



# Ohjelmistojärjestelmien valmistusprosessin tehostaminen

Viljami Kittilä

2020 Laurea





Laurea-ammattikorkeakoulu

## Ohjelmistojärjestelmien valmistusprosessin tehostaminen

Viljami Kittilä  
Tietojenkäsittelytradenomi  
Opinnäytetyö  
06, 2020

Viljami Kittilä

**Ohjelmistojärjestelmien valmistusprosessin tehostaminen**

Vuosi

2020

Sivumäärä 36

Tämän työn toimeksiantajana oli Prowledge Oy, joka on pieni ohjelmistoalan yritys. Tarkoituksena oli löytää ongelmia ja tarjota ratkaisuja toiminnanohjausjärjestelmien valmistusprosessissa, jota on tehty jo useamman vuoden ajan. Tavoitteena ei ollut löytää täysin uutta tapaa toimia kehityksessä, vaan ennemminkin kehittää vanhaa menetelmää siitä löytyvien heikkouksien pohjalta.

Teoriaosassa keskityttiin keräämään tietoa alalla vallitsevista tutkituista kehitysmenetelmistä, millaisia rooleja niistä löytyy ja miten kehitys etenee, mitkä ovat minkäkin roolin vastualueet. Lisäksi selvitettiin mitä etuja eri menetelmillä saavutetaan.

Tutkimus on laadullinen tutkimus, jossa keskitytään yksittäisen ilmiön tarkempaan tarkasteluun. Haastattelumenetelmänä toimi teemahaastattelut, joiden pohjalta tehtiin sisältöanalyysi, jossa puhtaaksi kirjoitetuista haastatteluista nostettiin esille halutut asiat. Näiden pohjalta tuotettujen yhteenvetojen pohjalta laadittiin rakenne, jonka tyypillinen projekti pitää sisällään.

Työn tuloksena syntyi projektin tyypillinen rakenne, sekä tietoa projekteissa esiintyvistä hankaluuksista. Tämän lisäksi syntyi kehitysehdotuksia, joiden pohjalta tilaajayritys voi kehittää toimintaansa. Lisäksi syntyi myös tietoa positiivisista asioista, jotka yrityksessä tehdään oikein, kuten asiakaslähtöinen projektinhallinta.

Työn tuloksena syntyneiden tuotosten arviointi on hankalaa, sillä kyseessä oli tutkimus, jossa tuotetaan tietoa, jonka pohjalta tehdään ehdotuksia. Tästä syystä konkretia onkin hyvin vähäistä ja varsinkin kehitystyö tapahtuu tutkimuksen ulkopuolella. Tuloksia voidaan arvioida sen mukaan, miten loogisia ne ovat suhteessa työn tavoitteeseen, sekä miten asiakasyritys ottaa ne vastaan. Johtopäätöksenä voidaan sanoa, että valmistusprosessissa on kehittämistä, niin roolien, kuin jatkuvan yhteistyön osalta kehittäjien ja asiakkaan välillä. Kehittämis ehdotuksena voisi kerätä määrällisesti asiakasyrityksiltä tietoa heidän kokemuksistaan ja tämän lisäksi keskittyä itse ohjelman teknisten osien paranteluun, joka suljettiin tässä tutkimuksessa ulos.

Asiasanat: Ohjelmistokehitys, ketterä kehitys, prosessinkuvaus

Viljami Kittilä

**Improving the Development Process of Software Systems**

Year 2020

Pages

36

---

The client for this bachelor's thesis was ProWledge Ltd., which is a small IT company. The goal was to find problems and offer solutions to the manufacturing process of the ERP systems, which has been done for several years. There was no goal to come up with a completely new approach to the process, but rather improve the old approach based on the weaknesses found in it.

The theoretical background section focuses in finding knowledge about the prevailing methods in the field, what kind of roles they have and how does the development advance, what are the responsibilities of each role. In addition, it was searched what benefits could be accomplished.

This is qualitative research, which focuses on a more detailed examination of an individual phenomenon. The interview method was theme interview, which was analyzed with a content analysis, where the desired issues were raised from the written interviews. From the summaries, made based on the analyzes, a structure of a typical process was created.

As the result a typical process structure was created and some knowledge of the issues that the projects have had. In addition, some improvement suggestions were made, based on which the client can improve their process. Furthermore, some positive notes were made about things that the client does well, like customer-oriented project management.

The evaluation of the research results is difficult because the nature of the research was to create knowledge and give suggestions based on the knowledge. Therefore, there is very little concrete things to show and the real development will be done outside of this research. The results can be evaluated based on how logical they are in comparison to the goals of the research, and how the client accepts them. In conclusion it can be said that the development process can be improved, some roles and continuing cooperation between client and the developers for example. To continue researching the next step could be to collect quantity of information from the client companies about their experiences and to concentrate on the technical improvements of the system, which were left out from this research.

Keywords: Software development, agile development, process overview

## Sisällys

1	Johdanto.....	7
2	Työn tausta ja tavoitteet .....	7
2.1	Tutkimuksen kohde, ongelma ja kysymykset.....	8
2.2	Aihealueen rajaus .....	8
2.3	Keskeiset käsitteet .....	8
2.4	Tutkimuksen rakenne.....	9
3	Ohjelmistokehitys ja menetelmät .....	9
3.1	Vesiputousmalli.....	10
3.2	Ketterät menetelmät .....	11
3.2.1	Kanban .....	12
3.2.2	Scrum.....	12
3.3	Lean IT.....	16
4	Tutkimusmenetelmät .....	19
4.1	Laadullinen tutkimus .....	20
4.2	Teemahaastattelu.....	20
4.3	Sisältöanalyysi .....	21
4.4	Reliabiliteetti ja validiteetti .....	22
5	Tiedonkeruu ja analyysi .....	23
5.1	Haastattelut .....	24
5.2	Haastatteluiden analyysi .....	24
6	Johtopäätökset .....	28
6.1	Ohjelmistokehitysmenetelmät ja niiden hyödyt .....	29
6.2	Tilaaajarityksen kehitysmenetelmä .....	29
6.3	Kehitysehdotukset .....	30
7	Pohdinta ja jatkokehitysehdotukset .....	32
	Lähteet.....	34
	Kuviot .....	36

## 1 Johdanto

Ohjelmistoprojektien valmistusprosessien kehittämistä on tutkittu jo paljon, Winston W. Royce julkaisi 70-luvulla artikkelin, jossa käsiteltiin ohjelmistokehitystä (Juvonen 2018, 16). Myöhemmin mm. Toyotan kehittämää Lean-ajattelutapaa on yhdistetty ohjelmistokehitykseen. Prowledge Oy:ssä toiminnanohjausjärjestelmiä on tehty jo 13 vuotta (Kittilä 2020), mutta viimeaikaisten ulkoisten tekijöiden vaikutuksesta on tarpeellista pyrkiä nopeuttamaan yrityksen toiminnanohjausjärjestelmän valmistusprosessia huomasti. Kokonaisprosessin tuntemuksen puute haittaa kehittämistä, sillä yrityksen tapaa valmistaa toiminnanohjausjärjestelmä, ei ole juurikaan tutkittu. Prosessin nopeutumisen mahdollistamiseksi on syytä tutkia koko prosessi, sen kaikki osatekijät ja niiden vastuut tähän saakka toteutetuissa projekteissa. Tämän tutkimisen avulla voidaan huomata mahdollisia puutteita ja heikkouksia nykyisessä menetelmässä, jolloin mahdollistetaan näihin puuttuminen ja sitä kautta alkava jatkuva kehittäminen. Ratkaisun avulla mahdollistetaan parempien ratkaisuiden ja palvelun tarjoaminen asiakasyrityksille. Opinnäytetyön aihe on tärkeä, sillä prosessin kehittämällä säästetään kustannuksissa, pienennetään kestoja ja löydetään erilaisia toimintatapoja, jotka perustuvat opittuihin tapoihin, joille ei ole olemassa loogista syytä. Ongelman ratkaisusta on hyötyä yritykselle siten, että heidän prosessinsa kuvataan, sitä analysoidaan ja annetaan kehitysehdotuksia, joiden avulla mahdollistetaan kestävä toiminnan kehittämisen periaatteiden tuominen osaksi jokapäiväistä työskentelyä. Asiakkaille hyötyä on siinä, että järjestelmät valmistuvat nopeammin ja luotettavammin. Yhteiskunnallisella tasolla hyötyä on tuotetun tiedon, sekä kilpailutilanteen parantumisen kannalta. Valmistusmenetelmiä tai -viitekehyksiä on tutkittu ainakin 70-luvulta lähtien. Kehityksen kulku on ollut käytännössä yleisestä mallista siirtyminen useampaan vaihtoehtoon, jotka nojaavat samoihin perusarvoihin, jotka kaikki nostavat esille jatkuvaa kehitystä, virheiden löytämistä ja niiden korjaamista. Työ liittyy ohjelmistoprojekteihin, eli tietojenkäsittelyalalla tehtäviin toistettavissa oleviin ohjelmistonkehitysprojekteihin.

## 2 Työn tausta ja tavoitteet

Tutkija on perehtynyt asiakasyritykseen aiemmassa työharjoittelussa, jonka hän teki kesällä 2019. Tutkija kiinnostui aiheesta harjoittelun aikana ja sopi asiakasyrityksen johdon kanssa tutkimuksen järjestämisestä. Tutkimuksen ajankohta on otollinen asiakasyritykselle, kun se suunnittelee oman toimintansa kehittämistä myös muilla toiminnanohjausjärjestelmiin liittyvillä osa-alueilla. Toiminnanohjausjärjestelmät, joita valmistetaan ovat itsenäisiä ohjelmistoja, jotka toimivat ulkoisilla palvelimilla. Järjestelmälle varataan verkkotunnus, jonka

kautta asiakkaat käyttävät valmista järjestelmää. Eri työntekijöille on olemassa eri tasoisia näkymiä, joissa he voivat merkitä työvaiheita ja edistymistä ylös.

## 2.1 Tutkimuksen kohde, ongelma ja kysymykset

Tutkimisen kohdeyritys on Prowledge Oy, joka työskentelee informaatiotekniikan alalla, yrityksen liikevaihto oli vuonna 2018 458000 € ja henkilöstömäärä samana vuonna 7. Yrityksen tuotteita ovat toiminnanohjausjärjestelmien lisäksi Know-ohjelmat, jotka ovat paperialan opetusohjelmia. (Kittilä 2020). Työn tutkimusongelma on toimintatapa, jolla toiminnanohjausjärjestelmiä valmistetaan Prowledge Oy:ssä.

Tutkimuskysymys onkin, mitä pitäisi muuttaa, jotta voitaisiin paremmin valmistaa toiminnanohjausjärjestelmiä Prowledge Oy:ssä. Apukysymyksenä ovat, mitkä ovat yleisesti käytettävät ohjelmistojärjestelmän valmistusmenetelmät, mitä hyötyä niiden käytöstä voi olla, sekä, miten Prowledge Oy:ssä tällä hetkellä valmistetaan toiminnanohjausjärjestelmiä. Työn mahdolliset tavoitteet ovat nykyisen menetelmän kuvaaminen, sekä siinä olevien ongelmien löytäminen, lisäksi nykyisen menetelmän vertaaminen valtakunnallisesti hyväksytyihin menetelmiin ja tämän vertailun pohjalta kehitysehdotuksien antaminen yritykselle.

## 2.2 Aihealueen rajaus

Ongelmasta tiedetään, että nykyinen tapa ei ole tarpeeksi tehokas vastaamaan volyyimia, jolla toiminnanohjausjärjestelmiä on tarkoitus tehdä. Aiheesta tiedetään sen verran, että on olemassa monia erilaisia toimintamalleja, joita alalla on kehitetty korvaamaan vanhentuneita toimintamalleja. Ongelmaa ei ole tutkittu juurikaan, joten ei ole tarpeellista selittää sitä perusteettomilla johtopäätöksillä.

Aihealue on hyvin tutkittua, sitä tutkitaan edelleen lisää, varsinkin alan isojen tekijöiden, kuten Spotifyn toimesta, joka on luonut oman ”Spotify mallinsa” (Ozkan & Tahrán 2019) tämän tutkimuksen kannalta rajaamme kuitenkin siten, että emme pyri löytämään koko alan kannalta uutta tutkimatonta osa-aluetta, vaan keskitymme yrityksen kannalta tutkimattomiin alueisiin. Tämä ei sulje pois sitä, etteikö alalle luotaisi uutta tutkimustietoa. Työn täsmällisestä aiheesta ei ole tehty varsinaista tutkimusta, sekä tutkimukset toimintamallien käyttöön-otoista ja toiminnasta keskittyvät pääasiassa suurempiin yrityksiin, jolloin tulokset eivät ole yhteneviä, sillä organisaatorakenteista johtuen, prosessit eivät ole samankaltaisia, jolloin tutkimustuloksia ei tule verrata keskenään väärinkäsityksien välttämiseksi.

## 2.3 Keskeiset käsitteet

Työn kannalta keskeisiä käsitteitä ovat ohjelmistokehitysmenetelmä, sekä ohjelmistonkehitysprosessi:



Ohjelmistonkehitysprosessi = tapahtuma sarja, jonka myötä valmistuu käyttökelpoinen ohjelmisto. Ohjelmistoprosessi kulku on pääpiirteiltään seuraavanlainen: Tarve, Määrittely, Suunnittelu, Toteutus, Käyttöönotto ja Päättäminen. (Juvonen 2018, 13.)

