

**Paavo Karppinen**

**Ohjelmistokehitysprojektien ammattilaisten näkemyksiä  
koronaviruspandemian aikaisesta projektinhallinnasta**

Tietotekniikan pro gradu -tutkielma

14. toukokuuta 2024

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

**Tekijä:** Paavo Karppinen

**Yhteystiedot:** paavo\_karppinen@outlook.com

**Ohjaajat:** Mirja Pulkkinen ja Tuomas Kokko

**Työn nimi:** Ohjelmistokehitysprojektien ammattilaisten näkemyksiä koronaviruspandemian aikaisesta projektinhallinnasta

**Title in English:** Views from software development project professionals about project management during the COVID-19 pandemic

**Työ:** Pro gradu -tutkielma

**Opintosuunta:** Ohjelmisto- ja tietoliikennetekniikka

**Sivumäärä:** 82+10

**Tiivistelmä:** Ohjelmistokehitysprojektit ovat tutkimusten mukaan olleet hyvin haasteellisia jo useiden vuosien ajan. Projektinhallinnan osa-alueista haasteellisia ovat olleet muun muassa aikataulun, kommunikaation ja vaatimusten hallinta. Ketterän kehityksen menetelmät ovat nykyaikana hyvin suosittuja ohjelmistokehitysprojekteissa, josta johtuen myös dokumentaation hallinta on ilmestynyt haasteeksi. Koronaviruspandemian myötä lisääntynyt etätyö on tuonut vielä omat ongelmansa projektinhallintaan. Jotta projektit voidaan viedä läpi onnistuneesti, on tiedettävä, missä haasteita ilmenee. Tutkimuksen tavoitteena olikin selvittää koronaviruspandemian aikaisten ohjelmistokehitysprojektien haasteita ja onnistumisia. Tutkimuksen teoriaosassa tarkasteltiin aluksi projekteissa käytettyjä menetelmiä niin perinteisten, kuin ketterien menetelmien osalta. Seuraavaksi selvitettiin mitä kirjallisuus kertoo ohjelmistokehitysprojektien haasteellisuudesta Scrum-kehityksen käytössä sekä yleisellä tasolla. Kirjallisuuskatsuksessa löydettiin tutkimusaukko koskien ohjelmistokehitysprojektien hallintaa koronaviruspandemian aikana. Tutkimusta koronaviruspandemian vaikutuksesta ohjelmistokehitykseen on tehty enemmän kehittäjien näkökulmasta ja kuinka se vaikutti heihin. Empiirinen tutkimus tehtiin kyselytutkimuksena. Kyselytutkimuksen kohderyhmä oli koronaviruspandemian aikaisissa ohjelmistokehitysprojekteissa työskennelleet henkilöt. Kyselyssä aihetta tarkasteltiin sekä määrällisesti, että laadullisesti, eli kyselyssä oli asteikollisia kysy-

myksiä ja avokysymyksiä. Kysely sai yhteensä 13 vastausta, joten alhaisen vastausmäärän takia, tutkimuksessa keskityttiin laadulliseen analyysiin, eikä määrällistä aineistoa käsitelty kovin syvällisesti. Tutkimustulokset osoittavat haasteiden olleen samankaltaisia, kuin aikaisemmin on tutkimuksissa löydetty.

**Avainsanat:** ohjelmistoprojekti, ohjelmistokehitysprojekti, projektinhallinta, koronaviruspandemia, etätyö, etätiimi, kysely

**Abstract:** Based on previous research, software development projects have been challenging for years. From the project management areas, schedule, communication and requirements management have been challenging. Agile methodologies are nowadays extremely popular in software development projects, which means that documentation management has become an issue as well. Remote work has increased due to the COVID-19 pandemic, which has brought its own set of issues to project management. So that projects can be successfully completed, we need to know where the challenges lie. The goal of this research was to find out the challenges and successes of projects done during the pandemic. The methodologies, both traditional and agile, used in projects have been examined in the theory part of the research. In the theory portion, previous research on project management challenges using Scrum and in general sense were presented. A research gap on the software development project management during the COVID-19 pandemic was discovered during the literature review. Research on the pandemic's effect have been made more on the point of view of the developers and how it affected them. The empirical study was performed as survey research. The target group for the study were people who worked on software development projects during the COVID-19 pandemic. The survey observed the topic qualitatively and quantitatively i.e. the survey had both scaled questions and open-ended questions. The survey received 13 responses in total. Due to the low response, the research focused on qualitative analysis and the quantitative data was not deeply processed. The research results indicate that the challenges during the pandemic were similar as to shown in previous research.

**Keywords:** software project, software development project, project management, COVID-19 pandemic, remote work, remote team, survey

## Termiluettelo

Etätyö	Työ, joka tehdään organisaation normaalin työtilojen ja -aikojen ulkopuolella.
Hybridityö	Etätyön ja toimistotyön välimuoto, esimerkiksi viikon aikana työskentely tapahtuu joinain päivinä toimistolla ja joinain etänä.
MVP	Lyhenne termille <i>Minimum Viable Product</i> , eli pienin mahdollinen tuote, jolla on mahdollista tuottaa arvoa sen käyttäjille.
Ohjelmistoprojekti	Ks. ohjelmistokehitysprojekti.
Ohjelmistokehitysprojekti	Väliaikainen pyrkimys, jolla on tietty alku- ja loppupäivämäärä, tuottaa jokin ohjelmisto tiettyyn tarkoitukseen tai parantamaan aikaisempaa ohjelmistoa.
PMBOK	Project Management Body of Knowledge, projektinhallinnan tietämyksen sisältävä teos.
Projekti	Väliaikainen pyrkimys tuottaa jokin jotain tuotetta, palvelua tai jotain muuta tulosta.
Projektinhallinta	Tiedon, taitojen, työkalujen ja tekniikoiden (johtamistapojen) soveltaminen, jotta projektin vaatimukset, tavoitteet ja päämäärä saavutetaan.

## Kuviot

Kuvio 1. Vesiputousmallin prosessi mukaillen Van Casteren (2017) ja Balaji ja Murugaiyan (2012).	7
Kuvio 2. V-mallin prosessi mukaillen Balaji ja Murugaiyan (2012).	9
Kuvio 3. W-mallin prosessi mukaillen Kargl ym. (2019).	10
Kuvio 4. Scrum-prosessi mukaillen Permana (2015).	15
Kuvio 5. Esimerkki Kanban-taulusta mukaillen Kirovska ja Koceski (2015).	20

## Taulukot

Taulukko 1. Scrum-kehityksen haasteita ja ongelmia mukaillen Anand ja Dinakaran (2015).	27
Taulukko 2. Ketteriin menetelmiin ja Scrum-kehitykseen siirtymisen mahdollisia ongelmia (López-Martínez ym. 2016).	29
Taulukko 3. Ohjelmistoprojektien hallinnan haasteista tehdyn kyselyn tulokset mukaillen Demir (2009).	31
Taulukko 4. Ketterän kehityksen laajuuden ja muutosten haasteita (Marnada ym. 2022).	36
Taulukko 5. Hajautetun ohjelmistokehityksen haasteita (Da Silva ym. 2010).	37
Taulukko 6. Analytiikka LinkedIn-julkaisuista.	42
Taulukko 7. Kyselyn vastaajien sukupuolijakauma.	45
Taulukko 8. Kyselyn vastaajien ikäjakauma.	45
Taulukko 9. Kyselyn vastaajien kokemusjakauma.	45
Taulukko 10. Kyselyn vastaajien projektien jäsenten määrän jakauma.	46
Taulukko 11. Kyselyn vastaajien organisaatioiden työntekijöiden jakauma.	47
Taulukko 12. Kyselyn vastaajien projektien työtavan jakauma.	47
Taulukko 13. Kyselyn vastaajien projektien projektinhallinnan menetelmien jakauma.	48
Taulukko 14. Kyselyn vastaajien projektien projektinhallinnan osa-alueiden haasteellisuus.	49
Taulukko 15. Kyselyn vastaajien projektien käytettyjen työkalujen ja teknologioiden vaikutus projektinhallintaan.	50
Taulukko 16. Kyselyn vastaajien projektien sisäisten tekijöiden vaikutus projektinhallintaan.	51
Taulukko 17. Kyselyn vastaajien projektien ulkoisten tekijöiden vaikutus projektinhallintaan.	52
Taulukko 18. Kyselyn vastaajien projektien muiden tekijöiden vaikutus projektinhallintaan.	53

# Sisällys

1	JOHDANTO .....	1
2	PROJEKTINHALLINTAMENETELMIÄ .....	6
2.1	Perinteisiä menetelmiä .....	6
2.1.1	Vesiputousmalli .....	6
2.1.2	V-malli .....	8
2.2	Ketterät menetelmät .....	11
2.2.1	Scrum.....	14
2.2.2	Kanban .....	19
2.3	Projektinhallintamenetelmien tiivistys ja merkitys .....	21
3	AIKAISEMPI TUTKIMUS PROJEKTINHALLINNAN HAASTEISTA .....	22
3.1	Projektinhallintamenetelmiin liittyvä tutkimus .....	22
3.2	Ohjelmistoprojektien hallintaan liittyvä tutkimus .....	30
3.3	Aikaisemman tutkimuksen tiivistys ja merkitys .....	37
4	TUTKIMUSASETELMA .....	39
4.1	Tutkimusmenetelmä .....	39
4.1.1	Kirjallisuuskatsaus.....	39
4.1.2	Kyselytutkimus .....	40
4.2	Aineiston keruu.....	42
4.3	Aineiston analyysi.....	43
5	TULOKSET.....	45
5.1	Kyselyn taustakysymysten tulokset .....	45
5.2	Kyselyn kysymysten 7 ja 11 tulokset.....	48
5.3	Työtap.....	52
5.4	Kommunikaatio .....	54
5.5	Ulkoiset tekijät ja tilaaja .....	55
5.6	Johtajuus projektissa .....	56
5.7	Käytetyt työkalut .....	57
5.8	Projektien sisäiset tekijät ja kuinka minimoida haasteita.....	58
5.9	Onnistumiset.....	61
5.10	Koronaviruspandemian vaikutus.....	62
6	POHDINTA .....	64
6.1	Vertailu aikaisempaan tutkimukseen .....	64
6.2	Tutkimuksen merkitys .....	65
6.3	Tutkimuksen rajoitukset .....	66
6.4	Jatkotutkimuskohteita .....	67
7	JOHTOPÄÄTÖKSET JA YHTEENVETO .....	68
	LÄHTEET .....	70

LIITTEET.....	76
A Tutkimuksen kyselylomake .....	76

# 1 Johdanto

Ohjelmistoprojektien epäonnistumisista on tehty vuosien mittaan useita tutkimuksia. Eräs tunnetuimmista tutkimuksista on Standish Groupin säännöllisesti julkaistu “The CHAOS report”, jonka vuoden 1994 mittausten mukaan 31,1 % projekteista peruutetaan ennen aikaisesti ennen niiden valmistumista (Clancy 1995). Kyseinen vuoden 1994 julkaisu on saanut kuitenkin kritiikkiä muun muassa projektin onnistumisen määrittämisen, ja kulujen arvioinnin suhteen (Eveleens ja Verhoef 2009; Jørgensen ja Moløkken-Østvold 2006). Vuoden 2015 jälkeen Standish Group on ottanut käyttöön uudenmallisen määritelmän projektin onnistumiselle, ja esimerkiksi vuoden 2015 osalta onnistuneiden, haasteellisten ja epäonnistuneiden projektien osuudet olivat seuraavat: 29 %, 52 % ja 19 % (*CHAOS Report 2015* 2015). El Emam ja Koru (2008) tutkivat ohjelmistoprojektien onnistumisista kahtena vuotena, 2005 ja 2007, ja heidän löydöstensä mukaan 11,5–15,1 % ohjelmistoprojekteista perutaan, ennen kuin mitään konkreettista toiminnallisuutta on saatu aikaan ensimmäiseen julkaisuun mennessä. Tutkijoiden mukaan numerot vastaavat muiden kyselyiden tuloksia, jotka oli tehty lähellä tutkimuksen ajankohtaa, jonka lisäksi projektien peruuttamisaste eri tutkimusten mukaan on ollut laskussa vuoden 1994 jälkeen. Jos kuitenkin otetaan mukaan epäonnistuneet projektit, osuus nousee jo 26–34 prosenttiin, joka El Emam ja Koru (2008) mukaan on kuitenkin korkea määrä. El Emam ja Koru (2008) kertovat vaatimusten ja laajuuden muutokset olevan pääsyyt projektien peruuntumiseen, jonka lisäksi ylemmän johtoportaan sitoutuminen ja epäpätevät johtamistaidot olivat muita syitä projektien peruuntumisessa. Onnistuneiden projektien suurimpana haasteena pidettiin aikataulussa pysyminen. Projektinhallinnan tietämys (PMBOK) kertoo projektinhallinnan koostuvan muun muassa laajuuden ja aikataulun hallinnasta (Guide 2001), jotka esiintyvät usein haasteina ohjelmistokehitysprojekteissa. PMBOK:n mukaan projektinhallinnan tietämysalueet, eli projektinhallinnan osa-alueet, ovat seuraavat: 1. Projektin kokonaisuuden hallinta, 2. laajuuden hallinta, 3. aikataulun hallinta, 4. kustannusten hallinta, 5. laadunhallinta, 6. henkilöstöresurssien hallinta, 7. kommunikation hallinta, 8. riskienhallinta ja 9. hankintojen hallinta. Jotta projektien epäonnistumisriskiä voidaan pienentää mahdollisimman paljon, on oltava tiedossa, missä projektinhallinnan osa-alueissa haasteita on koettu, ja tämän tietämyksen kautta riskiä voidaan pienentää, kunhan tarpeellisia toimintoja haasteiden minimoimiseen otetaan käyttöön. Big ja Small (2013) mu-



kaan projektinhallinnan tai prosessin hallinnan osaaminen on yksi tärkeistä tekijöistä, joka johtaa onnistuneeseen projektiin. Tietoisuus projektinhallinnan haasteista auttaa tässä asiassa.

Guide (2001) määrittelee projektin olevan väliaikainen (tietty alkua- ja päätöspäivä) pyrkimys tuottamaan jotain tuotetta, palvelua tai jotain muuta tulosta. Arto, Martinsuo, Kujala ym. (2008) kuitenkin kertovat projektille olevan useita eri määrittämiä, määrittäen projektin muun muassa seuraavanlaisesti: “johtamisympäristö, jonka tehtävänä on toimittaa yksi tai useampia tuloksia rajatun liiketoimintasuunnitelman mukaan”, toisissa määrittämissä projektia kuvataan ainutkertaisena toimeksiantona tiettyjen vaatimusten suhteen. Näiden määrittämysten lisäksi joissain tapauksissa korostetaan projektille erikseen määritettyä organisaatiota. Eri määrittämissä on kuitenkin samaa, että projektilla on alkua ja loppu. Tämän tutkielman käsittelyssä on projektit, joissa kehitetään jotain ohjelmistoa. Arto, Martinsuo, Kujala ym. (2008) kuvaavat kehitysprojektien olevan pyrkimys, jossa sovelletaan tietoa tuottamaan uusia tuotteita, palveluita, ratkaisuja, kyvykkyyksiä tai toiminnallisuuden luomista tai vanhojen parantelua. Tutkielmassa tulen käyttämään ohjelmistokehitysprojektin kanssa käsitettä “ohjelmistoprojekti”, tarkoittaen kuitenkin samaa asiaa. Projektinhallinta määritetään PM-BOK:ssa tiedon, taitojen, työkalujen ja projektiaktiiviteettien tekniikoiden hyväksikäyttöä, jotta projektin vaatimukset saavutetaan (Guide 2001).

Ketterän kehityksen mallit ovat nykypäivänä ohjelmistokehityksessä dominoivia. Vuoden 2021 alussa Digital.ai:n tekemän “State of Agile” kyselyn vastaajien mukaan 86 % ohjelmistokehitystiimeistä käyttävät ketteriä menetelmiä, kun taas vuoden 2019 lopussa kerätyn aineiston mukaan vastaava luku oli 37 (*15th State of Agile Report 2021*; *14th State of Agile Report 2020*). Vuoden 2022 vastaavan raportin mukaan yleisimmin käytetty ketterän kehityksen viitekehys oli Scrum (87 %) ja toiseksi käytetyin Kanban (56 %) (*16th State of Agile Report 2022*). Saman raportin mukaan skaalatuista kehyksistä Scaled Agile Framework (SAFe) oli suosituin 53 %:n suosiolla. Vuoden 2021 raportissa 74 % vastaajista kertoivat, että heillä joko on jo, tai ovat suunnittelemassa DevOps-aloitteita. Mikä tekee kehitysmenetelmästä ketterän, on seuraavanlaiset ohjelmistokehityksen ominaisuudet: inkrementaalisuus (pienet ohjelmistojulkaisut nopein syklein), yhteistyökykyisyys (asiakkaan ja kehittäjien tiivis yhteistyö), suoraviivaisuus (menetelmä/kehys on helppo oppia ja muokata sekä se on

hyvin dokumentoitu), ja mukautuvuus (mahdollisuus tehdä viime hetken muutoksia) (Abrahamsson ym. 2017). Keskeisintä ketterissä menetelmissä on iteratiivinen kehitysparadigma, jossa perinteisen vaatimusmäärittelyn ja -analyysin sijaan vaatimuksia voidaan arvioida iteraatiosta toiseen, jolloin vaatimuksia voidaan lisätä (uudet vaatimukset), muokata tai poistaa (Favare 2002).

Koronaviruspandemialla oli suuri vaikutus työelämään, kun ihmiset joutuivat usealla alalla jopa pakon edestä siirtymään toimistotiloista työskentelemään kotoa käsin. Tästä johtuvia pandemian vaikutukset olivat suurempia tai pienempiä riippuen toimialasta, lentoyhtiöiden toiminta pysähtyi aivan täysin, kun taas esimerkiksi ohjelmistoalalla, ja laajemmin IT-alalla, etätyö oli jo ennen pandemiaa hyväksytty tapa työskennellä ja pandemian aikana etätyö tuli välttämättömyydeksi (Kaushik ja Guleria 2020). Etätyöhön siirtymisen kautta pandemian aikana ihmisten kommunikaatio muuttui selkeästi, kun ihmiset eivät vain voi kävellä työ-kaverin työpisteelle ja aloittaa keskustelua, joka voi puolestaan vaikuttaa projektinhallinnassa kommunikaation hallintaan, kun kommunikaatiota ei voi ensinnäkään suorittaa yhtä helposti. Kirjallisuuskartoitusta tehdessäni otin huomioon, kuinka ohjelmistoalalla koronaviruspandemian vaikutuksia on tutkittu pääosin ihmislähtöisemmästä näkökulmasta, jonka lisäksi aikaisempi tutkimus on pääosin pandemian alkua ajoilta. Ihmislähtöistä tutkimusta on tehty esimerkiksi tutkien muun muassa yksilötasolla kehittäjien hyvinvointia ja erityisesti tuottavuutta pandemian aikana (Ralph ym. 2020; Ford ym. 2021). Koska pandemia on vaikuttanut suoraan ihmisiin, kuten tuottavuuteen tai hyvinvointiin, niin se vaikuttaa epäsuorasti myös projektinhallintaan muun muassa projektin aikataulutukseen. Vaikutuksista ketterään ohjelmistokehitykseen on tehty tutkimusta muun muassa siitä, kuinka pandemiasta johtuneet etätyö ja sosiaaliset rajoitteet vaikuttavat ketterään kehitykseen (Butt ym. 2021).

Etätyöllä tarkoitetaan organisaationallista työtä, joka tehdään organisaation normaalin työtilojen ja -aikojen ulkopuolella (Olson 1983). Hybridityö puolestaan tarkoittaa työtä, joka on toimisto- ja etätyön välissä, yleisellä tasolla tarkoittaen, että työntekijät työskentelevät osan viikon päivästä toimistolla ja osan etänä.

Tämän tutkielman tarkoituksena on tehdä tutkimusta enemmän asialähtöisemmästä näkökulmasta, tutkien pandemian vaikutusta ohjelmistoprojektien hallintaan tarkastelemalla aihetta pääosin etä- ja hybridityöhön siirtymisen kautta. Toisin sanoen tarkoituksena on tutkia, min-

kälaista vaikutusta pandemialla on ollut menetelmiin, käytäntöihin ja projektien vetämiseen ottamatta juurikaan kantaa projektin henkilöstöön kohdistuvaan vaikutukseen. Koronaviruspandemian vaikutusta tulen tutkimaan tekemällä katsauksen nykytilanteeseen projektinhallinnan osa-alueiden haasteista sekä eri tekijöiden vaikutusta projektinhallintaan. Tästä saadaan seuraava tutkimuskysymys:

Kuinka koronaviruspandemia on vaikuttanut ohjelmistoprojektien hallintaan?

Tähän tutkimuskysymykseen tullaan siis vastaamaan kartoittamalla ohjelmistoprojektien hallinnan haasteita ja onnistumisia nykytilanteessa, kun eletään hyvin vahvasti etä- ja hybridityössä, joihin siirtymiseen koronaviruspandemia oli erittäin suuri katalysaattori, sillä sen takia ohjelmistokehitystiimit joutuivat pandemian aikana siirtymään etätyöhön, kun he ennen sitä työskentelivät lähellä toisiaan toimistoilla (Miller ym. 2021). Ohjelmistoprojektien hallinnasta on tärkeää tehdä tutkimusta, sillä etä- ja, varsinkin, hybridityö on selkeästi uusi normaali, sillä jo useat suuretkin yritykset ovat siirtyneet pääsääntöisesti etätyöhön (Ford ym. 2021). Hyvin suuri projektipäälliköistä suosivat työskentelyä hybridimuodossa (Degerli 2022). Tarkoituksena on selvittää laajemmasta näkökulmasta ohjelmistoprojektien hallinnasta, kuin puolestaan selvittää tarkkoja haasteita hallintaan liittyen. Samankaltaista tutkimusta ohjelmistoprojektien hallinnasta ei ole juurikaan tehty niin pandemian aikana, kuin näin pandemian jälkeen, kun tilanne on hyvin normalisoitunut, joten tältä osin on kuvatus tutkimuksen tekeminen hyödyllistä. Tutkimusta on kuitenkin tehty eri tyyppisten haasteiden osalta pandemian ajalta, tai oikeastaan pandemian aikaisesta etätyöstä. Haasteellisten osa-alueiden ollessa tiedossa, on mahdollista tehdä jatkotutkimusta paljon syvällisemmin tietystä osa-alueesta siitä, mistä haasteet johtuvat. Projektinhallinnan osa-alueiden lisäksi on tarkoituksena selvittää, minkälainen vaikutus esimerkiksi eri työkaluilla on projektinhallintaan.

Tämä tutkielma jakautuu selkeästi kahteen vaiheeseen: Ensimmäinen vaihe on kirjallisuuskatsaus, jossa esitellään ohjelmistoprojektinhallintamenetelmiä sekä aikaisempaa tutkimusta projektinhallinnasta ja toinen vaihe on tutkielman empiirinen osio. Ensiksi luvussa 2 kuvataan eri projektinhallintamenetelmiä, pääpainona kuitenkin Scrum, sillä se on hyvin yleisessä käytössä ketteristä menetelmistä. Luvussa 3 esitellään aikaisempaa tutkimusta ensin Scrum-viitekehyksen käytössä löytyvistä haasteista, jonka jälkeen esitellään projektinhallinnan haasteista olevaa tutkimusta niin ennen pandemiaa, kuin pandemian aikaisesta pro-

jektimuotoisesta ohjelmistokehityksestä keskittyen korkean tason tutkimukseen, eli keskittymättä johonkin tiettyyn osa-alueeseen (kuten laadunhallinta). Luvussa 4 esitellään tutkielman metodologiaa siltä kannalta, kuinka aineisto kerätään ja analysoidaan sekä esitellään kirjallisuuskatsauksen ja kyselytutkimuksen taustat. Luvussa 5 esitellään aineiston analyysin tuloksia, jonka jälkeen luvussa 6 tehdään pohdintaa, kuinka tämän tutkimuksen tulokset vertautuvat aikaisempaan tutkimukseen projektinhallinnan haasteista sekä lopuksi pohditaan tutkimuksen toteutusta, luotettavuutta ja rajoituksia. Luvussa käsitellään lisäksi tutkimuksen merkitystä ja esitetään jatkotutkimuskohteita. Lopuksi luvussa 7 esitellään johtopäätökset, eli käydään vielä kertaalleen läpi, kuinka tutkimuskysymykseen vastattiin. Luvussa tehdään vielä lopuksi yhteenvetoa tutkimuksen tuloksista.

## 2 Projektinhallintamenetelmiä

Ohjelmistokehitysprojekteissa käytettävät hallintamenetelmät ovat siirtyneet perinteisistä menetelmistä, kuten vesiputousmallista, ketteriin menetelmiin. Tässä luvussa tullaan esittelemään ensin kaksi perinteistä menetelmää vesiputousmalli ja V-malli, jonka jälkeen esitellään ketterät menetelmät yleisellä tasolla. Yleisen esittelyn jälkeen esitellään ketteristä menetelmistä suosituimmat eli Scrum ja Kanban.

