2011) määritelmä, ja tutkija Seaman on osallisena kummassakin tutkimuksessa. Teknisen velan peruskäsitteet määritellään seuraavasti:

- **Pääoma.** Teknisen velan kerryttämisen hetkellä pääoma on myöhäistetyin toimenpitein avulla saatu hyöty, mutta pääoma muuttuu ajan myötä kustannuksiksi, joita myöhäistetyin toimenpitein toteuttaminen vaatisi kyseisellä hetkellä. Kustannuksia voidaan arvioida monilla mittareilla, kuten työaikana tai rahallisina kuluina.
- **Takaisinmaksu.** Teknisen velan takaisinmaksu on pääoman poistamiseksi vaadittavat toimenpiteet ja niiden kustannukset.
- **Korko.** Teknisen velan korko on pääoman takaisinmaksamatta jättämisestä aiheutuvat ohjelmistokehitykseen liittyvät ongelmat. Korko voi olla toistuvaa, kuten ongelmien korjaamiseen käytetyn ajan takia laskenut tuottavuus, tai kertyvää, kuten ongelmallisten ohjelmiston osa-alueiden perusteella kehitetyt uudet toiminnallisuudet.
- **Koron todennäköisyys.** Koron todennäköisyys kuvaa, millä todennäköisyydellä ja aikataululla teknisen velan korko tulisi maksettavaksi, mikäli pääoma jätetään maksamatta takaisin.

Kuten määrittelyistä huomataan, on tekninen velka aikasidonnaista. Pääoma ja korko, koron todennäköisyys sekä takaisinmaksun toimenpiteet siis vaihtelevat ajankulun ja ohjelmiston elinkaaren vaihteiden mukaan. Teknisen velan vertauskuvasta ja käsitteistöstä on kehittynyt aihepiirin tutkimuksen yleiskieltä, jota Cunninghamin (1992) jälkeiset tutkimukset käyttävät lähes poikkeuksetta ilmiötä tarkastellessaan. Kuitenkin itse teknisen velan termiä käytetään ilmiön kuvaamisen lisäksi myös synonyymina yksittäiselle teknisen velan ilmentymälle, joukolle ilmentymiä tai kuvaamaan kaikkien ilmentymien kokonaisuutta. Esimerkiksi Lim ym. (2012) kuvauksessaan, kuinka ”tietyt projektiryhmät kerryttivät teknistä velkaa, koska...” kuvaavat, kuinka teknisen velan kokonaisuus kasvatettiin organisaatiossa. Teknisen velan ilmentymä esitellään tarkemmin seuraavassa alaluvussa, mutta termin tavoiteltu merkitys on kuitenkin monesti helppo ymmärtää sen ympäröivän kontekstin perusteella ja tässäkin tutkielmassa teknistä velkaa käytetään sen kummassakin merkityksessä.

2.2 Teknisen velan teoreettiset mallit

Yksi varhaisimmista teknisen velan malleista on Tomin ym. (2013) teknisen velan perusmalli (kuvio 1), joka toimii monien myöhempien tutkimuksien teoreettisena perustana tai sen merkitys teknisen velan tutkimukselle tunnustetaan muutoin (mm. Alves, Ribeiro, Caires, Mendes & Spínola, 2014; Li ym. 2015; Ernst ym. 2015). Teknisen velan perusmallin (kuvio 1) voidaan katsoa koostaneen sitä edeltäneen tutkimuksen muotoon, jota huolimatta sen suhteellisesta yksinkertaisuudesta verrattuna myöhempiin tutkimuksiin, voidaan soveltaa tässäkin tutkielmassa teknisen velan vaiheiden ja koostumuksen kuvaamiseen.



KUVIO 1 Teknisen velan perusmalli (Tomin ym., 2013 mukaan)

Teknisen velan perusmalli (kuvio 1) kuvaa teknisen velan ilmentymää, eli yksittäistä teknisen velan yksikköä, jolle voidaan osoittaa yksi tai useampi laukaiseva tekijä ja lopputulos. Perusmallista voidaan havaita teknisen velan koostuvan monista muista osa-alueista kuin vain aiemmin esitellyistä pääomasta ja korosta. Perusmallin mukaan teknisen velan syntymiseen tarvitaan yksi tai useampi laukaiseva tekijä, eli teknisen velan synnyn mahdollistava käytäntö tai toimenpide. Laukaiseva tekijä voi olla esimerkiksi aikataulupaineista johtuva ohjelmistokehitystöiden priorisointi tai ohjelmistokehittäjien huolimattomuus. Syntynyt tekninen velka voidaan luokitella kahdella asteikolla: strateginen - taktinen ja kasaantuva - tahaton. Strateginen - taktinen -asteikko ottaa kantaa velan tavoitteeseen. Strateginen velka pyrkii hyödyttämään kehitystyötä pitkällä aikavälillä esimerkiksi jättämällä tietty kehitettävän ohjelmiston toiminnallisuus viimeistelemättä, jotta kehitystyössä voidaan keskittyä tärkeämmiksi koettuihin

toimintoihin. Taktisella velalla reagoidaan välittömiin tekijöihin, kuten lähestyviin määräpäiviin, joiden saavuttaminen aikataulussa olisi muutoin haastavaa. Kasaantuva velka on kehittäjien ottamia useita pieniä hyvien ohjelmistokehityskäytänteiden vastaisia ratkaisuja, jotka pitkällä aikavälillä kasaantuvat ja aiheuttavat velan kertymistä. Tahaton velka vastaa muutoin kasaantuvaa velkaa, mutta sen ottamisen syyt eivät ole tietoisia vaan johtuvat esimerkiksi kehittäjien osaamattomuudesta tai huolimattomuudesta. Monet seuraavat tutkimukset käyttävätkin kasaantuvan velan sijaan asteikkoa tietoinen - tahaton tekninen velka (Li ym., 2015). Teknisellä velalla on myös ohjelmistokehityksen ulottuvuuksia ja ohjelmistokehityksen ulkopuolisia piirteitä, joilla voidaan kuvata teknisen velan esiintymismuoto todellisessa maailmassa usein taloustieteen termistöä käyttäen. Teknisen velan ohjelmistokehityksen ulkopuolisiksi piirteiksi voidaan Tomin ym. (2013) mukaan laskea käytännössä kaikki teknisen velkaa kuvaavat tekijät, jotka eivät sisälly sen ohjelmistokehityksen ulottuvuuksiin. Velan lopputuloksena on positiivinen tai negatiivinen vaikutus ohjelmistokehitysorganisaation moraaliin sekä tuottavuuteen, ohjelmiston laatuun ja kehitysprossin riskeihin.

Li ym. (2015) sekä Alves ym. (2016) laajentavat teknisen velan perusmallissa (kuvio 1) esiteltyjä ohjelmistokehityksen ulottuvuuksia kategorisoimalla ne teknisen velan tyypeiksi. Kumpikin tutkimus on itsenäisesti koostanut velan tyyppit aiempien teknisen velan tutkimusten perusteella. Lin ym. (2015) jaottelussa tekninen velka on jaettu 10 tyyppiin, joihin Alves ym. (2016) löysivät viisi uutta tyyppiä nostaan tyyppien lukumäärän yhteensä 15 kappaleeseen. Alvesin ym. (2016) laajempi jaottelu perustuu sen myöhäisempään julkaisuajankohtaan, jonka väliaikana julkaistuista tutkimuksista löydettiin uusia teknisen velan tyyppejä. Molemmat jaottelut ovat laajasti käytössä teknisen velan tutkimuksessa. Tutkimusten jaottelut on koottu seuraavaan taulukkoon (taulukko 1).

TAULUKKO 1 Teknisen velan tyyppit
(Li ym., 2015; Alves ym., 2016 mukaan)

Teknisen velan tyyppi	Li ym. (2015)	Alves ym. (2016)
Arkkitehtuuri	X	X
Dokumentaatio	X	X
Henkilöstö		X
Infrastruktuuri	X	X
Koodi	X	X
Koontiversio	X	X
Käytettävyys		X
Ohjelmistotestaus	X	X
Palvelut		X
Rakenne	X	X
Testauksen automaatio		X
Toimintamallit		X
Tuotevirhe	X	X
Vaatimukset	X	X
Versiointi	X	X

Taulukon 1 teknisen velan tyypit kuvaavat, millä ohjelmistokehitysproessin osa-alueilla tekninen velka voi sijaita. Esimerkiksi ohjelmiston lähdekoodiin perustuvat laatuvirheet, kuten toistuvat koodirivit, luokitellaan koodityypin tekniseksi velaksi. Ohjelmistokehitysjärjestelmän henkilöstöön perustuva, kuten henkilöstön ammatillisen tietotaidon liiallinen kasaantuminen vain tietyille henkilöille tai ryhmälle, tekninen velka puolestaan luokitellaan henkilöstövelaksi.

Izurietan ym. (2016) nostaa esiin taulukon 1 tyyppien välisiä eroja sen perusteella, kuinka helposti velka on mitattavissa. Mittaamisella pyritään luomaan velasta laadullinen arvio, jota voidaan hyödyntää teknisen velan hallinnoinnissa (Seaman & Guo, 2011). Teknisen velan hallinnointia käsitellään seuraavassa alaluvussa. Teknisen velan mittaaminen riippuu velan tyypistä ja velan tyypistä riippuen voidaan käyttää eri mitta-asteikkoja. Esimerkiksi koodityypin velkaa voidaan mitata sen aiheuttamien järjestelmävirheiden tai havaittujen huonojen ohjelmointikäytänteiden määrän perusteella (Seaman & Guo, 2011). Izurietan ym. (2016) mukaan helposti mitattaviksi tyypeiksi esitetään juuri koodityypin velkaa, jota voidaan mitata erilaisten automaattisten työkalujen tuottamien numeraalisten tulosten muodossa. Vaikeasti mitattavia tyypejä on mm. edellisessä kappaleessa mainittu henkilöstövelka, joka vaatii henkilöstön sosiaalisen kanssakäymisen ja organisaation toimintamallien subjektiivista arviointia.

Tämän tutkielman kannalta keskeisimmät teknisen velan tyypit ovat arkkitehtuuri-, rakenne- ja koodivelka. Tutkielman kirjoittaja tulkitsee nämä teknisen velan tyypit kattamaan koko kehitettävän ohjelmiston toiminnallisuuden sisältämän teknisen velan, sillä Alvesin ym. (2016) mukaan ne kattavat yhdessä ohjelmiston komponenttien välisen vuorovaikutuksen, komponenttien sisäisten toiminnallisuuden vuorovaikutuksen sekä lähdekoodin jäsentämisen, tai yleiskielellä luettavuuden. Näistä kolmea edellä mainittua teknisen velan tyyppiä kuvataan jatkossa käsitteellä *ohjelmiston toiminnallisuuden velka*.

Vuonna 2016 järjestettiin teknisen velan johtavista tutkijoista, opiskelijoista, työkalujen toimittajista ja muista ammatinharjoittajista koostuva teknisen velan hallinnoinnin seminaari (Avgeriou ym., 2016), jossa esiteltiin teknisen velan käsitelmä (kuvio 2). Käsitelmä määrittelee teknisen velan keskeiset osa-alueet ja niiden väliset vuorovaikutussuhteet teknisen velan perusmallia (kuvio 1) tarkemmalla tasolla. Tämä tarkempi määrittely ja vuorovaikutussuhteiden kuvaus luo teoreettisen pohjan teknisen velan hallinnoinnin toteuttamiselle.

- **Teknisen velan aiheuttaja.** Teknisen velan aiheuttaja on myöhäistetty ohjelmistokehityksen toimenpide, joka voi johtua tietoisista päätöksistä, projektin aikataulutuksesta, käytetyistä työskentelymenetelmistä tai muista ohjelmistokehityksen toimenpidettä lykkäävistä syistä. Myöhäistetty toimenpide laukaisee teknisen velan kappaleen synnyn. Aiheuttaja on verrannollinen luvussa 2.1 kuvattuun samannimiseen toimenpiteeseen sekä ja kuvaa teknisen velan perusmallin (kuvio 1) laukaisevia tekijöitä perusmallia konkreettisemmalla tasolla.
- **Ohjelmistokehityksen artefakti.** Ohjelmistokehityksen artefakti kuvaa, mihin ohjelmistokehitysprosessin osa-alueeseen teknisen velan kappaleen aiheuttaja tuottaa teknisen velan pääomaa. Käsittemallin artefaktien, eli koodin, testauksen, dokumentaation ja tuotevirheiden voidaan katsoa vastaavan teknisen velan tyyppejä (taulukko 1), sillä kumpikin kuvaa millä teknisen velan ohjelmistokehitysprosessin osa-alueella velka sijaitsee. Tässä tutkimuksessa käytetäänkin jatkossa käsitettä *teknisen velan tyyppi* artefaktin käsitteen sijaan, ja määritelmä sisältää kaikki taulukossa 1 esitellyt tyypit. Myös Rios ym. (2018) ovat laajentaneet teknisen velan käsittemallia sisältämään taulukon 1 teknisen velan tyypit, mutta tutkimuksessa esitetty malli ei sovellu tähän tutkielmaan.
- **Teknisen velan seuraus.** Teknisen velan kappale voi aiheuttaa seurauksia ohjelmiston koettuun arvoon ja laatuun, ohjelmistokehitysprosessin aikatauluihin sekä jatkokehityksen kustannuksiin. Seuraukset vaikuttavat organisaation liiketoiminnallisiin tavoitteisiin välillisesti seurauksen sisältävän ohjelmiston toiminnallisuuden kautta.

Teknisen velan perusmalli ja käsittemalli ovat yleisesti käytettyjä teknisen velan osa-alueiden ja vuorovaikutussuhteiden havainnolistamisen menetelmiä. Kyseiset mallit, niiden luonteesta johtuen, käsittelevät teknistä velkaa vain teoreettisella tasolla. Mallien sisäistäminen kuitenkin avaa edellytykset teknisen velan hallinnoinnin ymmärtämiselle.

2.3 Teknisen velan hallinnointi

Teknisen velan hallinnointi voidaan määrittää joukoksi toimenpiteitä, joilla potentiaalisen teknisen velan kertymistä pyritään estämään ja pitämään olemassa oleva tekninen velka kohtuullisessa määrässä (Li ym., 2015). Kuten teknisen velan perusmallin (kuvio 1) ja käsittemallin (kuvio 2) perusteella voidaan havaita, tulee teknisen velan hallinnoinnin toimenpiteitä suunniteltaessa huomioida useita eri muuttujia ja vuorovaikutussuhteita. Vaikka Izurieta ym. (2016) esittivät aiemmassa luvussa tiettyjen teknisen velan tyyppien olevan helpommin mitattavia kuin toisten, esittävät mm. Seaman ja Guo (2011) sekä Marinescu (2012) teknisen velan olevan lähtökohtaisesti taloudellista velkaa vaikeammin mitattavissa. Esimerkiksi monet teknisen velan hallinnoinnille oleelliset tiedot eivät ole samalla tavalla suoraan saatavilla kuin esimerkiksi pankkilainan korko, vaan jo pelkän

teknisen velan koron määrän arviointiin saatetaan tarvita monimutkaisia ja paikoin subjektiivisiakin arvioita (Seaman & Guo, 2011).

