

VAASAN YLIOPISTO

TEKNILLINEN TIEDEKUNTA

OHJELMISTOTEKNIikka

Lähteinen Mari

**OHJELMISTOKEHITYSPROSESSIN LAADUN JA KUSTANNUSTEHOK-
KUUDEN KEHITTÄMINEN**

Diplomityö, joka on jätetty tarkastettavaksi diplomi-insinöörin tutkintoa varten
Vaasassa 19.3.2015.

Työn valvoja

Jouni Lampinen

Työn ohjaaja

Kaisa Ilola

SISÄLLYSLUETTELO

TIIVISTELMÄ.....	3
ABSTRACT	4
1. JOHDANTO.....	5
2. OHJELMISTOKEHITYSPROSESSIMALLEISTA	8
2.1 Vesiputousmalli	10
2.2 V-malli	11
3. OHJELMISTOKEHITYSPROSESSIN KEHITTÄMINEN.....	13
4. CASE: KANSAINVÄLISEN YRITYKSEN OHJELMISTOKEHITYSPROSESSIN KEHITTÄMINEN - TUTKIMUKSEN SUUNNITTELU	17
5. TUTKIMUKSEN TOTEUTTAMINEN	19
5.1 Prosessin kuvaus	19
5.2 Tutkimusaineisto.....	22
5.3 Tutkimuksen suorittaminen	25
6. TUTKIMUSTULOKSET	26
6.1 Havainnot etenemisen seurannan käytännöistä prosessissa.....	26
6.2 Havainnot töiden tiloista ja etenemisestä prosessissa.....	29
6.2.1 Jonossa oloajan analysointia	36
6.2.2 Pitkät ja pitkittyneet työt	37
6.2.3 Toteutuneet kestoajat ja alkuperäiset työmääräarviot	39
7. JOHTOPÄÄTÖKSET	42
LÄHTEET	45
LIITTEET	

VAASAN YLIOPISTO
Teknillinen tiedekunta

Tekijä:	Mari Lähteinen	
Diplomityön nimi:	Ohjelmistokehitysprosessin laadun ja kustannustehokkuuden kehittäminen	
Valvojan nimi:	Jouni Lampinen, professori	
Ohjaajan nimi:	Kaisa Ilola, KTM	
Tutkinto:	Diplomi-insinööri	
Ohjelma:	Tietotekniikan koulutusohjelma	
Suunta:	Ohjelmistotekniikka	
Opintojen aloitusvuosi:	2003	
Diplomityön valmistumisvuosi:	2015	Sivumäärä: 46

TIIVISTELMÄ

Tässä tutkielmassa on analysoitu erään kansainvälisen yrityksen Suomen liiketoimintayksikön ohjelmistokehitysprosessia. Tutkimuksen tavoitteena on löytää kehityskohteita ja tunnistaa käytännön toimenpiteitä, joiden avulla prosessin laatua ja kustannustehokkuutta voidaan parantaa.

Tutkielman aineistoksi koottiin liiketoimintayksikössä vuoden 2013 aikana käynnissä olleet ohjelmistokehitystyöt, jotka valmistuivat viikkoon 6/2014 mennessä. Näistä töistä selvitettiin eri prosessivaiheiden kestoajat viikon tarkkuudella, prosessin aikana uudelleen tehdyn työn osuus ja kesto aika sekä ne ajat, jolloin työ ei edennyt vaan odotti syötettä joltain prosessiosapuolelta. Saatuja tietoja analysoitiin tilastollisin menetelmin ja selvitettiin mm. riippuvuuksia eri tekijöiden välillä. Tilastollisia havaintoja täydennettiin prosessiosapuolten kanssa käytyjen keskustelujen pohjalta.

Tutkimusaineistosta voitiin todeta, että kehitystöihin sisältyi paljon odotusaikaa, jolloin työ ei edennyt. Keskimäärin odotusaika oli yli 30 % työn kestoajasta. Töiden välillä oli suurta vaihtelua odotusajan määrässä, joten aineiston perusteella näyttäisi siltä, että odotuksella on taipumus kasautua tietyille töille ja odotus jossain prosessivaiheessa saattaa ennakoida viivästyksiä myös jatkossa. Pitkään jonottaneissa töissä oli myös usein laadullisia ongelmia, kuten hyväksyntätestausvaiheessa löydettyjä ohjelmointivirheitä, jotka viivästyttivät työn etenemistä entisestään. Osa odotusajasta olisi voitu välttää prosessiosapuolten keskinäistä viestintää tehostamalla ja töiden etenemisen vieläkin systemaattisemmalla ja tarkemmalla seurannalla. Jokainen viivästys lisää työn kokonais kustannuksia ja heikentää kustannustehokkuutta. Näin ollen erityisesti prosessin johtamisen ja työn etenemisen seurantamenetelmien systemaattisella kehittämisellä voidaan saada hyviä tuloksia pyrittäessä parantamaan ohjelmistokehitysprosessin laatua ja kustannustehokkuutta.

