



Yritys X Oy Projektien kehittäminen

Harri Rajala

OPINNÄYTETYÖ
Toukokuu 2021

Teknologiaosaamisen johtamisen ylempi AMK-tutkinto

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Teknologiaosaamisen johtamisen ylempi AMK-tutkinto

RAJALA, HARRI:
Yritys X Oy Projektien kehittäminen

Opinnäytetyö 55 sivua, joista liitteitä 3 sivua
Toukokuu 2021

Tämän opinnäytetyön toimeksiantaja on yritys X Oy, joka tuottaa digitaalisia palveluita. Digitaaliset palvelut ovat haastavia ja laajoja ohjelmistokokonaisuuksia, jotka rakentuvat monenlaisista erilaisista komponenteista ja järjestelmistä. Suurin osa yritys X Oy:n liikevaihdosta muodostuu näistä digitaalisen palvelun projekteista ja tästä johtuen projektien onnistuminen on elintärkeää. Yleisesti ottaen ohjelmistoprojektien onnistuminen on hyvin haasteellista.

Tässä opinnäytetyössä tutkitaan ja tunnistetaan niitä tekijöitä ja syitä, jotka johtavat ohjelmistoprojektien onnistumiseen tai epäonnistumiseen. Lisäksi rakennetaan yritys X Oy:lle tarkistuslista projektien systemaattiseen aloittamiseen. Tarkistuslistaa on tarkoitus käyttää myös perehdytykseen kiinnitettäessä uusia työntekijöitä käynnissä oleviin projekteihin.

Tulosten mukaan yritys X Oy:ssä projektin tavoitteiden mielessä pitäminen, asiakaskommunikaatio ja ohjelmistokoodin katselmointi ovat onnistumisen kannalta oleellisia seikkoja, kun taas pieleen menemisen vaikuttaa heikko motivaatio, asiakkaan inaktiivisuus, testaamisen puute ja tekninen velka.

Digitaalisen palvelun ohjelmistoprojekti voidaan määritellä onnistuneeksi, kun projekti täyttää sille asetetut vaatimukset, pysyy budjetissa, valmistuu aikataulussa ja lopputuote täyttää asiakkaan tarpeet sekä on laadullisesti riittävä. Tämän opinnäytetyön sekä aikaisempien tutkimusten tulosten perusteella, voidaan sanoa, että selkeät ja saavutettavissa olevat tavoitteet sekä aktiivinen asiakaskommunikaatio ovat avaintekijöitä onnistuneen digitaalisen ohjelmistoprojektin kannalta.

Asiasanat: ohjelmistokehitys, projektin kehittäminen, digitaalinen palvelu, projektimalli

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Management of technology Expertise Master of Engineering

RAJALA, HARRI:
X Inc project development

Bachelor's thesis 55 pages, appendices 3 pages
May 2021

This thesis was commissioned by X Inc, which is a company that develops digital services. Digital services are challenging and extensive software packages consisting of a wide variety of different components and systems. Most of X Inc's revenue comes from these digital services projects. Consequently, the success of these projects is vital. In general terms, there are many challenges to realizing successful software projects.

This thesis examines and identifies the factors and reasons that lead to the success or failure of software projects. In addition, a checklist will be created for X Inc which enables the systematic launch of new projects. The checklist is also intended to be used for the orientation of new employees who are joining pre-existing projects.

According to the results, the factors which were associated with success at X Inc were keeping the project goals in mind, managing client communications, and reviewing the software code. Weak motivation, client inaction, lack of testing and technical debt were factors leading to failure.

A digital services software project can be defined as successful when the project meets the requirements set for it, stays on budget, is completed on schedule and the product meets the client's needs and is of adequate quality. Based on the results of this thesis as well as other previous studies, it can be stated that clear and achievable goals as well as active client communication are key factors for a successful digital software project.

Key words: software development, project development, digital service, project model

SISÄLLYS

1	JOHDANTO	7
	OPINNÄYTETYÖN TAUSTA JA TAVOITTEET	9
	1.1 Tavoitteet ja rajaukset	9
	1.2 Tutkimusmenetelmät.....	9
2	OHJELMISTOPROJEKTI	11
	2.1 Digitaalinen palvelu	11
	2.2 Projektin määritelmä	12
	2.3 Perinteisen projektin malli ja vaiheet.....	12
	2.4 Ohjelmistoprojektin vaiheet	15
	2.5 Vesiputousmalli.....	17
	2.6 Ketterämalli	18
	2.7 Tuotekehitys ja konseptointi	19
3	OHJELMISTOKEHITYKSEN MENETELMÄ - SCRUM.....	21
	3.1 Ketterä projektikehitys yleisesti	21
	3.2 Scrum.....	22
	3.3 Sprint Retrospektiivi	24
	3.4 Retrospektiivin vaiheet	25
	3.4.1 Viritä näyttämö.....	25
	3.4.2 Tietojen kerääminen	26
	3.4.3 Oivalluksien luominen.....	27
	3.4.4 Päätä mitä tehdään	27
	3.4.5 Retrospektiivin lopettaminen	27
4	OPPIMISEN TEORIA.....	28
	4.1 Oppimiskahvila.....	28
5	PROJEKTIN ALOITTAMISEN TUTKIMINEN KOHDEYRITYKSESSÄ	30
	5.1 Nykytilanteen kartoitus	30
	5.2 Esivalmistelut	30
	5.3 Retrospektiivin sisältö	31
	5.4 Retrospektiivin toteutus.....	32
6	TULOKSET	35
	6.1 Tiedon analysointi	35
	6.2 Tarkistuslistan muodostaminen projektin aloittamiseen	35
	6.3 Mitkä ovat projektien hyvät käytännöt?	37
	6.3.1 Tavoitteiden mielessä pitäminen ja seuraaminen	38
	6.3.2 Asiakkaan osallistuttaminen ja osallistumisen vaatiminen ..	39
	6.3.3 Katselmointi.....	41

6.4 Miksi projekti epäonnistuu?	41
6.4.1 Motivaatio	42
6.4.2 Testien puute.....	43
6.4.3 Asiakkaan inaktiivisuus.....	45
6.4.4 Tekninen velka	45
6.5 Yhteenveto.....	46
7 POHDINTA	49
LÄHTEET	51
LIITTEET	53
Liite 1. Ajatuskartta projektien aloittaminen.....	53
Liite 2. Ajatuskartta Mitkä ovat projektin hyvät käytännöt?.....	53
Liite 3. Ajatuskartta Miksi menee pieleen?.....	53
Liite 4 Tunnistettuja tehtäviä tarkistuslistaa varten.....	54
Liite 5 Miksi menee pieleen?.....	55
Liite 6 Retrospektiivin palaute	55

ERITYISSANASTO tai LYHENTEET JA TERMIT (valitse jompikumpi)

Agile	Ketterä
IEEE	kansainvälinen tekniikan alan järjestö
Scrum	projektinhallinnan viitekehys
MVP	Minimum viable product
Intra	Organisaation sisäinen lähiverkko
UX	Käyttäjäkokenemussuunnittelu
UI	Käyttöliittymäsuunnittelu
Web Frontend	Verkkoselaimessa ajettava koodi
Backend	Taustajärjestelmä
Refaktorointi	Koodin uudelleen kirjoittaminen
Business case	liiketoimintatarkastelu

1 JOHDANTO

Opinnäytetyön toimeksiantajana on vuonna 2009 perustettu it-alan yritys. Yrityksestä käytetään tässä työssä mainintaa yritys X Oy. Yritys X Oy fokusoitui aluksi pelien kääntämiseen eri alustoille. Muutama vuosi myöhemmin toiminta muuttui tavoitteellisemmaksi ja vuodesta 2015 lähtien yritys X Oy alkoi tehdä konsultointia mobiilialustoille. Konsultoinnin myötä vuonna 2020 yritys X Oy on kasvanut viiden henkilön tiimistä yli kaksikymmentä henkilöiseksi digitaalisia palveluita tarjoavaksi yritykseksi. Samana vuonna yritys X Oy fuusioitui isomman toimijan kanssa ja nykyään fuusioitunut toimija pystyy tarjoamaan palveluita aina mekaniikkasuunnittelusta digitaalisiin palveluihin. Tällä hetkellä fuusioitunut yritys työllistää yli 400 osaaajaa.

Opinnäytetyöntekijä on työskennellyt yritys X Oy:ssä vuodesta 2015 lähtien ohjelmistokonsulttina. Päivittäisen ohjelmistokehittämisen lisäksi opinnäytetyön tekijällä on halu parantaa olemassa olevia prosesseja ja käytäntöjä.

Yritys X Oy:n kasvaessa ja eri kokoisten projektien lisääntyessä toimintatavat ja prosessit lähtevät helposti pirstaloitumaan. Tämä aiheuttaa sen, että käytännöt ovat erilaisia projektien sisällä. Lisäksi projekteihin perehdyttäminen tapahtuu helposti siten, että asioista kerrotaan sitä mukaan, kun tulee tarvetta eikä tällä hetkellä ole systemaattista prosessia, miten perehdytys hoidetaan.

Opinnäytetyön tarkoituksena on ehkäistä pirstaloitumista ja yhtenäistää käytäntöjä sekä luoda digitaalisen palvelun ja projektin aloittamiseen toimintatapa projektin toteuttavan henkilöstön osalta. Lisäksi on tarkoitus tunnistaa tekijöitä, jotka ovat kulmakiviä onnistuneen digitaalisen palvelun tuottamisessa ja tekijöitä, jotka vaikuttavat osaltaan siihen, miksi projekti viivästyy aikataulustaan.

Opinnäytetyössä selvitettiin yritys X Oy:n nykytilanne projektin osalta tutkimalla sovelluskehittäjien ja myynnin toimintaa digitaalisen palvelun toteuttamisessa eri vaiheissa aina myyntiprosessista lähtien palvelun luovuttamiseen asiakkaalle asti. Yritys X Oy:llä on käytössään erilaisia järjestelmiä, joilla digitaalinen palvelu tuotetaan ja kehittämisen edistymistä sekä kuluja seurataan, mutta systemaattista ja yhtenäistä toimintakäytäntöä palvelun tuottamisen

aloittamiseen ei ollut. Näin ollen käytännölle oli kysyntää ja se voidaan jalkauttaa eri yksiköihin ja tällä tavoin yhtenäistää yritys X Oy:n toimintaa.

Ensiksi kuvataan suunnitelmallista digitaalista palvelun toteuttamista teoriassa, jonka jälkeen käydään läpi opinnäytetyössä käytetyt tiedon keruumenetelmät. Nykytilanteen kartoituksen jälkeen selvitettiin kehityskohteita työntekijöiden ja johdon yhteisellä retrospektiivillä yhdistettynä learning café eli oppimiskahvila oppimismenetelmään. Tuloksista laadittiin yrityksen projektien käynnistämiseksi yhteinen käytäntö.

Pohdintakappaleessa arvioidaan opinnäytetyön merkitystä yritykselle ja konkreettisia vaikutuksia työn tulosten perustella. Analysoinnissa on mukana myös kappale tavoitteiden saavuttamisesta.

OPINNÄYTETYÖN TAUSTA JA TAVOITTEET

Kun vuonna 2015 pieni tiimi teki pelkästään konsultointia, prosessit ja käytännöt tulivat asiakkaan puolelta. Yritys X Oy:n kasvaessa uusien ohjelmistopalveluprojektien myötä huomattiin, että on tarve kehittää toimintatapoja yhteisille prosesseille ja käytännöille. Esimerkkinä yrityksellä on projekteissa käytössä koodin suhteen Git -hajautettu versionhallintajärjestelmä. Olennaista on, että määritetään yhteinen käytäntö, miten kyseistä järjestelmää käytetään projektin sisällä.

1.1 Tavoitteet ja rajaukset

Tavoitteena on tunnistaa yritys X Oy:ssä ohjelmistoprojektien hyviä ja huonoja käytäntöjä sekä kehittää niitä ja luoda projektin aloituksen yhteyteen yhteinen tarkistuslista. Tarkistuslistan avulla projekteihin saadaan systemaattinen aloitustapa sekä selkeä malli projektiin liittyvälle uudelle työntekijälle. Tarkistuslistan merkittävä hyöty on myös, että sen läpikäymisen jälkeen saadaan projektiin liittyvät resurssit, työkalut ja menetelmät kirjattua ylös yritys X Oy:n intraan, josta se on käytettävissä koko henkilöstölle.

Työ rajataan uusien ohjelmistoprojektien käynnistämisen yhteyteen ja olemassa olevien projektien uusien henkilöiden perehdyttämiseen projektiin.

1.2 Tutkimusmenetelmät

Opinnäytetyön tutkimusmenetelmäksi valittiin kvalitatiivinen tutkimusmenetelmä eli laadullinen tutkimus, joka sisältää erilaisia lähestymistapoja sekä aineistonkeruu- ja analyysimenetelmiä. Lähtökohta kvalitatiiviselle tutkimukselle on todellisen elämän kuvaaminen. Tutkimuskohdetta on tarkoitus tutkia mahdollisimman kaikenkattavasti ja aineisto kootaan luonnollisessa ja todellisessa tilanteessa. Kvalitatiivisessa tutkimuksessa pyritään löytämään ja paljastamaan tosiasioita sekä todentaa jo olemassa olevia asioita (Hirsjärvi, Remes & Sajavaara 2007, 157). Tutkimusmenetelmän valinnan perustella projektin kehittämisen tutkimuskysymyksiksi asetettiin seuraavat kysymykset:

- 1) Mitkä ovat projektin hyvät käytännöt?

2) Miksi projekti epäonnistuu (Miksi menee pieleen?)

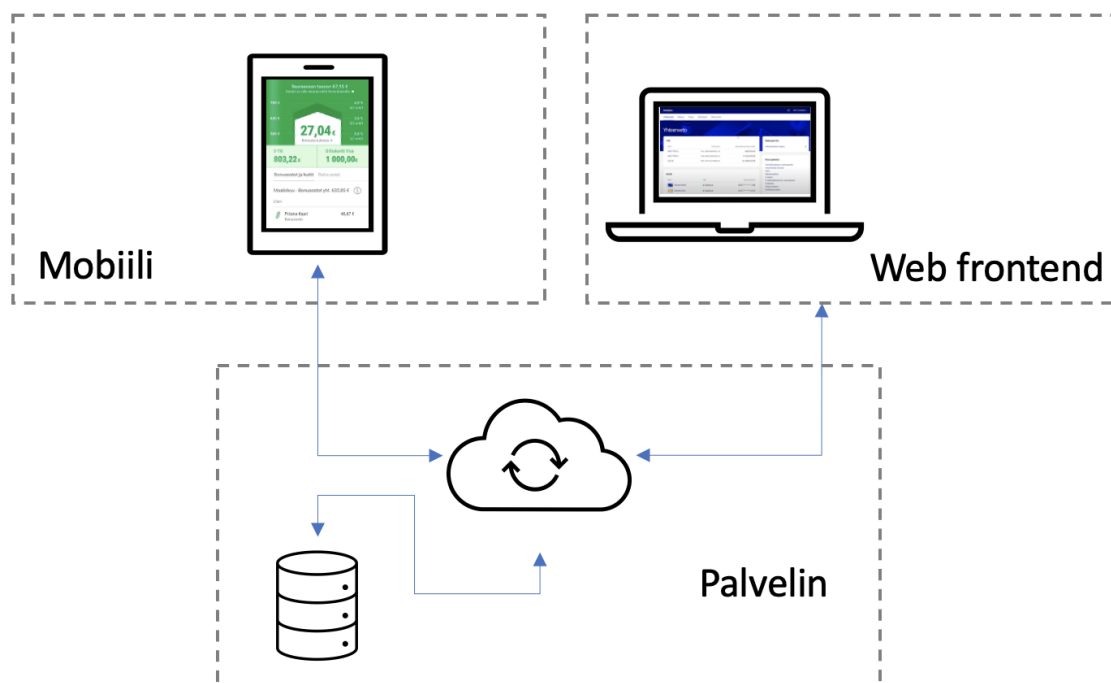
Päättötyön toteuttamistavasta yliopettaja Mäkilouko ehdotti oppimiskahvilaa eli keskusteluun sekä tiedon luomiseen ja siirtämiseen tarkoitettua menetelmää (Mäkilouko 2019). Nopealla tutkimisella menetelmä vaikutti hyvältä lisäykseltä retrospektiivin oheen, sillä molemmissa tutkimusmenetelmissä koostuu saman tyyppiset toimintamallit.