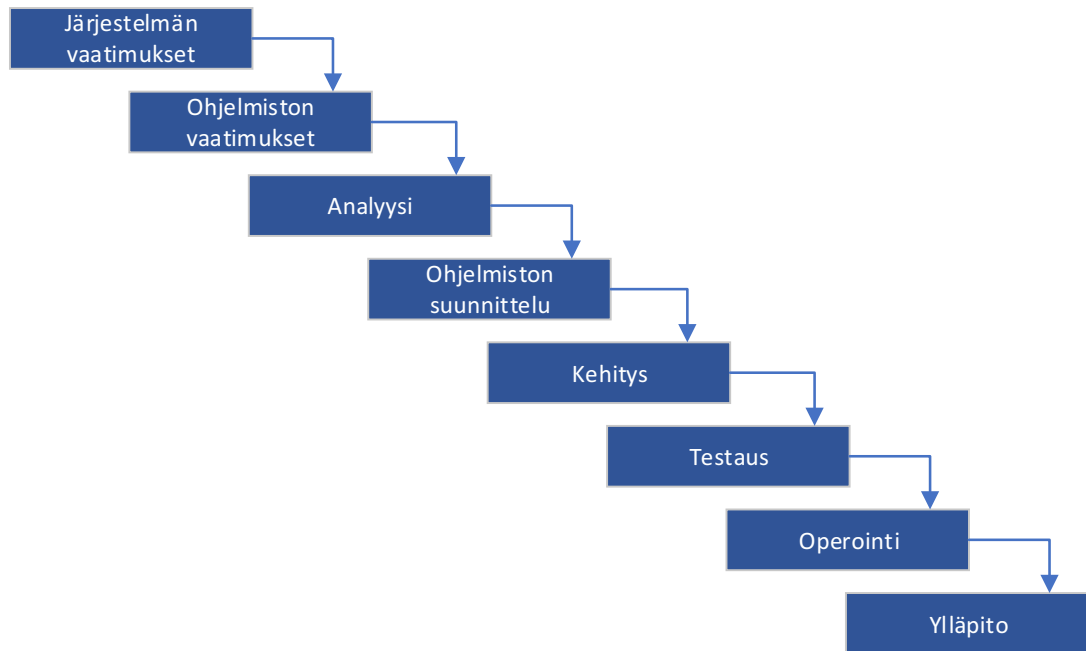
### 2.1 Perinteisiä menetelmiä

#### 2.1.1 Vesiputousmalli

Vesiputousmalli on esimerkki suunnitelmajohtoisesta projektinhallinnan menetelmästä. Tässä mallissa ohjelmistokehitystä tehdään seuraten tarkkaa suunnitelmaa, kuinka projekti tulee viedä eteenpäin. Vesiputousmallissa kehitys suunnitellaan kokonaisvaltaisesti projektin aloituksesta päätökseen asti. Suunnitelmassa erotellaan työpaketit, vastuut ja aikarajoitteet, ja tarkoituksena on, että ensimmäinen suunnitelma on mahdollisimman tarkka. Tarkkuuden kautta projektille saadaan vakautta ja rakennetta, ennakoituja resursseja ja dokumentoitua suunnittelua (Thesing, Feldmann ja Burchardt 2021).

Vesiputousmallia kuvataan nimeään muistuttavien vesiputouksia muistuttavien kuvioiden kautta. Tässä prosessimallissa vaiheet riippuvat lähteestä, mutta yhteistä monelle kuitenkin on analyysi, (järjestelmän) suunnittelu, kehitys, testaaminen, ja operointi/käyttöönotto (Van Casteren 2017; Balaji ja Murugaiyan 2012). Van Casteren (2017) lisäävät ennen analyysia vielä järjestelmän vaatimukset ja ohjelmiston vaatimukset ja Balaji ja Murugaiyan (2012) lisäävät vielä ylläpidon prosessin loppuun. Kuviossa 1 on esitetty näiden kahden kuvausten perusteella tehty kuvio.

Testaustasoihin kuuluvat yksikkötestaus, integraatiotestaus, järjestelmätestaus ja hyväksymistestaus. Yksikkötestauksella tarkoitetaan testausta, joka kohdistuu yksittäisiin ohjelmiston osiin. Yksikkötestauksessa testataan ohjelmiston osia, jotka toimivat itsenäisesti (Bentley, Bank ja Charlotte 2005). Seuraava testauksen taso, eli integraatiotestaus, kohdistuu yk-



Kuvio 1. Vesiputousmallin prosessi mukailien Van Casteren (2017) ja Balaji ja Murugaiyan (2012).

sikkötestausta suurempiin ohjelmiston osiin. Integraatiotestauksessa varmistetaan useamman ohjelmiston tai ohjelmiston osien toimivuus yhdessä ja se kohdistuu moduuleihin, toisin kuin yksikkötestaus, joka kohdistuu ohjelmiston lähdekoodin riveihin (Leung ja White 1990). Integraatiotestauksen jälkeen tulee järjestelmätestaus. Järjestelmätestauksessa testataan koko ohjelmiston toimivuus yhdessä, jonka jälkeen toteutetaan hyväksymistestaus (Bentley, Bank ja Charlotte 2005). Hyväksymistestaus suoritetaan projektin lopussa varmistamaan kehitetyn sovelluksen vastaavuus asiakkaan toivomuksiin. Käyttäjätarinat voivat olla oleellinen osa hyväksymistestausta (Lapointe-Boisvert, Mosser ja Trudel 2021). Yksikkö-, integraatio ja järjestelmätestausta suoritetaan aina, kun tehdään jotain uutta tai, kun olemassa olevaa muutetaan. Muutosten yhteydessä tehtyä testaamista kutsutaan myös regressiotestaamiseksi. Tätä testausta suoritetaan varmistamaan, ettei muutokset ole aiheuttaneet uusia virheitä, joita ei aiemmin testauksessa ilmennyt (Wong ym. 1997).

Vesiputousmallissa testaaminen suoritetaan vasta, kun koko ohjelmisto on saatu valmiiksi, ja testaamisvaihetta edeltäen kehitysvaihe “jäädätetään”, eli siihen ei enää palata. Tähän jäädyt-

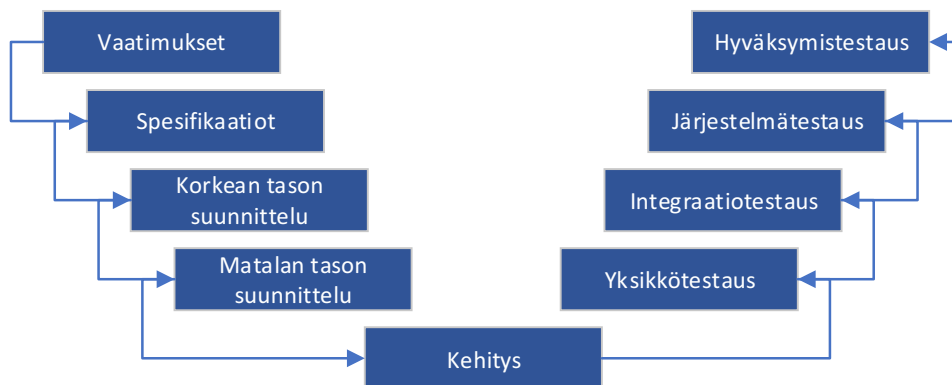
tämiseen myös liittyen, jokainen vaihe tulee suorittaa tietyn ajanjakson aikana, mikä niille on suunniteltu, eikä uusi vaihe voi alkaa ennen, kuin edellinen vaihe on saatu päätökseen. Koska testausvaihe suoritetaan vasta koko ohjelmiston kehityksen jälkeen, mahdolliset virheet huomataan vasta silloin, jolloin mahdolliset muutokset voivat tulla kalliiksi. Myöskään asiakkaan toivomia muutoksia ei voida ottaa huomioon jo käynnissä olevan kehitysprosessin aikana (Balaji ja Murugaiyan 2012).

Pargaonkar (2023) kertoo, mitkä ovat vesiputousmallin heikkouksia. Vesiputousmallin ongelmia ovat seuraavat: rajoitettu joustavuus, asiakkaan osallistamisen puute, korkea riski, pitkä kehitysaika ja sen huono soveltuvuus monimutkaisiin projekteihin. Joustavuus on ongelma, sillä vesiputousmallissa ei voida palata edelliseen vaiheeseen sen päätyttyä. Asiakas on puolestaan projektissa mukana vain sen alussa, joten jatkuvaa palautetta on hankala kerätä. Testaamisen ja validaation lopussa tapahtumisen takia kriittisten ongelmien ajoissa löytämisen mahdollisuus pienenee ja täten kasvattaen riskiä. Pargaonkarin mukaan vesiputousmalli voi johtaa pidempiin kehitysaikoihin verrattuna iteratiivisiin menetelmiin. Vesiputousmalli ei lisäksi sovellu projekteihin, joissa vaatimukset muuttuvat tai ovat epäselviä, sillä vesiputousmallissa vaatimukset tulisi suunnitella ja lyödä lukkoon ennen kehitystä. Balaji ja Murugaiyan (2012) kertovat lisäksi edellisten vaiheiden ongelmien ilmenemisen jälkimmäisissä vaiheissa olevan yksi vesiputousmallin ongelmista.

Balaji ja Murugaiyan kertovat vesiputousmallin vahvuuksia olevan vaatimusten selkeys ennen kehityksen aloittamista. Jokainen vaihe suorittaminen tietyssä ajanjaksossa, jonka jälkeen siirrytään seuraavaan vaiheeseen. Täten projektin etenemisen seuraamisen pitäisi olla selkeämpää. Vesiputousmallin käyttöönotto on helppoa sen lineaarisuuden takia. Vaadittujen resurssien määrä on minimaalinen ja jokaisessa vaiheessa tehdään kunnan dokumentaatio, joka edesauttaa kehityksen laatua.

### **2.1.2 V-malli**

V-malli, joka saa nimensä validaatiosta ja verifikaatiosta on vesiputousmallista kehitetty malli, jonka prosessikuviot muistuttavat V-kirjainta. V-mallin kuvio eroaa vesiputousmallista siten, että kehitysvaiheen jälkeen vaiheet kuvataan yläviistoon. Kehitystä edeltävät vaiheet



Kuvio 2. V-mallin prosessi mukailten Balaji ja Murugaiyan (2012).

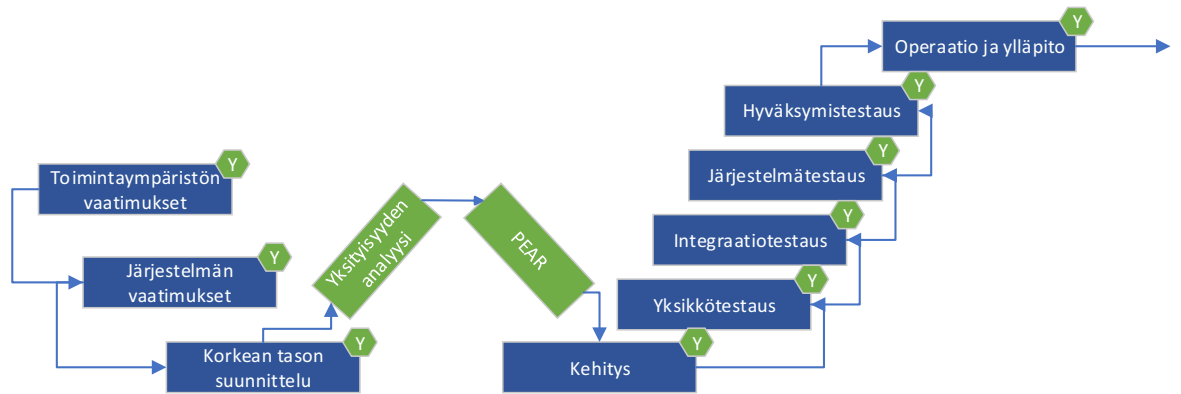
ovat seuraavat: vaatimukset, spesifikaatiot, korkean tason suunnittelu ja yksiköiden matalan tason suunnittelu. Kehityksen jälkeen puolestaan tulee eri testausvaiheet eli yksikkötestaus, integraatiotestaus, järjestelmätestaus ja hyväksymistestaus. V-mallin kehitysprosessi vaatii sen, että edellisen vaiheen tulokset varmistetaan verifikaation avulla, jolloin myös tulokset tulee hyväksyä, ennen kuin siirrytään seuraavaan vaiheeseen (Balaji ja Murugaiyan 2012). Kuviossa 2 on kuvattu V-mallin prosessi.

V-mallin vahvuuksina pidetään samoja, kuin vesiputousmallissa, jonka lisäksi testaaja on mukana jo vaatimusten määrittelyvaiheessa sekä toisin kuin vesiputousmalli, vaatimusmuutoksia voidaan tehdä jokaisessa vaiheessa (Balaji ja Murugaiyan 2012).

V-mallin heikkouksiin kuitenkin liittyy sen jäykkyys. Se ei ole niin joustava, kuin muut mallit. V-mallissa muutosten tullessa ilmi, dokumentaatiota joudutaan päivittämään niin vaatimusten, kuin testaamisen osalta. Projektin laajuuden osalta V-mallia ei suositella lyhytaikaisiin projekteihin, sillä V-malli vaatii tarkistuksia jokaisen vaiheen jälkeen (Balaji ja Murugaiyan 2012).

Uusien tietoturvalakien (kuten GDPR) tullessa voimaan, järjestelmäkehityksessä on otettava huomioon yksityisyys ja kuinka siitä pidetään kiinni. Kargl ym. (2019) ovat tästä johtuen kehittäneet V-mallista uudistetun mallin, joka keskittyy nimenomaan yksityisyyteen. Mallin nimeksi Kargl ym. ovat antaneet *W-malli*, jonka kuvio V-mallia seuraten muistuttaa W-kirjainta. V-mallia Kargl ym. lähtevät muokkaamaan ensinnäkin siten, että jokaisessa proses-





Kuvio 3. W-mallin prosessi mukailten Kargl ym. (2019).

sin vaiheessa yksityisyys otetaan huomioon. Kargl ym. (2019) kehittämän W-mallin vaiheet pysyvät samoina, eroten kuitenkin Balaji ja Murugaiyan (2012) esittelemästä V-mallista, että yksikkökohtaista matalan tason suunnittelua ei ole. Kargl ym. lisäävät korkean tason suunnittelun ja kehityksen väliin kaksi uutta vaihetta: yksityisyyden analyysi ja PEAR, joka tulee sanoista “Privacy-Enhanced Architecture”, eli yksityisyyttä tehostava arkkitehtuuri. Tämän mallin prosessikuvio on esitettyä kuviossa 3. Kuviossa Y-kirjaimella merkityt kuusikulmiot edustavat jokaisen vaiheen yksityisyyden huomioon ottamista. Yksityisyyden analyysivaiheessa yksityisyydensinööreistä ja juridisista avustajista koostuva tiimi analysoi järjestelmän suunnittelua tarkastaen, että se on yksityisyyden säilyttävä ja että se noudattaa lainsäädäntöä. Tässä vaiheessa on tarkoituksena paljastaa yksityisyyden uhkia alustavasta järjestelmäsuunnittelusta sekä löytää mahdollisia vastatoimenpiteitä löydetuille uhkille. PEAR on jaettu seuraaviin kahteen alavaiheeseen:

- Yksityisyyden säilyttävä korkeatasoinen suunnittelu. Tässä vaiheessa tiimi yksityisyys- ja järjestelmäinsinöörejä tarkastavat määritellyn järjestelmän ja yksityisyysvaatimukset ja reflektivat näitä edelliseen suunniteluun. Tämä vaihe kohdistuu järjestelmän korkeatasoisiin komponentteihin ja niiden vuorovaikutukseen.
- Yksityisyyden säilyttävä matalatasoinen suunnittelu. Tässä vaiheessa järjestelmä- ja yksityisyysinsinöörit rakentavat alemman abstraktion ylemmästä järjestelmäsuunnittelusta. Tarkoituksena on suunnitella korkeampaa tasoa vastaavat alemman tason komponentit, jotta kehitystyö seuraavassa vaiheessa on mahdollista.

## 2.2 Ketterät menetelmät

### Historiaa

Ketterä ohjelmistokehitys sai alkunsa, kun ketterän ohjelmistokehityksen julistus (engl. *Agile Manifesto*) julkaistiin (Beck ym. 2001). Julistuksen takana oli useampi johtava ohjelmistokehityksen henkilö. Julistus antaa ketterälle ohjelmistokehitykselle kaksitoista (12) periaatetta, jotka kuuluvat seuraavanlaisesti (listaus sama, kuin verkkosivulla (Beck ym. 2001)):

- Tärkein tavoitteemme on tyydyttää asiakas toimittamalla tämän tarpeet täyttäviä versioita ohjelmistosta aikaisessa vaiheessa ja säännöllisesti.
- Otamme vastaan muuttuvat vaatimukset myös kehityksen myöhäisessä vaiheessa. Ketterät menetelmät hyödyntävät muutosta asiakkaan kilpailukyvyn edistämiseksi.
- Toimitamme versioita toimivasta ohjelmistosta säännöllisesti, parin viikon tai kuukauden välein, ja suosimme lyhyempää aikaväliä.
- Liiketoiminnan edustajien ja ohjelmistokehittäjien tulee työskennellä yhdessä päivittäin koko projektin ajan.
- Rakennamme projektit motivoituneiden yksilöiden ympärille. Annamme heille puitteet ja tuen, jonka he tarvitsevat ja luotamme siihen, että he saavat työn tehtyä.
- Tehokkain ja toimivin tapa tiedon välittämiseksi kehitystiimille ja tiimin jäsenten kesken on kasvokkain käytävä keskustelu.
- Toimiva ohjelmisto on edistymisen ensisijainen mittari.
- Ketterät menetelmät kannustavat kestävään toimintatapaan. Hankkeen omistajien, kehittäjien ja ohjelmiston käyttäjien tulisi pystyä ylläpitämään työtahtinsa hamaan tulevaisuuteen.
- Teknisen laadun ja ohjelmiston hyvän rakenteen jatkuva huomiointi edesauttaa ketteryyttä.
- Yksinkertaisuus - tekemättä jätettävän työn maksimointi - on oleellista.
- Parhaat arkkitehtuurit, vaatimukset ja suunnitelmat syntyvät itseorganisoituvissa tiimeissä.
- Tiimi tarkastelee säännöllisesti, kuinka parantaa tehokkuuttaan, ja mukauttaa toimintaansa sen mukaisesti.

Näiden periaatteiden takana on neljä pääperiaatetta, jotka ovat seuraavat (Cervone 2011; Beck ym. 2001):

- Yksilöiden ja vuorovaikutuksen priorisointi ennen prosesseja ja työkaluja.
- Toimivan ohjelmiston priorisointi ennen kattavaa dokumentaatiota.
- Asiakkaan kanssa yhteistyön teon priorisointi ennen sopimusneuvotteluja.
- Muutokseen vastaamisen priorisointi ennen suunnitelman seuraamista.

Ketterän julistuksen tarkoituksena oli antaa raskaille, dokumenttipohjaisille ohjelmistokehitysprosesseille eriävä vaihtoehto (McKenna ja Whitty 2013). Julistusta pidettiin vallankumouksellisena, mutta McKenna ja Whitty pitivät sitä enemmänkin evolutiivisena ja, että ketteryyden piirteet eivät ole vääjäämättömiä, vaan kompromisseja eri sosiaalisia, kulttuurillisia ja teknologisia valintapaineita vastauksena ympäristöllisiin muutoksiin, missä niitä käytetään. Ketterälle kehitykselle on annettu kolme selkeää kehityksellistä lähdettä:

- Teollisen valmistuksen tekniikat ja lähestymistavat, etenkin Japanissa ilmestyneet toisen maailmansodan jälkeen.
- Inkrementaaliset ja iteratiiviset kehitystekniikat, jotka juontavat juurensa Yhdysvaltojen armeijan ohjelmiin 1950-luvulla.
- Poiminnot perinteisestä projektinhallinnasta.

Koska ketterä kehitys ei ole mikään uusi asia, vaan enemmänkin valintaprosessin tuotos, ketterä kehitys ei ole projektinhallintamenetelmien lopullinen tila (McKenna ja Whitty 2013). Tulevaisuudessa voidaankin puhua jostain muusta kehityksen toimintatavasta, kun ketterää kehitystä kehitetään eteenpäin. Ketterien menetelmien kehittymiseen annetaan kaksi pääsyytä, joista ensimmäinen on niiden käyttöönotto niin ohjelmistokehityksessä, kuin sen ulkopuolella. Toinen syy on muutokset sosiaalikultuurisessa ympäristössä, etenkin nopea teknologinen kehitys ja globalisaatio.

Nykyajan ketterän ohjelmistokehityksen viitekehykset, kuten Scrum, pohjautuvat julistukseen. Kevyet ja ketterät ohjelmistokehitysmenetelmät kehittyivät alun perin 1990-luvun puolivälissä reaktionä raskaisiin vesiputousmalleihin, joiden kritiikkeinä olivat korkea säädöstyly, kurinalaisuus ja mikrohallinta (Juricek 2014).

## **Ketterät periaatteet**

Juricek (2014) esittelee ketterien periaatteiden olevan erilainen tapa toteuttaa IT-projektien hallintaa.

Käyttäjien mukaanotto on ketterässä kehityksessä aktiivisessa roolissa. Käyttäjien tarpeiden ymmärrys vaatii heidän olevan mukana kehitysprosessissa. Jos loppukäyttäjiä ei ole mahdollista ottaa mukaan, on tärkeää, että mukana on kokenut henkilö edustamassa käyttäjiä.

Ketterässä kehityksessä tiimi on voimaannutettu siten, että se voi tehdä itse sitä koskevia päätöksiä. Tiimin on voitava itse yhdessä laatia ja selventää vaatimukset, priorisoida ne, päättää niiden toimittamiseen vaaditut tehtävät ja arvioida vaaditun panoksen määrä (Juricek 2014). Tämä tiimin itseohjautuvuus voidaan jäljittää ketterän kehityksen julistukseen.

Ketterässä kehityksessä vaatimukset voivat muuttua jatkuvasti ja myös nopeasti. Ketteryys toimii sillä periaatteella, että tiimi ei voi valmistautua mitenkään suurella analyysillä uusiin vaatimuksiin tai vanhojen kehittymiseen, sillä tiimi ei voi etukäteen tietää mitä he haluavat (tai mitä asiakas haluaa), ennen kuin tulosta käytetään. Ketterän kehityksen yhtenä perusajatuksena onkin, että vaatimukset eivät ole kiveen hakattuja, mutta kehitysaika on (enemmän tai vähemmän). Vaatimukset ovat ideaalisti näkyviä ja vain juuri niin riittäviä, jotta kehitys ja testaaminen on mahdollista (Juricek 2014).

Vaatimusten kehittyminen johtaa kehitykseen, joka on nopeaa, pientä, inkrementaalista ja iteratiivista. Ketterän kehityksen sykli, joka toistuu jokaisen ominaisuuden kohdalla, koostuu analyysistä, kehityksestä ja testauksesta. Tällainen kehitys antaa seuraavat hyödyt: Pienempi riski (selkeä näkyvyys mitä on tehty projektin aikana), kasvanut arvo (jonkintasoisen hyödyn toimitus aikaisiin), suurempi joustavuus (ohjelmiston käytön ja näkemisen kautta seuraavia iteraatioita voidaan muokata) ja parempi kustannusten hallinta (tilanteessa, jossa projektin budjetti ylitetään, jotain arvoa voidaan saavuttaa) (Highsmith 2002; Ceschi ym. 2005).

Ketterän kehityksen iteraatioissa kehitetyt ominaisuudet tulisi olla valmiita ennen seuraavaa kehityssykliä (engl. *Sprint*) (Fernandez ja Fernandez 2008). Vaatimukset siirretään tuotteen kehityslistaan ja suoritetaan niiden prioriteetin mukaan. Tästä seuraa se, että kaksi iteraatiota ei voi liittyä yhteen muodostaen yhden iteraation.

Ohjelmiston testaaminen on ketterässä kehityksessä jatkuvaa. Testausta suoritetaan koko kehityksen ajan, josta johtuu, että erillistä testausvaihetta ei ole. Suuren skaalan projektitutkimuksessa, projektitiimi onnistui vähentämään virheiden korjaamiseen vaadittua aikaa, virheiden pitkäaikaisuutta ja virheiden hallinnan rasitetta (Talby ym. 2006).

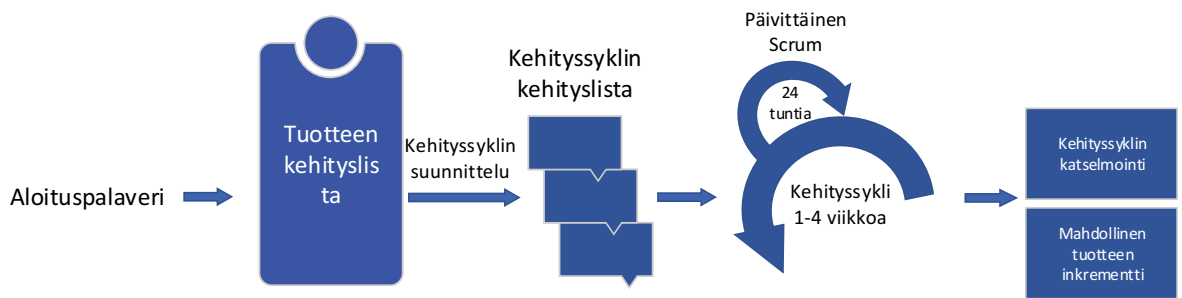
Ketterässä kehityksessä on eritoten tärkeää, että sidosryhmät toimivat kiinteästi yhdessä. Liiketoiminnan ihmisten ja kehittäjien on oltava jatkuvassa yhteistyössä. Parhaat arkkitehtuurit ja selkeimmät toiminnalliset vaatimukset ja niiden suunnittelu syntyvät itseohjautuvien tiimien avulla ja kaikista tehokkain tiedonjakaminen tapahtuu kasvotusten.

### **2.2.1 Scrum**

Kuten jo johdannossa tuli ilmi, Scrum on ketteristä ohjelmistokehityksen menetelmistä suosituin. Yksi Scrum-kehityksen tavoitteista ohjelmistokehityksessä on nopeuttaa projektia ja antaa sille joustavuutta (Permana 2015). Scrum-kehitys on saanut alkunsa Jeff Sutherlandin toimesta vuonna 1993 ja sen tarkoituksena on tulla kehitys- ja hallintamenetelmä, joka seuraa ketterän kehityksen periaatteita (Permana 2015).

Scrum-kehitys rakentuu kolmen pääkomponentin päälle: roolit, prosessi ja artefaktit (Cervone 2011). Rooleista *Scrum Master* päättyy tyypillisesti projektipäällikölle tai tiimin johtajalle. Cervone kertoo kyseisen henkilön olevan vastuussa useammasta tekijästä, joista tärkeimpiä ovat Scrum-kehityksen arvojen ja käytäntöjen toteuttaminen sekä esteiden poistaminen. Scrum-tiimi koostuu tyypillisesti eri kyvykkyyksistä ja se on kooltaan viidestä kymmeneen henkilöä, jonka lisäksi se on itseohjautuva. Itseohjautuvuus voi tarkoittaa, että tiimillä ei ole vakituista johtajaa, vaan se vaihtuu tarpeen vaatiessa (Cervone 2011). Scrum-kehityksessä on vielä lisäksi tuotteen omistaja, joka tietää mitä tulee kehittää, jotta projekti onnistuu ja kuinka kehitysketjun tulisi edetä (Cervone 2011). Tuotteen omistaja laatii tuotteen kehityslistan yhdessä tiimin kanssa, joka koostuu alkuperäisistä vaatimuksista (Permana 2015). Tiimi on vastuussa tuotteen kehityslistan toteuttamisesta ja Scrum Master on vastuussa Scrum-kehityksen esittelystä ja käyttöönotosta tiimille.