AVAINSANAT: ohjelmistotuotanto, ohjelmistoprosessi, ohjelmistoprojekti, ohjelmistokehitysprosessin kehittäminen

UNIVERSITY OF VAASA**Faculty of technology****Author:**

Mari Lähteinen

Topic of the Thesis:

Improvement of quality and cost efficiency in software engineering process

Supervisor:

Jouni Lampinen, Professor

Instructor:

Kaisa Ilola, M.Sc. (econ.)

Degree:

Master of Science in Technology

Degree Programme:

Degree Programme in Information Technology

Major of Subject:

Software Engineering

Year of Entering the University:

2003

Year of Completing the Thesis:

2015

Pages: 46

ABSTRACT

This thesis contains an analysis of software development process in a business unit of an internationally operating company in Finland. The purpose is to find points of improvement and recognize actions for better quality and cost efficiency in development process.

Used data contains all software development tasks in this business unit, which were ongoing during 2013 and were finished until week 6 /2014. Durations of different process phases were calculated in week level, as well as rework and waiting times during the process. Data was analyzed by statistical methods to find out possible correlations between different variables. These statistical observations were supplemented with discussions with people working in different parts of this development process.

Based on data, it was found out, that remarkable amount of waiting time with no progress in process was included in development work. Average amount of waiting time was over 30 % of total duration of development tasks. Variation between different tasks was wide, so it seemed that waiting time tends to cumulate. Development tasks with long waiting time had often problems with quality. E.g. remarkable amount of errors were found during test phase and that caused more delays to these tasks. At least some part of waiting time could have been avoided by more efficient internal communication between process actors and with more systematic and controlled progress follow-up. All delays tend to increase total costs of tasks and decrease cost efficiency. Therefore focusing specially into process management and follow-up of the task progress, it will be possible to gain good results when developing the quality and cost efficiency of software development process.

KEYWORDS: software engineering, software process, software process improvement

1. JOHDANTO

Ohjelmistokehitysprojekteihin kohdistuvat vaatimukset kasvavat yhä nopeammin muuttuvassa liiketoimintaympäristössä. Pitäisi pystyä tuottamaan uusia ohjelmistonosia aiempaa nopeammin ja kustannustehokkaammin, ilman että laatu kärsii (Fenech & De Raffaele 2013). Eräänä ratkaisuvaihtoehtona nopeus- ja laatuavoitteiden saavuttamiseksi on esitelty erilaisia ketteriä ohjelmistokehitysmenetelmiä (engl. Agile methods), kuten esimerkiksi Scrum (Scrum.org 2014) ja Dynamic Systems Development Method eli DSDM (DSDM Consortium 2014). Kaikkiin tilanteisiin nämä mallit eivät sellaisenaan kuitenkaan sovellu, vaan kehitystyö on tehtävä nk. perinteisiä malleja noudattaen. Tällainen tilanne voi olla esimerkiksi silloin, kun tuotteesta ei voida julkaista nk. beta – versiota, jota voidaan myöhemmissä kehitysvaiheissa parantaa. Tai kun tuote on itsessään laaja kokonaisuus, joka on rakennettava kokonaan valmiiksi ennen julkaisua. Näihin projekteihin, joissa ketterät menetelmät eivät sellaisenaan ole soveltuvia, on kuitenkin tarve löytää uusia toimintamalleja sekä kehitysprosessin variaatioita, joilla aiempaa parempi kustannustehokkuus ja toteutuksen suurempi nopeus voidaan saavuttaa.