Tutkimusmenetelmässä käytetään uudenlaista tutkimusasetelmaa. Tutkimus suoritettiin ketterästä ohjelmistoprojektista tunnetulla käytännöllä nimeltä sprint seremonia eli retrospektiivi ja oppimiskahvilaan pohjaavalla ryhmäkeskustelulla. Retrospektiivissä tavoitteena on tarkastella mennyttä ajanjaksoa sekä tunnistaa asioita, jotka menevät hyvin ja toisaalta myös niitä, jotka tarvitsevat kehittämistä. Oppimiskahvila -menetelmässä keskustellaan valmiiksi annetuista aihealueista ja haetaan erilaisia näkökulmia kehitettävän asian ratkaisuksi. Keskustelussa olevat henkilöt olivat yrityksen henkilöstöä eri osa-alueilta: myynti, ohjelmistonkehitys ja johto.

2 OHJELMISTOPROJEKTI

2.1 Digitaalinen palvelu

Tässä työssä projektina käsitellään digitaalisen palvelun toteuttamista asiakkaalle ketterän ohjelmistokehitysmenetelmän mukaan. Digitaalinen palvelu on kokonaisuus, joka pääsääntöisesti sisältää mobiili, Web frontend eli koodin, jota ajetaan verkkoselaimessa, kun käytät sivua, UX:n eli käyttäjäkokemussuunnittelun, UI:n eli käyttöliittymäsuunnittelun ja pilvipalvelinpuolen toteutuksen.



KUVA 1. Digitaalisen palvelun yleisen tason kuvaus

Yritys X Oy:n projektit sisältävät pääsääntöisesti kokonaisuuden tai sen osan toteuttamisen ja toimittamisen asiakkaalle. Onnistuneen digitaalisen palvelun tuottamisen osalta olennaista on alkuvaiheessa käydä läpi asiakkaan kanssa kokonaisuus löytämällä vastauksia kysymyksiin: Kenelle palvelu toteutetaan? Miksi se tehdään? Mitä sillä ratkotaan? Näin ollen tiedetään, että onko palvelua tai sen osaa järkevää edes lähteä toteuttamaan. Hyvinä tunnettuina esimerkkeinä

digitaalisista palveluista toimivat suomalaisten pankkien mobiili- ja verkkopankkisovellukset.

2.2 Projektin määritelmä

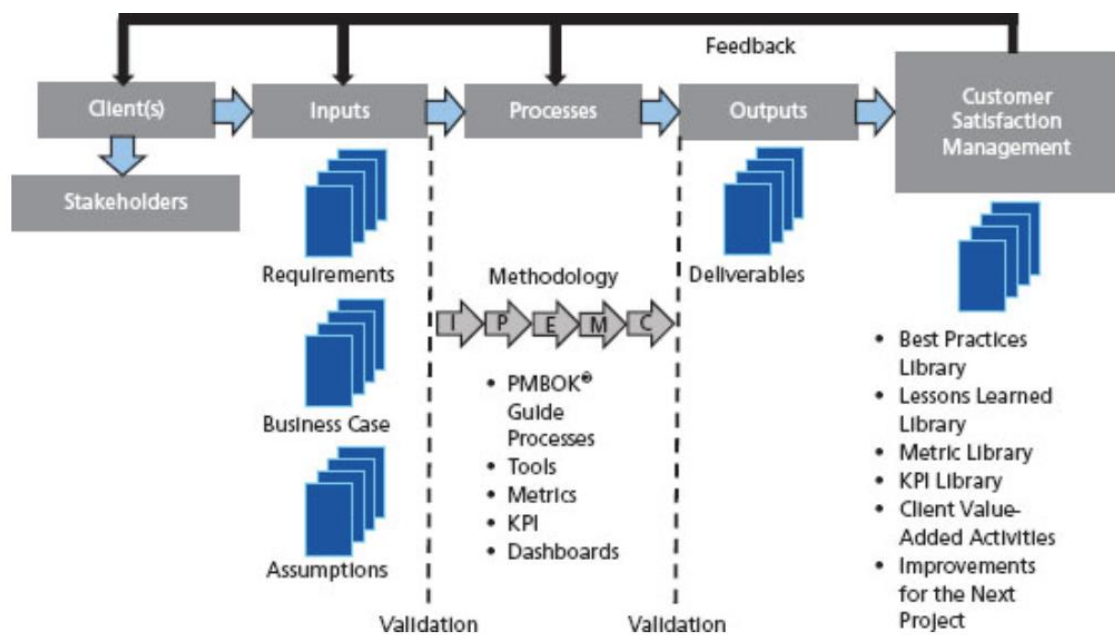
Mäkiloukon mukaan projekti määritellään siten, että se on suunnitelmallinen ja tarkasti rajattu kokonaisuus, jolla on määrätty aika, tavoite, rahoitus ja sisältää rajoitteita (Mäkilouko 2020). Wysocki sanoo, että ohjelmistokehitysprojekti on monimutkainen hanke ajan, budjetin sekä henkilöstöressurssien rajoissa ja tuottaa uutta tai paranneltuja ratkaisuja, joka lisää merkittävää liiketoiminnallista arvoa uudelle tai olemassa olevalle liiketoimintaprosessille (Wysocki 2006). Voidaan todeta, että projekti on ainutkertainen kokonaisuus, joka sisältää monimutkaisia ja toisiinsa liittyviä tehtäviä sekä tähtää ennalta määrättyyn päämäärään.

Digitaalisen palvelun osalta projektin lopun määrittelemisen voi olla haastavaa, koska projektin alkamisen jälkeen kehitystä voidaan jatkaa hyvin pitkään. Digitaalisen palvelun kehittäminen sisältää yleensä pieniä toiminnallisuuksia, joista rakentuu kokonaisuus. Digitaalisen palvelun (projektin) osalta päättyminen voidaan määritellä tapahtuneeksi, kun asiakas hyväksyy MVP:n (Minimum Viable Product). Termille ei ole vakiintunutta käännöstä suomen kielessä, mutta Project-it verkkojulkaisun mukaan se on ensimmäinen prototyyppi, pilottitoteutuksen kaltainen tuotos, jonka tarkoituksena on kerätä tietoa. MVP on kokonainen, tuotantokelpoinen, ehjä ja laadukas kokonaisuus, joka lähtökohtaisesti tuottaa maksimaalista arvoa ja on minimaalisella panoksella tuotettu (Project-IT 2021).

Kettunen luokittelee kirjassaan projektit kuuteen eri luokkaan: yrityksen sisäinen kehitys-, tutkimus-, toteuttamis-, rakennus-, toimitus- tai tuotekehitysprojekti (Kettunen 2009, 17). Digitaalisen palvelun kehittäminen kuuluu toimitusprojekteihin. Voidaan puhua kokonaistoimituksesta, joka käsittää sen, että koko palvelu tuotetaan asiakkaalle ja palvelun osat kehitetään kokonaan tai osittain talon sisällä.

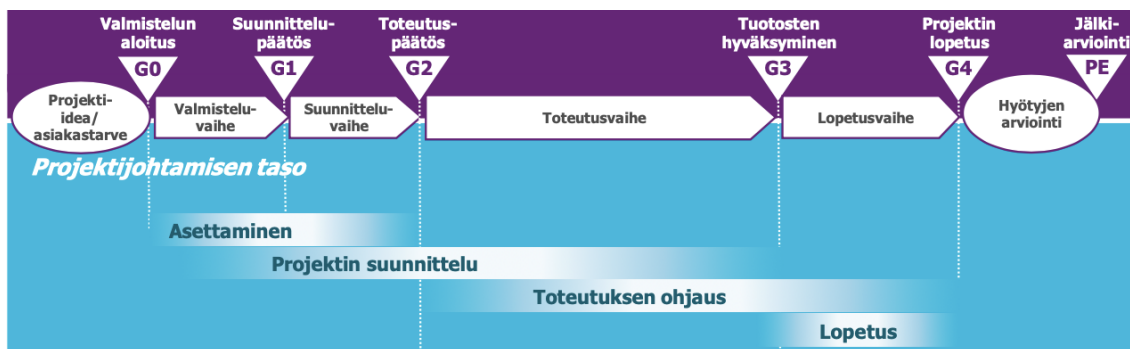
2.3 Perinteisen projektin malli ja vaiheet

Harold Kerzner kuvaa kirjassaan projektin vaiheet yleisellä tasolla seuraavasti: 1) Asiakas ja sidosryhmät 2) Syötteet sisältäen vaatimukset, business caset eli liiketoimintatarkastelut ja oletukset 3) prosessit ja toteuttaminen 4) tuote tai palvelu ja siihen sisältyvät toimitukset 5) asiakkaan tyytyväisyys. Ennen toteutusvaihetta numero kolme tehdään validoinnit kohdan kaksi syötteille ja näitä korjataan tarvittaessa. Samoin tehdään ennen tuotteen toimitusta, joten toteutusvaiheessa muutetaan tuotetta niin, että se vastaa haluttua lopputulosta. Kohdassa viisi varmistetaan asiakastyytyväisyys toimituksen ja tuotteen tai palvelun osalta. (Kerzner, 2011, 25)



KUVA 2 Projektin yleinen rakenne (Kerzner, 2011, 26/241)

Suomen projekti-instituutin kehittämän ABC-projektimallin mukaan projektin vaiheet menevät seuraavasti: Asiakastarve, valmistelu ja määrittely, suunnittelu, toteutus, lopetusvaihe ja jälkiarviointi. Vaiheiden välillä on päätöksentekopisteet eli portit (G), joiden kautta projektin eri vaiheet etenevät. Projektijohtamismallissa määritellään, millä organisaatiotasolla mikäkin porttipäätös (G0-G4) tehdään. Jokaisessa päätöksentekopisteessä päätetään, voiko projekti edetä, lopetetaanko projekti, tarvitaanko etenemistä varten lisäselvityksiä vai pysäytetäänkö projekti tilapäisesti (Suomen projekti-instituutti).



KUVA 3. Projektin vaiheet ABC-Malli (Suomen projekti-instituutti, 2017, 25/47)

Projektin käynnistämisen taustalla on vastata asiakastarpeeseen tai markkinoilla tunnistettu mahdollisuus. Projektiehdotus voi myös perustua havaittuun muutostarpeeseen tai ideaan. Projektiehdotuksesta kuvataan hyötytavoitteet ja muut alkuvaiheessa tiedossa olevat lähtökohdat ja sen hyväksyminen aloittaa valmisteluvaiheen. (Adapro, 2021).

Valmisteluvaiheen aikana projektiehdotusta työstetään edelleen, suoritetaan tarpeelliset esiselvitykset liiketoiminnallisista lähtökohdista ja dokumentoidaan valmistelutyön tulokset projektikuvaukseksi. (Adapro, 2021).

Projektsuunnitelma on realistinen suunnitelma, jonka tarkoitus on kuvata projektin tilaajalle ja ohjausryhmän jäsenille, miten projektin tavoitteet vastaavat olemassa olevaan tarpeeseen ja millä aikataululla, toiminta- ja toteutustavoilla sekä budjetilla projektin tavoitteisiin päästään. Lisäksi suunnitelma sisältää projektin laajuuden ja siihen liittyvät rajaukset. (Adapro, 2021).

Toteutusvaiheessa suoritetaan hyväksytyyn projektsuunnitelman kaikki tehtävät ja tehtävien suorittaminen on projektipäällikön ja projektitiimin vastuulla. Projektipäällikkö viestii projektin edistymisestä ohjausryhmälle ja sidosryhmille. (Adapro, 2021).

Projektin lopetusvaihe alkaa, kun projektin lopputuotokset on hyväksytty. Lopetusvaiheen aikana laaditaan projektin loppuraportti, joka sisältää suunnitelman projektin jälkeisten liiketoimintahyötyjen arvioinnista. Lopetusvaiheen aikana varmistetaan, että vastuu projektin tuotoksista on siirretty

pysyvälle organisaatiolle ja kaikki tarvittavat lopetustoimet on tehty. (Adapro, 2021).

Jälkiarviointi tehdään jonkin ajan kuluttua projektin päätöksestä (6-12kk), kun nähdään, ovatko projektin hyödyt toteutuneet. Jälkiarvioinnin idea on tarkastella projektin alkuperäisiä hyötyjä ja strategisia päämääriä. Sen aikana tulisi saada vastaukset seuraaviin kysymyksiin:

- Onko tuotos käytössä?
- Kannattiko projektin toteutus?
- Toteutuiko business case?

Jälkiarvioinnin suunnitelma voidaan sisällyttää projektin loppuraporttiin ja tämän suorittaminen on ensisijaisesti omistajan vastuulla. (Adapro, 2021).

2.4 Ohjelmistoprojektin vaiheet

Digitaalisen palvelun kehittäminen on osa ohjelmistotuotantoa ja Institute of Electrical and Electronics Engineers eli kansainvälinen tekniikan alan järjestö määrittelee tämän siten, että se on kurinalaista, järjestämällistä, mitattavissa olevaa tapaa ohjelmistojen kehittämiseen, käyttöön ja ylläpitoon. (IEEE: Ontology driven software engineering 2012). Käytännössä ohjelmistotuotanto sisältää kaiken tietokoneohjelmistojen valmistukseen käytettävän prosessinhallinnan ja erilaiset ohjelmistokehitysmenetelmät.

Pierre Bourque ja Richard E. Fairley kuvaavat päivitetystä kansainvälisessä ohjelmistotekniikan ohjeessaan eli SWEBOK:ssa ohjelmistotuotannon vaiheet seuraavasti:

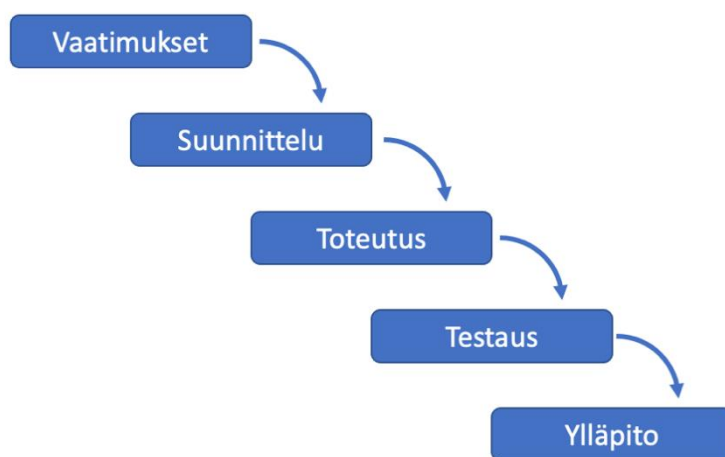
1. Software Requirements - Vaatimukset: a) esittävät ominaisuuksia, jotka ratkaisevat ongelman todellisessa maailmassa. b) miten ohjelmiston / palvelun tulee toimia loppukäyttäjän tai tilaajan näkökulmasta.
2. Software Design - Malli: kuvaa ohjelmistoarkkitehtuuria eli kuinka ohjelmisto organisoidaan komponenteiksi - ja näiden komponenttien välisiä rajapintoja sillä tasolla, että niiden tekeminen on mahdollista.