Ohjelmistokehitysmenetelmä on käsite, joka kuvaa menetelmää, periaatteita tai tapoja, jolla ohjelmistoa valmistetaan. Ohjelmistokehitysmenetelmiä on useampia ja niitä kehitetään alan huippujen toimesta jatkuvasti eteenpäin. Ohjelmistokehitysmenetelmiä ovat esimerkiksi vesiputousmalli, sekä ketterät menetelmät. (Juvonen 2018.)

Keskeiset käsitteet pitävät sisällään sekä teoriapohjan keskeisimmän osuuden, että empiirisen osan keskeisimmän osuuden. Mikäli nämä kaksi käsitettä saadaan täsmällisesti selvitettyä, on tutkimuksella mahdollisuus onnistua.

## 2.4 Tutkimuksen rakenne

Tutkimus etenee ajallisessa järjestyksessä tutkimuksen sisällysluettelon mukaisesti. Ensiksi tutkija päättää, mitä halutaan tutkia ja tekee selväksi, kenelle tutkitaan ja mistä näkökulmasta. Tämän jälkeen tutkija päättää tutkimismenetelmät. Kolmanneksi tutkija perehtyy teoriaan, joka koskee tutkimusta, teorialla vastataan kysymyksiin yleisesti käytettävistä kehitysmenetelmistä ja mitä hyötyä niiden käytöstä voi olla. Tämän lisäksi hän perehtyy myös tutkimusmenetelmien teoriaan, jotta osaa toteuttaa niitä oikealla tavalla. Teoria-osan jälkeen tutkija lähtee haastattelemaan henkilöitä, jotka ovat avain asemassa järjestelmän valmistusprosessissa. Haastatteluista saatu tieto analysoidaan ja analyysien pohjalta tuodaan vastaus kysymykseen, miten Prowledge Oy:ssä valmistetaan toiminnanohjausjärjestelmiä nykyään. Seuraavassa vaiheessa tutkija vertaa nykyistä kehitysmenetelmää teoriapohjassa valittuihin menetelmiin. Tämän avulla saadaan vastattua siihen, miten Prowledgen kehitys eroaa teoriapohjaan valituista menetelmistä. Tämän pohjalta tehdään löydöksiä, joiden avulla voidaan tehdä perusteltuja kehitysehdotuksia, varsinkin jos asiakasyrityksellä on ollut ongelmia eroihin liittyvillä osa-alueilla. Ehdotukset vastaavat kysymykseen, mitä voidaan tehdä toisin, jotta kehittäminen olisi tehokkaampaa.

## 3 Ohjelmistokehitys ja menetelmät

Ohjelmistokehitys tarkoittaa koko prosessia, joka tehdään, jonka tuloksena valmistuu ohjelma. Ohjelmistokehitysprosessi sisältää sarjan erilaisia aktiviteetteja, kuten vaatimusten määrittelyn, suunnittelun, koodaamisen ja testaamisen. Vaatimusten määrittelyssä selvitetään, mitä ohjelman täytyy tehdä, mikä sen tarkoitus on. Suunnitteluvaiheessa päätetään, miten vaatimukset kohdataan, eli miten ohjelma tehdään. Koodaamisvaiheessa tehdäänkin jo

varsinainen ohjelma. Testausvaiheessa testataan, että ohjelma toimii oikein. Ohjelmistokehitykseen kuuluu siis koko prosessi siitä hetkestä, kun asiakas hyväksyy tarjouksen, siihen hetkeen, kun valmis järjestelmä otetaan käyttöön.

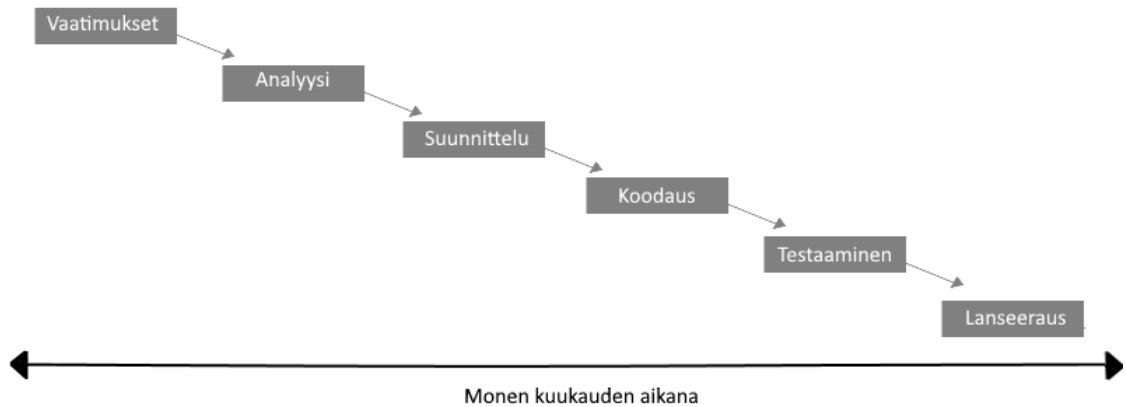
Vaikka ohjelmistokehitys on sinällään yksinkertainen prosessi, sen tekemiseen on monia erilaisia menetelmiä ja ne eroavat käytännössä jokaisen toimijan kohdalla. Yleisiä menetelmiä ovat vesiputousmalli, sekä ketterät menetelmät. Tämän lisäksi esimerkiksi testaukseen ja suunnitteluun itseensä kuuluu useita erilaisia menetelmiä, kuten vaikka protoilu, jossa asiakkaalle esitetään yksinkertaistettu versio tuotteesta, jolloin asiakas ja kehittäjä osaavat hahmottaa tuotteen paremmin ja puhuvat samoista asioista. Vaikka kahdella eri toimijalla olisi käytössä sama ohjelmistokehitysmenetelmä, saattaa heidän kehitystapansa silti poiketa toisistaan. (Juvonen 2018.)

Tutkimukseen valikoituneissa menetelmissä on mukana vesiputousmalli, eli se, josta pyritään pääsemään eroon. Vesiputousmalli antaa tutkimukseen näkökulmaa vanhasta tavasta, sen avulla pyritään tuomaan esille asioita, joita kannattaisi kehittää. Lisäksi mukana on Scrum, joka edustaa tyypillistä ketterää menetelmää, jossa on otettu huomioon asioita hyvin laajasti, niin itse kehittämisen, kuin asiakaskontaktin ja testaamisen kannalta. Kanban on mukana siitä syystä, että se on asiakasyrityksellä tietyn muotoisena käytössä, sekä se sisältyy moneen muihunketterään menetelmään ikään kuin lisänä. Lean on todella laaja kehitysmenetelmä tai viitekehitys, jolla taas pyritään tuomaan koko prosessin kehitystä. Lean tuo näkökulmaa koko prosessin kehittämiseen, joka pitää jossain määrin sisällään kaikki muut kehitysmenetelmät.

### 3.1 Vesiputousmalli

Vesiputousmalli (kuvio 1) on perinteinen ohjelmistokehitysmenetelmä, jonka perustaja on Winston W. Royce, joka tiedosti mallin heikkoudet ja julkaisi siitä toimivankin version. Huonojen siteerausten takia väärä versio sai jalan sijaan alalla ja muodostui standardiksi. (Juvonen 2018, 16) Vesiputousmallissa myynnistä vastaava henkilöstö, työnjohtaja ja asiakas suunnittelevat järjestelmän valmistuksen keston ja resurssit. Tämän jälkeen sovituista asioista koostetaan analyysi, jonka pohjalta suunnitellaan, millainen järjestelmä tehdään. Tämän jälkeen suunniteltu järjestelmä viedään ohjelmistokehittäjille, jotka kehittävät sen annetun ajan ja resurssien puitteissa. Kun järjestelmä on valmis, se testataan ja siirretään asiakkaalle. Tässä vaiheessa aikaa on yleensä kulunut liian kauan ja asiakkaan vaatimukset järjestelmää kohtaan ovat muuttuneet, sillä hän ei alussa osannut oikeasti sanoa, mitä he järjestelmältä halusivat, joten hän ilmoitti varmuuden vuoksi mahdollisimman kattavalla listalla, mitä siinä voisi olla. Tämän listan ansiosta kehittäjät saivat itselleen ison läjän töitä ja tiukat resurssit. Näiden kahden yhteen tulemana rakennettiin järjestelmä, jossa on kyllä paljon asioita, mutta on myös paljon virheitä ja epäolennaisuuksia. Asiakas on pettynyt ja ottaa yhteyttä kehittäjiin ja

vaatii korjauksia, jolloin järjestelmä on niin suuri, että jokainen korjaus on kallis. (Juvonen 2018.)



Kuvio 1: Vesiputousmallin rakenne (mukaiillen Bell, Bell & Orzen, 173).

On toki olemassa projekteja, jotka on viety vesiputousmallin avulla maaliin, mutta se vaatii pieniä projekteja ja onnistumisia. Kuitenkin n.50% järjestelmään päätyvistä asioista eivät tuota asiakkaalle todellista arvoa, budjetista noin 80% menee virheiden korjaamiseen ja kun testaus on painotettu projektin loppupäähän, niiden korjaaminen on kallista ja hankalaa. Vesiputousmallia käytetään usein vertauskohteina nykyisille menetelmille, sillä useat toimijat ovat pysyneet vesiputousmallin tyylisissä ratkaisuisissa, joissa on saatettu tähdätä johonkin uudempaan toimintamalliin, mutta on pikkuhiljaa palattu takaisin vanhoihin tapoihin. Vesiputousmalli vaatii paljon työjohtoa, hallintaa ja datan vaihtoa, päätökset tehdään työjohdon puolella ja kehittäjien tehtävänä on pärjätä niiden kanssa. (Bell & Bell & Orzen 2010,172.)

### 3.2 Ketterät menetelmät

Ketterät menetelmät ovat vuonna 2001 Utahissa kokoontuneiden ohjelmistokehittäjien parhaaksi kokemia menetelmiä, joiden edistämiseen on perustettu järjestö nimeltä Agile Alliance. Menetelmän pohjana käytettävät arvot on määritetty Agile Manifestossa (Agile Alliance 2001): "Löydämme parempia tapoja tehdä ohjelmistokehitystä, kun teemme sitä itse ja autamme muita siinä. Kokemuksemme perusteella arvostamme: yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja, toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota, asiakasyhteistyötä enemmän kuin sopimusneuvotteluja, vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa. Jälkimmäisilläkin asioilla on arvoa, mutta arvostamme ensiksi mainittuja enemmän." (Kananen 2017, 17.)

Ketteriä menetelmiä ymmärretään usein väärin, kuten ajatellaan, ettei ohjelmaa tarvitse suunnitella tai asiakas ajattelee, että projekti toteutetaan ketterästi, mutta ei ymmärrä, että ketterä vaatii myös hänen jatkuvaa osallistumistaan. Ketterät menetelmät vaativat kuitenkin

suunnittelua, sekä asiakkaan osallistumista projektiin, jotta projektissa saadaan kehitettyä asiakkaalle oikeanlainen lopputulos. Eräitä ketteriä menetelmiä ovat esimerkiksi Kanban ja Scrum, joista kerrotaan myöhemmin tarkemmin. Lean ei sinällään ole ketterä menetelmä, mutta Lean-ohjelmistokehitys hyödyntää ketteriä menetelmiä, sekä rohkaisee niiden käyttöön. (Juvonen 2018.)

### 3.2.1 Kanban

Kanban on näkökulma ohjelmistokehitykseen, joka perustuu Lean-periaatteisiin. Kanban toimii kaikenlaisissa projekteissa, sekä sen käyttö voidaan aloittaa suoraan, ilman uusia rooleja tai organisointia. Alussa tehdään taulu, jolla visualisoidaan työtä ja sen virtausta työvirran lävitse. Taulun tekoon tarvitaan kaikki työvirtaukseen osallistuvat työntekijät, jotta saadaan koko virtaus kuvattua todenmukaisesti. Kanban on japania ja koostuu sanoista Kan, joka tarkoittaa visuaalista ja Ban, joka tarkoittaa korttia. Kanban on kehitetty Toyotalla, jossa sitä käytettiin aikataulutusrjestelmänä. Kanban ei sisällä monia sääntöjä ja toimia, mutta sen avulla voidaan silti kehittyä valtavasti noudattamalla sen periaatteita. Visuaalisesta prosessin kuvauksesta, kuten taulusta, jossa kaikki tehtävät ovat järjestettynä, nousee esille häiriöt, sekä useimmiten jopa isoimmat ongelmat. (Hammarberg & Sunden 2014, 49.)

