

Jaana Lehojärvi

**TEKNINEN VELKA HALLINNOINNIN NÄKÖKUL-
MASTA**

TUNNUSTA, TUNNISTA JA RAJOITA



JYVÄSKYLÄN YLIOPISTO
TIETOJENKÄSITTELYTIETEIDEN LAITOS
2016

TIIVISTELMÄ

Lehojärvi, Jaana

Tekninen velka hallinnoinnin näkökulmasta - tunnusta, tunnista ja rajoita

Jyväskylä: Jyväskylän yliopisto, 2016, 37 s.

Tietojärjestelmätiede, kandidaatin tutkielma

Ohjaaja(t): Seppänen, Ville

Teknisen velan määrää nykyisissä ohjelmistoissa ei tiedetä tarkasti, eikä ole tarkkaa tietoa myöskään sen hinnasta, mutta globaalin IT-velan on ennustettu mahdollisesti nousevan yhteen triljoonaan dollariin vuoden 2015 aikana. Teknisen velan vaikutukset taas ulottuvat kaikkiin ohjelmistotuotantoyrityksiin ja niiden kautta käyttäjiin ja näin ollen koko ympäröivään maailman. Tutkielmasa käsitellään teknistä velkaa, mitä tekninen velka on, mistä teknisen velan hallinnointi koostuu ja kuinka teknistä velkaa hallinnoidaan. Tutkielman tutkimusmenetelmänä on kirjallisuuskatsaus, jonka kautta asioita käsitellään. Teknisen velan metafora on esitelty jo 1990-luvulla, jolloin sen perusteena oli hätäisesti tehty koodi. Tämän jälkeen metaforaa, ja määritelmää yleisesti, on laajennettu koskemaan lähes kaikkia ohjelmistotuotantoon kuuluvia asioita ja tätäkin laajemmin. Yleistäen voitaneen sanoa, että tekninen velka on jokin ohjelmistossa oleva tai siihen liittyvä asia, joka haittaa sujuvaa ohjelmiston jatkokehitystä tai ylläpitoa ja lisäksi sillä on merkittävä rahallinen vaikutus. Toisaalta tekninen velka on hyvin monitahoinen, vivahteikas ja hankalakin asia, johon ei löydy yhtä yksiselitteistä vastausta, vaan se on aina riippuvainen myös kontekstista, jossa sitä käytetään. On hyvä myös huomata, että teknistä velkaa syntyy kaikilla organisaation tasoilla ja että tekninen velka on koko organisaation yhteinen asia. Teknisen velan hallinnoinnilla hoidetaan ja seurataan muun muassa teknisten velkojen hinnan muodostumista, priorisointia, takaisinmaksua ja koron kasvua. Teknisen velan hallinnointi nähdään hyvin tärkeänä välineenä hallita niin ohjelmistojen laatua kuin niiden kustannuksiakin, unohtamatta kilpailutilanteita ja liiketoiminnallisia strategioita. Teknisen velan hallinnointiin on vähänlaisesti työkaluja, eivätkä ne kata kaikkia hallinnoinnin osa-alueita. Toisaalta se, millä välineillä hallinnointia hoidetaan, ei lopulta ole ratkaisevaa, kunhan kaikki osapuolet ovat aina tietoisia teknisen velan kulloisestakin tilanteesta. Tutkielman lopputuloksena ovat yleiskuvaukset teknisestä velasta, sen hallinnoinnista sekä näiden merkityksestä.

Asiasanat: tekninen velka, teknisen velan hallinnointi

ABSTRACT

Lehojärvi, Jaana

Management aspect on technical debt - admit, identify and prevent

Jyväskylä: University of Jyväskylä, 2016, 37 p.

Information systems science, bachelor's thesis

Supervisor(s): Seppänen, Ville

The amount of the technical debt in existing software is not known, and there is no accurate information on either of its price, but it has been estimated that the global IT debt has potential to rise to one trillion dollars in 2015. Implications of technical debt are reaching out to all software engineering companies and through them to the surrounding world. This thesis concerns technical debt and technical debt management. Research method used in this thesis is a literature review, through which the issues will be addressed. Technical debt metaphor has been introduced already in the 1990s, when it based on the 'not-so-good' code. After this, the metaphor, and the definition in general, has been extended to almost all matters falling within the software development process and even more broadly. In general, it can be said that the technical debt is any issue in the software or any related issue that hampers the smooth software further development or its maintenance, and in addition, that issue has a significant financial impact. On the other hand, the technical debt is very complex, nuanced and a difficult subject. Technical debt is also always dependent on the contexts in which it is used. It is also very important to notice that technical debt can be caused by all levels of an organization. Technical debt management is used for example to handle technical debt principal, prioritization, repayment and interest rate growth. Management of technical debt is seen as a very important tool to manage both software quality and its costs, not forgetting the competition conditions and business strategies. The management of technical debt has only few tools, and those do not cover all areas of technical debt management. On the other hand, it is not important which tools are used to manage technical debt, as long as all stakeholders are always aware of the situation with the respective technical debt. As a result, this thesis gives general descriptions of the technical debt and its management with associated activities.

Keywords: technical debt, technical debt management

KUVIOT

KUVIO 1 Tekninen velka ja sen korko.....	10
KUVIO 2 Teknisen velan puitteet.....	12
KUVIO 3 Yhdistelmä teknisen velan kahdesta perustyyppistä eri määrittelemien mukaan.....	13
KUVIO 4 Teknisen velan jako tietoisiin ja tahattomiin.....	14
KUVIO 5 Teknisen velan syitä ja seurauksia jaettuna kahteen perustyyppiin...	19
KUVIO 6 Malli teknisen velan hallinnoinnista.....	23
KUVIO 7 Teknisen velan hallinnointia tukevat työkalut toiminnoittain.....	30
KUVIO 8 Teknisen velan hallinnointia tukevat työkalut tyypeittäin.....	31

TAULUKOT

TAULUKKO 1 Yhdistelmä teknisen velan ominaisuuksista sekä huomioita näistä	15
TAULUKKO 2 Teknisen velan hallinnoinnin jako kahdeksaan osaan	22

SISÄLLYS

TIIVISTELMÄ.....	2
ABSTRACT.....	3
KUVIOT	4
TAULUKOT	4
SISÄLLYS.....	5
1 JOHDANTO	6
2 TEKNINEN VELKA	8
2.1 Määritelmät	8
2.2 Luokittelut.....	12
2.3 Ominaisuudet.....	15
2.4 Hinta	16
2.5 Syntyminen.....	17
3 TEKNISEN VELAN HALLINNOINTI	21
3.1 Hallinnoinnin osa-alueet.....	22
3.2 Tunnistaminen, ehkäiseminen ja mittaaminen	24
3.3 Priorisointi, takaisinmaksu ja seuranta.....	26
3.4 Viestintä ja dokumentointi.....	28
3.5 Strategiat ja työkalut.....	29
4 YHTEENVETO JA POHDINTA	32
LÄHTEET	35

1 JOHDANTO

Tekninen velka (Technical Debt, TD) on kiinnostava aihe sen vaikutusten ulottuessa kaikkiin ohjelmistotuotantoyrityksiin ja niiden kautta kaikkien ohjelmistojen käyttäjiin asti ja näin ollen koko ympäröivään maailmaan. Toisaalta tekninen velka on hyvin monitahoinen, vivahteikas ja hankalakin asia, johon ei löydy yhtä yksiselitteistä vastausta, vaan se on aina riippuvainen myös kontekstista, jossa sitä käytetään. Yksinkertaistettuna voidaan sanoa, että teknistä velkaa on esimerkiksi puutteellisesti ja oikein tehtyjen koodien erotus. Tuo erotus on eräänlainen velka, joka on maksettava takaisin eli puutteellisesti tehdyt koodit on jossain vaiheessa korjattava. Velalle alkaa myös kasvaa korkoa, joka taas kuvaa sitä lisäpanostusta, jonka myöhemmin tehtävä työ vaatii, mitä myöhemmin se tehdään sitä kalliimpaa ja hankalampaa se on. Teknisen velan määrää nykyisissä ohjelmistoissa ei tiedetä tarkasti, eikä ole tarkkaa tietoa myöskään sen hinnasta, mutta Gartner (Stamford, 2010) on vuonna 2010 ennustanut globaalin IT-velan mahdollisesti nousevan 1 triljoonaan dollariin vuoden 2015 aikana. Ensimmäisen kerran teknisen velan metafora esiteltiin, Ward Cunninghamin toimesta, jo vuonna 1992. 2000-luvulla on alettu tutkia enemmän asiaa, mutta vasta viime vuosina tekninen velka on saanut paljon enemmän huomiota ja tutkimusta, kun myös yritysmaailma on ilmaissut kiinnostuksensa asiaan. Li, Avgeriou ja Liang (2015) olivat myös tutkimuksessaan havainneet teknisestä velasta julkaistujen tutkimuksien määrän lisääntyneen merkittävästi vuodesta 2008 vuoteen 2013 tultaessa.

Lim, Taksande ja Seamanin (2012) mukaan teknisen velan hallinnointia (Technical Debt Management, TDM) on parhaan kompromissin löytäminen kullekin projektille. Heidän mukaansa siihen kuuluu myös halukkuus hyväksyä joitakin teknisiä riskejä liiketoiminnan tavoitteiden saavuttamiseksi, ja toisaalta ymmärtää tarve asiakkaiden odotusten lieventämiselle, jotta voidaan taata riittävä ohjelmiston laatu.

Tutkimuksen tavoitteena on antaa yleiskuva teknisestä velasta hallinnoinnin näkökulmasta, mitä tekninen velka on ja kuinka teknistä velkaa hallinnoidaan. Tutkimuskysymys on:

1. Miten teknistä velkaa hallinnoidaan?

Lisäksi ongelman kuvaamista helpottamaan on käytetty alatutkimuskysymyksiä:

1.1. Mitä tekninen velka on?

1.2. Mistä teknisen velan hallinnointi koostuu?

Tutkielman tutkimusmenetelmänä on kirjallisuuskatsaus, jonka kautta asioita käsitellään. Kirjallisuutta etsittäessä on käytetty seuraavia hakusanoja "Debt", "Technical debt", "Design", "Architecture", "Management", "Prioritizing", "Monitoring" sekä erilaisia yhdistelmiä näistä. Aineistoa on etsitty muun muassa seuraavista kannoista ACM, IEEE Xplore ja Elsevier. Lisäksi on hyödynnetty jo löydettyjen artikkeleiden lähdeluetteloita.

Tutkielma etenee siten, että toisessa luvussa käydään läpi teknistä velkaa ja sen käsitettä määritelmien, luokittelujen, ominaisuuksien, hinnan ja syntyminen kautta. Kolmannessa luvussa tarkastellaan teknisen velan hallinnointia, tutkien mitä hallinnoinnilla tarkoitetaan, mistä osista se koostuu ja mitä nämä yksittäiset osat ovat. Neljännessä luvussa esitetään tutkielman yhteenveto sekä pohditaan saatujen tulosten valossa teknisen velan olemusta ja sen tulevaisuuden näkymiä.

2 TEKNINEN VELKA

Tässä luvussa tutkitaan teknisen velan käsitettä erilaisten näkökulmien kautta, jotta saadaan selkeämpi käsitys siitä mitä tekninen velka on. Teknistä velkaa voidaan tarkastella monesta eri näkökulmasta tai mittakaavasta, esimerkiksi: teoreettisesti, käytännön kautta, projektitasolla tai koko ekosysteemin kattavana. Se kuinka teknistä velkaa tarkastellaan määrittää myös kyseisessä yhteydessä tarvittavia yksityiskohtia, rajouksia ja näkemyksiä. Monen muun tavoin myös Li ym. (2015) ovat havainneet tutkimuksessaan, että vaikka tekninen velka on saanut merkittävää huomiota viime vuosina, niin tutkijat ja käytännön tekijät, ohjelmistoalan ammattilaiset, hahmottavat teknisen velan käsitettä edelleen hieman eri tavoin. Arkkitehtuurisen teknisen velan lisäksi ohjelmistoalan ammattilaiset eivät ole täysin samaa mieltä muista teknisen velan tyypeistä (Ernst, Bellomo, Ozkaya, Nord, & Gorton, 2015). Tämän lisäksi samanaikaisesti on tutkijoiden kesken olemassa vielä erilaisia epäselvyyksiä teknisen velan olemuksesta ja sen rajoista. Esimerkiksi, on edelleen vielä hieman epäselvää, mitä voidaan ohjelmistokehityksestä luokitella tekniseksi velaksi ja mitä ei. Kyse on eräänlaisista kompromisseista liittyen siihen mitä järjestelmän laatuominaisuuksia pidetään teknisenä velkana ja minkä taas voidaan ajatella olevan teknisen velan metaforan rajoja. (Li ym., 2015.)

Seuraavassa alaluvussa käydään läpi teknisen velan määritelmiä, jonka jälkeen paneudutaan vielä tarkemmin tekniseen velkaan sen erilaisten luokittelujen kautta, josta edetään teknisen velan ominaisuuksien ja hinnan kautta siihen kuinka ja miksi teknistä velkaa syntyy.

2.1 Määritelmät

Cunninghamin (1992) mukaan koodin lähettäminen ensimmäistä kertaa tuotantoon on kuin tekisi velkaa. Pieni velka nopeuttaa kehitystä, kunhan se vain maksetaan viipymättä ja paranneltuna takaisin. Hänen mukaansa vaara piilee siinä, jos velkaa ei makseta takaisin. Jokainen minuutti, joka on vietetty ei-ihan-

oikeanlaisen koodin kanssa, lasketaan korkona velalle. (Cunningham, 1992.) Tekninen velka voidaan kuvata myös kaikkina niinä oikoteinä, jotka säästävät rahaa tai nopeuttavat kehitystä tänään, riskinä se, että se mahdollisesti lisää kustannuksia tai hidastaa kehitystä monesti epäselvässä tulevaisuudessa. Tekninen velka on väistämätön ja voi olla jopa hyvä asia, kunhan se hallinnoidaan oikein. Hallinnoiminen voi kuitenkin olla hankalaa, sillä teknistä velkaa syntyy lukuisista eri syistä, ja velan vaikutuksien ennustaminen on usein vaikeaa; tämä sisältää yleensä myös uhkapeliä siitä, mitä tulevaisuudessa tapahtuu. (Allman, 2012.) Tekninen velka kuvastaa teknisiä kompromisseja, jotka voivat tuottaa lyhyen tähtäimen hyötyjä, mutta toisaalta voivat vahingoittaa ohjelmiston terveyttä pitkällä aikavälillä (Li ym., 2015). Tekninen velka voidaan määritellä myös ohjelmistoa koskevien menneiden päätösten näkymättöminä seurauksina, jotka vaikuttavat ohjelmiston tulevaisuuteen (Falessi, Kruchten, Nord & Ozkaya, 2014).