3. Software Construction - Rakentaminen on toimivan ohjelmiston yksityiskohtainen tekeminen koodauksen, katselmoinnin, yksikkötestauksen, integraatiotestauksen ja virheenkorjauksen yhdistelmiä.
4. Software Testing - Testaaminen on joukko menetelmiä, joilla varmistetaan siitä, että ohjelmiston ja tai palvelun toiminnallisuudet toimivat halutulla tavalla ja että se on riittävän virheetön käytettäväksi.
5. Software Maintenance - Ylläpito on olennainen osa ohjelmiston elinkaarta. Käyttöönoton jälkeen siirrytään ylläpitovaiheeseen, jolloin korjataan virheitä ja toteutetaan uusia toiminnallisuuksia.
6. Software Configuration Management - Ohjelmistokokoonpanon hallinta on ohjelmistoja tukeva elinkaari prosessi, joka hyödyttää projektinhallintaa, kehitys- ja ylläpitotoimia, laadunvarmistustoimia sekä lopputuotteen asiakkaita ja käyttäjiä.
7. Software Engineering Management - Ohjelmistotuotannon hallinta kattaa johtamistoimien: suunnittelun, koordinoinnin, mittaamisen, seurannan, valvonnan ja raportoinnin sekä varmistaa, että ohjelmistotuotteet ja ohjelmistotuotepalvelut toimitetaan tehokkaasti, tuloksellisesti ja hyödyttävät sidosryhmiä.
8. Software Engineering Process - Ohjelmistosuunnitteluprosessit koskevat menetelmiä, miten kehittäjiä tulee suorittaa ja ajoittaa eri tehtäviä kuten vaatimukset, suunnittelu, kehittäminen, testaus ja kokoonpanon hallinta.
9. Software Engineering Models and Methods Software Quality - Ohjelmistotuotantomallit ja -menetelmät kuvaavat tarkemmin ohjelmistokehityksessä käytettäviä tapoja, joita kuten mallinnusta (kaaviot) ja erilaisia suunnittelumenetelmiä esimerkkinä prototyyppi ja ketterät menetelmät.
10. Software Quality - Ohjelmistojen laatu vastaa testausta laajemmin kysymykseen onko ohjelmisto/palvelu hyvä, vastaako tuote vaatimuksia, käyttömukavuutta, toiminnallisuutta ja loppukäyttäjän tarpeeseen. (Bourque & Fairley 2014. kappaleet 1-10)

Edellä kuvatuista osa-alueista viidestä ensimmäisestä rakentuu ohjelmistoprojektin tai -palvelun elinkaari, jonka kautta projektin toteuttamisvaihe etenee.

2.5 Vesiputousmalli

Perinteistä projektin kulkua ohjelmistotuotannossa vastaa vesiputousmalli. Youssef Bassil kirjoittaa artikkelissaan, että vesiputousmalli on peräkkäinen ohjelmistokehitysprosessi, joka kulkee ylhäältä alaspäin mallissa määritellyistä vaiheista kuin vesiputous. Mallin esitti Winston W. Royce vuonna 1970 ohjelmistotekniikan kuvaamiseksi. Vesiputousmalli määrittelee useita peräkkäisiä vaiheita, jotka on suoritettava peräkkäin ja siirryttävä seuraavaan vaiheeseen vasta, kun sen edellinen vaihe on valmis. Tästä syystä vesiputousmalli on rekursiivinen siinä mielessä, että jokaista vaihetta voidaan toistaa loputtomasti, kunnes se on täydellinen. (Bassil 2012). Kuva viisi esittää vesiputousmallin vaiheet.



KUVA 4. Vesiputousmalli

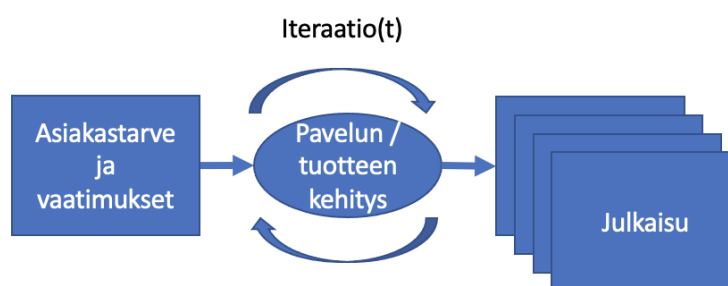
Malli sisältää viisi eri vaihetta: vaatimusmäärittelyn, suunnittelun, toteutuksen, testauksen ja ylläpidon. Vesiputousmallin ongelmana voidaan pitää kehityksen hitautta, muutoksien tekemisen kankeutta ja lopputuotteen valmiiksi saattamista. Esimerkkinä jos testausvaiheessa huomataan, että vaatimus on määritelty väärin tai ristiriidassa toisen vaatimuksen kanssa, korjaaminen vaatii koko prosessin läpikäymisen uudelleen. Tästä johtuen koko tuotteen julkaiseminen viivästyy.

2.6 Ketterämalli

Rachelle Lynnin verkkojulkaisun mukaan ketteräohjelmistokehitys sai alkunsa, kun seitsemäntoista ohjelmistokehittäjää mukaan lukien Martin Fowler, Jim Highsmith, Jon Kern, Jeff Sutherland, Ken Schwaber tapasivat hiihtokeskuksessa Utahissa keväällä 2000, keskustellakseen siitä, miten ohjelmiston kehitysaikaa voidaan nopeuttaa ja kuinka saadaan tuotettua uusia ohjelmistoja markkinoille nopeammin. (Lynn 2021). Ketterän ohjelmistokehityksen periaatteet vastaavat vesiputousmallin ongelmakohtiin. Ketterän kehityksen pääperiaatteet ovat seuraavat:

- 1) nopea julkaisutahti kahden viikon – kuukauden välein
- 2) muuttuviin vaatimuksiin reagoiminen myös myöhäisessä vaiheessa
- 3) sidosryhmien ja ohjelmoijien päivittäinen yhteistyö
- 4) tiedonvälitys tiimin kesken tehdään kasvokkain
- 5) toimiva ohjelmisto on keskeinen mittari ja yksinkertaisuus on oleellista eli tekemättä jätettävän työn maksimointi. (AgileManifesto 2001)

Ketterän ohjelmistokehityksen periaatteet yhdistettynä kuvassa kolme esitettyyn vaihekuvaan saadaan korkean tason malli ketterälle menetelmälle.



KUVA 5. Ketterämalli

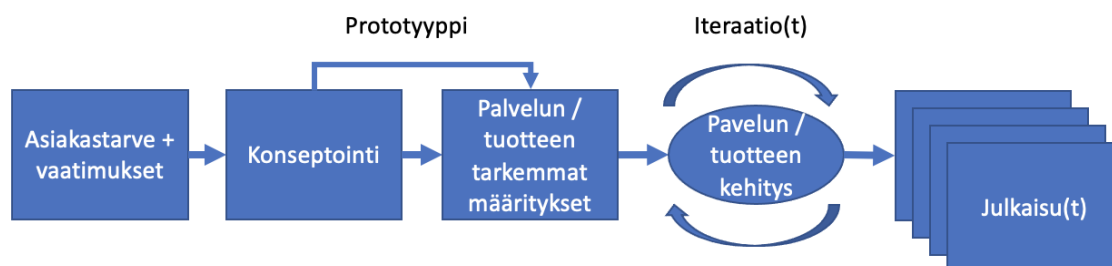
Ketterässä toimintamallissa kartoitetaan asiakastarpeet, määritellään vaatimukset ja reunaehdot sekä tavoitteet riittävällä tasolla projektin suunnitelman kannalta. Hyvin tehdyn määrittelyn jälkeen on koossa spesifikaatio, jossa halutut toiminnallisuudet on esitelty. Toiminnallisuudet priorisoidaan tiimin kesken ja tiimi valitsee kehitettävät toiminnallisuudet, jotka katsotaan valmistuvan

iteraation eli kehityssyklin aikana. Iteraation jälkeen tehdään tuotteesta julkaisu, jolloin asiakas pääsee kokeilemaan toteutettuja toiminnallisuuksia.

Esimerkkinä ajatellaan tilannetta, että on olemassa ohjelmistoprojekti, jossa on 100 toiminnallisuutta. Vesiputousmallin mukaan jokainen toiminnallisuus pitää olla toteutettuna ennen julkaisua. Ketterän toimintamallin mukaan riittää minimimäärä toiminnallisuuksia, että asiakkaalle pystytään näyttämään tuotteen tai palvelun kehittymisen etenemistä. Tästä johtuen asiakas pystyy nopeasti antamaan palautetta ja tiimi pystyy nopeasti reagoimaan muutoksiin. (Eric Verzuh s.328 2012).

2.7 Tuotekehitys ja konseptointi

Seija Bergströmin ja Arja Leppäsen mukaan *”tuotekehitys tarkoittaa täysin uusien tuotteiden eli innovaatioiden tuottamista ja entisten parantamistyötä eli muutoksen aikaansaamista”* (Bergström & Leppänen s.154 2018). Tuotekehityksen vaiheita ovat: 1) ideoiminen ja arvioiminen 2) kehitteleminen ja testaaminen 3) tuotteistaminen 4) lanseeraaminen (Bergström & Leppänen s.160 2018). Perinteiset tavaratuotannon tuotekehitysmallit eivät välttämättä sovi palveluiden kehittämiseen ja tuotteistamiseen. Niihin liittyviä toimia nimitetään konseptoinniksi tai palvelujen systematisoinniksi. (Bergström & Leppänen 2018, 167). Näitä yhdistämällä ketterään malliin saadaan palvelun kehittämiseen sopiva malli.



KUVA 6 Palveluiden kehittäminen yhdistettynä ketterään malliin

Konseptoinnissa aloitetaan palvelun ja sovelluksen varsinainen ideointi. Vaatimuksien täyttämiseksi suunnitellaan ne ratkaisut, joilla tavoiteltu

toiminnallisuus saadaan aikaiseksi. Tässä vaiheessa palveluun tai tuotteeseen on vielä helppo tehdä muutoksia ja lopputuloksena olevaa suunnitelmaa voidaan jo käyttää ensimmäisen prototyypin rakentamiseen.

Prototyypin tarkoituksena on yleensä testata erilaisia ratkaisuja käyttöliittymän toteutukseen. Käyttöliittymästä toteutetaan nopeasti kuvia ja demoja, joiden avulla saadaan selville, miten hyvin hahmoteltu toteutus osuu asiakkaan näkemykseen. Juvosen mukaan prototyypivaiheen edetessä väärinymmärrykset vähenevät ja vaatimukset tarkentuvat, että päästään paremmin haluttuun lopputulokseen (Juvonen 2018). Tämän jälkeen palvelun kehittämistä jatketaan ketterän ohjelmistomallin mukaan.

Esimerkkinä taksipalvelu, jossa asiakkaan tarve on saada digitaalinen palvelu, jossa yrityksen autot ovat kartalla näkyvissä ja niihin voi tehdä tilauksen mobiilisovelluksen kautta. Konseptoinnin avulla on mahdollista tuottaa nopea prototyyppi mobiilisovelluksen käyttöliittymästä ja siinä tapahtuvista interaktioista. Lisäksi konseptointi auttaa ymmärtämään paremmin asiakastarpeen ja vahvistamaan tai muuttamaan tarvittaessa business casea. Prototyypin kautta on mahdollista huomata aikaisessa vaiheessa ongelmakohtia eli mitä on tarvetta muuttaa ja täten pystytään reagoimaan ketterästi asiakkaan palautteeseen.

3 OHJELMISTOKEHITYKSEN MENETELMÄ - SCRUM

Kappaleen tarkoitus on kuvata yleisesti Scrumin ketterän ohjelmistokehityksen menetelmää ja antaa kuva, mitä ketterä ohjelmistokehitys on. Lisäksi kappale toimii johdantona työssä käytettyyn datankeräysmenetelmään eli retrospektriiviin. Ketterä ohjelmistokehitys on de facto standardi kaikissa nykypäivän ohjelmistotaloissa, jotka tuottavat digitaalisia palveluita.

3.1 Ketterä projektikehitys yleisesti

Hugh Beyer toteaa kirjassaan, että ketterät menetelmät ovat suosittuja, koska ne ratkaisevat sekä ohjelmistokehittäjien että johtamisen ongelmia.

Ohjelmistokehittäjät loivat menetelmät, koska he tunsivat olevansa voimattomia ja kykenemättömiä hallitsemaan projekteja. Heidän mielestään aikataulut saneltiin heille ja vaatimuksia sekä projektin laajuutta muutettiin koko projektin ajan. Usein projektin (tuotteen) tai palvelun julkaisun yhteydessä käyttäjät eivät halunneetkaan sitä. Ohjelmistokehityksen systemaattisen tekemisen keinoja ovat projektin jakaminen lyhyiksi sprinteiksi ja toiminnallisuuden jäädyttäminen ennen sprintti syklin alkua sekä palautteen saaminen sprintin aikana ja jälkeen (Beyer 2010, 3).

Esimerkki epäonnistuneesta ohjelmistoprojektista: Yritys kehitti tuotteen demosta, joka oli alun perin rakennettu kovalla kiireellä messu- ja tapahtumakäyttöön. Ensimmäinen ongelma projektille (tuotteelle) oli ohjelmiston arkkitehtuuri, jota ei suunniteltu millään tavalla. Tämän jälkeen saneltiin, mitä toiminnallisuuksia siinä haluttiin olevan ja tuotteen käyttötarkoitukset muuttuivat projektin elinkaaren aikana. Tämän lisäksi tuotteen julkaisut venyivät kolmesta neljän kuukauden sykleiksi.

Johto pitää ketterästä kehityksestä, koska se antaa heille käsityksen ja hallinnan ohjelmistoprojektista. Kahden viikon tai kuukauden kehityssykliä antavat paremmin tilannetietoa kehityksestä sidosryhmille kuin se, että odotettaisiin kuukausia tai vuosia havaitakseen, että lopputuote on edelleen vielä kuukausien tai vuosien päässä. Näin ollen sidosryhmät tietävät tarkalleen,

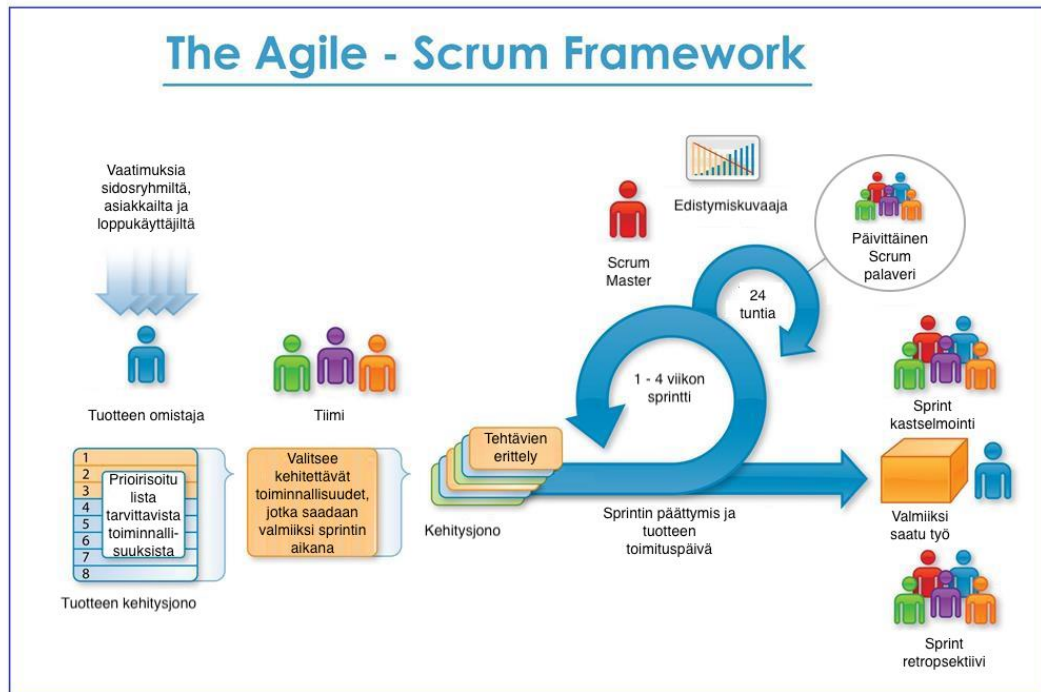
mitä sprintissä tehdään ja milloin sprintti suoritetaan. Lisäksi mahdolliset ongelmat ja muutokset ilmenevät nopeasti. (Beyer 2010, 3).

Eräässä maksamiseen liittyvässä ohjelmistoprojektissa tuotekehitystä toteutettiin Scrumilla kahden viikon iteraatioissa eli kehityssykleissä. Sprint suunnittelussa ohjelmistosuunnittelijat kävivät läpi, mitä ominaisuuksia tässä ajassa saadaan toteutettua ja mitä ominaisuuksia tarvitsee jakaa pienempiin osiin sekä siirtää seuraavaan sprinttiin. Tuotekehitykseen sprintin tehtäväjono oli johdon nähtävillä, joten johdossa oli reaaliaikainen tieto, mitä toiminallisuuksia tulee seuraavaan julkaisuun.