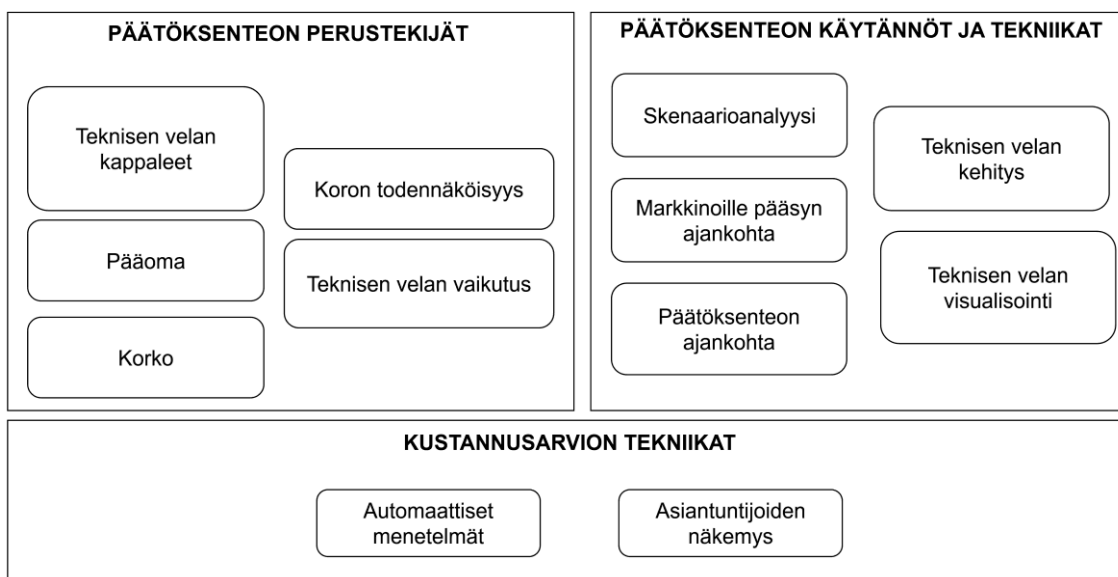
Teknisen velan hallinnoinnin ongelmien ratkaisun lähtökohdaksi Avgeriou ym. (2016) esittävät teknisen velan hallinnoinnin keskeiseksi sisällöksi teknisen velan tunnistamisen, teknisen velan laadun analysoinnin, teknisen velan mittaamisen sekä teknisen velan seurannan. Näitä osa-alueita on kuvattu tarkemmin muissa tutkimuksissa, joiden kautta alueet esitellään seuraavana.

- **Teknisen velan tunnistaminen.** Teknisen velan tunnistaminen käsittää teknisen velan kappaleiden havaitsemisen ohjelmistokehitysprosessista. Kappaleita voidaan etsiä joko automaattisilla työkaluilla tai manuaalisesti. Spínola, Zazworka, Vetro, Shull ja Seaman (2019) tarkentavat tunnistamisen menetelmiä vertailemalla automaattisia työkaluja manuaaliseen tunnistamiseen. Automaattisilla työkaluilla etsitään usein niiden teknisen velan tyyppien kappaleita, joista pystytään löytämään numeraalisesti mitattavia muuttujia. Automaattisten työkalujen käyttö painottuu vahvasti luvussa 2.2 esiteltyyn ohjelmiston toiminnallisuuksien velkaan (Rios ym., 2018). Manuaalinen etsintä puolestaan on ohjelmiston kehittäjän tai muun vastuutahon suorittamaa teknisen velan kappaleiden paikantamista. Manuaalinen etsintä on usein hitaampaa ja epätarkempaa kuin automaattisilla työkaluilla. Kaikissa kehitysorganisaatioissa työkaluja ei kuitenkaan ole saatavilla eikä monia teknisen velan tyyppisiä pystytä havaitsemaan automaattisesti.
- **Teknisen velan laadun analysointi ja mittaaminen.** Teknisen velan mittaamista alustettiin edellisessä luvussa 2.3. Havaitusta teknisen velan kappaleesta voidaan arvioida sekä laadullisesti että määrällisesti, mutta molempien arviointimenetelmien tavoitteena on tuottaa havaitusta teknisen velan kappaleesta sen hallinnointia hyödyttävää informaatiota. Yleisimpiä arvioinnin keinoja ovat erilaisten matemaattisen laskentakaavojen ja mallien hyödyntäminen (Li ym., 2015). Esimerkkinä matemaattisesta lähestymistavasta on SQALE-menetelmä, jota käyttämällä ohjelmiston toiminnallisen velan kappaleesta voidaan automaattisin työkaluin laskea mm. sen takaisinmaksun kustannukset (Letouzey, 2012). Ilman automaattisia työkaluja suoritettava, asiantuntijoiden "valistuneisiin arvauksiin" perustuva teknisen velan kappaleen arviointi on kuitenkin monesti yleisempi lähestymistapa (Spínola ym., 2019). Spínolan ym. (2019) mukaan tärkeimmät teknisen velan kappaleesta selvittävät tiedot ovat kappaleen pääoma, korko ja koron todennäköisyys. Edellä mainituista kolmesta arvosta pystytään koostamaan mm. teknisen velan listaus sekä johtamaan muita teknisen velan hallinnoinnin työkaluja.
- **Teknisen velan seuranta.** Teknisen velan seurannalla velan pääomaa ja tyyppisiä arvioidaan jatkuvasti sekä pyritään havaitsemaan takaisinmaksun kohteita. Arvanitou, Ampatzoglou, Bibi, Chatzigeorgiou ja Stamelos (2019) esittävät tarkentavan määritelmän seurattavista kohteista. Erityisen seurannan kohteena tulee olla teknisen velan kappaleiden kokonaismäärä,

korkean teknisen velan pääoman kappaleiden lukumäärä, korkean koron ja koron todennäköisyyden kappaleet sekä painotettu velan kokonaismäärä. Painotetussa velan kokonaismäärässä teknisen velan kappaleet pisteytetään niiden pääoman mukaisesti ja pisteet lasketaan yhteen. Havainnoimalla edellä mainittuja mittareita voidaan teknisen velan kehitystä seurata ja ennakoida. Li ym. (2015) lisäävät teknisen velan hallinnoinnin eroavan monista muista ohjelmointikehityksen toimintatavoista sillä, että teknisen velan hallinnointi on läsnä koko ohjelmiston elinkaaren ajan suunnitteluvaiheesta ylläpitovaiheeseen ja vaatii aktiivista seuranta-aikaa aina siihen asti, kunnes ohjelmisto poistuu käytöstä.

Edellä kuvatuilla toimenpiteillä pystytään keräämään teknisen velan kappaleiden sisältämä informaatio teknisen velan kerryttämisen tai takaisinmaksun päätöksistä.

Pelkkä teknisen velan kappaleiden tarkastelu ei vielä tuota kaikkea tarvittavaa informaatiota teknisen velan tehokkaaseen hallinnointiin, vaan on huomioidava myös teknisen velan käsitteellisissä (kuvio 2) esitetyt muut kappaleiden epäsuorat vuorovaikutussuhteet ja etenkin vaikutukset kehitysorganisaation liiketoimintaan. Fernández-Sánchez ym. (2017) esittelevätkin teknisen velan hallinnoinnin päätöksenteon elementit (kuvio 3), joihin sisältyy jo tässä tutkielmassa esiteltyjen elementtien lisäksi myös elementtejä, joista ei ole mahdollista saada tietoa vain teknisen velan kappaleita arvioimalla, kuten mikä on organisaatiossa kehitettävän ohjelmistotuotteen arvioitu markkinoille pääsyn ajankohta ja miten se vaikuttaa tekniseen velkaan liittyviin päätöksiin.



KUVIO 3 Teknisen velan päätöksenteon elementit (Fernández-Sánchezin ym., 2017 mukaan)

Kuvion 3 päätöksenteon elementeillä kuvataan teknisen velan hallinnoinnin päätöksentekoon vaikuttavia tekijöitä. Elementit on kategorisoitu päätöksenteon perustekijöihin, kustannusarvioinnin tekniikoihin sekä päätöksenteon käytäntöihin ja tekniikoihin. Päätöksentekijöitä kuvaillaan seuraavasti:

- **Päätöksenteon perustekijät.** Päätöksenteon perustekijät kuvaavat ohjelmistokehitysprosessiin kertynyttä teknistä velkaa. Perustekijät koostuvat jo aiemmin tässä tutkimuksessa esitellyistä teknisen velan kappaleista ja niiden vaikutuksista sekä velan pääomasta, korosta ja koron takaisinmaksun todennäköisyydestä.
- **Kustannusarvion tekniikat.** Kustannusarvion tekniikat ovat edellytyksiä teknisen velan hallinnoinnin käytännön soveltamiselle. Kustannusarvion tekniikat sisältävät teknisen velan hallinnoinnissa käytettävät menetelmät ja työkalut, automaattisten menetelmien kuvatessa ohjelmistokehittäjien päivittäistyötä häiritsemättömiä teknisen velan tiedonkeruun menetelmiä ja asiantuntijoiden näkemys tilanteita, joissa päätöksentekoon tarvittava tieto vaatii asiantuntijan, kuten ohjelmistokehittäjän, arviointia.
- **Päätöksenteon käytännöt ja tekniikat.** Päätöksenteon käytännöt ja tekniikat kuvaavat ohjelmiston ulkopuolisten tekijöiden vaikutusta teknisen velan hallintaan. Edes täydellinen tietoisuus ohjelmistokehitysprosessin teknisestä velasta ei ole riittävä päätöksenteon perusta, vaan myös ohjelmistokehitysprosessin hallinnolliset tekijät tulee huomioida osana päätöksentekoa.

Fernández-Sánchez ym. (2017) painottavat, että elementtien olemassaolon tiedostamista vaaditaan kaikilta ohjelmistokehityksen toimijoilta. Teknisen velan hallinnoinnin menetelmien ja työkalujen soveltaminen ei ilman elementtien tuntemusta ole mahdollista, käsittäen myös teknisen velan viestinnän. Käytännössä teknisen velan hallinnointia harvoin toteutetaan Avgeriou ym. (2016) esittämien lähtökohtien mukaisesti eivätkä Fernández-Sánchezin ym. (2017) esittämät vaatimukset päätöksenteon elementtien tuntemuksesta täyty. Ennen siirtymistä teknisen velan viestintään onkin syytä tarkastella, mitä aiempia tutkimuksia teknisestä velasta ja sen hallinnoinnista on tehty sekä miksi ehdotettuja hallinnoinnin toimenpiteiden toteuttamista ei ohjelmistoteollisuudessa noudateta.

3 AIEMPI TUTKIMUS

Teknisen velan tutkimusta vaivaa usea tutkimusaiheen kehittymistä hidastava tutkimuskäytänne. Ristiriitaisten tutkimusten ja tutkimusaiheiden epäsuhtaisen painottumisen lisäksi myös etenkin käytännössä sovellettavat empiiriset teknisen velan tutkimukset ovat harvinaisia. Huonoja tutkimuskäytänteitä ja puutteellista tutkimustietoa voidaan pitää vähintäänkin osasyllisenä siihen, että tehokas teknisen velan hallinnointi ohjelmistokehitysteollisuudessa on enemmän poikkeus kuin sääntö. Tässä luvussa esitellään teknisen velan ja sen hallinnoinnin aiempia tutkimuksia. Tämän luvun sisältämät aiemman tutkimuksen epäkohdat ovat perustana tutkielman seuraavassa pääluvussa esiteltäville teknisen velan viestinnän ongelmille.

3.1 Aiemmat tutkimukset teknisestä velasta

Suurin osa teknisen velan tutkimuksesta on peräisin 2010-luvulta. Tutkimuksien suhteellisen nuoruuden ja vähäisen määrän johdosta teknisen velan tutkimus koetaan usein puutteelliseksi (Nielsen ym., 2020). Keskeisiksi tutkimusaiheen ongelmiksi nostetaan usein yhtenäisten määritelmien puute, julkaistujen tutkimusten vähäinen lukumäärä etenkin käytännössä sovellettavien teorioiden osalta ja tutkimuksen keskittyminen uusien teoreettisten mallien luomiseen käytännön sovellusten kustannuksella.

Kuten luvussa 2.1 esitettiin, on taloustieteen käsitteistö vakiintunut teknisen velan tutkimuksen yleiskieleksi, jota lähes kaikki teknisen velan vertauskuvan esitteleen Cunninghamin (1992) jälkeiset tutkimukset ovat käyttäneet. Vaikka lähes kaikki teknisen velan tutkimukset käyttävät samaa käsitteistöä, määritellään ne monesti tutkimuskohtaisesti eri tavoin. Esimerkiksi Kruchten, Nord, Ozkaya ja Falessi (2013) sekä Li ym. (2015) kritisoivat teknisen velan tutkimusta siitä, että monesti lähes mikä tahansa ohjelmistotuotteen tai sen kehityksen haaste on liitetty kuuluvaksi teknisen velan piiriin. Li ym. (2015) mm. havaitsivat, että eräissä tutkimuksissa kaikki tuotevirheet sisällytettiin teknisen

velan alaisuuteen, vaikka tutkimuksessa käsitellyt tuotevirheet eivät täyttäneet tutkijoiden itsensä esittelemää teknisen velan määritelmää. Ongelman syyksi on ehdotettu, että koska tekninen velka esiteltiin alunperin vertauskuvana, on sitä jatkotutkimussa myös käsitelty sellaisena jokaisen tutkimuksen tehdessä omat tulkintansa sen konkreettisesta sisällöstä. Teknisen velan käsitelmällin (kuvio 2) julkaisemisen jälkeenkin mm. Rios ym. (2018) esittävät yhtenäisten määritelmien puutteen säilyneen ongelmana.

Teknisen velan empiiriset tutkimukset ovat harvinaisia (Nielsen ym., 2020). Harvat empiiriset tutkimukset painottuvat lisäksi vain muutamaaan aihealueeseen, etenkin käsittelemään koodi- ja rakennetyypin velan työkaluja (Rios ym., 2018). Näitä työkaluja käsittelevien tutkimuksien tulokset eivät myöskään usein ole vertailukelpoisia keskenään, sillä ne perustuvat kyseistä tutkimusta varten räätälöityyn tietojoukkoon, eivätkä tietojoukot myöskään ole monesti julkisesti saatavilla (Lenarduzzi, Saarimäki & Taibi, 2019).

Empiiristen tutkimusten puute on mahdollisesti syynä Nielsenin ym. (2020) havaitsemaan ilmiöön, jossa huomattava osa teknisen velan tutkimuksesta on olemassa olevien työskentelymenetelmien ja työkalujen soveltuvuuden arviointia erilaisissa teoreettisissa ympäristöissä. Nielsenin ym. (2020) mukaan olemassa olevia teknisen velan malleja ei hyödynnetä vaan tutkimukset kehittävät uusia malleja, jotka toimivat vain tutkimuksessa esitetyissä tilanteissa.

Siponen ja Baskerville (2018) kuvaavat ilmiön, jossa jokaisen tutkimuksen tulisi tuottaa uusi teoria, olleen pitkän aikaa vakiintuneena ja jopa toivottavana käytäntönä laajemmassa tietojärjestelmätieteen tutkimuksessa. Omien teorioiden lukumäärää on pidetty osana uuden tieteenalan vakiintumista osaksi laajempaa akateemista ympäristöä. Tämä pelkkä teorioiden määrän käyttäminen tieteenalan vakiintumisen mittarina teorioiden laadun sijasta on viime vuosina saanut kritiikkiä mm. Sipoelta ja Baskervilleltä (2018). Tämä tietojärjestelmätieteen alkuajoilta peräisin oleva ilmiö on kuitenkin tunnistettavissa myös teknisen velan tutkimuksessa.

Vähäinen teknisen velan tutkimus vaikuttaa tutkimusyhteisön lisäksi myös ohjelmistokehitysteollisuuden toimijoihin. Teollisuudessa tiedostetaan teknisen velan olevan kriittinen ongelma, mutta sitä ei puutteellisen tiedon takia osata hallinnoida tehokkaasti (Besker, Martini & Bosch, 2017). Keskimäärin 23 % ohjelmistokehittäjien työajasta kuluu teknisen velan hallintaan ja riittämätön velan takaisinmaksu pakottaa kehittäjiä ottamaan jatkuvasti lisää velkaa (Besker, Martini & Bosch, 2019a). Beskerin ym. (2019a) mukaan ohjelmiston vikojen korjaamiseen kulutettu työaika vaikuttaa negatiivisesti kehittäjien tuottavuuteen ja voi aiheuttaa merkittäviä rahallisia kustannuksia. Tomin ym. (2013) mukaan ICT-alan tutkimus- ja konsultointiyritys Gartner on arvionut koko ICT-alan teknisen velan takaisinmaksun rahallisten kustannusten olleen vuonna 2010 yli 500 miljardia dollaria ja ennakoii velan kaksinkertaistuvan vuoteen 2015 mennessä. Teknisen velan lisätutkimukselle on esitetty olevan tarvetta, sillä nykyisen tutkimuksen ei koeta vastaavan ohjelmistoteollisuuden tarpeita (Nielsen ym., 2020).