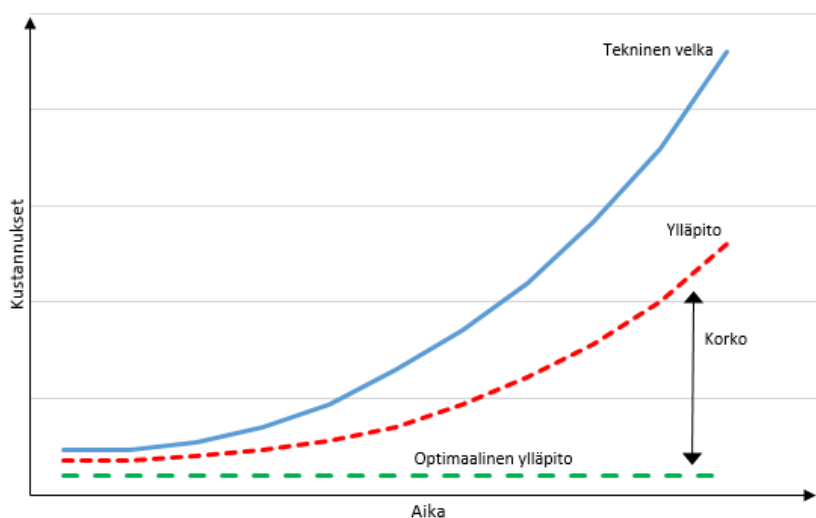
Toisaalta teknistä velkaa voidaan tarkastella myös käytännönläheisemmin. Tekninen velka on metafora, joka kuvaa tehtyjä kompromisseja teknisen kehityksen toiminnoissa, jotka taas viivästyvät, jotta saadaan lyhyen tähtäimen etuja, kuten ajoissa oleva ohjelmistoversio. Termiä "velka" käytetään kuvaamaan miten nämä ohitetut toiminnot voisi maksaa pois lyhyellä aikavälillä. Kuten esimerkiksi otettaessa uuden auton ostoon rahoitusvelkaa, joka on maksettava myöhemmin takaisin. (Zazworka, Seaman & Shull, 2011.) Vastaavasti, viivästyttämekö toimitusta päivällä saadaksemme kommentoitua ohjelmamoduulin koodin tiukan julkaisuajataulun paineessa, vai lähetämmekö kommentoitoman ohjelmamoduulin tuotteen mukana (McGregor, Monteith & Zhang, 2012)? Entä käytämmekö suunnittelumalleja, jotka voidaan sijoittaa nykyiseen malliin, vai sellaisia, jotka parantavat enemmän haluttuja ominaisuuksia, mutta aiheuttavat merkittäviä muutostarpeita? Myös tällaiset skenaariot ovat olennainen osa teknistä velkaa. (McGregor ym., 2012.)

Teknistä velkaa voidaan tarkastella myös suuremmassa mittakaavassa, ekosysteemitasolla. McGregor ym. (2012) määrittelevät ekosysteemin asiayhteydeksi, jossa organisaatiot tuottavat ja käyttävät ohjelmistoja. Ne harjoittavat liiketoimintaa toistensa kanssa ja suorittavat näitä toimia uusilla ja innovatiivisilla tavoilla. Suhteet tällaisissa ekosysteemeissä luovat suoria ja epäsuoria riippuvuuksia. Ekosysteemi kattaa myös organisaatioiden toimitus- ja arvoketjut. (McGregor, ym., 2012.) Tämä merkitsee, että päätös, jonka yksi organisaatio on tehnyt voi levitä läpi ekosysteemin ja vaikuttaa tahattomasti muihin organisaatioihin (McGregor ym., 2012). Tämä tarkoittaa myös sitä, että teknistäkin velkaa siirtyy osapuolelta toiselle. McGregor ym. (2012) havaitsivat, että kun tekninen velka tietyssä komponentissa saattaa heikentyä siirtyessään, komponenttien yhdistämisessä tekninen velka voi pahentua. Merkiten, että teknisen velan siirtyessä seuraavaan solmuun, on se suurempi kuin yhdistelmä teknisistä veloista toimitussolmuissa.

Wolff ja Johann (2015) tuovat tutkimuksessaan esille, että tekninen velka kytkeytyy luonnollisesti ohjelmiston laatuun ja että sitä on olemassa kahta tyyppiä. Ensimmäinen tyyppi on ulkoinen laatu, joka koetaan käyttäjän tai asi-

akkaan toimesta. Sitä voi olla ohjelmiston suorituskyky, turvallisuus, skaalautuvuus, vakaus ja niin edelleen. Tällaista laatua voidaan mitata ja käyttäjät kokevat sen ohjelmistoa käyttäessään. Toinen tyyppi on sisäinen laatu, jonka voivat havaita vain kehittäjät. Se on jotain, mikä tekee koodin jatkokehityksestä ja ylläpitämisestä joko helpompaa tai vaikeampaa. Hankalaa ohjelmistokehityksessä on se, että sisäistä laatua ei oikeastaan voi havaita kuin tekniset henkilöt. Muiden kuin teknisten henkilöiden on vaikea nähdä, mitä tämä sisäinen laatu todella on ja miten se itse asiassa vaikuttaa kehitysprosessiin. (Wolff & Johann, 2015.)

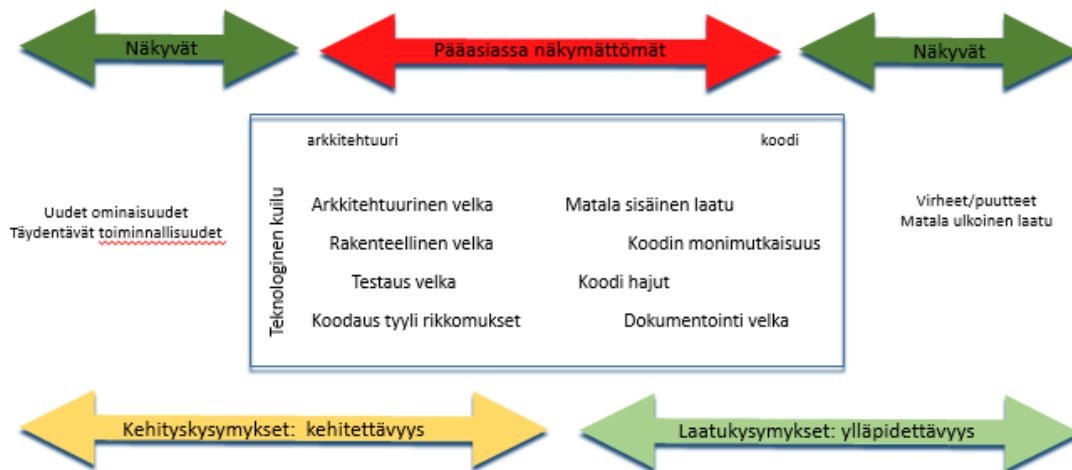
Teknisen velan määritelmää on Cunninghamin metaforan jälkeen sekä laajennettu että supistettu useasti eri ihmisten toimesta. McConnell (2007) määritteli teknisen velan suunnittelun tai rakentamisen lähestymistavaksi, joka voi olla tarkoituksenmukainen lyhyellä aikavälillä. Samalla kuitenkin luoden erilaisia teknisiä konteksteja, jotka merkitsevät, että tuon saman työn tekeminen myöhemmin maksaa enemmän kuin jos se olisi tehty heti. Tekninen velka voidaan määritellä myös niinä kustannuksina, joita on tullut, kun ohjelmiston teknistä laatua on parannettu optimaaliselle tasolle. Teknisen velan korkoa ovat taas ne ylimääräiset kustannukset, jotka on käytetty ohjelmiston ylläpitoon sen huonon teknisen laadun takia. Tekninen velka ja sen korko tyypillisesti kasvavat jatkuvasti ajan kuluessa, ellei niitä ratkaista kuten kuvio 1) nähdään. (Nugroho, Visser, & Kuipers, 2011, 2.) Toisaalta teknisen velan voidaan ajatella olevan niiden tuotannossa olevien koodien korjaamisen kustannuksia, jotka johtuvat koodien rakenteellisista laatuongelmista. Nimenomaan niiden ongelmien, jotka jo tiedetään poistettaviksi, jotta voidaan valvoa kehityskustannuksia tai välttää toiminnalliset ongelmat. (Curtis, Sappidi & Szykarski, 2012.) Tosin on huomattava, että tekninen velka ei ole pelkästään huonoa laatua, se ei myöskään ole virheitä ja se on vielä hankalasti mitattavissa (Kruchten, Nord, Ozkaya, & Falessi, 2013).



KUVIO 1 Tekninen velka ja sen korko (Nugroho ym., 2011, 2)

Brown ym. (2010) mukaan tekninen velka on yksi ulottuvuus suuremmissa arvostusprosessissa, eräänlainen linssi, jonka kautta voidaan tarkkailla ohjelmistokehitystä taloudellisen strategian näkymästä. Tämä taas edellyttää keinoja arvioida arvon luomisen mahdollisuuksia ohjelmistoissa. Siinä hankkeita verrataan toisiin harkinnanalaisiin hankkeisiin näiden kustannusten suhteen. (Brown, ym., 2010.) Voidaan ajatella myös, että tekninen velka on mikä tahansa kuuluu teknologiainfrastruktuurin sisällä tai sellainen sen toteutus, jolla on materiaallinen vaikutus vaadittuun laatuun (Theodoropoulos, Hofberg & Kern, 2011). Tässä ajatuksena on, että sitomalla vaadittu laatu määritelmään, luodaan arvopohjainen näkemys, joka on linjassa sidosryhmien omien näkökulmien kanssa (Theodoropoulos ym., 2011). Kun sidosryhmille annetaan mahdollisuus laadullisten haasteiden arviointiin ja hallintaan johdonmukaisesti läpi koko yrityksen, saadaan mahdollisuus tehokkaampaan yhteistyöhön kohti yhteisiä tavoitteita. Kun lisäksi yhdistetään asiantuntemukset liiketoiminnan, teknologian ja riskienhallinnan keinoista, maksimoidaan vaikuttavuutta yhtenäisyydellä teknistä velkaa vastaan. (Theodoropoulos ym., 2011.) Yritysten sovellussalkuissa piilossa olevat kustannukset ja riskit täytyy tehdä näkyviksi, sekä rakentaa vertailukohtia investointien päätöksenteolle, jotka koskevat sovellusten laatua, erityisesti rakenteellista laatua (Curtis ym., 2012).

Yhdistävänä näkökulmana on syntymässä ajatus teknisestä velasta ohjelmiston aiemmin tehtyjen päätösten summana, joka vaikuttaa sen tulevaisuuteen. Kruchen ym. (2013, 52) esittelivät teknisen velan puitteet kuvaaman teknisen velan konteksteja sekä konkretisoimaan teknisen velan käsitettä (ks. kuvio 2). Heidän mukaansa ohjelmistojen erilaiset kehityskivut, ongelmakohtat, voidaan löytää paremmin tunnistamalla konkreettisia teknisen velan yksiköitä, tarkastelemalla niiden vaikutuksia sekä arvioimalla elinkaaren niihin liittyviä kustannuksia. Ongelmia löydetään myös esittelemällä erilaisia mekanismeja, joilla voidaan kuvata teknistä velkaa sekä arvioimalla velan vaikutuksia. Näin teknisen velan vähentämistä voidaan sitten suunnitella yhdessä muiden kohteiden, kuten uusien ominaisuuksien, vikojen ja arkkitehtuuristen elementtien kanssa. (Kruchen, ym., 2013.)



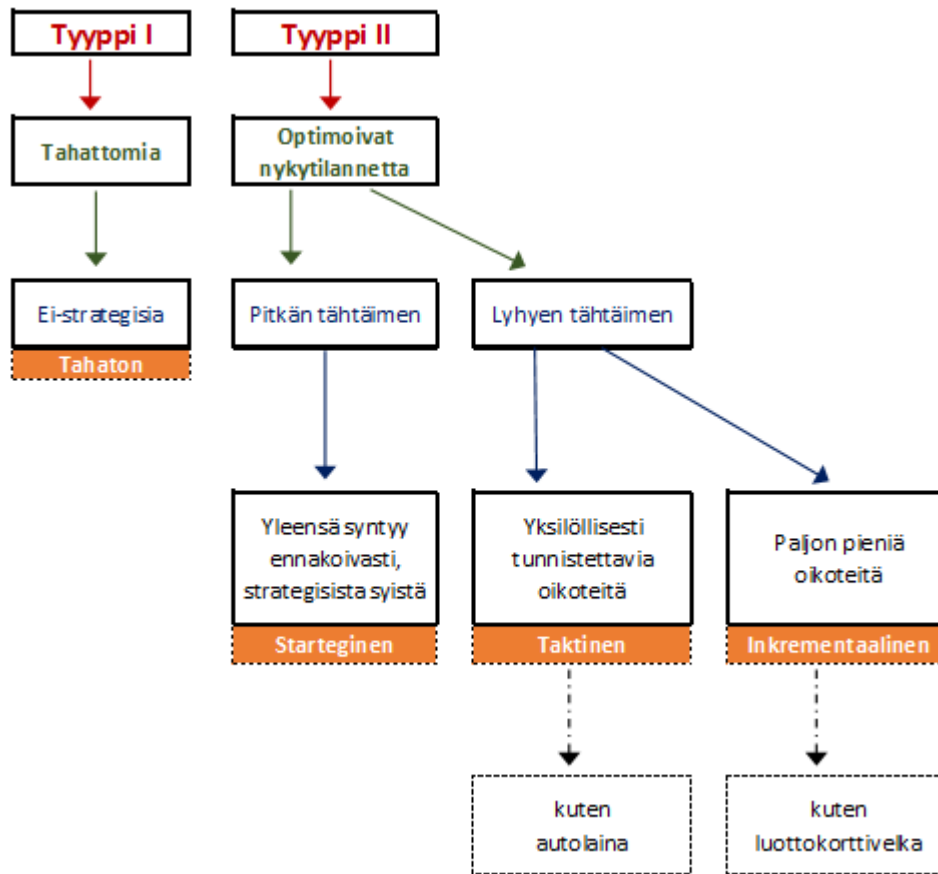
KUVIO 2 Teknisen velan puitteet (Kruchten ym., 2013, 52)

Yhteenvedon voidaan todeta, että teknisen velan määritelmiä on useita ja osittain ne ovat eri kohderyhmille suunnattuja. Kaikissa näissä on kuitenkin pyrkimyksenä tuottamaan yhteinen ymmärrys ja kieli, jotta erilaisen taustan omaavat henkilöt voivat keskustella samalla kielellä, samasta asiasta. Yhteistä kieltä ja kommunikointia tarvitaan, kun tehdään päätöksiä teknisen velan suhteen. Yleistäen voitaisiin sanoa, että teknistä velkaa on kaikki ne järjestelmään liittyvät asiat, jotka heikentävät tai huonontavat järjestelmän kehittämistä ja ylläpitoa. Seuraavassa alaluvussa jatketaan teknisen velan tarkastelua ja kuinka sitä voidaan määritellä luokittelujen kautta.