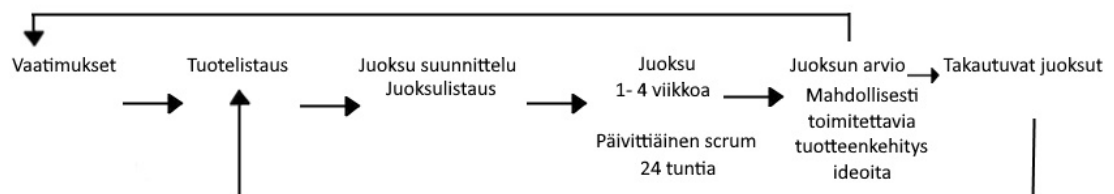
Kanban projektissa on kolme elementtiä: visualisointi, prosessissa olevan työn rajaaminen ja työ virtauksen hallitseminen. Visualisoinnilla tarkoitetaan sitä, että tuodaan tehtävä työ kaikkien näkyville, jolloin jokainen hahmottaa kokonaisprosessin, visualisoinnin avulla voidaan tuoda esille myös käytäntöjä, joita työntekijöillä on, sekä keskustella niiden hyvistä ja huonoista puolista. Prosessissa olevan työn rajaaminen, eli aseta rajamäärä, jonka verran tehtäviä sinulla voi olla samanaikaisesti käynnissä. Kun rajamäärät on asetettu, käynnissä olevat tehtävät suoritetaan nopeammin. Rajoittamalla tehtävien määriä prosessissa, saadaan olevassa olevat ongelmat nousemaan pintaan. Jotkut tehtävät siirtyvät hitaammin prosessin lävitse, tai palaavat prosessin alkuun tai pysähtyvät kokonaan. Mitä tahansa noista asioista tapahtuukin, se indikoi sitä, että menetelmää voidaan kehittää. Viimeinen periaate on työ virtauksen hallitseminen, siirretään työ sujuvasti ilman pysähdyksiä koko prosessin lävitse. Tästä alkaa jatkuvan kehittämisen kokemus, sillä kehitys ei ole koskaan valmista. Työvirtausta voidaan näet aina kehittää, aina on olemassa jokin hidastava tekijä (Hammarberg & Sunden 2014, 50). Kanban tarjoaa hyviä työkaluja projektin näkyväksi tekemiseksi, tutkimusyrityksessä on käytössä Trello, joka on eräänlainen verkkotyökalu, jossa voidaan soveltaa Kanban periaatteita. Kanban myös sopii muiden menetelmien kanssa hyvin yhteen, joten siinä on paljon potentiaalia.

### 3.2.2 Scrum

Scrum on ketteräkehitysmenetelmä, jossa tehtävät jaetaan osiin ja osat priorisoidaan. Korkeimmalle priorisoidut tehtävät tehdään aina ensin, jolloin asiakas saa aina arvokkaimmat

ominaisuudet tuotteesta ensimmäisenä. Scrumissa asiakas antaa palautetta tehdystä työvaiheesta, jolloin korkeimmalle priorisoitujen töiden kehitysehdotukset saadaan myös priorisoidua muun työn seassa, jolloin työskennellään taas korkeimmalle priorisoitujen tehtävien parissa. (kuvio 2) Scrum ei kerro, kuinka jokin projekti tehdään, vaan auttaa tekijöitä näkemään, minne projektissa on tarkoitus edetä. (Juvonen 2018, 18-22.)

Scrumissa projekti alkaa siten, että keksitään tavoite, johon pyritään pääsemään, sopiva maali, näkemys, mitä projektilta halutaan. Tämän jälkeen tämä tavoite jaetaan kaikkien projektissa työskentelevien ihmisten kesken, jolloin kaikki työskentelevät samaan suuntaan. Tämän jälkeen tavoite jaetaan vaatimuksiin, joita lopputuotteelta halutaan. Nämä vaatimukset toimivat tuotelistauksena, joka kertoo, millainen tuote on. Tuotelistaus saattaa muuttua projektin aikana, mutta se nähdään scrumissa positiivisena asiana, koska silloin tuote kehittyy edelleen. Tuotelistaus jaetaan julkaisutavoitteisiin, jotka ovat vaiheita, jossa on tarkoitus julkaista tuotteesta versio. Julkaisutavoitteet jaetaan juoksuihin, jotka ovat keskimäärin 1-2 viikkoa mittaisia kehitysjaksoja, joissa on tavoite saada jokin vaatimus valmiiksi. Juoksujen aikana kehitystiimi pitää päivittäisiä scrumeja, joissa käydään läpi päivittäin kehittämisen etenemistä. Päivittäisen scrumin lisäksi juoksuun sisältyy juoksun suunnittelu, juoksun arviointi ja takautuva juoksu. Suunnitteluvaiheessa tiimi suunnittelee tavoitteet seuraavalle juoksulle. Arvioinnissa katsotaan, miten hyvin näihin tavoitteisiin on päästy. Takautuvassa juoksussa pohditaan, mitä on tehty hyvin ja missä on kehittämisen varaa, sekä sovitaan askeleita, joita otetaan, jotta kehitytään edelleen. Lopulta, kun tarpeeksi monta juoksua on tehty, on edetty julkaisuvaiheeseen, jossa julkaistaan tuotteesta siihen asti valmistetut ominaisuudet. Tämän jälkeen tehdään seuraavaksi korkeimmalle priorisoituja vaatimuksia, jotka julkaistaan seuraavassa julkaisutavoitteessa. Julkaisutavoite määrittyy päätettyjen ominaisuuksien ja niille arvioitun ajan mukaan. (Layton & Morrow 2018, 2.)



Kuvio 2: Scrumin rakenne (mukaillen Layton & Morrow 2018, 10).

Scrum tiimissä on kolme roolia, kehittäjät, scrummaster ja tuotteenomistaja (kuvio 3). Toimintaan scrum vaatii jokaiselta jäseneltään omistautumista, rohkeutta muutokselle, arvostusta toisia kohtaan, avoimuutta ja fokuoitumista. Nämä viisi asiaa tunnetaan myös scrum arvoina. Jokaisella jäsenellä on selkeä rooli ja roolilla selkeät päivittäiset tavoitteet. Kaikki roolit ovat

keskenään tasa-arvoisia, jolloin jokainen on vastuussa omasta suorituksestaan. Tuotteen omistaja asettaa tavoitteet ja näkemyksen tuotteelle ja pitää niistä kirjaa. Omistaja myös ylläpitää tuotelistausta, tekee päätöksiä ja valintoja, vastaa kehitysjonon luonnista ja sen näkyvyydestä, optimoi kehitystiimin työskentelyä, ottaa vastaan kehitysehdotuksia ja priorisoi ne muiden vaatimusten kanssa, asettaa tavoitteet julkaisuille ja juoksuille, valitsee juoksun aikana kehitettävät tuotteen osat, vastaa talouspuolesta ja kommunikoi asiakkaiden ja sijoittajien kanssa. Hänen täytyy myös olla saatavilla päivittäin ja hän päättää hyväksytäänkö jokin valmiiksi saatu vaihe vaiko ei. Scrummaster vastaa scrumin opettamisesta ja järjestää scrum tapahtumat. Scrummaster ei tee päätöksiä tai kehitä tuotetta, vaan auttaa kehittäjiä kehittämään tuotetta auttamalla heitä löytämään ratkaisuja. Kehittäjät kehittävät tuotetta, sekä päättävät miten kehitys tehdään ja miten paljon kehitystä tehdään. Kehittäjät sopivat keskenään, mitä kukin tekee päivittäisissä scrumeissa, jolloin saadaan aina keskitettyä työvoima korkeimman prioriteetin omaavaan työvaiheeseen. Näiden roolien lisäksi projekteihin liittyy aina asiakkaat ja sijoittajat, jotka antavat palautetta aina juoksujen arvioinnissa, kun kehitystiimi esittää valmiiksi saatuja vaiheita. Lisäksi varsinkin alkuvaiheessa, kun yritys siirtyy scrumiin, voidaan käyttää scrum mentoria, joka seuraa sitä, kuinka hyvin jäsenet noudattavat scrumia ja tunnistaa vanhoja tapoja, jotta niistä päästäisiin eroon. Scrum mentori opastaa myös epäselvissä tilanteissa ja helpottaa siirtymävaihetta monella tapaa. (Layton & Morrow 2018, 14; 33-34.)



Kuvio 3: Scrumin roolit (mukaillen Layton & Morrow 2018, 11).

Scrumin vahvuuksia vanhaan ei ketterään kehitysmalliin, eli vesiputousmalliin ovat ne, että vanhassa vesiputousmallissa kehittäjät näkivät sen, mitä ovat kehittämässä paperilla, kun taas scrumissa kehittäjät näkevät sen tehdessä tuotetta. Toisin sanoen vanhassa mallissa

tuote suunniteltiin ja annettiin jollekin tiimille tehtäväksi aikarajan ja resurssien sisällä, joka oli sovittu asiakkaan ja projektijohdon kanssa. Tämän jälkeen tuote valmistettiin ja lähetettiin asiakkaalle, joka näki tuotteen silloin ensimmäistä kertaa. Projektin kestosta riippuen virhemarginaali kasvaa, jolloin koko projekti voi epäonnistua tai vaatia isoja korjauksia. Scrumissa tuotetta ei niinkään suunnitella, tiedetään vain, että asiakas mitä tuotteella pitää kyetä tekemään. Nämä ovat tuotteen vaatimuksia, jotka sitten priorisoidaan ja jaetaan juoksuille. Priorisoinnissa käytetään vaatimuksena korkeimman taloudellisen arvon ja korkeimman riksien omaavia vaatimuksia. On järkevää tehdä korkeimman taloudellisen arvon omaavia vaiheita, koska niiden julkaisun jälkeen saadaan rahaa, mutta on myös kannattavaa tehdä riskialtteimmat vaiheet alussa, jotta voidaan käyttää mahdollisimman paljon resursseja niiden toimivuuden varmistamiseksi tai todeta ne mahdottomiksi ja lopettaa tuotteen valmistus ajoissa, ennen kuin on tuhlatu resursseja liiaksi. Tämän lisäksi kehittäjillä on joka juoksun jälkeen mahdollisuus esittää jotain toimivaa ominaisuutta sijoittajille ja asiakkaalle ja näin saada palautetta siitä, toimiiko se oikein, vai tarvitseeko sitä vielä kehittää. Näin ollen asiakas saa itselleen aina haluamaansa laatua, ja väärinymmärryksistä johtuvat virheet jäävät pois. (Layton & Morrow 2018, 31.)

Scrumiin siirtyminen alkaa siitä, että hankitaan tarkastus, jossa määritetään oma projekti, metodit ja resurssit, sekä muut projektin tekijät. Tämän jälkeen yritys etsii itselleen sertifioidun scrummasterin ja varmistaa, että kaikki projektiin osallistuvat saavat koulutuksen, jossa käsitellään scrumia. Seuraavaksi luodaan tiimi, jonka vastuulla on muutoksen aikaansaaminen ja tarvittavien muutosten määrittäminen, mahdollisesti hankitaan scrum mentori. Käyttöönotto aloitetaan yhdellä projektilla, joka on tarpeeksi haastava, mutta sisältää hieman kokeiluvaraa resurssien ja ajan kanssa. Luodaan toimiva ympäristö, jossa koko scrum tiimi työskentelee, sosiaaliset työtilat ja tarvittava tekninen tuki. Tehdään varjostamista ja tuplakehitystä, jotta kaikki kehittäjät oppivat työskentelemään eri kehitystyön osa-alueilla. Tehdään selväksi, mikä on valmiin vaatimustaso, tutkitaan työskentelyä ja mukaudutaan huomattujen asioiden perusteella. (Layton & Morrow 2018, 327.)

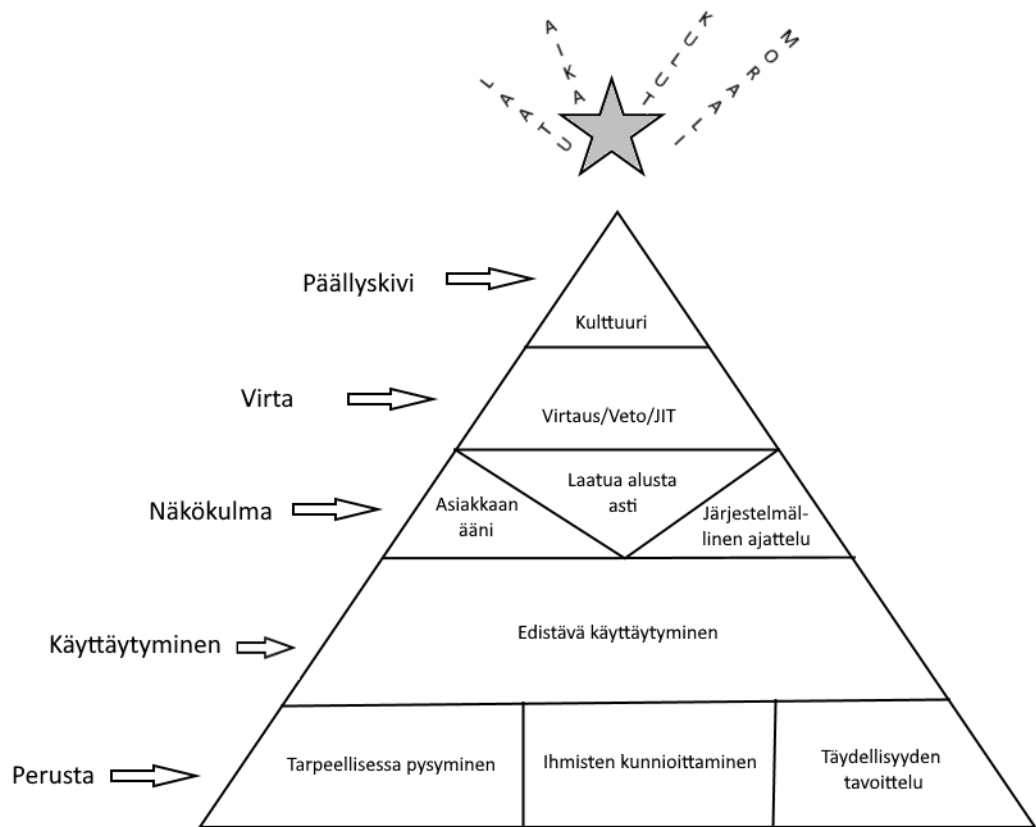
Scrum tarjoaa myös vaihtoehdon yrityksille, jotka eivät tee projekteja näin mittavissa puitteissa, vaan kaikenkokoiset projektit voidaan toteuttaa scrumilla. Scrumin ohessa voidaan käyttää monia teknisiä työkaluja, mutta scrumiin riittää myös kynä ja muistilaput. Muistilapuille kirjataan ominaisuudet ja ne sijoitetaan ilmoitustaululle priorisoinnin mukaan. Scrumissa suositaan kommunikointia kasvotusten, joten viestintätyökalutkaan eivät ole välttämättömiä. Scrum on muihin menetelmiin verrattuna hyvin ohjeistettu, scrumin toteuttaminen vaatii käytännössä yhteistä työpistettä kaikilta scrum-tiimiläisiltä, joka ei asiakasyrityksessä ole mahdollista. Verkkosovelluksilla on mahdollista tehdä jonkinlaista scrumia, mutta optimaalista ympäristöä sen harjoittamiseksi ei ole.