3.2 Scrum

Ken Schwaber ja Jeff Sutherland kirjoittavat verkkojulkaisussaan, että Scrum on kevyt kehys, joka auttaa ihmisiä, tiimejä ja organisaatioita tuottamaan arvoa mukautuvilla ratkaisuilla monimutkaisiin ongelmiin (Schwaber & Sutherland 2020). Scrum-viitekehys antaa suuntaviivat tuotteen tai palvelun kehitykseen ja sen sisällä voidaan käyttää erilaisia prosesseja, tekniikoita ja menetelmiä. Nämä tulee valita projektin tai tuotteen kannalta tapauskohtaisesti. Scrum käyttää iteratiivista, inkrementaalista lähestymistapaa ennustettavuuden optimoimiseksi ja riskien hallitsemiseksi. Scrumin päämäärä on tuoda käytettyjen työtapojen suorituskyky esille mahdollistaen työympäristön sekä tuotteen jatkuvan parantamisen.

Alla oleva kuva antaa kokonaiskuvan Scrum viitekehyyksen toimijoista ja yleisistä toimintaperiaatteista.



KUVA 7. Scrum viitekehys (Lara, 2018 vapaasti suomennettu)

Tuoteomistaja vastaa siitä, että Scrum-tiimi tuottaa maksimaalisen arvon.

Hän selvittää sidosryhmien ja loppukäyttäjien vaatimukset sekä hallinnoi ja priorisoi ne siten, että asiakas saa tuotteen kehityksestä parhaimman hyödyn. Hän vastaa tuotteen kehitysjonon läpinäkyvyydestä, saatavuudesta ja ymmärrettävyydestä. (Scrum-Opas 2020, 6). Tuoteomistaja toimii käytännössä kahteen suuntaan. Hänellä on vastuu huolehtia siitä, että eri sidosryhmiltä tulevat vaatimukset ovat selkeitä ja järkevästi priorisoitu tiimin suuntaan. Toisaalta hän kommunikoi Scrum-tiimiltä tulevaa palautetta sidosryhmille, mikäli jokin vaatimuksen toteuttaminen ei ole realistista tai sen prioriteettia tarvitsee muuttaa.

Sprint on yhdestä neljään viikkoon kestävä iteraatio. Luultavasti suosituin sprintin pituus on kaksi viikkoa.

Tiimi Scrum-tiimin jäsenien menestyminen on yhtä kuin tiimin menestyminen ja näin ollen tiimi on sitoutunut tekemiseen. Tiimi päättää sprintin suunnittelupalaverissa, miten paljon toiminnallisuuksia he pystyvät toteuttamaan yhdessä sprintissä. (Beyer 2010, 4). Käytännössä hyvin toimivissa tiimissä sen jäsenet määrittävät yhdessä toiminnallisuuksille työmääräarviot, jonka

perusteella pystytään rajaamaan sprintin kokonaisuus siten, että se on toteutuksen kannalta järkevä.

Kehitysjono. Scrum-projektit alkavat tuote- ja sprinttitapahtumista. Jokainen kehitettävä asia on työn yksikkö, joka on suoritettava sprintin aikana. Projektin alussa tiimi listaa tehtävät ja priorisoi kehitysjonon; jokaisen sprintin alussa tiimi valitsee kohteet, jotka toteutetaan sprintissä. (Beyer 2010, 5).

Tiimiä ohjaa **Scrum Master**, joka hallitsee prosessia. Hän johtaa päivittäisen scrum kokouksen ja ohjaa suunnittelukokoukset tuotekehityksen tilanteen määrittämiseksi. Scrum master voi olla projektipäällikkö, mutta hän ei voi tällöin toimia tuotteen omistajana. (Beyer 2010, 4).

Scrum määrittelee erilaisia projektinhallintatyökaluja edistymisen seuraamiseksi sprintin aikana. Esimerkkinä sprintin vähenevä edistymiskuvaaja, joka kertoo sidosryhmille ja tiimille, mikä on kehityksen sen hetkinen tilanne.

Päivittäinen scrum on palaveri, jossa seurataan edistymistä ja suunnitellaan päivän tehtävät. (Beyer 2010, 4). Palaverissa käydään läpi jokaisen tiimin jäsenen tilanne: mitä he ovat tehneet, mitä he aikovat tehdä tänään ja onko heillä jotain esteitä.

Sprintin lopussa on katsaus sprintissä tehtyyn työhön tuotteen omistajan ja sidosryhmien kanssa. Tämä on lyhyt läpikäynti, maksimissaan neljä tuntia. Teoriassa tässä vaiheessa kehityksen ja tiimin suuntaa arvioidaan uudelleen. Kehitysjonoa priorisoidaan uudelleen ja uusia tehtäviä voidaan kirjoittaa vastaamaan palautteeseen. (Beyer 2010, 5). Käytännössä sprintin lopussa pohditaan omia prosesseja ja koitetaan tunnistaa ongelmia sekä keskustellaan mahdollisista muutoksista seuraavaa sprinttiä varten. Aika usein lopussa keskitytään enemmän myöhässä olevien kehitystehtävien priorisointiin kuin projektisuunnan uudelleensuunnitteluun.

3.3 Sprint Retrospektiivi

Schwaberin ja Sutherlandin mukaan: Sprint-retrospektiivin tarkoituksena on suunnitella tapoja parantaa laatua ja tehokkuutta (Schwaber & Sutherland 2020).

Scrum-tiimi tarkastaa, miten viimeinen sprintti sujui yksilöiden, vuorovaikutuksen, prosessien, työkalujen ja tiimin toteutuksen jälkeen. Scrum-tiimi keskustelee siitä, mikä sujui hyvin sprintin aikana, mitä ongelmia se koki ja miten nämä ongelmat ratkaistiin tai ei ratkaistu (Schwaber & Sutherland 2020).

3.4 Retrospektiivin vaiheet

Diana Larsen, Esther Derby ja Ken Schwaber ovat jakaneet kirjassaan retrospektiiviin viiteen kappaleeseen ja näistä jäsentyy tapahtumalle rakenne, joka on kuvattu alla olevassa kuvassa (Larsen, Derby & Schwaber 2006).



KUVA 8. Retrospektiivin rakenne

3.4.1 Viritä näyttämö

Näyttämön virittämisen tarkoitus on herätellä ihmisiä kyseiseen tapahtumaan. Ensimmäisessä vaiheessa on tarkoitus rikkoa jää jollain helpolla kysymyksellä tai tehtävällä. Tämä auttaa luomaan avoimen ilmapiirin, jossa ihmisten on helpompi keskustella asioista. Aloituskysymys kannattaa valita siten, että jokainen pääsee vastaamaan ja osallistumaan keskusteluun, sillä retrospektiivin tarkoituksena on auttaa ryhmää ajattelemaan ja oppimaan yhdessä (Larsen ym. 2006).

Seuraavaksi hahmotellaan istunnon kulku sekä osioiden ajallinen kesto. Aika on arvokasta, ja ihmiset haluavat tietää, että heidän aikansa kuluu hyvin.

Lähestymistavan tunteminen auttaa varmistamaan, ettei tapahtuma ole uusi päämäärätön tapaaminen. (Larsen ym. 2006).

Yleensä retrospektiiville on varattu aika jostain kohtaa keskeltä työpäivää, joten on oleellista kiinnittää huomiota siihen, että ihmiset herätellään tapahtumaan ja karistetaan ajatukset työtehtävistä. Tämän lisäksi on oleellista sopia, ettei esimerkiksi puhelimia ja kannettavia tietokoneita oteta mukaan tapahtumaan.

Eräässä hyvässä retrospektiivissa oli tapahtuman alun kulku seuravanlainen: Alussa käytiin retrospektiivin sisältö läpi aikatauluineen, jonka jälkeen jatkettiin avoimiin kysymyksiin. Kun kysymykset oli käyty läpi, siirryttiin ihmisten herättely vaiheeseen. Tässä ideana oli muodostaa jono siinä järjestyksessä, missä henkilö oli aloittanut yrityksen palveluksessa. Tiimissä aloitettiin keskustella ja järjestää jonoa tehtävän mukaan. Näin ollen tapahtuma alkoi rakentavasti yhteistyöllä ja retrospektiiviin saatiin oikea tunnelma.

3.4.2 Tietojen kerääminen

Larsenin ym. mukaan tietojen kerääminen iteraation jälkeen, jonka pituus on viikosta kahteen, voi tuntua turhalta. Kuitenkin jos joku on ollut poissa päivän tai kaksi, niin hän menettää 20–40 % tapahtumista ja vuorovaikutuksesta. Lisäksi on oleellista, että eri ihmisillä on erilaiset näkökulmat samaan tapahtumaan. Tietojen kerääminen luo yhteisen kuvan tapahtuneesta ja ilman yhteistä kuvaa ihmiset pyrkivät tarkistamaan omat mielipiteensä toisiltaan. Tietojen keräämisen idea on laajentaa kaikkien näkökulmaa. (Larsen ym. 2006).

Tunnin pituisen retrospektiivin aikana on hyvä saada ihmiset kertomaan asioista ja tapahtumista käyttäen apuna esimerkiksi fläppitaulua sekä suuria näkyviä kaavioita. Yleisesti visuaalinen kuvaus aikajanalla helpottaa ihmisten näkemyksiä tiedossa ja tai asioissa sekä auttaa yhteyksien luomisessa. Sen lisäksi on oleellista pitää mielessä, että tosiasiat ovat vain osa tiedoista ja tunteet ovat ainakin puolet tarinasta. Tunteet kertovat ihmisille tärkeistä asioista ja tiimin avoimuudesta sekä dynamiikasta. (Larsen ym. 2006).

3.4.3 Oivalluksien luominen

Retrospektiivin tässä vaiheessa on aika kysyä "Miksi?" ja alkaa miettiä, mitä tehdään toisin. Luodessaan oivalluksia tiimi ottaa huomioon tiedot edellisen kehityssyklin (iteraation) vahvuuksista ja heikkouksista ongelmien tunnistamiseksi. (Larsen ym. 2006).

Ihmisten on helppo siirtyä ratkaisuihin ongelmien ilmaantuessa. Ensimmäiset ratkaisut voivat olla oikeita, mutta useimmiten eivät. Tämän vaiheen tehtävä on tarkastella mahdollisuuksia, syitä ja seurauksia sekä miettiä niitä analyttisesti. Tästä löydetty oivallukset auttavat tiimiä näkemään, kuinka työskennellään tehokkaammin, mikä on minkä tahansa retrospektiivin perimmäinen tavoite. (Larsen ym. 2006).

3.4.4 Päätä mitä tehdään

Tässä vaiheessa tiimillä on luettelo mahdollisista uusista ideoista, kokeiluista ja tehtävistä. Tämän jälkeen valitaan äänestämällä suosituimmat kohteet. Äänestyksen lopputuloksesta valitaan enintään yksi tai kaksi toteutettavaa ehdotusta ja suunnitellaan, mitä näille tehdään. Fasilitaattorin eli tilaisuuden vetäjän ensisijainen tarkoitus on tarjota rakenteita ja ohjeita toimien suunnitteluun sekä auttaa tiimiä valitsemaan kohteet, joihin he voivat sitoutua. Ilman henkilökohtaista sitoutumista ihmiset olettavat helposti, että "tiimi" suorittaa tehtävän, eikä kukaan tee sitä. (Larsen ym. 2006).

3.4.5 Retrospektiivin lopettaminen

Kaikki hyvät asiat loppuvat aikanaan, jopa retrospektiivit. On tärkeää kertoa retrospektiivin lopussa, miten tapahtuma dokumentoidaan ja miten edetään jatkossa. Tässä on hyvä antaa arvostusta tiimille siitä kovasta työstä, jota kaikki tekivät sekä iteraation että retrospektiivin aikana. Lopuksi on hyvä pyytää tiimiltä palautetta retrospektiivin kulusta. Tällöin fasilitaattori tietää, mikä sujui hyvin ja mitä tarvitsee tehdä toisin seuraavassa tapahtumassa. Jatkuva parantaminen kuuluu olennaisena osana myös retrospektiiviin (Larsen ym. 2006).

4 OPPIMISEN TEORIA

Kappaleen teema on taustoittaa opinnäytetyössä käytettyä oppimiskahvilamenetelmää ja tuoda ilmi mistä menetelmä on saanut alkunsa.

Oppimiskahvilan tietoteoria perustuu Nonakan ja Takeuchin luomaan kaksiulotteiseen teoriaan. He määrittivät, että tiedon luomisen prosessi yrityksissä perustuu kahteen pääulottuvuuteen. Ensimmäinen on epistemologinen ulottuvuus: kriittinen oletus, että ihmisen tieto syntyy ja laajenee hiljaisen tiedon ja nimenomaisen tiedon sosiaalisen vuorovaikutuksen kautta ("tiedon muuntaminen") (Nonaka & Takeuchi, 1995, 59-60). Voidaan ajatella, että esimerkiksi kahvipöytäkeskusteluilla on iso merkitys hiljaisen tiedon siirtymiseen varsinkin siinä tapauksessa, kun yksilö nostaa esille työtehtävissä olevan haasteellisen ongelman johon kollega pystyy antamaan erilaisia ratkaisuvaihtoehtoja.

Toinen pääulottuvuus on ontologinen ulottuvuus, joka koskee tietämystä luovien yksiköiden tasoista, kuten yksilöllinen, ryhmä-, organisaatio- ja organisaatioiden välinen taso. (Nonaka & Takeuchi, 1995, 59-60). Tämä voidaan ajatella siten, että yksittäisen henkilön tai ryhmän tietämystä ja osaamista siirretään organisaation välillä esimerkiksi sisäisten koulutusten tai tietoiskujen avulla. Tieto siis alkaa yksilöstä ja organisaation tieto rakentuu sekä yksilön että yrityksen välisestä vuorovaikutuksesta ja ryhmällä on tärkeä rooli vuorovaikutuksen vaikuttamisessa.

4.1 Oppimiskahvila

World Café kommuuni -säätio, joka käyttää oppimiskahvilasta nimeä World café kuvaa oppimiskahvilamenetelmää seuraavasti: se on yksinkertainen, tehokas ja joustava muoto suurryhmäkeskustelujen järjestämiseen. Jokaisella menetelmän elementillä on erityinen tarkoitus ja se vastaa yhtä tai useampaa suunnitteluperiaatetta (World Cafe 2021).

Innokylän verkkosivustolla mainitaan, että menetelmässä on oleellista keskustelu, omien mielipiteiden kertominen ja yhteisen konsensuksen

löytäminen. Menetelmä ohjeistaa yhteisten ratkaisujen tekemiseen. Toisten mielipiteitä ja näkemyksiä on mahdollista kommentoida sekä kyseenalaistaa, mutta tärkeintä on ryhmän yhteisen mielipiteen löytäminen. (Innokylä 2021). Menetelmän helppous on se, että sitä voidaan muokata halutulla tavalla, kun suunnittelussa ja kysymyksissä otetaan huomioon asian konteksti, tarkoitus ja sijainti.

Oppimiskahvilamenetelmässä keskitytään joidenkin teemojen tai kysymysten ratkaisuun ryhmässä. Lähtökohtana on jakaantua pienryhmiin pöytäkunnittain. Prosessin edetessä pöytäryhmien jäsenet vaihtavat pöydästä toiseen, jolloin uusi pöydän ympärille muotoutunut ryhmä pohtii aiempia tuotoksia ja ideoi niitä eteenpäin. (Innokylä 2021).

Oppimiskahvilan prosessi etenee seuraavalla tavalla. Teemat ja kysymykset pöytäryhmille on määritellyt tapahtuman vetäjä. Vetäjän tehtävä on myös kertoa, mitä tehdään, mitä tavoitellaan ja mikä on keskustelun aikataulu. Vetäjän lisäksi jokaisesta ryhmästä valitaan kirjuri, joka huolehtii, että kaikkien ajatukset tulevat kirjattua ylös. Sopiva pöytäryhmän koko on noin neljästä kuuteen henkilöä. Ajatuskierroksen jälkeen tehdään yhteenveto ja esitellään tulokset kaikille osallistujille.