2.2 Luokittelut

Mitä tarkemmalle tasolle teknisen velan määrittelyssä mennään, sitä enemmän tarvitaan kuvauksia, rajoituksia ja sääntöjä. Teknistä velkaa voidaan jakaa, luokitella ja rajata monin eri tavoin, perustuen ominaisuuksiin, syihin tai ilmenemispaiikkaan. Luokittelujen ja esimerkkien avulla on helpompi konkretisoida ja tuoda lähemmäs käytäntöä teknistä käsitteistöä ja määritelmiä. Tällaiset kuvat ja listat ovat yksi tutkijoiden ja ohjelmistoammattilaisten välistä keskustelua helpottava asia (Li ym., 2015.) McConnell (2007) jakoi teknisen velan kahteen perustyyppiin. Tyypin I tekniset velat ovat tahattomia ja ei-strategisia. Tyypin II tekniset velat taas optimoivat nykytilannetta ja ovat joko lyhyen tai pitkän tähtäimen velkoja. Tom, Aurum ja Vidgen (2013) ottivat pohjaksi McConnellin teknisen velan jaon kahteen perustyyppiin ja selkeyttääkseen jakoa uudelleen nimesivät luokittelut, yhdistettynä kuviossa (kuvio 3). He säilyttivät Tyypin I tekniset velat tahattomina, mutta uudelleen nimesivät Tyypin II tekniset velat niiden ominaispiirteiden mukaan. Pitkän tähtäimen tekniset velat nimettiin Strategisiksi veloiksi. Lyhyen tähtäimen, yksilöllisesti tunnistettavia oikoteitä

käyttävät tekniset velat nimettiin Taktisiksi veloiksi. Lyhyen tähtäimen, paljon pieniä oikoteitä sisältävät tekniset velat nimettiin inkrementaaliksi veloiksi. (Tom ym., 2013.)



KUVIO 3 Yhdistelmä teknisen velan kahdesta perustyyppistä eri määrittelemien mukaan

Toisaalta Fowlerin (2009) mukaan tekninen velka voidaan jakaa neljään erilaiseen tyyppiin (ks. kuvio 4). Ensimmäinen tyyppi on "tietoinen – uhkarohkea", tässä otetaan tietoinen mutta hallitsematon riski. Toinen tyyppi on "tietoinen – viisas", tässäkin otetaan tietoinen riski, mutta se myös hallinnoidaan ja hoidetaan. Kolmantena tyyppinä on "tahaton – uhkarohkea", tämä syntyy tahattomasti ja tietämättömyydestä/osaamattomuudesta. Neljäs tyyppi on "tahaton – viisas", tämäkin syntyy tahattomasti, mutta tästä otetaan opiksi ja myös hoidetaan se. (Fowler, 2009.)



KUVIO 4 Teknisen velan jako tietoisiin ja tahattomiin (Fowler, 2009)

Siirryttäessä konkreettisempiin teknisen velan yksiköihin, voidaan niitä luokitella esimerkiksi velan sijainnin mukaisesti. Li ym. (2015) käyttivät teknisen velan luokittelussaan velan sijaintia jakaessaan teknisen velan karkeasti kymmenen erilaiseen tyyppiin. Nämä teknisen velan kymmenen eri tyyppiä ovat:

1. vaatimusperustaiset
2. arkkitehtuuriset
3. suunnittelu
4. koodaus
5. testaus
6. julkaisu
7. dokumentaatio
8. infrastruktuuri
9. versioitumiseen liittyvät
10. viat.

Li ym. (2015) mukaan jokainen näistä tyypeistä voidaan vielä jakaa edelleen useaan alatyyppeihin perustuen teknisen velan syntysyihin. Jos tyyppille ei ole määritelty erityistä syntysyytä, sen alatyyppejä on silloin "ei määritelty". Muita alatyyppejä ovat esimerkiksi "hyvien käytänteiden rikkomukset" ja "kompleksisuus". Teknisten velkojen ulkopuolelle rajattiin viat, toteuttamattomat ominaisuudet, tukevien prosessien puute, keskeneräiset tehtävät kehitysprosessissa, triviaalit koodin laatukysymykset sekä ohjelmistojen alhainen ulkoinen laatu. (Li ym., 2015.) Tutkimuksessaan Li ym. (2015) havaitsivat myös sen kuinka jakautunut tutkijakunta on edelleen sen suhteen ovatko viat teknistä velkaa vai eivät. Tämä näkyy myös heidän tekemissään "teknisen velan tyyppi" ja "ei tekninen velka" listauksissa. Huomionarvoista on se, että viat esiintyvät molemmissa listauksissa.

Yhteenvedon sanottakoon, että teknistä velkaa voidaan jakaa, luokitella ja rajata esimerkiksi perustuen ominaisuuksiin, syihin tai ilmenemiskohtaan. Tällaisilla luokittelulla pyritään konkretisoimaan teknisen velan käsitteistöä ja

määritelmiä. Tekniset velat voidaan jakaa esimerkiksi tahattomiin, strategiisiin, taktisiin ja inkrementaalisiin velkoihin. Seuraavassa alaluvussa jatketaan teknisen velan tarkastelua velan ominaisuuksien kautta.

2.3 Ominaisuudet

Sekä käsitteen määrittelyssä että kuvatessa erityyppisiä teknisiä velkoja, on hyödyllistä viitata useisiin teknisen velan ominaisuuksiin. Vaikka täydellinen lista teknisen velan ominaisuuksista on vielä tutkimuksen alla, useat velan ominaisuudet ovat jo selkiytymässä. Tutkimuksen tarkoituksena tällä alueella on ollut löytää keinoja taata, että tekninen velka saavuttaa riittävän näkyvyyden, jotta se huomioidaan järjestelmätason päätöksentekoprosesseissa. (Brown, ym., 2010.) Brown ym. (2010) listasivat seitsemän tekniseen velkaan liittyvää ominaisuutta. Nämä ominaisuudet ovat näkyvyys, arvo, nykyarvo, velan lisääntyminen, ympäristö, velan alkuperä sekä velan vaikutukset. Myös Tom ym. (2013) keräsivät tekniseen velkaan liittyviä ominaisuuksia, joita ovat rahallinen kustannus, anteeksianto, konkurssi, korko, pääoma, vipuvoima, lyhennykset ja nostot. Seuraavaan taulukkoon (taulukko 1) on yhdistetty ominaisuuksia ja huomioita edellä mainituista tutkimuksista. Tiivistettynä taulukon huomioista voidaan sanoa, että velan tulee olla näkyvää, täytyy omata tarkka kuva velan hinnasta ja sen vaikutuksista ja velkaa täytyy hallinnoida. Tekninen velka ei välttämättä ole vain huono asia, vaan sitä voidaan käyttää myös strategisena tai taktisena välineenä sekä arvon luomisessa. (Brown, ym., 2010., Tom ym., 2013.)

TAULUKKO 1 Yhdistelmä teknisen velan ominaisuuksista sekä huomioita näistä

Ominaisuus	Huomioita
Näkyvyys (Visibility)	Huomattavia ongelmia syntyy silloin, kun velka ei ole näkyvä. Monissa tapauksissa se on joidenkin ihmisten tiedossa, mutta se ei näy riittävästi niille, jotka lopulta joutuvat maksamaan velkaa.
Arvo (Value)	Tekninen velka, kun se hallinnoidaan oikein, on taloudellisessa käytössä työkalu, jolla luodaan arvoa. Arvo on oletetun ympäristön nykyisen järjestelmän ja ideaalitilassa olevan järjestelmän taloudellinen ero.
Nykyarvo (Present value)	Teknisen velan mahdollistaman kokonaistuottopotentialin lisäksi velasta aiheutuneiden kustannusten nykyarvo, mukaan lukien "time-to-impact" ja vaikutusten epävarmuus, on sijoitettava yleiseen kustannushyötyanalyysiin.
Velan lisääntyminen (Debt accretion)	Velka ei välttämättä lisäännä yhdistyessä, mutta yli-lisääntyy siinä mielessä, että otettaessa liian paljon velkaa johtaa se järjestelmän huonoon, ehkä korjaamattomaan tilaan.
Ympäristö (Environment)	Ohjelmistojen suunnitteluprojekteissa velka on suhteessa tiettyyn tai oletettuun ympäristöön.
Velan alkuperä (Origin of debt)	On tärkeää erottaa jyrkästi strateginen velka ja tahaton velka. Strateginen velka otetaan jonkin edun takia. Tahaton velka johtuu huo-

	noista käytänteistä tai ympäristön muuttumisesta kesken kaiken siten, että se luo epäkohtia, jotka vähentävät järjestelmän arvoa.
Velan vaikutukset (Impact of debt)	Velan sijainti on tärkeää: ovatko elementit, joita täytyy muuttaa velkaa maksettaessa paikallisia vai hyvin hajallaan?
Rahallinen kustannus (Monetary cost)	Tekninen velka vaikuttaa lopulta kielteisesti moraaliin, tuottavuuteen, laatuun ja riskeihin.
Anteeksianto (Amnesty)	Vaikka velan anteeksiantaminen on yksi mahdollisuus, sitä ei pidä käyttää tekosyynä kerryttää teknistä velkaa piittaamattomasti.
Konkurssi (Bankruptcy)	Tilanne, jossa velan korko on kasvanut ylivoimaiseksi, edistyminen on pysähtynyt ja ohjelmiston uudelleen kirjoittaminen kokonaan on tarpeen.
Korko ja Pääoma (Interest and principal)	Korkomenot ja pääoman lyhennykset. Teknisen velan korkomenot liittyvät velan vaikutukseen moraaliin, tuottavuuteen ja laatuun.
Vipuvoima (Leverage)	Tekninen velka on usein luotu vipuvoimaksi ja se voi lisätä tuottavuutta lyhyellä aikavälillä. Voidaan strategisesti tai taktisesti käyttää oikoteitä ja "lainatun ajan" vastineeksi kerryttää teknistä velkaa.
Lyhennykset ja Nostot (Repayment and withdrawal)	Paljon pieniä (satoja, tuhansia) velkoja, joille on tehtävä lyhennyksiä tai maksettava pois. Osa huonosti näkyviä, mikä vaikeuttaa tunnistamaan ne velat, jotka tulisi maksaa ensin. Verrattavissa luotokorttiosastoihin.

Tekninen velka on pohjimmiltaan tasapainoilua ohjelmistojen laadun ja liike-toiminnan todellisuuden kanssa. Toisaalta se on prosessien, laajuuden, testauksen ja suunnittelun kompromisseja. (Lim ym., 2012.)

Yhteenvetona voidaan todeta, että negatiivisten vaikutusten lisäksi teknisellä velalla voidaan saavuttaa myös merkittäviä hyötyjä, kuten esimerkiksi kilpailuetua. Tarkasteltaessa teknistä velkaa sen ominaisuuksien kautta voidaan nähdä, että käytettävä käsitteistö vielä enemmän talouspuolella käytettävää käsitteistöä. Tämä taas helpottaa teknisten ja ei-teknisten henkilöiden välistä kommunikointia. Seuraavassa alaluvussa tarkastellaan teknisen velan hinnan muodostumista.

2.4 Hinta

Tekninen velka ei oikeastaan ole kehittäjän ongelma; se on koko yrityksen laajuinen ongelma. Jos yrityksellä on liikaa teknistä velkaa, niin ääritapauksessa, kokonaiset suunnitteluosastot voivat pysähtyä. (Wolff & Johann, 2015.) Toisaalta tekninen velka on arvokas valuutta tasapainottamaan kehityshankkeen liike-toiminnallisia ja organisatorisia tekijöitä ja sen teknisten näkökohtien kanssa maksimoimaan taloudellista menestystä (Buschmann, 2011). Ymmärtämällä, viestimällä ja teknisen velan hallinnoinnilla voi olla valtava merkitys järjestelmän sekä lyhyen että pitkän tähtäimen menestykseen (Allman, 2012). Liiketoiminnan näkökohdat ovat etusijalla, kun käsitellään teknistä velkaa, varsinkin jos velka on tahallinen. Tässä tilanteessa on mahdollista verrata velan ve-

roetuuksia asianomaisen liiketoiminnan ajurien kanssa, kuten velan huoltokustannukset ja velan poistamiskustannukset. (Buschmann, 2011.)