### 3.3 Lean IT

Lean IT pyrkii tuomaan Lean-mallia IT-näkökulmaan. Lean on Japanissa Toyotan kehittämä menetelmä, jossa pyritään luomaan asiakkaalle mahdollisimman paljon arvoa poistamalla hukkaa arvovirrasta. Arvovirta koostuu kaikesta, mitä tehdään, jotta paperilla oleva suunnitelma siirtyy valmiiksi tuotteeksi. Jotta arvovirtaa voidaan kehittää, täytyy kyetä tekemään yhteistyötä, jonka avulla kehittäjät siirtyvät paikallisesta oman toimialan ajattelusta kokonaiskuvan ajatteluun ja kehittämiseen. Hukka jaetaan kolmen M-kirjaimen mukaan, Mura, Muri, Muda. Mura tarkoittaa epätasaisuutta työn virtauksessa, joka johtuu kysynnästä, vaihtelusta ja laadusta. Asiakkaat toivovat erilaisuutta ja joustavuutta, mutta niiden saavuttamisessa pitää välttää turhaa monimutkaisuutta ja sekasortoa. Muri tarkoittaa ylikuormitusta, jossa työntekijät ja työvälineet laitetaan liian suuren työtaakan alle, jolloin tuloksena on paljon huonolaatuista korjaustarpeen omaavaa työtä. Muda on turha työ, eli työ, joka ei tuota arvoa asiakkaalle. Muda johtuu ylituotannosta, liiasta varastoinnista, odottelusta, kuljettamisesta, yliprosessoinnista, liikkeestä ja korjaamisesta, IT:ssä lisätään myös käyttämätön ihmisten luovuus ja potentiaali. Näitä kolmea hukkaa pyritään poistamaan standardisoiduilla tuotteilla, prosessin suunnittelulla, tukemalla tuotteiden palveluiden ja informaation sujuvaa virtausta, sekä kouluttamalla ihmisiä. (Loader 2019, 26.)

Lean yrityksen pyramidi (kuvio 4) rakentuu periaatteiden avulla, jota yritys ylläpitää. Kolme elementtiä, jotka tukevat vahvaa sosiaalista rakennetta Lean yritykselle ovat tarpeellisuudessa pysyminen, ihmisten kunnioittaminen ja jatkuva kehittäminen yhdistettynä täydellisyyden tavoitteluun. Tarpeellisuudessa pysyminen painottaa tärkeisiin asioihin keskittymistä, varsinkin, kun kyseessä on isompia projekteja. Tarpeellisuudessa pysymällä tehdään tarpeelliset asiat, jolloin turhat asiat jätetään tekemättä, jolloin voidaan keskittyä asioihin, jotka tuottavat asiakkaalle arvoa. Ihmisten kunnioittaminen keskittyy siihen, että kun ihmisiä kunnioitetaan, he ovat motivoituneempia työskentelemään yhdessä. Kun ihmiset työskentelevät yhdessä, voidaan ratkaista ongelmia, jotka käsittävät useampia kuin yhden osan yrityksen työntekijöistä. Työntekijä, joka tuntee, että häntä kunnioitetaan on aina valmiimpi tuomaan itsestään enemmän esille, jolloin saadaan piilossa olevia taitoja ja luovuutta paremmin esille. Jatkuva kehittäminen ja täydellisyyteen pyrkiminen lisää oman toiminnan kehittämistä, kun ajatellaan, että ei olla ikinä täydellisiä, niin ei tyydytä pienellä ajanjaksolla saatuihin tuloksiin, vaan pysytään edelleen kehityksessä mukana. (Bell & Bell & Orzen 2010 ,18-24.)



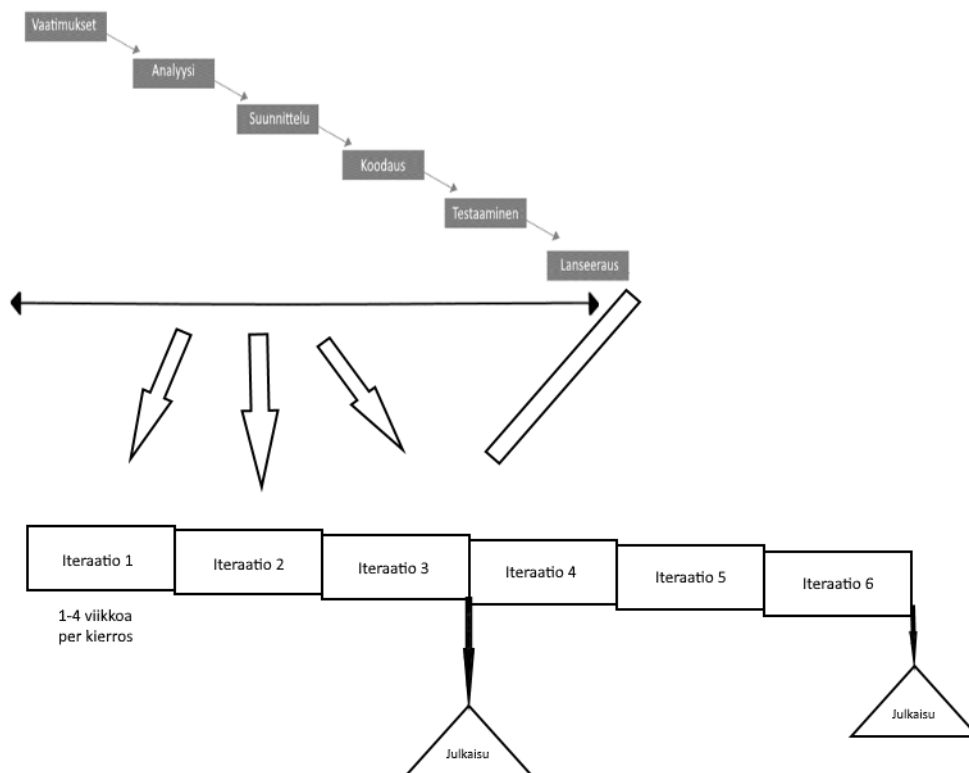


Kuvio 4: Lean pyramidi (mukaillen Bell & Bell & Orzen 2010, 18).

Seuraava kerros, edistävä käyttäytyminen rakentuu kolmen elementin päälle, edistävällä käyttäytymisellä tarkoitetaan sitä, kun keskitytään tulipalon sammuttamisen sijasta tulen aiheuttajaan. Edistävässä käyttäytymisessä kysytään paljon kysymyksiä, miksi jotakin tapahtuu (Bell & Bell & Orzen 2010, 24-26). Seuraavalla tasolla Lean-pyramidia on asiakkaan ääni, Laatu alusta alkaen ja Järjestelmällinen ajattelu. Kuulemalla asiakkaan ääntä kehittäjät keskittyvät oikeisiin ongelmiin ja tekevät ratkaisuja, joita nykyiset ja tulevat asiakkaat arvostavat. Ymmärtämällä asiakkaiden tarpeet ja halut saadaan vastuullisuutta, ketteryyttä ja kilpailullista etua kilpailijoihin nähden. Tuottamalla laatua alusta alkaen vähennetään uudelleen tekemistä, jolloin tiimillä jää aikaa tehdä asioita, jotka edistävät arvon tuomista asiakkaalle. Järjestelmällisessä ajattelussa ajatellaan ratkaisuja isossa kuvassa, mikä vaikuttaa koko yrityksen arvovirtaan (Bell & Bell & Orzen 2010, 26-28). Seuraavassa kerroksessa on Virtaus/Veto/Juuri ajoissa. Näillä tarkoitetaan sitä, että työskentely virtaa, ei ole turhia keskeytyksiä ja tehdään asioita oikea-aikaisesti, jolloin odottelu ja muu hukka poistuu työskentelystä (Bell & Bell & Orzen 2010, 29-30). Ylimpänä meillä on kulttuuri, joka koostuu perususkomuksista, jotka inspiroivat kunnioitusta, kehittävää käyttäytymistä, innovointia ja jatkuvaa

oppimista. Näillä päästään hyvälle alulle kohti kestävää toiminnan laadukkuutta. (Bell & Bell & Orzen 2010, 30-33)

Lean IT:n yksi osa on Lean ohjelmistokehitys, joka lähtee katsomaan kokonaisuutta asiakkaan näkökulmasta kehittäjän sijaan. Ketterät menetelmät ovat osana Lean-ohjelmistokehitystä, mutta Lean menee Ketteriä pidemmälle antaen laajemman näkökulman ja tätä kautta mahdollistaa Ketterien menetelmien menestyksen. Lean-ohjelmistokehityksessä keskitytään kehittämään koko arvovirtaa tilauksesta valmiiseen toimitettuun ohjelmistoon saakka. Lean-ohjelmistokehitys lähtee siitä, että ensin löydetään 20 % koodista, joka tuottaa 80 % tuotteen tai palvelun arvosta ja keskitytään sen toimittamiseen juuri ajoissa. Lean kehottaa tekemään vasta, kun asiakas vaatii jotain ja testaamaan jokaisessa iteraatiossa (kuvio 5). Tällöin saadaan ratkaisuita, jotka asiakas hyväksyy, tai antaa kehitysehdotuksia, jolloin edelleen kehittäminen on mahdollista jo, kun järjestelmä ei ole suuri kokonaisuus, eivätkä pienet korjaukset vaadi kokonaisen järjestelmän logiikan muuttamista. (Bell & Bell & Orzen 2010, 174-177.)



Kuvio 5: nopeiden iteraatioiden malli (mukaillen Bell & Bell & Orzen 2010, 176).

Lean-yritys voi olla minkä tahansa kokoinen, sekä projektikoolla ei ole väliä. Lean-kehitystiimi on hyvin samanlainen kuin esimerkiksi Scrum-tiimi. Siinä on tuotemanageri, sekä järjestelijä, joiden vastuut ovat käytännössä samat kuin scrumissa, eli manageri on vastuussa siitä,

mitä tehdään ja milloin tehdään. Järjestelijä järjestää kokouksia, kuten päivittäisiä palautekeskusteluja ja poistaa häiriötekijöitä, sekä opastaa kehittäjiä (Bell & Bell & Orzen 2010, 177-178). Lean ohjelmistoprojekteissa käytetään Scrumin lisäksi Kanban-tyylisiä tehtävien jako menetelmiä, kuten tauluja, joissa on tehtävät, jotka on järjestetty niille tarvittavien kehitysaikataulujen mukaisesti. Lean kuitenkin painottaa kokonaisvaltaista kehitystä arvovirralla ja tähtää virtauksen kehittämiseen joka osa-alueella, ei vain yhdessä osassa kehitystä. Kuten huomaamme, Lean kulkee hyvin käsi kädessä kahden muun valitun ohjelmistokehitysmenetelmän kanssa, antaen hieman laajempaa perspektiiviä hyödyntäen osia niistä, sekä lisäksi kehittämällä koko arvovirtaa tuotekehityksen kannalta. Tutkimuksen kannalta on oleellista kehittää koko prosessia, johon Lean tarjoaa hyviä ideoita.

#### 4 Tutkimusmenetelmät

Työn tutkimusotteena on kvalitatiivinen tutkimus, sillä kyseessä ei ole laajamittainen ilmiö, josta olisi saatavilla tilastoja ja näin ollen, kun teorioita, joilla selitettäisiin toiminnanohjausjärjestelmien valmistusta Prowledge Oy:ssä ei ole olemassa, on kyseessä laadullinen tutkimus. (Kananen 2017, 32.)

Työn aineistonkeruumenetelmänä toimivat teemahaastattelut, joissa haastattelen ilmiötä kokeneita työntekijöitä, asiakkaita ja toimihenkilöitä. Teemahaastattelut toteutetaan verkkoversioina, sillä opinnäytetyön teon aikana vallitseva Covid-19 pandemia estää henkilöiden haastattelun paikan päällä. ”Teemahaastatteluilla pyritään ymmärtämään ja saamaan käsitystä tutkimisen kohteena olevasta ilmiöstä” (Kananen 2017, 90)., tutkijana tavoitteenani haastattelujen osalta on saada mahdollisimman kattava ymmärrys ja käsitys siitä, miten toiminnanohjausjärjestelmän valmistus etenee koko prosessin lävitse, jotta voin tässä onnistua, on minun kuultava jokaisen mukana olleen osatekijän näkökulmasta, miten he ovat ilmiön kokeneet. Tilanteen autenttisuuden takaamiseksi en voi esittää valmiita kysymyksiä, sillä ne ohjaavat haastateltavia kertomaan ilmiöstä minun ennako-odotusteni pohjalta. Teemahaastatteluissa pääsen kuulemaan ilmiöstä, kuten haastateltava on sen kokenut, jolloin todellinen valmistusprosessi saadaan kuvattua mahdollisimman tarkasti. Haastattelut otetaan videotalenteina kokonaisuudessaan ylös analysointia varten. (Kananen 2017.)