5 PROJEKTIN ALOITTAMISEN TUTKIMINEN KOHDEYRITYKSESSÄ

Opinnäytetyön aihetta kartoitettaessa käytiin keskusteluja eri vaihtoehtoista yritystä X Oy:n johdon kanssa. Eräs vaihtoehto olisi ollut kehittää osaamisen siirtämistä henkilöstön välillä, mutta ensisijaisena kehityskohteena nähtiin projektien yhtenäisten käytäntöjen luominen, koska yritys X Oy:lle projektit ovat tärkeässä asemassa niin tulonlähteen kuin referenssien osalta.

5.1 Nykytilanteen kartoitus

Opinnäytetyöntekijä on pitkän työhistoriansa aikana huomannut, että projektien toteutus alkaa useimmiten siten, että tehdään sovelluksen runko ja lisätään se versionhallintaan. Yhteisiä toimintatapoja ei ole yleensä käyty järjestelmällisesti läpi ennen projektin alkua. Osaltaan tähän varmasti vaikuttaa se, että yrityksen työntekijät ovat kokeneita ohjelmistoalan tekijöitä ja osaavat adaptoitua projektissa käynnissä oleviin toimintamalleihin. Tämä ei kuitenkaan poista sitä tosiasiaa, että järjestelmälliselle toimintatavalle ei olisi tarvetta. Tämän lisäksi projektin sisällä olevat käytännöt saattavat vaihdella laidasta laitaan, mikä on ymmärrettävää eri kokoisten projektien ja eri asiakkaiden johdosta. On kuitenkin selvää, että eri projektien välillä on yhteisiä piirteitä, joista saadaan rakennettua yhteinen käytäntö.

Retrospektiiviin kutsu lähetettiin yritys X Oy:n koko henkilöstölle ja siihen toivottiin vähintään kymmentä osallistujaa. Ennen kutsujen lähettämistä kirjoitettiin yritys X Oy:n Slack-kanavalle pieni tietoisuus siitä, mitä tapahtumassa käsitellään, mikä sen tarkoitus on ja mitä tuloksia siitä halutaan. Tuleva retrospektiivi synnytti kahvipöytäkeskusteluja jo etukäteen, mikä osaltaan kertoi, että aihe kiinnosti henkilöstöä. Kutsussa toivottiin, että ihmiset tulevat tapahtumaan avoimin mielin. Retrospektiivi toteutettiin tammikuussa 2020.

5.2 Esivalmistelut

Opinnäytetyöntekijä oli kysellyt ja kartoittanut retrospektiivin pitämiseen liittyviä ohjeita asiakasprojektissa monia retrospektiivejä fasilitoineelta kollegalta. Keskustelun perusteella kolme tärkeintä ohjetta tapahtuman läpiviemiseen olivat:

1) rento ja jämpä ote 2) harjoitusten kellottaminen aikataulussa pysymiseksi 3) sopivien osien valitseminen kyseiseen tapahtumaan. Lisäksi tuli ilmi, että tapahtumaan sopivan variantin voi rakentaa sivustolta www.retromat.org. Opinnäytetyöntekijä valitsi sivustolta löytyvän suosituksen aloittelijalle, koska aikaisempaa kokemusta ei ollut retrospektiivin vetämisestä.

5.3 Retrospektiivin sisältö

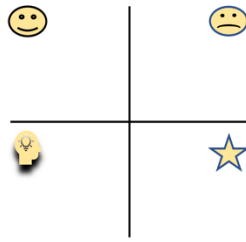
Corinna Baldauf on luonut blogisivun retromat.org:n, joka kuvaa aloittelijan retrospektiivinä. Ensimmäisen harjoituksen tarkoituksena on antaa virikkeitä ja herättää retrospektiiviin. Harjoite on nimeltään ”Positiivista ja totta” ja sen ydin on luoda positiivista energiaa tapahtumaan ja saada jokainen osallistumaan sekä kuulluksi. Toteutus tapahtuu siten, että vieruskaveri kysyy ennalta räätälöidyn kysymyksen, johon saadaan positiivinen vastaus. (Baldauf, C 2018)

- *Mitä olet tehnyt todella hyvin viimeisessä projektissa?*
- *Mikä on jotain sellaista, joka tekee sinut todella onnelliseksi?*
- *Mitä hienoa teit jollekin toiselle edellisessä projektissa?*

Tätä jatketaan niin kauan, kunnes kaikki ovat vastanneet ja kysyneet. Harjoitus antaa kaikille vauhtia ja positiivista mieltä sekä johtaa parempiin tuloksiin (Baldauf, C 2018).

Vaiheessa kaksi yhdistetään tietojen kerääminen ja luo näkemyksiä -harjoitukset yhdeksi. Retromat -verkkosivustolla kuvataan lean coffee -harjoituksen sisällöstä näin: listataan yksi aihe yhdelle Post-it lapulle ja kuvataan aiheen sisältöä muutamalla lauseella. Jokainen osallistuja vie vuorotellen lappunsa näkyviin seinälle ja yhdistää mielestään samaa aihealuetta käsittelevät aiheet (post it - laput) yhteen sarakkeeseen. Tämän jälkeen jokainen osallistuja äänestää kahta aihetta, josta he haluavat keskustella. (Baldauf, C 2018)

Learn matrix -harjoitus koostuu neljästä kategoriasta, jotka on jaettu taulukkoon alla olevan kuvan mukaisesti.



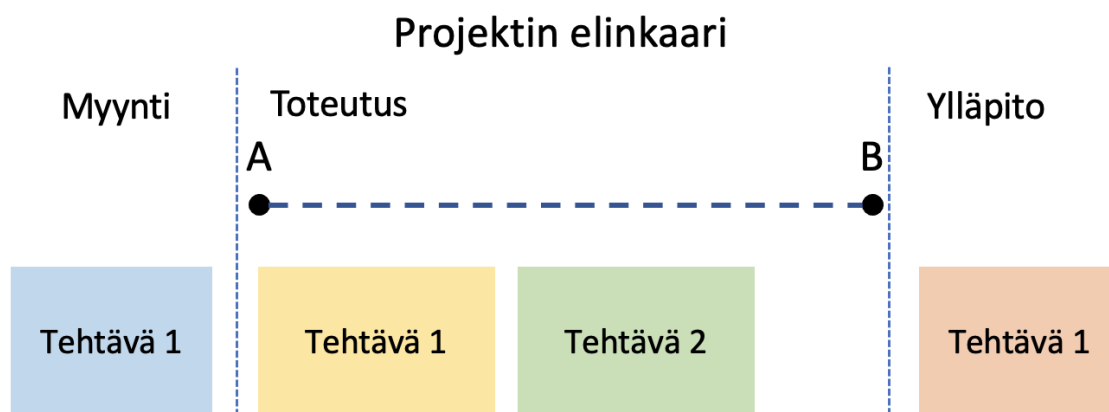
KUVA 9. Learn matrix -taulukko

Learn matrix -taulukon idea on tuoda esille asioita 1) missä onnistuttiin 2) mikä meni pieleen 3) ideoita 4) tähtipelaaja. Tiimin jäsenet kirjoittavat aiheen per lappu ja koittavat tuottaa mahdollisimman monta lappua määrättyssä aikarajassa. Aiheet yhdistellään ja käydään läpi harjoituksessa. Tämä vaihe aikataulutetaan viidestä kymmeneen minuuttiin.

5.4 Retrospektiivin toteutus

Alkukartoitus toteutettiin retrospektiivilla, johon osallistui yritys X Oy:stä kymmenen henkilöä, joilla on kokemusta ohjelmistoalalta yli kymmenen vuoden ajalta. Kaikilla oli kokemusta ohjelmistokehityksestä ja -projekteista sekä muutamilla henkilöillä lisäksi myynnistä ja palveluiden konseptoinnista. Henkilöiden osaaminen ja kokemus antaa varsin laajan näkemyksen erilaisista ohjelmistoprojekteista.

Toteutus tapahtui hieman muunnetusti edellä kuvatusta retrospektiivistä. Tarkoitus oli aloittaa tapahtuman näyttämön virittäminen harjoitteella ”positiivista ja totta”, mutta tiimi lähti välittömästi kyselemään tarkemmin retrospektiivin osioista. Näin ollen kerrottiin, että tarkoitus on tunnistaa yleisiä tehtäviä, joita projektin aloitusvaiheessa tarvitsee tehdä riippumatta projektin elinkaaren vaiheista. Kuva 10 havainnollistaa tehtävän alustamista. Tämän lisäksi jokaisen tarvitsi itsenäisesti miettiä eri näkökulmista (tiiminvetäjä, ohjelmistokehittäjä, myynti) näitä tehtäviä.



Kuva 10. Tehtävien tunnistaminen ennen projektia, sen aikana sekä jälkeen

Tehtävän määrittelyä kerrottaessa osa tiimin jäsenistä alkoi täyttää Post-it lappuja ja osa tiimistä jatkoi vielä tarkentavilla kysymyksillä, joihin vastattiin parhaalla mahdollisella tavalla. Näin ollen retrospektiiviin käynnistyminen tapahtui eri tavalla kuin oli suunniteltu. Koska oli päästy vauhtiin, laitettiin kello käyntiin ja annettiin tiimin käyttää tehtävään viisi minuuttia. Harjoituksen jälkeen tiimi jaettiin kolmen henkilön ryhmiin, joissa koostettiin yhteisiä kokonaisuuksia tunnistetuista asioista. Koostamiseen käytettiin aikaa viisi minuuttia.

Viimeisenä vaiheena jokainen ryhmä esitteli löydöksensä ja nämä koostettiin yhteiselle alueelle seinälle kahden fläppitaulun kokoiselle paperiarkille ja jokainen ryhmä ryhmitteli samankaltaiset tulokset. Liitteessä neljä on kuvattuna toteutuksen lopputulos. Tästä saadun datan perusteella on tarkoitus jalostaa yrityksen intran sivulle kattava käytäntö löydetyistä tehtävistä ja organisoida ne loogiseen järjestykseen ketterän ohjelmistoprojektin aloittamisen tarkistuslistaksi.

Retrospektiivin toinen osa käsitteli ohjelmistokehityksen käytäntöjä. Toisessa osassa käytettiin oppimiskahvilaa ja aihealueiksi oli valittu seuraavat teemat:

- 1) Mitkä ovat ohjelmistoprojektin hyvät käytännöt?
- 2) Miksi projekti epäonnistuu? (Miksi menee pieleen?)

Harjoituksen alussa tiimi jaettiin kahteen viiden henkilön ryhmään, joista toinen ryhmä keskusteli kysymyksestä numero yksi ja toinen kysymyksestä numero

kaksi. Tehtävän kesto oli kymmenen minuuttia. Tämän jälkeen tiimit vaihtoivat puolia ja keskustelua jatkettiin kymmenen minuutin ajan puitteissa. Tiimit löysivät monia hyviä tekijöitä molempien kohtien alle. Tämän jälkeen jatkettiin siten, että tiimien jäsenet pisteyttivät mielestään tärkeimmät tekijät tärkeysjärjestykseen pisteäänestyksellä.

Jokaisella tiimin jäsenellä oli mahdollisuus antaa pisteitä yhdestä kolmeen Post-it -lapulla olevan asian tärkeyden osalta. Näin ollen löydettiin tärkeimmät käytännöt, jotka tiimi kokee toimivaksi ja mitkä tarvitsevat eniten kehittämistä.

Pisteytyksen jälkeen otettiin toisesta kategoriasta kolme tärkeintä ja toisesta neljä eniten pistettä saanutta Post-it lappua ja pohdittiin ryhmissä konkreettisia toimia, miten hyvät käytännöt pysyvät jatkossa yrityksessä käytössä ja miten hidastavia tekijöitä ennaltaehkäistään. Tämä tehtävä oli ajastettu viiden minuutin aikaikkunaan. Tehtävän suorituksen aikana molemmat ryhmät löysivät monia toimenpiteitä näiden seitsemän käytännön parantamiseksi.

Viimeisenä tehtävänä oli kerätä tiimiltä palautetta retrospektiivistä ja sen fasilitoinnista. Oveen oli järjestetty arvosteluasteikko yhdestä viiteen. Jokainen tiimin jäsen sai kirjoittaa palautteen lappuun ja lähtiessään asettaa sen haluamansa numeron kohdalle. Asteikon kuva löytyy liitteestä x

Palautteen keskiarvoksi tuli hieman päälle kolme, mikä on hyvä arvosana, kun ottaa huomioon, että tämä oli opinnäytetyöntekijälle ensimmäinen retrospektiivi fasilitaattorin roolissa. Lisäksi palautteesta kävi ilmi, että tiimi piti aihetta tärkeänä ja löydettiin hyviä kehityskohteita. Retrospektiivin pohjustaminen olisi tarvinnut enemmän aikaa ja tehtävien määrittäminen olisi voinut olla vielä selkeämpää.

6 TULOKSET

6.1 Tiedon analysointi

Retrospektiivistä kerätystä tutkimusaineistosta rakennettiin kolme käsitekarttaa. Ensimmäinen liittyi projektin tarkistuslistaan lisättäviin tehtäviin ja osittain sivuaa päättötyössä esitettyjä tutkimuskysymyksiä. Kaksi seuraavaa käsitekarttaa vastaavat suoraan opinnäytetyön tutkimuskysymyksiin. Käsitekartat koostettiin yhteen teemoittelulla. Teemoittelun tavoitteena on löytää ja analysoida aineistosta yhteneviä tekijöitä ja teemoja sekä organisoida niitä. Käsitekartassa korostuivat tietyt teemat, jotka nousivat selkeästi esiin. Käsitekarttaan haluttiin sisällyttää tutkimusaineistossa löytyneet tekijät, joten teemoittelun myötä tärkeimmät teemat onnistuttiin löytämään. Retrospektiivin tulokset on esitelty liitteiden kaksi ja kolme käsitekartoissa.

Retrospektiivin jälkeen sovittiin aika maaliskuulle, johon mennessä kerätty data on analysoitu ja koostettu esitettävään muotoon sekä valmiina esitettäväksi yrityksen viikkopalaverissa. Datan analysointi tapahtui aikataulussa ja esitys toteutettiin 3/2020.

6.2 Tarkistuslistan muodostaminen projektin aloittamiseen

Tarkistuslistan laatiminen aloitettiin liitteen numero yksi käsitekartan pohjalta. Käsitekartassa on painoarvot jokaiselle vastaukselle. Painoarvo on kuvattu käsitekartassa numerolla, joka on ympyrän sisällä. Käsitekartan painoarvoilla ei listan kannalta ole merkitystä, koska tärkeämpää on, että asia on sisällytetty listaan. Painoarvo kuvaa enemmän sitä, kuinka monen henkilön kannalta tietty asia nousi esille ja kuinka moni oli sitä mieltä, että asia on syytä käydä läpi ennen projektin aloittamista.

Uuden projektin käynnistyksen tai henkilömuutoksen yhteydessä pidetään projektille kick off -tapahtuma asiakkaan kanssa sekä sisäisesti. Tämä siitä johtuen, että mikäli kaikki eivät pääse osallistumaan asiakaspalaveriin niin jokaiselle a) tulee yhteinen käsitys projektiin liittyvistä käytännöistä ja b)

saadaan projektille yritys X Oy:n intraan eli sisäiselle verkkosivustolle sivu, jossa tieto on koko henkilöstön saatavilla.