3.2 Aiemmat tutkimukset teknisen velan hallinnoinnista

Avgeriou ym. (2016) esittävät myös näkemyksensä teknisen velan hallinnoinnin hetkisestä nykytilasta ja tulevaisuudesta. Näkemyksen mukaan teknistä velkaa ei hallinnoida yhtä tehokkaasti kuin muita ohjelmistokehityksessä esiintyviä ongelmia ja teknisen velan hallinnoinnin keskeisinä haasteina ovat muun muassa

- Kommunikaatio ohjelmistokehittäjien ja organisaation hallinnon välillä
- Soveltuvien työkalujen ja työskentelymenetelmien tunnistaminen
- Soveltuvien työkalujen käyttäminen kaikilla organisaatiotasolla

Edellä lueteltuihin haasteisiin on tarjolla vain rajallinen määrä tutkimustietoa, sillä kyseiset haasteet eivät kuulu teknisen velan hallinnoinnin nykytutkimuksen painopistealueisiin ja kuten aiemmassa alaluvussa todettiin, painopistealueiden ulkopuolinen tutkimus on harvinaista.

Teknisen velan hallinnoinnin ongelmien osasyynä voi olla tutkimuksen painottuminen tiettyihin teknisen velan tyyppeihin (taulukko 1). Alves ym. (2016) toteavat ohjelmiston rakenteeseen, arkkitehtuuriin, dokumentaatioon, testaukseen ja koodiin liittyvien teknisen velan tyyppien vastaavan valtaosaa käsitellyistä tutkimuksista, kun esimerkiksi henkilöstön välisen vuorovaikutuksen ja käytettyjen ohjelmistokehityksen toimintatapojen vaikutusten tutkimus on lähes olematonta. Alvesin ym. (2016) lisäksi myös Rios ym. (2018) havaitsivat saman ilmiön käyttäen omaa teknisen velan tyyppien jaotteluaan ja toteavat lisäksi, että teknisen velan hallinnointiin käytettäviä menetelmiä ja työkaluja on tutkittu pääosin omina itsenäisinä kokonaisuuksinaan sekä kyseisten tutkimusten käsittelevän menetelmiä ja työkaluja vain tietyn teknisen velan hallinnoinnin toimenpiteen näkökulmasta. Kuten teknisen velan, niin myös teknisen velan hallinnoinnista tehdyt empiiriset tutkimukset ovat harvinaisia ja Bersker, Martini ja Bosch (2022) eivät esimerkiksi omassa tutkimuksessaan löytäneet yhtään teknisen velan ja henkilöstöhallinnon strategioita yhteen sovittavia empiirisiä tutkimuksia.

Teknisen velan hallinnoinnin tutkimuksen voidaan siis katsoa käsittelevän suurilta osin vain tiettyjä aihealueita. Hallinnoinnin kokonaiskuvaava selventäviä tutkimuksia on tehty, mutta niidenkin sisällössä on puutteita teknisen velan hallinnoinnin käytännön sovellusten osalta. Rios ym. (2018) ehdottavat teknisen velan hallinnointiin liittyviä konkreettisia menetelmiä ja työkaluja, mutta eivät ota kantaa näiden käytännön toteuttajiin. Fernández-Sánchez ym. (2017) puolestaan käsittelevät päätöksentekoprosessiin vaikuttavia muuttujia sekä kunkin päätöksen toteuttavaa vastuutahoa, mutta eivät suoraan anna konkreettisia esimerkkejä kunkin tahon noudattamista menetelmistä tai käyttämistä työkaluista.

Teknisen velan hallinnoinnissa pätee myös Siposen ja Baskervillen (2018) kuvaama laajempi tietojärjestelmätieteen ilmiö uusien teorioiden tuottamisesta. Esimerkkinä Riosin ym. (2018) tutkimuksessa käytetään mallia, jossa monista muista jaotteluista poiketen teknisen velan tunnistamista käsitellään omana teknisen velan hallinnoinnista irrallisena kokonaisuutenaan.

4 TEKNISEN VELAN VASTUUTAHOT JA VIESTINTÄ

Teknisen velan hallinnoinnin tulisi olla koko ohjelmistokehitysorganisaation vastuulla. Kehitysorganisaation eri vastuutahoilla on kuitenkin erilainen teknisen velan hallinnoinnin tuntemus ja etenkin vastuutahojen näkemykset teknisen velan vaikutuksista eroavat usein kriittisen paljon. Silloinkin, kun vastuutahot ovat yksimielisiä teknisen velan merkityksestä organisaation toiminnalle, tekevät vastuutahot päätöksiä ensisijaisesti omaa etuaan ajatellen. Syynä ongelmiin on usein teknisen velan vuorovaikutussuhteiden puutteellinen tuntemus tai tietämättömyys organisaatiolle kertyneestä teknisestä velasta, joiden ratkaisemiseen teknisen velan viestintä on keskeisin työkalu. Tässä luvussa käsitelläänkin teknisen velan viestintää ohjelmistokehitysorganisaatiossa. Kolmessa ensimmäisessä alaluvussa esitellään ohjelmistokehitysorganisaation teknisen velan vastuutahot, heidän teknisen velan näkemyksensä sekä teknisen velan päätöksentekoon vaikuttavat intressit. Viimeisessä alaluvussa esitellään tämän tutkielman kirjoittajan oma näkemys onnistuneen teknisen viestinnän lähtökohdista aiempaan tutkimukseen perustuen. Lähtökohdat ovat perustana tämän tutkielman seuraavassa ja myös viimeisessä sisältöluvussa esiteltävien teknisen velan viestinnän menetelmien ja työkalujen esittelylle.

4.1 Vastuutahojen esittely

Ohjelmistokehitystä harjoittavan organisaation henkilöstö voidaan jakaa erilaisiin vastuutahoihin riippumatta siitä, miten organisaatio on järjestäytynyt sisäisesti. Vastuutaho on yksi tai useampi henkilö tai ryhmä, jolla on intressi ja päätöksentekovoimaa johonkin ohjelmistokehitysprosessin osa-alueeseen. Vastuutahot voidaan määrittää mm. henkilöstön ammatillisen roolin perusteella, kuten Fernández-Sánchez ym. (2017) tekivät Clementsin ja Northropin (2002) vastuutahojen määritelmää käyttäen. Tämä tutkielma käyttää edellä mainittua

vastuutahojen jakoa. Ohjelmistokehitysorganisaatio on siinä jaettuna kolmeen vastuutahoon, joita ovat

- **Ohjelmistokehityksen vastuutaho.** Ohjelmistokehityksen vastuutaho toteuttaa ohjelmiston suunnitteluun ja rakentamiseen liittyvät toimenpiteet. Teknisen velan päätöksenteon perustekijöihin liittyvät toimenpiteet (kuvio 3), kuten velan kappaleiden tunnistaminen ja velan takaisinmaksu, toteutetaan usein ohjelmistokehityksen vastuutahon toimesta.
- **Ohjelmistokehityksen hallinto.** Ohjelmistokehityksen hallinnon vastaa ovat ohjelmistokehityksessä käytetyistä toimintamalleista ja kehitysprojektin seurannasta. Myös ohjelmiston laadun varmistus, vaatimustenmukaisuus ja muut ohjelmiston laadulliset kriteerit ovat ohjelmistokehityksen hallinnon vastuualueena.
- **Liiketoiminnan hallinto.** Liiketoiminnan hallinto vastaa ohjelmiston soveltuvuudesta liiketoiminnallisiin tavoitteisiin ja kehitystyön resurssoinnista tavoitteisiin pääsemiseksi asettaen mm. ohjelmiston toiminnallisuuksien vaatimukset ja kehityksen aikamääreet.

Myös Yli-Huumo ym. (2016) ovat itsenäisesti löytäneet muutoin samat vastuutahot tutkimuksessaan, mutta ohjelmistokehityksen vastuutaho eroteltiin ohjelmistokehitys- ja arkkitehtuuriryhmiin. Ohjelmistokehityksen vastuutahojen tarkempi erottelu ei kuitenkaan tämän tutkielman kontekstissa ole tarpeellista ja jatkossa käytetään Fernández-Sánchezin ym. (2017) määritelmää. Jatkossa myös termi *hallinto* käsittää yhteisesti sekä ohjelmistokehityksen hallinnon että liiketoiminnan hallinnon vastuutahot, ellei niitä tekstissä ole erikseen eroteltu.

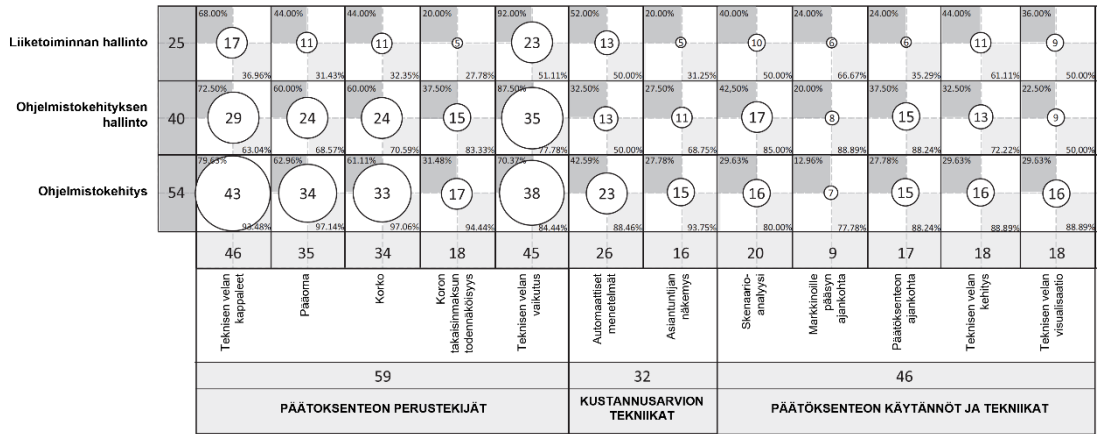
Edellä mainittujen kolmen vastuutahon lisäksi myös organisaation sidosryhmät, kuten asiakkaat ja ohjelmiston loppukäyttäjät, voidaan mieltää vastuutahoiksi. Liiketoiminnan tahon asettamat määritykset perustuvatkin usein asiakkaalta saatuihin vaatimuksiin (Codabux & Williams, 2013). Asiakasta koskevat teknisen velan hallinnoinnin toimenpiteet voidaan kuitenkin mieltää myös kuuluvaksi liiketoiminnan vastuutahon alaisuuteen. Loppukäyttäjältä saatu palaute ohjelmiston teknisen velan näkyvyydestä voi olla arvokasta tietoa, mutta tekninen velka ilmenee harvoin loppukäyttäjälle näkyvässä ohjelmiston käyttöliittymässä. Käyttöliittymän virheetön toimivuus on usein ohjelmistokehittäjille ensiarvoisen tärkeää ja teknistä velkaa pyritään siksi kerryttämään loppukäyttäjälle näkymättömillä suunnitteluratkaisuilla (Lim ym., 2012). Tässä tutkimuksessa keskitytäänkin vain ohjelmistokehitysorganisaation sisäisiin kolmeen vastuutahoon ja sidosryhmiä käsitellään tästä tarpeen mukaan esiteltyjen vastuutahojen näkökulmasta.

4.2 Vastuutahot ja päätöksenteko

Kukin vastuutaho tekee teknisen velan hallinnoinnin päätöksiään omien intressiensä mukaisesti, eli sen perusteella mitkä päätökset tuottavat eniten hyötyä kyseiselle vastuutaholle (Fernández-Sánchez ym., 2017). Esimerkiksi liiketoiminnan hallinto ei välttämättä koe tarvetta lisätä ohjelmistokehityksen resursseja teknisen velan hallintaan, ellei sen koeta tuottavan lisäarvoa liiketoiminnan tavoitteisiin. Tekninen velka vaikuttaa kuitenkin moniin organisaation toimintoihin vähintäänkin epäsuorasti ja näitä vuorovaikutussuhteita kuvattiin teknisen velan käsitelmissä (kuvio 2). Vastuutahojen itsekäiden päätöksiensä syynä onkin usein puutteellinen teknisen velan vuorovaikutussuhteiden tuntemus, kuten seuraavassa kappaleessa havainnoidaan.

Ernstin ym. (2015) mukaan teknisen velan hallinnoinnin toimenpiteitä ei osata toteuttaa ennakoivasti vaan tekniseen velkaan reagoidaan vasta, kun sen vaikutukset muuttuvat häiritseviksi. Syyksi tähän voidaan esittää, että liiketoiminnan hallinnolla ei ole tietoa teknisen velan vuorovaikutussuhteista liiketoimintaan ja Guon, Spínolan ja Seamanin (2016) mukaan hallinto tekeekin päätöksiä useammin oman kokemuksensa ja intuitionsa pohjalta kuin teknisen velan hallinnoinnin toimenpiteillä saadun tiedon perusteella. Ernstin ym. (2015) mukaan lähes puolet ohjelmistokehittäjistä ja ohjelmistokehityksen hallinnosta kokevat, ettei liiketoiminnan hallinto ole tietoinen teknisen velan hallitsemattoman kertymisen tuomista ongelmista ja velan hallinnoinnin hyödyistä. Puutteellinen informaatio ei ole vain hallinnon ongelma, vaan myös ohjelmistokehityksen tulee huomioida se omassa työskentelyssään. Ilman laajempaa organisaation liiketoiminnan tavoitteiden tuntemusta eivät kehittäjät pysty peilaamaan saamiaan vaatimuksia siihen, mitä hallinto niillä pyrkii tavoittelemaan. Kehittäjät joutuvatkin monesti luottamaan intuihionsa siitä, mitkä tekniset ratkaisut parhaiten edistävät organisaation etua (Brenner, 2019).

Fernández-Sánchez ym. (2017) jaottelevat aiemmin esitellyistä teknisen velan päätöksenteon elementeistä (kuvio 3) tehtyjä tutkimuksia vastuutahoittain sekä analysoivat kustakin elementistä tehtyjen tutkimusten lukumäärän kullekin vastuutaholle (kuvio 4). Tutkimusten lukumäärästä kullekin vastuutaholle voidaan tehdä päätelmiä, mitkä teknisen velan päätöksenteon elementit koetaan tärkeiksi milläkin vastuutaholla. Aliedustettujen elementtien perusteella voidaan myös selittää mahdollista teknisen velan tietämyksen puutetta kyseisen elementin osalta.