Kerätty aineisto käsitellään sisältöanalyysillä, aluksi se litteroidaan, eli kirjataan ylös, jonka jälkeen siitä tehdään sisältöanalyysi, jossa pyritään kuvaamaan koko ilmiötä kokonaiskuvana. Sisältöanalyysi sopii menetelmäksi, sillä sen avulla saadaan kuvattua kokonaisprosessia (Saaranen-Kauppinen & Puusniekka 2006), jota on hyvä lähteä vertaamaan yleisesti käytettyihin menetelmiin rakentaessa pohjaa kehitysehdotuksille. Teemahaastateltavat valikoituvat sen pohjalta, ketkä kuuluvat toiminnanohjausjärjestelmien valmistusprosessiin mukaan. Tässä tutki-

muksessa keskitytään tutkimuksen jälkeisessä vaiheessa tulevien projektien pohjana käytettäviin järjestelmiin, joten niiden toteutuksessa mukana olleet henkilöt valikoituvat haastateltaviksi.

#### 4.1 Laadullinen tutkimus

Laadullinen tutkimus on empiirinen tutkimusmuoto, joka on saanut alkunsa Euroopan suurten tieteilsofilosofisten suuntausten pohjalta. (Tuomi & Sarajärvi 2018). Laadullisella tutkimuksella tarkoitetaan tutkimuksia, joissa pyritään löydöksiin ilman tilastoja tai muita määrällisiä keinoja (Kananen 2017). Lukemissani useammassa lähteessä pohdittiin syvällisesti sitä, mitä laadullinen tutkimus on, sekä mistä se on tullut ja mistä se on saanut vaikutteita. Positivismi mainitaan useammassa lähteessä yhtenä vaikuttajana 1800-luvulla. Kuitenkin laadulliseen tutkimukseen ei ole olemassa yhtä tarkkaa määritelmää, jota voitaisiin yleisesti käyttää. Laadullisen tutkimuksen piirteitä ovat toiminnallisuus, eli empiria, sanat ja lauseet lukujen ja yleistysten sijasta sekä yksittäisen tapauksen tutkiminen. Määrällinen tutkimus eroaa laadullisesta siten, että siinä käytetään lukuja ja tilastoja, joista tehdään yleistyksiä. Näitä yleistyksiä voivat olla esimerkiksi jonkin tapahtuman ennustaminen todennäköisyyden pohjalta. Vaikka määrällinen ja laadullinen voidaan erottaa toisistaan, niin niitä voidaan soveltaa samassa tutkimuksessa ja analysoinnissa. Tutkimuksen tarkoituksena on joka tapauksessa tuottaa perusteltua tietoa, joten nämä kaksi tutkimustapaa eivät sulje toisiaan pois. (Alasuutari 2011)

Valitsin tutkimusmenetelmäkseni kvalitatiivisen tutkimustavan, sillä tutkimukseni aihe on prosessi, jossa olevia ongelmia halutaan kehittää. ”Laadullisessa tutkimuksessa tutkitaan pääasiassa prosesseja, joihin määrällisessä tutkimuksessa tilastollisin analyysin on lähes mahdollonta pureutua prosessien ja ilmiöiden monimutkaisuuden vuoksi” (Kananen 2017, 36). Jotta kykenemme ymmärtämään ilmiötä, josta ei ole tarjolla määrällisiä arvoja, on tarpeen käyttää sanoja näkemysten keräämisessä tilastojen sijaan. ”Kvalitatiivisessa tutkimuksessa tutkittavien määrä ja tilastollinen argumentaatio ei ole tarpeellista tai mahdollista” (Alasuutari 2011). Tutkimuksen aihe ei siis mahdollista tilastojen keruuta, sillä prosessit ovat monimutkaisia kokonaisuuksia, joissa ei voida korvata yksittäisen ihmisen käsitystä ilmiöstä usean ihmisen mielipiteellä jostain ilmiön osasta.

#### 4.2 Teemahaastattelu

Teemahaastattelu, eli puolistrukturoitu haastattelumenetelmä. Pohjautuu Mertonin, Fiskin ja Kendalliin (1956) julkaisemaan kirjaan ”The Focused Interview”, joka on ilmestynyt uutena painoksena vuonna 1990. Kirjassa kuvataan menetelmää seuraavanlaisesti:

- Ensinnäkin tiedetään, että haastateltavat ovat kokeneet jonkin tietyn tilanteen
- Toiseksi tutkija on alustavasti selvitetty tutkittavan ilmiön oletettavasti tärkeitä osia, rakenteita, prosesseja, sekä kokonaisuutta. Tämän sisällön- tai tilanneanalyysin

avulla hän on päätenyt tiettyihin oletuksiin tilanteen määräävien piirteiden seurauksista siinä mukana olleille.

- Kolmanneksi hän analyysin avulla luo haastattelun rungon
- Neljänneksi ja viimeiseksi haastattelu suunnataan tutkittavien henkilöiden kokemuksiin tilanteista, jotka tutkija on ennalta analysoinut.

(Merton, Fiske & Kendall 1956, 3-4.)

Kuten nimi kertoo, niin teemahaastattelu ei kulje tiettyjen kysymysten mukaan, vaan etenee tiettyjen keskeisten aihepiirien varassa. Tämä pääosin poistaa tutkijan näkökulman ja tuo esille tutkittavien äänen. Teemahaastattelu ottaa huomioon ja tekee keskeiseksi ihmisten tulkinnat ja heidän asioille antamat merkitykset, sekä merkityksen syntymisen. Teemahaastattelu on lähempänä strukturoimatonta haastattelua, se eroaa muista puolistrukturoiduista haastatteluista, sillä siinä kysymykset ja niiden muoto eivät ole samanlaisia, teemahaastattelu ei kuitenkaan ole täysin vapaa, niin kuin syvähaastattelu (Hirsjärvi & Hurme 2008,47-48). Aineistonkeruun menetelmäksi valikoitui teemahaastattelu siitä syystä, että tutkimuksen tarkoituksena on kyetä ymmärtämään toiminnanohjausjärjestelmän valmistusprosessia ja siinä esiintyviä ongelmia. Näiden ongelmien syntyperään päästään parhaiten käsiksi kuvaamalla prosessia, sekä löytämällä sieltä pullonkauloja ja niiden syy ja seuraussuhteita. Paras tapa saada kokonaiskuva prosessista on haastatella sen osatekijöitä, jolloin saadaan kokonaisvaltainen kuvaus. Tämän kuvauksen pohjalta on helppoa lähteä analysoimaan, missä kohdissa kehitystä on ollut hidasteita. Teemahaastatteluissa saan oman mielipiteeni jäämään pois tutkimusaineistosta, joka lisää tutkimuksen luotettavuutta, kun tieto on kerätty suoraan lähteestä. Haastatteluissa, sekä niiden valmisteluissa tulee ottaa huomioon, ettei eräiden haastateltavien äidinkielenä ole suomi. Tästä johtuen on noudatettava varovaisuutta ja varautuneisuutta näistä lähteistä saatuihin tietoihin analyysivaiheessa. Syvähaastatteluiden käyttö tutkimuksessa olisi ollut toinen vaihtoehto. Kuitenkin kehittääksemme koko prosessia emme voi keskittyä vain tiettyjen henkilöiden näkemyksiin, vaan haluamme tietoa myyjältä, asiakkaalta, ohjelmistokehittäjältä, sekä työnjohdolta.

#### 4.3 Sisältöanalyysi

Laadullisessa tutkimuksessa voidaan analyysimenetelmät jakaa kahteen ryhmään, teoriaan perustuvat, joissa pyritään soveltamaan jotakin perusteoriaa lähdemateriaaliin, sekä vapaampiin, joissa voidaan soveltaa erilaisia teorioita. Sisältöanalyysi kuuluu vapaisiin menetelmiin. Laadullisessa tutkimuksessa analyysin perusrakenne on seuraavanlainen:

1. Päätä mikä tässä aineistossa kiinnostaa ja tee VAHVA PÄÄTÖS!
2.
  - a. Käy läpi kaikki aineisto, erota ja merkitse ne asiat, jotka sisältyvät kiinnostukseesi.

- b. Kaikki muu jää pois tästä tutkimuksesta!
  - c. Kerää merkityt asiat yhteen ja erilleen muusta aineistosta
3. Luokittele, teemoita tai tyyppitele aineisto
  4. Kirjoita yhteenveto  
(Tuomi 2018.)

Sisältöanalyysi on menetelmä, jota voidaan käyttää kaikissa laadullisen tutkimuksen perinteissä. Se on sekä yksittäinen metodi, että väljä teoreettinen kehys, joka on liitettävissä erilaisiin analyysikokonaisuuksiin. Sisältöanalyysillä tarkoitettaessa väljää kuullun, kirjoitetun tai nähdyn analyysin viitekehystä, voidaan kaikkien laadullisen tutkimuksen analyysimenetelmien sanoa perustuvan sisältöanalyysiin. Sisältöanalyysi sopii strukturoimattomalle tutkimusaineistolle, sillä se auttaa tiedon järjestelyssä, asettaen sen selkeään ja tiiviiseen muotoon, kadottamatta kuitenkaan sen sisältämää informaatioita. Aineisto voi olla kirjoitettua, puhetta tai haastattelu. (Tuomi 2018.)

Sisältöanalyysi sopii tutkimukseen, sillä sen avulla voidaan järjestää tietoa, teemahaastattelut antavat puolistrukturoitua haastatteluaineistoa, jonka analysoiminen järjestykseen ja tiiviiseen muotoon tuo esille prosessin eri vaiheet, sekä sisältää kustakin vaiheesta kerätyt tiedot. Tämän jälkeen voidaan alkaa tarkastelemaan, onko prosessissa jotakin vaiheita, joita ei yleisissä standardeissa ole, tai onko vaiheissa joitakin yleisillä standardeilla turhiksi tai ongelmallisiksi katsottuja osia.

#### 4.4 Reliabiliteetti ja validiteetti

Luotettavuuteen varaudutaan tutustumalla luotettavuutta käsittelevään aineistoon, sekä noudattamalla hyviä käytäntöjä tutkiessa, julkaisemalla lähteet oikein, sekä tehden tutkimus tutkimuksessa kerrottujen menetelmien ja tietojen puitteissa. Lisäksi perustellaan valitut menetelmät ja niiden oikeellisuus. Lisäksi pyritään täyttämään luotettavuuskriteereitä (Kananen 2017, 176.):

- Informantin vahvistus, eli luetetaan kerätty aineisto ja siitä tehty tulkinta haastattelutavalla, jotta varmistetaan saadun käsityksen oikeellisuus.
- Vahvistettavuus, eli pyritään takaamaan saatua tietoa aikaisemmalla tiedolla, tai muista lähteistä saadulla tiedolla.
- Arvioitavuus, eli tehdään riittävä dokumentaatio kaikesta, jota on tutkimuksen aikana tehty, jotta tulokset ovat jäljitettävissä, säilytetään kerätty tieto oikein, jotta se on saatavilla ja aitous voidaan todentaa.
- Tulkinnan ristiriidattomuus, eli varmistamalla toisen tutkijan avulla, että hän pääsee myös samaan lopputulokseen.

- Saturaatio, eli kerätään niin kauan tietoa, että eri lähteiden tarjoamat tulokset alkavat toistua, jolla varmistetaan se, että tutkimustietoa on kerätty tarpeeksi.
- Aikaisemmat tutkimukset, eli vahvistetaan omia tutkimustuloksia aiemmin tehdyillä tutkimuksilla, joissa on päädytty samankaltaisiin ratkaisuihin.
- Kaiken tämän lisäksi tehdään lopussa myös luotettavuustarkastelu, jossa pohditaan kunkin osion osalta, kuinka hyvin jokaista mittaria on onnistuttu noudattamaan.

Reliabiliteettia ja validiteettia käytetään mittareina eri vaiheessa työskentelyä, niillä pyritään varmistamaan tutkimuksen luotettavuutta ja laatua. Validiteetin merkitys koostuu työn suunnittelu vaiheessa, sillä sen avulla mitataan onko tutkittu oikeita asioita (Kananen 2017, 175). Validiteetin todentamiseksi perustellaan valitut menetelmät suhteessa tutkittavaan asiaan, jotta saadaan selvitettyä, miksi tutkijan mielestä asiaa kannattaa tutkia tällä tavalla. Reliabiliteetti otetaan huomioon työn tekovaiheessa, jossa dokumentoidaan tutkijan työskentelyä riittävästi, jotta voidaan todentaa, että tutkija on tehnyt tutkimusta oikealla tavalla. Täten voidaan varmistua siitä, että mikäli joku tutkii asiaa uudelleen samoilla menetelmillä, on lopputulos sama tai samankaltainen (Kananen 2017, 175).