Scrum-prosessiin kuuluu viisi pääaktiviteettia: aloituspalaveri, jossa kuvataan kehitettävä järjestelmä (Permana 2015), kehityssyklin suunnittelu, kehityssykli, päiväpalaveri, ja kehi-



Kuvio 4. Scrum-prosessi mukaillen Permana (2015).

tyssykin katselmointi (Cervone 2011). *Kehityssykli* on enimmillään kuukauden kestävä kehitysjakso, jossa kehitettävää järjestelmää työstetään eteenpäin (Permana 2015). Kuviossa 4 esitetään nämä vaiheet.

Kehityssykin sisällä pidetään aluksi kehityssykin suunnitteleva tapaaminen, jossa Scrum-tiimi, Scrum Master, ja tuotteen omistaja tapaavat, ja päättävät tuotteen kehityslistan sekä kehityssykin tavoitteen, eli mitä kehityssykin tulisi saavuttaa. Näiden jälkeen tiimi päättää kehityssykin kehityslistan, eli mitä vaatimuksia kyseisessä kehityssyklissä työstetään. Scrum-prosessin aloitus (engl. *the kickoff*) seuraa vastaavaa kaavaa, sillä erolla, että ryhmä päättää projektin korkeatasoisesta kehityslistasta ja päätavoitteista (Cervone 2011).

Monen projektin kehityssykin aluksi pidetään *päiväpalaveri*, joka kestää maksimissaan 15 minuuttia. Päiväpalaveri pidetään joka päivä Scrum Masterin ja tiimin kesken (Cervone 2011). Tässä palaverissa jokainen tiimin jäsen vastaa seuraaviin kysymyksiin:

- Mitä sinä teit viime päiväpalaverin jälkeen?
- Mitä sinä teet seuraavaan päiväpalaveriin mennessä?
- Mikä estää sinua tekemästä työtäsi?

*Kehityssykin katselmointi* tapahtuu jokaisen kehityssykin päätteeksi. Tässä tapaamisessa on tarkoitus esitellä valmistunutta toiminnallisuutta tuotteen omistajalle (Cervone 2011).

Scrum-kehityksen viimeinen komponentti on artefaktit, joihin kuuluvat tuotteen kehityslista, kehityssykin kehityslista ja edistymiskäyrä. Tuotteen kehityslista on lista järjestelmän vaa-

timuksista, jotka ovat asetettu tärkeysjärjestykseen. Tuotteen kehityslistaa hallitsee ja omistaa tuotteen omistaja (tuoteomistaja, engl. *product owner*). Tuotteen kehityslista on kehityssyklin suunnittelevan tapaamisen tuotos, ja täten sitä ei voi muokata ennen seuraavaa vastaavaa tapaamista. Tiimi arvioi listan jokaista kohtaa joko asiantuntija-arvion kautta tai tekemällä työnosituksen, joiden tarkoituksena on antaa ennusteita, eikä tarkkoja mittauksia kunkin kohdan vaativuudesta. Tämän arvioinnin tarkoituksena on selvittää, kuinka paljon työtä ja aikaa menee kunkin ominaisuuden toteuttamiseen, ja täten selvittää tiimiltä vaadittavan panoksen määrä yhden kehityssyklin aikana (Cervone 2011).

Kehityssyklin kehityslista eroaa tuotteen kehityslistasta siltä osin, että se luodaan Scrum-tiimin toimesta, ja sitä päivitetään ideaalissa tilanteessa päivittäin. Scrum-tiimi voi ilman tuotteen omistajan ohjausta poistaa kohtia kehityslistasta tai lisätä kohtia kehityslistaan, jos he niin päättävät (Cervone 2011).

Edistymiskäyriä käytetään Scrum-kehyksessä kolmella tapaa: kehityssyklin edistymiskäyrä dokumentoimaan kehityssyklin edistymistä, julkaisun edistymiskäyrä dokumentoimaan julkaisun edistymistä sekä tuotteen edistymiskäyrä dokumentoimaan koko projektin edistymistä. Edistymiskäyrän tavoitteena on antaa informaatiota selkokielellisesti (Cervone 2011).

Scrum-kehysten käytössä on myös ongelmansa, jonka takia Ashraf ja Aftab esittelevät oman parannetun versionsa kehuksesta. Ashraf ja Aftab (2017) tunnistavat ongelmia dokumentaatiossa, työn näkyvyydessä, ohjelmiston laadussa ja kehysten käyttöönotossa ylläpitotehtävissä. Dokumentaation puute on ongelma, joka johtuu ajan puutteesta kehityssyklin aikana, kun työ keskittyy muihin kehityssyklin tehtäviin, eikä täten tarkan dokumentaation tekemiseen ole aikaa. Ketterän kehityksen periaatteiden mukaiset jatkuvat muutokset lisäksi heikentävät dokumentaation tekemistä. Tuotteen kehityslistan päivittämisessä on myös ongelmia, jonka lisäksi muutosten jäljitettävyyys jälkikäteen on haasteellista. Lopuksi, eri tiimin jäsenillä on omat dokumentaatiotyylinsä, joka johtaa dokumentaatiostandardin puuttumiseen. Työn näkyvyyden puolesta ongelmia on siinä, että tiimeillä ei ole tiedossa, mitä muut tiimit tekevät samaan aikaan. Suuri määrä kehityksessä olevista toiminnoista eivät ole koherentissa tilassa, eikä tiimiläisillä ole tarpeeksi aikaa tutustua kaikkiin tehtäviin ja pyytää tietoa heille annettusta tehtävästä niin tiimin sisältä, kuin ulkopuolelta. Ohjelmiston laatu voi kärsiä, jos kehitystiimin ja testaajien välinen kommunikaatio on heikkoa, jonka lisäksi kattava regressiotestaus

puuttuu Scrum-kehyksessä. Ashraf ja Aftab (2017) kertovat Scrum-kehityksen olevan vaikeaa ottaa käyttöön ylläpidossa. Heidän mukaan ylläpidon tehtävät eivät liity toisiinsa toisin, kuin Scrum-kehityksen kehityssykliden tehtävät. Ylläpitoa tekevät ryhmät lisäksi toimivat useiden asiakkaiden kanssa, jotka ovat kaukana heistä, eivätkä he kommunikoi usein keskenään kasvotusten.

Ashraf ja Aftab (2017) esittelevätkin *IScrum*-nimisen kehityksen, jonka tavoitteena on korjata Scrum-kehityksen ongelmia. *IScrum* on tarkoitettu tuottamaan korkeatasoista ohjelmistoa pienimmässä mahdollisessa ajassa projekteissa, jotka ovat pieni- ja keskikokoisia. Ashraf ja Aftab ottavat perinteisen Scrum-kehityksen ja muokkaavat sitä keinoihin, jotka ilmenevät vaatimustenhallinnassa (engl. *requirements engineering*). Dokumentaatiota varten rooleihin lisätään tekninen kirjoittaja (engl. *Technical Writer*), jonka lisäksi Scrum Master-rooli muokataan tekemään analyysia sidosryhmien ja tiimin osalta sekä suorittamaan koulutusta tiimille kehitykseen liittyen.

Scrum-osaamisen puute on tunnistettu ongelma, joka voi johtaa tiimin epäonnistumiseen tavoitteen saavuttamisen suhteen (Bootla, Rojanapornpun ja Mongkolnam 2015). Ashraf ja Aftab (2017) lisäävätkin esittelemässään parannuksessaan Scrum-koulutuksen, jossa Scrum Master kouluttaa tiimin jäsenet hyväksikäyttämään kehystä sekä tarvittaessa antaa teknistä koulutusta. Haastetta lisää myös se tekijä, että usein käytettävään malliin lisätään viitteitä muista kehityksistä, jolloin syntyy hybridimalleja, kuten ScrumBan, jonka kaikki ominaisuudet eivät ole hallussa.

Teknisen kirjoittajan roolin myötä dokumentaatiota pyritään parantamaan. Tekninen kirjoittaja on mukana kehitysprosessissa alusta loppuun ja hän voi esimerkiksi huomioida tehtäviä, jotka vaativat lisää dokumentaatiota. Tämän lisäksi tekninen kirjoittaja voi kirjoittaa testitapauksia, jotka palvelevat teknisenä dokumentaationa. Tekninen kirjoittaja on yhteyksissä kehitystiimin ja testaajien kanssa koko kehityssyklin ajan, jotta tiimin yhteistyö on onnistunut tulosten suhteen (Ashraf ja Aftab 2017).

Testauksen osalta testihenkilöstö on *IScrum*-kehityksessä mukana koko kehityssyklin ajan, eikä vain sen lopussa. Testaajat ovat yhteyksissä kehitystiimin kanssa ja he yhdessä päättävät ohjelmiston sisäisen julkaisun, jossa voi olla esimerkiksi puolet kehitetyistä ominaisuuksis-



ta. Sisäisen julkaisun päättämisen jälkeen testaajat pääsevät käsiksi tähän julkaisuun ja täten ulkoisen julkaisun ollessa valmis, sen testaus vie vähemmän aikaa, kuin “normaalissa” Scrumissa. Kehityssyklin lopussa suoritetaan regressiotestaus, jotta varmistetaan, että muutokset eivät ole tuottaneet lisää virheitä, jotka eivät olleet aikaisemmin ohjelmistossa.

I-Scrum-kehyksessä dokumentaation tekemiseen annetaan seuraavat kriteerit:

- Keskustelun puute tai vähäisyys.
- Dokumentaatio välittää projektin tavoitteen.
- Dokumentaatio voidaan muuttaa suoritettavaksi spesifikaatioksi, eli vaatimukseksi, arkkitehtuuriksi ja suunnittelun spesifikaatioksi testien muodossa.
- Tietty kohde/konsepti/vaatimus on vakaa.
- Dokumentaatio on asiakkaan vaatimus.
- Dokumentaatio vaaditaan toimialan säännöstelyn tai sopimuksen takia.

Tuotteen kehitysjonon suunnittelun yhteydessä tekninen kirjoittaja tekee vaatimusanalyysin, niiden spesifikaation, ja kirjoittaa muun dokumentaation annettujen kriteerien mukaan. Jälkeenpäin tekninen kirjoittaja dokumentoi kaiken, jolla on arvoa tiimille ja asiakkaalle (Ashraf ja Aftab 2017).

Päivittäisen palaverin Ashraf ja Aftab ehdottavat siirrettäväksi päivän loppuun sen alun sijaan ja palaverissa tiimin jäsenet vastaavat kahteen kysymykseen: “Mitä saavutit tänään?” ja “Onko työlle esteitä?”. Tämän jälkeen tehtävät suunnitellaan uudelleen siten, että tehtävät, joissa on joko esteitä tai virheitä (engl. *bug*), asetetaan prioriteetissa ensisijaiseksi. Työn alla olevat tehtävät sijoitetaan tauluun, ja täten tiimillä on selkeä käsitys tehtävästään ja auttaa yhteistyössä.

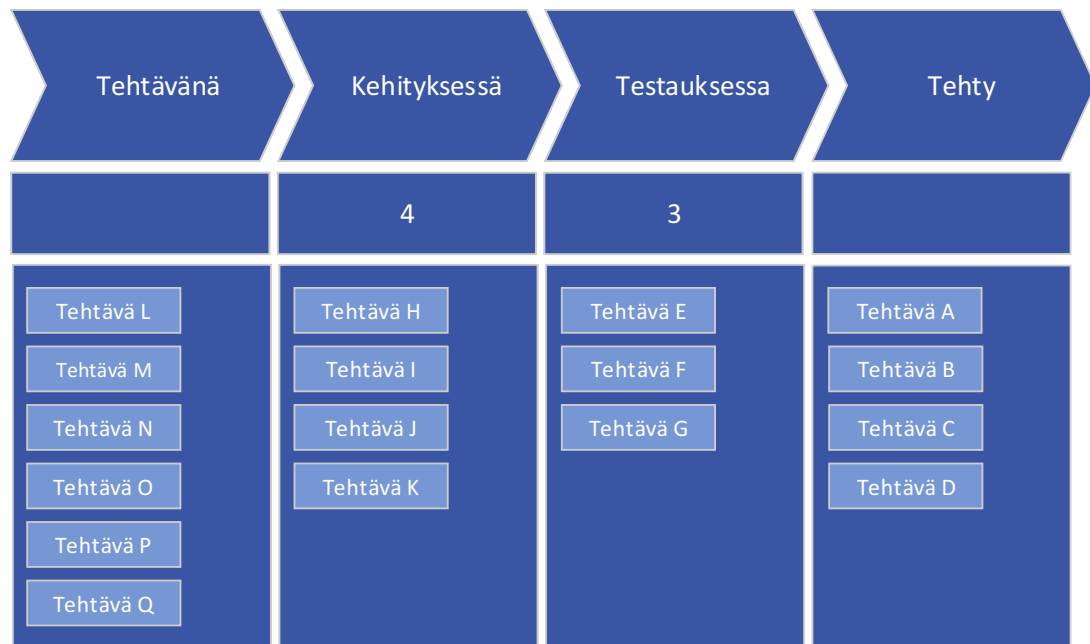
Työn näkyvyyden puolesta kaikki tiimin tekemä työ tulee olla esitettyä tuotteen kehitysjonossa. Kehityksen aikana eri lähteistä tulevat uudet kohteet tulee päivittää kehitysjonoon, jotta asiakkaalla on kuva siitä, mitä on tehty ja mitä tullaan vielä tekemään, sillä asiakkaalla ei ole pääsyä tiimin sisälle kehityssyklin aikana (Ashraf ja Aftab 2017).

### 2.2.2 Kanban

Kanban on toinen hyvin suosittu ketterä menetelmä (*16th State of Agile Report 2022*). “Kanban” tulee japanin kielestä tarkoittaen kylttiä (Ikonen ym. 2011). Kanban-kehiksen on alun perin kehittänyt Taiichi Ohno ja sen juuret ovat 1950-luvun Toyotan tuotannossa (Kirovska ja Koceski 2015). Kanban olikin alkujaan yksi Lean-tuotannon menetelmistä, joista se oli yksi tärkeimmistä työkaluista (Ikonen ym. 2011; Kirovska ja Koceski 2015). Lean-tuotannon viisi periaatetta ovat seuraavat: 1. tuotteen asiakkaalle tuottaman arvon selvittäminen, 2. tuotteen arvoketjun selvittäminen ja selventäminen, 3. nopean ja keskeyttämättömän arvoketjun järjestäminen, 4. asiakkaiden salliminen hankkimaan arvoa toimittajalta ja 5. loistavuuteen pyrkiminen. Lean-hallinta on puolestaan hallinnan menetelmä hallitsemaan yrityksiä mukautumalla todelliseen markkinatilanteeseen organisaatiollisten ja toiminnallisten muutosten kautta. Dekier (2012) antaa seuraavat säännökset, joita tulee seurata:

- Hyvä ilmapiiri työpaikalla: työnantajan tulee luoda työpaikalle positiivinen ilmapiiri.
- Tavoitteiden määrittäminen: työntekijän tulee tietää työn tavoitteet toimiakseen tehokkaasti.
- Kommunikaatio: johtajien tulee määrittää järjestelmä tiedon välittämiseksi.
- Kunnollinen motivaatio: johtajien tulee välttää perinteisiä motivaation menetelmiä, kuten erilaisia “bonus”-järjestelmiä.
- Ihmispotentiaalın hukkaaminen: ulkoisten toimijoiden hyödyntämisen sijaan, toiminnan kehittämisessä tulee keskustella suoraan työntekijöiden kanssa, joilla on ongelmia jonkin asian kanssa.
- Työntekijöiden kehittäminen: johtajien tulee tehdä kaikkensa, minkä he voivat antaa työntekijöille keinoja kehittymiseen.
- Johtajuus: johtajien tulee olla jatkuvasti tietoisia, mitä johtamisen tyyliä he käyttävät alaisten johtamiseen.

Kanban-kehiksen taustalla on seitsemän periaatetta, jotka kuuluvat seuraavasti: 1. hukan vähentäminen, 2. oppimisen vahvistaminen, 3. päättäminen mahdollisimman myöhään, 4. tuottaminen mahdollisimman nopeasti, 5. tiimin voimaannuttaminen, 6. laadun tuottaminen ja 7. kokonaisuuden näkeminen. Kanban on saanut ohjelmistokehityksessä alkunsa David J. Andersonin toimesta (Kirovska ja Koceski 2015). Kanban-kehiksen taustalla on ajatus,



Kuvio 5. Esimerkki Kanban-taulusta mukaillen Kirovska ja Koceski (2015).

että tiimit voivat visualisoida tekemisensä mahdollisesti parantaen työn sujuvuutta (Ikonen ym. 2011; Kirovska ja Koceski 2015). Visualisointia varten tiimit luovat *Kanban-taulun*, jossa tehtävät ovat jaettu eri vaiheisiin (Ikonen ym. 2011; Kirovska ja Koceski 2015). Kanban-taulun sisältöön ei kuitenkaan ole mitään tiettyjä sääntöjä vaan tiimit rakentava taulun itse, kuinka he parhaaksi näkevät (Ikonen ym. 2011). Kanban-taulu voi lisäksi olla joko virtuaalinen (Kirovska ja Koceski 2015) tai fyysinen taulu (Ikonen ym. 2011). Taulussa tehtäviä kuitenkin kuvataan lipukkeilla, joita siirretään vaiheesta toiseen peilaten tuotteen kehitystä (Ikonen ym. 2011). Kuviossa 5 on esitetty esimerkki Kanban-taulusta.

Ikonen ym. kertovat Kanban-kehityksen hyötyjä olevan parannettu työn virtaus, liikatuotannon estäminen, operaatiotason hallinta, visualisoitu aikataulu ja prosessien hallinta ja parannettu reagoitakyky muutoksiin. Täten Kanban-kehityksen käytöllä on mahdollista laskea kuluja, parantaa laatua ja nopeuttaa kehitysaikaa (Ikonen ym. 2011). Kanban-kehitys vaikuttaa projektityön yhdeksään näkökulmaan ja näitä vaikutuksia esitellään seuraavaksi Ikonen ym. (2011) mukaan. *Dokumentaatiota* tehdään vain, jos asiakas tarvitsee sitä ja puoltaa sen tekemistä. *Ongelmanratkaisun* kohdalla ongelmat nähdään nopeasti Kanban-taulusta ja tä-

ten niitä voidaan lähteä ratkomaan välittömästi. *Visualisaatio* tapahtuu jo aiemmin mainitun Kanban-taulun kautta. *Kokonaisuuden ymmärtäminen* tulee Lean-ajattelusta, eikä Kanban itsessään anna tähän työkaluja. *Kommunikaatio* on Ikonen ym. mukaan nopeaa ja runsasta. *Metodin hyväksyntä* tapahtuu Kanban-kehityksen intuitiivisuuden kautta. *Palaute* on kommunikaation tapaan nopeaa ja runsasta, jonka lisäksi Kanban-kehys tukee säännöllisiä tapauksia asiakkaan kanssa. Kanban-kehys ei käytä mitään monimutkaista *hyväksymisprosessia*, vaan paras asiantuntemus on jokaisella kehittäjällä. Koska Kanban on yksi ketterän kehityksen menetelmä, niin tiimin itseohjautuvuus on siinäkin läsnä, joten *työtehtävien valinta* on jokaisen kehittäjän omalla vastuulla.

### **2.3 Projektinhallintamenetelmien tiivistys ja merkitys**

Tässä luvussa esiteltiin nykyisin vallassa olevaa ketterän kehityksen periaatetta ja siitä kaksi yleisintä viitekehystä, Scrum ja Kanban, joita ohjelmistokehitysprojekteissa on käytössä. Jotta ymmärretään, mistä ollaan ohjelmistokehityksessä tultu ja miksi ketterät menetelmät on kehitetty, luvussa esitettiin kaksi perinteistä menetelmää, vesiputousmalli ja V-malli. Menetelmät on hyvä tuntea, jotta ohjelmistokehityksen haasteita (ja onnistumisia) on mahdollista peilata takaisin itse menetelmiin. Menetelmät on hyvä tuntea myös siksi, että meillä on ymmärrys, kuinka ohjelmistokehitystä toteutetaan, kun käytetään jotain tiettyä menetelmää. Menetelmien tuntemus antaa vankan pohjan, kun mietitään mitä haasteita ohjelmistokehitysprojektien hallinnassa on olemassa. Ketterän kehityksen viitekehykset liittyvät oleellisesti, kun tutkitaan ohjelmistokehitysprojektien hallintaa, sillä käytetty viitekehys ohjaa paljon erityisesti ohjelmistokehitystä, mutta myös projektinhallintaa. On tärkeää ymmärtää viitekehykset heikkouksineen ja vahvuuksineen, jotta voidaan ymmärtää miksi ohjelmistokehitystä ja näiden projektien hallintaa tehdään kuten niitä tehdään.

### 3 Aikaisempi tutkimus projektinhallinnan haasteista

Aikaisempi tutkimus on jaettu kahteen kategoriaan: tiettyihin projektinhallintamenetelmiin liittyvä tutkimus, kuten esimerkiksi Scrumiin liittyvä tutkimus, ja ohjelmistoprojektien hallintaan liittyvä tutkimus kattaen projektinhallintaan liittyvät tutkimukset ohjelmistokehitykseen liittyen, jossa esitellään myös yleisellä tasolla olevaa ketterän kehityksen tutkimusta. Projektinhallintamenetelmistä oleva alaluku keskittyy Scrum-kehykseen, sillä se on suosituin ketterän kehityksen menetelmä. Projektinhallintaan liittyvän tutkimuksen alaluvun pääpainona on pandemiasta riippumaton tutkimus ottaen kuitenkin kantaa, mitä haasteita etätyössä voi olla ohjelmistokehitysprojekteissa ja ketterässä kehityksessä.

#### 3.1 Projektinhallintamenetelmiin liittyvä tutkimus

Scrum-kehityksen ongelmista ja mahdollisista haasteista ollaan jo hieman sivuttu luvussa 2.2.1, johon liittyen Cho (2010) on tehnyt tutkivan (engl. *exploratory*) tutkimuksen ketterän ohjelmistokehityksen, kun käytössä on Scrum. Chon tekemässä tutkimuksessa tarkastelun kohteena on kaksi toisistaan erilaista ohjelmistoyritystä: suuria ja tehtäväkriittisiä ohjelmistoja tarjoava (yritys ABC) sekä pieniä ja keskisuuria ohjelmistoja tarjoava (yritys XYZ). Yrityksen ABC tarkastelussa oleva projekti oli kestoaltaan yhdestä kahteen vuotta. Projektien ohjelmistokehitystiimit oli hajautettu kahteen eri sijaintiin. Yrityksen XYZ projekti oli kestoaltaan paljon lyhyempiä, maksimissaan puoli vuotta. Yritys XYZ on näistä kahdesta yrityksestä kokeneempi Scrum-kehityksen käytössä, he ovat käyttäneet sitä useamman vuoden, kun taas ABC oli ottanut Scrum-kehityksen vasta tarkastelussa olleessa projektissa. Yrityksen ABC projektien Scrum-kehityksen käytöstä tuli ilmi seuraavia ongelmia:

- Tiimien jäsenten uudelleenjärjestäminen huomioimatta heidän taitojaan.
- Tiimin itseohjautuvuus ei ole toimivaa.
- Kehittäjät ja testaajat eivät ole keskustelussa keskenään eikä heidän yhteistyö ole toimivaa.
- Tiimin jäsenet eivät tutustu etänä toimiviin tiimin jäseniin.
- Kehityssyklin kehityslistan kohteita ei saada valmiiksi eikä kukaan ota vastuuta myöhässä olevista kohteista.

- Projektipäälliköllä ei ole auktoriteettia kehittäjiä kohtaan.
- Scrum Master ei pysty poistamaan kehittäjien esteitä ja scrum Masterin valinta ei ole kehittäjien mieleen.
- Päivittäinen palaveri ja kehityssyklin suunnitteleva tapaaminen vievät liikaa aikaa.
- Dokumentaation puute voi johtaa ongelmiin projektin muissa osissa ja lisääntyneisiin virheisiin testauksessa.
- Asiakkaat eivät voi osallistua Scrum-tapaamisiin, jonka takia projektipäällikkö viettää liikaa aikaa asiakkaiden tiloissa.
- Scrum-kehityksen käyttöönoton myötä testauksessa on löytynyt kaksi kertaa enemmän virheitä, kuin aikaisemmin.