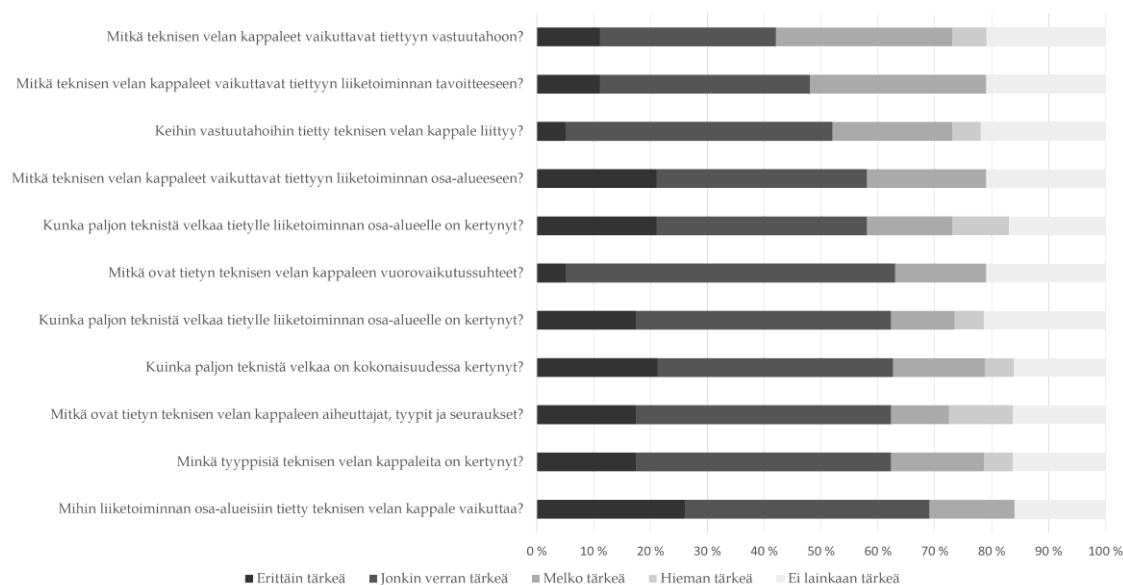
KUVIO 4 Teknisen velan päätöksenteon elementit vastuutahoittain (Fernández-Sánchezin ym., 2017 mukaan)

Kuvion 4 tutkimusten lukumäärän perusteella voidaan arvioida, mitkä päätöksenteon elementit kukin vastuutaho kokee tärkeiksi. Kuvion rivit kuvaavat vastuutahoa ja vastuutahon otsikon vieressä oleva luku kyseisestä tahoa käsittelevien tutkimusten kokonaismäärää. Kuvion sarakkeet vastaavat teknisen velan elementtiä ja sarakeotsikon yläpuolella oleva luku on kyseistä elementtiä käsittelevien tutkimusten kokonaismäärää. Rivien ja sarakkeiden risteykset ovat tutkimuksia, jotka käsittelevät kyseistä vastuutahoa sekä elementtiä. Risteyksessä esitetty luku on kyseistä vastuutahoa sekä elementtiä käsittelevien tutkimusten kokonaismäärää. Risteyksen vasemman yläneljänneksen prosentuaalinen arvo kuvaa, kuinka usea kyseisen vastuutahon tutkimus käsittelee kyseistä elementtiä. Vastavuoroisesti vasemman alaneljänneksen prosentuaalinen arvo kuvaa, kuinka usea kyseistä elementtiä käsittelevä tutkimus käsittelee myös saman rivin vastuutahoa. Esimerkiksi ohjelmistokehityksen hallinnon tahoa ja teknisen velan vaikutusten elementtiä käsitteleviä tutkimuksia on yhteensä 35 kappaletta. 87,50 % kaikista ohjelmistokehityksen hallintoa käsittelevistä tutkimuksista liittyy teknisen velan vaikutuksiin ja 77,78 % teknisen velan vaikutuksia käsittelevistä tutkimuksista liittyy ohjelmistokehityksen hallintoon.

Kuviosta 4 voidaan havainnoida teknisen velan hallinnoinnin tutkimuksen painottuvan ohjelmistokehittäjien vastuutahoon, vaikka monien tutkimusten, kuten Brenner (2019), mukaan teknisen velan hallinnoinnin tulisi olla kaikkien tahojen vastuulla. Lisäksi kuvio tukee havaintoa, jonka mukaan teknisen velan käytännön menetelmien ja työkalujen, eli kustannusarvion tekniikoiden, tutkimus on huomattavasti harvinaisempaa kuin teknisen velan päätöksenteon perustekijöiden tutkimus.

Osittain Fernández-Sánchezin ym. (2017) tutkimukseen pohjaten Pacheco, Marín-Raventós ja López (2018) selvittivät, mistä teknisen velan osa-alueista viestintä koetaan yhteisesti tärkeiksi eri vastuutahojen välillä (kuvio 5) osana laajempaa teknisen velan visualisointia käsittelevää tutkimusta. Pachecoon ym. (2018) tutkimus toteutettiin kyselytutkimuksena ja siihen vastasi it-alan ammattilaisia kaikilta tässäkin tutkimuksessa esitellyiltä vastuutahoilta. Tutkimuksessa koostettiin kysymyspatteristo, jonka yksittäisiin kysymyksiin kukin vastaaja

ilmoitti näkemyksensä siihen, kuinka tärkeänä huolenaiheena hän kokee kunkin kysymyksen kuvaavan teknisen velan aihealueen.



KUVIO 5 Teknisen velan vastuutahojen yhteiset intressit (Pachecon ym., 2018 mukaan)

Kuvion 5 vastauksien jakaumasta voidaan havainnoida, että tekninen velka koetaan yleisesti vastuutahosta riippumatta tärkeäksi asiaksi. Pacheco ym. (2018) huomioivat, että vastaajista yli puolet piti kaikkien paitsi kahden kysymyksen kuvaamia aiheita joko erittäin tärkeinä tai jonkin verran tärkeinä huolenaiheina. Tutkimus tukee tässä tutkielmassa aiemmin esitettyä havaintoa, jonka mukaan tekninen velan hallitsematon kertyminen tiedostetaan ohjelmistoteollisuudessa mittavaksi ongelmaksi, mutta jonka hallinnointi on vielä puutteellista etenkin sen kertymiseen ja vuorovaikutussuhteiden ymmärtämisen osalta. Tutkimus ei kuitenkaan huomioi, millä organisaation vastuutaholla on tarvittava tietotaito kyseisten visualisointien laatimiseen. Tutkimuksen visualisoinnit laadittiin tutkijoiden toimesta ja vain esiteltiin vastuutahoille palautteen saamista varten. Tutkimuksen tuloksia voidaan kuitenkin hyödyntää erilaisia visualisointimalleja luodessa, vaikkei se tarjoa varsinaisia menetelmiä tai työkaluja.

4.3 Viestintä vastuutahojen välillä

Teknisen velan hallinnointiin liittyvän päätöksenteon edellytyksenä on, että tieto tunnistetusta ja analysoidusta velasta saavuttaa oikean vastuutahon. Myös teknisen velan muutoksia tulee seurata ja tiedottaa vastuutahoille. Pelkkä tiedottaminen ei riitä, vaan tietoa vastaanottavan vastuutahon tulee myös ymmärtää kyseisen teknisen velan vaikutukset. Teknisen velan viestintä määritelläänkin vastuutahon suorittamaksi toimenpiteeksi, jolla tunnistettu tekninen velka saatetaan muiden vastuutahojen tietoon, jotta sitä voidaan hallinnoida. Apan ym. (2020)

mukaan teknisen velan viestintä onkin yksi tärkeimmistä teknisen velan hallinnon toimenpiteistä.

Informaatiota, jota teknisen velan viestinnällä pyritään välittämään, voidaan tarkastella teknisen velan päätöksenteon elementtien kautta (kuvio 3 ja kuvio 4). Fernández-Sánchez ym. (2017) nostavat viestinnän kannalta keskeisimmiksi päätöksenteon elementeiksi teknisen velan vaikutuksen ja teknisen velan visualisoinnin. Teknisen velan vaikutuksien ja visualisoinnin korostaminen ei silti tarkoita, että muut elementit voidaan jättää huomioimatta, vaan että vaikutusten ja visualisoinnin elementit kokoavat yhteen monien muiden elementtien kriittisimmät ohjelmistokehityksen ja hallinnon vastuutahojen välisen viestinnän tiedot. Kuten aiemmin tässä tutkielmassa esitettiin, teknisen velan laadullisen ja määrällisen arvioinnin tärkeimpänä tehtävä on tuottaa tieto teknisen velan pääomasta, korosta ja koron todennäköisyydestä. Näiden tietojen avulla pystytään jo itsessään välittämään tietoa teknisen velan vaikutuksista teknisen velan kokonaisarvion muodossa, mutta niistä voidaan myös lisätiedon avulla johtaa tehokkaampia teknisen velan viestinnän työkaluja, kuten kustannushyöty-analyysi.

Teknisen velan vaikutuksen elementti sisältääkin arvion teknisen velan vaikutuksista organisaation liiketoiminnallisiin tavoitteisiin. Fernández-Sánchez ym. (2017) esittävät arvion laatimiseksi nimenomaan edellä mainittua kustannushyöty-analyysiä. Yksinkertaistettuna kustannushyöty-analyysi on teknisen velan kokonaisarvio, johon on sisällytetty myös teknisen velan kappaleiden vuorovaikutussuhteet. Tällöin teknisen velan kappaleet voidaan listata tärkeysjärjestykseen niiden liiketoiminnan vaikutusten perusteella. Tämä mahdollistaa kappaleiden takaisinmaksun priorisoinnin ja aikataulutuksen. Kustannushyöty-analyysi esitellään tarkemmin seuraavassa luvussa. Teknisen velan vaikutusten ohella myös teknisen velan visualisointi koetaan tehokkaaksi keinoksi tuoda tekninen velka ja sen vuorovaikutussuhteet näkyville (Fernández-Sánchez ym., 2017). Myös teknisen velan visualisoinnin keinoja käsitellään tarkemmin seuraavassa luvussa.

Kuten myös edellisessä luvussa esitettiin, on tekniseen velkaan liittyvä viestintä vastuutahojen välillä havaittu keskeiseksi teknisen velan tehokasta hallinnointia estäväksi ongelmaksi (Avgeriou ym., 2016; Rios ym., 2018). Haasteita kommunikaatiossa esiintyy etenkin ohjelmistokehityksen vastuutahon ja liiketoiminnan hallinnon välillä, sillä heidän intressinsä ja näkemyksensä teknisestä velasta perustuvat eri lähtökohtiin (Klinger, Tarr, Wagstrom & Williams, 2011; Ampatzoglou ym., 2015). Kuten kuvio 4 osoittaa, keskittyy teknisen velan tutkimus ohjelmistokehittäjien osalta teknisen velan kappaleiden ja liiketoiminnan hallinnolla teknisen velan vaikutusten elementteihin. Elementtejä käsittelevien tutkimusten lukumäärän perusteella voidaan päätellä, että vastuutahot painottavat näitä elementtejä arvioidessaan teknisen velan hallinnointia koskevia päätöksiä myös työtehtävissään.

Ernstin ym. (2015) mukaan vain 10 % liiketoiminnan hallinnon henkilöstöstä osallistuu aktiivisesti teknisen velan hallinnointiin. Suurimman osan teknisestä velasta esitetäänkin syntyneen hallinnon ei-teknisen henkilöstön päätösten takia (Klinger ym., 2011; Ernst ym., 2015). Syyksi tähän on esitetty

kommunikaatiokuilua ohjelmistokehittäjien ja hallinnon välillä, kun vastuutahot yrittävät viestiä omista tarpeistaan toiselle osapuolelle vierain termein ja konseptein (Ampatzoglou ym., 2015). Ohjelmistokehittäjät eivät usein ole tietoisia organisaation liiketoiminnallista tavoitteista, jotka ovat hallinnon keskeisinä intresseinä, eivätkä siten osaa esittää teknisen velan kappaleiden vaikutuksia hallinnolle relevantissa muodossa. On myös esitetty, että ohjelmistokehittäjät käyttävät liian teknistä sanastoa kuvaillessaan teknisen velan kappaleita, jolloin hallinnolle ei synny käsitystä niiden vaikutuksista ohjelmistoon ja siten myös välillisesti liiketoiminnallisiin tavoitteisiin. Yleisempänä näkemyksenä kuitenkin on, että liiketoiminnan hallinto ei ymmärrä teknisen velan vaikutuksia, ellei se vaikuta suoraan liiketoimintaan (Lim ym., 2012). Tietoisuuden puutteen takia hallinto usein nähdäänkin kykenemättömänä arvioimaan teknisen velan hallinnon päätöksiä.

Ohjelmistokehittäjillä ei siis ole luottamusta hallinnon teknisestä velasta tekemiin päätöksiin. Hallinto ei myöskään aina tiedota muita vastuutahoja siitä, onko teknisen velan hallinnointi huomioitu päätöksenteossa johtaen epätietoisuuteen hallinnon esittämien vaatimusten tavoitteista (Brenner, 2019; Besker ym., 2022). Hallinnon teknisen velan heikko tuntemus voi johtaa väärinymmärryksiin, kun hallinto esittää teknisen velan hallinnon toimenpiteitä. Esimerkiksi Beskerin ym. (2022) mukaan organisaatioissa, joissa hallinto osallistuu aktiivisesti teknisen velan hallinnointiin, hallinto kokee kannustavansa kehittäjiä pienimuotoisen teknisen velan takaisinmaksuun, kun taas kehittäjät kokevat, että hallinto ei kannusta heitä laajamuotoisen teknisen velan takaisinmaksuun.

4.4 Teknisen velan viestinnän lähestymistavat

Teknisen velan viestintää voidaan toteuttaa monilla eri tavoilla. Monesti ehdotetaan keskusteluyhteyden avaamista ohjelmistokehityksen ja hallinnon vastuutahojen välille muusta ohjelmistokehityksestä erillisten vain teknistä velkaa käsittelevien tapaamisten muodossa. Tässä tutkielmassa ei kuitenkaan lähtökohtaisesti käsitellä eri viestintäkanavien, kuten kokousten tai sähköpostin, vaikutusta viestinnän hyödyllisyyteen vaan konkreettisten teknisen velan hallinnon menetelmien vaikutuksia teknisen velan viestintään.

Tässä tutkielmassa teknisen velan viestintää tarkastellaan käyttäen teoreettisena perustana Lin ym. (2015) löytämiä teknisen velan viestinnän lähestymistapoja. Lin ym. (2015) lähestymistavat ovat

- **Teknisen velan koostenäkymä.** Teknisen velan koostenäkymä on kooste havaituista teknisen velan kappaleista, tyypeistä ja määrästä.
- **Teknisen velan työlista.** Työlistaan kirjataan havaitut teknisen velan kappaleet. Teknisen velan työlistan sisältöä tulee käsitellä samanarvoisena kuin muita työlistoja, kuten bugilistaa tai suunnitelmaa uusien toiminnallisuuksien lisäämisestä.

- **Ohjelmiston rakenteen suhteiden visualisointi.** Ohjelmiston rakenteen visualisointi kuvaa ohjelmiston sisäisten osien, kuten erilaisten teknisten komponenttien ja pakettien väliset vuorovaikutussuhteet. Tämän visualisoinnin voidaan ymmärtää olevan kuvaus arkkitehtuurityypin velasta.
- **Koodianalyysin visualisointi.** Koodianalyysin visualisointi kuvaa ohjelmiston lähdekoodiin liittyviä ongelmia. Tämän voidaan ymmärtää olevan kuvaus rakenne- ja koodityypin velasta.
- **Teknisen velan kappaleiden listaaminen.** Löydetyt teknisen velan kappaleet kirjataan listaukseen. Listauksen perusteella voidaan toteuttaa teknisen velan kappaleiden seuranta, kuten luvussa 2.3 esitettiin.
- **Teknisen velan kappaleiden välisen vaikutuksen visualisointi.** Teknisen velan kappaleet voivat vaikuttaa myös toisiinsa ja tämä vuorovaikutus voidaan visualisoida. Esimerkkeinä tästä on teknisestä velasta oireilevan toiminnallisuuden päälle rakennettu lisätoiminnallisuus ja muut kasaantuvan koron kohteet.

Yllä esitetyillä lähestymistavoilla tunnistettu tekninen velka pystytään tekemään näkyväksi vastuutahoille Lin ym. (2015) mukaan. Yhteisenä piirteenä lähestymistavoissa on, että tunnistettu tekninen velka vaikutuksineen ja vuorovaikutussuhteineen listataan sekä visualisoidaan niin, että myös ei-tekninen henkilöstö, kuten monesti organisaation hallinto, pystyy ymmärtämään ne.