Käsitekartan pohjalta, liite numero yksi, rakennettiin tarkistuslista. Tarkistuslista pitää sisällään seuraavat asiat:

- Vastuuhenkilöt
- Asiakkaan vastuuhenkilöt
- Aikataulu
- Kommunikaatiokanavat
- Tarvittavat tunnukset
- Työskentelymallin sopiminen
- Lopputestaus
- Projektinhallinta työkalut
- Käyttöliittymän suunnittelu - design
- Arkkitehtuurista päättäminen
- CI / CD – Jatkuva integrointi ja jatkuva julkaiseminen
- Dokumentaatio
- Retrospektiivi asiakkaan kanssa

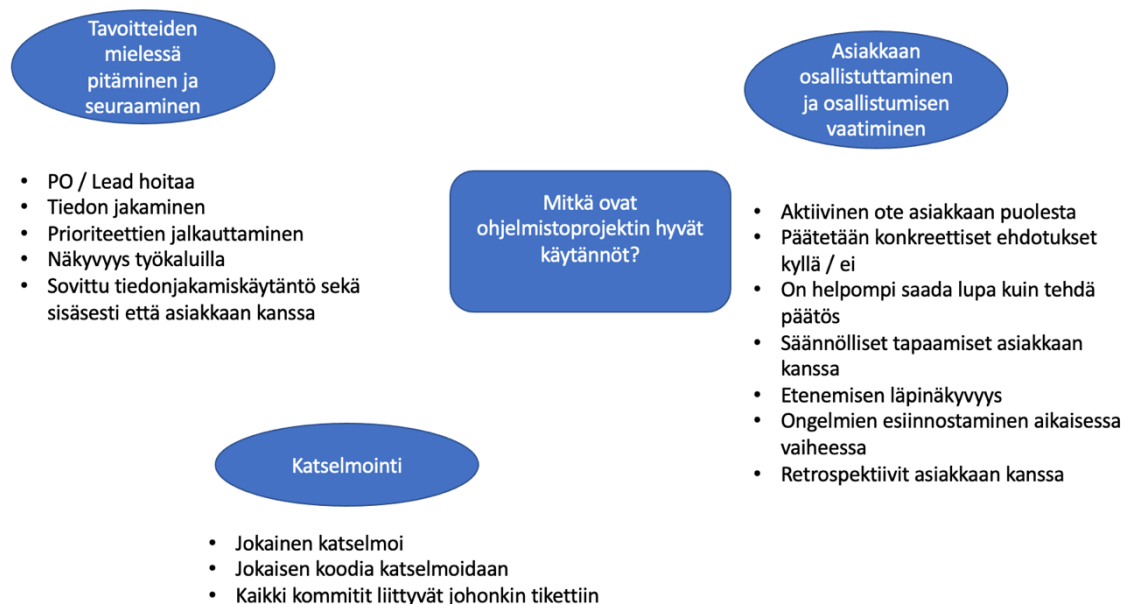
Vastuuhenkilöt kohdassa kirjataan projektinvetäjä ja esitellään tiimi sekä tiimihenkilöiden vastuualueet. Asiakkaan vastuuhenkilöt sisältävät asiakkaan kontaktihenkilöt ja mahdolliset kolmannen osapuolen kontaktihenkilöt sekä vastuualueet. Aikataulu osiossa määritellään ohjelmiston demonstraatiot asiakkaalle ja päivä- sekä viikkopalaverien ajankohdat. Kommunikaatiokanavat määrittävät käytettyjen viestisovelluksien käyttämisen niin sisäisesti kuin asiakkaan kanssa. Tarvittavien tunnuksien kohdassa käydään läpi sovelluksen julkaisemiseen liittyvät toimenpiteet ja asiakaspuolelta tarvittavat oikeudet. Työskentelymalli määrittää yhteiset käytännöt versionhallinnan käyttämiseen ja sovelluskoodin katselmointiin sekä sovelluksien julkaisemiseen. Projektinhallintatyökalut päätetään projektille käytettävät työkalut. Käyttöliittymän suunnittelu kohdassa -osiossa tarkastetaan, että sisältyykö suunnittelu asiakkaalle vai yritys X Oy:lle ja määritetään käytettävät työkalut sekä kirjataan käyttöliittymän suunnittelulle vastuuhenkilö. Arkkitehtuurista päättäminen kohdassa määritellään sovelluksien rakenne ja tarvittavat

kolmannen osapuolen komponentit. CI / CD vaiheessa päätetään vastuhenkilö sovelluksien julkaisuputken rakentamisesta joko sisäisesti tai ulkoisesti. Dokumentaatio kohdassa kirjataan ohjeet, jotka tarvitaan projektissa sisäisesti. Asiakasretrospektiivi vaiheessa kirjataan alustavat päivämäärät näille tapahtumille. Mikäli osa listan asioista jää avoimeksi, ne kirjataan projektitehtäviksi alkuun, jotka projektipäällikön tai tiiminvetäjän tulee hoitaa kuntoon.

6.3 Mitkä ovat projektien hyvät käytännöt?

Käsittekartan, liite numero kaksi analysoinnin perusteella piirrettiin kuva 11, joka ilmentää tiimin mielestä tärkeimpiä asioita, jotka vaikuttavat projektin onnistumiseen:

- 1) Tavoitteiden mielessä pitäminen ja seuraaminen
- 2) Asiakkaan osallistuttaminen ja osallistumisen vaatiminen
- 3) Katselmointi



KUVA 11. Ohjelmistokehityksen hyviä käytäntöjä

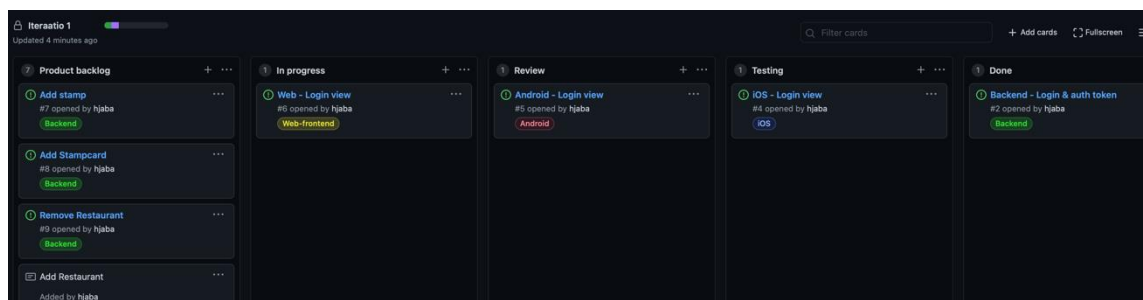
Seuraavissa alaotsikoissa käydään tarkemmin läpi tuloksia ja peilataan niitä vastaavan teoriaan.

6.3.1 Tavoitteiden mielessä pitäminen ja seuraaminen

Tavoitteet ohjaavat projektin tekemistä ja tavoitteiden täytyy olla saavutettavissa. Onnistuneen projektin kannalta määrätyt tavoitteet tulee olla selkeänä mielessä koko tiimille projektin koko elinkaaren ajan. Epäselvästi määritetyt ja tai väärinymmärretyt tavoitteet vaikuttavat siihen, että yksittäiset asiat alkavat helposti mennä pieleen. Ensimmäisenä ja tärkeimpänä tekijänä käsitekartasta nousi esille projektin tavoitteiden mielessä pitäminen ja niiden seuraaminen.

Aleksi Pulkkasen verkkoartikkelin mukaan, että SMART-tavoitteet ovat suosittuja. Nämä viittaavat tarkkoihin (specific), mitattaviin (measureable), jaettuihin (agreed upon), realistisiin (realistic) ja aikaan sidottuihin (time-based) tavoitteisiin. Lisäksi hän täsmentää, että tavoitteet ohjaavat tekemistä ja ne täytyy olla saavutettavissa. (Aleksi Pulkkanen 2020).

Yritys X Oy:ssä on onnistuttu pitämään eri projektien tavoitteet selkeinä Scrum käytäntöjen eli päivittäisten ja viikkopalaverien sekä projektissa käytettyjen projektihallinta työkalujen avulla. Projektihallintatyökalut vastaavat suoraan edellä kuvattuun SMART- tavoitteisiin. Kuva 12 havainnollistaa yleistä näkymää eräästä käytössä olevasta projektin hallintatyökalusta. Kuvassa yksittäinen tehtävä vastaa suoraan Smart-tavoitteeseen.



KUVA 12. Esimerkki Github projektihallintanäkymä

Projektinhallintanäkymä antaa kokonaiskuvan iteraatiossa olevan tuotteen kehitysjonosta (sarake 1), käynnissä olevasta kehityksestä (sarake 2), katselmoinnissa olevasta toteutuksesta (sarake 3), testauksessa olevasta toiminnallisuudesta (sarake 4) ja valmiista toiminnallisuudesta (sarake 5).

Joidenkin pienten projektien kohdalla keskusteluissa huomattiin, että vastuualueiden selkeä määrittely tarvitsee kirkastamista projektin alussa. Lisäksi keskusteltiin siitä, miten kehittäjä keskittyy tekemään oleellista toiminnallisuutta ilman, että toiminnallisuutta hiotaan loputtomasti tai miten päätetään kolmannen osapuolen ohjelmistokirjaston käyttämisen etu projektin ollessa käynnissä.

Edellä mainittujen epäkohtien ratkaisemiseksi päätettiin, että nimetään vastuuhenkilö kick off -palaverin tarkistuslistalle. Tämän lisäksi tarkistuslistan kautta määritellään projektissa käytettävät tiedon jakamiskäytännöt. Projektin vastuuhenkilö vastaa tehtävien prioriteettien jalkauttamisesta ja käyttämisestä projektiin työkalujen avulla tai sovituissa päiväpalavereiden käytännöissä.

6.3.2 Asiakkaan osallistuttaminen ja osallistumisen vaatiminen

Asiakkaan osallistuttaminen on tärkeä osa projektin onnistumisen kannalta. Asiakkaan on mahdollista omalla toiminnallaan vaikuttaa projektin lopputulokseen. Hyvällä asiakasyhteistyöllä varmistetaan projektin muutoshallinnan ja -lopputuloksen onnistuminen sekä asiakastyytyväisyys.

Toisena tekijänä nähtiin asiakkaan osallistuttaminen ja osallistumisen vaatiminen. Hanna Ahtolan verkkojulkaisussa kiteytetään osallistuttamisen hyödyt näin:

- Asiakkaan ja palvelun välille syntyy yleensä erityinen side
- Osallistumalla asiakas pääsee vaikuttamaan ja kehittämään
- Osallistuminen usein selkeyttää myös asiakkaan roolia palvelussa

(Hanna Ahtola 2018).

Suomidigi julkaisussa kiteytetään seuraavasti: keskeinen osa asiakaslähtöistä digitaalista palveluntarjontaa sekä sen suunnittelua ja toteuttamista on

asiakstarpeen ymmärtäminen sekä tarpeeseen vastaaminen. Tämän lisäksi asiakaslähtöisyys tulee olla palvelukehittämisen ohjaavana voimana (Suomidigi 5 2020).

Yrityksen strategia sisältää ihmisläheisen ohjelmistokehityksen, joka on onnistuttu jalkauttamaan yrityksen henkilöstöön. Henkilöstö on motivoitunut haastamaan asiakasta ja haluaa, että asiakas on mukana ohjelmistokehityksessä. Asiakasosallistuttamisen kulmakivet yrityksessä ovat:

- Scrum-mallin mukaiset päivä- ja viikkopalaverit
- Kommunikaatiokanavista sopiminen
- Demonstraatiot eri osista

Yrityksessä on tullut muutamissa projekteissa esiin tilanteita, joissa asiakkaan yhteyshenkilöä on ollut haasteellista saada osallistumaan palavereihin tai asiakasretrospektiiviin, ja tämä koettiin henkilöstön kannalta turhauttavaksi. Tästä johtuen keskusteltiin siitä, miten ottaa vielä aktiivisempi ote asiakkaan puolesta. Tiimin ajatukset kursivoituna:

- *Päätetään konkreettiset ehdotukset kyllä ja ei*
Tehdään selkeitä suljettuja kysymyksiä asiakkaalle.
- *On helpompi saada lupa kuin tehdä päätös*
Useimmin päätöksiin sisältyy byrokratiaa ja sidosryhmien kanssa koordinoitua, joten luvan saaminen on helpompaa ja tehokkaampaa kuin kokonaan uuden päätösprosessin läpivienti erinäisissä sidosryhmien kokouksissa.
- *Etenemisen läpinäkyvyys*
Tehdään säännöllisiä demonstraatioita kaikissa projekteissa ja otetaan asiakasretrospektiivit myös käyttöön.
- *Ongelmien esiin nostaminen aikaisessa vaiheessa*
Kirjataan ongelmat ylös ja seurataan palavereissa tai jaetaan ja sidosryhmien kesken välittömästi.

6.3.3 Katselmointi

Projekteissa katselmointi toimii laadun varmistamisena ja tuotetun projektin kokonaisuuden tai sen osan hyväksymisenä. Katselmointia voidaan pitää yhtenä kulmakivenä onnistuneen projektin maaliin saattamisessa

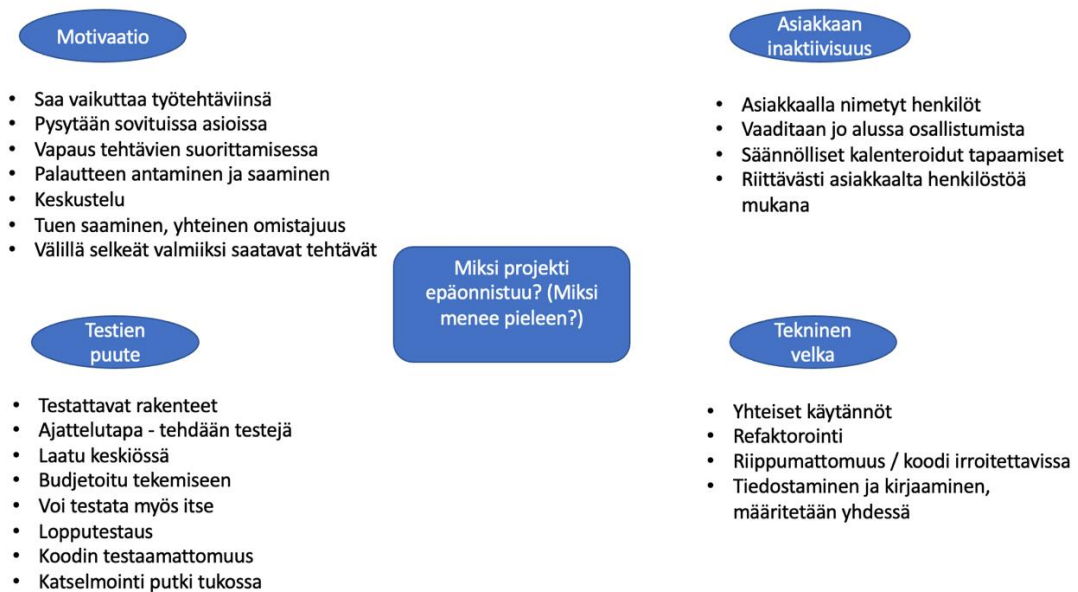
Kolmanneksi tärkeimpänä asiana tiimillä nousi katselmointi. Katselmointi on erittäin tehokas ja olennainen osa ohjelmiston laadunvarmistusprosessia. Katselmoinnilla varmistetaan, että koodi noudattaa sovittua tyyliä, koodi on ylläpidettävää ja siitä on huomattavissa mahdolliset ongelmat. Panu Leppäniemi kirjoittaa verkkosivullaan seuraavasti: Katselmointi on hyvä tapa oppia ja sillä pystytään yhtenäistämään käytännöt nopeasti. Katselmointi on erityisen hyödyllistä varsinkin, kun projekti alkaa uudella ryhmällä. (Panu Leppäniemi 2020). Toimeksiannon kohteena olevassa yrityksessä katselmointi hoidettiin niin, että toinen sovelluskehittäjä katselmoi kirjoitetun sovelluskoodin ja testasi koodin toimivuuden.

Yritys X Oy:ssä on kautta linjan ollut isoissa ja keskisuurissa projekteissa katselmointikäytäntö mukana ohjelmistokehityksessä, koska tiimien suuremman koon vuoksi resurssit ovat mahdollistaneet tämän. Pienemmissä yhden tai kahden henkilön projekteissa koodikatselmoinnin käytännön toteutuminen on ollut hieman puutteellista. Kehitysehdotuksena tämän puutteen korjaamiseksi ehdotettiin projektille mentori eli tukihenkilöä, joka hoitaisi katselmoinnin. Toisena vaihtoehtoina mainittiin myös yhden henkilön projektien tiimiyttäminen eli yhdistetään yhden ja kahden henkilön projekteissa toimivat henkilöt isommaksi yhdeksi tiimiksi, jolloin resursseja riittää koodin katselmointia varten.

6.4 Miksi projekti epäonnistuu?

Käsittekartan, liite numero 3 analysoinnin perusteella luotiin kuva numero 13, joka kertoo tiimin näkemykset projektin epäonnistumiselle:

1. Motivaatio
2. Asiakkaan inaktiivisuus
3. Testien puute
4. Tekninen velka



Kuva 13. Miksi menee pieleen?