Tiimien uudelleenjärjestely johti siihen, että jäsenet joutuivat opettelemaan ohjelmiston toimintaympäristön sekä uusia ohjelmistotyökaluja ja -kieliä. Tämä johti siihen, että kehittäminen vei enemmän aikaa ja rahaa. Scrum-kehityksen yksi tärkeimmistä osa-alueista on tiimien itseohjautuvuus, joka saa juurensa ketterän kehityksen julistuksesta. Yrityksen ABC projektissa itseohjautuvuus kuitenkin nähtiin haittana ja kehittäjät toivoivat nimettyä esihenkilöä, joka johtaa tiimiä. Koska ABC oli aikaisemmin käyttänyt vesiputousmallia heidän projekteissaan, kyseessä voi olla pelkästään kokemuksen puutteesta itseohjautuvuuden saralla. Testaaminen oli yrityksessä ABC järjestetty alkuun siten, että jokaisessa tiimissä oli yksi testaaja, mutta tämä järjestely vaihtui yhteen Scrum-tiimiin, jossa oli kaikki testaajat. Tämän Scrum-tiimin tehtävänä oli testata kaikkien yritysten toteuttama lähdekoodi. Ongelmana tässä järjestelyssä oli se, että testaajat eivät olleet tietoisia, mitä ohjelman osia muutokset koskevat, sillä kehittäjät eivät halunneet palata kirjoittamaansa koodiin kehityssyklin jälkeen. Tässä tilanteessa testaaminen kärsii, kun he eivät voi testata kaikkia ohjelmiston alueita, sillä ne voivat olla mahdollisesti jonkin verran virheellisiä. Testauksessa ilmeni lisäksi kaksi kertaa enemmän virheitä verrattuna aikaisempaan. Dokumentaation puute oli yksi syy virheiden lisääntymiseen, sillä ohjelmiston suunnittelusta ja kehityksestä ei tehty teknistä dokumentaatiota, jolloin ominaisuudet, kenttien nimet ja virheviestit ovat erilaisia. Yhteistyötä testaa- jien ja kehittäjien välillä haittasi myös testaa- jien ja kehittäjien sijainnin eroavaisuus. Testaa- jien kehittäjien sijainnin eroavaisuuden lisäksi kehitystiimi oli kahdessa eri sijainnissa, joten myös kehitystiimin yhteistyö koki haittaa. Tällainen tiimin jakaminen johti myös siihen, et- tä toiseen sijaintiin palkatut uudet henkilöt eivät tulleet tutuiksi toisen sijainnin kehittäjille

heidän oman haluttomuudensa takia. Tiimin sisäinen yhteistyö oli haasteellista myös kehityssyklin kehityslistan kohtien myöhästymisessä, sillä kehittäjät eivät ota vastuuta tekemättömistä kohteista. Projektipäällikkö ei voinut myöskään auttaa tässä tilanteessa, sillä heillä ei ollut valtaa kehittäjien työpanoksen suhteen. Ongelma johtuu jälleen Chon mukaan tiimin itseohjautuvuudesta. Projektipäälliköt lisäksi joutuivat matkustamaan asiakkaiden omiin tiloihin, sillä heillä ei ollut pääsyä tiimien Scrum-tapaamisiin. Tämä johti siihen, että projektipäälliköt viettivät suuren osan heidän työajastansa matkustamiseen asiakkaiden luokse ja heidän tiloissaan tapaamiseen. Kehittäjien suhde Scrum Masteriin ei myöskään ollut myönteinen. Koska Scrum Masteriksi oli valittu ei-tekninen henkilö, se ei tuottanut kehittäjiin luottamusta. Scrum Master ei pystynyt poistamaan kehittäjien esteitä, eikä myöskään informoimaan kehittäjiä sovitusta asioista. Esteiden poistaminen johtui erään kehittäjän mielestä riittämättömästä teknisestä osaamisesta. Luottamusta myös kehittäjien kesken laski se tekijä, että kehittäjät eivät saaneet heille sovittuja asioita tehdyksi. Päivittäiset Scrum-tapaamiset olivat osalle kehittäjistä mieleisiä, mutta osa piti niitä turhiksi, jos jotain tiettyä agendaa ei ollut käsiteltävänä. Osa kehittäjistä olivat sitä mieltä, että päivittäiset Scrum-tapaamiset vievät liikaa aikaa ja niitä olisi voinut pitää harvemmin.

Yrityksellä XYZ oli erilaisia haasteita verrattuna yritykseen ABC. Yrityksellä XYZ oli viisi Scrum-tiimiä, joissa oli kolmesta neljään kehittäjää, tuoteomistaja ja Scrum Master. Yrityksessä yksi testaaja vastasi kaikkien tiimien ohjelmistojen testaamisesta. Neljä tiimeistä työskenteli uusien projektien parissa ja viimeinen työskenteli ylläpidon parissa. Yrityksen XYZ haasteita Scrum-kehityksen käytössä olivat seuraavat:

- Yhteistyö ja kommunikaatio tuoteomistajan ja projektipäällikön kanssa oli haasteellista.
- Päivittäiset Scrum-tapaamiset, kehityssyklin suunnittelutapaaminen ja kehityssyklin katselmointi olivat joskus tehottomia.
- Dokumentaation puute aiheutti ongelmia järjestelmän viimeistelyssä.
- Asiakkaan mukaanotto ohjelmiston kehittämiseen oli kaikilta osin haasteellista.
- Tiimien välinen kommunikaatio oli haasteellista.

Kehittäjien keskeinen yhteistyö oli Chon mukaan yrityksessä XYZ hyvällä tasolla, mutta projektipäällikön ja tuoteomistajan yhteistyössä kehittäjien kanssa oli parannettavaa. Projek-

tipäällikkö ei antanut tarpeeksi laadukasta koulutusta kehittäjille Scrum-kehyksen käytöstä. Yrityksessä yksi henkilö tiimissä toimi projektipäällikkönä, tuoteomistajana ja kirjanpitojohtajana. Kyseisellä henkilöllä oli selkeästi liikaa vastuita, sillä hänellä ei ollut aikaa kehittäjille ja hän oli projekteissa pullonkaula. Myös kommunikaatio kyseisen henkilön kanssa oli haasteellista liian monen tehtävän takia. Eri Scrum-tapaamiset oli nähty positiivisena, mutta osa ei nähnyt syytä viettää koko päivää suunnitteluun. Päivittäiset Scrum-tapaamiset lisäksi olivat välillä kestäneet pidempäänkin kuin tavoiteltu 15 minuuttia. Eräs kehittäjä sanoi seuraavasti projektin ja tiimin koosta ketterän kehityksen osalta: “Ketterä kehitys on erittäin vaikeaa isoissa projekteissa. Suurien tiimien on vaikeaa olla ketteriä” (Cho 2010). Kehittäjät ymmärtävät dokumentaation tärkeyden ja myös yrityksessä XYZ dokumentaation puute aiheutti uusien kehittäjien osalta hämmennystä toteutustapojen osalta. Tämän hämmennyksen takia uusien toimintojen kehittäminen ohjelmiston osaan, johon kehittäjä ei ole aiemmin koskenut voi olla vaikeaa ja etenkin uusille kehittäjille, jotka ovat uusia koko projektiin. Ketterässä kehityksessä asiakkaan mukaanotto tulisi olla erittäin tärkeää. Yrityksen XYZ yhden projektipäällikön mukaan asiakkaat olivat mukana päätöksenteossa vasta, kun päätöksenteon prosessi on valmis. Yrityksessä XYZ kuitenkin oli halua ottaa asiakkaat mukaan prosessiin, mutta asiakkaat olivat haluttomia yhteistyöhön. Asiakkaat eivät tieneet mitä he haluavat, joten heillä oli vaikeuksia kommunikoida vaatimuksia toiminnoille, jonka lisäksi annettu informaatio oli epäselvää. Kommunikaatio oli myös ongelmana tiimien välisessä kommunikaatiossa, jolloin tiimit päätyvät tekemään samaa työtä useaan kertaan. Kommunikaatio oli kuitenkin hyvällä tasolla tiimien sisäisesti. Eräs kehittäjä sanoi seuraavasti kommunikaatiosta: “suurimmat kommunikaatio-ongelmat meillä on asiakkaan kanssa, koska he eivät anna tarpeeksi palautetta” (Cho 2010).

Näiden kahden yrityksen kautta nähdään, että Scrum-kehyksen käytössä voi olla hyvin erilaisia haasteita riippuen kehyksen käytön kokemuksen määrästä ja myös siitä, minkälaista ohjelmistoa yritys kehittää. Hyvin tärkeää on, että yritykset antavat kehittäjille tarpeeksi koulutusta Scrum-kehyksen käytöstä, vetovastuussa oleva henkilö tulee valita siten, että hän voi palvella kehittäjiä tehokkaasti sekä auttaa heitä ongelmatilanteissa ja projektin vetovastuussa olevalle henkilölle ei tule antaa liikaa tehtävää, jotta hänellä on aikaa kehittäjille.

Scrum-kehyksen käytössä voi olla haasteena myös käyttäjätarinoiden muuntaminen tuotteen



kehityslistan kohdiksi (Anand ja Dinakaran 2015). Anand ja Dinakaran lisäävät haasteeksi kommunikaatio-ongelmat, kun Scrum-kehystä käytetään hajautetusti, joka nähtiin myös Cho (2010) tekemässä tutkimuksessa. Anand ja Dinakaran kertovat testauksen olevan yksi suurimmista ongelmista Scrum-kehityksessä. Koska ketterässä kehityksessä vaatimukset voivat muuttua kehityksen aikana, ohjelmiston regressiotestaus on erittäin tärkeää, jotta muutokset eivät aiheuta ongelmia jo tehdyissä ohjelmiston osissa. Aika tulee kuitenkin tässä ongelmaksi (etenkin suurissa ja monimutkaisissa ohjelmistoissa), sillä kehityksen edetessä regressiotestauksen määrä paisuu suureksi, jolloin se vie enemmän aikaa (ja myös rahaa). Toinen testauksen ongelma Anand ja Dinakaran mukaan on testaajien joutuminen odottamaan kehittäjiä saamaan kehitys valmiiksi testausta varten. Tämä ilmeni myös yrityksessä ABC, kun Scrum-tiimeissä alkuun olleet testaajat joutuivat odottamaan kehityssyklin päättymistä ennen kuin he pääsivät testaamaan ohjelmistoa, jolloin heillä ei ollut mitään tekemistä pitkään aikaan (Cho 2010). Anand ja Dinakaran (2015) kertovat niin ikään dokumentaation olevan haasteena Scrum-kehityksessä, tosin heidän mukaansa ongelma ilmenee liian monena palaverina. Anand ja Dinakaran sanovatkin ongelman olevan dokumentaation ja tapaamisten määrän tasapainon löytäminen. Moni näistä haasteista on pyritty korjaamaan aikaisemmin esitetyssä IScrum-kehityksessä. Taulukossa 1 on esitetty muita haasteita tai ongelmia Scrum-kehityksen käytössä voi ilmetä (Anand ja Dinakaran 2015).

Anandin ja Dinakaranin esittämissä haasteissa esiintyy etenkin Scrum Masterin ja tuoteomistajan rooli. Heillä tulee olla aktiivinen ote kehityksessä, jotta haasteita ei ilmenisi tai niistä olisi ainakin helpompi toipua.

Scrum-kehityksen käyttöön liittyvät ongelmat eivät kuitenkaan jää tähän, vaan sen käytössä on enemmänkin ongelmia. Kehittäjien tietotaidot liittyen Scrum-kehitykseen ja ketterään kehitykseen, omiin kehitystaitoihin sekä ohjelmiston toimintaympäristöön ovat tunnistettuja ongelmia (López-Martínez ym. 2016). López-Martínez ym. kertovat myös heikentyneen päätöksenteon, lisääntyneen tarpeen sosiaalisille taidoille sekä välttämättömyyden tiimien väliselle kommunikaatiolle ja ketterän kehityksen toiminnan suorituskyvyn noudattamiselle. López-Martínez ym. lisäävät, kuten on jo tullut ilmi, testaamisen hankaluuden, sillä testaaminen tulisi suorittaa saman iteraation sisällä, kun kehitystä suoritetaan. López-Martínez ym. (2016) suorittivat systemaattisen kirjallisuuskatsauksen ketteriin menetelmiin

<b>Haaste tai ongelma</b>	<b>Haasteen tai ongelman kuvaus</b>
Ongelmanratkaisu päiväpalaverissa	Päiväpalavereissa ei pitäisi keskustella ratkaisujen löytämisestä ongelmiin. Keskustelua ongelmanratkaisusta tuli keskustella vain asianomaisten kanssa.
Tehtävien määrittäminen	Itseohjautuvan tiimin tehtävien määrittäminen tasaisesti ei ole todennäköistä, joten Scrum Masterin tulisi seurata rajoitteita saavuttaakseen parempia tuloksia.
Kehityssyklin epäonnistunut uudelleenkäynnistys	Jos kehityssykli peruutetaan, tiimin tulisi välittömästi jatkaa työtä eikä odottaa täydellistä tilannetta.
Tuoteomistajan näkymättömyys	Tuoteomistajan tulisi olla paikalla kehityssykleissä, jotta tiimi voi poistaa epäselvyyksiä ja tuotteen hyväksymistestaus olisi helpompi suorittaa.
Yksittäisten henkilöiden liiallinen työmäärä	Kehityksessä ei tulisi olla yksittäistä henkilöä, joka tekee suuren osan ohjelmiston tehtävistä. Tiimi voi yhdessä tehdä enemmän kuin vaadittu, mutta ei yksittäinen henkilö.
Tuotteen kehityslistan organisointi tiimin toimesta	Tiimin ei tulisi saada organisoida tuotteen kehityslistaa, sillä heillä ei ole tarvittavia tietoja asiakkaan tarpeista, vain tuoteomistaja ja Scrum Master voi organisoida tuotteen kehityslistaa.
Tiimin vaihtuvuus	Tiimin jäsenten vaihtuessa kehityksen aikana, tiimin muodostusprosessi käynnistyy uudelleen.

Taulukko 1. Scrum-kehityksen haasteita ja ongelmia mukailten Anand ja Dinakaran (2015).

ja Scrum-kehykseen siirtymisestä, jonka kirjallisuuskartoituksen tulokset löytyvät taulukossa 2. López-Martínez ym. määrittivät löytyneet eri kategorioihin, joihin löytyneet artikkelit kuuluvat.

Kuten tästä taulukosta nähdään, tutkimusta on tehty hyvinkin paljon keskittyen ihmisiin liittyvään tutkimukseen. Näistä ongelmista lisäksi nähdään, että yleisenä ongelmana on ketterien menetelmien ymmärryksen ja niiden käytön tuen puute. Kommunikaatio on lisäksi ongelma, joka tulee ilmi tutkimuksessa. Kommunikaatio on tullut ilmi haasteena myös tässä tutkielmassa. On selkeää, että organisaatioiden tulee panostaa vielä paremmin tehokkaaseen kommunikaatioon, jotta siitä ei tule ongelmia. Organisaatiokulttuuri tuli ilmi yrityksen ABC toiminnassakin (Cho 2010), sillä he eivät olleet aiemmin käyttäneet ketteriä menetelmiä, jolloin organisaatiokulttuuri ei tukenut Scrum-kehyksen käyttöä.

Organisaatiokulttuuri luontevasti vaikuttaa siihen, kuinka Scrum-kehyksen tiimien itseohjautuvuus nähdään. Akif ja Majeed (2012) löysivätkin kyselytutkimuksen muodossa Scrum-kehyksen olevan liian idealistinen olettaen, että kehityksessä oleva tiimi ja sen jäsenet kykenevät olemaan itseohjautuvia. Tiimin koulutus Scrum-kehyksen käyttöön onkin olennaista, sillä silloin tiimi tietää Scrum-kehyksen käytön ja mitä heidän tulee tehdä (Akif ja Majeed 2012). Akif ja Majeed löysivät tähänkin liittyen, että tiimin kokemus Scrum-kehyksen käytöstä vaikuttaa siihen, kuinka paljon ongelmia he kohtaavat. Tuoteomistajaan ja Scrum Masteriin liittyviä ongelmia löytyy myös. Perinteisten menetelmien aikaisemman yleisyyden takia Akif ja Majeed löysivät, että tuoteomistaja ja Scrum Master keskeyttävät kehittäjien työtä liikaa kysymällä kysymyksiä, jotka eivät ole oleellisia. Scrum Masterin tulee lisäksi määrittää kehityssyklin kesto oikein, jotta kehittäjille ei tule ongelmia koodin laadun kanssa, joka liian kireällä aikataululla on ongelma (Akif ja Majeed 2012). Akif ja Majeed (2012) löysivät dokumentaation tulevan ongelmaksi silloin, kun vaatimuksia muutetaan sähköpostien välityksellä, eikä muutoksia tehdä tuotteen kehityslistaan. Tuoteomistajan tulee olla aktiivinen päivittämässä tuotteen kehityslistaa, jotta kehittäjillä on selkeästi tiedossa, mitä tulee kehittää.

<b>Ongelmat ketterään kehitykseen siirtymisessä</b>	<b>Kategoria</b>
Organisaatiokulttuuri ei tue ketteriä menetelmiä	Organisaatio (3 paperia)
Kapasiteetin puutos organisaatiokulttuurin muutokseen	Organisaatio (2 paperia)
Organisaation ongelmat	Organisaatio (2 paperia)
Johdon tuen puuttuminen	Organisaatio (1 paperi)
Ulkoinen paine käyttämään perinteisiä menetelmiä	Organisaatio (2 paperia)
Yhteistyön ja kommunikaation puute asiakkaan kanssa	Ihmiset (2 paperia)
Tuoteomistajan ja asiakkaan koulutuksen puute	Ihmiset (1 paperi)
Tiimin koko	Ihmiset (4 paperia)
Tiimin säätämättömyys	Ihmiset (1 paperi)
Työkalujen määrä	Ihmiset (1 paperi)
Tiimin vaihtuvuus	Ihmiset (2 paperia)
Ketterien menetelmien käytön kokemuksen puute	Ihmiset (2 paperia)
Koulutetun henkilöstön saatavuus	Ihmiset (2 paperia)
Tehokkaan kommunikaation puute	Ihmiset (3 paperia)
Ketterien arvojen ymmärryksen puute	Ihmiset (1 paperi)
Heikko ja toimimaton koulutus	Ihmiset (1 paperi)
Muutosvastarinta	Ihmiset (2 paperia)
Päätöksiin heikko sitoutuminen	Ihmiset (1 paperi)
Asiakkaan jatkuva osallistuminen	Ihmiset (1 paperi)
Projektin koko	Projekti (3 paperia)
Ketteryyden aste	Prosessi (1 paperi)
Vastamallit	Prosessi (1 paperi)

Taulukko 2. Ketteriin menetelmiin ja Scrum-kehikseen siirtymisen mahdollisia ongelmia (López-Martínez ym. 2016)

## 3.2 Ohjelmistoprojektien hallintaan liittyvä tutkimus

Demir (2009) suoritti kyselytutkimuksen kartoittamaan, missä osa-alueissa ohjelmistoprojekteissa löytyy haasteita. Tutkimus suoritettiin vuoden 2007 alussa, ja siihen tuli 78 vastausta. Vastaajien kokemus oli vaihtelevaa, niiden joukossa oli niin sovelluskehittäjiä eri rooleissa, kuin myös johtamisasemassa olevia. Kysely oli toteutettu siten, että henkilöt vastasivat siihen edelliseen projektiin peilaten. Projektin taustatietoa Demir kysyi projektiin osallistuvan tiimin koon puolesta, projektin koon lähdekoodin rivien määrän puolesta sekä organisaatiotyypistä, joka jakautui kyseisessä tapauksessa julkisen (kyselyssä käytetty termiä "Government"), yksityisen (kyselyssä käytetty termiä "Commercial") ja julkisen puolen sopimukseen pohjautuviin. Tämän kyselyn kokonaisosuudet näkyvät taulukossa 3.

*Laajuuden hallinta* oli kaikissa projektityypeissä kolmen yleisimmän joukossa ja se vaihteli 61,5 ja 41,7 % välillä.

*Vaatimusten hallinta* puolestaan löytyi kaikkien projektityyppien kahden yleisimmän haasteellisen alueen joukossa osuuden vaihdellen 61,1 ja 45,8 prosenttiyksikön välillä.

*Projektin suunnittelu ja arviointi* riippui suuresti projektityypistä. Julkisen puolen projekteista puolessa projekteissa tämä oli haasteena, kun taas projektit, joissa oli vähemmän kuin 20 tuhatta koodiriviä, suunnittelu ja arviointi oli haasteena vain 30,8 prosentissa projekteista.

*Kommunikaatiota* on pidetty Demirin mukaan tekijänä, mitä ei ole otettu mukaan vastaavissa kyselyissä, eikä hän odottanutkaan sen olevan ongelmana pienemmissä organisaatioissa. Kyselyn organisaation koon vaihtoehtojen 1–10 ja 11–100 välillä ei ollut eroavaisuuksia, eikä myöskään organisaatiotyypin välillä. Sopimusprojekteissa puolestaan kommunikaatio oli haasteellista puolessa projekteista. Pienen koon ohjelmistojen osalta kommunikaatio ei ollut niin haasteellista, ja tämän alueen osalta alhaisin vastausprosentti oli 21,3.

*Henkilöstön palkkaus* vaihtelee organisaatiotyypin mukaan, onko tässä alueessa ongelmia. Julkisen puolen organisaatioissa palkkaus on ongelmana puolessa projekteista, kun taas yksityisen puolen organisaation projekteissa vastaavaa on vain neljänneksessä.

*Projektin monitorointi ja kontrollointi* oli ongelmana 33,3–22,2 % projekteista riippumatta projektityypistä, mutta organisaation tyyppi jälleen näyttää eronsa, kun julkisen puolen (mu-

<b>Projektinhallinnan osa-alue</b>	<b>Vastausten osuus</b>	<b>Vastausten määrä</b>
Laajuuden hallinta	52,6 %	41
Vaatimusten hallinta	51,3 %	40
Projektin suunnittelu ja arviointi	41,0 %	32
Kommunikaatio	38,5 %	30
Henkilöstön palkkaus	33,3 %	26
Projektin monitorointi ja kontrollointi	28,2 %	22
Riskienhallinta	26,9 %	21
Tekninen monimutkaisuus	26,9 %	21
Sidosryhmien osallistuminen	25,6 %	20
Johtajuus	25,6 %	20
Konfiguraatioiden hallinta	25,6 %	20
Organisaation sitoutuminen	24,4 %	19
Laadullinen kehitystyö	23,1 %	18
Tiimityö	21,8 %	17
Riskien arviointi	19,2 %	15
Projektipäällikkö	14,1 %	11
Muut	10,3 %	8
Tukitoimet	9,0 %	7
Projektissa ei ongelmia	2,6 %	2

Taulukko 3. Ohjelmistoprojektien hallinnan haasteista tehdyn kyselyn tulokset mukailien Demir (2009).

kaan lukien sopimusprojektit) projektien osuus on 33,3 % ja yksityisen puolen projektien osuus 22,2 %. Henkilöstön koko ei vaikuta juurikaan haasteellisuuteen, 1–10 ja 11–100 henkilön projekteissa osuudet ovat 28,1 ja 29,3 % tässä järjestyksessä, ja sama toistuu pienten ja keskisuurten sovellusprojektien osalta, kun osuudet ovat 30,8 ja 27,4 %.

*Riskienhallinta* vaihteli 38,5 ja 24,2 % välillä, ja pienten sovellusprojektien osuus oli suurempi, kuin muissa. Muissa kategorioissa osuudet olivat lähes identtisiä, 24,2 ja 27,8 % välillä.

*Teknisen monimutkaisuuden* osuus oli samankaltaista pienissä ja keskisuurissa sovellusprojekteissa, osuuksien ollen 23,1 ja 25,8 %. Demir pitää mahdollisena sitä, että tämä johtuu vastauksien vähäisyydestä pienten sovellusprojektien osalta, ja hän uskoo, että tulokset vaativat lisätutkimusta niiden validoimiseen. Julkisen puolen projektit ovat vapaampia tämän alueen osalta, kun osuudet vaihtelivat kokonaisuudessaan 34,1 ja 23,5 % välillä.

*Sidosryhmien osallistuminen* oli haasteellisena eniten julkisen puolen projekteissa osuuden ollen 25 %, kun taas yksityisen puolen projekteissa vastaava oli 19,4. Kokonaisuudessaan osuudet vaihtelivat 38,9 ja 15,4 % välillä.

*Johtajuuden* osuus oli kategorioista korkein julkisen puolen sopimusprojekteissa ja alhaisin pienissä sovellusprojekteissa (7,7 %). Keskisuurissa sovellusprojekteissa osuus oli puolestaan 28,9. Osuudet olivat suhteellisen läheisiä muissa kategorioissa niiden ollen 28,9 ja 19,4 % välillä. Kokonaisuudessaan osuudet vaihtelivat 38,9 ja 7,7 % välillä.

*Konfiguraatioiden hallinnan* osuudet vaihtelivat 33,3 ja 15,4 prosenttiyksikön välillä, eikä kategorioiden välillä ollut suurempia eroavaisuuksia, mutta osuus oli suurempi keskisuurissa sovellusprojekteissa verrattuna pieniin (25,8 ja 15,4 %).

*Organisaation sitoutuminen* oli korkeinta julkisen puolen sopimusprojekteissa ja alhaisin yksityisen puolen projekteissa. Henkilöstön määrän osalta 11–100 henkilön projekteissa osuus oli hieman suurempi (24,4 %), kuin 1–10 henkilön projekteissa (18,6 %). Osuudet näkivät suurehkoa vaihtelua niiden ollessa 44,4 ja 13,9 % välillä kokonaisuudessaan.

*Laadullinen kehitystyö*-termiä käytettiin tutkimuksessa laadunvarmistuksen sijaan jälkimmäisen ollessa erillinen prosessi yrityksissä Demir mukaan. Osuudet vaihtelivat 33,3 ja 12,5

% välillä sen ollessa korkein julkisen puolen sopimusprojekteissa ja alhaisin julkisen puolen organisaatioissa.

*Tiimityön* osuudet vaihtelivat 30,8 ja 15,6 % välillä ja se oli korkeinta pienissä sovellusprojekteissa, kun taas keskikokoisissa projekteissa osuus oli 19,4 %. Organisaatiotyyppi ei vaikuttanut osuuksiin suuresti (16,7–25,0 %) ja henkilöstön osalta osuus oli 15,6 % 1–10 henkilön projekteissa ja 22,0 % 11–100 henkilön projekteissa.

*Riskien arvioinnissa* päästään jo alle viidennekseen osuuksien osalta. Osuudet vaihtelivat tämän osalta 30,8 ja 12,5 % välillä, joista suurin osa oli alhaisia. Pienten ja keskisuurten sovellusprojektien välillä osuudet ovat suuria niiden ollessa 30,4 ja 16,1 % tässä järjestyksessä.

*Projektipääällikkö* on ensimmäinen alue, jossa eri kategorioiden välillä päästään osuuksissa nolnaan prosenttiin vaihtelun ollessa 22,2 ja 0 %. Nollan prosentin osuus löytyy pienissä sovellusprojekteissa ja myös 1–10 henkilön projekteissa osuus oli pientä, 6,3 %. Demir antaa mahdollisen selityksen tälle, että pienissä projekteissa projektipääällikkö pääsee enemmän mukaan, jolloin hänen puolestansa ongelmia ilmenee vähemmän.

*Muut* haasteet joita vastaajat mainitsivat koostuivat seuraavista: korkea vaihtuvuus niin toimittajan, kuin tiimin osalta, priorisointi, mikrohallinta (engl. *micromanagement*), hajautettu rahoituksen kontrollointi, laitteistokonfiguraatiot, avainhenkilöiden menetys, ja alirahoitettu ohjelmiston turvallisuuden kehitys.

Vain kahdessa projektissa ei ilmennyt ongelmia ja niistä molemmat olivat 1–10 henkilön projekteja. Muiden kategorioiden osalta projekteista toinen oli pienikokoinen sovellusprojekti ja toinen keskisuuri, toinen oli yksityisen organisaation projekti ja toinen julkisen puolen.