Tämän tutkielman selkeyttämiseksi on osa Lin ym. (2015) löytämistä lähestymistavoista yhdistetty. Kolme eniten tutkittua teknisen velan tyyppiä ovat arkkitehtuurivelka, rakennevelka ja koodivelka (Rios ym. 2018), eli luvussa 2.2 määritelty ohjelmiston toiminnallisuuksien velka. Näiden teknisen velan tyyppien voidaan ymmärtää sisältävän pääosan ohjelmiston sisältämästä automaattisilla työkaluilla havaittavasta teknisestä velasta ja monet automaattisia työkaluja käsittelevät tutkimukset onkin laajudeltaan rajattu käsittelemään nimenomaan näitä kolmea teknisen velan tyyppiä (Avgeriou ym., 2020). Ohjelmiston rakenteen suhteiden visualisointi ja koodianalyysin visualisointi pyrkivät visualisoimaan juurikin näitä kolmea teknisen velan tyyppiä, joten kyseiset visualisoinnit on yhdistetty toiminnallisuuksien velan visualisoinniksi.

Teknisen velan kappaleiden listaamisella ja kappaleiden välisen vaikutuksen visualisoinnilla koostetaan teknisen velan kappaleet ja niiden vuorovaikutussuhteet sekä tehdään ne näkyviksi eri vastuutahoille. Lin ym. (2015) lähestymistavat eivät kuitenkaan huomioi ohjelmistosta riippumattomia teknisen velan piirteitä vaan kappaleet vuorovaikuttavat Lin ym. (2015) lähestymistavoissa vain toisiinsa. Teknisen velan käsittemallista (kuvio 2) voidaan kuitenkin havaita, että teknisen velan kappaleet aiheuttavat suoria seurauksia ohjelmistokehitykseen sekä vaikuttavat välillisesti myös organisaation liiketoiminnallisiin tavoitteisiin. Näiden vuorovaikutussuhteiden

havainnollistaminen on jo aiemmin tässä tutkimuksessa todettu teknisen velan viestinnässä mittavaksi haasteeksi. Tämän johdosta on teknisen kappaleiden välisen vaikutuksen visualisointi sisällytetty teknisen velan kappaleiden listaukseen ja listaus sisältää myös muut kuin teknisen velan kappaleiden väliset vuorovaikutussuhteet.

Viimeisenä muutoksena lähestymistapoihin on Lin ym. (2015) suppeahko teknisen velan koostenäkymän määritelmä tässä tutkielmassa korvattu Shullin, Falessin, Seamanin, Diepin ja Laymanin (2013) koostenäkymän vaatimuksilla, jotka ovat

- **Automaattinen tiedonkeruu.** Tiedonkeruun tulee perustua olemassa oleviin tietolähteisiin, kuten edellä mainittuun Jira-järjestelmään, joista tietoa teknisestä velasta voidaan kerätä automaattisesti. Kerätystä tiedosta, kuten teknisen velan hallinnointiin käytetystä työajasta tai teknisen velan kappaleiden lukumäärästä, tulee pystyä koostamaan kutakin vastuutahoa hyödyttävää informaatiota. Shull ym. (2013) havainnoivat tämän tiedonkeruuprosessin manuaalisen toteuttamisen olevan harvoin hyödyllistä ajankäyttöä ja lisäksi Ernst ym. (2015) toteavat automaattisten työkalujen datan manuaalisen analysoinnin ylikuormittavan käyttäjän tarkkaavaisuutta tehden datasta käyttökelvotonta.
- **Helppo mukautettavuus.** Koska dataa kerätään useista tietolähteistä, tulee uusien lähteiden lisäämisen sekä vanhojen poistamisen olla helposti toteutettavissa.
- **Tiedon luotettavuus.** Kerätylle datalle tulee toteuttaa automaattisia oikeellisuustarkastuksia.
- **“Details-on-demand”.** Koostenäkymästä tulee tarjota teknisen velan yleiskatsauksen lisäksi myös pääsy tarkentaville tasoille. Näiden tasojen tulee sisältää kutakin vastuutahoa hyödyttävää tarkempaa informaatiota, kuten yksittäisten teknisen velan kappaleiden kuvaukset niiden alkuperäisen syötön järjestelmissä.

Shullin ym. (2013) vaatimuksia voidaan pitää totuudenmukaisempina toimivan teknisen velan koostenäkymän ehtoina kuin Lin ym. (2015) määritelmää. Tämän tutkielman teknisen velan viestinnän lähtökohtina käytetään siis seuraavia määritelmiä

- **Teknisen velan kappaleiden listaus.** Lista, joka sisältää kaikki havaitut teknisen velan kappaleet ja niiden vuorovaikutussuhteet.
- **Teknisen velan koostenäkymä.** Sama kuin Shullin ym. (2013) vaatimukset.
- **Teknisen velan työlista.** Sama kuin Lin ym. (2015) määritelmä.
- **Toiminnallisuuksien velan visualisointi.** Visualisointi, joka käsittää teknisen velan arkkitehtuuri-, rakenne- ja koodivelan sekä esittää niiden vuorovaikutussuhteet.

- **Teknisen velan vaikutusten kuvaus.** Teknisen velan kappaleiden vuorovaikutussuhteiden visualisointi.

Edellä esityt lähtökohdat luovat tämän tutkielman kirjoittajan mielestä hyvät edellytykset toteuttaa onnistunutta teknisen velan viestintää ja niitä voidaan soveltaa Lin ym. (2015) alkuperäistä laajemmassa teknisen velan ympäristössä huomioiden oleellimmat teknisen velan vuorovaikutussuhteet sekä teknisen velan hallinnoinnin päätöksenteon elementit.

5 TEKNISEN VELAN VIESTINNÄN MENETELMÄT JA TYÖKALUT

Teknisen velan viestintään on tarjolla monia erilaisia menetelmiä ja työkaluja. Tässä luvussa esitellään viisi teknisen velan viestinnän menetelmää ja työkalua, joita tarkastellaan aiemmassa luvussa määriteltyjen teknisen velan viestinnän lähestymistapojen kautta. Menetelmät ja työkalut on valittu kuvaamaan teknisen velan viestinnän monia eri toteuttamistapoja. Menetelmien ja työkalujen tulee myös olla soveltuvia sekä saatavilla ohjelmistoteollisuuden käyttöön. Ensimmäiset kaksi lukua esittelevät valitut teknisen velan viestinnän menetelmät ja koostaa niistä matriisin, jonka perusteella havainnoidaan menetelmien eroavaisuuksia ja yhtäläisyyksiä. Jälkimmäiset kaksi lukua noudattavat samaa rakennetta, mutta käsittelevät teknisen velan viestinnän työkaluja.

5.1 Teknisen velan viestinnän menetelmien esittely

Teknisen velan viestinnän menetelmät on määritelty käsittämään tässä luvussa teknisen velan viestinnän toimenpiteet, joiden toteuttamiseen ei vaadita ohjelmistotyökaluja. Tekniset työkalut voivat kuitenkin avustaa menetelmien toteuttamista. Esiteltävät menetelmät ovat

- Teknisen velan koostenäkymän laatiminen
- Teknisen velan kustannushyöty-analyysi
- Teknisen velan hallinnointi liiketoiminnan prosesseilla
- Hyvät hallintokäytänteet
- Asiantuntija-arvio

Esiteltävät menetelmät on valittu niiden vastuutahojen välisen teknisen velan viestinnän soveltuvuuden mukaan. Valinnoissa on pyritty valitsemaan mahdollisimman erilaisia menetelmiä eikä esimerkiksi useita

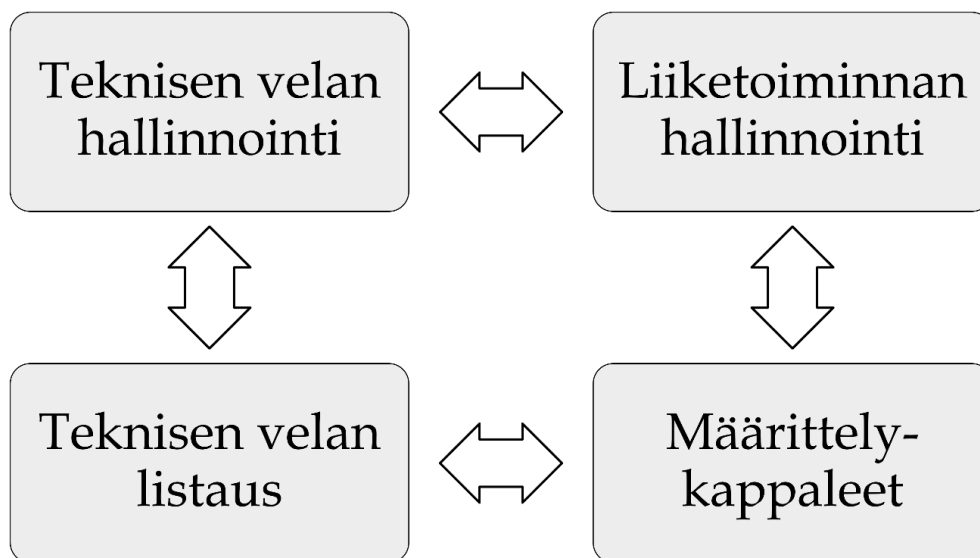
kustannusarvioiden tekniikoita, kuten esiteltävän kustannushyöty-analyysin vaihtoehtoja analyttistä hierarkiaprosessia (lyh. AHP) tai teknisen velan salkunhallintamenetelmää. Poikkeuksen tästä tekee hyvät hallintokäytänteet ja asiantuntija-arvio, jotka molemmat voidaan mieltää keskittyvän ihmisten väliseen vuorovaikutukseen selkeiden ohjesääntöjen noudattamisen sijasta. Näitä menetelmiä toteuttavat vastuutahojen teknisen velan hallinnoinnin näkökulmat ovat kuitenkin niin erilaisia, että molemmat on päätetty sisällyttää esiteltäviin menetelmiin. Asiantuntija-arvio on menetelmien lisäksi sisällytetty myös esiteltäviin teknisen velan viestinnän työkaluihin, mutta kuten menetelmän ja työkalujen määritelmät rajoittavat, toteuttaa asiantuntija viestintää eri roolissa.

Shullin ym. (2013) ehdot täydellisesti täyttäviä teknisen velan koostenäkymiä ei tämän tutkielman yhteydessä löytynyt yhtäkään kappaletta. Koostenäkymä on kuitenkin sekä Lin ym. (2015) että Shullin ym. (2013) mukaan keskeinen teknisen velan viestinnän osa, joten koostenäkymää käsitellään sen suunnittelun menetelmänä, sillä näkymien suunnittelun pystyy toteuttamaan myös ilman ohjelmistotyökaluja. Täten, kun koostenäkymän toiminnalliset vaatimukset on toteutettu, voidaan keskittyä teknisestä velasta esitettävään informaatioon ja sen esitystapoihin. Kuten aiemmin tässä tutkielmassa on kerrottu, on yhtenä teknisen velan viestinnän ongelmana vastuutahojen intressien sekä intresseille relevanttien teknisen velan vuorovaikutussuhteiden tuntemus. Kuviossa 5 esitelty kysymyspatteristo on kuitenkin sen esitelleen Pacheco ym. (2018) mukaan kattava kuvaus ohjelmistokehityksen ja hallinnon yhteisistä intresseistä, joihin vastaamisen tulisi olla teknisestä velasta laadittujen visualisointien keskeisenä sisältönä vastuutahojen välisessä viestinnässä. Pacheco ym. (2018) esittelevät myös erilaisia visualisoinnin esitystapoja ja niiden soveltumista kuvion 5 kysymyksiin vastattaessa. Esitystapoja ei tutkimuksessa eritelty esitettyjen kysymysten perusteella, vaan tulosten perusteella todettiin yleisesti yli 80 % vastaajista pitävän diagrammeja, kaavioita ja graafeja hyödyllisinä esitystapoina sekä erityisesti puu-, kalanruoto-, palkki- ja Sankey-kaavioiden olevan toivottuja esitysmuotoja. Vaikka tässä kappaleessa esitetyt tutkimukset korostavat teknisen velan vuorovaikutussuhteiden havainnollistamisen tärkeyttä, eivät ne tarjoa menetelmiä tai työkaluja niiden selvittämiseen.

Kuten monet muut tutkimukset (mm. Ampatzoglou ym., 2015; Fernández-Sánchez ym., 2017), esittävät myös de Almeida, Kulesza, Treude ja Lima (2018), että teknisen velan hallinnoinnin tulisi perustua velan liiketoiminnallisiin vaikutuksiin. Teknisen velan liiketoiminnan tavoitteisiin liittyvien vuorovaikutussuhteiden selvittäminen on kuitenkin todettu haastavaksi, sillä sen toteuttaminen luotettavasti tähänhetkisillä automaattisilla työkaluilla ei ole mahdollista (Rios ym., 2018). Tämän takia on arviointia pakko toteuttaa vastuutahojen subjektiivisiin näkemyksiin perustuen ja Fernández-Sánchez ym. (2017) ehdottavat ratkaisuksi vaikutusten mittaamista teknisen velan perusmallin (kuvio 1) lopputulosten kautta ja vaikutusten visualisointia. Ohjelmiston ulkopuolisten tekijöiden arvioinnin haastavuudesta huolimatta monet teknistä velkaa käsittelevät tutkimukset ehdottavat teknisen velan kustannushyöty-analyysiä viestinnän välineeksi. Kustannushyöty-analyysi sisältää teknisen velan kokonaisarvion eli

vähintäänkin arviot kaikkien havaittujen teknisen velan kappaleiden pääomasta ja korosta, ja johon on huomioitu myös ohjelmistokehitysprosessin ei-tekniset rajoitteet, kuten aikamäärät ja budjetti. Kustannushyötyanalyysi koostaa teknisen velan määrän, vaikutukset ja takaisinmaksun vaatimukset kaikille vastuutahoille ymmärrettävään muotoon; kokonaisarvio paljastaa teknisen velan määrän ja kriittiset teknisen velan kappaleet, kun taas ei-tekniset rajoitukset mahdollistavat kustannusten määrittämisen kappaleelle ja sen takaisinmaksulle. Kustannuksia ja vuorovaikutussuhteita voidaan esittää edellämäinituilla visualisoinnin keinoilla.

Vaikka tekninen velka on tunnistettu ja havainnollistettu onnistuneesti vastuutahoille, ei tämä vielä itsessään ole tae onnistuneelle viestinnälle. Aiemmin tässä mainittujen tutkimuksien lisäksi myös de Almeida ym. (2018) huomioivat, että monesti eri vastuutahojen intressit vaikuttavat huomattavasti päätöksentekoon, vaikka teknisen velan määrä ja vaikutukset olisivatkin selvillä. De Almeida ym. (2018) havaitsivat, että vaikka organisaation teknistä velkaa seurattaisiin ja havainnollistettaisiin aiemmin esitellyillä menetelmillä, kuten tutkimuksen yhden case-ryhmän käyttämällä Trello-tehtävienhallintaohjelmistolla, voidaan teknisen velan hallinnoinnissa toteuttaa väriä toimenpiteitä vastuutahojen intressien takia. Eräs tutkimukseen osallistunut liiketoiminnan vastuutahon edustaja totesi, että ”on yleistä tarkastella tietojärjestelmiä vain oman liiketoiminnan osaluensa kannalta...” jatkaen tämän johtavan ristiriitoihin muiden liiketoiminnan osien kanssa. Vastuutahojen edustaja toteaa myös liiketoiminnan kokonaiskuvan selvittämisen vaikeaksi, sillä nykyisin it-alan työntekijät ovat äärimmäisen erikoistuneita omiin toimenkuviinsa. De Almeida ym. (2018) ehdottavat ratkaisuksi teknisen velan hallinnointia liiketoiminnan prosesseilla (kuvio 6).