6.4.1 Motivaatio

Henkilön motivaatio vaikuttaa olennaisesti projektin epäonnistumiseen varsinkin silloin, kun kyseessä on pienemmät projektit. Yksinkertaistettuna esimerkkinä voidaan ajatella, että valmis projekti on yhtä kuin sata prosenttia ja jokaisen tiimin jäsenen prosentuaalinen työpanos saadaan, kun sata jaetaan tiimin henkilölukumäärällä. Tällöin viiden henkilön projekteissa yhden henkilön työpanos vaikuttaa 20% lopputulokseen. Loppuun vielä mainittakoon, että tämä oli vain karkea laskennallinen ja havainnollistava esimerkki, mikä vaikutus motivaatiolla voi olla. On myös muistettava, että projektityöntekijöiden työpanokset eivät välttämättä ole keskenään yhtä suuret, jolloin yhden henkilön motivaatiolla voi olla pienempi tai suurempi vaikutus kuin toisella projektityöntekijällä.

Tiimin mielipide oli, että projektihenkilöiden motivaatio on merkittävin tekijä projektin epäonnistumiselle. Motivaation osuus korostuu, sillä projektien yleisin koko yrityksessä on yhdestä viiteen henkilöä. Mitä pienempi projektitiimi on kyseessä, sitä suhteellisesti suurempi vaikutus yksittäisen henkilön motivaatiolla on. Isoissa projekteissa muu tiimi pystyy helpommin kompensoimaan yhden henkilön motivaation puutetta, mutta pienemmässä ryhmässä tämä on vaikeampaa.

Motivaation vaikuttavia tekijöitä tiimi listasi seuraavasti:

- Vaikuttaminen omaan työtehtävään
- Vapaus tehtävien tekemisessä
- Pysytään sovituissa asioissa
- Palautteen antaminen ja saaminen
- Keskustelu
- Tuen saaminen, yhteinen omistajuus
- Välillä selkeät valmiiksi saatavat tehtävät

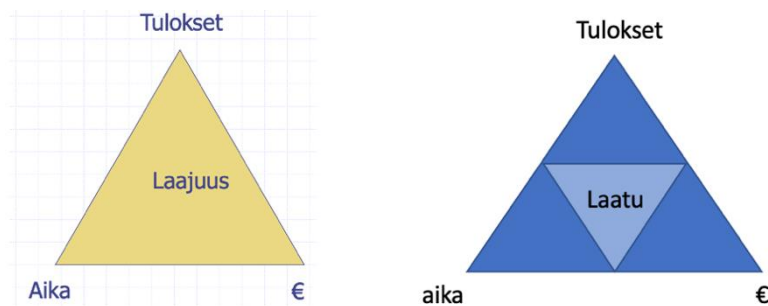
Projektin toteuttamisen kannalta henkilön on mahdollista vaikuttaa omaan tekemiseen kuten työaikaan ja toteutuksen sisältöön. Vapaus työn tekemisessä onkin ollut alusta lähtien osa yrityksen käytäntöjä. Tämä näkyy siten, että tiimi saa kehittää projektin sisäisiä työtapoja ja tehtävien toteuttaminen voi tapahtua henkilön oman päivärytmin mukaan. Ossi Aura on kirjoittanut blogitekstissään henkilöstökyselystä, jossa kartoitettiin työhön vaikuttamisen / kehittämisen mahdollisuutta, ammattitaidon kokemusta ja työn innostavuutta eli motivaatiota. Tulosten perusteella motivaatio on erinomainen 55 %:lla ryhmästä, joka koki vaikutusmahdollisuuksien olevan erinomaiset. (Ossi Aura 2020)

Tekemisen osalta nähtiin, että motivaatio hiipuu, mikäli toteutettavat tehtävät ovat liian isoja. Tällöin tekemiseen turhautuu ja tuntuu ettei projekti etene. Tämän lisäksi oleellisena seikkana nähtiin, että tehdään sovittuja asioita eikä viilata toiminnallisuutta loputtomiin. Parannettavaa motivaation kohottamisen osalta nähtiin palautteen, keskustelun, tuen ja yhteisen omistajuuden puutteissa. Palautteen osalta kehitettiin kiitoskaappi, josta voidaan jakaa pieniä kiitollisuudenosoituksia (elokuvalippu, smoothielippu, suklaapatukka) kollegoille.

6.4.2 Testien puute

Projektin onnistumista mitataan usein rahan, aikataulun ja tuloksien perusteella. Mäkilouko esitti kurssillaan projektikolmio -käsitteen, jossa projektin laajuus sisältyy kolmion keskelle ja kolmion kärjillä kuvataan tuloksia, aikaa ja rahaa. Kuva 14 olevassa kuvassa vasemmalla oleva kolmio. Oikealla puolella kuvatussa

kolmiossa keskiöön on laajuuden sijasta nostettu projektin laatu. Näin ollen voidaan kysyä, onko projekti onnistunut, mikäli se on pysynyt sekä aikataulussa että budjetissa ja täyttää sisältövaatimukset, mutta on laadullisesti huono eli sovellus sisältää huomattavia virheitä ja on käyttäjän kannalta hankala käyttää. Ohjelmistoprojekteissa tämä näkyy siten, että loppukäyttäjät eivät halua käyttää tuotetta tai palvelua. Jos projektin aikataulu on venynyt, mutta se muuten täyttää kriteerit ja etenkin laatu- ja laajuuskulman (tuotettu palvelu on käyttökelpoinen), projekti voidaan silti nähdä onnistuneena



Kuva 14 projektikolmiot

Ohjelmistoprojektin olennaisena laatuksena on sovelluksen tai palvelun testaaminen. Testaamisella yritetään löytää ohjelmistosta siellä vielä olevia virheitä tai osoittaa, että ohjelmisto täyttää jo sille asetetut vaatimukset ja se on riittävän toimiva käytettäväksi. Testaaminen voidaan jakaa moneen eri osaluokkaan: yksikkö-, integraatio-, järjestelmä-, suorituskyky-, käytettävyys-, ja hyväksyntätestaukseen.

Toisena merkittävänä tekijänä projektin epäonnistumiselle nähtiin testien puute. Tämä näkyy etenkin pienemmissä projekteissa siten, ettei testaamista olla budjetoitu mukaan tekemiseen. Kehitysehdotuksena testaamisen parantamiseen tiimi listasi 1) ajatustavan muutoksen testaamisen suhteen 2) tehdään testattavia rakenteita 3) testataan myös itse. Testaamisen kehittämisen ratkaisuksi suunniteltiin tiimien rakentamista pienten projektien osalta, jolloin tiimin sisällä on resursseja testaamisen suorittamiseen. Lisäksi pienten projektien kohdalla tämä huomioidaan jatkossa projektien budjetoinnissa.

6.4.3 Asiakkaan inaktiivisuus

Asiakkaan inaktiivisuuden eli passivisuuden vaikutus näkyy eritoten silloin, kun projektiin liittyvissä kriittisissä päätöksissä ei saada asiakkaalta vastausta. Tällöin projekti siirtyy tyhjäkäyntitilaan ja asiat eivät etene tavoitelulla tavalla.

Kolmantena tärkeänä asiana nousi esille asiakkaan inaktiivisuus. Vaikka yleisesti oli koettu, että asiakas saadaan osallistutettua projekteihin mukaan, oli joissakin projekteissa haasteita saada asiakaspuolelta henkilöä osallistumaan kehitykseen, palavereihin ja retrospektiiviin. Selkeitä kehitysedotuksia tähän olivat

- 1) Asiakkaalta nimetyt ja useammat henkilöt mukaan projektiin.
- 2) Vaaditaan asiakasta alusta lähtien mukaan ja sovitaan säännölliset tapaamiset.

Useamman henkilön kautta on mahdollista välttää sudenkuopat siinä vaiheessa, jos asiakasrajapinnan kontaktihenkilön motivaatio ja tai osaaminen ei ole vaadittavalla tasolla. Täten projektia voidaan viedä sujuvasti eteenpäin. Sovitaan proaktiivisesti itse asiakkaan ja asiakkaan sidosryhmien kanssa säännölliset tapaamiset.

6.4.4 Tekninen velka

Ward Cunnighamin artikkelissa esiintyy ensimmäistä kertaa tekninen velka kielikuva. Tämä tarkoittaa viimeisteleämätöntä nopeasti tuotettua koodia, joka toimitetaan tuotantoon. (Ward Cunningham 1992). Nykyään termin sisälle voidaan sisältää myös 1) ratkaisut, jotka ohjelmoija valitsee toteutusta tehdessä, 2) vanhentuneet ohjelmistokirjastot, jotka on otettu projektiin käyttöön ja 3) ohjelmiston suunnittelumallit, jotka ovat saattaneet muuttua ohjelmiston elinkaaren aikana.

Tekninen velka vaikuttaa suoraan ohjelmistototeutuspuoleen, mutta sillä voi olla haitallisia vaikutuksia myös projektin liiketoiminnan puolelle. Sillä on vaikutusta sovelluksen vakauteen, muokattavuuteen, suorituskykyyn, tietoturvaan ja uusien vikojen todennäköiseen syntymiseen. Tekninen velka korostuu siinä vaiheessa, kun on tarve saada nopealla aikataululla toiminnallisuus julkaistuksi

tai projekti on kestänyt pitkään ja se sisältää kolmannen osapuolen ohjelmistokirjastoja, joita ei ole päivitetty ajan tasalle. Yleensä tekninen velka jää huomioimatta pitkäkestoisissa projekteissa eikä sen kitkemiseen ole varattu budjettia, koska olennaisempaa on saada loppukäyttäjän kannalta uusia toiminnallisuuksia tuotteeseen. Käytännössä teknisen velan poistaminen ei suoraan näy loppukäyttäjälle tai sovelluksessa itsessään, mutta sillä saattaa olla suuri vaikutus sovelluskehityksen ja sen kompleksisuuden kannalta. Tämän lisäksi se vaikuttaa ohjelmiston vakauteen, suorituskykyyn ja tietoturvaan.

Teknisen velan kitkemiseksi tiimin kehitysjatoksia olivat:

- Yhteisten käytäntöjen luominen ja määrittely
- Refaktorointi eli koodin uudelleen kirjoittaminen
- Koodin modulaarisuus eli riippumattomuus

Teknisen velan haasteellisuuden osalta suunniteltiin tiimiä, joka luo yhteisiä käytäntöjä, miten teknistä velkaa pyritään välttämään. Muutamia alustavia avainkohtia ovat 1) kopioidun koodin välttäminen 2) objektien ja funktioiden rajaaminen vastuultaan pieneksi 3) käyttämättömän koodin poistaminen 4) refaktoroinnin läpikäyminen kollegan kanssa, jolla on tietoa kyseisestä teknologiasta.

6.5 Yhteenveto

Digitaalisten palveluohjelmistoprojektien onnistuminen on pääsääntöisesti haastavaa, koska ne ovat monimutkaisia kokonaisuuksia. Digitaalisten projektien osalta pätee sama kuin perinteisten projektien eli jokainen on oma kokonaisuus eikä toteutusta voida kopioida seuraavan projektin kohdalla. Lisäksi ohjelmistoprojektien budjetti on korkea, joten projektin onnistuminen on tavoiteltua.

Ohjelmistoprojektien onnistumista on tutkittu paljon ja Standish Groupin eli kansainvälisen tutkimusneuvontaorganisaation viimeisimmän tutkimuksen mukaan vuosina 2011-2015 ketterä ohjelmistokehitysmalli on tuottanut parempia lopputuloksia verrattuna vesiputousmalliin. Peilaten edellä mainittuun tutkimukseen, yritys X Oy:ssä on valittu hyvä ja toimiva ohjelmistokehitysmalli.

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

KUVA 15. Projektien onnistuminen ketterä vs vesiputous (Standish Group, 2015)

Nerg (2019) on tutkinut tietojärjestelmäprojektien onnistumiseen ja epäonnistumiseen vaikuttavia tekijöitä. Tutkimuksen mukaan onnistumisen kannalta on oleellista, että ”*projektille asetettujen tavoitteiden pitää olla selvillä tilaaja- ja toimittaja osapuolille, jotta tiedetään, mihin pyritään*”. Lisäksi viestinnän tärkeyden merkitys kävi ilmi niin Nergin työssä kuin Lauri Müllerin (2020) Pro Gradu tutkielmassa it-projektin onnistuminen. Samoihin lopputuloksiin, eli tavoitteiden selkeyteen ja viestinnän aktiivisuuteen, päädyttiin myös tässä opinnäytetyössä. Tässä työssä löydettyjä tavoitteiden mielessä pitämistä halutaan kehittää entisestään projektityökalujen avulla. Nergin ja Müllerin tutkimuksista poiketen, yritys X Oy:ssä viestinnän tärkeys painottui enemmän yrityksen ja asiakkaan väliseen kommunikointiin eikä niinkään tiimin sisäiseen viestintään.

Yritys X Oy:ssä epäonnistumisen syitä löytyi huomattavasti enemmän kuin onnistumisen. Syy saattaa olla siinä, että tiimillä oli kokemuksia ja yleisesti luettua tietoa ohjelmistoprojektin epäonnistumisen syistä jo valmiiksi. Rami Juvonen kirjoittaa kirjassaan, että projektien suuruusluokan kasvaessa sen hallinta on hankalampaa ja Standish Groupin tutkimuksen mukaan suurista yli kymmenen

miljoonan dollarin projekteista vain kahdeksan prosenttia onnistuu. Muita sudenkuoppia projektien epäonnistumiselle Juvosen mukaan on:

- 1) heikko business case, joka vaikuttaa projektihenkilöstön motivaatioon
- 2) priorisointi (aikataulu, budjetti, sisältö)
- 3) loppukäyttäjän kuunteleminen eli projektin vaatimukset määritelty osittain ilman loppukäyttäjän näkökulmia
- 4) yllätykset eli erilaisiin löydöksiin ei varata aikaa. (Rami Juvonen 2020).

Anders Borgström listaa opinnäytetyössään yleisimmät it-projektien epäonnistumisen tekijät, jotka ovat 1) projektisuunnittelun ongelmat 2) johtamisongelmat ja 3) kehittäjien ja loppukäyttäjien välisen yhteistyön ongelmat. (Anders Borgström 2015) Näiden lisäksi BMBOK:n mukaan yleinen juurisyy epäonnistumiselle on sidosryhmien vähäinen tuki tai puuttuminen kokonaan. Tämä saattaa näkyä sidosryhmien epäselvinä vaatimuksina tai passiivisuutena henkilömuutosten yhteydessä. (BMBOK 2018) Juvosen kirjassa ja Borgströmin tutkimuksessa yhtenäisiä syitä projektien epäonnistumiselle listataan projektin valmistelun ja suunnittelun puutteet ja laiminlyönnit sekä loppukäyttäjän näkökulmien huomioitta jättäminen.

Yritys X Oy:ssä toteutetussa retrospektiivissä listatuista syistä ainoa yhtenevä aihe Juvosen ja Borgströmin luokittelun kanssa oli motivaatiotekijät. Muilta osin vastaukset eivät olleet yhteneväisiä. Motivaatiotekijöinä tiimi listasi myös erilaisia ajatuksia kuin Juvonen tai Borgström. Muun muassa heikko business case ei tullut keskustelussa lainkaan puheeksi, vaikka Juvonen määrittelee sen merkittävänä tekijänä henkilöstön motivaation kannalta.

Yritys X Oy:n motto, ihmisläheinen ohjelmistokehitys, on osana yrityskulttuuria, joten loppukäyttäjät ovat keskiössä ja tältä osin Juvosen ja Borgströmin esiin nostama sudenkuoppa liittyen loppukäyttäjän kuuntelemiseen on onnistuttu välttämään yritys X Oy:ssä. Johtamisongelma ja muutostenhallinta on onnistuttu ratkaisemaan hyvin pitkälti ohjelmistosuunnittelijoiden vankalla kokemuksella ja yritys X Oy:ssä tehtävällä hyvällä ohjelmistodesignilla sekä tiimien itseohjautuvuudella.