Projekteista viisi olivat sellaisia, *joissa oli 101 tai enemmän henkilöä*, joista kolme oli keskisuurta sovellusprojektiä ja kaksi suurta. Yksi projekteista, oli haasteellinen kaikissa alueissa, jossa kehitettiin suuren skaalan hajautettua simulaatiota julkisen puolen sopimusprojektissa. Toinen projekteista sisälsi yli 2 miljoonaa koodiriviä, jossa kehitettiin yksityisellä puolella verkkopohjaista reaaliaikaista ohjelmistoa, ja se oli haasteellinen vain sidosryhmien osalta.

Kolme projektia sattuivat kategoriaan *projektit, joissa yli 2 miljoonaa koodiriviä*, eikä tästä kategoriasta Demir anna minkäänlaisia johtopäätöksiä kategorian suppeuden takia.



Raharjo ja Purwandari (2020) tutkivat ketterien projektien hallintaa systemaattisen kirjallisuuskatsauksen muodossa, jonka seurauksena he löysivät projektinhallinnan alueita, jotka olivat haasteellisia. Tämän tutkimuksen projektinhallinnan alueet perustuivat PMBOK:n tietoihin. Raharjo ja Purwandari tekemän kartoituksen perusteella haasteellisin alue projektinhallinnassa oli sidosryhmien hallinta kahdeksalla paperilla. Seuraavaksi yleisin alue oli resurssien ja kommunikaation hallinta yhdellä vähemmällä paperilla. Seuraavat kolme olivat integraation hallinta, laajuuden hallinta ja aikataulun hallinta, neljän, kahden ja kahden paperin muodossa. Demir (2009) kyselytutkimukseen verrattuna samankaltaisuutta löytyy laajuuden ja kommunikaation hallinnan kanssa, kun ne olivat vielä Demir tekemän kyselyn aikaan yleisiä haasteita. Sidosryhmien osalta kuitenkin tulee suuri ero. Eroa voi selittää sillä, että Raharjo ja Purwandari ottavat huomioon koko alueen, kun taas Demir on muotoillut tekijän sidosryhmien osallistumisena. Myös ajanjakso voi vaikuttaa tuloksiin, Demir suoritti kyselyn vuonna 2007, kun taas Raharjo ja Purwandari tutkimuksen vanhin paperi aiheen osalta oli vuodelta 2015.

Ketterästä kehityksestä ovat niin ikään Santos, Figueiredo ja Marques (2023) tehneet tutkimusta tutkien, mitä haasteita ketterän kehityksen käytänteissä on ollut pandemian kontekstissa. Heidän tutkimuksensa kohteena oli iso Brasilian valtion omistama ohjelmistoyritys, jolla on käytössään ketteriä käytänteitä. Santos, Figueiredo ja Marques suorittivat tutkimuksen haastatteluiden muodossa, joiden kohteena oli viisi yrityksen tiimiä (ja täten projektia). Projektien ajankohdat vaihtelivat huhtikuusta 2019 joulukuuhun 2021 päättyneiden osalta, tutkimuksen kirjoitushetkellä yksi projekti oli vielä kesken, joka oli alkanut kesäkuussa 2021. Näiden ajankohtien perusteella pandemian vaikutus on hyvinkin voinut näkyä projektien hallinnassa. Pandemian vaikutus on näkynyt oletettavasti etätyöhön siirtymisellä. Yritys siirtyi kokonaan etätyöhön maaliskuun puolella välissä vuonna 2020 ja hybridimuotoon lokakuussa 2021. Santos, Figueiredo ja Marques löysivät jo tähänkin mennessä tutun haasteen, kommunikaation. Kommunikaatio nousee jälleen haasteeksi, kun työskennellään etänä, sillä yrityksen tiimit ovat tottuneet työskentelemään samassa tilassa, jolloin kommunikaatioon on matala kynnys ja sitä voidaan helposti suorittaa. Tutkimuksen yrityksen tilannetta tekee vaikeammaksi se, että he joutuivat mukautumaan etätyöhön, jolloin tekniset ongelmat olivat yleisiä. Tämä tietenkin vaikuttaa projektinhallintaan, jos teknisten ongelmien takia ei voi esimerkiksi ajankohtaista tilannetta saada helposti selville ja ongelmien ratkaisu

vie aikaa pois kehityksestä, jolloin se voi vaikuttaa projektin aikatauluun. Henkilöstön vaihtuvuus tunnistettiin ongelmana, josta Santos, Figueiredo ja Marques sanovatkin seuraavasti: “pandemia itsessään ei lisännyt vaihtuvuutta; . . . ammattilaisten päivittäisten kontaktin puute ja riippuvuus teknologisista resursseista vaarantavat lisää palveluvaltuuttamista ja tiimin muuttamista”. Ketterän kehityksen käytänteisiin vaikutuksista Santos, Figueiredo ja Marques lisäävät vielä projektin vaiheen olevan isossa roolissa, kuinka etätyö voi vaikuttaa projektinhallintaan. Projektin alussa tehtävä tarvekartoitus voi olla haastavaa tehdä etänä mikä saattaa johtaa projektin epäonnistumiseen (Santos, Figueiredo ja Marques 2023). Projektinhallinnan osalta etätyön vaikutukset ovat näkyneet tässä tutkimuksessa siirtymisessä digitaalisiin palveluihin, palaverit pidettiin jonkin kommunikaatiosovelluksen kautta ja ketterän kehityksen artefaktit ym. digitoitiin (Santos, Figueiredo ja Marques 2023).

Marnada ym. (2022) tekivät ketterästä kehityksestä systemaattisen kirjallisuuskatsauksen tutkien, kuinka ketterässä projektinhallinnassa projektin laajuus ja muutokset ovat olleet haasteellisia. Marnada ym. kategorioivat 18 haastetta seitsemään eri kategoriaan. Kategoriat ja haasteet ovat esitettyinä taulukossa 4. Tässä tutkimuksessa jälleen kommunikaatio esiintyy haasteena, joten on selkeää, että se on yleisesti ottaen hyvin yleinen haaste etätyössä. Yksi ketterän kehityksen periaatteita on dokumentaation jättäminen taka-alalle, joten ei ole ihme, että se Marnada ym. tekemän tutkimuksen mukaan on haaste.

Hajautetusta ohjelmistokehityksestä Da Silva ym. (2010) tekivät systemaattisen kirjallisuuskatsauksen tutkien, mitä haasteita, parhaita käytäntöjä, malleja ja työkaluja siihen liittyen on olemassa. Da Silva ym. löysivät yhteensä 30 haastetta aiheeseen liittyen. Vähintään kymmenessä tutkimuksessa esiintyneet haasteet sekä niiden esiintymistiheys on listattuna taulukossa 5. Kuten taulukosta nähdään, hajautetuissa ohjelmistoprojekteissa esiintyy erilaisia haasteita, kuin normaalissa, ei-hajautetussa, ohjelmistokehityksessä. Tutkimuksesta voidaan kuitenkin nähdä, että esimerkiksi kommunikaatio nousee suurimmaksi haasteeksi, kun se Demir (2009) tekemässä tutkimuksessa oli niin ikään haasteellinen tekijä. Eri tietotasoja voidaan pitää myös joksenkin samankaltaisena tekniseen monimutkaisuuteen, jos verrataan Demirin tutkimukseen ohjelmistokehitysprojektin haasteista.

Kotoa tehtävän työn (engl. *work-from-home*, *WFH*) vaikutusta ohjelmistoprojekteihin on tutkittu muun muassa Nguyen-Duc ym. (2022) toimesta. Tässä tutkimuksessa heillä on tarkoi-

<b>Kategoria</b>	<b>Haasteet</b>
Ihmiset ja organisaatio	Laajuuden ja muutosten hallintaan ei tarkoitettua roolia Ei selvää menettelyä laajuuden ja muutosten hallintaan Epäselvät roolit ja vastuut tiimin jäsenten välillä Aikaa vievä byrokratia
Käyttäjien vaatimusten priorisointi	Muutoksen prioriteetin määrittämisen menettelyn puute
Laajuuden ylittävät vaatimukset	Kohtuuttomat ja suunnittelemattomat muutospyyntö Huonosti määritellyt tavoitteet Kehityssyökiien epäselvät valmiusasteet Dokumentaation puutteet versiohistoriasta Tuotteen heikko tietämys
Työkalut ja prosessi	Työkalujen puute versiohallintaan Työkalujen puute laajuuden hallintaan Uudemman teknologian saattamattomuus muutospöynnön käsittelyyn
Tuotteen kehityslista	Jokaisen iteraation selkeän vaiheen puute
Kommunikaatio ja koordinaatio	Jäsenten ja sidosryhmien heikko osallistuminen Kommunikaation puute muutosten yhteydessä
Kulttuuri ja käyttäytyminen	Tiimin jäsen tottumaton muutokseen Vastenmielisyydet muutokselle

Taulukko 4. Ketterän kehityksen laajuuden ja muutosten haasteita (Marnada ym. 2022).

Haaste	Tutkimusten määrä
Tehokas kommunikaatio	34
Kulttuurilliset erot	31
Koordinaatio	23
Aikavyöhykkeiden erot	19
Luottamus	13
Prosessien, säännösten ja standardien asymmetria	13
Fyysinen etäisyys	13
IT-infrastruktuuri	13
Eri tietotasot tai tiedon jakaminen	11
Seuranta ja hallinta	10
Yhteistyö	10

Taulukko 5. Hajautetun ohjelmistokehityksen haasteita (Da Silva ym. 2010).

tuksena selvittää, kuinka WFH on vaikuttanut projektinhallintaa ja ohjelmistotuotantoa. Kyselyssä on kyselytutkimus, jonka kohderyhmänä on ne ohjelmistokehittäjät, jotka vaihtoivat toimistotyön kotoa tehdyksi työksi koronaviruksen takia. Projektinhallinnan osalta tutkimuksessa Nguyen-Duc ym. kysivät neljää asiaa: 1. projektin tarvitseman kykyjen löytäminen, 2. organisaation kulttuurin/ilmapiiirin säilyttäminen, 3. projektin resurssien, aikataulun, riskien ja saavutusten suunnittelu ja 4. ymmärrys siitä, kuka tekee mitään projektissa. Projektien aikataulutuksen suhteen vastauksista ilmeni, että 47,2 % projekteista olivat kokeneet viivästyksiä työtapamuutoksen takia. 13,4 % vastaajista koki merkittäviä viivästyksiä, kun taas 3 % vastaajista koki projektin lakkautuksen kokonaan. 25 % vastaajista puolestaan ei nähnyt vaikutusta projektin aikatauluun. Erään vastaajan mukaan työtavan vaikutus projektinhallintaan on sekoitettua siten, että tiimin ja tehtävien tietoisuus ei ole ongelma, mutta projektin suunnittelu on puolestaan haasteellisempaa, kuin aikaisemmin (Nguyen-Duc ym. 2022).

### 3.3 Aikaisemman tutkimuksen tiivistys ja merkitys

Tässä luvussa on käsitelty mitä haasteita Scrum-kehityksen käytössä ja käyttöönotossa on ilmennyt niin perinteissä toimistotyössä, kuin hajautettuna. Luvussa käsiteltiin lisäksi tut-

kimusta ohjelmistokehitysprojektien hallinnan haasteista niin ennen pandemiaa, kuin myös pandemian aikana ja etenkin pandemian aikaisessa etätyössä. Kuten luvussa on huomattu, dokumentaation puute, käytetyn viitekehityksen koulutuksen puute ja heikosti valitut projektin vetäjät (projektipäällikkö ja tuoteomistaja) ovat yleisiä haasteiden lähteitä projektinhallinnassa. Nämä tekijät liittyvät käytettyyn viitekehitykseen, mutta myös yleisesti projektinhallintaan, joten on hyvä huomioida tämä yhteys. Dokumentaation puute haasteiden taustalla pohjautuu selkeästi siihen ketterän kehityksen periaatteeseen, että toimivaa tuotetta priorisoidaan dokumentoinnin sijaan. Tämän tutkielman kannalta on erittäin tärkeää tietää, mitä aiheesta tiedetään jo, jotta tutkielman tuloksia voidaan verrata aikaisempaan tutkimukseen. Jotta analyysin tulosta voidaan pohtia tehokkaasti, on myös tärkeää tuntea, mitkä haasteet johtuvat pelkästään käytetystä viitekehityksestä. Aikaisemman tutkimuksen tuntemus auttaa myös siinä, mitä uutta tietoa tämä tutkielma tuo tutkimusalalle ja käytäntöön.

## 4 Tutkimusasetelma

Tässä luvussa esitellään ensiksi, kuinka kirjallisuuskatsausta on tehty esittäen muun muassa kirjallisuuskartoituksen tekemistä. Luvussa esitellään lisäksi tutkielman empiirinen osa.

### 4.1 Tutkimusmenetelmä

Tutkielmassa käytettiin hyväksi kirjallisuuskatsausta tutkimaan, mitä aiheesta on jo tutkittu ja kyselytutkimusta hankkimaan ajankohtaista ja merkityksellistä informaatiota.

#### 4.1.1 Kirjallisuuskatsaus

Ohjelmistotekniikan systemaattisten kirjallisuuskatsausten tekemiseen on omat ohjeistuksensa (Keele ym. 2007). Keele ym. antavat kolme vaihetta kirjallisuuskatsausten tekemiseen: suunnittelu, suoritus ja raportointi. Suunnitteluvaiheessa varmistetaan katsauksen tekemisen aiheellisuus, täsmennetään tutkimuskysymys sekä suunnitellaan ja arvioidaan katsausprotokolla. Toteutusvaiheessa tunnistetaan tehty aiempi tutkimus, valitaan ensisijaiset tutkimukset, tehdään tutkimusten laadunvarmistusta, luodaan kirjallisuudesta aineisto ja lopuksi yhdistellään se. Raportoinnissa lopuksi määritellään, kuinka katsaus levitetään sekä formatoidaan tutkimusraportti.

Tässä tutkimuksessa on mukailtu edellä kuvattua systemaattisten kirjallisuuskatsausten toteutusta. Tutkielman kirjallisuuskatsauksen lähteitä etsittiin Google Scholar -palvelun avulla sekä Scopus-tutkimusportaalista. Google Scholar -palvelusta lähteitä etsittiin seuraavilla hakulauseilla: “software project management challenges”, “agile software project management challenges” sekä lisäämällä näihin lauseisiin sanoja “covid” ja “pandemic”. Scopus-tutkimusportaalin hakulauseke oli seuraava: “Project management” AND “software engineering” AND ((distributed OR distance OR remote) AND (team OR work)), joka tuottaa 23 dokumenttia, kun tuloksia rajataan vuosiin 2020–2023 ja rajoittamalla tutkimusalat seuraaviin: *Computer Science*, *Engineering* ja *Decision Sciences*, kun etsitään otsikoista, tiivistelmistä ja avainsanoista. Jos näistä tuloksista rajataan pois tutkimukset opetuksesta, eli lisäämällä hakulausekkeeseen “AND NOT education”, haku tuottaa 15 dokumenttia. Näistä

15:sta dokumentista muutamat olivat joko maksullisia artikkeleita tai osa jotain maksullista kirjaa. Näistä dokumenteista tunnistettiin tutkimuksen kannalta oleellisiksi vain kolme dokumenttia, muut joko eivät käsitelleet projektinhallintaa, tai ne käsitelivät projektinhallintaa sellaisesta näkökulmasta, joka ei vastannut tutkimuksen tavoitteita. Toinen hakulause, jota hyödynsin oli seuraava: “project management” AND (remote OR distance OR distributed) AND “software engineering” AND (challenges OR issues), mutta tämä tuotti usean saman lähteen, kuin aiempi ja siinä oli samat ongelmat, kuin aiemmassa. Hyödynsin näitä hakulausekkeita myös Google Scholar -palvelun haussa, joka tuotti pari uutta lähdettä aikaisempaan verrattuna. Scrum-kehikseen olevista haasteista on olemassa enemmänkin tutkimuksia, mutta tässä tutkielmassakin on nähty, kuinka samankaltaisia haasteita on olemassa sen käytössä ja käyttöönotossa.

Näiden hakujen tulokset kannattavat väitettä, ettei tutkimusta koronaviruspandemian vaikutuksista ohjelmistokehitysprojektien hallintaan ole juurikaan tehty. Koska tutkimuksen määrä on vähäinen, tämä tutkielma tuo arvokasta tietoa tutkimusalalle pelkästään sen uutuuden takia.

#### **4.1.2 Kyselytutkimus**

Tutkielman empiirinen osa suoritettiin kyselytutkimuksena. Kyselytutkimuksen ideana on tutkia tiettyä populaatiota joko selittävien mallien kehittämiseen tai varmistamaan olemassa olevaa tietoa ja sitä voidaan käyttää niin määrällisesti, kuin laadullisesti (Molléri, Petersen ja Mendes 2020). Kyselytutkimuksen instrumentiksi valittiin sähköinen kyselylomake, joka jaettiin internetin välityksellä. Kyselytutkimuksen avulla oli mahdollista saada ajankohtaista informaatiota ohjelmistokehitysprojekteista ja niiden hallinnasta. Ajankohtaisen tiedon hankkiminen on tärkeää, jotta tutkimuksessa ei pohjauduta ainoastaan olemassa olevaan tutkimukseen, joka ei välttämättä pidä enää paikkaansa. Kyselyn kohderyhmänä olivat ohjelmistokehitysprojekteissa mukana olleet henkilöt, eli vastauksia ei haettu pelkästään projektipäälliköiltä vaan myös muun muassa ohjelmistokehittäjiltä, suunnittelijoilta ja testaaajilta. Jotta vastaukset liittyivät nimenomaan koronaviruspandemian aikaisiin projekteihin, projektit rajattiin aikavälille 2020-2023.

Kyselytutkimuksen valinta menetelmäksi on perusteltua sekä käytännöllisestä lähestymistavasta, että tieteellisesti. Kyselytutkimuksen avulla voidaan vastata luvussa 1 esitettyyn tutkimuskysymykseen, koska sen avulla voidaan kysyä ohjelmistokehitysprojekteissa mukana olleiden henkilöiden mielteitä sekä rajata saatava tieto juuri koronaviruspandemian aikaan. Aiempaa tutkimusta on myöskin tehty nimenomaan kyselytutkimusten kautta, joten on luontevaa, että tässäkin tutkielmassa käytettiin hyväksi kyselytutkimusta. Kyselytutkimuksen kautta saadaan käytäntöön arvokasta informaatiota, kuinka ohjelmistokehitysprojektien hallinta sujuu. Tieteellisesti kyselytutkimus antaa myös oivaa näkemystä, sillä sen avulla voidaan nähdä, missä haasteita ilmenee ja täten saada tietoa tutkimusalalle sekä nähdä, mitä kannattaa tutkia lisää.

Molléri, Petersen ja Mendes (2020) antavat kyselytutkimuksen prosessin, johon kuuluu 10 tehtävää. Ensimmäiset kuusi ovat sen suunnittelua ja ne kuuluvat seuraavasti: tutkimustavoitteen määrittäminen, tutkimussuunnitelman teko, kohderyhmän tunnistaminen, näytteenoton suunnittelu, kyselytutkimuksen instrumentin suunnittelu ja viimeiseksi, instrumentin validatio. Seuraavat kaksi tehtävää liittyvät tutkimuksen toteuttamiseen. Aluksi tehdään osallistujien rekrytointia jonka jälkeen tehdään vastausten hallintaa. Toteutuksen jälkeen siirrytään analysointiin, jonka jälkeen tulokset raportoidaan.

Kyselylomake laadittiin tutkimusta varten Webropol-kyselyohjelmiston avulla. Lomakkeessa kysyttiin vastaajien demografiaan liittyviä kysymyksiä, eli sukupuolta, ikää ja kokemusta. Vastaajia pyydettiin ilmaisemaan projektien ajanjakso ja heidän vastualueensa projektissa. Lomakkeessa kysyttiin projektien ja organisaation kokoa, jossa vastaaja työskentelee. Projekteista kysyttiin lisäksi niissä käytettyä projektinhallintamenetelmää (Scrum, Kanban ym.) sekä työtapaa eli työskentelikö vastaaja projektissa etänä, toimistossa vai näiden yhdistelmää eli hybridimuodossa. Projektinhallinnan haasteita ja onnistumisia kysyttiin ensin suljettuihin kysymyksiin projektinhallinnan osa-alueiden perusteella ja projektinhallintaan vaikuttavien tekijöiden perusteella. Nämä tekijät pohjautuvat muun muassa tekijöihin, joiden takia projektit voivat epäonnistua. Näitä tekijöitä laadittiin tutkielman ohjaajien avustuksella. Suljettujen kysymysten jälkeen vastaajille esitettiin avoimia kysymyksiä liittyen suljettuihin kysymyksiin projektinhallinnasta. Avoimia kysymyksiä hyödynnettiin myös kysymään, kuinka tekijöiden vaikutusta on pyritty minimoimaan, mitkä tekijät tai projektinhallinnan osa-alueet



Julkaisu	Näyttökerrat	Jyväskylän seutu	Helsingin seutu
Ensimmäinen julkaisu	1291	26,9 %	8,3 %
Toinen julkaisu	102	34,5 %	24,1 %

Taulukko 6. Analytiikka LinkedIn-julkaisuista.

ovat johtaneet onnistumisiin projekteissa, kuinka työtapana on vaikuttanut projektin hallintaan sekä lopuksi, kuinka koronaviruspandemia on näkynyt projektinhallinnassa. Lomake on kokonaisuudessaan näkyvissä tutkielman liitteessä A. Kyselylomakkeen määrällisten kysymysten tarkoituksena oli saada aineistolle taustatietoja sekä selvittää, missä on yleisellä tasolla esiintynyt haasteita tai onnistumisia projektinhallinnan osalta.

## 4.2 Aineiston keruu

Aineiston kerääminen aloitettiin 11.1.2024, jolloin kyselyyn tuli ensimmäinen vastaus ja viimeinen vastaus saatiin 4.4.2024. Yhteensä vastauksia tällä ajanjaksolla tuli 13. Kyselyn vastaajilla yhtä lukuun ottamatta on tausta ohjelmistokehityksessä. Yhden vastaajan projektien vastualueet koostuivat pääosin projektipäällikkönä toimimisesta, jonka lisäksi useammalla vastaajalla on myös ollut projekteissa projektin vetämiseen liittyviä vastualueita.

Kyselyyn tavoiteltiin vastaajia useammasta ohjelmistokehitysyrityksestä, joissa työskennellään pääosin projektimuotoisesti. Kyselyn saaminen kiertoon yritysten sisällä osoittautui kuitenkin hyvin haasteelliseksi, vain muutama yrityksen henkilö sanoi suoraan, että kysely meni jakoon yrityksen sisällä. Sähköpostien lisäksi vastauksia pyrittiin hakemaan myös LinkedIn-palvelusta kahden julkaisun voimin, joista ensimmäinen saavutti 1291 näyttökertaa ja jälkimmäinen 102. LinkedIn antaa julkaisuille analytiikkaa, jonka sijainnin mukaista analytiikka verrattuna julkaisujen näyttökertojen määrään voidaan nähdä taulukossa 6. Kyselyä jaettiin myös Slack-ryhmässä *Koodiklinikka*, kanavalla *yleinen*, jossa oli viestin lähetyshetkellä 6106 jäsentä. Koodiklinikka on heidän sanojensa ([www.koodiklinikka.fi](http://www.koodiklinikka.fi)) mukaan ”Suomen suurin ohjelmistoalan yhteisö”. Alkuperäisten viestien ja sähköpostien lisäksi hyödynnettiin muistutusviestejä, jotka eivät lopulta antaneet uusia vastauksia yhtä lukuun ottamatta.

Kysely avattiin vastaajien toimesta yhteensä 84 kertaa, joista 50 henkilöä aloitti vastaamisen.

Vastausmäärän ollessa 13, vastausprosentiksi vastaamisen aloittaneista tulee lopulta 26 %.

### 4.3 Aineiston analyysi

Alhaisen vastausmäärän takia aineiston analyysissä keskitytään kyselyn avokysymyksiin analysoimalla ne laadullisin analyysin keinoin. Kyselyn rakenteen takia vastauksia on mahdollista teemoitella, itse kyselylomakkeen rakenteessa jo alustavaa teemoittelua onkin tehty.

Laadulliseen analyysiin löytyy neljä yleisesti pätevää vaihetta, jotka kuuluvat seuraavasti: 1. Aineiston valmistelu, 2. Lukeminen ja reflektointi, 3. Koodaaminen, kategorisoiminen ja muistiinpano sekä 4. Teemojen/konseptuaalisten mallien tai teorioiden kehittäminen (Ravindran 2019).

Aineiston valmistelu koskee etenkin haastattelututkimuksia. Tässä vaiheessa koko aineistosta valmistellaan pienempiä kokonaisuuksia, eli käytetään tiettyjä osia haastatteluista. Hyväksyttävä tapa on myös jättää pienempien osien muodostaminen väliin ja käyttää koko aineistoa (Ravindran 2019).

Lukemis- ja reflektointivaiheessa luetaan koko aineisto läpi refleктоimalla, mitä vastaajat jakavat aiheesta. Tämä antaa lähtökohdan aineiston ymmärrykselle. Tämä on tärkeä vaihe, ennen kuin teemoittelua tehdään, jotta ymmärretään aineiston taustat (Ravindran 2019).

Aineiston järjestäminen aloitetaan koodaamisella. Koodit ovat lipukkeita, jotka liitetään vastauksiin. Koodaamisen jälkeen siirrytään kategorioimiseen. Tässä kohdassa samankaltaiset koodit ryhmitellään ja muodostetaan ymmärrettävä sarja aineistosta. Kattegoria on kokoelma samankaltaista dataa yhdessä paikassa. Kategoriointivaiheen lopussa on koolla aineisto, jota käytetään tutkimuksessa. Tutkija voi kirjoittaa muistiota analyysistä, ajatuksista, tulkinnoista, kysymyksistä ja ohjauksista lisädata keräyksestä (Ravindran 2019).

Teemoitteluvaiheessa tuodaan yhteen kategorioiden taustalla olevat tarkoitukset. Teemoittelussa integroidaan kategoriat ja tulkitaan näiden kategorioiden sisältämät suhteet. Teemoittelua tehdessä tutkijan tulee selvittää, mikä on vastaajien kohtaama ydinongelma. Teemoja syntyy yleensä yhdestä kolmeen ja liian monen kategorian tai teeman olemassaolo voi viijaillla analyysin liian aikaisesta lopettamisesta (Ravindran 2019).