KUVIO 6 Teknisen velan hallinnointi liiketoiminnan prosesseilla
(de Almeidan ym., 2018 mukaan)

Kuvion 6 mukaan teknisen velan kappaleita seurataan teknisen velan listauksella. Teknisen velan kappaleelle tulee pystyä osoittamaan määrittelykappale, joka voi

olla teknisen velan tyyppi, tietojärjestelmä tai prosessi, johon teknisen velan kappale vaikuttaa millään tavoin. Jokaiselle määrittelykappaleelle tulee liiketoiminnan hallinnollinen prosessi, jossa huomioidaan määrittelykappaleen olemassaolo. Liiketoiminnan prosessin tulee pystyä tunnistamaan määrittelykappaleen perusteella, mitkä päätöksenteon tekijät, kuten rahalliset kustannukset tai aikataulullinen kiireisyys, ovat päätöksenteon keskeisimmät tekijät. Näitä päätöksenteon tekijöitä ei tule sekoittaa teknisen velan päätöksenteon elementteihin (kuvio 3). Liiketoiminnan prosessi määrää toteutettavat teknisen velan hallinnoinnin toimenpiteet. Menetelmän avulla voidaan havaitulle tekniselle velalle toteuttaa vaaditut toimenpiteet useiden vastuutahojen intressit huomioiden.

Teknisen velan viestinnän tulee myös huomioida muita kuin suoraan tekniseen velkaan tai sen liiketoiminnan seurauksiin vaikuttavia tekijöitä eikä hyvien teknisestä velasta riippumattomien organisaation hallinnointimenetelmien merkitystä tule jättää huomioimatta (Brenner, 2019). Brenner (2019) muun muassa huomioi, että mikäli teknisen velan hallinnoinnin prosesseja kehitetään vain ohjelmistokehittäjien työnkuviin liittyen, asetetaan heille jatkuvasti teknisen velan ottamiseen kannustavia tiukempia aikamääreitä eikä teknisen velan hallinnoinnin hyötyjä saavuteta. Hallinnon tehtäväksi esitetään tarvittavien resurssien tarjoamista kehittäjille sekä säännöllistä teknisen velan takaisinmaksun vaatimista. Pitkän aikavälin liiketoimintastrategian tiedottaminen kehittäjille koetaan tarpeelliseksi, sillä kun kehittäjillä on selkeä käsitys yrityksen tulevaisuuden suunnitelmista, osaavat he rajoittaa teknisen velan kappaleiden luomista ohjelmiston liiketoiminnalle kriittisillä osa-alueilla. Muita epäsuorasti teknisen velan kertymiseen johtavia syitä on esimerkiksi henkilöstömuutoksista johtuva osaamisen poistuminen organisaatiosta, johon pystytään vaikuttamaan organisaation henkilöstöpolitiikalla (Brenner, 2019). Lisäksi Besker ym. (2022) esittävät erilaisia palkitsemis- ja rankaisemismenetelmiä, joilla hallinto pyrkii ohjaamaan ohjelmistokehityksen vastuutahoa tehokkaampaan teknisen velan hallinointiin.

Asiantuntija-arvio voidaan tämän tutkimuksen kontekstissa ymmärtää joko menetelmäksi tai työkaluksi, riippuen mihin toimenpiteisiin asiantuntijaa käytetään. Kuten aiemmin tässä tutkimuksessa on todettu, varsinkaan teknisen velan liiketoiminnallisten vaikutusten arviointiin on tarjolla hyvin rajattu määrä työkaluja tai menetelmiä. Silloin kehitettävän tietojärjestelmän ja/tai kehitettävän organisaation liiketoiminnan tuntevan asiantuntijan näkemys voi olla tarkempi ja helpommin ymmärrettävä kuvaus teknisestä velasta kuin mitä muilla keinoin saavutettaisiin. Asiantuntija kuvaakin tässä kontekstissa organisaation vastuutahoa, jolla on paras kyky arvioida käsiteltävää teknisen velan osa-alueita. Asiantuntijoita voidaan käyttää myös viestintään sidosryhmien välillä, mikä on tärkeä teknisen velan viestinnän osa-alue. Vaikkei sidosryhmiin viestintä olekaan organisaation sisäistä viestintää, vaikuttaa se olennaisesti eri vastuutahojen toisilleen esittämiin vaatimuksiin. Esimerkiksi teknistä velkaa aiheuttavat vaatimukset johtuvat monissa tapauksissa asiakkaan tarpeista, jotka hallinto välittää ohjelmistokehityksen vastuutaholle. Hallinnolla tulee olla kokonaiskuva ohjelmiston teknisestä velasta, jotta asiakkaan tarpeet osataan

resurssoida ohjelmistokehitysprosessissa oikein. Asiakas ei yleensä ole tietoinen ohjelmiston teknisen velan määrästä eikä osaa huomioida sitä vaatimuksissaan. Asiakas ei myöskään usein ole tietoinen tarpeistaan, jolloin teknistä velkaa saatetaan kehittää vahingossa ellei asiakkaan tarpeita ole kartoitettu tarkasti. Lim ym. (2012) painottavat, että kun teknisen velan todellinen määrä pystytään tiedottamaan asiakkaalle, asiakas on usein joustavampi vaatimuksiensa suhteen, kuten aikatauluissa ja uusien toiminnallisuuden kehittämisessä, mahdollistaen näin myös tehokkaamman teknisen velan hallinnoinnin, kun eri vastuutahojen työkuormaa voidaan resurssoida tarkemmin. Teknisen velan viestinnän tuleekin olla kaikkien vastuutahojen yhteinen ja keskeinen toimenpide osana teknisen velan hallinnointia, jotta myös ulkopuolisten tekijöiden vaikutus pystytään minimoimaan.

5.2 Johtopäätökset teknisen velan viestinnän menetelmistä

Tässä tutkielmassa esiteltyt teknisen velan viestinnän menetelmät ovat koostenäkymän suunnittelu, kustannushyöty-analyysi, teknisen velan hallinnointi liiketoiminnan prosesseilla, hyvät hallintokäytänteet ja asiantuntija-arvio. Eritelty menetelmät on listattu alla olevaan teknisen velan viestinnän menetelmien koostetaulukkoon (taulukko 2).

TAULUKKO 2 Teknisen velan viestinnän menetelmien kooste

	Teknisen velan kappaleiden listaus	Teknisen velan koostenäkymä	Teknisen velan työlista	Ohjelmiston sisäisen teknisen velan kuvaus	Teknisen velan vaikutusten kuvaus
Koostenäkymän suunnittelu	X	X	X	X	X
Kustannushyöty-analyysi			X		X
Teknisen velan hallinnointi liiketoiminnan prosesseilla	X		X		X
Hyvät hallintokäytänteet				X	X
Asiantuntija-arvio				X	X

Taulukossa 2 on esitetty tässä tutkimuksessa esiteltyjä teknisen velan viestinnän menetelmiä. Kunkin menetelmän osalta on tehty merkintä "X" kuvaamaan menetelmän soveltuvuutta aiemmin luvussa 4 esiteltyyn teknisen velan viestinnän lähestymistapaan.

- **Koostenäkymän suunnittelu.** Shullin ym. (2013) vaatimukset täyttävää koostenäkymää voidaan käyttää kaikissa teknisen velan viestinnän lähestymistavoissa. Koostenäkymän laatimisen ongelmana onkin enemmän teknisen toteutuksen mahdollisuus kuin vaatimusten puutteellisuus.
- **Kustannushyöty-analyysi.** Kustannushyöty-analyysi on monissa tutkimuksissa (mm. Li ym., 2015; Ampatzoglou ym., 2015; Fernández-Sánchez ym., 2017) mainittu yhdeksi keskeisimmäksi teknisen velan vaikutusten arvioinnin työkaluksi ja Fernández-Sánchez ym. (2017) nostavat sen erikseen esiin keskeisimmäksi teknisen velan vaikutusten kuvaajaksi. Näin ollen, oikein toteutettuna, kustannushyöty-analyysin voidaan nähdä kuvaavan teknisen velan vaikutukset etenkin liiketoiminnan tavoitteisiin erittäin tehokkaasti. Analyysi kuitenkin tekee tämän yhden osa-alueen erittäin hyvin, mutta on työlistausta lukuun ottamatta soveltumaton muihin lähtökohtiin. Analyysin soveltuminen työlistaukseksi riippuu pitkälti ohjelmistokehitysorganisaation käytännöistä, kuten kuinka usein ja kenen toimesta sekä työlistoja että kustannushyöty-analyysia päivitetään.
- **Teknisen velan hallinnointi liiketoiminnan prosesseilla** sisältää jo vaatimuksiensa puolesta teknisen velan listauksen, työlistan ja vaikutusten kuvauksen.
- **Hyvät hallintokäytänteet ja Asiantuntija-arvio** käsittelevät molemmat ihmisten välistä vuorovaikutusta, mutta eri näkökulmista. Hyvät hallintokäytänteet pyrkivät tekemään teknistä velkaa näkyväksi organisaation käytäntein, kun taas asiantuntija-arvio on pienemmän mittakaavan tapauskohtaista teknisen velan viestintää. Kummatkin soveltuvat niin toiminnallisuuksien velan kuin vuorovaikutussuhteiden kuvaamiseen viestintää toteuttavien tahojen pätevyys huomioiden.

5.3 Teknisen velan viestinnän työkalujen esittely

Teknisen velan viestinnän työkalut on määritelty käsittämään tässä luvussa teknisen velan viestinnän toimenpiteet, joiden toteuttamisen vaatimuksena on jonkin ohjelmistotyökalun käyttäminen. Työkalun tulee tuottaa usealla vastuutaholla hyödynnettävää teknisen velan tietoa joko täysin automaattisesti tai asiantuntijan kohtuullisella avustuksella. Esiteltävät työkalut ovat

- Asiantuntija-arvio
- SQALE-menetelmä
- Tehtävienhallintaohjelmisto Jira
- Themis
- DebtFlag

Esiteltävät työkalut on valittu niiden vastuutahojen välisen teknisen velan viestinnän soveltuvuuden mukaan. Työkalujen valinnassa on pyritty

valitsemaan mahdollisimman erilaisia työkaluja eikä esimerkiksi useita SQALE-menetelmää soveltavia lähdekoodin analysoinnin työkaluja, vaikka lähdekoodin analysoinnin työkaluista tehtyjä tutkimuksia on muihin teknisen velan hallinnoinnin osa-alueisiin verrattuna moninkertainen määrä (mm. Li ym., 2015; Alves ym., 2016; Rios ym., 2018). Asiantuntija-arvio on työkalujen lisäksi sisällytetty myös esiteltäviin teknisen velan viestinnän menetelmiin, mutta kuten menetelmän ja työkalujen määritelmät myös rajoittavat, toteuttaa asiantuntija viestintää eri roolissa.

Pelkkä teknisen velan kappaleen tunnistaminen ja mittaaminen ei tuota riittävästi tietoa viestinnän mahdollistamiseksi (Yli-Huumo ym., 2016). Esimerkiksi Debt Symptoms Index -viitekehysellä voidaan yksiselitteisesti mitata ja analysoida ohjelmiston rakennevelka (Marinescu, 2012), mutta viitekehysten tulosten tulkinta edellyttää rakennevirheitä kuvaavien mallien ja mittareiden syvää tuntemusta, eikä siten sovellu vastuutahojen väliseen viestintään. Ernstin ym. (2015) mukaan työkalujen heikkoutena onkin, että niiden tuottamien tulosten oikeellisuuden varmistaminen vaatii asiantuntijoiden ammattitaitoa ja työkalut tulee räätälöidä erikseen jokaista tutkittavaa ohjelmistoa varten. Yksinkertaisin menetelmä soveltaa tällaista tietoa on pyytää kyseisen teknisen velan tyypin asiantuntijaa, joka on usein ohjelmistokehityksen vastuutaho, tulkaamaan työkalun tulokset päätöksiä tekevän vastuutahon ymmärrettävään muotoon joko sanallisesti, visuaalisin avuin tai muilla soveltuvilla keinoilla. Asiantuntija voi myös antaa näkemyksensä velan vaatimista toimenpiteistä (Fernández-Sánchez ym., 2017). Viestintään soveltuvien ohjelmistotyökalujen hyödyntäminen on kuitenkin usein asiantuntijoiden käyttöä tehokkaampi ja tarkempi viestinnän menetelmä.

Kuten edellä mainittu, monet teknisen velan mittaamiseen käytetyt automaattiset työkalut eivät suoraan tuota vastuutahojen väliseen viestintään soveltuvaa informaatiota. Tähän ongelmaan on ehdotettu ratkaisuksi SQALE-menetelmää (lyh. Software Quality Assessment based on Life Cycle Expectations), jonka avulla automaattisten työkalujen tuottama data voidaan muuntaa viestintään hyödynnettäväksi informaatioksi, kuten teknisen velan takaisinmaksuun vaadituiksi rahallisiksi kustannuksiksi tai työtunneiksi. SQALE-menetelmä on saatavilla monissa staattisesti koodia analysoivissa työkaluissa, kuten yhden eniten käytetyimmän analysointiohjelmisto SonarCuben lisäosana (BenIdris, 2020). Kustannusarvioiden laatimisen lisäksi Letouzey (2012) ehdottaa seuraavia toimenpiteitä SQALE-menetelmän käyttöön, jotka osin kattavat myös Lin ym. (2015) esittämät lähestymistavat:

- Organisaatiolla tulee olla selkeät säännöt ja prosessit teknisen velan hallintoihin
- Pilotointiprojekteja käytetään varmistamaan SQALE-arvion totuudenmukaisuus
- SQALE-tuloksista laaditaan koostenäkymä, josta kaikki vastuutahot voivat tarkistaa teknisen velan määrän

- Laadun arviointia ja seuranta automatisoidaan mahdollisimman paljon ja muutosten tulee päivittyä koostenäkymään
- Organisaation henkilöstö on perehdytetty teknisen velan hallinnasta kaikilla organisaatiotasolla ja lisäkoulusta järjestetään tarvittaessa

Ennen SQALE-menetelmää ei ohjelmiston lähdekoodin laadun arviointiin ollut tarjolla malleja, sillä mm. ohjelmistojen laadun arvioinnin ISO-standardit havaittiin puuttellisiksi eivätkä ne tarjonneet luotettavia käytännön menetelmiä (Letouzey, 2012), kuten monissa muissakin tässä tutkimuksessa käsitellyissä teknisen velan aihepiireissä on havaittu. SQALE-menetelmän heikkoutena voidaan nähdä, että se tarjoaa työkaluja vain lähdekoodin analysointiin ja sen perusteella tehtäviin toimenpiteisiin. Esimerkiksi liiketoiminnan tarpeita ja muita ohjelmistosta riippumattomia tekijöitä ei huomioida SQALE-menetelmässä.

Yli-Huumo ym. (2016) havaitsivat, että monet organisaatiot käyttivät teknisen velan hallinnoinnin työkalunaan Jira-tehtävienhallintaohjelmistoa. Kehittäjät dokumentoivat teknistä velkaa Jiraan, jonka perusteella ohjelmistokehityksen ja liiketoiminnan hallinnot pystyvät arvioimaan teknisen velan määrää. Jiran käytön tehokkuus havaittiin kuitenkin vaihtelevaksi. Joissakin yrityksissä ohjelmistokehityksen hallinto teki Jira-kirjaukset, jolloin teknisen velan tyyppi ja määrä arvioitiin väärin. Pelkkä Jirasta saatu tieto ei myöskään aina ollut riittävää liiketoiminnan päätöksentekoa varten, sillä kehittäjät kirjasivat sinne ennen kaikkea teknisiä ongelmia ja niiden vaatiman työpanoksen, mutta vaikuttavuuden arviointia ei huomioitu vaikeuttaen teknisen velan takaisinmaksun priorisointia. Toisessa tutkimuksessa Besker, Martini ja Bosch (2019b) havainnoivat, ettei ketteriä ohjelmistokehitysmenetelmiä käyttävissä organisaatioissa teknisen velan kappaleita usein kirjata sprinttien omiin työlisteroihin vaan ohjelmistokehityksen vastuutaho ylläpitää erillistä teknisen velan varjolistausta, joka on pelkästään kyseisen kehitystiimin saatavilla. Beskerin ym. (2019b) mukaan nämä varjolistat tulisi joko liittää osaksi varsinaisia työlisteroita tai vähintäänkin tehdä yhtä näkyviksi sekä samanarvoisiksi teknisen velan viestinnän osiksi kuin sprinttien varsinaiset työlisterot.