## 5 Tiedonkeruu ja analyysi

Tiedonkeruu vaiheessa järjestettiin teemahaastattelut viikolla 19, haastatteluissa haastateltiin yhtä asiakasyrityksen työntekijää, kahta ohjelmistokehittäjää, Prowledgen toimitusjohtajaa, sekä asiakasyrityksen toimitusjohtajaa, joka työskentelee myyntihenkilönä toiminnanohjausjärjestelmien myyntipuolella. Tutkija kirjoitti haastattelut puhtaaksi analyysivaihetta varten, joka toteutettiin sisältöanalyysinä. Analyyseista tehtyjen yhteenvetojen pohjalta tutkija tuotti tyypillisen toiminnanohjausjärjestelmän kehitysprojektin, jossa on mukana eri roolit, sekä vaiheet ja kehittämistä vaativat asiat.

Haastatteluihin valitut henkilöt valittiin heidän roolinsa perusteella suhteessa kehitettyihin toiminnanohjausjärjestelmiin. Esimerkiksi ohjelmistokehittäjiä on asiakasyrityksessä enemmän kuin kaksi, mutta nämä kehittäjät valittiin, sillä toinen heistä on ollut pääkehittäjänä tämänlaisissa ohjelmistoissa alusta asti. Toinen kehittäjä on uusin kehittäjä, joka on yritykseen tullut, joten häneltä pyrittiin saamaan näkökulmaa sopeutumisesta ja asioista, jotka saattavat tuntua pidempään mukana olleelle kehittäjälle tavallisilta, mutta nuoremmasta hieman kummallisilta. Toimitusjohtajan ja myyjän tai asiakkaan haastattelut pohjautuvat siihen, että he ovat ainoat roolinsa edustajat tämänkaltaisessa prosessissa. Asentaja valittiin vapaaehtoisuuteen pohjautuen.

## 5.1 Haastattelut

Haastattelujen pohjana toimi teemahaastattelu-tiedosto (liite 1), joka tehtiin varmuuden vuoksi kahdelle kielelle, jotta Ruotsia äidinkielenään puhuvat ohjelmistokehittäjät voivat halutessaan osallistua haastatteluun englanniksi. Englanti valikoitui toiseksi kieleksi tutkijan ja ohjelmistokehittäjien aikaisemman kommunikaation pohjalta. Haastatteluista kaksi pystyttiin järjestämään kasvotusten ja kolme niistä pidettiin Skype-videopuheluna mahdollisimman aidon haastattelutilanteen luomiseksi. Haastattelut sujuivat pääasiassa mutkattomasti, tekniikka toimi kohtalaisen hyvin, ainoana ongelmana tutkijalla oli alussa verkossa pidettävissä haastatteluissa haastateltavan äänen tallentaminen.

Haastatteluja pitäessä tutkija törmäsi omaan kokemattomuuteensa haastattelijana ja joutui välillä pitämään pieniä mietintätaukoja, jotta saisi haastattelun pidettyä teeman sisällä. Tämän lisäksi tutkija huomasi, että ei osannut kohdistaa kaikkia kysymyksiä kyllin tarkasti, joten välillä haastattelut saattoivat lähteä pois kohdealueelta. Haastatteluista saatiin kuitenkin tietoa toiminnanohjausjärjestelmien valmistusprosessista. Tutkijan olisi syytä seuraavalla kerralla miettiä haastattelujärjestystä, sekä valmistautua paremmin.

Ensimmäisessä haastattelussa keskityttiin siihen, miten toiminnanohjausjärjestelmän valmistus ja käyttöönotto näkyi asentajan, eli järjestelmää käyttävän työntekijän näkökulmasta. Toisessa haastattelussa haastateltiin pääkehittäjää, joka kertoi alkuperäisen järjestelmän kehittämisestä, sekä uudempien järjestelmien kehittämisestä. Kolmannessa haastattelussa haastateltiin nuorempaa kehittäjää, joka oli tullut taloon, kun monia järjestelmiä oli jo tehty. Häneltä kerättiin tietoa eri tehtävistä, joita hänellä on ollut, sekä alussa tapahtuneesta tutustumisesta järjestelmään ja sen kehittämiseen osallistumisesta. Neljännessä haastattelussa haastateltiin Prowledge Oy:n toimitusjohtajaa, joka on toiminut myyntitehtävissä, sekä yhteishenkilönä asiakkaan ja kehittäjien välillä. Häneltä kerättiin näkemystä projektinhallinnasta ja yhteistyöstä asiakkaiden kanssa. Viidennessä ja viimeisessä haastattelussa haastateltiin asiakasyrityksen toimitusjohtajaa, joka on toiminut pohjajärjestelmän projektinjohtajana, sekä myyntihenkilönä uusien järjestelmien kanssa. Häneltä kerättiin kokemuksia asiakkaana, sekä näkemystä projektinjohdosta ja myyntityöstä, ensikosketuksesta asiakkaaseen, sekä yhteistyöstä varsinaisten ohjelmistokehittäjien kanssa.

## 5.2 Haastatteluiden analyysi

Haastattelujen analyysi perustuu sisältöanalyysin rakenteeseen. Tutkimuksessa päätettiin keskittyä toiminnanohjausjärjestelmien valmistusprosessin kuvaamiseen saadun tiedon pohjalta. Tutkija päätti, että tähän kuvaamiseen tarvittavat tiedot koskevat nykyisen prosessin vaiheita, kommunikaatiota sekä henkilöstöä käsittelevät ja kuvaavat tiedot, kaikki muu jäi aineistossa tässä vaiheessa pois.



Ensimmäisen haastattelun analyysi on aika lyhyt, sillä asentajalla ei ollut kovin suurta roolia järjestelmän valmistuksessa. Yhteenvedon sisältöä ei tarkastettu hänellä, sillä muiden haastateltavien antamat tiedot täsmentävät niitä. Häneltä saadut tiedot pohjautuvat lähinnä järjestelmän käyttöönottoon. Yhteenvetona valmistusprosessi asentajan näkökulmasta:

”Firman johto tiedotti työntekijöille, että firma on siirtymässä sähköiseen järjestelmään. Tämän jälkeen firman johto tiedotti työntekijöille, mitä kehitystyössä tapahtui. Firman johto esitteli työntekijöille järjestelmän ja opasti sen käytössä, käyttöönotto onnistui työntekijöillä kertasiirtymällä. Järjestelmässä on joillakin työntekijöillä ollut ongelmia, laitteistoista tai jostakin muusta johtuen.” (Asentaja 2020.)

Toisen haastattelun haastateltava oli ohjelmistokehittäjä, joka on ollut talossa muutaman vuoden ajan, hän on tullut uutena mukaan kehittämiseen, ja työpaikka oli hänelle alan ensimmäinen. Hän kokee sopeutuneensa järjestelmän kehittämiseen hyvin, ja hän on ollut mukana useammassa toiminnanohjausjärjestelmän kehityksessä. Hänelle lähetettiin yhteenvedo tarkennuksia varten, jotta väärinymmärryksiltä vältytään. Yhteenvetona hän kuvaa kehityksen vaiheita ja kommunikaatiota asiakkaan kanssa seuraavanlaisesti:

”Projektin pohjana käytetään työlistausta, joka on tehty aloituspalaverissa, alussa tarvitaan asiakkaan apua, että tiedetään, mitä tarkalleen halutaan. Työvaiheen valmistuttua ilmoitetaan asiakkaalle, että voisitteko testaila sitä ja sitä mukaa, kun asiakkaalla tulee mieleen kehitysehdotuksia tai muita, niin niitä saa laittaa projektihallintatyökaluun, tämän avulla varmistetaan, että asiakas ja ohjelmistokehittäjä ovat ymmärtäneet asian samalla tavalla. Kehitys jatkuu samalla tavalla myös ensimmäisen käyttöönoton jälkeen. Käyttöönoton kesto riippuu pohjajärjestelmään tehtävistä muutoksista. Kommunikaatio kehittäjien ja asiakkaan välillä on yleensä toiminut, tosin projektihallintatyökalusta siirrytään usein puhelin ja sähköpostiviestintään, vaikka kehittäjät pyrkivät sen pitämään projektihallintatyökalussa. Eräissä projekteissa asiakas oli muuttanut vaatimuksia työvaiheelle lennosta, jolloin kehittäjät olivat saaneet häneltä palautetta, kyseessä ei ollut uutta toiminnallisuutta, vaan sisältöpäivitys. Kommunikaatio keskittyy yleensä projektin alkuun, joskus asiakasta ei saa alun jälkeen kiinni lainkaan, joskus tapahtuu paljon kommunikaatiota kerralla, sitten voi olla taas taukoa, eli kommunikaatio elää jatkuvasti. Yleensä asiakkaat ovat kuitenkin olleet ymmärtäväisiä ja kommunikaatio on toiminut.” (Nuorempi ohjelmistokehittäjä 2020.)

Kolmannessa haastattelussa haastateltavana oli vanhempi ohjelmistokehittäjä, joka on ollut kehittämässä ensimmäistä, muiden järjestelmien pohjana ollutta järjestelmää. Hän on siis nähnyt järjestelmän kehitystä käytännössä jokaisen uuden pohjan mukaan tehdyn projektin osalta, joten hänellä on kattavat tiedot siitä, miten projekteja toteutetaan, sekä miten niiden kehitys on muuttunut matkan varrella. Hän luki myös yhteenvedon ja täsmensi siinä olevat tiedot oikeiksi. Yhteenvetona hän kuvaa kehitystä ja kommunikaatiota seuraavanlaisesti:

”Kun Prowledge kehittää toiminnanohjausjärjestelmää, se kehittää sitä asiakkaalle. Tästä johtuen järjestelmä on asiakkaan, joka päättää mitä järjestelmään tulee. Top-järjestelmien ensimmäinen asiakas on siirtynyt järjestelmien myyntityöhön, mutta projektinjohtajana on edelleen uusi asiakas. Asiakkaalla on käytössään testiversio, jota voi katsoa ja kokeilla. Kun käyttöönotto on tehty, niin erilliselle testiversiolle ei ole tarvetta, muutokset laitetaan suoraan tuotantoon ja jos ongelmia havaitaan, niin niistä tehdään korjaus. Yhden järjestelmän kehityksessä oli joka toinen viikko palaveri asiakkaan kanssa, mikä toimi hyvin tiedottamisen ja projektinseurannan kannalta. Projektinhallintatyökaluna toimii Trello, jota asiakkaat käyttävät eritasoisesti, kehittäjät ja asiakkaat käyttävät muitakin työkaluja, kuin Trelloa tehtävien siirtämiseksi. Tieto kulkee myös välillä yhteyshenkilön, Prowledge Oy:n toimitusjohtajan kautta, sillä jotkut asiakkaat ovat fyysisesti lähempänä Etelä-Suomen toimipistettä.” (Vanhempia ohjelmistokehittäjiä 2020.)

Neljännessä haastattelussa haastateltiin Prowledge Oy:n toimitusjohtajaa, joka on toiminut tulkkina asiakkaan ja ohjelmistokehittäjien välillä. Hän on eräänlainen yhteyshenkilö, joka osallistuu projektien suunnitteluun, sekä hallintaan. Hän on ollut kehittämisessä mukana alusta saakka, joten hän on nähnyt useampia projekteja. Hän tutustui myös yhteenvetoon ja varmensi siinä esitetyt asiat oikeiksi. Hän kuvaa projektin kulkua ja kokemuksia eri projekteista seuraavalla tavalla:

” Alussa firman yhteyshenkilö tekee asiakkaan kanssa yhteistyössä työstä, jossa sovitaan, mitä asioita järjestelmään tulee. Tämän jälkeen aloitetaan näiden työvaiheiden teko, sekä laitetaan asiakkaalle testiversio, jossa hän voi käydä testaamassa järjestelmän osia. Testausvaiheesta siirrytään käyttöönottoon, jossa yhteistyökumppani käy esittelemässä järjestelmän toiminnan asiakkaalle. Asiakkaalle voi olla vaikea nähdä järjestelmän kehittämisestä koituvia kustannuksia. Kriittisten toimintojen, kuten palkanlaskun siirtäminen järjestelmään vaatii luottamusta asiakkaan ja kehittäjien välillä. Asioiden dokumentoiminen paperille, ennen kun aletaan tekemään on kannattavaa ja se toteutetaan projektinhallintatyökalun avulla, joka esitellään asiakkaalle aloituskokouksessa. Tyypillisin yhteydenotto menetelmä on soitto, jonka seurauksena tehdään projektinhallintatyökaluun kortti, jonka asiakas ja koodari käyvät katsomassa ja johon koodari laittaa työaika-arvion ja jonka asiakas hyväksyy, asiakkaat harvemmin itse laittavat työkaluun kortteja. Asiakas saattaa soittaa suoraan koodarille, tai toimitusjohtajalle, tai laittaa vaikka WhatsApp ryhmään viestin. Ennen käyttöönottoa saattaa joutua asiakasta hätistelemään testaamiseen, toivottavaa on, että asiakas tekee testaamisen kunnolla, että järjestelmä on testattu, kun se otetaan käyttöön. Muutamien asiakkaiden kohdalla on käynyt niin, että palaute ja jatkokehitys on sovittu projektinhallintatyökalussa asiakkaan ja kehittäjien välillä. Asiakkaiden on vaikeaa nähdä järjestelmän suunnittelun arvoa, sillä järjestelmä on heidän ensimmäinen ja toivottavasti viimeinen toiminnanohjausjärjestelmän kehitysprojekti. Yhteistyöongelmat johtuvat siitä, että asiakkaan yhteyshenkilö ei ole

kyllin kiinnostunut järjestelmästä, jolloin saattaa olla kannattavampaa jättää järjestelmä tekemättä.” (Toimitusjohtaja 2020.)