Analyysi aloitettiin lukemalla avokysymyksiin saadut vastaukset ja tarkastelemalla, mihin teemaan vastaukset liittyvät. Vastauksia sen jälkeen jaoteltiin eri teemoihin, jotka ilmenevät vastauksista sekä itse kyselyn rakenteesta. Aineiston teemoina ovat muun muassa työtapa, teknologiavalinnat, henkilöstösuhteet ja vaihtuvuus sekä projektin vetäjä. Analyysissä esille tulleet teemat ovat samoja, mitä on aikaisemmin kirjallisuudessa esiintynyt. Kaikista vastauksista valittiin sellaiset, jotka liittyvät tutkimuskysymykseen sekä, jotka liittyvät selkeästi johonkin teemaan.

Teemoittelussa otettiin huomioon, kuinka hyvin vastaukset antavat näkemystä, tekijöiden vaikutukseen projektinhallintaan. Useampi vastauksista käsitteli useampaa kuin yhtä teemaa, joten samoja vastauksia tai niiden osia liittyvät useaan teemaan. Avokysymysten vastaukset lajiteltiin teemoihin Microsoft Excel -työkalun avulla, jos vastaus ei liittynyt suoraan mihinkään teemaan, niitä ei sijoitettu Excel-tilukseen. Osa vastauksista oli hyvin geneerisiä, eivätkä ne antaneet juuri mitään näkemystä siihen, mitä kysymys koski.

Määrällisten kysymysten osalta Webropol-kyselyohjelmisto antaa kyselylle raportin, jossa kerrotaan vastausvaihtoehtojen jakaumat. Tämän raportin pohjalta laadittiin määrällisen aineiston tulosten taulukot.

## 5 Tulokset

Tässä luvussa esitellään kyselytutkimuksen määrällisen ja laadullisen aineiston analyysin tulokset. Määrällistä aineistoa ei tarkastella kovinkaan syvällisesti, vaan analyysissa on keskitytty kyselyn laadulliseen aineistoon, joka syntyi kyselylomakkeen avokysymyksistä.

### 5.1 Kyselyn taustakysymysten tulokset

Kyselyssä kysymykset 1–6 ja 9–10 antavat taustaa vastaajasta ja hänen projekteista. Sukupuolijakauma nähdään taulukossa 7. Kuten taulukosta nähdään, suurin osa vastaajista olivat miespuolisia.

Mies	Nainen	Muu
8	4	1

Taulukko 7. Kyselyn vastaajien sukupuolijakauma.

Vastaajien iän suhteen kaikki vastausvaihtoehdot saivat vähintään yhden vastauksen ja ikäjakauma nähdään kokonaisuutena taulukossa 8. Vastausvaihtoehtoina ovat yleisesti käytössä olevat ikäryhmät.

18-24 vuotta	25-34 vuotta	35-44 vuotta	45-54 vuotta	55+ vuotta
2	6	2	2	1

Taulukko 8. Kyselyn vastaajien ikäjakauma.

Ikäjakaumaa sivuten, kyselyn vastaajat olivat suhteellisen kokeneita. Suurimmalla osalla vastaajista oli vähintään viiden vuoden kokemus ohjelmistokehityksessä. Kokonainen kokemusjakauma näkyy taulukossa 9. Vastausvaihtoehtoina käytettiin samoja, mitä oli esiintynyt aikaisemmassa tutkimuksessa.

alle 2 vuotta	2-5 vuotta	5-10 vuotta	10+ vuotta
2	2	4	5

Taulukko 9. Kyselyn vastaajien kokemusjakauma.

Vastaajien vastualueissa oli paljon samankaltaisuutta. 5 vastaajaa ei antanut projekteille ajanjaksoa, jolloin he osallistuivat niihin ja yksi vastaaja ei antanut ollenkaan projektin vastuuetua. Tätä henkilö laskematta mukaan vain yhdellä henkilöllä ei ollut projekteja, joissa hän oli ohjelmistokehittäjänä, vaan hän oli projektipäällikkönä parissa projektissa. Muiden vastaajien vastualueet koostuivat pääosin ohjelmistokehityksestä sekä osalla oli lisäksi projekteja, joissa he olivat projektipäällikkönä. Vastuualueiden kysymyksen vastauksista voidaan nähdä, että yhteensä projekteja vastaajien kesken oli vähintään 30, mahdollisesti enemmän.

Ketterän kehityksen periaatteiden mukaisesti suurimman osan vastaajien projekteista oli pieniä eli niissä oli 1–10 jäsentä. Kokonainen jakauma on näkyvillä taulukossa 10. Koska kysymyksessä oli mahdollista valita useampi vaihtoehto, kysymykseen tuli 16 valintaa, vaikka vastaajia oli 13.

<b>Jäsenten määrä</b>	<b>Vastausten määrä</b>	<b>Osuus vastauksista</b>
1–10 jäsentä	11	84,6 %
11–50 jäsentä	4	30,8 %
51–100 jäsentä	1	7,7 %
100+ jäsentä	0	0 %

Taulukko 10. Kyselyn vastaajien projektien jäsenten määrän jakauma.

Jos projektit olivat pääsääntöisesti vastaajilla pieniä, niin heidän organisaatioiden koko oli sen sijaan isompi. Eniten valintoja sai edelleen 1–10 työntekijää, mutta vastaajien joukossa yli 250:n työntekijän organisaatiot ovat enemmän edustettuina. Koko jakauma on näkyvillä taulukossa 11. Tässä kysymyksessä oli jälleen mahdollista valita useampi vaihtoehto, mutta mielenkiintoista on, että valittuja vastauksia on kuitenkin yksi enemmän, kuin projektien kohdalla, eli yhteensä 17.

Kysymyksessä 9 kysyttiin projektien työtappaa, joista hybridityöskentely oli suosituin valinta. Valinnat näkyvät taulukossa 12. Kysymyksessä oli jälleen mahdollista valita useampi vaihtoehto ja tähän kysymykseen tulikin yhteensä 26 valintaa. Taulukkoa katsellessa huomataan, kuinka suosittua etätyö on. Selkeää kuitenkin on, että hyvin useassa projektissa kuitenkin käydään osittain toimistolla, sillä hybridityön osuus on erittäin korkea.

Työntekijöiden määrä	Vastausten määrä	Osuus vastauksista
1–10 työntekijää	5	38,5 %
11–50 työntekijää	1	7,7 %
51–250 työntekijää	3	23,1 %
251–1000 työntekijää	4	30,8 %
1000+ työntekijää	4	30,8 %

Taulukko 11. Kyselyn vastaajien organisaatioiden työntekijöiden jakauma.

Työtapa	Vastausten määrä	Osuus vastauksista
Etätyö	9	69,2 %
Toimistotyö	6	46,2 %
Hybridityö	11	84,6 %

Taulukko 12. Kyselyn vastaajien projektien työtavan jakauma.

Seuraavaksi kysyttiin projektien projektinhallinnanmenetelmää. Kuten ollaan jo aiemmin nähneet, Scrum ja Kanban ovat suosittuja ketteristä menetelmistä, eikä tässä kyselyssä siitä poikettu. Scrum oli suosituin käytetty viitekehys, jota seurasi Kanban. Vastaukset kokonaisuudessaan nähdään taulukossa 13. Vastausvaihtoehdot ovat suosituimmat projektinhallintakehykset *State of Agile*-raportin mukaan, lisäksi perinteiset vesiputousmalli ja V-malli sekä skaalatut kehykset. Lisäksi on vastausvaihtoehto, jos listatut eivät vastaa projektissa käytettyä menetelmää.

Mielenkiintoista on nähdä, että kahdessa projektissa on vielä käytetty jotain perinteistä menetelmää jaon mennessä tasan vesiputousmallin ja V-mallin välillä. Kolme vastaajaa ilmoitti vastausvaihtoehdon “Jokin muu, mikä?” yhteydessä, mitä menetelmää he käyttivät. Yksi vastaaja kertoi menetelmän olevan “sekamallinen agile”, toinen kertoi menetelmän olevan *ScrumBan*, eli Scrumin ja Kanbanin yhdistelmä ja kolmas kertoi, ettei kyseessä ollut mikään varsinainen menetelmä, mutta projektissa käytettiin Kanbanin taulujärjestelmää muistuttavaa työnkulkua.

Projektinhallinnan menetelmä	Vastausten määrä	Osuus vastauksista
Scrum	9	69,2 %
Kanban	7	53,8 %
Lean Startup	0	0 %
XP	0	0 %
Vesiputousmalli	1	7,7 %
V-malli	1	7,7 %
Skaalatut kehykset	4	30,8 %
Jokin muu	4	30,8 %

Taulukko 13. Kyselyn vastaajien projektien projektinhallinnan menetelmien jakauma.

## 5.2 Kyselyn kysymysten 7 ja 11 tulokset

Kysymykset 7 ja 11 käsittelivät projektinhallinnan osa-alueiden haasteellisuutta ja niihin vaikuttavia tekijöitä. Kysymyksessä ohjattiin vastaajaa pohtimaan osa-alueiden haasteellisuutta heidän projektien keskiarvollisen haasteellisuuden suhteen. Taulukossa 14 esitetään vastausten jakauma kysymykselle projektinhallinnan osa-alueista. Vastausvaihtoehdot ovat samoja, kuin projektinhallinnan osa-alueet, joiden lisäksi vaatimusten hallinta on eroteltu laajuuden hallinnasta. Lisäyksenä on myös muutosten, sidosryhmien ja dokumentaation hallinta, jotka ovat esiintyneet kirjallisuudessa haasteina.

Taulukosta huomataan, kuinka suuressa osassa koetaan ainakin jonkintasoista haasteellisuutta. Varsinkin vaatimukseen liittyvät osa-alueet (laajuuden hallinta, vaatimusten hallinta ja muutosten hallinta) nähdään vähintään joksenkin haasteellisena. Kommunikaation hallinta, riskienhallinta ja laadunhallinta ovat myös sellaisia, jotka nähdään eniten joksenkin haasteellisena. Sidosryhmien hallinta puolestaan koetaan erityisen haasteellisena: se on nähty eniten erittäin haasteellisena. Dokumentaation hallinta on myös haasteellinen osa-alue, joka ei ole yllättävää, sillä suurimmassa osassa projekteissa oli käytetty jotain ketterän kehityksen menetelmää. Kulujen hallinta puolestaan nähdään vähiten haasteellisena: 46,1 % ilmoitti, ettei siinä ollut haasteita. Vaatimusten hallinta (7,7 %), laajuuden hallinta (15,4 %) sekä aikataulun hallinta ja muutosten hallinta (23,1 %) saivat puolestaan alhaisimman vastausprosentin kohtaan "1: Ei haasteellista". Kuten vastausvaihtoehtojen keskiarvosta nähdään, ohjelmis-

<b>Projektin- hallinnan osa-alue</b>	<b>1: Ei haas- teellista</b>	<b>2: Joksenkin haasteellista</b>	<b>3: Erittäin haasteellista</b>	<b>4: En osaa sa- noa</b>
Kommuni- kaation hallin- ta	30,8 %	46,1 %	23,1 %	0 %
Aikataulun hallinta	23,1 %	53,8 %	15,4 %	7,7 %
Laajuuden hallinta	15,4 %	69,2 %	15,4 %	0 %
Vaatimusten hallinta	7,7 %	61,5 %	30,8 %	0 %
Henkilöstö- resurssien hallinta	30,7 %	15,4 %	23,1 %	30,8 %
Kulujen hal- linta	46,1 %	23,1 %	15,4 %	15,4 %
Riskienhallinta	30,8 %	46,1 %	23,1 %	0 %
Hankintojen hallinta	38,4 %	23,1 %	7,7 %	30,8 %
Laadunhallinta	30,8 %	53,8 %	7,7 %	7,7 %
Muutosten hallinta	23,1 %	46,1 %	23,1 %	7,7 %
Sidosryhmien hallinta	30,8 %	23,1 %	38,4 %	7,7 %
Dokumentaa- tion hallinta	38,4 %	30,8 %	30,8 %	0 %
<b>Yhteensä</b>	<b>28,8 %</b>	<b>41,0 %</b>	<b>21,2 %</b>	<b>9,0 %</b>

Taulukko 14. Kyselyn vastaajien projektien projektinhallinnan osa-alueiden haasteellisuus.



1: Merkittävästi negatiivinen, 2: Joksenkin negatiivinen, 3: Ei vaikutusta, 4: Joksenkin positiivinen, 5: Merkittävästi positiivinen, 6: En osaa sanoa						
<b>Projektinhallintaan vaikuttava tekijä</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
Projektin tehtävienhallintatyökalut	0 %	0 %	7,7 %	38,5 %	53,8 %	0 %
Tiiminhallinta- ja kommunikointityökalut	0 %	0 %	0 %	23,1 %	76,9 %	0 %
Teknologiaosaaminen	0 %	0 %	7,7	38,5 %	46,1 %	7,7 %
Uudet teknologiat	0 %	7,7 %	46,1 %	15,4 %	15,4 %	15,4 %

Taulukko 15. Kyselyn vastaajien projektien käytettyjen työkalujen ja teknologioiden vaikutus projektinhallintaan.

tokehitysprojektit koronaviruspandemian aikaan on nähty haasteellisina projektinhallinnan osa-alueiden osalta. Taulukoissa 15–18 on esitettyä kysymyksen 11, eli eri tekijöiden joko positiivinen tai negatiivinen vaikutus, vastausten valinnat. Tekijöitä on johdettu yleisistä projektien epäonnistumisen syistä (Hughes, Rana ja Simintiras 2017). Vaihtoehtoisiksi valittiin lisäksi tekijöitä, jotka ovat esiintyneet aikaisemmassa tutkimuksessa haasteina. Arviointiasteikko tekijöille oli 1-6, numeroiden vastaten seuraavaa: 1: Merkittävästi negatiivinen, 2: Joksenkin negatiivinen, 3: Ei vaikutusta, 4: Joksenkin positiivinen, 5: Merkittävästi positiivinen, 6: En osaa sanoa. Asteikko on lisäksi esillä jokaisen erillisen taulukon yläpuolella.

Kuten taulukosta 15 nähdään, vastaajien projekteissa käytetyt työkalut viestimiseen ja projektinhallintaan on nähty positiivisena tekijänä. Myös teknologiaosaaminen on nähty positiivisena, kun taas uudet teknologiat eivät ole vaikuttaneet juurikaan projekteihin.

Projektien sisäisten tekijöiden vaikutuksen näkemys on paljon jakautuneempaa, kuin käytetyissä työkaluissa ja teknologioissa (taulukko 16). Monella tekijällä on kuitenkin nähty olevan positiivinen vaikutus, kuten henkilöstösuhteet, työtapo, projektin vetäjä ja vastualueiden selkeys. Myös ylemmän johdon tuki on nähty enemmän positiivisena vaikuttajana, kuin negatiivisena. Vaihtuvuus puolestaan on sellainen tekijä, joka on nähty negatiivisena vaikut-

1: Merkittävästi negatiivinen, 2: Joksenkin negatiivinen, 3: Ei vaikutusta, 4: Joksenkin positiivinen, 5: Merkittävästi positiivinen, 6: En osaa sanoa						
<b>Projektinhallintaan vaikuttava tekijä</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
Henkilöstösuhteet	7,7 %	0 %	15,4 %	46,1 %	23,1 %	7,7 %
Työtapa	0 %	7,7 %	30,8 %	15,4 %	46,1 %	0 %
Projektin vetäjä	15,4 %	0 %	15,4 %	38,4 %	30,8 %	0 %
Ylemmän johdon tuki	0 %	23,1 %	23,1 %	30,7 %	23,1 %	0 %
Henkilöstön palkkaus	7,7 %	7,7 %	38,4 %	30,8 %	7,7 %	7,7 %
Vaihtuvuus	15,4 %	30,8 %	30,8 %	15,4 %	0 %	7,7 %
Vastuualueiden selkeys	23,1 %	7,7 %	7,7 %	30,7 %	30,8 %	0 %
Sisäiset sidosryhmät	0 %	30,7 %	23,1 %	7,7 %	15,4 %	23,1 %
Organisaatorakenne	0 %	15,4 %	23,1 %	0 %	15,4 %	46,1 %

Taulukko 16. Kyselyn vastaajien projektien sisäisten tekijöiden vaikutus projektinhallintaan.

tajana, mutta toisaalta henkilöstön palkkaus on ollut joko enemmän positiivinen tai sillä ei ole ollut vaikutusta. Organisaatorakenne on puolestaan tekijä, joita kyselyn vastaajat eivät osanneet arvioida, kuinka se on vaikuttanut projektien hallintaan.

Projektien ulkoiset tekijät ovat selkeästi hyvin hajautunutta (taulukko 17). Vastausvaihtoehdot ovat saaneet hyvin tasaisesti vastauksia, jolloin tilaajan sitouttaminen ja odotukset ovat nähty sekä positiivisena vaikuttajana, että negatiivisena, mutta myös sellaisina, joilla ei ole ollut vaikutusta. Ulkoiset sidosryhmät puolestaan on nähty hieman enemmän negatiivisena, mutta myös sellaisena tekijänä, joita vastaajat eivät ole osanneet arvioida.

Muiden tekijöiden osalta moni tekijä on nähty positiivisena vaikuttajana, kuten tiedon keruu ja hallinta, projektin suunnittelu ja projektinhallintakehys (taulukko 18). Tekijöistä vaatimusten muuttuminen ja muutokset ylipäättänsä ovat nähneet negatiivisena vaikuttajana projektinhallintaan. Tavoitteiden selkeys on selkeästi jakautunutta, osa on nähnyt sen negatiivisena, kun taas osa positiivisena. Rahoituskuviot olivat tästä joukosta sellainen tekijä, joita vastaajat eivät osanneet arvioida.

1: Merkittävästi negatiivinen, 2: Joksenkin negatiivinen, 3: Ei vaikutusta, 4: Joksenkin positiivinen, 5: Merkittävästi positiivinen, 6: En osaa sanoa						
<b>Projektinhallintaan vaikuttava tekijä</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
Tilaaajan sitouttaminen	15,4 %	7,7 %	30,7 %	7,7 %	30,8 %	7,7 %
Odotukset	7,7 %	23,1 %	30,7 %	23,1 %	7,7 %	7,7 %
Ulkoiset sidosryhmät	0 %	30,8 %	15,4 %	7,7 %	7,7 %	38,4 %

Taulukko 17. Kyselyn vastaajien projektien ulkoisten tekijöiden vaikutus projektinhallintaan.

Koska kyselyn vastaajien määrä oli alhainen ja vaikka vastaajien projektien määrä oli korkeampi, niin nämä määrälliset tilastot on syytä nähdä vain suuntaa antavina. Näistä tilastoista ei siis kannata vetää suurempia johtopäätöksiä, toisin kuin käytännön tasolla tarkastella kriittisesti omaa projektia, ja pohtia, missä haasteita voisi olla ja mitkä tekijät vaikuttavat mitenkään projektinhallintaan. Yksi vastaaja huomioikin projektien moniluontoisuuden ja sen, ettei kahdella projektilla aina ole samoja haasteita:

Osa-alueiden haasteellisuus vaihtelee projektista, omasta firmasta ja projektin tilaajasta toiseen paljonkin, joten on erittäin hankala antaa yhteenkään yleispätevää vastausta.

### 5.3 Työtapa

Työtapa oli yksi useimmiten nousseista tekijöistä, joka nähtiin koronaviruspandemian vaikutuksena. Lähes kaikki vastaajat huomioivat, kuinka etätyö tai etätyön mahdollistaminen on ollut positiivinen tekijä projektinhallinnassa tai pelkästään ohjelmistokehityksessä. Kaikki vastaajat eivät ole kuitenkaan nähneet etätyötä positiivisena, vaan he nostavat negatiivisia vaikutuksia etätyöstä. Esimerkiksi yksi vastaus puhuu työilmapiirin luomisen haastavuudesta etätyössä:

Turvallisen, arvostavan eli tehokkaan devausilmapiirin luominen etänä on todella hankalaa.

1: Merkittävästi negatiivinen, 2: Joksenkin negatiivinen, 3: Ei vaikutusta, 4: Joksenkin positiivinen, 5: Merkittävästi positiivinen, 6: En osaa sanoa						
<b>Projektinhallintaan vaikuttava tekijä</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
Projektinhallintakehys	0 %	7,7 %	23,1 %	15,4 %	46,1 %	7,7 %
Tiedon keruu	0 %	0 %	23,1 %	38,4 %	30,8 %	7,7 %
Tiedon hallinta	0 %	7,7 %	23,1 %	38,4 %	23,1 %	7,7 %
Projektin suunnittelu	0 %	7,7 %	23,1 %	38,4 %	23,1 %	7,7 %
Vaatimusten ymmärrys	7,7 %	23,1 %	15,4 %	23,1 %	30,7 %	0 %
Vaatimusten muuttuminen	15,4 %	61,5 %	7,7 %	7,7 %	7,7 %	0 %
Muutokset	7,7 %	30,8 %	38,4 %	7,7 %	7,7 %	7,7 %
Tavoitteiden selkeys	0 %	38,4 %	7,7 %	38,5 %	15,4 %	0 %
Liiketoimintaympäristön ymmärrys	0 %	23,1 %	7,7 %	46,1 %	7,7 %	15,4 %
Rahoituskuviot	0 %	7,7 %	15,4 %	15,4 %	0 %	61,5 %

Taulukko 18. Kyselyn vastaajien projektien muiden tekijöiden vaikutus projektinhallintaan.

Toinen vastaaja huomioi etätyössä ihmisten tai sidosryhmien havaitsemisen haastavuuden etätyössä. Vastaajan mukaan etätyötä tehdessä viestien huomioiminen ja tiedon ylläpito on haastavampaa etätyössä verrattuna toimistotyöhön etäviestintien käytön myötä. Kommunikaatio on lisäksi projektinhallinnan alue, joka on huomattu muuttuvan etätyössä. Eräs vastaaja kuitenkin näkee etätyöhön siirtymisen positiivisena asiana liittyen muun muassa kommunikaatioon:

Pakon sanelema etätyö paljasti toimistotyön tehottomuuden. Työnantajan yritykset tuoda toimistotyötä takaisin ovat osoittautuneet huonoksi ajatukseksi; asynkroninen kommunikaatio on itseään dokumentoivaa tekemistä eikä asiat jää enää sen varaan että mitä kukakin muistaa kuulleensa jossain palaverissa. Myöskään ohituskaistoja ei enää samalla tavalla ole, kun cowboy-PO:t [tuoteomistajat] eivät voi hyökätä seniorikehittäjän pöydälle keskeyttämään töitä omilla vaatimuksillaan.

Vastauksista huomaakin, että etätyö tai etenkin etätyön mahdollistaminen on selkeästi nähty positiivisena muutoksena. Kuten viimeisestä vastauksesta näkee, työnantajien tulee mahdollistaa tulevaisuudessakin etätyömahdollisuus, vaikka välillä etätyö voikin tuoda omia haasteitaan. Työntekijät selkeästi arvostavat työtavan valitsemisen vapautta.

## 5.4 Kommunikaatio

Jo luvussa 5.3 nähtiin, kuinka kommunikaatio on muuttunut, eikä aina positiiviseen suuntaan. Aineistossa nähdään juuri kommunikaation haasteellisuus koronaviruspandemian aikaisissa ohjelmistokehitysprojekteissa. Eräs vastaaja antaa suoran näkemyksensä kommunikaatiosta eräällä alalla:

Kommunikaatio nykyisellään teollisuuden ohjelmistoyrityksissä on todella heikkoa ja vaikuttaa suoraan resurssien hallintaan.

Tarvittavien henkilöiden tavoittaminen korostuu kommunikaatioon liittyvissä vastauksissa. Monet vastaajat huomioivat, kuinka tarvittavia henkilöitä on välillä vaikea tavoittaa, kun työskennellään etänä sekä monesta eri paikasta. Henkilöiden tavoittamisen haastavuus näkyy erään vastaajan mukaan toiminnan hidastumisena. Henkilöiden tavoittamisen haasteet eivät liity pelkästään työkavereihin, vaan myös asiakkaaseen ja heidän projektiryhmiinsä kuten eräs vastaaja sanoo:

Asiakkaan eri projektiryhmien kanssa kommunikointi on ollut haastavaa, tietoa on joskus joutunut kysymään useamman kerran ja välillä ei ole selvää keneltä tietoa pitäisi kysyä. Samoin hankala saada yhteyttä toisiin tiimeihin, vaikka olisi yhteinen chat-kanava Teamsissa, jos kukaan tiimistä ei ota vastuulleen vastata kysymyksiin. Tämä ei ole muuttunut korona-aikana verrattuna aiempaan eikä sen jälkeen.

Moni vastaaja huomioi kuitenkin myös etäviestintien positiiviset puolet kertoen muun muassa, kuinka viestiketjut ovat mahdollista avata jälkeinpäin, jolloin mahdollisesti unohdettu asia palaa mieleen. Asiakkaan kanssa kommunikoimisesta eräs vastaaja kertoo, kuinka heidän kanssa kommunikoimalla edesautetaan projektin edistymistä ja tarpeiden toteutumista. Eräs vastaaja korostaa lisäksi avoimen kommunikaation merkityksen:

Kun voi luottaa, että jokin on tehty ajallaan, stressi vähenee ja tiimihenki paranee. Avoin kommunikaatio on mahdollistanut ongelmien esille ottamisen aikaisessa vaiheessa, jolloin tarvittavia korjausliikkeitä on pystytty tekemään ajoissa.

On siis selkeää, että etänä tehtävissä projekteissa kommunikaatio on ollut haasteellista. Vastauksissa kuitenkin tulee ilmi, että kommunikaatio voi olla onnistunutta ja myös edistävä tekijä projekteissa. Kommunikaatio nähtiin haasteellisena myös kysymyksen 7 vastauksissa (taulukko 14), joka vahvistaa avokysymysten vastauksia kommunikaation haasteellisuudesta. Vastauksista huomataan, että etenkin asiakkaan kanssa kommunikaatio tulee toteuttaa tehokkaasti. Suurimpana haasteena etäviestintien kautta tehdyssä kommunikaatiossa nähdäänkin oikeiden henkilöiden tavoittaminen.