Tutkimusten mukaan vain pienehkö osa projekteista täyttää niille asetetut tavoitteet. Yhdysvaltalaisen The Standish Groupin tekemän tutkimuksen mukaan vain 32 prosenttia tietoteknisistä projekteista todettiin onnistuneeksi vuonna 2009, haasteita oli 44 prosentissa projekteista ja 24 prosenttia epäonnistui. Tämän tutkimuksen perusteella on siis merkittävästi todennäköisempää, että projekti epäonnistuu ainakin osittain (68 %) kuin että se onnistuu (32 %). Projektin onnistumisen kriteereiksi Standish Groupin tutkimuksessa määritellään: budjetin mukainen, suunnitellussa aikataulussa toteutettu, vaatimusmäärittelyjä vastaava toteutus, yleinen tyytyväisyys (asteikolla erittäin korkeasta erittäin alhaiseen). Tähän arviointikriteerien listaan lisättiin vielä organisaation strategian mukaisten tavoitteiden saavuttaminen ja arvon tuottaminen organisaatiolle. Näiden arviointikriteerien merkityksestä tehtiin kysely Standish User Research Forum käyttäjille, joita pyydettiin valitsemaan näistä kolme mielestään tärkeintä kriteeriä. Kyselyn tuloksena kolmeksi tärkeimmäksi kriteeriksi nousivat arvon tuottaminen organisaatiolle 52 %, yleinen tyytyväisyys 41 %, budjetissa pysyminen 32 %. Kun kriteerejä verrattiin

Standish Groupin tutkimustaan varten keräämään projektitietokantaan, vain 1,2 % projekteista täytti kaikki kriteerit ja yksittäisen kriteerin kohdalla esim. budjetissa pysyminen toteutui 42 % projekteista. Tähän pohjautuen Standish Group suosittaakin, että yksittäisten projektien onnistumisen arvioinnin sijaan yritysten kannattaisi keskittyä kehittämään projektisalkkunsa arvoa kokonaisuutena. (Eveleens & Verhoef 2010; Standish Group 2014.)

Projektin onnistumista voidaan mitata eri tavoin. Projektille voidaan asettaa, mittareita, joilla pyritään todentamaan esimerkiksi projektin laatu ja kustannustehokkuus. Laatu voidaan niin ikään määritellä usein eri tavoin. Laadukkaana pidetään tuotetta tai palvelua, joka on virheetön. Ohjelmistotuotteiden kohdalla laatuna pidetään usein sitä, miten hyvin tuote täyttää käyttäjänsä tarpeet, kohtuulliset toiveet ja odotukset. (Haikala & Märijärvi 2004:48; Pressman 2005.) Watsonin (2005:18–19) mukaan voidaan myös ajatella, että laatu on mitä tahansa, mitä asiakas haluaa tai pitää riittävänä laatuna, koska laadun ja virheettömyyden määrittelee tilaaja/asiakas.

Tehokkuutta voidaan mitata projektin ajallisella kestolla: miten hyvin projekti pysyy suunnitellussa aikataulussa? Tosin projektin aikatauluarvio voi olla epäonnistunut tai epärealistinen, joten pelkästään aikataulun tarkasteleminen ei kerro projektin tehokkuudesta. Kaikkia projektiin liittyviä tehtäviä ei aina pystytä tunnistamaan projektin alkuvaiheessa eikä niiden ajallista kestoa pystytä välttämättä arvioimaan riittävän tarkasti. Tehokkuutta on prosessimielessä vaihejakomallin mukaan etenevässä kehitystyössä myös se, että projekti etenee mallin mukaisesti. Silloin projektin aikana ei jouduta palaamaan edelliseen vaiheeseen eli uudelleen tehtävän työn (engl. rework) osuus prosessissa on mahdollisimman vähäinen. Tällöin projektin tuottavuus on mahdollisimman suuri käytettyihin resursseihin (aika, raha, ihmiset jne.) nähden (Haikala ym. 2004), joten viimekädessä tehokkuus johtaa aiempaa alhaisempiin kustannuksiin. Deming (1981) yhdistää nämä toteamalla, että tuottavuutta ja kustannustehokkuutta voidaan parantaa kehittämällä laatua. Samalla, kun parannetaan laatua, voidaan hänen mukaansa alentaa kustannuksia, parantaa kilpailukykyä ja henkilöstön tyytyväisyyttä työhönsä. Hänen mukaansa laatua parannetaan kehittämällä prosessia siten, että poistetaan tekemisestä hukkaa (engl. waste) ja karsitaan prosessissa tapahtuvien virheiden määrää.