Teknisen velan hinnan arviointiin on eri tutkimuksissa tarjottu erilaisia menetelmiä. Ampatzoglou, Ampatzoglou, Chatzigeorgiou, ja Avgeriou (2015) ovat tutkineet millaisia lähestymistapoja on käytetty muissa tutkimuksissa teknisen velan hallinnointiin. Eniten käytettyjä ovat olleet salkunhoito (portfolio management), kustannus-hyöty-analyysi sekä arvo-analyysi (arvoperustainen analyysi). Ongelmaksi he näkevät sen, että näistä puuttuu johdonmukaisuus, eri tutkimuksissa on lähestymistapoja sovellettu eri tavoilla, ja joistakin tutkimuksista on puuttunut selkeä kytkeä taloudellisten ja ohjelmistosuunnittelun käsitteiden kesken. (Ampatzoglou ym., 2015.) Toisaalta juuri soveltaminen, tutkiminen ja sitä kautta kehittäminen olisivat tärkeitä tällä alueella, koska kuten Ampatzoglou ym. (2015) yllä totesivat, ei teknisen velan hallinnointiin ja hinnan muodostumisen arviointiin ole vielä kehitetty sille parhaiten soveltuvaa lähestymistapaa. Hinnan arviointimenetelmiksi on joissakin tutkimuksissa tarjottu myös erilaisia koodirivien määrään perustuvia arviointitapoja. Curtis ym. (2012) käyttivät seuraavan laista kaavaa:

$$\begin{aligned} & ((\text{vakavat rikkeet} \times 0,5) \times 1\text{h}) \times 75\$ + \\ & ((\text{keskivakavat rikkeet} \times 0,25) \times 1\text{h}) \times 75\$ + \\ & ((\text{matalat rikkeet} \times 0,1) \times 1\text{h}) \times 75\$ \end{aligned}$$

Curtis ym. (2012) oletuksena on, että erilaisista rikkeistä korjataan tietty prosenttiosuus, kuten vakavista rikkeistä 50 %. Lisäksi tiedossa on oltava jonkinlainen arvio erilaisten rikkeiden määrästä, esimerkiksi staattisten koodianalyysaattoreiden avulla saatuna. Korjaamisen aikayksikkönä käytetään yhtä tuntia ja tuntihintana 75.00\$. Kaavan laskutuloksena he saivat keskimääräiseksi hinnaksi 3.61\$ / koodirivi ja tällä kaavalla laskettuna tyyppillisessä ohjelmistossa on 361,000.00\$ edestä teknistä velkaa jokaista sataa tuhatta (100 000) koodiriviä kohti. Tosin he huomauttavat joidenkin asiantuntijoiden saaneen yhden koodirivin hinnaksi 10.00\$, ollen kuitenkin sitä mieltä, että perusarvion saamiseksi kyseistä 3.61\$ koodirivin hintaa voidaan käyttää, kunhan huomioidaan teknologiat ja käytetty ohjelmointikieli. (Curtis ym., 2012.)

Yhteenvetona voidaan todeta, että teknisen velan hinnan arviointi on vaikeaa, erityisesti jos kyseessä on tahaton kauan piilossa ollut velka. Velan hinnan ja vaikutusten arviointi on helpompaa, ja yleensä jo etukäteen tehty, jos kyseessä on tietoinen velka. Teknisen velan hinnan arviointiin on ehdotettu ja tutkittu erilaisia menetelmiä, mutta teknisen velan hinnan arviointiin ei ole vielä kehitetty omaa siihen hyvin soveltuvaa menetelmää. Seuraavassa alaluvussa siirrytään tarkastelemaan miksi ja kuinka teknistä velkaa syntyy.

2.5 Syntyminen

Jotta teknisen velan hallinnointi olisi mahdollista, on tärkeää tietää miten, milloin ja miksi teknistä velkaa syntyy. Hyvin usein tekninen velka voidaan havai-

ta jo suoraan järjestelmän rakenteen perusteella, esimerkiksi sen monimutkaisen rakenteen perusteella. Rakenteellinen monimutkaisuus taas syntyy tahattomasti usein silloin, kun järjestelmän osien riippuvuuksien määrä kasvaa ja suunnittelutavoitteita rikotaan. Tällaiset monimutkaistumiset tekevät järjestelmistä jäykkiä (vaikea muuttaa), hauraita (kukin muutos rikkoo jotain muuta), viskooseja (on vaikeampaa tehdä asiat oikein) ja läpinäkymättömpiä (vaikea ymmärtää). (Brown, ym., 2010.) Tällaisessa tapauksessa voidaan todeta, että teknistä velkaa on syntynyt. Siis pohjimmiltaan tekninen velka voi syntyä huonosta rakenteellisesta laadusta, jolloin se vaikuttaa liiketoimintaan sekä tietotekniikan kustannuksina että liiketoiminnan riskeinä (Curtis ym., 2012). Toisaalta karkeasti ottaen voidaan sanoa, että teknistä velkaa on hankittu, kun suunnittelijat ovat käyttäneet erilaisia oikoteitä, eivätkä ole käyttäneet sovittuja tai hyväksi havaittuja käytänteitä (Allman, 2012). Teknistä velkaa syntyy myös silloin kun käytetään yksinkertaista, mutta hidasta algoritmia sen oikean, mutta työläämmän sijaan (Allman, 2012).

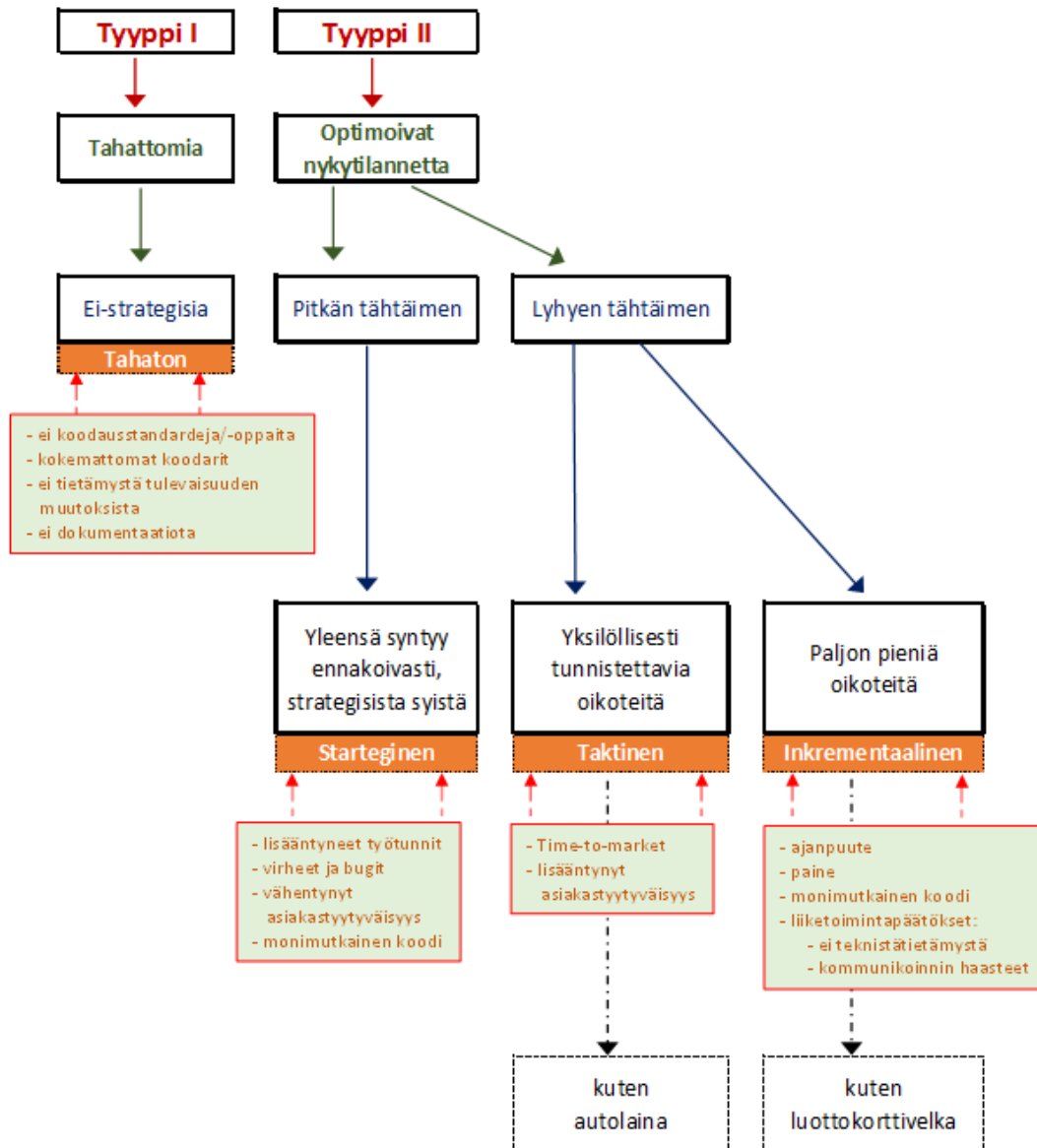
Heikko sisäinen koodin laatu on käytännössä eräänlaista teknistä velkaa, ehkä vallitseva sellainen. Työkalut, kuten staattiset koodianalysointit, auttavat tunnistamaan näitä heikosta sisäisestä laadusta johtuvia ongelmia ja niihin liittyviä dokumentaatio- ja testausongelmia. (Kruchten ym., 2013.) Teknistä velkaa syntyy myös seuraavista syistä (Allman, 2012):

- käytetään epämääräisiä käsitteitä
- käytetään hämääviä tai epätäydellisiä virheilmoituksia
- käytetään ei tyyppiturvallisia ratkaisuja
- käytetään järjestelmään sopimattomia käännoistyökaluja
- tingitään hyvistä tietoturvakäytännöistä
- jätetään yksikkötestit kirjoittamatta.

McGregor ym. (2012) mukaan ohjelmistojen kehittämisessä tehdään jatkuvasti päätöksiä, jotka vaikuttavat projektin tavoitteisiin, komponenttien kypsyyteen ja julkaisuaikatauluihin. He tarkentavat myös, että usein nämä päätökset edellyttävät useiden eri ominaisuuksien kompromisseja. Ohjelmistoissa tekninen velka voi aiheutua myös muutoksista ympäristötekijöissä, joihin kehitystiimin kontrolli ei ylety, ei vaikka olisi tehty hyviä päätöksiä. Jos järjestelmää ei kehitetä, niin uudet ympäristöolosuhteet voivat alkaa luoda korkeita korkomenoja. (Brown ym., 2010.) Yli-Huumo, Maglyas & Smolander (2014) toteavat omassa tutkimuksessaan vakaasti uskoen, että tekninen velka ei liity ainoastaan tekniisiin päätöksiin vaan myös johdon ja liiketoiminnan päätöksiin. Kuten myös, että päätös ottaa teknistä velkaa voi olla tietoinen, jotta voidaan toimittaa tuote nopeammin markkinoille. On todennäköisempää, että ohjelmistoprojekti hankkii teknistä velkaa ohjelmiston infrastruktuuriin kuin sen käyttöliittymään (Lim ym., 2012).

Yli-Huumo ym. (2014) ovat havainnoineet myös, ettei tekninen velka näytä liittyvän vain koodaukseen, jossa koodaaja käyttää oikoteitä lähdekoodissa. Heidän mukaansa tulokset osoittavat, että samanlaisia oikoteiden käyttämisiä tapahtuisi eri vaiheissa ohjelmistokehityksen elinkaarta, kuten vaatimus-, arkki-

tehtuuri- ja testausvaiheissa. Yli-Huumo ym. (2014) mukaan teknistä velkaa ei välttämättä aiheuta yksi erityinen syy, tosin ohjelmistoprojektin teknisen velan ensisijaiseksi syyksi he havaitsivat ajanpuutteen. Liiketoimintapäätökset taas olivat yksi kehitystiimin ajanpuutteen ja paineen yleisistä syistä (Yli-Huumo ym., 2014). Kuviossa (kuvio 5) on esitetty teknisen velan jakaminen kahteen perustyyppiin McConnell (2007) ja Tom ym. (2013) mukaan täydennettynä Yli-Huumo ym. (2014) tutkimuksessa havaittujen teknisen velan syiden ja seurausten suhteilla. He löysivät seuraavia syitä Tyypin I tahattomille veloille: koodausstandardien ja ohjeiden puuttuminen, kokemattomat koodaajat, puuttuva dokumentaatio sekä tietämättömyys tulevaisuuden muutoksista. Syitä Tyypin II erilaisille teknisille veloille olivat muun muassa ajanpuute, paine, virheet, monimutkainen koodi ja markkinoille tuloaika (time-to-market). (Yli-Huumo ym., 2014.)



KUVIO 5 Teknisen velan syitä ja seurauksia jaettuna kahteen perustyyppiin

Yhteenvedona voidaan todeta, että teknistä velkaa syntyy lukuisista eri sysistä ja usein syntymistä ei aiheuta vain yksi syy, vaan se on tulos monen eri syyn yhteisvaikutuksesta. Teknisen velan syntymistä edesauttavia asioita näyttäisivät olevan muun muassa kommunikoinnin haasteet, ajan puute, monimutkainen koodi sekä koodausstandardien ja -ohjeiden puute. Seuraavassa luvussa tarkastellaan teknisen velan hallinnointia ja mistä osista hallinnointi koostuu.

3 TEKNISEN VELAN HALLINNOINTI

Tässä luvussa tutkitaan mitä teknisen velan hallinnointi on ja kuinka sitä voidaan tehdä. Brown (2010) oli havainnut jo 2010, että teknisen velan käsite oli saamassa vetovoimaa keinona keskittyä lyhyen tähtäimen kompromissien luomien satunnaisten monimutkaisuuksien pitkän tähtäimen hallintaan. Hän oli tutkimuksessaan havainnut myös, että tällaisten velkojen tehokasta hallinnointia pidetään yleisesti kriittisenä yritettäessä saavuttaa ja ylläpitää ohjelmistojen laatua. Syynä tähän on, että hallinnoimaton velka luo merkittäviä pitkän tähtäimen ongelmia, kuten lisääntyneet ylläpitokustannukset. Kun teknisen velan metafora korostaa, että kuten rahoitusvelat, tekninenkin velka syntyy korkoina aikaisempien nopeiden, hätäisten ja vajavaisten suunnittelujen ja toteutuksien valintojen takia, näkyen lisääntyneinä tulevaisuuden kustannuksina. (Brown, ym., 2010.) Kuten taloudellinen velka, joskus tekninenkin velka voi olla tarpeellinen. Voi valita jatkaa koron maksua, tai maksaa pois pääomaa refaktoroimalla ja muokkaamalla arkkitehtuuria vähentääkseen tulevia korkomaksuja. (Brown, ym., 2010.)