Viimeisessä haastattelussa haastateltiin henkilöä, joka toimi alkuperäisen järjestelmän asiakkaana, sekä projektinjohtajana ja työskentelee nykyisin myös myyntipuolella. Häneltä saatiin hyvää näkemystä sekä asiakkaana olemisesta, että uudempien asiakkaiden kanssa tehdystä yhteistyöstä, sekä lähtökohdista, joihin järjestelmää aletaan kehittämään. Hänelle lähetettiin yhteenveto tarkastusta varten, mutta hän ei kommentoinut sitä millään tavalla. Yhteenvetona hän kertoi järjestelmän kehityksestä asiakkaana seuraavanlaisesti:

” Järjestelmän kehityksen alkuvaiheessa tehtiin toiminnan kuvaus, joka muuttui matkan varrella, mikä on täysin normaalia. Asiakkaana vaadittiin paljon työtä, että järjestelmästä tuli sellainen, minkälaisen asiakas halusi, asiakas kertoi aina, miten jonkin pitää toimia ja miltä sen pitää näyttää, koodarin ja asiakkaan välillä toimi yhteyshenkilö ikään kuin tulkkina, jotta kommunikaatio on sujunut. Tämän tyyppinen järjestelmä vaatii asiakkaalta aktiivisuutta, jotta sen valmistaminen on ylipäättänsä mahdollista, sillä kehitettävät asiat ovat hyvin abstrakteja suunnittelu vaiheessa ja tarvittavista asioista on tiedot asiakkaalla, mutta asiakas ei tiedä, mitä on mahdollista tehdä ja mitä minkäkin asian teko maksaa. Asiakkaan suuntaan pitäisi olla heti alusta asti aktiivisempi, jotta asiakas pääsisi koko projektin ajan testaamaan järjestelmää ja voisi antaa palautetta koko kehityksen ajan. Tällä hetkellä koodarit tekevät pätkän töitä ja ilmoittavat asiakkaalle, että nyt kaikki on valmista, käy testaamassa järjestelmä, jotta voidaan aloittaa käyttöönotto, asiakas ei välttämättä osaa testata kunnolla, jolloin hänellä tulee kiire, kun pitäisi osata käyttää järjestelmää, heti alusta asti, vaikka sen käyttö vaatii oppimista.” (Asiakas 2020.)

Myyjän näkökulmasta yhteenvetona:

” Aluksi myyntihenkilö käy tutustumassa asiakkaan liiketoimintaan, jotta saadaan selville, mihin asiakkaan bisnes perustuu, mitä eri osa-alueita siinä on ja minkälaiseen ongelmaan järjestelmä tarjoaa ratkaisua. Tämän jälkeen tehdään demo tai näytetään valmista järjestelmää, miten se toimii, tärkeätä on selvittää mitä asiakas tarvitsee ja osoittaa, minkälaista lisäarvoa järjestelmä tarjoaa ja tätä kautta auttaa asiakasta päättämään, ottaako hän järjestelmän itselleen vai ei. Asiakas ja myyjä voivat keskenään sopia, mitä osia asiakas tarvitsee järjestelmään. Asiakkaan ja koodarin kommunikaatioon liittyy jokapäiväistä ongelmaa, sillä asiakkaan täytyy opetella käyttämään projektinhallintatyökalua, jotta hän voi ohjata ja seurata projektin kulkua reaaliaikaisesti, asiakas ei halua opetella käyttämään Trelloa, vaan haluaa järjestelmän valmiiksi, koodari haluaa tehdä järjestelmän, mutta tarvitsee asiakkaalta tietoa. Ylipäättänsä kommunikaatiossa tarvitaan enemmän avoimuutta, ohjelmistokehittäjien täytyy kyetä antamaan hinta-arvioita lisätöistä ja sopimaan asioita asiakkaan kanssa tehokkaammin.” (Myyjä 2020.)

Haastattelujen pohjalta voidaan useamman henkilön varmentamana tehdä tyypillinen projekti, jossa rooleja on:

1. Myyjä, joka tutustuu asiakkaan liiketoimintaan ja selvittää asiakkaan kanssa, mitä ongelmia järjestelmän täytyy ratkaista.
2. Yhteyshenkilö, joka tekee asiakkaan kanssa tarkemmin työtä, jotta asiakas oppii ymmärtämään järjestelmää
3. Ohjelmistokehittäjä, joka kehittää järjestelmää, toimii välillä yhteyshenkilönä, sekä osana suunnittelua
4. Asiakas, joka toimii projektin projektipäällikkönä, tietää mitä järjestelmään tulee, hoitaa testauksen ja hyväksyy työjäljen

Projektin vaiheita on:

1. Myyntitapahtuma, jossa myyjä ja asiakas selvittävät, miksi järjestelmää tarvitaan
2. Tarvelistaus, jossa yhteyshenkilö ja asiakas selvittävät järjestelmän toimintaa
3. Kehitysvaihe, jossa kehittäjät lähtevät tekemään tarvikelistauksen tuotteita, asiakkaalla on testiversio, jossa kehitettyjä osioita voi testata jo tässä vaiheessa
4. Testausvaihe, jossa asiakas testaa tuotteen eri osiot ja antaa niistä palautetta
5. Käyttöönotto, jossa järjestelmän kriittiset osat ovat valmiina asiakkaan käyttöön ja jossa yhteyshenkilö käy esittelemässä järjestelmän toiminnan asiakasyrityksessä
6. Jatkokehitysvaihe, jossa asiakas yrityksen kasvaessa, tai erinäköisten tarpeiden muututtua antaa lisätöinä kehittäjille erilaisia lisäkehitystehtäviä

Asioita, joissa on kehitettävää:

1. Pysyvä ja kyllin avoin kommunikaatio, asiakkaan ja kehittäjän välillä
2. Asiakkaan kokemattomuus työkalujen ja projektin osien välillä ja tästä johtuen kommunikoinnin siirtyminen tai puutuminen projektinhallintatyökalusta, josta johtuen sovittujen asioiden dokumentointi puuttuu.
3. Testausvaiheen sujuvuus ja kiireellisyys asiakkaan näkökulmasta
4. Asiakkaan ymmärtämättömyys, josta johtuen järjestelmän teko on saattanut päättyä kokonaan

## 6 Johtopäätökset

Yleisellä tasolla nykyinen kehitysmenetelmä eroaa tyypillisestä ohjelmistokehityksen kehitysmenetelmästä ainakin roolien osalta siten, että asiakas on tässä tapauksessa projektinjohtaja, eikä Prowledge Oy:ssä ole projektinjohtajaa. Yrityksen toimitusjohtaja toimii eräänlaisena

projektinjohtajana, sekä yhteyshenkilönä kehittäjän ja asiakkaan välillä, mutta yrityksen sisällä ei ole nimetty projekteille varsinaista projektinjohtajaa, vaan tämä tehtävä annetaan asiakkaalle. Tämä perustellaan yrityksessä sillä, että järjestelmä tehdään asiakkaalle, joten hänellä täytyy olla valta päättää siitä, mitä sinne tulee. Projektit vaativat onnistuakseen vahvan asiakkaan, joka on valmis perehtymään, sekä johtamaan kehitystä, erään tuotteen jatkokehitys oli lopetettu, kun tämä asiakas jäi pois virastaan.

Yrityksen kehitysmenetelmää ei voida kohtuudella väittää vesiputousmalliseksi, sillä siinä on testausta, jota toteutetaan kehityksen aikana. Tämän lisäksi jatkokehitys toteutetaan myös ketteriä menetelmiä mukaillen, jolloin asiakkaan ääni on kuuluvissa ja kehitetyt osat menevät testaukseen. Kehitysmenetelmä on lähellä ketteriä menetelmiä ja siitä löytyykin monia niiden osia.

### 6.1 Ohjelmistokehitysmenetelmät ja niiden hyödyt

Tutkimuksessa selvitettiin, mitä ovat nykyiset kehitysmenetelmät, sekä mitä etuja niistä on, Teoriaosassa käytiin lävitse, että kehitysmenetelmiä on vanha vesiputousmalli, sekä ketterät menetelmät. Ketteriksi menetelmiksi valittiin teoriaosaan Scrum, Kanban ja Lean, joka ei välttämättä lukeudu pelkästään ketteräksi menetelmäksi, mutta ajaa samoja asioita. Menetelmiä on vielä lisää, mutta tutkimuksen resursseista johtuen rajaamme menetelmät näihin kolmeen. Nämä kolme tuovat jokainen omaa näkökulmaansa siten, että Kanban on hyvin vapaa menetelmä, joka sisältää samoja arvioita, kuin muut, mutta vaatii toteuttajaltaan paljon vähemmän. Scrum taas keskittyy hyvin intensiivisesti aktiivikehitysvaiheen kehittämiseen. Scrum vaatii kehittäjiltä hieman enemmän, sen roolitukset ja toimipisteoptimiit eivät vastaa asiakasyritystä, mutta se antaa näkökulmaa tiivistä kehitysyhteistyöstä. Lean valikoitui kolmanneksi, sillä se antaa laajemman näkökulman koko prosessiin. Itse ohjelmistokehityksen kannalta Lean ja Scrum ovat hyvin samankaltaisia, mutta tutkimuksessa halutaan kehittää koko prosessia, ei vain ohjelmistokehitystä.

Valittujen kehitysmenetelmien hyödyksi on löydetty häiriöiden ja ongelmien esiin nousuista, työntekijöiden käytäntöjen esiin nousuista. Arvon tuottamista asiakkaalle alusta saakka. Priorisoinnin ansiosta arvokkaimman osien työstäminen, sekä priorisointi kehitysehdotusten ansiosta. Ketteryys, eli voidaan reagoida muutokseen, mikäli sille on tarvetta. Hukan poisto, eli kun löydetään turhia asioita, niin niitä voidaan poistaa.

### 6.2 Tilajayrityksen kehitysmenetelmä

Tutkimuksessa tutkittiin, miten tilajayrityksessä valmistetaan ohjelmistoja, tämä toteutettiin haastattelujen avulla, joiden pohjalta luotiin tyypillinen projektin rakenne, joka etenee tarpeiden selvittämisestä myyntiin, jonka jälkeen tehdään toimintasuunnitelma, tuotelistaus, jossa selvitetään yksityiskohtaisesti, miten mikäkin järjestelmän osio tulee toimimaan. Tämän

jälkeen alkaa intensiivikehitysvaihe, jonka aikana voidaan ja saatetaan toteuttaa testaamista. Intensiivikehityksen jälkeen, jos asioita ei ole testattu, niin asiakkaalle ilmoitetaan testaamisen tarpeesta. Testaamisen jälkeen tehdään jatkokehitystä, jonka jälkeen suoritetaan käyttöönotto, jolloin yhteistyökumppani käy esittelemässä järjestelmän toimintaa asiakasyrityksessä. Käyttöönoton jälkeen tehdään jatkokehitystä asiakkaan tarpeiden mukaisesti. Tilajayrityksessä on myös tehty asiakkaan kanssa kokoustamista yhdessä projektissa, jossa aina joka toinen viikko käytiin yhdessä läpi, mitä on tehty ja saatiin keskustelua tehdyistä asioista.

### 6.3 Kehitysehdotukset

Tutkimuksessa tutkittiin myös, mitä Prowledge oy:ssä voidaan tehdä, jotta saadaan tehostetua toiminnanohjausjärjestelmien valmistusprosessia. Nykyisessä toiminnassa esille nousseita kehityskohtia oli:

1. Pysyvä ja kyllin avoin kommunikaatio, asiakkaan ja kehittäjän välillä
2. Asiakkaan kokemattomuus työkalujen ja projektin osien välillä ja tästä johtuen kommunikoinnin siirtyminen tai puuttuminen projektinhallintatyökalusta, josta johtuen sovitujen asioiden dokumentointi puuttuu.
3. Testausvaiheen sujuvuus ja kiireellisyys asiakkaan näkökulmasta
4. Asiakkaan ymmärtämättömyys, josta johtuen järjestelmän teko on saattanut päättyä kokonaan.

Pysyvän ja kyllin avoimen kommunikaation luomiseksi asiakkaan ja kehittäjän välillä on ketterissä menetelmissä käytetty juoksuja tai iteraatiokierroksia, joiden yhteydessä pidetään eräänlaisia kokouksia, joissa käydään lävitse kehitettyjä osia. Tämänkaltaisia kokouksia pidetään säännöllisesti, kuten Scrumissa n. kahden viikon kehitysjakson jälkeen, jolloin kommunikaatio on pysyvää, avoimuutta lisää kommunikaatiossa se, että kokoontaan yhteen, jolloin kaikkien mukana olevien osapuolien täytyy keskittyä kokouksessa käsiteltäviin asioihin.