Yksi harvoja erityisesti ohjelmistokehittäjien ja hallinnon väliseen teknisen velan viestintään erikoistuneita ohjelmistoja on Foucaultin, Blancin, Storeyn, Fallerin ja Teytonin (2018) kehittämä Themis-ohjelmisto (nyk. Promyze). Themis on tunnettu vuodesta 2020 alkaen nimellä Promyze ja sen ominaisuudet sekä sen tukemat ohjelmistoalustat ovat laajentuneet Foucaultin ym. (2018) jälkeen. Tässä tutkimuksessa Themista käsitellään kuitenkin vain Foucaultin ym. (2018) tutkimukseen perustuen. Themis yhdistää tutkimuskohteena olleen yrityksen versiohallintajärjestelmän sekä edellä mainitun lähdekoodin analysointityökalu SonarCuben samaan ohjelmistoon ja lisää näiden päälle teknisen velan hallinnointia pelillistäviä (engl. gamification) toiminnallisuuksia. Themiksen keskiössä on koko kehitysorganisaatiolle näkyvä pistetaulukko, joka mittaa ohjelmistokehittäjien toteuttamia teknisen velan hallinnoinnin toimenpiteitä pistein. Ohjelmistokehittäjien sijoitus pistetaulukossa määräytyy heidän teknisen velkansa

hallinnointiin toteuttamien toimenpiteiden mukaan. Ohjelmistokehittäjä kirjaa Themikseen tekemänsä hallinnoinnin toimenpiteen, jonka perusteella hallinto joko myöntää tai vähentää pisteitä kehittäjältä. Hallinto pystyy asettamaan myös pelinsisäisiä haasteita tai lisätehtäviä, kuten tietyn velan kappaleen takaisinmaksun tietyssä aikamääreessä, minkä toteuttamalla kehittäjä voi saada lisäpisteitä. Koska Themis sisältää myös lähdekoodin analysointiohjelmiston, pystytään sen datan perusteella automaattisesti vähentämään kehittäjien pisteitä, mikäli hänen havaitaan luovan teknistä velkaa lisääviä kohteita, tai luomaan lisätehtäviä havaittujen kehityskohteiden perusteella. Themiksen koettiin tuovan läpinäkyvyyttä organisaation teknisen velan hallinnointiin. Hallinto pystyi lähes reaaliajassa seuraamaan teknisen velan määrää ja takaisinmaksua, kun taas kehittäjät kokivat pelillistämisen kannustavan heitä velan välttämiseen ja takaisinmaksuun. Kehittäjät myös toivat tekniseen velkaan liittyviä ongelmia aktiivisemmin hallinnon tietoisuuteen. Teknisen velan seuranta ja priorisointi koettiin hallinnon taholta myös helpommaksi pelillistettyjen elementtien ansiosta. Kuten SQALE-menetelmän, niin myös Themiksen heikkoutena on sen keskittyminen ohjelmiston toiminnallisuuksien velan tyyppeihin.

Viimeisenä esiteltävänä työkaluna on Eclipse-ohjelmointiympäristöön lisäosana saatavilla oleva DebtFlag, jolla lähdekoodin analysoinnista löytyvästä teknisestä velasta voidaan automaattisesti koota teknisen velan kappaleiden listaus sekä visualisoida kappaleiden vuorovaikutussuhteet toisiinsa. Lisäosan toiminta perustuu DebtFlag-elementteihin, joiden perusteella teknisen velan kappale lisätään listaukseen. Teknisen velan kappale sisältää yhden tai useamman DebtFlag-elementin, joka puolestaan sisältää viitteen ohjelmiston lähdekoodin velkaa aiheuttavaan sijaintiin. DebtFlag-elementti toisin sanoen linkittää havaitun teknisen velan kappaleen sen aiheuttavaan lähdekoodin osaan. DebtFlag-elementille voidaan myös määrittää tyyppi. DebtFlag-tyyppi ei ole sama asia kuin aiemmin tässä tutkimuksessa käsitelty teknisen velan tyyppi, vaan se on sääntö, jonka perusteella elementin vaikutuksia käsitellään. DebtFlag-elementin linkitysten ja tyyppien perusteella voidaan kyseiseen DebtFlagiin liittyvälle teknisen velan kappaleelle laskea pääoma ja korko sekä arvioida ja kuvata kappaleen vaikutuksia muihin teknisen velan kappaleisiin.

5.4 Johtopäätökset teknisen velan viestinnän työkaluista

Tässä tutkielmassa esiteltyt teknisen velan viestinnän työkalut ovat asiantuntija-arvio, SQALE-menetelmä, Jira-tehtävienhallintaohjelmisto, Themis sekä DebtFlag. Esiteltyt työkalut on listattu alla olevaan teknisen velan viestinnän työkalujen koostetaulukkoon (taulukko 3)

TAULUKKO 3 Teknisen velan viestinnän työkalujen kooste

	Teknisen velan kappaleiden listaus	Teknisen velan koosteenäkymä	Teknisen velan työlista	Ohjelmiston sisäisen teknisen velan kuvaus	Teknisen velan vaikutusten kuvaus
Asiantuntija-arvio				X	X*
SQALE-menetelmä				X	
Jira	X*		X*	X	X*
Themis	X	X	X	X	
DebtFlag	X		X	X	

Taulukossa 3 on esitetty tässä tutkimuksessa esiteltyjä teknisen velan viestinnän työkaluja. Kunkin työkalun osalta on tehty merkintä "X" kuvaamaan työkalun soveltuvuutta aiemmin luvussa 4 esiteltyyn teknisen velan viestinnän lähestymistapaan. Kuten aiemmin tässäkin tutkielmassa on mainittu, suurin osa tarjolla olevista teknisen velan hallinnoinnin ohjelmistotyökaluista rajoittuu käsittelemään vain ohjelmiston toiminnallisuuksien velan tyyppisiä. Tämän takia taulukossa esitetyt merkinnät kuvaavat lähtökohtaisesti vain työkalun soveltuvuutta teknisen velan viestintään vain kyseisten teknisen velan tyyppien osalta. Mikäli työkalu soveltuu kyseiseen lähestymistapaan myös muiden kuin edellä mainittujen kolmen teknisen velan tyyppien osalta, on sen perään merkitty asteriski (*). Esimerkiksi DebtFlag soveltuu listaamaan teknisen velan kappaleista vain arkkitehtuuri-, rakenne- ja koodityypin kappaleita, kun taas tehtävienhallintaohjelmistoja voidaan käyttää myös muiden teknisen velan tyyppien listaamiseen.

Yleisesti voidaan huomata, että kaikki esitellyt työkalut soveltuvat ohjelmiston sisäisen teknisen velan kuvaukseen. Tätä ei voida pitää yllättävänä, sillä kuten tässäkin tutkielmassa on useasti mainittu, suurin osa teknisen velan hallintoihin tarjolla olevista työkaluista on suunniteltu ohjelmiston toiminnallisuuksien velan tunnistamiseen sekä analysointiin ja nämä kolme tyyppiä kattavat ohjelmiston sisäisen teknisen velan lähes täysin. Vastavuoroisesti mitään pelkästään ohjelmiston sisäisiä koodityyppejä käsitteleviä työkaluja ei pystytä käyttämään teknisen velan laajempien vuorovaikutussuhteiden kuvaamiseen. Myöskään Themistä lukuun ottamatta työkalut eivät täytä Shullin ym. (2013) esittämiä koostenäkymän vaatimuksia.

- **Asiantuntija-arvio.** Asiantuntija-arviolla pystytään muuntamaan muiden työkalujen jo tuottama data viestintään hyödynnettäväksi informaatioksi. Tämän takia asiantuntija-arvio soveltuu parhaiten ohjelmiston sisäisen teknisen velan sekä yleisemmin velan vaikutusten kuvaamiseen, mutta sillä ei pystytä tuottamaan kohtuullisella työmäärällä tarvittavaa tietoa muiden lähestymistapojen soveltamiseen.

- **SQALE-menetelmä.** SQALE-menetelmää käytetään usein yhdessä muiden teknisen velan hallinnoin työkalujen, kuten SonarCube-lähdekoodin analysointiohjelman, yhteydessä. SQALE-menetelmällä näiden ohjelmistojen tuottama data pystytään muuntamaan esimerkiksi työtunneiksi tai muiksi kustannuksiksi. Periaatteessa SQALE-arvion edellytyksenä on, että arvioitu kappale on jo tunnistettu. Vaikka tunnistaminen ja SQALE-arvio voidaan toteuttaa lähes samanaikaisesti SQALE:a soveltavan ohjelmistotyökalun toimesta, voidaan SQALE-menetelmä mieltää ennemminkin kyseisen ohjelmistotyökalun viestinnälliseksi lisäosaksi kuin keskeiseksi osaksi itse tunnistamisprosessia, eikä SQALE-menetelmä täten sovellu kuin tunnistetun velan kuvaukseen.
- **Tehtävienhallintaohjelmistot.** Tehtävienhallintaohjelmistot, kuten Jira, ovat yleiskäyttöisiä projektinhallintatyökaluja, joita voidaan soveltaa myös teknisen velan hallintaan ja viestintään. Ainakaan edellä mainitut tehtävienhallintaohjelmistot eivät oletusmäärityksiltään täytä Shullin ym. (2013) teknisen velan koostenäkymän vaatimuksia, mutta tehtävienhallintaohjelmistot sisältävät yleensä erilaisia ohjelmointirajapintoja, joita hyödyntämällä koostenäkymä pystytään rakentamaan. Tähän tutkimukseen ei kuitenkaan löytynyt yhtäkään esimerkkiä, joissa ohjelmistoja olisi käytetty koostenäkymänä tai niiden olisi tunnistettu soveltuvan koostenäkymien luomiseen, minkä takia kyseinen lähestymistapa on jätetty merkittämättä. Löydettyjen esimerkkitapausten perusteella tehtävienhallintaohjelmistojen soveltuvuus teknisen velan viestintään vaikuttaakin riippuvan enemmän ohjelmistoa käyttävästä organisaatiosta kuin ohjelmiston toiminnallisuuksista itsessään.
- **Themis.** Themis kehitettiin erityisesti teknisen velan viestintä huomioiden, joten ei olekaan yllättävää, että sitä voidaan soveltaa lähes kaikkiin esitettyihin viestinnän lähestymistapoihin. Themiksen soveltumisesta koostenäkymäksi voidaan argumentoida puolesta ja vastaan. Isoimpana vastaargumenttina voidaan nähdä Themiksen rajoittuneen edellä mainittuihin kolmeen teknisen velan tyyppiin. Tämä heikkous on tosin yhteinen monien muiden työkalujen kanssa. Themis kuitenkin täyttää Shullin ym. (2013) vaatimukset näiden kohdetyyppien osalta; tietoa kerätään automaattisesti kohdeohjelmiston lähdekoodista, tietolähteitä voidaan vaihtaa helposti, kerätylle tiedolle toteutetaan oikeellisuustarkistuksia ja Themiksen käyttäjälle näkyvä tieto on mukautettavissa aina hallinnon tarpeista ohjelmistokehittäjiin. Tämän vuoksi Themis on merkitty koostenäkymäksi, joskin vain sen käsittelemien velkatyyppien osalta.
- **DebtFlag.** DebtFlag on muihin esiteltyihin työkaluihin verrattuna selkeästi eniten ohjelmistokehityksen vastuutaholle suunnattu työkalu sen päätoiminnallisuuksien keskittyessä lähdekoodin analysointiin. DebtFlagin toiminta kuitenkin perustuu sen tuottamaan teknisen velan listaukseen, ja DebtFlag-linkitykset koostavat teknisen velan kappaleista viestintään hyödynnettävää tietoa. Mikäli velan vaikutuksia tarkentavia työkaluja tai menetelmiä ei organisaatiossa ole käytössä, voidaan DebtFlagin

tuottamien arvojen perusteella laatia pääomaan, korkoon ja velan ohjelmiston sisäisiin vuorovaikutuksiin perustuva ohjelmistokehittäjien teknisen velan työlista.

6 YHTEENVETO JA POHDINTA

Tässä tutkielmassa esiteltiin teknisen velan käsite ja mitä teknisen velan hallinnoinnilla tarkoitetaan. Teknisen velan havaittiin olevan keskeinen ohjelmistokehityksen ongelma, mutta sen hallinnoinnin menetelmät ja työkalut eivät vastaa käytännön tarpeita. Tiedeyhteisön tarjoama teknisen velan tutkimustieto voidaan nähdä puutteellisena eikä ohjelmistoteollisuudessa ole tietotaitoa tarjolla olevien tutkimusten käytännön soveltamiseen. Teknisen velan hallinnoinnin tutkimuksen keskeisenä ongelmana on käytäntöön soveltuvien menetelmien puute, ja tutkimukset keskittyvätkin usein tarjoamaan vain uusia teoreettisia malleja.

Teknisen velan hallitsemattoman kertymisen yhdeksi suurimmaksi aiheuttajaksi koetaan riittämätön viestintä ohjelmistokehityksen ja hallinnon vastuutahojen välillä. Useat tutkimukset havainnoivat, että ohjelmistokehityksen taho ei osaa viestiä teknisen velan määrää ja vaikutuksia liiketoiminnan hallinnolle. Liiketoiminnan hallinto ei kehittäjiltä saamansa tiedon perusteella osaa tehdä oikeita päätöksiä teknisen velan kerryttämisestä ja takaisinmaksusta, vaikka olisikin tietoinen teknisen velan merkityksestä yleisellä tasolla. Viestinnän keskeiseksi ongelmaksi nähdään kuilu kehittäjien ja hallinnon välisessä tietoisuudessa toistensa intresseistä, joiden perusteella tehdyt päätökset eivät huomioi teknisen velan kappaleiden vuorovaikutussuhteita johtaen hallitsemattomaan tekniseen velkaan.

Ohjelmistokehityksen vastuutahoilla tulee olla yhtenäinen käsitys teknisen velan määrästä ja vuorovaikutussuhteista sekä keinot havainnollistaa velan vaikutukset toisen vastuutahon ymmärrettävässä muodossa. Silloin teknisen velan viestintää voidaan toteuttaa tehokkaasti. Tämä tutkielma esittää teknisen velan viestintään menetelmiä ja työkaluja, joita voidaan soveltaa myös käytännössä. Tässä tutkielmassa esiteltyjä menetelmiä ja työkaluja olivat mm. SQALE-menetelmä, Jira-tehtävienhallintajärjestelmä, teknisen velan hallinnoinnin pelillistävä Themis-ohjelmisto, teknisen velan kustannushyötyanalyysi sekä hallinnon yleiset tekniseen velkaan liittyvät viestintäkäytännöt. Aiemman tutkimuksen perusteella ei teknisen velan viestinnän käytännön sovelluksia ole esitelty koostetusti yhdessä tutkimuksessa. Tämä tutkielma on pyrkinyt paikkaamaan kyseistä puutetta teknisen velan tutkimuksessa.

Tämän tutkielman luotettavuuden arvioinnissa huomioitiin seuraavia tekijöitä. Lähteiden laadun arvioinnissa ilmeni, että tämän tutkimuksen kannalta keskeisen lähteen Avgerioun ym. (2016) julkaisu *Dagstuhl reports* on saanut Julkaisufoorumin laatuluokituksen nolla, eli se ei täytä Julkaisufoorumin laatukriteerien vähimmäisvaatimuksia. Avgerioun ym. (2016) tutkimus on kuitenkin Google Scholar- sekä Semantic Scholar -hakupalveluiden mukaan, sen julkaisuajankohta huomioiden, teknisen velan tutkimukseksi laajasti viitattu ja tutkimuksessa esiteltyä teknisen velan käsitelmää sekä teknisen velan hallinnoinnin sisältöä voidaan viittausten perusteella pitää yhtenä aihealueen perusteoksena. Lisäksi monien tutkimuksen tekijöiden tutkimuksia on julkaistu Julkaisufoorumin parhaimman laatuluokitus kolmen omaavissa julkaisuissa, kuten Lin ym. (2015) tutkimukseen osallistunut tutkija Avgeriou julkaisussa *Journal of Systems and Software* sekä Guon ym. (2016) tutkimukseen osallistunut tutkija Seaman julkaisussa *Empirical Software Engineering*. Julkaisufoorumin laatuluokitus ei myöskään eritellyt syytä julkaisun *Dagstuhl reports* heikolle luokitukselle. Tämän tutkimuksen kirjoittaja ei myöskään havainnut muita laadullisia tai julkaisuteknisiä poikkeamia, joten edellä kuvattujen syiden perusteella Avgeriou ym. (2016) on arvioitu laadukkaaksi ja sisälletty lähdeaineistoon Julkaisufoorumin laatuluokituksesta huolimatta.