Tämän kolmannen luvun ensimmäisessä alaluvussa käsitellään yleisesti teknisen velan hallinnointia ja sitä millaisista osista hallinnointi koostuu. Tämän jälkeen seuraavissa alaluvuissa käydään läpi näitä eri hallinnoinnin osa-alueita, aloittaen toisessa alaluvussa tunnistamisesta, ehkäisemisestä ja mittaamisesta. Kolmannessa alaluvussa käsitellään teknisen velan priorisointia, takaisinmaksua ja seurantaa. Neljännessä alaluvussa tarkastellaan kommunikaatiota ja dokumentointia. Viimeisessä viidennessä alaluvussa tarkastellaan vielä lopuksi hieman erilaisia strategioita ja työkaluja teknisen velan hallinnointiin liittyen. Seuraavaksi siis siirrytään tutkimaan mistä teknisen velan hallinnointi koostuu, mitä osa-alueita siihen kuuluu.

3.1 Hallinnoinnin osa-alueet

Teknisen velan hallinnointi (Technical Debt Management, TDM) sisältää toimia, joilla estetään mahdollisten uusien teknisten velkojen (sekä tahallisten että tahattomien) syntyminen. Siihen kuuluvat myös toimet, jotka käsittelevät jo kertynyttä teknistä velkaa, tehden sen näkyväksi ja hallittavaksi sekä toimet, joilla pidetään ohjelmistoprojektin kustannukset ja arvo tasapainossa. (Li ym., 2015.) Teknisen velan hallinnointi voidaan Li ym. (2015) mukaan jakaa kahdeksaan osa-alueeseen, jotka on järjestelty taulukoksi (taulukko 2). Heidän mukaansa teknisen velan hallinnointi on tunnistamista, mittaamista, priorisointia, seuranta, ehkäisemistä, dokumentointia ja viestintää.

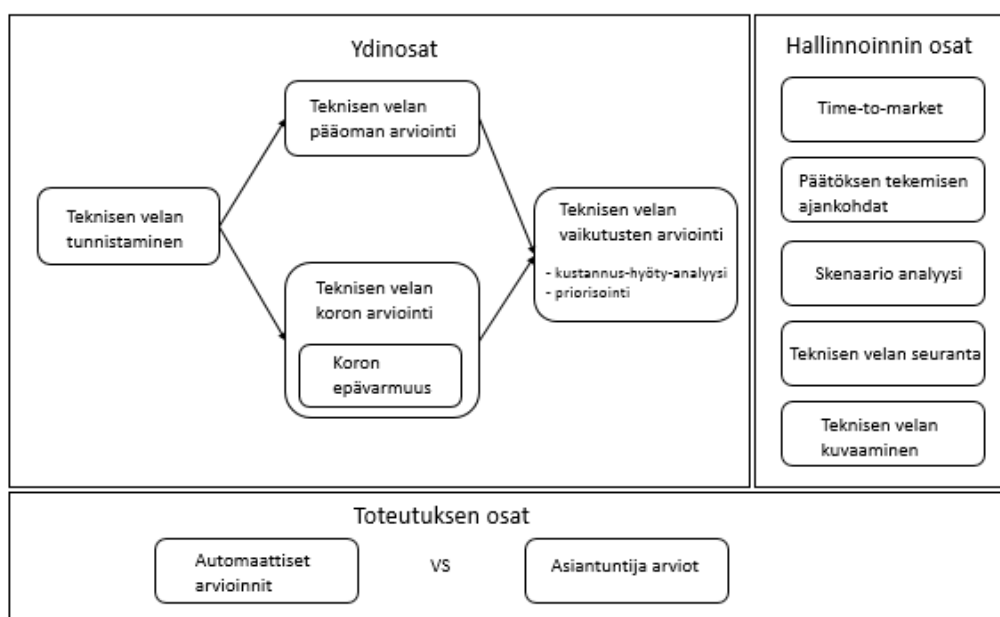
TAULUKKO 2 Teknisen velan hallinnoinnin jako kahdeksaan osaan

Hallinnoinnin osa	Selite
1 Tunnistaminen	➔ Havaitaan syntyneet velat
2 Ehkäiseminen	➔ Pyritään ehkäisemään uusien velkojen syntyminen
3 Mittaaminen	➔ Arvioidaan hyödyt / kustannukset
4 Priorisointi	➔ Ennalta määriteltyjen sääntöjen perusteella järjestellään velat
5 Takaisinmaksu	➔ Korjataan tai lievennetään teknisiä velkoja, esim. uudelleensuunnittelulla
6 Seuranta	➔ Seurataan hyötyjen ja kustannusten muutoksia ratkaisemattomien velkojen osalta
7 Kommunikointi	➔ Tehdään tunnistetut tekniset velat näkyviksi kaikille sidosryhmille, mahdollistaa keskustelun ja edelleen hallinnoinnin
8 Dokumentointi / esittäminen	➔ Yhdenmukainen tapa esittää ja yksilöidä tekniset velat

Fernandez-Sanchez, Garbajosa, Vidal ja Yague (2015) mukaan käytettävissä olevat teknisen velan hallinnointitekniikat ovat kasvattaneet erilaisten tekniikoiden hajontaa. Mistä johtuen johtajat kaipaavatkin nyt ohjausta siitä kuinka rakentaa sellainen tekniikoiden kokoelma, joka huomioi organisaation tavoitteet, liiketoimintastrategiat, aikataulut, riskitekijät, taloudelliset rajoitteet ja verotukselliset näkökohdat ja lisäksi mahdollistaisi teknisen velan hallinnoinnin. Fernandez-Sanchez ym. (2015) toivat esille tutkimuksessaan, että teknisen velan hallinnoinnin elementit edustavat niitä konkreettisia vaatimuksia, jotka tulee saavuttaa, jotta teknistä velkaa voidaan hallinnoida. He tunnistivat kolme ryhmää elementtejä. Nämä ovat:

1. Ydinkohdat: keskittyvät arvioimaan järjestelmien teknistä velkaa.
2. Hallinnoinnin osat: yksilöivät ne asiat, joita konkreettisissa projekteissa teknistä velkaa hallinnoitaessa tarvitaan.
3. Täytäntöönpanon osat: osoittavat, kuinka nämä elementit on toteutettu.

Fernandez-Sanchez ym. (2015) jakavat hallinnoinnin osa-alueen seuraaviin osiin: markkinoille tuloaika (time-to-market), teknistä velkaa koskevien päätösten tekoajankohdat, skenaarioanalyysi, teknisen velan seuranta sekä teknisen velan kuvaaminen. Ydinosat he jakavat tunnistamiseen, velan arviointiin (mittaaminen) sekä vaikutusten arviointiin (priorisointi). Toteutuksen osat he jakavat automaattiseen arviointiin ja asiantuntija arvioihin. Fernandez-Sanchez ym. (2015, 23) näkemyksen mukainen malli teknisen velan hallinnoinnista esitetään seuraavassa kuviossa (kuvio 6). Niin Li ym. (2015) kuin Fernandez-Sanchez ym. (2015) näkemykset teknisen velan hallinnoinnin osa-alueista ovat pitkälti hyvin samantyyppiset, vaikkakin hieman eri tavoin jaettuina ja osin eri tavalla nimettyinä.



KUVIO 6 Fernandez-Sanchez ym. (2015, 23) malli teknisen velan hallinnoinnista

Brownin ja muiden (2010) mukaan ketterät käytänteet (refaktorointi, testivetoinen ohjelmakehitys (TDD) ja iteraatioiden hallinta) yhdessä intuitiivisen teknisen velan käsityksen kanssa koetaan riittävänä hallita teknistä velkaa pienimuotoisissa hankkeissa. Kenties tästä riittävyyden tunteesta johtuen ei ketteriin käytänteisiin ole tehty teknisen velan tiukempaa konseptin määritelmää ja validointia tai heuristista käytänteitä, joita sen hallinnointi suuremmassa mittakaavassa sitten käytännössä tarkoittaisi. Ketterillä ohjelmistokehitysmenetelmillä on ollut merkittävä vaikutus ohjelmistokehityksen käytäntöihin muutamien viime vuosien aikana. Samalla vaikka ketterien tekniikoiden skaalaus on saanut vetovoimaa, eivät taas teknisen velan hallinnointitekniikat ole saaneet samantyyppistä huomiota. Vaarana tässä on, että tarve saada teknisen velan hallinnointi-ongelmaan tehokkaita ratkaisuja, johtaa siihen, että hyväksytään intuitiivisesti houkutteleva, mutta vain osittain sopiva heuristiikka. (Brown ym., 2010.)

Allman (2012) toteaa, että teknisen velan hallinnoinnissa on paljon samaa kuin riskienhallinnassa, joten vastaavia tekniikoita voidaan soveltaa. Jos teknistä velkaa ei hallinnoida, niin se yleensä ajan mittaan kasvaa asteittain, kunnes

mahdollisesti saavutetaan kriisitilanne. Teknistä velkaa voidaan tarkastella monella eri tavalla ja sitä voi syntyä kaikilla organisaation tasoilla. Sitä voidaan hallinnoida kunnolla vain kaikkien tasojen tuella ja ymmärryksellä. Erityisen tärkeää on auttaa ei-tekniisiä osapuolia ymmärtämään kustannuksia, joita voi syntyä, jos velka jätetään hallinnoimatta. (Allman, 2012.) Vähän teknistä velkaa ei ehkä ole ongelma, mutta siitä tulee ongelma, kun velkaa on "liian paljon". Tämä merkitsee sitä, että täytyy olla jotain sääntöjä siitä, miltä "liika" velka näyttää kuten hyväksyttävyyssraja. (Brown ym., 2010.) Snipes, Robinson, Guo ja Seaman (2012) tuovat taas tutkimuksessaan esille, että kustannushyötyanalyysin liittäminen päätöksentekoprosessiin voisi johtaa parempiin päätöksiin kuten tuotteen kannattavuuteen pitkällä aikavälillä sekä asiakastyytyväisyyteen. He myös kehittivät kustannushyötyanalyysiä niin, että se sisältää tekniseen velkaan liittyviä tunnistettuja kustannustekijöitä. Teknisen velan hallinnointi on tarpeen, jotta velka voidaan pitää hallinnassa. Teknisen velan hallinnointi muodostuu lisäkustannusten lähteen selvittämisestä silloin kun ohjelmistoa muutetaan ja analysoidaan, jotta nähdään milloin on kannattavaa sijoittaa panoksia (investoida panoksia) ohjelmiston parantamiseen. Käytännössä hyvä teknisen velan hallinnointi tarkoittaa, että projekteilla on teknistä velkaa sellainen määrä, joka ei estä organisaatiota saavuttamasta sen liiketoiminnallisia tavoitteita. (Fernandez-Sanchez ym., 2015.)

Tyypillisesti tiukat ylläpidon budjetit ja aikataulut väistämättä aiheuttavat muutoksia, jotka eivät täysin ota laatuvaatimuksia huomioon. Käyttämällä teknisen velan metaforaa selittämään tätä ilmiötä, on käyttäjien ja johtajien mahdollista huomioida se suunnittelussa ja päätöksenteossa. (Seaman & Guo, 2011.) Tekniset velat ovat usein seurausta korkean tason teknisten ja ei-tekniisten johtajien päätöksistä, ja teknisen johtajuuden ja johtamisen asenteet vaikuttavat voimakkaasti päätöksiin ottaa teknistä velkaa (Lim ym., 2012). Teknisten velkojen tehokasta hallinnointia taas pidetään kriittisenä yritettäessä saavuttaa ja ylläpitää ohjelmistojen laatua. Hallinnoimaton velka luo merkittäviä pitkän aikavälin ongelmia, kuten lisääntyneet ylläpitokustannukset. (Brown, ym., 2010.)

Yhteenvedona voidaan todeta, että teknisen velan hallinnointi voidaan jakaa useammalla eri tavalla, mutta jakotavasta ja nimeämisestä riippumatta sieltä löytyy samoja toimenpiteitä. Hallinnointiin kuuluu toimia, joilla tunnistetaan, ehkäistään ja rajoitetaan, mitataan, priorisoidaan, hoidetaan takaisinmaksua, seurataan, viestitään ja dokumentoidaan teknistä velkaa. Ensimmäinen edellytys teknisen velan hallinnoinnille on tunnustaminen, teknisestä velasta vapaata järjestelmää ei ole. Seuraavissa alaluvuissa tutkitaan tarkemmin teknisen velan hallinnoinnin osa-alueita. Ensimmäisenä tarkastellaan teknisen velan tunnistamista, ehkäisemistä ja mittaamista.

3.2 Tunnistaminen, ehkäiseminen ja mittaaminen

Ensimmäinen askel teknisen velan hallinnointiin on erilaisten teknisten velkojen tunnistaminen. Et voi hallinnoida tekniisiä velkaa ellet tunnista sitä sekä

kommunikoi niiden kanssa, jotka voivat tehdä päätöksiä tekniseen velkaan liittyen. (Lim ym. 2012) Guo ja Seaman (2011) ovat tutkineet miten erilaisia teknisiä velkoja voidaan tunnistaa ja he ovat listanneet seuraavat tavat:

1. Suunnitteluvelka voidaan tunnistaa staattisesti, Eisenberg (2012) on lisännyt vielä dynaamisen tunnistamisen, analysoimalla lähdekoodia.
2. Testausvelka voidaan tunnistaa vertaamalla testisuunnitelmia ja testituloksia. Suunnitellut, mutta ajamattomat testit ovat testausvelkaa.
3. Vikavelka voidaan tunnistaa vertaamalla testituloksia muutosraportteihin. Viat, jotka on todettu, mutta ei korjattu, ovat vikavelkaa. Toisaalta Kruchten ym. (2013) ovat rajanneet itse viat teknisen velan ulkopuolelle.
4. Vertaamalla muutosraportteja dokumentaation versiohistoriaan, dokumentaatiovelka voidaan tunnistaa. Jos muutoksia on tehty päivittämättä dokumentaatiota, on kyseinen päivittämätön dokumentaatio dokumentointivelkaa.