7 POHDINTA

Opinnäytetyössä tutkittiin ohjelmistoprojektien onnistumisen ja epäonnistumisien syitä yritys X Oy:ssä sekä kehitettiin projektimallin toteutukseen liittyvää osaluuetta. Tutkimustulosten perusteella voidaan löytää pääkohdat jotka:

- 1) tarvitaan ennen projektin toteutusvaiheen alkua
- 2) vaikuttavat yritys X Oy:n projektien onnistumiseen
- 3) vaikuttavat yritys X Oy:n projektien epäonnistumiseen

Tulosten perusteella oli suoraviivaista rakentaa projektimallin aloitukseen liittyvä tarkistuslista ja jalkauttaa se projektikäytäntöihin. Tarkistuslista on ollut käytössä sen luomisesta lähtien ja se on otettu hyvin vastaan yritys X Oy:ssä sekä koettu tarpeelliseksi.

Kohdeyrityksen prosessit ja käytännöt ovat pääsääntöisesti hyvässä kunnossa. Projekteissa on käytössä Scrum-viitekehys, joka Standish Group:n tutkimuksen mukaan tuottaa parempia lopputuloksia kuin perinteinen vesiputousmalli. Projektien kannalta on kuitenkin syytä painottaa projektin tavoitteita ja asiakasyhteistyön tiivistämistä. Projektin tavoitteiden mielessä pitämiseen voidaan vaikuttaa lisäämällä läpinäkyvyyttä projektihallintatyökalujen kautta. Asiakasyhteistyön tiivistämisen parantamiseksi ehdotettu retrospektiivi on syytä ottaa laajemmin käyttöön.

Lisäksi tuloksista kävi ilmi, mitkä asiat tarvitsevat muutoksia. Selkeitä toimenpiteitä tarvitaan eritoten pienissä yhdestä kahteen hengen projekteissa. Näiden projektien ongelmat kasaantuvat työtehtävien määrän sekä resurssien vähyyden vuoksi. Esimerkiksi kahden henkilön projektissa, jossa molemmat tekevät eri alustoja, hoitavat asiakasrajapinnan kommunikointia sekä sovellukseen liittyviä muutoksia ja julkaisuja, aikaa kattavaan testaamiseen sekä koodin katselmointiin ei juurikaan ole. Tämä näkyy siinä, että katselmointia ja testaamista ei pystytä tekemään halutulla tasolla. Pienissä projekteissa puuttuu ihmisten välinen sosiaalisuus, joka on tärkeä osa yhteisöllisyyttä ja yrityskulttuuria. Oleellista olisi myös tieto siitä, keneltä löytyy apu ongelmatilanteissa, mutta sosiaalisuuden puutteen vuoksi tämäkin tieto jää yleensä saamatta. Lisäksi palautteen saaminen, joka on yksi onnistuneen

projektin avaintekijöistä, on vähäistä, mikä osaltaan vaikuttaa henkilöiden motivaatioon.

Edellä tunnistettuihin ongelmiin on lähdetty hakemaan ratkaisua tiimien rakentamisella eli niputetaan pieniä projekteja yhteen, että saadaan yhteinen isompi tiimi. Tämän lisäksi jokaiseen tiimiin on tarkoitus lisätä kompetenssivetäjä, joka pystyy auttamaan eri projektien ongelmatilanteissa. Tulosten pohjalta yritys X Oy:n tiloihin tuotiin kiitoskaappi, josta on mahdollista jakaa pieni muistaminen kollegalle hyvin tehdystä työstä. Kiitoskaappi on yksi keino parantaa projektitiimiläisten motivaatiota, sillä tällä tavalla palautteen saaminen ja antaminen on helppoa ja nopeaa.

Koska opinnäytetyön tekemisen aikana yritys fuusioitui isomman toimijan kanssa, jatkokehitettäviä ideoita projektien kehittämisen osalta voisivat olla: 1) projektimallien ja -käytäntöjen yhdistäminen yritysfuusion jälkeen 2) projektihallintatyökalujen yhtenäistäminen 3) ketterän menetelmän jalkauttaminen yritykseen.

Päättötyön tekeminen alkoi keväällä 2020 ja aluksi työ koki vähän haasteita johtuen meneillään olevista opiskelukursseista ja töissä olevista projektikiireistä. Näistä seikoista johtuen toteuttaminen pääsi kunnolla käyntiin vasta keväällä 2021. Työssä oli mielenkiintoista pureutua ketterän mallin teoriaan ja abc-projektimalliin sekä syventää tietoa niin projektihallinnan työkaluista kuin sen menetelmistä. Työhön liittyen luettiin muutama todella mielenkiintoinen kirja kuten Harold Kerznerin *A Guide to Measuring and Monitoring Project Performance* ja Rami Juvosen *ohjelmistoprojektin sudenkuopat ja miten ne vältetään*. Työ saavutti sille asetetut tavoitteet ja hyödytti konkreettisesti kohdeyritystä.

LÄHTEET

Ahtola, H. 2018 Miksi ottaa asiakas mukaan palvelun kehitysprosessiin? Luettu 1.4.2021 <https://www.arter.fi/miksi-ottaa-asiakas-mukaan-palvelun-kehitysprosessiin/>

Baldauf, C. 2018. The best retrospective for beginners. Luettu 2.2.2020 <https://retromat.org/blog/best-retrospective-for-beginners/>

Bassil, Y. 2012. A Simulation Model for the Waterfall Software Development Life Cycle: Libanon luettu 17.4.2020 <https://arxiv.org/pdf/1205.6904.pdf>

Beyer, H. 2010. User-Centered Agile Methods. SYNTHESIS LECTURES ON HUMAN-CENTERED INFORMATICS. U.S.A: Morgan & Claypool.

Bourque, P., Fairley, R.E. 2014. Guide to the Software Engineering Body of Knowledge, Version 3.0, IEEE Computer Society

Hirsjärvi, S., Remes, P., Sajavaara, P. & Sinivuori, E. 2009. Tutki ja kirjoita. 15. uud. p. Helsinki: Tammi

IEEE, Ontology driven software engineering. 2012. IEEE Canadian Conference on Electrical and Computer Engineering. Montreal, QC, Canada:IEEE. Luettu 8.2.2021. <https://ieeexplore.ieee.org/document/6334938>

Innokylä. 2021. Learning cafe eli oppimiskahvila Luettu 2.2.2021 <https://innokyla.fi/fi/tyokalut/learning-cafe-eli-oppimiskahvila>

Juvonen R. 2018. Ohjelmistoprojektin sudenkuopat ja miten ne vältetään. Suomi: Books on Demand

Kerzner, H. 2017. Project Management Metrics, KPIs, and Dashboards: A Guide to Measuring and Monitoring Project Performance. U.S.A: Wiley

Kettunen, S. 2009. Onnistu projektissa. Juva: WS Bookwell Oy

Lara, J. 2018. Agile Scrum Process in a Nutshell. Luettu 16.2.2021 <https://medium.com/@realjoselara/agile-scrum-process-in-a-nutshell-6ec32a59efb>

Larsen, D., Derby, E. & Schwaber, K. 2006. Agile Retrospectives: Making Good Teams Great. U.S.A: The Pragmatic Programmers

Leppäniemi, P. 2020 Laatu tekniikoilla, konventioilla ja suunnittelumalleilla Luettu 2.4.2021 <http://ohjelmistotuotanto.panuleppaniemi.com/laatu-ohjelmistoissa/>

Lynn, R. n.d The History of Agile. Luettu 4.4.2021 <https://www.planview.com/resources/guide/agile-methodologies-a-beginners-guide/history-of-agile/>

Mäkilouko, M. yliopettaja 2019. Opinnäytetyön tutkimusmenetelät. Sähköpostiviesti. Luettu 28.9.2019.

Mäkilouko, M. 2020. Teknologiaprojektien johtaminen -kurssin luentomateriaali.

Müller, L. 2020. IT-Projektin onnistuminen - projektihallinnan työkalut. Infomaatioteknologian tiedekunta. Jyväskylän yliopisto. Pro gradu

Nerg, M. 2019 Tietojärjestelmäprojektin onnistumiseen tai epäonnistumiseen vaikuttavat tekijät. Liiketalouden tutkinto-ohjelma. Jyväskylän ammattikorkeakoulu. Opinnäytetyö

Nonaka, I., & Takeuchi H. 1995. The knowledge-Creating Company: How Japanese Companies create the Dynamics of Innovation. Oxford University Press.

Pulkkanen, A. 2020. 7 toimintatapaa, joilla varmistat projektisi lähtevän oikeaan suuntaan Luettu 16.3.2021

<https://www.agendum.com/projektinhallinta/avaintiedot>

Projekti-instituutti, https://www.projekti-instituutti.fi/files/58/Omistajuus_PRY0206.pdf

Schwaber, K., & Sutherland J. 2020. Scrum-opas -Scrumin määritelmä ja pelisäännöt Luettu 16.2.2021

<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Finnish.pdf>

Suomidigi 2.3.2021 Digitaalisten palvelujen asiakaslähtöinen suunnittelu Luettu 2.4.2021 <https://www.suomidigi.fi/sites/default/files/2020-03/Digitaalisten%20palvelujen%20asiakaslähtöinen%20suunnittelu.pdf>

World Café. 2021. World Cafe Method. Luettu 2.3.2021.

<http://www.theworldcafe.com/key-concepts-resources/world-cafe-method/>

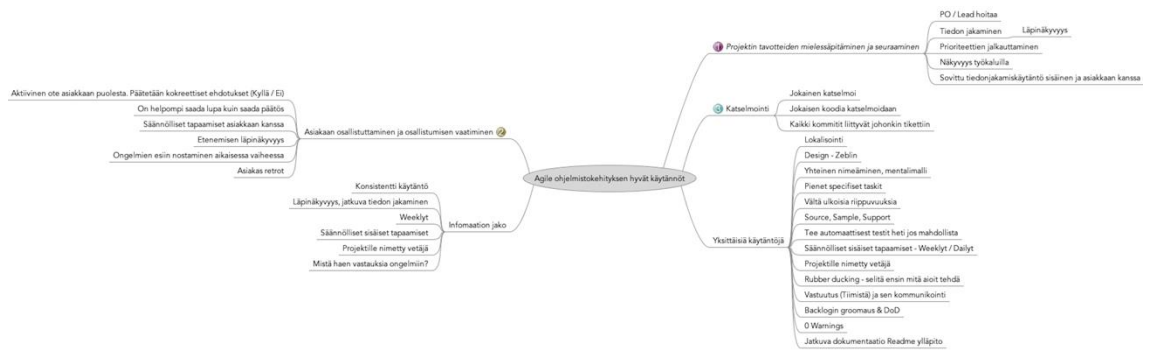
Wysocki, R. 2006 Effective Software Project Management. U.S.A: Wiley

LIITTEET

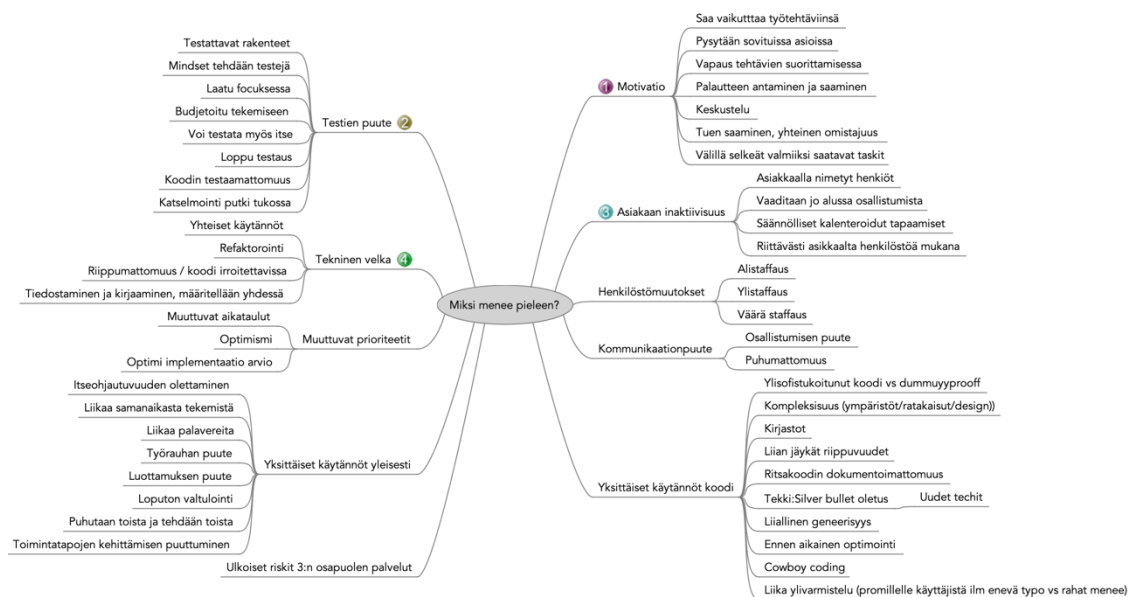
Liite 1. Ajatuskartta projektien aloittaminen



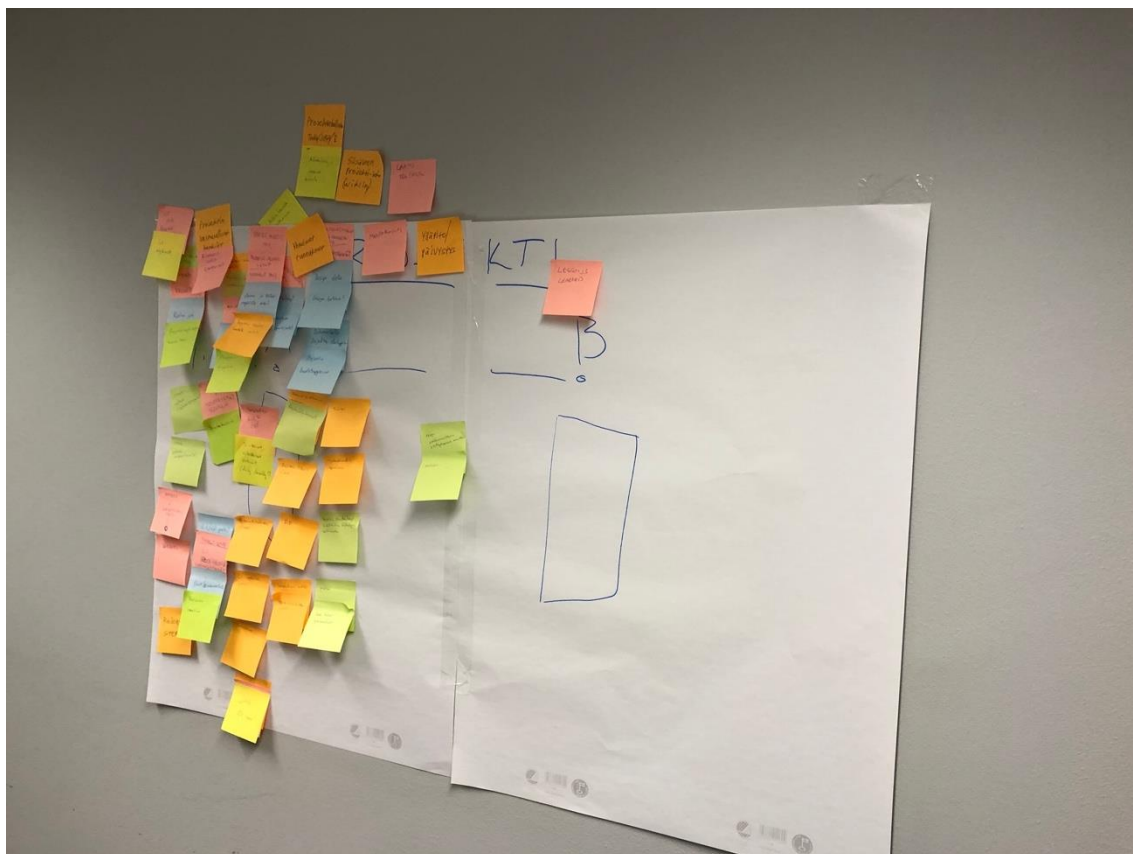
Liite 2. Ajatuskartta Mitkä ovat projektin hyvät käytännöt?



Liite 3. Ajatuskartta Miksi menee pieleen?



Liite 4 Tunnistettuja tehtäviä tarkistuslistaa varten



Liite 5. Projektien hyvät käytännöt



Liite 5 Miksi menee pieleen?



Liite 6 Retrospektiivin palaute