Kuten aiemmissa luvuissa on useasti korostettu, eivät tekniseen velkaan liittyvät määritelmät ole yhdenmukaisia eri tutkimusten välillä. Tässä tutkimuksessa käytettyjen määritelmien nimet on lähtökohtaisesti valittu kyseisen aihealueen eriten viitatuimmasta tutkimuksesta. Esimerkiksi ennen Lin ym. (2015) teknisen velan tyyppijaottelua on samaa määritelmää kuvattu mm. käsitteillä teknisen velan "lähde" (Brown ym., 2010) ja "ohjelmistokehityksen ulottuvuus" (Tom ym., 2013). Osa tutkimuksista puolestaan käyttää identtisiä käsitteitä, mutta eri määritelmiä. Jälleen Lin ym. (2015) jaottelua käyttäen, on arkkitehtuurityypin tekninen velka määritetty liittyväksi ohjelmiston komponenttien vuorovaikutukseen ja ohjelmiston rakennetyypin velka komponentin sisäisten toiminnallisuuksien vuorovaikutukseen. Koodityypin tekninen velka käsitteää tässä määrittelyssä vain koodin luettavuuden ilman sen teknisiä toiminnallisuuksia ja luettavuutta käsitellään samoin kuin minkä tahansa muunkin tekstilajin luettavuutta. Kuitenkin esimerkiksi Ernst ym. (2015) sisällyttävät kaikki kolme edellä mainittua määritelmää arkkitehtuurikäsitteen alaisuuteen ilman erottelua. Käsite- ja määrittelyeroissa on tämän tutkielman kirjoittaja pyrkinyt yhtenäistämään käsitteet tutkielman ymmärrettävyyden vuoksi.

Käsitteiden käännökset eivät myöskään ole yksiselitteisiä. Alvesin ym. (2016) tyyppijaottelu sisältää alkukielellä tyyppin "design debt", jonka suora suomennos olisi suunnitteluelka. Alves ym. (2016) kuitenkin kuvaavat käsitteen määrittelyssään kyseisen velkatyyppin sisältävän velkaa, joka "voidaan havaita lähdekoodin analysoinnin työkaluilla ja jonka suunnitteluratkaisut ovat hyvien olio-ohjelmoinnin periaatteiden vastaisia (esim. erittäin laajat [koodi]luokat ja [koodi]luokkien tiiviit kytkennät)". Käännös suunnitteluelaksi saattaa antaa lukijalle käsityksen, että kyseisen tyyppin velkaa käsittävät kaikki ohjelmiston tekniset suunnitteluratkaisut, vaikka ne on erikseen jaoteltu omiin

kokonaisuuksiinsa, kuten edellisen kappaleen arkkitehtuurivelan esimerkistä voidaan havaita. ”Design debt” viittaa selkeästi vain ohjelman lähdekoodin jäsentämiseen ja olioiden välisiin suhteisiin, jota käännöksen rakennevelka voidaan perustella kuvaavan paremmin.

Teknisen velan viestintä, kuten monet muutkin teknisen velan aihealueet, on vielä suhteellisen uusi tutkimuksen kohde ja etenkin empiiristen tutkimusten toteuttamisesta on esitetty laajasti toiveita tutkimusyhteisössä. Tämän tutkielman perusteella voidaankin ehdottaa jatkotutkimuksen kohteeksi tässä tutkielmassa esiteltyjen teknisen velan viestinnän menetelmien laajamittaisempaa vaikutusten arviointia varsinkin, kun teknisen velan viestintä on päätutkimusaiheena harvinaisuus. Toimivan teknisen velan viestinnän edellytyksenä tosin on eri vastuutahojen syvällisempi perehdyttäminen teknisen velan hallinnoinnin periaatteista. Tietämättömyys teknisestä velasta havaittiin monissa tutkimuksissa isoksi viestinnän esteeksi. Tämän koulutuksen vaikutus teknisen velan hallinnoinnin tehokkuuteen on myös potentiaalisesti sekä akateemista yhteisöä että ohjelmistoteollisuutta hyödyttävä tutkimuskohde.

Monet etenkin varhaisemmat teknisen velan viestintää sivuavat tutkimukset, kuten Lim ym. (2012) korostivat avoimen vuoropuhelun merkitystä vastuutahojen välillä käyttämättä valmiita viestinnän malleja tai viitekehyksiä. Nykyisin taas monet tutkimukset ehdottavat paikoin hyvinkin monimutkaisia kustannusarvioiden menetelmiä parhaimmaksi keinoksi teknisen velan viestintään. Tieteenalan kehittyminen on varmastikin vaikuttanut tutkimusten keskittyvän entistä enemmän marginaalisesti edellistä menetelmää tehokkaamman menetelmän löytämiseen, mutta tutkimusaiheeseen vasta perehtyneenä lukijana heräsi ajatuksia, onko tutkimuksista jossakin vaiheessa unohtunut esiteltyjen menetelmien käytännön soveltuvuus. Tämä kysymys osaltaan oli keskeisenä henkilökohtaisena motivaationa tutkimusaiheen valinnassa.

Päätän tutkielmani seuraavaan huomioon: de Almeida ym. (2018) esittävät, että ”tekniset johtajat ja järjestelmäarkkitehdit ovat vastuussa teknisen velan vaikutuksien arvioinnista”. Samassa tutkimuksessa, yhden kappaleen erottaessa edellä mainittu väite, toteaa eräs tutkimukseen haastateltu liiketoiminnan vastuutahon edustaja ”nykyään ihmiset ovat erittäin erikoistuneita omiin [tehtävä]alueisiinsa ja liiketoiminnan kokonaiskuvan hahmottaminen on haastavaa” kuvaillessaan liiketoiminnan kokonaiskuvan saavuttamisen haastavuutta. Liiketoiminnan vastuutahoilla ei siis ole kokonaiskuvaa liiketoiminnasta, joten miten teknisen puolen henkilöstön voidaan olettaa pystyvän arvioimaan teknisen velan liiketoiminnallisia vaikutuksia, kuten tässäkin tutkielmassa usein on esitetty? Ja mikäli liiketoiminnalla ei ole käsitystä liiketoiminnan kokonaiskuvasta, niin kenellä on? Teknisen velan tutkimukselle saattaisi tästä näkökulmasta olla hyödyllistä arvioida uudelleen, mitkä ovat ohjelmistoteollisuuden käytännön ympäristöissä vallitsevat edellytykset ehdotettujen mallien soveltamiselle.

LÄHTEET

- de Almeida, R. R., Kulesza, U., Treude, C. & Lima, A. H. G. (2018, September). Aligning technical debt prioritization with business objectives: A multiple-case study. Teoksessa *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 655-664). IEEE.
- Alves, N. S., Mendes, T. S., de Mendonça, M. G., Spínola, R. O., Shull, F. & Seaman, C. (2016). Identification and management of technical debt: A systematic mapping study. *Information and Software Technology*, 70, 100-121.
- Alves, N. S., Ribeiro, L. F., Caires, V., Mendes, T. S. & Spínola, R. O. (2014, September). Towards an ontology of terms on technical debt. Teoksessa *2014 Sixth International Workshop on Managing Technical Debt* (pp. 1-7). IEEE.
- Ampatzoglou, A., Ampatzoglou, A., Chatzigeorgiou, A. & Avgeriou, P. (2015). The financial aspect of managing technical debt: A systematic literature review. *Information and Software Technology*, 64, 52-73.
- Apa, C., Jeronimo, H., Nascimento, L. M., Vallespir, D. & Travassos, G. H. (2020). The perception and management of technical debt in software startups. Teoksessa *Fundamentals of Software Startups* (pp. 61-78). Springer, Cham.
- Arvanitou, E. M., Ampatzoglou, A., Bibi, S., Chatzigeorgiou, A. & Stamelos, I. (2019). Monitoring technical debt in an industrial setting. Teoksessa *Proceedings of the Evaluation and Assessment on Software Engineering* (pp. 123-132).
- Avgeriou, P., Kruchten, P., Ozkaya, I. & Seaman, C. (2016). Managing technical debt in software engineering (dagstuhl seminar 16162). Teoksessa *Dagstuhl reports* (Vol. 6, No. 4). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- Avgeriou, P., Taibi, D., Ampatzoglou, A., Fontana, F. A., Besker, T., Chatzigeorgiou, A., Lenarduzzi V., Martini, A., Moschou, N., Pigazzini I., Saarimäki, N., Sas, D., de Toledo, S.S. & Tsintzira, A. (2020). An overview and comparison of technical debt measurement tools. *Ieee software*, 38(3), 61-71.
- Behutiye, W. N., Rodríguez, P., Oivo, M. & Tosun, A. (2017). Analyzing the concept of technical debt in the context of agile software development: A systematic literature review. *Information and Software Technology*, 82, 139-158.
- BenIdris, M. (2020). Investigate, identify and estimate the technical debt: a systematic mapping study. *Available at SSRN 3606172*.
- Besker, T., Martini, A. & Bosch, J. (2017, May). Time to Pay Up: Technical Debt from a Software Quality Perspective. In *CibSE* (pp. 235-248).

- Besker, T., Martini, A. & Bosch, J. (2019a). Software developer productivity loss due to technical debt – A replication and extension study examining developers' development work. *Journal of Systems and Software*, 156, 41-61.
- Besker, T., Martini, A. & Bosch, J. (2019b, May). Technical debt triage in backlog management. Teoksessa *2019 IEEE/ACM International Conference on Technical Debt (TechDebt)* (pp. 13-22). IEEE.
- Besker, T., Martini, A. & Bosch, J. (2022). The use of incentives to promote technical debt management. *Information and Software Technology*, 142, 106740.
- Brenner, R. (2019, May). Balancing resources and load: Eleven nontechnical phenomena that contribute to formation or persistence of technical debt. Teoksessa *2019 IEEE/ACM International Conference on Technical Debt (TechDebt)* (pp. 38-47). IEEE.
- Brown, N., Cai, Y., Guo, Y., Kazman, R., Kim, M., Kruchten, P., Lim E., MacCormack A., Nord, R., Ozkaya, I., Sangwan, R., Seaman, C., Sullivan, K. & Zazworka, N. (2010, marraskuu). Managing technical debt in software-reliant systems. Teoksessa *Proceedings of the FSE/SDP workshop on Future of software engineering research* (47-52). Clements, P. & Northrop, L. (2002). *Software product lines*. Boston: Addison-Wesley.
- Codabux, Z. & Williams, B. (2013, May). Managing technical debt: An industrial case study. Teoksessa *2013 4th International Workshop on Managing Technical Debt (MTD)* (pp. 8-15). IEEE.
- Cunningham, W. (1992). The WyCash portfolio management system. *ACM SIGPLAN OOPS Messenger*, 4(2), 29-30.
- Ernst, N. A., Bellomo, S., Ozkaya, I., Nord, R. L. & Gorton, I. (2015, August). Measure it? manage it? ignore it? software practitioners and technical debt. Teoksessa *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering* (pp. 50-60).
- Fernández-Sánchez, C., Garbajosa, J., Yagüe, A. & Perez, J. (2017). Identification and analysis of the elements required to manage technical debt by means of a systematic mapping study. *Journal of Systems and Software*, 124, 22-38.
- Foucault, M., Blanc, X., Storey, M. A., Falleri, J. R. & Teyton, C. (2018). Gamification: a game changer for managing technical debt? a design study. *arXiv preprint arXiv:1802.02693*.
- Guo, Y., Spínola, R. O. & Seaman, C. (2016). Exploring the costs of technical debt management—a case study. *Empirical Software Engineering*, 21(1), 159-182.
- Izurietta, C., Ozkaya, I., Seaman, C., Kruchten, P., Nord, R., Snipes, W. & Avgeriou, P. (2016, January). Perspectives on managing technical debt: A transition point and roadmap from Dagstuhl. Teoksessa *Joint of the 4th International Workshop on Quantitative Approaches to Software Quality*,

QuASoQ 2016 and 1st International Workshop on Technical Debt Analytics, TDA 2016 (pp. 84-87).

- Klinger, T., Tarr, P., Wagstrom, P. & Williams, C. (2011, May). An enterprise perspective on technical debt. *Teoksessa Proceedings of the 2nd Workshop on managing technical debt* (pp. 35-38).
- Kruchten, P., Nord, R. L. & Ozkaya, I. (2012). Technical debt: From metaphor to theory and practice. *Ieee software*, 29(6), 18-21.
- Kruchten, P., Nord, R. L., Ozkaya, I. & Falessi, D. (2013). Technical debt: towards a crisper definition report on the 4th international workshop on managing technical debt. *ACM SIGSOFT Software Engineering Notes*, 38(5), 51-54.
- Lenarduzzi, V., Saarimäki, N. & Taibi, D. (2019, September). The technical debt dataset. *Teoksessa Proceedings of the Fifteenth International Conference on Predictive Models and Data Analytics in Software Engineering* (pp. 2-11).
- Letouzey, J. L. (2012, June). The SQALE method for evaluating technical debt. *Teoksessa 2012 Third International Workshop on Managing Technical Debt (MTD)* (pp. 31-36). IEEE.
- Li, Z., Avgeriou, P. & Liang, P. (2015). A systematic mapping study on technical debt and its management. *Journal of Systems and Software*, 101, 193-220.
- Lim, E., Taksande, N. & Seaman, C. (2012). A balancing act: What software practitioners have to say about technical debt. *IEEE software*, 29(6), 22-27.
- Marinescu, R. (2012). Assessing technical debt by identifying design flaws in software systems. *IBM Journal of Research and Development*, 56(5), 9-1.
- Nielsen, M. E., Østergaard Madsen, C. & Lungu, M. F. (2020, August). Technical debt management: A systematic literature review and research agenda for digital government. *Teoksessa International Conference on Electronic Government* (pp. 121-137). Springer, Cham.
- Pacheco, A., Marín-Raventós, G. & López, G. (2018, September). Designing a technical debt visualization tool to improve stakeholder communication in the decision-making process: a case study. *Teoksessa International Conference on Research and Practical Issues of Enterprise Information Systems* (pp. 15-26). Springer, Cham.
- Rios, N., de Mendonça Neto, M. G. & Spínola, R. O. (2018). A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners. *Information and Software Technology*, 102, 117-145.
- Seaman, C. & Guo, Y. (2011). Measuring and monitoring technical debt. *Teoksessa Advances in Computers* (Vol. 82, pp. 25-46). Elsevier.
- Shull, F., Falessi, D., Seaman, C., Diep, M. & Layman, L. (2013). Technical debt: Showing the way for better transfer of empirical results. *Teoksessa*

Perspectives on the Future of Software Engineering (pp. 179-190). Springer, Berlin, Heidelberg.

- Siponen, M. & Baskerville, R. L. (2018). Intervention effect rates as a path to research relevance: Information systems security example. *Journal of the Association for information Systems*, 19(4).
- Spínola, R. O., Zazworka, N., Vetro, A., Shull, F. & Seaman, C. (2019). Understanding automated and human-based technical debt identification approaches-a two-phase study. *Journal of the Brazilian Computer Society*, 25(1), 1-21.
- Tom, E., Aurum, A. & Vidgen, R. (2013). An exploration of technical debt. *Journal of Systems and Software*, 86(6), 1498-1516.
- Yli-Huumo, J., Maglyas, A. & Smolander, K. (2016). How do software development teams manage technical debt?—An empirical study. *Journal of Systems and Software*, 120, 195-218.