Koodin analysointiin on tarjolla paljon erilaisia työkaluja. Analysoinnissa tulisi-kin käyttää useampaa välinettä, sillä työkalut osoittavat erilaisiin koodin ongelmakohtiin. Yhden työkalun käyttö ei yleensä ole riittävä osoittamaan projektin kannalta kaikkia tärkeitä teknisen velan ongelmakohtia. (Zazworka ym., 2014.) Useat analysointityökalut sisältävät myös jonkinlaisen mittaamisen työkalun tai menetelmän, jonka avulla järjestelmän teknisen velan määrää voidaan mitata. Saatavilla on myös erikseen hyödynnettäviä menetelmiä. Eniten käytetty lienee SQALE (software quality assessment based on life-cycle expectations), joka löytyy myös joistakin analysointityökaluista plugin versiona (Letouzey & Ilkiewicz, 2012). Teknisen velan tunnistamisessa myös kehittäjät ovat avainasemassa. Kehittäjillä on usein jo tiedossa olevia ongelmakohtia, jotka usein ovat myös sellaisia, joita työkalut eivät löydä, kuten esimerkiksi useaan paikkaan tallennettu tieto tai muistinkäyttö ongelmat (Zazworka, Spinola, Vetro, Shull & Seaman, 2013). Teknisen velan yleisiin syntyminen syihin vastaamalla voidaan rajoittaa ainakin jonkin verran syntyvien velkojen määrää. Tällaisia ehkäisemisen ja rajoittamisen menetelmiä voisivat siten olla esimerkiksi koodausstandardien ja -ohjeiden kirjoittaminen, parikoodaus (kokematon-kokenut), dokumentointi ja koodin refaktorointi (Yli-Huumo ym., 2014).

Teknisen velan olemassaolo vaikeuttaa ohjelmistojen hallintaa, koska johtajien on päätettävä maksetaanko velkaa pois, kuinka paljon ja milloin. Jotta voidaan tehdä perusteltuja päätöksiä, on välttämätöntä saada perusteellinen tietämys nykyiselle ja tulevalle arvon määrälle, joka projektilla tällä hetkellä teknistä velkaa on. Siksi tunnistaminen, mittaaminen ja teknisen velan seuranta auttaisi johtajia tekemään perusteltuja päätöksiä, jolloin saataisiin laadukkaampia ylläpidettyjä ohjelmistoja ja suurempi ylläpidon tuottavuus. (Seaman & Guo, 2011.) Teknistä velkaa voidaan pitää tietyn tyyppisenä riskinä ohjelmistojen ylläpidossa, koska sillä on riskien peruselementit: mahdollinen tappio / rangaistus ja siihen liittyvä epävarmuus. Siksi teknisen velan mittaamisen ja hallinnan ongelma liittyy läheisesti riskienhallintaan sekä perusteltujen päätösten tekemiseen, jotka liittyvät siihen mitkä viivästytyt tehtävät tarvitsee suo-

rittää loppuun ja milloin. (Seaman & Guo, 2011.) Avain teknisen velan kuvantamiselle, ei niinkään ole mitata sitä pala palalta, kun se on toteutunut, vaan mitata sen kumulatiivista vaikutusta ajan suhteen (Lim ym., 2012).

Yhteenvedona voidaan todeta, että teknisen velan tunnistamisen ja mitaamisen avuksi, erityisesti koodin, löytyy useita työkaluja. Tosin myös kehittäjien tietämystä tulee teknisen velan tunnistamisessa hyödyntää. Teknisen velan ehkäisemisen ja rajoittamisen toimenpiteitä voisivat olla esimerkiksi koodausstandardien ja -ohjeiden kirjoittaminen sekä monimutkaisten koodien refaktorointi. Seuraavassa alaluvussa tarkastellaan teknisen velan priorisointia, takaisinmaksua ja seuranta.

3.3 Priorisointi, takaisinmaksu ja seuranta

Teknisen velan metaforaa on pidetty hyödyllisenä kommunikoitaessa osa-optimaalisten ratkaisujen riskeistä teknisten ja ei-teknisten sidosryhmien välillä. On kuitenkin olennaista ymmärtää asiaan liittyvien sidosryhmien tietotarpeet, jotta voidaan tuottaa teknisen velan mittauksia, jotka sallivat asianmukaisen viestinnän ja tietoihin perustuvan priorisoinnin. (Martini & Bosch, 2015.)

Martini ja Bosch (2015) ovat tutkineet mitkä ovat ne keskeiset tiedot, joita ketterien projektien tuotteen omistajat ja ohjelmistojen arkkitehdit tarvitsevat voidakseen priorisoida riskialttiita arkkitehtuurisia teknisiä velkoja suhteessa ominaisuuksien kehittämiseen. He määrittelivät ne priorisoinnin näkökohdat, joita voitiin käyttää vertailupainoarvoina arkkitehtien ja tuotteenomistajien välillä. Kuten myös saatavilla olevat tiedot arkkitehtuuristen teknisten velkojen vaikutuksista, joita voidaan käyttää priorisoinnin aikana. Priorisoinnin näkökohdat Martini ja Bosch (2015) mukaan ovat:

1. kilpailuetu
2. pitkäaikainen asiakastyytyväisyys
3. läpimenoaika
4. ylläpitokustannukset
5. vetovoima markkinoille
6. rangaistus
7. riskit
8. arvo tietylle asiakkaalle
9. epävakaus / vaihtelevuus.

Lim ym. (2012) tutkimuksen mukaan riskienhallinnan lähestymistapaa voisi käyttää arvioimaan ja priorisoimaan teknisen velan kustannuksia ja arvoa. He myös ehdottavat 5 – 10 % varaamista kustakin julkaisusyklistä teknisen velan käsittelyyn. Martin ja Bosch (2015) tutkimuksen mukaan luku voisi olla 10 - 20 % resursseista ja tämä osoitettuna nimenomaan arkkitehtuuriselle tekniselle velalle.

Arkkitehtuurisen teknisen velan priorisointiprosessissa käytettäviä tietoja ovat asiakkaalta saatavat tiedot sekä arkkitehtuurisen teknisen velan vaikutusten tiedot, kuten refaktorointikustannukset ja velan pääomakustannukset. Näiden tietojen perusteella ja priorisointinäkökohdat huomioiden arkkitehdit ja tuoteomistajat arvioivat ja tekevät päätöksiä arkkitehtuurisen teknisen velan suhteen. (Martini & Bosch, 2015.) Priorisoinnin näkökohdista läpimenoaika, ylläpitokustannukset ja riskit näyttäisivät hyötyvän paljon arkkitehtuurisen teknisen velan vaikutuksiin liittyvistä tiedoista. Edellisillä kolmella näkökohdalla on myös eniten merkitystä priorisointiprosessissa. (Martini & Bosch, 2015.) Toisin kuin tuoteomistajat, arkkitehdit näkevät arkkitehtuurisen teknisen velan vaikutusten tiedot enemmän hyödyllisenä priorisointiprosessissa, erityisesti silloin kun priorisointinäkökohtina ovat ylläpitokustannukset tai rangaistus (Martini & Bosch, 2015).

Kun projektiryhmä päättää maksaa takaisin teknistä velkaa, on arkkitehtien vastuulla auttaa valitsemaan sopiva toteutus. Nyrkkisääntönä on olla "minimaalisesti kajoava" - pidetään kustannukset alhaisina, mutta myös koska ei ole vain voitettavaa, vaan on myös hävittävää. (Buschmann, 2011.) Ihannetapauksessa, ohjelmistojen johtajat voivat valita ottavansa vain sellaista teknistä velkaa, joka ei kasva korkoa tai valita viivyttävänsä velan takaisin maksua keskittymällä sellaisiin teknisiin velkoihin, jolle on rangaistus. Vaikeutena kuitenkin on se, että harvoin on tiedossa etukäteen, onko tiettyssä osassa järjestelmää puutteita tai onko tiettyyn moduuliin koskaan tulossa muutoksia. (Seaman & Guo, 2011.)

Vaikka teknistä velkaa ei tarvitse maksaa takaisin millään kiinteällä aikataululla, yleensä se pitää maksaa. Korkoa tekniselle velalle kertyy aina kun joku järjestelmän kanssa työskentelevä viivästyy esimerkiksi virheiden, suorituskykyongelmien tai ylläpidettävän koodin takia. (Allman, 2012.) Buschmann (2011), kuten myös Allman (2012) sekä Curtis ym. (2012), mukaan teknisen velan hoitaminen voidaan jakaa kolmeen erilaiseen vaihtoehtoon. Ensimmäisenä on teknisen velan takaisinmaksu, jossa uudelleen kirjoitetaan tai korvataan todella huono koodin osa tai kokonaisuus vakaalla ja hyvällä suunnittelulla. Toisaalta tällaiseen tulisi lähteä vain siinä tapauksessa, että koodi on todella huonoa tai, että siihen joudutaan jatkuvasti lisäämään uusia toiminnallisuuksia. Toisena vaihtoehtona on teknisen velan muuntaminen. Kun jossain järjestelmän osassa on hyvin korkea tekninen velka, mutta järjestelmän osan vaihtaminen ei ole mahdollista, takaisinmaksu voi olla liian kallista ja riskialtista. Silloin voidaan yrittää muuntaa järjestelmän osaa parempaan vaan ei täydelliseen ratkaisuun, jossa on pienempi tekninen velka. Näin järjestelmän osa ei edelleenkään ole täydellinen, mutta paljon parempi kuin vanha. Uuden teknisen velan korko on paljon alhaisempi ja sopii siten paremmin myös talousarvioon. Viimeisenä, kolmantena vaihtoehtona on teknisen velan koron maksu. Joskus saavutettavissa olevat hyödyt eivät kata uudelleensuunnittelun liittyviä kustannuksia, jolloin voidaan hyväksyä tekninen velka ja vain jatkaa sen koron maksua. Tässä tapauksessa tiedetään, ettei koodi ole kovin hyvää, mutta liiketoiminnan kannalta se on halvin vaihtoehto.

Jos et voi välttää teknistä velkaa, sinun täytyy hallita sitä. Tämä tarkoittaa, että tunnustetaan, tunnistetaan, seurataan, tehdään perusteltuja päätöksiä liittyen tekniseen velkaan ja estetään sen pahimmat seuraukset (Lim ym., 2012). Tärkeää olisi myös analysoida kuinka erilaisia asiakkaisiin liittyviä tietoja, mm asiakasvaatimuksia, käytetään priorisoinnin aikana ja kuinka niitä verrataan teknisen velan vaikutuksiin. Näin ymmärretään kuinka teknisen velan takaisinmaksutarpeet voidaan priorisoida ominaisuuskehityksen edelle (Martini & Bosch, 2015). Teknisen velan seurannan avuksi Lim ym. (2012) ehdottavat Wikiä, tehtävälistaa (backlog) tai tehtävätaulua (task board).

Yhteenvedona voidaan sanoa, että teknisen velan priorisointi ja takaisinmaksu ovat asioita, jotka vaativat jatkuvaa ja avointa viestintää kaikkien asianosaisten kanssa. Haastavan asiasta tekee teknisten ja ei-teknisten osallisten kommunikointi teknisistä ja taloudellisista asioista, tähän hyvä työväline on teknisen velan metafora. Päätöksiä tehtäessä on myös tärkeää, että käytettävissä on ajantasainen ja riittävä tietämys velasta. Päätösten jälkeen on myös tärkeää seurata velkaa, sen takaisinmaksua, koron kehittymistä ja muita velan ominaisuuksia. Seuraavassa alaluvussa tarkastellaan teknisen velan viestintää ja dokumentointia.

3.4 Viestintä ja dokumentointi

Tekninen velka ei ole yhtä näkyvä ulkoisille sidosryhmille, kuten asiakkaille ja johdolle, kuin mitä se on kehittäjille, jotka työskentelevät koodin kanssa päivittäin (Lim ym., 2012). Tom ym. (2012) tutkimuksessa tulee esille se, että järjestelmän sisältämän teknisen velan laajuuden tunnistaminen, määrällistäminen ja tarkka viestintä tuovat merkittäviä haasteita. Lim ym. (2012) kehottavat hallitsemaan asiakkaiden ja ei-teknisten sidosryhmien odotuksia tekemällä heistä tasavertaisia kumppaneita sekä näin helpottamaan avointa vuoropuhelua teknisen velan vaikutuksista.

Yli-Huumo ym. (2014) tuovat ilmi, että tekninen velka on jotain, mitä yritykset eivät voi välttää ohjelmistokehitysprojektien aikana. Heidän tapaustutkimuksessaan tekninen velka syntyi eri johtotason päätöksien seurauksena, joita tehtiin hankkeen aikana, jotta saavutettaisiin määräajat tai tietämättään, tietämyksen puutteesta johtuen (Yli-Huumo ym., 2014). Heidän tutkimuksessaan tulee esille hyviä viestintärakenteita, joissa projektipäälliköt toimivat tehokkaina suodattimina yritysjohtajien ja kehitystiimien välillä teknistä velkaa koskevissa päätöksissä (Yli-Huumo ym., 2014). Lisäksi tulee ilmi, että tiedonantorakenteet ovat pienissä yrityksissä paljon parempia kuin suurissa yrityksissä, joissa mielipiteen saaminen teknisestä velasta on haastavampaa monimutkaisemman ja suuremman viestintärakenteen takia (Yli-Huumo ym., 2014). Yli-Huumo ym. (2014) mukaan yritysjohtajille ja liikkeenjohdon ihmisille oli vaikea ilmaista teknisiä mielipiteitensä ja tällä tavalla saada enemmän kehitysaikaa. Toisaalta kehitystiimi kuitenkin mainitaan usein tekniseen velkaan liittyvissä tilanteissa (Yli-Huumo ym., 2014). On olennaista ymmärtää mukana olevien sidosryhmien

tietotarpeita, jotta voidaan laatia teknisen velan mittauksia, joka sallivat asianmukaisen viestinnän ja tiedon priorisoinnin (Martini & Bosch, 2015). Teknisen velan tasapainottaminen vaatii, että ohjelmistoalan ammattilaiset esittävät teknisen velan täsmällisesti, kommunikoivat sen kustannukset ja hyödyt kaikille sidosryhmille sekä hallinnoivat väistämättömät kompromissit (Lim ym., 2012).