Tämän tutkielman tarkoituksena on analysoida erään kansainvälisen yrityksen yhden liiketoiminta-alueen IT –kehitysprosessia. Työn lopputuloksena syntyy selvitys, jossa on tunnistettu kehityskohteet, joihin keskittymällä nykyistä prosessia voidaan parantaa. Parannusehdotukset tähtäävät sekä projektien laadun että kustannustehokkuuden parantamiseen prosessia kehittämällä. Demingin (1981) ajatusten mukaisesti parempaa laatua ja sitä kautta parempaa tuottavuutta haetaan hukkatyön tunnistamisella (mm. odotusaika prosessissa, uudelleen tekeminen) ja karsimisella sekä virheettömyyteen pyrkimisellä.

Olen tehnyt analysointi- ja tutkimustyön kyseisen yrityksen palveluksessa työskennellen ja prosessissa osallisena olevana, millä on vaikutusta sekä työn etenemiseen, johtopäätöksiin että lopputuloksiin. Osa tutkimuksen aikana tehdyistä havainnoista on viety eteenpäin jo työn aikana ja kehitystoimenpiteet käynnistetty saman tien. Liitteenä (liite 1) olevassa toimenpidesuosituksen listassa on kuitenkin lueteltu kaikki kehityssuositukset riippumatta siitä, onko niitä jo aloitettu vai ei.

2. OHJELMISTOKEHITYSPROSESSIMALLEISTA

Määritelmän mukaan ”prosessi on systemaattinen toimintatapa jonkin tavoitteen saavuttamiseksi”. Prosessi muodostuu ihmisistä, tekemisistä, järjestelmistä, toimintatavoista, työkaluista ja toisista prosesseista. (Haikala & Mikkonen 2011:137.) Toimenpiteet suoritetaan ennalta määrättyssä järjestyksessä ja ne ovat toistettavissa yhä uudelleen. Projekti on prosessin ilmentymä, kertaluontoinen tehtävä, jolla on nimetty organisaatio sekä annetut resurssit ja aikataulu. Projekti suoritetaan prosessin mukaisesti. Prosessia voidaan kierrättää eli sen mukaisesti voidaan toteuttaa erilaisia projekteja. (Haikala ym. 2004)

Ohjelmistokehitysprosessi voidaan määritellä toisiaan seuraavien toimintojen ketjuksi, joka muuttaa käyttäjän tarpeet ohjelmistotuotteeksi. Ohjelmistokehitysprojekti puolestaan on aikataulun, sisällön ja resurssien osalta rajattu kokonaisuus, joka toteutetaan sovitun prosessin noudattaen eli se on ohjelmistokehitysprosessin ilmentymä. Samaa määriteltä kehitysprosessia noudattaen voidaan toteuttaa keskenään erilaisia ohjelmistokehitysprojekteja. Prosessia voidaan kehittää ja parantaa, ja kehityksen tulisi olla jatkuvaa. Kehitys tapahtuu erillään projekteista eli kehitystyöllä ei yleensä muuteta meneillään olevien projektien tekemistä, mutta prosessiin tehdyt parannukset vaikuttavat tuleviin projekteihin. (IEEE 1990; Haikala ym. 2004; Haikala ym. 2011 PMBOK® Guide 2008.) Ohjelmistokehitysprosessin kehittäminen voi kohdistua joko kehitystyön hallinnan prosessiin tai varsinaisen ohjelmointityön prosessiin. Tämä työ keskittyy projektin hallinnan prosessiin eli siihen, miten kehitystyötä seurataan ja edistetään eri prosessiosapuolten toimesta sekä kuinka tehokkaasti ja suoraviivaisesti työ kulkee prosessiketjun läpi. Ohjelmointityön prosessin kehityksessä keskitytään esimerkiksi työn mallintamisen ja määrittelytyön kehittämiseen sekä niihin liittyviin työkaluihin ja menetelmiin (esim. UML – Universal Modeling Language, SPEM – Software Process Engineering Metamodel), joiden käyttöä ja vaikuttavuutta on tutkinut tarkemmin mm. Mäkilä (2012).