Asiakkaan kokemattomuutta projektinhallintatyökalujen ja projektin osien välillä ja tästä johtuvaa kommunikaation siirtymistä on ketterissä menetelmissä ratkaistu yksinkertaisemmillä työkaluilla, kuten kynä ja paperi, sekä siirtämällä projektinhallintaa esimerkiksi tuotteenomistajalle Scrumissa, sekä Lean-IT:ssä. Tilajayritys kuitenkin tarvitsee verkkotyökalun, sillä kehittäjät ja asiakkaat ovat eri paikoissa, joten jotta kaikilla on sama näkymä projektista, täytyy projektinhallinnan olla verkossa. Mikäli asiakas kuitenkin tarvitsee puhelimitse käytävää keskustelua, täytyy kehittäjien kyetä tarjoamaan siihen mahdollisuus. Tämän jälkeen on kuitenkin huolehdittava, että sovitut asiat tulevat projektinhallintatyökaluun, jotta oikeanlainen dokumentaatio säilyy. Scrumissa ja Lean-IT:ssä projektinjohtaja käyttää tätä työkalua ja on yhteydessä asiakkaaseen, joka mahdollistaa kehittäjien keskittymistä kehittämiseen ja asiakkaan keskittymistä muihin asioihin, kuin työkalun käytön oppimiseen. Tilajayritys haluaa

kuitenkin opettaa asiakkaan käyttämään työkaluja jatkokehitystä varten, joka tapahtuu täysin asiakkaan tarpeiden mukaan.

Testausvaiheen sujuvuus ja kiireellisyys asiakkaan näkökulmasta johtuu siitä, että asiakas ei osaa käyttää järjestelmää, joten hän ei osaa sitä myöskään testata. Kehittäjät tarvitsevat asiakkaan testaamista, jotta käyttöönotto saadaan toteutettua sujuvasti ja intensiivikehitysvaihe saadaan siirrettyä pois kehittäjien hartioilta. Asiakas joutuu oppimaan järjestelmän käytön joka tapauksessa, mutta ei välttämättä kerkeä, tai viitsi testata keskeneräistä järjestelmää tai opetella sen käyttöä. Kuten ensimmäisessäkin kohdassa, niin tässäkin palataan siihen, että kun kehittäjät ja asiakkaat kokoutuvat yhdessä tutkimaan, mitä kehitysajaksella on kehitetty, saadaan testaaminen tulemaan osaksi kehittämistä jatkuvassa muodossa, eikä asiakkaan tarvitse itse testata tai opetella kaikkea, vaan kehittäjät voivat kokouksessa demonstroida ja selittää asioita asiakkaalle, joka voi kysyä tarkennuksia epäselvyyksiin.

Asiakkaan ymmärtämättömyyttä ja siitä johtuneita järjestelmien valmistuksen peruuntumisia on vaikeata ratkaista. Mikäli asiakas ja järjestelmä eivät kohtaa, on muutettava jompaakumpaa, joko järjestelmää tai asiakasta. On olemassa toiminnanohjausjärjestelmiä, jotka sopivat varmasti paremmin tämänkaltaisille asiakkaille, sillä ne eivät vaadi alussa asiakkaalta mitään. Kuitenkin, kun emme tutkimuksessa voi muuttaa asiakasta tai järjestelmää, niin muutamme koko menetelmän rakennetta siten, että asiakkaalla ei tarvitse olla ymmärrystä koko järjestelmästä ja hän voi vain opetella käyttämään valmista järjestelmää. Tämänkaltaiset projektit sopivat hieman paremmin Scrumin kaltaiseen kehitykseen, jossa asiakas ei käytännössä tee muuta, kun arvioi ja kehitystiimi kehittää koko järjestelmän alusta loppuun saakka. Tässä tapauksessa tarvitaan nykyiseen malliin vahva projektinjohtaja, joka tekee yhteistyötä asiakkaan kanssa koko projektin läpi ja korvaa vahvat asiakkaat, joita järjestelmien valmistus on ennen vaatinut.

Erilaisten ongelmien ratkaisussa huomaamme, että Scrum ja Lean tarjoavat ratkaisuja moniin esille nousseisiin ongelmiin. Tutkimusryityksessä on käytössä Kanbanin kaltaista projektinhallintaa, mutta siinä nousee esille ongelmia, kuten asioiden siirtymistä projektinhallinnan ja dokumentaation ulkopuolelle. Iteraatiot ja juoksut auttaisivat pitämään asiat kaikkien näkyvillä, kunhan muistetaan pitää dokumentaatiota esille nousseista asioista. Lisäämällä roolin kehittämiseen, jonka vastuuna on hallita projektia, eli pitää huolta kommunikaatiosta, varmistaa, että asiat toteutuvat sovitulla tavalla, varmistaa, että asiakas ymmärtää, mitä häneltä vaaditaan ja auttaa asiakasta, mikäli hänelle tulee vaikeuksia, sekä kouluttaa asiakkaalle työkalut jatkokehitystä varten saadaan tilaajaryityksessä turvaa ja rauhaa siitä, että intensiivikehitys, joka toteutetaan ennen ensimmäistä käyttöönottoa toteutuu heidän aikataulullaan ja heidän ja asiakkaan sopimien resurssien sisällä, sekä sovitut asiat dokumentoituna. Asiakas, jolla on yritys hoidettavanaan saa osallistua kehittämiseen niin paljon kuin tahtoo, mutta häneltä vaadittavat asiat eivät ole niin raskaita, että hän ei ehtisi niitä hoitaa, mikäli näin kuitenkin on,

voi hän sopia, että projektijohtaja hoitaa nämä asiat tarvittaessa hänen puolestaan. Jatkokehitys on kuitenkin mahdollista toteuttaa asiakkaan ehdolla, kunhan varmistutaan intensiivikehityksen aikana siitä, että asiakas on kykenevä johtamaan oman järjestelmänsä kehitystä.

## 7 Pohdinta ja jatkokehitysehdotukset

Tutkimuksessa nousi esille asioita, jotka tukivat teoriapohjan käsitystä alan eri yritysten toiminnassa. Haastatteluissa huomataan, miten yritys on aloittanut kehittämään järjestelmiä yritys ja erehdys periaatteella, jossa on aina opittu vanhasta ja pyritty kehittämään toimintaa aina ongelmien tullessa kohdalle. Kuitenkin, kuten teoriapohjassa tuli esille, tämän kaltainen tulipalojen sammutus saattaa ajaa kehitystä kauemmaksi todellisista ongelmista. Asiakasyrityksen kehittäjien ja asiakkaan kommunikaatioon liittyviä ongelmia on paikattu siirtämällä kommunikaatiota eri viestintämenetelmiin ja jakamalla vastuuta kehittäjille ja asiakkaalle enemmän. Tämän kaltainen kommunikaatio yhdistettynä asiakkaan vastuulla ja välillä kykenemättömyydellä täyttää näitä vastuita ovat johtaneet ristiriitoihin, joskus jopa väärinkäyttöön. Lisäksi tämä sulkee pois myös asiakkaita, joilla ei ole valmiuksia johtaa oman järjestelmän kehitystä, jolloin se syö myös koko palvelun markkinakelpoisuutta. Puhelimitse käytyä viestintää on myös vaikeata dokumentoida, jolloin väärinkäytösten mahdollisuus kasvaa. Lisäksi prosessin intensiivikehitysvaiheen päättäminen, eli käyttöönotto on viivästynyt, kun tarvittavia keinoja ja aikatauluja testaamisen aikaan saamiseksi ei ole normalisoitu. Tässä suhteessa on kuitenkin toteutettu jo testaamista ja siinä mielessä kehityssuunta on oikea. Ketterien menetelmien pohjalta huomataan, että siirtämällä kehitystä juoksu/iteraatiomalliseksi saadaan tehostettua kommunikaatiota ja testausta projektin aikana. Lisäksi äärimmäisissä tapauksissa projektinjohdon siirtäminen kehittäjäyrityksen kontolle ratkaisisi asiakkaiden kokemattomuudesta johtuvia ongelmia, sekä lisäisi palvelun markkinakelpoisuutta.

Tutkimuksen tekeminen sujui sopivalla tahdilla, valtakunnallinen pandemia häiritsi haastattelujen järjestämistä, joten osa toteutettiin verkkohaastatteluina. Haastatteluissa haastattelijan oli vaikea saada haastateltavia näkemään itsensä objektiivisena tutkijana, sillä haastateltavat ja haastattelija olivat usein tuttuja entuudestaan. Luotettavuuden näkökulmasta tutkijan sukulaissuhde asiakasyrityksen toimitusjohtajaan on voinut vaikuttaa joihinkin vastauksiin haastatteluissa, mutta tutkijana olen pyrkinyt sulkemaan sen vaikutukset pois, eli en ole suosinut asiakasyritystä arvostelussani tai suoristanut mitään mutkia sen kummemmin. Luotettavuutta myös varjostaa se, että tutkija haastatteli haastateltavat anonyymina, pelkkään virkaansa pohjautuen, joten uudelleen tehtäessä voi olla, että täysin samat henkilöt eivät valikoidu haastateltaviksi. Toisaalta tämä antoi mahdollisuuden vastata kysymyksiin hieman avoimemmin. Haastattelujen yhteenvedot luetutettiin haastateltavilla asentajaa lukuun ottamatta, sillä häneltä saadut tiedot täsmennettiin asiakasyrityksen toimitusjohtajan haastattelussa, eivätkä ne olleet lopputuloksen kannalta kovinkaan kriittisessä roolissa.



Tutkija on tallentanut kaikki työvaiheet ylös, niistä jokaisen tekoa on tuotu työssä esille ja niiden tekeminen on perusteltu, sekä kuvattu. Tällä varmistutaan siitä, että jokainen työvaihe on nähtävissä, eikä tutkija ole tehnyt omia ratkaisujaan tai oikonut mutkia suoriksi. Tutkimusmenetelmien perustelu ja niiden käyttö toteutui kohtalaisella tavalla, haastatelluista saatu tieto ja kirjoitetut yhteenvedot antoivat tutkijalle hyvät lähtökohdat vertailua varten. Tästä syystä voi sanoa, että niihin valitut menetelmät ovat toimineet tutkimuksessa ainakin kohtalaisella tasolla. Laadullisen tutkimuksen näkökulma antaa hyvät lähtökohdat lähteä tutkimaan asiaa ja saada siitä yksilökohtaista tietoa esille, jotta voidaan kuvata todellisia tapahtumia.

Seuraavassa vaiheessa voitaisiin tehdä määrällistä tutkimusta asiakasyrityksien kanssa, jotta saataisiin numeraalista vahvistusta tutkimuksessa saaduille tuloksille, lisäksi normaalista poikkeava asiakkaan rooli toi uutta tietoa teoriapohjaan siitä, miten asioita voidaan tehdä eri tavalla. Lisätutkimusta voisi tehdä siitä, miten toimiva ratkaisu se on pitkässä juoksussa, se on kuitenkin asiakaslähtöisin tapa tuottaa järjestelmiä, sillä asiakas päättää kaikesta, mitä järjestelmä pitää sisällään. Lisäksi voidaan tutkia sitä, onko näin persoonallisten järjestelmien tuottaminen paras mahdollinen ratkaisu ja mitä asioita teknisellä puolella voidaan muokata itse järjestelmän osalta.

## Lähteet

### Painetut

Alasuutari, P. 2011 Laadullinen tutkimus 2.0

Bell, Steve; Bell, Steven C; Orzen Michael A. 2010: Lean IT

Hammaberg, M. & Sundén, J. 2014 Kanban IN ACTION

Hirsjärvi, S. & Hurme, H. 2008 Tutkimushaastattelu: teemahaastattelun teoria ja käytäntö

Juvonen, R. 2018 Ohjelmistoprojektien sudenkuopat ja miten ne vältetään

Kananen, J. 2017 Laadullinen tutkimus pro graduna ja opinnäytetyönä

Layton, Mark C; Morrow, David 2018: Scrum for dummies

Loader, N. 2019 The Lean IT Expert: Leading the Transformation to High Performance IT

Merton, R. K., Fiske, M. & Kendall, P. L. 1990 The Focused Interview. A manual of problems and procedures, Glencoe IL; Free Press.

Tuomi, J. 2018 Laadullinen tutkimus ja sisällönanalyysi. Ellibs-kirjastot

### Sähköiset

Ozkan, Necmettin; Tahran, Ayca 09.2019 A Review of Scaling Approaches to Agile Software Development Models. ProQuest: <https://search-proquest-com.nelli.laurea.fi/central/docview/2312156880/5F832E5530354B17PQ/1?accountid=12003>

### Julkaisemattomat

Kittilä, M. Prowledge OY, yritykseltä hankittu tieto

Asentaja 2020, Vaskisepät Oy. Haastattelu 05.05.2020

Nuorempi ohjelmistokehittäjä 2020, Prowledge Oy. Haastattelu 07.05.2020

Vanhempi ohjelmistokehittäjä 2020, Prowledge Oy. Haastattelu 05.05.2020

Toimitusjohtaja 2020, Prowledge Oy. Haastattelu 08.05.2020

Asiakas 2020, Vaskisepät Oy. Haastattelu 08.05.2020

Myyjä 2020, Prowledge Oy:n yhteistyö, Haastattelu 08.05.2020

## Kuviot

Kuvio 1: Vesiputousmallin rakenne (mukaillen Bell, Bell & Orzen, 173.).....	11
Kuvio 2: Scrumin rakenne (mukaillen Layton & Morrow 2018, 10.).....	13
Kuvio 3: Scrumin roolit (mukaillen Layton & Morrow 2018, 11.).....	14
Kuvio 4: Lean pyramidi (mukaillen Bell & Bell & Orzen 2010, 18.).....	17
Kuvio 5: nopeiden iteraatioiden malli (mukaillen Bell & Bell & Orzen 2010, 176.) .....	18