Yhteenvedona voidaan sanoa, että teknisen velan dokumentointi ja kommunikointi ovat erittäin tärkeitä osa-alueita, mutta samalla hanakalia toteuttaa. Onnistuneeseen kommunikointiin vaikuttavat niin organisaation viestintärakenteet kuin teknisen velan näkyvyys ja vaikutukset. Seuraavassa alaluvussa tarkastellaan tekniseen velkaan liittyviä strategioita ja työkaluja.

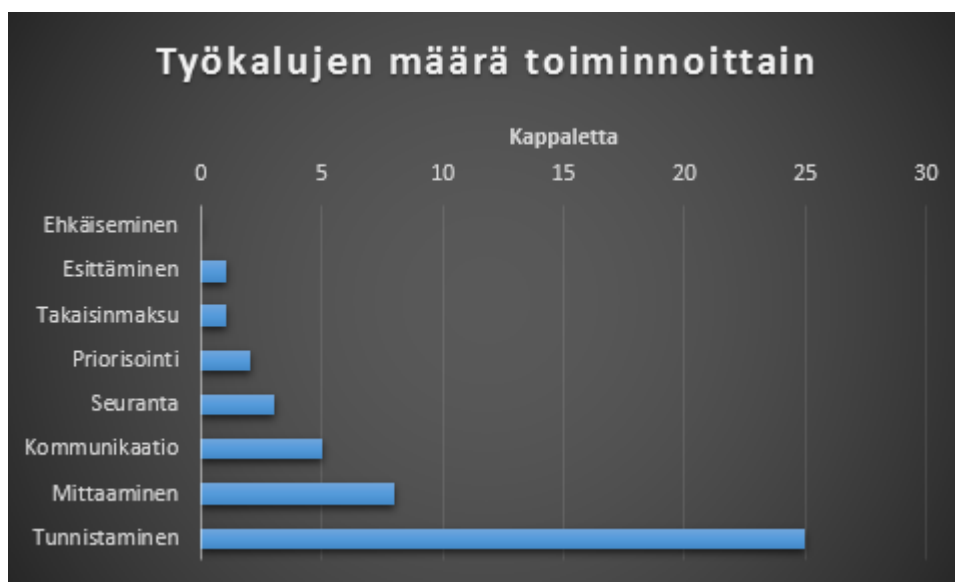
3.5 Strategiat ja työkalut

Tekninen velka on arvokas valuutta tasapainottamaan kehityshankkeen liike-toiminnallisia ja organisatorisia tekijöitä ja sen teknisten näkökohtien kanssa maksimoimaan taloudellista menestystä (Buschmann, 2011). Ymmärtämällä, viestimällä ja teknisen velan hallinnoinnilla voi olla valtava merkitys järjestelmän sekä lyhyen että pitkän tähtäimen menestykseen (Allman, 2012). Tekninen velka on usein tarkoituksella luotu vipuvoimaksi ja se voi lisätä tuottavuutta lyhyellä aikavälillä. Näin tiimit voivat käyttää strategisesti tai taktisesti oikoteitä ja lykätä työtä sekä kerätä teknistä velkaa vastineeksi "lainatusta ajasta" (Tom ym., 2013) tai strategiana on säästää aikaa tai rahaa (Allman, 2012). Tekninen velka sisältää usein myös uhkapeliä siitä, mitä tulevaisuudessa tapahtuu (Allman, 2012). Tyypillisesti regressiot ja virheet näkyvät pitkällä aikavälillä, lisääntyneen monimutkaisuuden seurauksena. Tahaton tekninen velka ei vaikuta tuottavuuteen lyhyellä aikavälillä, koska se ei tyypillisesti johda nousuun tai laskuun nopeudessa. Moraaliin taas ei ole vaikutuksia lyhyellä aikavälillä, koska tahaton tekninen velka saattaa löytyä ja realisoitua vasta useiden seuraavien iteraatioiden päästä. (Tom ym., 2013.) Kun velka on hallitsematonta, päästään optimointiongelmaan, jossa lyhyen tähtäimen optimoinnilla saatetaan talous ja tekniikka vaaraan pitkällä tähtäimellä. Suunnittelun oikotiet taas saattavat antaa käsityksen menestyksestä, kunnes niiden seuraukset alkavat hidastaa projekteja. Kehityspäätökset, erityisesti arkkitehtuuriset, taas tulee olla aktiivisesti hallinnoituja ja jatkuvasti määrällisesti analysoituja, koska ne aiheuttavat kustannuksia, arvoa ja velkaa. (Brown ym., 2010.)

Käsite velasta investoinnin tyyppinä, joka mahdollistaa kasvun kehityksen nopeasti lyhyellä aikavälillä, luo houkutuksen antaa velan kertyä. Tämä väistämättä hidastaa nopeutta ja kasvattaa markkinoille tuloaikaa (time-to-market) pitkällä aikavälillä, ellei sitä makseta takaisin. (Tom ym., 2013.) Buschmann (2011) mukaan yllättäen monet laskelmat osoittavat, että usein on parempi ottaa velkaa ja maksaa se myöhemmin pois. Hyvä asia teknisessä velassa on se, että se poistuu automaattisesti järjestelmän elinkaaren lopussa, toisin kuin taloudellinen velka. Teknisten näkökohtien pitää vaikuttaa, mutta ne eivät saa määrittää miten velkaa tulee käsitellä. Vaikka teknisen velan poismaksu on suotuisaa,

niin se ei aina tarjoa parasta arvoa. (Buschmann, 2011.) Teknisen velan kielteiset vaikutukset näyttävät osuvan pitkälle aikavälille. Nämä johtavat huonoon asiakaskäsitykseen ja heijastuvat siten asiakaspalveluun, jossa tarvitaan myöhemmin enemmän työtä ja kustannuksia, koska joudutaan käsittelemään teknistä velkaa asiakkaan tiloissa. (Lim ym., 2012.) Kun yksi ohjelmistokehitysorganisaatio tekee päätöksen ottaa teknistä velkaa saavuttaakseen tietyn tavoitteen, kuten tietyn toimituspäivän, ei päätös ole erillinen yksittäinen tapahtuma, se vaikuttaa myös muihin ekosysteemin jäseniin (McGregor ym., 2012).

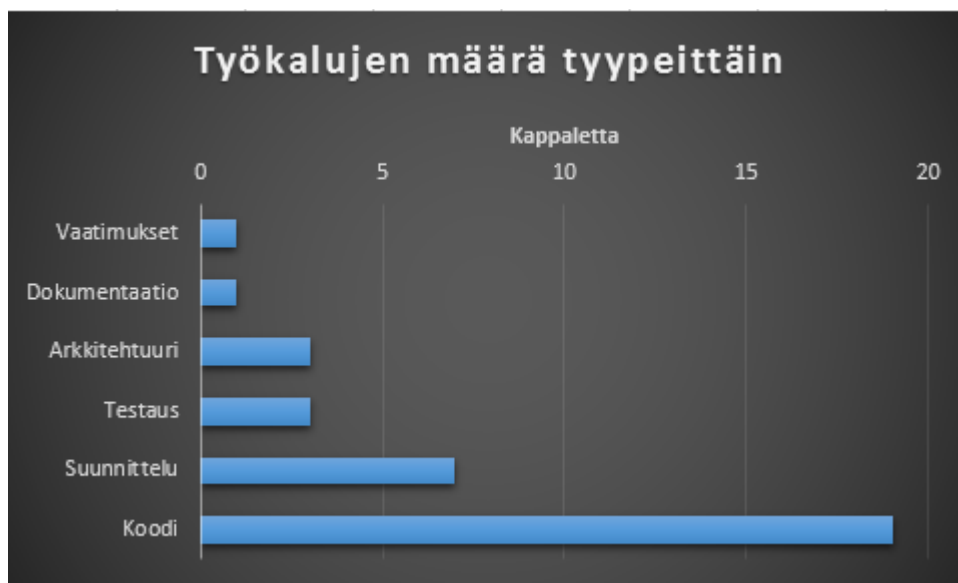
Jotta voidaan systemaattisesti hallinnoida teknistä velkaa, on syytä omata selkeä ja perusteellinen käsitys viimeisimmästä teknisen velan tilanteesta. Eri-laisia menetelmiä ja työkaluja on käytetty, ehdotettu ja kehitetty teknisen velan hallinnointiin, mutta on epäselvää kuinka näitä menetelmiä ja välineitä voidaan kytkeä teknisen velan hallinnoinnin toimintoihin. (Li ym., 2015.) Kuviossa (kuvio 7) on kuvattu Li ym. (2015) tutkimuksessaan tarkastelemien kahdenkymmenen yhdeksän työkalun jakautuminen teknisen velan hallinnointitoiminnoittain. Kuvioista on selvästi havaittavissa kuinka työkalut ovat painottuneet teknisen velan tunnistamiseen, kun taas esittämistä eli kuvaamista tai takaisinmaksua tukee vain yksi työkalu ja teknisen velan ehkäisemistä ei yksikään. Li ym. (2015) mukaan näistä kahdestakymmenestä yhdeksästä työkalusta vain neljä työkalua oli omistettu teknisen velan hallinnointiin, kun suurin osa kerätyistä työkaluista oli muilta aloilta lainattuja, kuten esimerkiksi koodin laadun analysointiin. He myös totesivat, että useimmat työkaluista käyttivät syötteenään lähdekoodia.



KUVIO 7 Teknisen velan hallinnointia tukevat työkalut toiminnoittain (Li ym., 2015, 211)

Kuviossa (kuvio 8) on kuvattu Li ym. (2015) tutkimuksessaan tarkastelemien kahdenkymmenen yhdeksän työkalun jakautuminen teknisen velan tyyppien kesken. Kuvioista on selvästi nähtävissä kuinka koodipainottuneita työkalut

ovat, kun taas teknisen velan vaatimus- ja dokumentaatiotyypeille tukea löytyy vain yhdestä työkalusta. Li ym. (2015) painottavat, että toisaalta jokainen näistä neljästä teknisen velan hallinnointiin omistetusta työkalusta tuki useampaa kuin kahta teknisen velan hallinnointitoimintoa. Kun taas muut työkalut tukivat vain yhtä tai kahta teknisen velan hallinnointitoimintoa. Sen sijaan useimmat työkaluista tukivat teknisen velan koodin ja suunnittelun hallinnointia, toisaalta harvat työkalut tukivat muita teknisen velan tyyppisiä (esimerkiksi arkkitehtuurista teknistä velkaa). (Li ym., 2015.) Ernst, Bellomo, Ozkaya, Nord ja Gorton (2015) taas havaitsivat tutkimuksessaan, että arkkitehtuurisen teknisen velan hallinnointiin käytettiin harvoin työkaluja. Heidän mukaansa työkalut olivat, joko hankalia käyttää ja asentaa tai eivät vastanneet tarpeita.



KUVIO 8 Teknisen velan hallinnointia tukevat työkalut tyypeittäin (Li ym., 2015, 211)

Yhteenvedona voidaan todeta, että teknistä velkaa voidaan käyttää myös strategisena tai taktisena välineenä organisaation lyhyen ja pitkän tähtäimen suunnittelussa. Tosin vain asianmukaisesti hallinnoidulla velalla voidaan saavuttaa tuloksia, hallinnoimaton velka voi johtaa mittaviin tappioihin. Teknisen velan hallinnoinnin tueksi on tarjolla vähänlaisesti työkaluja, varsinkin teknisen velan hallinnointiin erikoistuneita. Toisaalta koodin analysoinnin tueksi on tarjolla useita erilaisia työkaluja. Seuraava luku sisältää tutkimuksen yhteenvedon sekä pohdintaa teknisestä velasta.

4 YHTEENVETO JA POHDINTA

Tässä tutkielmassa oli tarkoituksena kirjallisuuteen perehtymällä tarkastella teknistä velkaa hallinnoinnin näkökulmasta. Tutkimuskysymyksenä oli Miten teknistä velkaa hallinnoidaan? Aiheeseen lähdettiin tutustumaan apututkimuskysymyksen Mitä tekninen velka on? kautta luvussa kaksi.

Teknisen velan määritelmiä on useita, mutta kaikissa niissä kuitenkin puhutaan yleisellä tasolla samasta asiasta, siis kaikkien niiden aiempien päätösten summana, joka vaikuttaa ohjelmiston tulevaisuuteen. Sen sijaan täysin tarkkaa yhteisymmärrystä ei ole siitä mitä kaikkea tarkemmalla tasolla lasketaan kuuluvaksi tekniseen velkaan ja mitä ei. Itse teknisen velan metafora on hyvä väline teknisen velan viestinnän apuvälineeksi keskusteltaessa eri sidosryhmien välillä. Se antaa myös varsin hyvät puitteet teknisen velan konkreettisten yksiköiden määrittelemiseksi. Lisäksi on huomattava, että jokin tietty tekninen velka voidaan joissakin tapauksissa luokitella kuuluvaksi moneen eri tyyppiin (arkkitehtuuri, vaatimus, suunnittelu, koodi) ja se voi olla syntynyt myös usean eri syyn seurauksena. Käytännön työssä ei aina kannata lähteä miettimään jokaisen yksittäisen yksityiskohdan kanssa kuuluuko se niin sanotusti määriteltyjen teknisten velkojen joukkoon, ja kenen määritelmän mukaan. Käyttämällä metaforan määrittämiä yleisiä puitteita sekä yrityksen sisäisesti päättämiä yksityiskohtia päästään hyvin alkuun.