## **5.5 Ulkoiset tekijät ja tilaaja**

Suoraan tilaajaan liittyviä vastauksia ei aineistossa tullut paljoa, joka voi mahdollisesti kertoa projektien onnistumisesta tilaajan saralla. Parin vastauksen yhdistävä tekijä löytyy tilaajan taloudellisesta tilanteesta. Eräs vastaaja kertoo, kuinka tilaajan muuttunut taloustilanne oli vaikuttanut tekijän näkemykseen, tulisiko kehitystä vähentää. Toinen vastaaja kertoo suoraan, kuinka taloudellinen tilanne yhdessä projektissa vaikutti suoraan kehitykseen. Vastaajan mukaan tässä projektissa kehitys laitettiin jäihin, kun rahat loppuivat ja heidän olisi pitänyt jatkaa kehitystä täydellä voimalla, kun toimintaan tuli lisää rahaa. Toisessa projektissa taas omistajanvaihdos näkyi kehityksen ulkoistamisen muodossa halvempaan maahan. Tilaajan sitouttamisesta eräs vastaaja puhuu ketterästä kehityksestä ja, että sitouttamisella voidaan varmistaa toiminnallisuuden vastaavuus tilaajan odotuksia vastaan.

Eräs vastaaja puhuu myös useamman tiimin ollessa mukana kehityksessä:

Kun tekemisessä on ollut mukana useampi tiimi, on ollut hankala seurata, koska toinen tiimi on saanut valmiiksi osuutensa, ja milloin oma tiimi pääsee tekemään omaa osuuttaan. Vaikka aikataulu olisi suunniteltu etukäteen, on toisten tiimien tekeminen välillä vaihtunut ja oman tiimin tekeminen myöhästynyt, kun sovitut asiat ovat myöhästyneet tai toteutettu eri tavalla kuin on sovittu.

Tilaajan taloudellinen tilanne vaikuttaakin suoraan siihen, kuinka onnistuneesti projekti voidaan vetää läpi. Vastauksista on huomattu, että taloudellinen tilanne on vaikuttanut suoraan kehitykseen ja täten projektien läpiviemiseen. Myös kehityksen ulkoistaminen halvempaan maahan on nähty todellisena haasteena.

## **5.6 Johtajuus projektissa**

Projektipäällikön ja tuoteomistajan valinta on nähty jo kirjallisuudessa tärkeänä osana projektia. Aineistossakin huomioidaan projektin vetäjän valinnan tärkeys. Eräs vastaaja kertoo, kuinka hyvä projektin vetäjä saa aikaan merkittävää parannusta projektin sujumisessa. Etätyössä projektin vetäjän valinta on ollut erittäin tärkeää. Vastaaja sanoo suoraan, kuinka sen puuttuminen on johtanut haasteisiin, jotka korostuvat etätyössä. Eräs vastaaja sanoo myös tuoteomistajan kunnollisen valinnan olevan tärkeää:

Odotusten hallinta tilaajan suhteen on olennaista heti alusta alkaen. Pätevän tuoteomistajan saaminen tilaajan puolelle on tässä ehkä tärkeimmästä päästä, mutta valitettavan usein hankalaa.

Projektin vetäjän merkitystä nostetaan lisäksi sisäisten tekijöiden suhteen. Erään vastaajan mukaan yhdessä projektissa koko muu tiimi lähti pois, koska yksi jäsenistä aiheutti haittaa muille jäsenille. Tilanteessa projektin vetäjä ei tehnyt tilanteelle mitään, vaikka asiasta sanottiin hänelle suoraan.

Projektin vetäjän valinta onkin tärkeä osa projektinhallintaa. Parhaimmassa tapauksessa hyvin valittu projektipäällikkö edistää projektin sujumista, kun taas pahimmassa tapauksessa huonosti valittu on estävä tekijä projektin edistymisessä. On kuitenkin hyvä muistaa, että johtajuus on taito siinä missä muutkin, joten sitä on mahdollista kehittää. Oikea valinta projektin johtoon onkin sellainen henkilö, joka ei välttämättä ole kokenein, vaan sellainen, joka osaa oppia virheistään ja kehittää toimintaansa palautteen mukaan. Myös tuoteomistajan kanssa ongelmia oli ilmennyt, joten sama pätee myös häneen.

## 5.7 Käytetyt työkalut

Käytetyt työkalut voivat hyvinkin vaikuttaa projektinhallintaan ja projektien edistymiseen. Aineistossa esiintyy useasti maininta Jirasta ja sen tärkeydestä. Jira nähdään tärkeänä työkaluna muutosten hallinnassa, jonka lisäksi se tuo erään vastaajan mukaan näkyvyyttä ja parantaa projektin kokonaiskuvan ymmärtämistä. Jira on Atlassianin kehittämä sovellus, jolla hallitaan projektin tehtäviä ja sovelluksen vaatimuksia/kehityslistaa. Eräs vastaaja sanookin Jiran käytöstä seuraavanlaisesti:

Jira on ollut ehdottoman tärkeä työkalu muutosten hallinnassa usealle eri asiakkaalle.

Eräs vastaaja sanoo yleisesti käytettyjen työkalujen helpottaneen työskentelyä ja selkeyttäneen osa-alueita. Aina työkalut eivät kuitenkaan ole nähty positiivisena. Erään vastaajan mukaan projektissa työkaluja on vaihdettu tilaajan toimesta, joka on ollut vaikeuttavana tekijänä tekemisen seuraamisessa, kuten seuraavasta lainauksesta nähdäänkin:

Asiakas on vaihtanut projektin aikana käytettäviä työkaluja pari kertaa, joka on vaikeuttanut projektin tekemisen seuraamista. Vaihto työkalusta toiseen on vaikeuttanut muutosten hallintaa, kun uusi työkalu (Azure devops) ei ole vastannut käyttökokemukseltaan Jiraa.

Eräs vastaaja nostaakin käytettyjen työkalujen tuttuuden olevan positiivinen tekijä:

Ketterät menetelmät ja niiden hallinnoinnin työkalut lisäävät läpinäkyvyyttä ja mahdollisuutta seurata, missä mennään ja mikä vaikuttaa mihinkin. Käytetyt teknologiat ovat olleet tuttuja ja pitkän aikaa ylläpidettyjä, jonka vuoksi talossa on tietämystä niistä joka auttaa ylläpidossa/muutosten tekemisissä.

Viestinten käyttö on nähty positiivisena tekijänä projekteissa. Esimerkiksi Slack-viestipalvelun käyttö on mahdollistanut samanaikaisen ja reaaliaikaisen kommunikaation sekä auttanut parityöskentelyä. Toinen vastaaja kertoo, kuinka viestinten käyttö on mahdollistanut nopean kommunikaation. Viestipalveluiden käyttö nähdäänkin tärkeänä osana projektinhallinnassa kommunikaation osalta sekä myös helpottavana tekijänä. Etenkin tiedon löytäminen kirjallisena on asia, joka on nostettu esille useamman vastaajan toimesta.



Suunnittelun osalta Miro- ja Mural-ohjelmistoja nostetaan esille, kuinka tärkeitä ne ovat, kuten seuraavassa lainauksessa nähdäänkin:

Suunnittelussa ja retroissa käytetyt Miro/Mural ovat olleet hyvin toimivia, kun jokainen pääsee täydentämään tietoja ja näkee kokonaisuuden.

Käytetyt työkalut tulevat esille myös vaatimusten muuttumisessa, johon yksi vastaaja kertoo Jira-palvelun olevan tärkeä työkalu. Eräs vastaaja kertoo projektin sisäisten tekijöiden tuovan ongelmia vaatimusten muuttuessa. Toinen vastaaja lisää, että vaatimusten muuttuminen on tyypillisin tapa tuottaa ongelmia projektiin, varsinkin jos muutoksia ja niiden seurauksia ei ymmärretä. Vastauksessa annetaankin neuvo, että muutoksenhallintaan tulisi panostaa, jotta ongelmia ei synny. Eräs vastaaja sanoo vaatimusten muuttumisen olleen syynä MVP:n (lyhenne *minimum viable product*) laajentumiseen ja julkaisemisen myöhästymiseen. MVP tarkoittaa tuotetta, jossa on pienin määrä ominaisuuksia, joilla on mahdollista tuottaa arvoa sen käyttäjille.

Projekteissa käytetyt työkalut on kuitenkin nähty suurimmalta osin positiivisina, kuten taulukko 15 näyttää. Työkalujen käyttö nykyajan ohjelmistokehitysohjelmistoprojekteissa onkin suuressa roolissa, ja niiden toimivuus vaikuttaa olennaisesti projektin läpiviemiseen. Aina käytetyt työkalut eivät kuitenkaan ole nähty positiivisessa valossa, joten ne tulee valita, joko projektin jäsenten osaamisen mukaan tai, kuinka helppoa ne ovat oppia. Vastauksista tulee esille myös työkalujen käyttö dokumentaatioon. Ketterien menetelmien ollessa vallassa, dokumentaation hallinta on yleinen haaste, joten kaikki mitä sen pienentämisen eteen voidaan tehdä, pitääkin tehdä. Työkalujen käyttö dokumentaation hallinnan tukena onkin yksi mahdollinen vaihtoehto.

## **5.8 Projektien sisäiset tekijät ja kuinka minimoida haasteita**

Aineistossa on kuvailtu useampiakin eri keinoja minimoimaan eri tekijöiden negatiivisia vaikutuksia projektinhallintaan. Muun muassa avoimuus nostetaan keinona, jolla on pyritty minimoimaan negatiivisten tekijöiden vaikutusta projektinhallintaan. Avoimuus on ilmennyt esimerkiksi ilmoittamalla suoraan sidosryhmille omat rajoitteet ja säännöstelyt. Palaverissa avoimuus on erään vastaajan mukaan ollut tärkeä tekijä tällä saralla sekä olemalla avoimessa

yhteydessä asiakkaan kanssa.

Palaverit tuleekin esille useamman vastaajan toimesta. Vastauksissa nousee esimerkiksi aikataulutettujen palaverien tilaajan kanssa olevan tärkeää. Toinen vastaaja kertoo ajastettujen ja ajastamattomien palaverien olevan haasteiden pienentävä tekijä. Palavereihin liittyen myös pikaviestinten ahkera käyttö nostetaan keinona pienentää negatiivisia vaikutuksia projektinhallintaan.

Dokumentaation rooli nousee kahden vastaajan toimesta, joista yksi vastaaja kertoo dokumentaation kehittämisen olevan keino pienentää negatiivisia vaikutuksia projektinhallintaan. Dokumentaation puutehan onkin tuonut haasteita yleisesti ketterässä kehityksessä. Yksi vastaajista kertoo kaiken oleellisen tiedon kirjoittamisen Jira-lipukkeille olevan yksi dokumentaation keinoista. Myös etäviestinten käyttö on nähty eräänä dokumentaation keinona:

Toisaalta Teams-viesteillä kommunikaation saa avattua, kun taas kasvoikkain asiat voi unohtua, jos niitä ei heti pääse kertomaan.

sekä:

... asynkroninen kommunikaatio on itseään dokumentoivaa tekemistä eikä asiat jää enää sen varaan että mitä kukakin muistaa kuulleen jossain palaverissa. . .

Muita keinoja vaikutusten minimoimiseen on myös esimerkiksi määrällisen analyysin tekeminen toistuvista murheista, jolloin vastaajan mukaan he saivat vähennettyä seremonioiden määrää ja siirryttyä asynkroniseen työn tekemiseen. Samankaltaista oli myös toisella vastaajalla, sillä he olivat kehittäneet heidän projektinhallintaansa koronapandemian aikana, jolloin etätyöskentely sujui vaivattomasti.

Yhdessä vastauksessa nostettiin myös esihenkilön rooli. He olivat olleet muistuttamassa pitämään taukoja myös etätyössä. Myös asiakkaan kanssa toimimisesta on vastaajan mukaan keskusteltu projektissa:

Projektin sisällä omassa tiimissä on keskusteltu, että asiakkaan muiden tiimien toimintaan ei voi vaikuttaa, ja se ei ole oma syy, jos tekeminen myöhästyy siksi, ettei toinen tiimi ole toteuttanut osuuttaan ajoissa/sovitulla tavalla.

Projektin sisäisistä tekijöistä moni nostaa esille henkilöstösuhteet. Moni vastaaja nostaa, kuinka hyvät suhteet toisten kanssa auttaa projektissa toimimista, kuten seuraavasta lainauksesta nähdään:

Jos projektitiimin sisäinen dynamiikka toimii ja huumori kukoistaa sopivissa paikoissa, niin sillä on dramaattinen vaikutus projektin onnistumismahdollisuuksiin. Tässä taas on avainroolissa minimissään aloituspalaveri naamakkain ml. jonkinlainen vapaa-ajan aktiviteetti, jossa voidaan tutustua porukkana toisiimme.

Yhteiset toimintatavat nostetaan yhdessä vastauksessa, jolloin toiset voivat aiheuttaa ongelmia, jos niitä ei seurata. Toinen vastaaja nostaa myös positiivisten henkilöstösuhteiden mahdollisen haitan:

Henkilöstösuhteet ja tutun osaavan porukan pysyminen ovat mielestäni tärkeitä projektin edistämiseksi. Täytyy kuitenkin myös pysyä ammattimaisena, ettei homma meni liian kavereiluksi eikä puutteista uskalleta mainita.

Vaihtuvuus on nostettu sisäisten tekijöiden suurimpana ongelmana. Yksi vastaaja nostaa, kuinka valittujen teknologioiden takia osaavien tekijöiden saatavuus on ollut haastavaa ja toinen vastaaja kertoo suoraan työntekijöiden saatavuudesta:

Kaikki osapuolet eivät aina ole asioista kartalla tarpeeksi ja työntekijöitä ei aina ole tarpeeksi.

Myös ylempää johtoa kritisoidaan lisäten myös henkilöstön vaihtuvuuteen:

Ylemmällä johdolla puolestaan on mahdollisuus sössiä hyvin toimiva programmikin omalla "visionääriseksi" mainostetulla tavallaan toimia.

Samanlaista myrkkyä tekemiselle on epäterveessä määrin tapahtuva henkilöstön vaihtuvuus projektihenkilöstössä. Keskimäärin resilienssi tuntuu olevan varsin heikko isoissakin firmoissa, eli monessa kohti yksi ihminen vastaa jopa jostain kriittisestä toiminnosta.

Ylemmän johdon tulisi olla yrityksissä työtä edistävä, mutta selkeää on, ettei näin ole (aina-

kaan yhden vastaajan yrityksessä). Ylempi johto monesti päättää mitä projekteja tehdään ja kuinka, joten heillä on suurta päätösvaltaa ja olisi hyvä, että he käyttävät tätä valtaa oikein. Vaihtuvuuden suhteen selkeä ongelma on se, jos vain yksi henkilö tietää, kuinka jokin toiminto toimii, jolloin kyseinen henkilö on suuri riski. Onko ratkaisu tähän palkata enemmän työntekijöitä ja jokaisesta toiminnosta vastaa vähintään kaksi henkilöä? Se on yksi vaihtoehto, varmaa kuitenkin on, ettei lainauksessa esiintynyt tilanne ole ideaali.

Kuten on nyt nähty, vastaajien projekteissa sisäiset tekijät ovat olleet enimmäkseen positiivisia, mutta myös haasteita on ilmennyt. Suurimpia haasteita näyttää olevan vaihtuvuus ja henkilöstön palkkaus. Negatiivisten tekijöiden vaikutuksen minimoimiseen on kuitenkin löydetty keinoja. Vastauksissa nousee esille avoimuus ja säännölliset palaverit keinoina, joilla negatiivisia vaikutuksia on pyritty pitämään kurissa. Dokumentaation puutetta on pyritty paikkaamaan etäviestintien käytön kautta.

## **5.9 Onnistumiset**

Onnistumisten saralla aineistossa ilmenee useasta eri aiheesta, mutta Scrum tulee useammissa vastauksessa esille. Scrum oli vastaajien joukossa yleisin ketterän kehityksen viitekehys: 26:sta projektista Scrum oli käytössä yhdeksässä. Vastaajien mukaan Scrum on tehnyt tekemisestä selkeämpää ja antanut myös hyvää rytmiä työn tekoon. Scrum mainitaan myös tehtävien suunnittelun ja läpikäynnin yhteydessä, joka vastaajan mukaan auttaa aikatauluttamaan tekemistä. Eräs vastaaja puhuu onnistuneesta kehityssyklisen suunnittelusta. Kehityssyklisen suunnittelu oli nähty onnistuvan kaikille mielekkään tekemisen löytämisessä ja etukäteen suunnittelussa, missä syklissä mikäkin ominaisuus kehitetään ja missä testataan. Tämä antaa vastaajan mukaan kaikille kokonaiskuvan, milloin minkäkin ominaisuuden tulisi olla valmis. Säännölliset palaverit antoivat vastaajan mukaan hyvä valmiudet suunnitella omaa tekemistä ja ajankäyttöä. Toinen vastaaja niin ikään nosti tilanpalaverit onnistuneiksi.

Ketterän kehityksen periaatteisiin liittyen pieni ja osaava tiimi on nähty onnistumisena. Toisaan arvostava tiimi nähdään turvallisenä paikkana kehittää asioita ja kommunikaatiota. Kommunikaation onnistumista puoltaa myös toinen vastaaja sanoen keskusteluiden ja työnteon vaiheiden säilyvyyden muistissa auttavan kokonaisuuden hallitsemisessa. Projektinhallinta oli

nähty erään vastaajan mukaan onnistuneena vaikkakin se oli ollut kokonaan etätyössä.

Muita onnistumisen kohteita olivat seuraavat:

- selkeät vaatimukset ja tehtävät, iteratiivinen toiminnan tarkastelu, avun saaminen tarvittaessa,
- teknologinen osaaminen ja laaja kokemus osalla tiimistä,
- tiivis yhteistyö asiakkaan kanssa,
- etätyömahdollisuus sekä
- Asynkronisempi malli on tuonut näkyvyyttä, joustavuutta ja tehokkuutta, joka on johtanut onnistumisiin sovellusten kehityksessä ja työntekijöiden tyytyväisyydessä.

## **5.10 Koronaviruspandemian vaikutus**

Koronaviruspandemian vaikutus nähtiin odotetusti etätyöhön siirtymisellä. Moni vastaajista kertoo, kuinka etätyön haasteet olivat jo ennen pandemiaa tuttuja ja toisaalta toisista vastauksista tulee ilmi, kuinka etätyöhön siirryttiin vasta pandemian myötä. Yksi vastaajista kertoo seuraavasti siirtymisestä:

Aluksi vaati opettelua, miten kahvitunteja ja käytäväkeskusteluita saadaan nostettua esiin liimaamaan porukkaa kasaan. Aluksi online-kahvitunnit tuntuivat teennäisiltä, eikä niissä varmasti koskaan saavuteta samalla tavalla sitä hyötyä ja rentoutta kuin paikan päällä, mutta kyllä ne olivat paljon tyhjää parempia. SAFen seremonioiden vetäminen verkossa tuntui aluksi hurjalta, kun oli juuri tottunut tekemään homman upeasti paikan päällä, mutta alustat kuten Miro auttoivat tässä todella paljon. Kyllä se verkossa on edelleen paljon hankalampaa eikä yhtä tuottoisaa kuin paikan päällä, mutta työkalujen kehitys ja niihin tottuminen on auttanut saamaan tehoja irti myös etäsessioista.

Moni vastaaja puhuu etätyön lisäksi hybridityöskentelystä, kuinka osa tekijöistä kutsutaan läsnä-palaveriihin ja yksi jopa harmittelee, kuinka ihmiset eivät käy samalla tapaa toimistolla, jolloin yhteishenki kärsii ja on hankalampi huomata, jos työkaverilla on liikaa työkuormaa.

Koronaviruspandemian uutisointiinkin oli eräässä yrityksessä keksitty ratkaisu:

Työchatissa oli erikseen varattu kanava korona-uutisille ja tehtiin yhteinen linjaus, että korona-uutisia ei linkitettäisi ja koronasta ei keskusteltaisi muilla kanavilla, jotta niitä ei olisi kaikkien pakko seurata

Tulevaisuudessa samankaltaista ratkaisua voidaan hyödyntää muissakin tilanteissa, eikä vain jonkin sairauden aiheuttamassa pandemiassa. Lainausta esittääkin neroutta, jota löytyy yritysten henkilöistä.

Eräs vastaaja puhuu lisäksi ohjelmistokehittäjän näkökulmasta pandemian negatiivisesta vaikutuksesta:

Ohjelmistokehittäjälle se näkyi lähinnä isoina epäonnistumisina. Introvertit ennestään siiloutuneet putkiaivot ovat voittaneet ja päässeet valtaan eli uusien ajatusten ja moninaisuuden virta tuotekehitykseen on onnistuttu torppaamaan lähes täysin. Kukaan ei halua luopua saavutetuista eduista (teams palaverissa ketään ei pysty edes yrittämään osallistamaan). Etukäteen suunnittelu loppuu, koska osaajat saavat tehdä kuten ennenkin ja siihen ei kaivata ideoita sekoittamaan. Hiljainen tieto ei siirry ja on pelkkiä mastereita (jotka kohta eläköityy) ja kukaan ei opasta uusia - paitsi jos sattuu olemaan oma poika.

Edelliseen lainaukseen liittyen etätyö ja työhön liittyvät eri alueet pitää saada toimimaan kunnolla, jotta erilaiset henkilöt otetaan huomioon. Eihän toimistotyössäkään toimita samalla tavalla kaikkien kohdalla, joten toimintatavoissa tulee miettiä, mitkä ovat parhaita juuri tiettyä henkilöä varten.

Vaikka pandemian vaikutus näkyikin suuresti etätyöhön siirtymisessä, on tärkeää huomata, kuinka se vaikuttaa projektinhallintaan ja itse työntekoon. Viimeinen lainaus lisäksi antaa hyvän näkökulman vanhempien työntekijöiden eläköitymisestä. Jos haluamme, että tulevaisuudessa riittää tekijöitä, on nuoremmat henkilöt otettava paremmin huomioon. Jatkossa on keksittävä keinoja, joilla hiljaista tietoa saadaan liikutettua muille etätyössä ja, kuinka muiden opastaminen onnistuu parhaiten. Selkeästi on kuitenkin tiedossa, kuinka aiemmin toimistossa tehty työ on mahdollista tehdä joko kokonaan etätyönä tai sitten hybridityönä.

## 6 Pohdinta

Tässä luvussa tarkastellaan tutkimuksen tulosten suhdetta aikaisempaan tutkimukseen, pohditaan tutkimuksen merkitystä niin käytännön osalta, kuin tutkimuksellisesti. Lisäksi pohditaan, mitä rajoituksia tällä tutkimuksella esiintyy sekä lopuksi annetaan jatkotutkimuskohteita.

### 6.1 Vertailu aikaisempaan tutkimukseen

Jos tutkimuksen määrällisiä tuloksia verrataan aikaisempaan tutkimukseen, niin kommunikaation sekä vaatimusten ja laajuuden hallinta ovat selkeitä kohteita, jotka nousevat samankaltaisina (Demir 2009). Demirin tutkimuksessa laajuuden ja vaatimusten hallinta sekä kommunikaatio olivat yleisimpiä haasteiden kohteita, ja tässäkin tutkimuksessa kommunikaation, laajuuden ja vaatimusten hallinta sekä näihin liittyen muutosten hallinta nähtiin haasteellisimpina. Kommunikaatio yleisesti sekä muutosten yhteydessä on nähty useammassakin tutkimuksessa haasteena ohjelmistokehitysprojektien hallinnassa (Cho 2010; Santos, Figueiredo ja Marques 2023; Marnada ym. 2022). Huomionarvoisena tutkimuksena kommunikaatioon liittyen on Santos, Figueiredo ja Marques (2023), sillä se tutki samoin koronaviruspandemian aikaisia projekteja. Riskienhallinta oli samoin tässä ja Demirin tutkimuksessa haasteiden joukossa.

Demir (2009) raportoi henkilöstön palkkauksen olleen viiden haasteellisimman alueen sisällä, kun taas tässä tutkimuksessa henkilöstöresurssien hallinta ei juurikaan koettu haasteelliseksi. Vaihtuvuus on puolestaan ongelma, joka ilmenee yleisenä ongelmana aikaisemmassa tutkimuksessa (Anand ja Dinakaran 2015; López-Martínez ym. 2016). Tässäkin tutkimuksessa vaihtuvuus oli yleisesti ilmaistu haaste yhdistettynä työntekijöiden määrään.

Demir (2009) näki projektin suunnittelun ja arvioinnin olevan suuri haaste projektinhallinnassa. Tämän tutkimuksen kohdalla on huomattu, kuinka projektin suunnittelu on nähty positiivisena tekijänä projektinhallinnassa, kuten taulukosta 18 nähdään.

Projektin johtajuuteen liittyvät haasteet esiintyvät myös Cho (2010) ja Anand ja Dinakaran

(2015) tekemissä tutkimuksissa liittyen Scrum-kehukseen. Johtajuus nähtiin Demirin tekemässä tutkimuksessa haasteena neljänneksessä projekteista. Projektin vetäjä ei puolestaan nähty haasteellisena tekijänä Demirin tekemässä tutkimuksessa ja tässäkin tutkimuksessa se nähtiin hyvin positiivisena vaikuttajana. Akif ja Majeed (2012) löysivät tuoteomistajan ja Scrum Masterin keskeyttävän työtä turhaan, joka ilmeni myös erään vastaajan toimesta siten, että se vähentyi etätyöhön siirtymisen takia.

Projektinhallintakehys nähtiin tässä tutkimuksessa erittäin positiivisena tekijänä, vain yhdellä vastaajalla oli negatiivinen kokemus kehysten käytöstä. Aikaisemmassa tutkimuksessa Scrum-kehys on nähty haasteellisena (Cho 2010). Esimerkiksi kehysten seremonioiden määrä tuli esille, joka on vaikuttanut negatiivisesti.

Ketterään kehitykseen ja sen viitekehukseen liittyen dokumentaatio on tunnistettu aikaisemmassa tutkimuksessa haasteeksi, eikä tämä tutkimus ole poikkeus. Dokumentaation hallinta oli vastaajien mukaan yksi haasteellisimpia osa-alueita.

## **6.2 Tutkimuksen merkitys**

Tällä tutkimuksella on selkeä merkitys niin tieteellisessä mielessä, kuin käytännön. Tutkimuksessa saatiin ohjelmistokehitysprojektien ammattilaisten näkemyksiä, kuinka projektinhallintaan liittyvät asiat ovat sujuneet koronaviruspandemian aikaan. Avokysymysten vastaukset antavan selkeän kuvan missä on onnistuttu, mutta etenkin missä on parannettavaa. Ohjelmistokehitysprojektien haasteellisuus on selkeästi tiedossa, joten tutkimus tuo tärkeää tietoa käytäntöön, missä haasteita voi ilmetä ja myös kuinka niitä voidaan hallita. Koska etä- ja hybridityö on hyvin suosittua nykyään, eikä tulevaisuudessa kovinkaan suurta muutosta ole nähtävissä, tämä tutkimus tulee olemaan relevanttia, vaikka kyseessä olikin koronaviruspandemian aikaiset projektit. Syynä tähän on se, että suurin osa haasteista ja ongelmista ilmeni työtavan takia.