Pfleeger ja Atlee (2010:72) totesivat, että prosessiin kuuluvat toimenpiteet ja tekemiset, jotka tarvitaan prosessin läpiviemiseksi. Prosessi voi koostua toisiinsa linkittyvistä aliprosesseista ja se voi olla hierarkkisesti määritelty siten, että jokaisella aliprosessilla on oma prosessimallinsa. Jokaisella prosessin toiminnolla on määritelty käynnistymisen (sisääntulo, engl. entry) ja lopetuksen (poistuminen, engl. exit) ehdot, jolloin aktiviteetin alku ja loppu tunnustetaan. Prosessin toiminnot on järjestetty sarjaksi, jolloin tiedetään millä tavoin toiminnot ovat riippuvaisia toisistaan. Prosessin jokaiselle toiminteelle on määritelty tavoitteet, jotka kuvataan prosessiohjeissa. Toimintoihin, resursseihin tai tuotteeseen itseensä voi liittyä rajoitteita ja kontrollipisteitä.

Prosessit ovat tärkeitä, koska ne antavat toimintojen sarjalle rakenteen ja vakauden. Ne helpottavat myös tiedon keräämistä ja jakamista, kun toteutunutta tekemistä voidaan verrata kuvattuun prosessiin ja havaintojen kautta voidaan oppia itse ja opettaa muita. Aiemmistä projekteista oppiminen ja havaintojen perusteella tapahtuva kehittäminen lisäävät prosessin ymmärtämistä, kontrollissa pysymistä ja mahdollistavat prosessin lopputuloksen paranemisen (tuotteen laadun parantamisen) tulevissa projekteissa (Pfleeger *et al.* 2010: 72–73).

Ohjelmistokehitysprosessiin kuuluu tavallisimmin seuraavat vaiheet: vaatimusten analysointi ja määrittely, suunnittelu (systeemisuunnittelu, ohjelman suunnittelu), ohjelmointi, testaus (yksikkötestit, integrointitestit, systeemitestit, hyväksymistestit), järjestelmän toimittaminen ja ylläpito. Nämä vaiheet ovat käytössä etenkin silloin, kun toimitetaan kokonainen uusi järjestelmä. Jokainen edellä mainitusta vaiheista on oma osaprosessinsa ja voidaan kuvata toimenpiteiden sarjana, joihin liittyy rajoitteita, syötteitä, tuloksia ja resursseja. Yhdessä ne muodostavat ohjelmistokehityksen prosessikokonaisuuden, jonka mukaisesti ohjelmistoprojekteja ja kehitystöitä voidaan toteuttaa. (Haikala ym. 2004; Pfleeger *et al.* 2010:73.)

Prosessi voidaan mallintaa eri tavoilla. Näistä malleista esitellään jäljempänä tässä luvussa tarkemmin ne mallit, joihin tutkimuksen kohteena olevan IT-töiden prosessin voidaan ensisijaisesti ajatella pohjautuvan. Tämän tutkielman kohteena ovat olemassa olevan järjestelmän pienkehitystyöt (uudet tuotteet ja uudet toiminnot tai päivitykset), joten

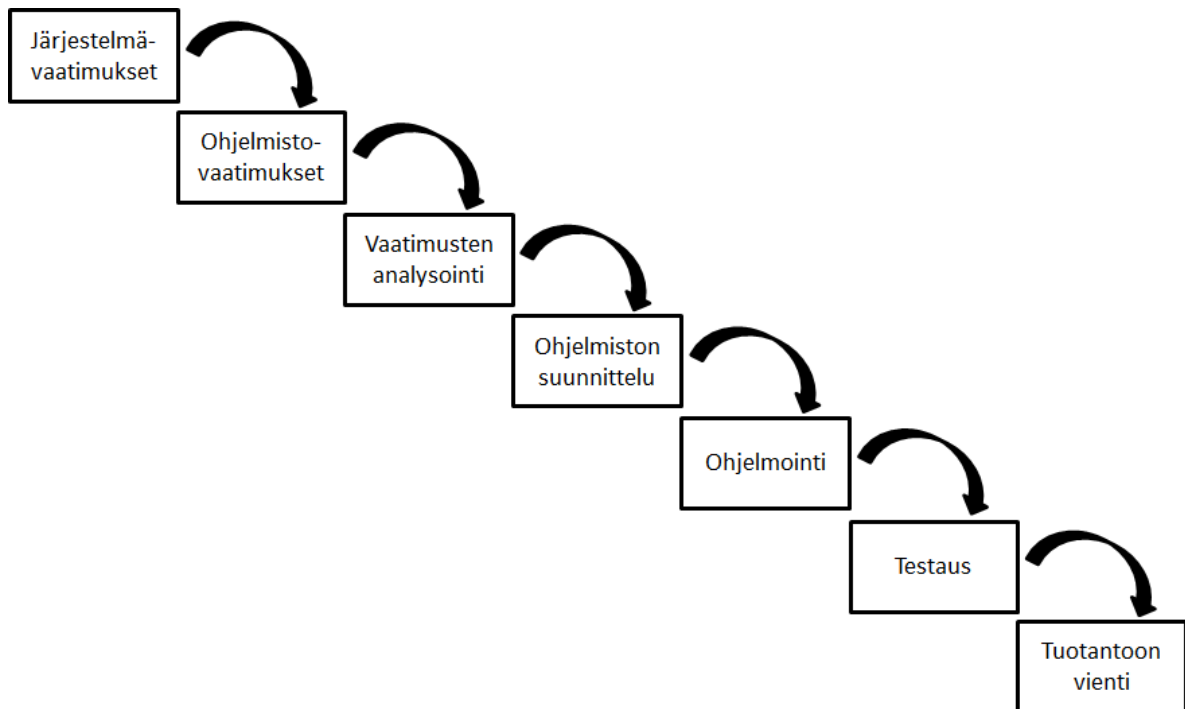
prosessi on karsittu versio teoreettisesta optimista, mutta sisältää olennaisimmat prosessin osat. Analysoinnin kohteena olevan prosessin tarkempi rakenne on esitelty luvussa 5.