Toisena apukysymyksenä oli "Mistä teknisen velan hallinnointi koostuu?", johon lähdettiin hakemaan vastausta luvussa kolme ja sen ensimmäisessä alaluvussa. Näissä tarkasteltiin mitä hallinnointi tarkoittaa ja millaisiin osiin sitä voidaan jakaa. Tarkastelun alla oli kaksi erilaista jakamistapaa, jotka lopulta olivat kuitenkin hyvin samanlaiset, erilaisesta nimeämisestä huolimatta. Varsinaiset toiminnalliset osat, se mitä teknisen velan hallinnointiin kuuluu, olivat molemmissa samansuuntaiset. Teknisen velan hallinnointi voidaan jakaa esimerkiksi seuraaviin osiin: tunnistaminen, mittaaminen, priorisointi, seuranta, ehkäiseminen, dokumentointi ja viestintä.

Tutkimuskysymyksenä oli Miten teknistä velkaa hallinnoidaan? Jotta teknisen velan hallinnointi olisi mahdollista, on tärkeää tietää miten, milloin ja miksi teknistä velkaa syntyy. Toisaalta teknisen velan hallinnointi vaatii muun

muassa tunnustamista, tunnistamista ja rajoittamista. Tunnustaminen: teknisestä velasta vapaata ohjelmistoa ei olekaan. Tunnistaminen: missä ja millaista teknistä velkaa ohjelmistossa on. Rajoittaminen: tahattomien ja/tai ei suunniteltujen teknisten velkojen rajoittaminen. Hallinnoinnilla hoidetaan ja seurataan muun muassa teknisten velkojen hinnan muodostumista, priorisointia, takaisinmaksua ja koron kasvua. Suunnittelemalla, ennakoimalla ja tekemällä päätöksiä siitä kuinka ja milloin teknistä velkaa poistetaan vaikuttaa ohjelmiston koko elinkaareen. Teknisen velan hallinnointiin vaikuttaa myös muun muassa organisaation liiketoiminnalliset tavoitteet ja kilpailutilanne. Teknisellä velalla voidaan myös saavuttaa taloudellista hyötyä tai ratkaisevaa etua markkinoille tuloajassa, silloin on kyseessä hallinnoitu ja suunniteltu strateginen tekninen velka. Teknisen velan hallinnointiin ja laskentaan on tarjolla vähänlaisesti omia työkaluja tai menetelmiä. Lähinnä rahoitus- ja riskienhallinnan puolelta on haettu menetelmiä lähtökohdaksi tähän soveltuvien kehysten ja menetelmien kehittämiseksi, ja näitä tullaan varmasti kehittämään ja tutkimaan lisää tarpeiden kasvaessa. Toisaalta se, millä välineillä hallinnointia hoidetaan, ei lopulta ole ratkaisevaa, kunhan kaikki osapuolet ovat aina tietoisia teknisen velan tilanteesta. Tässä taas avainasemassa on jatkuva viestintä kaikkien osapuolten kesken. Toisaalta monipuolinen, hallinnoinnin eri osa-alueita tukeva ohjelmisto helpottaisi teknisen velan hallinnointia.

Teknisen velan yhteydessä ratkaistavana on aina suhde ohjelmiston laadun ja liiketoiminnan tavoitteiden kanssa. Toisaalta siihen liittyy aina myös erilaisia kompromisseja niin prosessien, toiminnallisuuksien, testauksen kuin suunnittelunkin suhteen. Teknisen velan hallinnoinnin tärkeys korostuu jatkuvasti kiristyvässä kilpailutilanteessa, hallinnoimattomaan velkaan liittyy aina yllättäviä ja suuria menoeriä ja aikarasisiteita. Hallinnoimattoman velan esimerkiksi estäessä tärkeän asiakkaan uusien ominaisuuksien toimittamisen, voivat projektin kustannukset nousta kriittisesti, jolloin projekti jää tappiolliseksi. Kuten alussa mainittiin teknisen velan hinnan mahdollisesti yltävän vuonna 2015 noin yhteen triljoonaan dollariin, on hyvin ymmärrettävää, että kiinnostus tekniseen velkaan kasvaa myös yritysten puolella. Ohjelmistosta riippuen teknisen velan vaikutukset kustannuspuolella ovat kasvaneet runsaasti, eivätkä ne lähiaikoina ainakaan laskemaan tule. Teknisen velan huomioimatta jättämisellä taas voi olla hyvinkin radikaaleja vaikutuksia, ääritapauksessa jopa yritysten konkurssseja.

Tämän tutkimuksen tarkoituksena on ollut antaa yleiskuva siitä kuinka teknistä velkaa hallinnoidaan ja mistä osista hallinnointi koostuu. Tutkimus on pyrkinyt myös kuvaamaan, sitä kuinka moniulotteinen asia tekninen velka on ja millaisiin asioihin on syytä kiinnittää huomiota hallinnoitaessa teknistä velkaa. Yleiskuvasta johtuen tutkimus ei ole sisältänyt syvällisiä, yksityiskohtaisia ratkaisumalleja tai tietoja siitä kuinka teknisen velan hallinnointi tulisi ratkaista tai järjestää yksittäisissä tilanteissa. Tutkimus ei myöskään ole ottanut kantaa siihen millä työkalulla tai -kaluilla hallinnointia tulisi tehdä, jotain esimerkkejä työkaluista eri osa-alueille on lueteltu. Tässä tutkimuksessa ei ole tarkasteltu yksittäisiä teknisen velan hallinnointiin soveltuvia työkaluja, tutkimuksessa

käytetty tieto työkaluista perustuu lähteinä olleisiin tutkimuksiin. Työkalujen osalta myös uutta tutkimustietoa ja/tai työkaluja on voinut tulla tarjolle käytetyn lähdemateriaalin julkaisun jälkeen sekä tämän tutkimuksen aikana. Mahdollisista työkaluista ja niiden soveltuvuudesta sekä niiden erilaisista integrointimahdollisuuksista tulisi kirjoittaa oma tutkimuksensa. Tutkimuksessa ei myöskään ole yleiskuvaa tarkemmin perehdytty yksittäisiin erilaisiin laskentamalleihin /-menetelmiin. Aiheen ajankohtaisuuden ja kiinnostavuuden vuoksi relevantteja uusia tutkimustuloksia on voitu myös julkaista tämän tutkimuksen kirjoittamisen aikana. Lisäksi joitakin olennaisia tutkimuksia on voinut jäädä löytymättä hakusanavalintojen vuoksi.

Vaikka teknistä velkaa on tutkittu jo suhteellisen kauan, on vielä paljon asioita, joita täytyy tutkia ja kehittää lisää. Tällaisia ovat esimerkiksi teknisen velan hintaan ja sen muodostumiseen tai priorisointiin liittyen, mutta myös käytännön tutkimuksia siitä miten teknistä velkaa hallinnoidaan yrityksissä ja millaisia työkaluja yritykset kaipaavat hallinnoinnin tueksi. Teknisen velan hallinnoinnin työkalujen puute on myös avannut uusia mahdollisuuksia erilaisten hallinnointiin soveltuvien työkalujen ja sovellusten valmistamiselle.

Jatkotutkimuskohteeksi on valikoitunut teknisen velan hallinnointi IT-alan yrityksissä, rajauksena Keski-Suomi. Muita mahdollisia tutkimusaiheita voisivat olla esimerkiksi:

1. Millaisia laskentamalleja yritykset käyttävät teknisen velan hallinnoinnissa?
2. Miten teknistä velkaa hallinnoidaan pienissä/suurissa yrityksissä?
3. Millaisia työkaluja ja tekniikoita yrityksissä käytetään teknisen velan hallinnointiin?
4. Millaisia työkaluja ja tekniikoita haluttaisiin teknisen velan hallinnoinnin tueksi?
5. Miten ja millaisilla työkaluilla teknisen velan viestintää ja dokumentointia voidaan tukea?
6. Tekninen velka yritysjohdon näkökulmasta.

LÄHTEET

- Allman, E. (2012). Managing technical debt. *Commun. ACM* 55(5), 50-55.
- 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.
- Brown, N., Cai, Y., Guo, Y., Kazman, R., Kim, M., Kruchten, P., Lim, E., McCormack, A., Nord, R., Ozkaya, I., Sangwan, R., Seaman, C., Sullivan, K., Zazworka, N. (2010). Managing technical debt in software-reliant systems. *Teoksessa Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, Santa Fe, New Mexico, USA (47-52). New York, NY, USA: ACM.
- Buschmann, F. (2011). To pay or not to pay technical debt. *IEEE Software*, 28(6), 29-31.
- Cunningham, W. (1992). The WyCash portfolio management system. *Teoksessa Addendum to the Proceedings on Object-Oriented Programming Systems, Languages, and Applications (Addendum)*, Vancouver, British Columbia, Canada (29-30). New York, NY, USA: ACM.
- Curtis, B., Sappidi, J. & Szyrkarski, A. (2012). Estimating the size, cost, and types of technical debt. *Managing Technical Debt (MTD), 2012 Third International Workshop On*, (49-53). Zurich, Switzerland.
- Eisenberg, R. J. (2012). A threshold based approach to technical debt. *SIGSOFT Softw.Eng.Notes*, 37(2), 1-6.
- Ernst, N. A., Bellomo, S., Ozkaya, I., Nord, R. L. & Gorton, I. (2015). Measure it? manage it? ignore it? software practitioners and technical debt. *Teoksessa Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, Bergamo, Italy (50-60). New York, NY, USA: ACM.
- Falessi, D., Kruchten, P., Nord, R. L. & Ozkaya, I. (2014). Technical debt at the crossroads of research and practice: Report on the fifth international workshop on managing technical debt. *SIGSOFT Softw.Eng.Notes*, 39(2), 31-33.
- Fernandez-Sanchez, C., Garbajosa, J., Vidal, C. & Yague, A. (2015). An analysis of techniques and methods for technical debt management: A reflection from the architecture perspective. *Software Architecture and Metrics (SAM), 2015 IEEE/ACM 2nd International Workshop on*, (22-28).
- Fowler, M. (2009, 14. lokakuuta, reposted 2014, 19. marraskuuta) Technical Debt Quadrant. Bliki. Haettu 1.11.2015 osoitteesta <http://www.martinfowler.com/bliki/TechnicalDebtQuadrant.html>
- Guo, Y. & Seaman, C. (2011). A portfolio approach to technical debt management. *Teoksessa Proceedings of the 2Nd Workshop on Managing Technical Debt*, Waikiki, Honolulu, HI, USA (31-34). New York, NY, USA: ACM.

- 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. *SIGSOFT Softw.Eng. Notes*, 38(5), 51-54.
- Letouzey, J. & Ilkiewicz, M. (2012). Managing technical debt with the SQALE method. *Software, IEEE*, 29(6), 44-51.
- 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. *Software, IEEE*, 29(6), 22-27.
- Martini, A. & Bosch, J. (2015). Towards prioritizing architecture technical debt: Information needs of architects and product owners. Teoksessa *Software Engineering and Advanced Applications (SEAA), 2015 41st Euromicro Conference on*, (422-429).
- McConnell, Steve (2007, 1. marraskuuta). Technical Debt. 10x Software Development. Haettu 1.11.2015 osoitteesta http://www.construx.com/10x_Software_Development/Technical_Debt/
- McGregor, J. D., Monteith, J. Y. & Zhang, J. (2012). Technical debt aggregation in ecosystems. Teoksessa *Proceedings of the Third International Workshop on Managing Technical Debt*, Zurich, Switzerland (27-30). Piscataway, NJ, USA: IEEE Press.
- Nugroho, A., Visser, J. & Kuipers, T. (2011). An empirical model of technical debt and interest. Teoksessa *Proceedings of the 2Nd Workshop on Managing Technical Debt*, Waikiki, Honolulu, HI, USA (1-8). New York, NY, USA: ACM.
- Seaman, C. & Guo, Y. (2011). Measuring and monitoring technical debt. Teoksessa *Advances in Computers*, 82, 25-46.
- Snipes, W., Robinson, B., Guo, Y. & Seaman, C. (2012). Defining the decision factors for managing defects: A technical debt perspective. *Managing Technical Debt (MTD), 2012 Third International Workshop on*, (54-60). Stamford, Conn. (2010). Gartner Estimates Global 'IT Debt' to Be \$500 Billion This Year, with Potential to Grow to \$1 Trillion by 2015. Gartner. Haettu 29.10.2015 osoitteesta <http://www.gartner.com/newsroom/id/1439513>
- Theodoropoulos, T., Hofberg, M. & Kern, D. (2011). Technical debt from the stakeholder perspective. Teoksessa *Proceedings of the 2Nd Workshop on Managing Technical Debt*, Waikiki, Honolulu, HI, USA (43-46). New York, NY, USA: ACM.
- Tom, E., Aurum, A. & Vidgen, R. (2012). A CONSOLIDATED UNDERSTANDING OF TECHNICAL DEBT. Teoksessa *ECIS 2012 Proceedings*. Paper 16.
- Tom, E., Aurum, A. & Vidgen, R. (2013). An exploration of technical debt. *Journal of Systems and Software*, 86(6), 1498-1516.
- Wolff, E. & Johann, S. (2015). Technical debt. *Software, IEEE*, 32(4), 94-c3.
- Yli-Huumo, J., Maglyas, A. & Smolander, K. (2014). The sources and approaches to management of technical debt: A case study of two product lines in a middle-size finnish software company. Teoksessa A. Jedlitschka, P. Kuvaja, M. Kuhrmann, T. Männistö, J. Münch & M. Raatikainen (toim.), *Lecture Notes in Computer Science* (s. 93-107) Springer International Publishing.

- Zazworka, N., Seaman, C. & Shull, F. (2011). Prioritizing design debt investment opportunities. Teoksessa *Proceedings of the 2Nd Workshop on Managing Technical Debt*, Waikiki, Honolulu, HI, USA (39-42). New York, NY, USA: ACM.
- Zazworka, N., Spinola, R. O., Vetro', A., Shull, F. & Seaman, C. (2013). A case study on effectively identifying technical debt. Teoksessa *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, Porto de Galinhas, Brazil (42-47). New York, NY, USA: ACM.
- Zazworka, N., Vetro', A., Izurieta, C., Wong, S., Cai, Y., Seaman, C. & Shull, F. (2014). Comparing four approaches for technical debt identification. *Software Quality Control*, 22(3), 403-426.