Koska aikaisemmasta tutkimuksesta löytyy aukko koskien projektinhallintaa koronaviruspandemian aikaan, tieteellisesti katsottuna tämä tutkimus antaa uutta tietoa, kuinka koronaviruspandemian aikainen projektityöskentely on sujunut. Tämä tutkimus tuo myös yleisellä tasolla ajankohtaista tietoa tutkimusalalle, kun puhutaan korkealla tasolla ohjelmistokehi-



tysprojektien hallinnasta, eikä suinkaan mennä tarkasti johonkin osa-alueeseen. Tutkimuksen myötä tulee myös mahdollisia jatkotutkimuskohteita, kommunikaatioon liittyen yksi selkeimmistä.

### **6.3 Tutkimuksen rajoitukset**

Tutkielma suoritettiin kyselytutkimuksen muodossa, joka kyllä on sopiva menetelmä tämän tyyppisille tutkimuksille, joissa halutaan hankkia tietoa alan ammattilaisilta. Kyselytutkimuksen tekemisen on erittäin helppo epäonnistua. Tämä kyselytutkimus olisi ollut mahdollista toteuttaa paremmin, jonka puolesta selkeästi puhuu vastausten vähyys, joka onkin selkein rajoitus tutkimuksessa.

Tutkimuksen rajoituksia vastausten määrän lisäksi löytyy testaamisen puolelta. Kyselylomaketta ei testattu kohderyhmän kanssa, jolloin mahdollisia lomakkeen puutteita ei saatu kitkettyä ennen vastausten keräämisen aloittamista. Lomaketta oli kuitenkin mahdollista muokata sen lähettämisen jälkeen, jota tehtiinkin hiukan, mutta se ei poista rajoitusta testaamisen puuttumisesta. Toinen rajoitus oli kyselylomakkeen lähettäminen kohderyhmälle. LinkedIn-julkaisut osoittautuivat erittäin huonoksi ratkaisuksi, sillä niistä ei tullut vastauksia yhtä lukuun ottamatta. Myös sähköpostien lähettäminen siinä toivossa, että kyselyä olisi voinut jakaa eri yritysten sisällä osoittautui haasteelliseksi. Vain muutamasta yrityksestä tuli selkeä viesti, että kyselyä oli jaettu eteenpäin. Molléri, Petersen ja Mendes (2020) raportoimien kyselytutkimuksen tehtävistä tosiaan vain instrumentin validointi jäi tekemättä, joten tämä kyselytutkimus on toteutettu ohjeiden mukaisesti.

Itse lomake on myös yksi mahdollinen rajoittava tekijä tutkimuksessa. Kyselyssä oli lopulta 20 kysymystä, joista useampi oli avokysymyksiä, joten on mahdollista, että vaikka kyselyn aloittaneita on kohtuullinen määrä, he eivät lopulta lähettäneet vastauksia juuri sen pituuden takia. Kyselyn pituuden puolesta puhuu se seikka, että analytiikan mukaan noin puolet vastaajista käyttivät vastaamiseen yli puoli tuntia.

Vaikka kyselyn vastausmäärä on alhainen, tutkimuksen tulokset ovat valideja aineiston laadullisen luonteen takia. Vastaukset puolestaan voidaan nähdä luotettavina koskien niiden liittymistä koronaviruspandemian aikaiseen projektityöskentelyyn, sillä suurin osa vastaajis-

ta oli antanut ajankohdan, milloin he työskentelivät projektissa. Reliabiliteettia eli luotettavuutta haastaa lisäksi se tekijä, että projektien ajankohdasta on jopa vuosia, joten vastaukset voivat olla vääristyneitä riippuen vastaajien muistista. Koska vastaukset olivat kuitenkin suhteellisen yksityiskohtaisia, ei tämä riski ole oleellinen. Aineiston analyysissä on lisäksi otettu huomioon vain ne vastaukset, jotka antavat selkeästi jotain näkökulmaa koskien aihetta. Laadullisen tutkimuksen luonteen takia vastaukset ovat subjektiivisia, mutta kuten on nähty, vastaukset ovat samankaltaisia, kuin on kirjallisuudessa tullut esille. Määrällisen aineiston puolesta ei kuitenkaan voi johtaa mitään yleisiä johtopäätöksiä, vaan ne toimivat enemmänkin suuntaa näyttävinä ja ideoita antavina.

## **6.4 Jatkotutkimuskohteita**

Selkein jatkotutkimuskohde on kommunikaatio. Se on tullut vastaan haasteena jo monessa tutkimuksessa vuosien varrella tämän tutkimuksen lisäksi. Kommunikaatiosta tulisi selvittää mistä haasteet johtuvat tutkimalla esimerkiksi yritysten kommunikointikäytänteitä. Kommunikaation osalta voi keskittyä joko toteuttajaorganisaation sisäiseen kommunikaatioon tai kommunikaatioon toteuttajan ja tilaajan välillä, sillä molemmat ovat kohteita, joissa haasteita esiintyy. Toinen jatkotutkimuskohde on vaatimusten hallinta. Vaatimusten hallinta nähtiin tässä tutkimuksessa yhtenä haasteellisimmista niin yleisesti vaatimusten ja laajuuden osalta, mutta myös niiden muutosten osalta. Vaatimusten ja niiden muutosten hallinta on ketterässä kehityksessä keskeisessä roolissa, joten on tärkeä selvittää, mistä haasteet johtuvat. Selvityksen lisäksi ratkaisukeinoja tulee tutkia. Koska määrälliset tulokset näyttävät, että haasteita ilmenee hyvin usein projekteissa, kokonaisvaltainen tutkimus ohjelmistokehitysprojektien haasteellisuudesta olisi paikallaan. Olisi syytä tutkia, mitkä ovat juurisyöt projektien haasteellisuudelle, sillä samat teemat toistuvat tutkimuksissa. Hybridityön ollessa kyselyn vastaajien kesken suosituin työtapo herättää kysymyksen: Yritetäänkö juuri hybridityöllä pienentää etätyön negatiivisia vaikutuksia? Siinä voisi olla ideoita jatkotutkimukselle.

## 7 Johtopäätökset ja yhteenveto

Tämän pro gradu -tutkielman tarkoituksena oli selvittää kyselytutkimuksen avulla missä ohjelmistokehitysprojektien hallinnassa on haasteita sekä selvittää, kuinka haasteita on pyritty minimoimaan. Kolmantena tarkoituksena oli selvittää missä oli kuitenkin koettu onnistumisia. Tutkimuksessa projektit rajattiin koronaviruspandemian aikaisiin, jolloin työskentely siirtyi etätyöhön ja myöhemmässä vaiheessa hybridityöhön. Täten tutkimuksen tulokset kertovat etenkin etä- ja hybridityön haasteista. Tutkielman tutkimuskysymys oli seuraava:

Kuinka koronaviruspandemia on vaikuttanut ohjelmistoprojektien hallintaan?

Koronaviruspandemian vaikutus näkyy etenkin etätyöhön siirtymisellä, jonka seurauksena haasteita ilmenee. Suurimmat haasteet ilmenevät kommunikaatiossa ja ryhmäytymisessä, kun mietitään pandemian vaikutusta.

Tutkielma aloitettiin kirjallisuuskatsauksella antamaan pohja projektinhallinnalle ohjelmistotalalla ensin menetelmien muodossa, jonka jälkeen siirryttiin tutkimukseen haasteista. Aikaisemman tutkimuksen pohjalta tuli esille missä alueissa on ollut haasteita niin yleisesti ohjelmistokehitysprojektien hallinnassa, kuin Scrum-kehyksen kohdalla. Aikaisemman tutkimuksen esittelyn jälkeen siirryttiin esittelemään tämän tutkimuksen tekeminen, jonka jälkeen esiteltiin aineiston analyysin tuloksia. Tulosten pohjalta haasteellisimpia alueita on seuraavat:

- kommunikaatio,
- aikataulun hallinta
- vaatimusten hallinta,
- laajuuden hallinta,
- muutosten hallinta,
- laadunhallinta,
- riskienhallinta sekä
- dokumentaation hallinta.

Vaikka kommunikaatio nähtiin haasteellisena, viestintäpalveluiden, kuten Microsoft Teams

ja Slack, nähtiin positiivisena vaikuttajana projektinhallintaan. Eri työkalujen käyttö ylipääntänsä nähtiin positiivisena kuten myös teknologiaosaaminen. Vastauksissa nostettiin etenkin Jiran roolia vaatimusten muuttumisessa ja hallinnassa.

Avokysymysten pohjalta nähdään, että työtavan valinnanvapaus nähdään positiivisena ja edistävänä tekijänä projektinhallinnassa. Avokysymyksissä nostetaan, kuinka etätyö on vaikuttanut positiivisesti projektinhallintaan, mutta myös negatiivisia vaikutuksia on esitetty.

Viestintäpalveluiden käyttö nähdään positiivisena tekijänä esimerkiksi dokumentaation hallinnan suhteen. Etäviestinten käyttö yhdistettynä etätyöhön on tuottanut haasteita etenkin vaadittujen henkilöiden tavoittamiseen.

Haasteena pidettiin myös työntekijöiden riittävyyttä projekteissa sekä henkilöstön vaihtuvuus. Sisäisen tekijän haasteena yhdistettynä etätyöhön pidettiin myös ryhmäytymistä. Etätyössä ryhmäytymistä on haasteellisempaa toteuttaa, kun kaikki projektin jäsenet eivät ole samassa sijainnissa.

Tämä tutkimus toi lisää ymmärrystä ohjelmistokehitysprojektien hallinnan haasteista, onnistumisista ja haasteiden minimoinnista. Tutkimuksen pohjalta tuli esille mahdollisia jatkotutkimusmahdollisuuksia, joita on syytä tutkia, parantaaksemme esimerkiksi projektien onnistumista. Tutkimus toi lisäksi tietoa käytäntöön, sillä haasteet ja onnistumiset ovat reaailmaailman asioita, joita on jo oikeasti tapahtunut.

## Lähteet

*14th State of Agile Report*. 2020. <https://info.digital.ai/rs/981-LQX-968/images/SOA14.pdf>.

*15th State of Agile Report*. 2021. <https://info.digital.ai/rs/981-LQX-968/images/SOA15.pdf>.

*16th State of Agile Report*. 2022. <https://info.digital.ai/rs/981-LQX-968/images/SOA16.pdf>.

Abrahamsson, Pekka, Outi Salo, Jussi Ronkainen ja Juhani Warsta. 2017. “Agile software development methods: Review and analysis”. *arXiv preprint arXiv:1709.08439*.

Akif, Raza ja Hammad Majeed. 2012. “Issues and challenges in Scrum implementation”. *International Journal of Scientific & Engineering Research* 3 (8): 1–4.

Anand, R Vijay ja M Dinakaran. 2015. “Issues in scrum agile development principles and practices in software development”. *Indian Journal of Science and Technology* 8 (35): 1–5.

Artto, Carlos, Miia Martinsuo, Jaakko Kujala ym. 2008. *Projektiliiketoiminta*. Aalto University.

Ashraf, Sara ja Shabib Aftab. 2017. “IScrum: An improved scrum process model”. *International Journal of Modern Education and Computer Science* 9 (8): 16.

Balaji, Sundramoorthy ja M Sundararajan Murugaiyan. 2012. “Waterfall vs. V-Model vs. Agile: A comparative study on SDLC”. *International Journal of Information Technology and Business Management* 2 (1): 26–30.

Beck, Kent, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries ym. 2001. *The agile manifesto*. <https://agilemanifesto.org/iso/fi/manifesto.html>.

Bentley, John E, Wachovia Bank ja NC Charlotte. 2005. “Software testing fundamentals—concepts, roles, and terminology”. Teoksessa *Proceedings of SAS Conference*, 1–12.

Big, Think ja Act Small. 2013. “CHAOS MANIFESTO 2013”. *CHAOS MANIFESTO* 2:52.

- Bootla, Penprapa, Olarn Rojanapornpun ja Pornchai Mongkolnam. 2015. “Necessary skills and attitudes for development team members in Scrum: Thai experts’ and practitioners’s perspectives”. Teoksessa *2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 184–189. IEEE.
- Butt, Shariq Aziz, Sanjay Misra, Muhammad Waqas Anjum ja Syed Areeb Hassan. 2021. “Agile project development issues during COVID-19”. Teoksessa *Lean and Agile Software Development: 5th International Conference, LASD 2021, Virtual Event, January 23, 2021, Proceedings 5*, 59–70. Springer.
- Cervone, H Frank. 2011. “Understanding agile project management methods using Scrum”. *OCLC Systems & Services: International digital library perspectives* 27 (1): 18–22.
- Ceschi, Martina, Alberto Sillitti, Giancarlo Succi ja Stefano De Panfilis. 2005. “Project management in plan-based and agile companies”. *IEEE software* 22 (3): 21–27.
- CHAOS Report 2015*. 2015. [https://standishgroup.com/sample\\_research\\_files/CHAOSReport2015-Final.pdf](https://standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf).
- Cho, Juyun Joey. 2010. “An exploratory study on issues and challenges of agile software development with scrum”. *All Graduate theses and dissertations*, 599.
- Clancy, Tom. 1995. “The chaos report”. *The Standish Group*.
- Da Silva, Fabio QB, Catarina Costa, A Cesar C Franca ja Rafael Prikladinicki. 2010. “Challenges and solutions in distributed software development project management: A systematic literature review”. Teoksessa *2010 5th IEEE International Conference on Global Software Engineering*, 87–96. IEEE.
- Degerli, Mustafa. 2022. “Declarations of Software Engineering Project Managers Managing Remotely: Provisions for Hybrid Working”. Teoksessa *2022 3rd International Informatics and Software Engineering Conference (IISEC)*, 1–5. <https://doi.org/10.1109/IISEC56263.2022.9998217>.
- Dekier, Łukasz. 2012. “The origins and evolution of Lean Management system”. *Journal of International Studies* 5 (1): 46–51.

- Demir, Kadir Alpaslan. 2009. “A Survey on Challenges of Software Project Management.” *Software engineering research and practice* 2009 (1): 579–585.
- El Emam, Khaled ja A Günes Koru. 2008. “A replicated survey of IT software project failures”. *IEEE software* 25 (5): 84–90.
- Eveleens, Johan ja Chris Verhoef. 2009. “The rise and fall of the chaos report figures”. *IEEE software* 27 (1): 30–36.
- Favare, J. 2002. “Managing requirements for business value”. *IEEE software* 19 (2): 15–17.
- Fernandez, Daniel J ja John D Fernandez. 2008. “Agile project management—agilism versus traditional approaches”. *Journal of computer information systems* 49 (2): 10–17.
- Ford, Denae, Margaret-Anne Storey, Thomas Zimmermann, Christian Bird, Sonia Jaffe, Chandra Maddila, Jenna L Butler, Brian Houck ja Nachiappan Nagappan. 2021. “A tale of two cities: Software developers working from home during the covid-19 pandemic”. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 31 (2): 1–37.
- Guide, A. 2001. “Project management body of knowledge (pmbok® guide)”. Teoksessa *Project Management Institute*, 11:7–8.
- Highsmith, James A. 2002. *Agile software development ecosystems*. Addison-Wesley Professional.
- Hughes, D Laurie, Nripendra P Rana ja Antonis C Simintiras. 2017. “The changing landscape of IS project failure: an examination of the key factors”. *Journal of Enterprise Information Management* 30 (1): 142–165.
- Ikonen, Marko, Elena Pirinen, Fabian Fagerholm, Petri Kettunen ja Pekka Abrahamsson. 2011. “On the impact of Kanban on software project work: An empirical case study investigation”. Teoksessa *2011 16th IEEE international conference on engineering of complex computer systems*, 305–314. IEEE.
- Juricek, Jan. 2014. “Agile project management principles”. *Lecture Notes on Software Engineering* 2 (2): 172.

- Jørgensen, Magne ja Kjetil Moløkken-Østvold. 2006. “How large are software cost overruns? A review of the 1994 CHAOS report”. *Information and software technology* 48 (4): 297–301.
- Kargl, Frank, Robert Schmidt, Antonio Kung, Christoph Bösch ym. 2019. “A privacy-aware V-model for software development”. Teoksessa *2019 IEEE Security and Privacy Workshops (SPW)*, 100–104. IEEE.
- Kaushik, Meenakshi ja Neha Guleria. 2020. “The impact of pandemic COVID-19 in workplace”. *European Journal of Business and Management* 12 (15): 1–10.
- Keele, Staffs ym. 2007. *Guidelines for performing systematic literature reviews in software engineering*.
- Kirovska, Nevenka ja Saso Koceski. 2015. “Usage of Kanban methodology at software development teams”. *Journal of applied economics and business* 3 (3): 25–34.
- Lapointe-Boisvert, Alexandra, Sébastien Mosser ja Sylvie Trudel. 2021. “Towards modeling acceptance tests as a support for software measurement”. Teoksessa *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 827–832. IEEE.
- Leung, Hareton KN ja Lee White. 1990. “A study of integration testing and software regression at the integration level”. Teoksessa *Proceedings. Conference on Software Maintenance 1990*, 290–301. IEEE.
- López-Martínez, Janeth, Reyes Juárez-Ramírez, Carlos Huertas, Samantha Jiménez ja Cesar Guerra-García. 2016. “Problems in the adoption of agile-scrum methodologies: A systematic literature review”. Teoksessa *2016 4th international conference in software engineering research and innovation (conisoft)*, 141–148. IEEE.
- Marnada, Primadhika, Teguh Raharjo, Bob Hardian ja Adi Prasetyo. 2022. “Agile project management challenge in handling scope and change: A systematic literature review”. *Procedia Computer Science* 197:290–300.
- McKenna, T ja SJ Whitty. 2013. “Agile is not the end-game of project management methodologies”. Teoksessa *Proceedings of the 10th Annual Project Management Australia Conference (PMOz 2013)*. University of Southern Queensland.



- Miller, Courtney, Paige Rodeghero, Margaret-Anne Storey, Denae Ford ja Thomas Zimmermann. 2021. “"how was your weekend?" software development teams working from home during covid-19”. Teoksessa *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, 624–636. IEEE.
- Molléri, Jefferson Seide, Kai Petersen ja Emilia Mendes. 2020. “An empirically evaluated checklist for surveys in software engineering”. *Information and Software Technology* 119:106240.
- Nguyen-Duc, Anh, Dron Khanna, Des Greer, Xiaofeng Wang, Luciana Martinez Zaina, Gerardo Matturro, Jorge Melegati, Eduardo Guerra, Giang Huong Le, Petri Kettunen ym. 2022. “Work-from-home and its implication for project management, resilience and innovation—a global survey on software companies”. *arXiv preprint arXiv:2202.04950*.
- Olson, Margrethe H. 1983. “Remote office work: Changing work patterns in space and time”. *Communications of the ACM* 26 (3): 182–187.
- Pargaonkar, Shraavan. 2023. “A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering”. *International Journal of Scientific and Research Publications (IJSRP)* 13 (08).
- Permana, Putu Adi Guna. 2015. “Scrum method implementation in a software development project management”. *International Journal of Advanced Computer Science and Applications* 6 (9): 198–204.
- Raharjo, Teguh ja Betty Purwandari. 2020. “Agile project management challenges and mapping solutions: A systematic literature review”. Teoksessa *Proceedings of the 3rd International Conference on Software Engineering and Information Management*, 123–129.
- Ralph, Paul, Sebastian Baltes, Gianisa Adisaputri, Richard Torkar, Vladimir Kovalenko, Marcos Kalinowski, Nicole Novielli, Shin Yoo, Xavier Devroey, Xin Tan ym. 2020. “Pandemic programming: How COVID-19 affects software developers and how their organizations can help”. *Empirical software engineering* 25:4927–4961.

- Ravindran, Vinitha. 2019. "Data analysis in qualitative research". *Indian Journal of Continuing Nursing Education* 20 (1): 40–45.
- Santos, Renata Bittencourt Mendonça dos, Paulo Soares Figueiredo ja Felipe Tumenas Marques. 2023. "Challenges to agile software project management practices in the context of the COVID-19 pandemic". *Gestão & Produção* 30:e9722.
- Talby, David, Arie Keren, Orit Hazzan ja Yael Dubinsky. 2006. "Agile software testing in a large-scale project". *IEEE software* 23 (4): 30–37.
- Thesing, Theo, Carsten Feldmann ja Martin Burchardt. 2021. "Agile versus waterfall project management: decision model for selecting the appropriate approach to a project". *Procedia Computer Science* 181:746–756.
- Van Casteren, Wilfred. 2017. "The Waterfall Model and the Agile Methodologies: A comparison by project characteristics". *Research Gate* 2:1–6.
- Wong, W Eric, Joseph R Horgan, Saul London ja Hiralal Agrawal. 1997. "A study of effective regression testing in practice". Teoksessa *PROCEEDINGS The Eighth International Symposium On Software Reliability Engineering*, 264–274. IEEE.

## **Liitteet**

### **A Tutkimuksen kyselylomake**



## Koronaviruspandemian vaikutus ohjelmistoprojektienhallintaan

Hei!

Tämä kysely on osa minun pro gradu -tutkielmaani. Kyselyn vastaamisen kautta annat minulle arvokasta dataa tutkielmani empiiriseen osioon. Tutkielmassani käsittelen kuinka koronaviruspandemia on vaikuttanut ohjelmistokehitysprojekteihin.

Vastaathan kyselyyn, jos olet ollut mukana vähintään yhdessä ohjelmistokehitysprojektissa kehityksessä (rooleina esim. ohjelmistokehittäjä, suunnittelija, testaaja) tai projektinhallinnassa (rooleina esim. Scrum master, projektipäällikkö) vuodesta 2020 lähtien. **Kyselyyn vastaamisen kautta saadaan tärkeää tietoa projektien onnistumisista ja haasteista, ja tämä tieto edesauttaa projektien hallintaa!**

Kyselyyn vastataan anonyymisti, ja siihen vastaamiseen menee noin 15-20 minuuttia.

Halutessasi voit osallistua **50 €:n S-ryhmän lahjakortin arvontaan, jonka lomakkeeseen sinut ohjataan tämän kyselyn päätökseksi**. Tämän kyselyn vastauksia ei voida yhdistää arvontakyselyn vastauksiin. Arvonnan tietosuojailmoituksen voit lukea [tästä](#). S-ryhmä ei ole osallisena arvonnassa tai kyselyssä.

Kyselyssä tulet ensin antamaan yksinkertaista taustatietoa (sukupuoli, ikä, kokemus), jonka jälkeen tulet arvioimaan viimeisimpiä ohjelmistokehitysprojekteja, jossa olit mukana. Arvioinnissa pohdit, kuinka haastavia eri projektinhallinnan osa-alueet olivat projektissa (numeerinen asteikko), mitkä tekijät vaikuttivat hallintaan ja kuinka (positiivinen vai negatiivinen vaikutus, numeerinen asteikko), ja kuinka näitä haasteita pyrittiin minimoimaan, sekä myös mitkä eri tekijöistä johtivat projektien onnistumiseen, jonka lisäksi kysyn projektien ja organisaation kokoa, työtapaa, projekteissa käytössä olleita projektinhallintamenetelmiä (esim. Scrum). Lisäksi voit antaa vapaamuotoista kommenttia projekteihin liittyen projektinhallinnan osa-alueista, sekä vaikuttavista tekijöistä. Lopuksi pohdit vapaamuotoisesti, kuinka työtapo on vaikuttanut projekteihin, sekä voit antaa kommenttia, kuinka pandemia on näkynyt projektinhallinnassa.

Paina "Seuraava"-painiketta aloittaaksesi. **Kiitos, kun osallistut!**

Yhteystietoni:

Paavo Karppinen, tietotekniikan maisteriopiskelija

paavo.a.karppinen@student.jyu.fi

p. 0401902842

1. Sukupuoli \*

Mies

Nainen

Muu

**2. Ikä \***

- 18-24 vuotta
- 25-34 vuotta
- 35-44 vuotta
- 45-54 vuotta
- 55+ vuotta

**3. Kokemus \***

- alle 2 vuotta
- 2-5 vuotta
- 5-10 vuotta
- 10+ vuotta

**4. Mikä on ollut vastualueesi viimeisimmissäsi projekteissa, joissa olet ollut mukana? Yksilöi vastualueet projekteittain ajanjakso(arvio)n kanssa (esim. projekti 1: ohjelmistokehittäjä, 2020-2021, projekti 2..., jne.) \***

---

---

---

---

---

---

**5. Kuinka monta jäsentä on projekteissa ollut mukana? Yksilöi projektit tekstikentässä kysymyksen 4 projektien nimien mukaan \***

- 1-10 jäsentä \_\_\_\_\_
- 11-50 jäsentä \_\_\_\_\_
- 51-100 jäsentä \_\_\_\_\_
- 100+ jäsentä \_\_\_\_\_

**6. Kuinka monta työntekijää on työskentelemiesi projektien organisaatioissa ollut? Yksilöi projektit tekstikentässä kysymyksen 4 projektien nimien mukaan, jos useampi \***





1

2

3

4

5

6

**Projektin sisäiset tekijät**

Henkilöstösuhteet	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Työtapa (etätyö/toimistotyö/hybridi)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Projektin vetäjä	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ylemmän johdon tuki	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Henkilöstön palkkaus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Vaihtuvuus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Vastuualueiden selkeys	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sisäiset sidosryhmät	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Organisaatorakenne	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Projektin ulkoiset tekijät**

Tilaaajan sitouttaminen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Odotukset (tilaaja vs toimittaja)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ulkoiset sidosryhmät (esim. konsultit, rahoittajat, yms.)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Muut**

Projektinhallintakehys (Scrum, XP, Kanban ja vastaavat)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tiedon keruu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tiedon hallinta	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Projektin suunnittelu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Vaatimusten ymmärrys	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Vaatimusten muuttuminen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Muutokset	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tavoitteiden selkeys	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Liiketoimintaympäristön ymmärrys	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Rahoituskuviot	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

12. Millä tavalla käytetyt työkalut ja teknologiat (kysymys 11 kohdan "Käytetyt työkalut ja teknologiat" alla) ovat vaikuttaneet projekteissa? \*

---



---



---



---



---