Ohjelmistokehityksen prosessimallit ovat usein nk. vaihejakomalleja, jolla tarkoitetaan tapaa, miten ohjelmiston tuottaminen jaetaan vaiheisiin. Malleja voidaan soveltaa myös ohjelmiston koko elinkaareen eli kehitystyön aloittamisesta ohjelmiston poistamiseen käytöstä. Vaihejakomallin rinnalla käytetään tästä syystä myös termiä elinkaarimalli. Vaihejakomallin perustyyppinä ovat lineaarinen malli, inkrementaalinen malli, evoluutiomalli, protoilumalli ja iteratiivinen malli. Lisäksi kehityksessä käytetään edellä mainittujen mallien yhdistelmiä. Lineaarisista malleista tunnetuin on vesiputousmalli, joka on myös tämän tutkielman kohteena olevan ohjelmistokehitysprosessin ohjaavana mallina ja johon syvennyttään tarkemmin jäljempänä tässä luvussa. Lisäksi tässä yhteydessä esitellään tarkemmin myös vesiputousmallin laajenuksena pidettyä V-mallia, koska se havainnollistaa hyvin käyttöönottoon valmistautumisen ja testausvaiheiden takaisinkytkennän määrittelyyn ja toteutukseen. (Haikala ym. 2004.)

2.1 Vesiputousmalli

Vesiputousmallina tunnettu prosessikehitysmalli on yksi ohjelmistokehityksen perinteisistä malleista. Malli on saanut nimensä vesiputousmaisesta kuvaustavasta, jossa edetään vaihejakomallille tyypillisesti yksi vaihe kerrallaan ja siirrytään seuraavaan vaiheeseen, kun edeltävä vaihe on valmis (kuva 1). Sen esitteli ensimmäisenä Winston Royce artikkelissaan *Managing the Development of Large Software Systems* (Royce 1970) ja se vakiintui ohjelmistokehityksen malliksi, vaikka Royce totesi jo mallia kuvatessaan, että siinä on heikkouksia. Hän totesi mm. että, ”vapaiden ohjaksien antaminen ohjelman tuottajalle järjestelmävaatimusten ja tuotantoon viennin väliseksi ajaksi on ongelmien kerjäämistä”. Eli mallin vaiheittaisesta kuvaustavasta huolimatta, Royce piti tärkeänä, että asiakas on tiiviisti mukana kehitystyössä koko prosessin ajan ja että arviointikierroksia käydään jatkuvasti prosessin aikana, jotta etenemisen oikea suunta voidaan varmistaa kaiken aikaa. Jo mallia esitellessään Royce siis painotti asiakkaan osallistumisen

ja jatkuvan kommunikoinnin merkitystä. Eli hän painotti samoja asioita, joita on viime vuosikymmeninä korostettu erityisesti ketterissä kehitysmalleissa (engl. Agile Development) kuten mm. Crystal, Scrum ja XP-malleissa. (Agile Alliance 2004; Pfleeger *et al.* 2010: 85.)

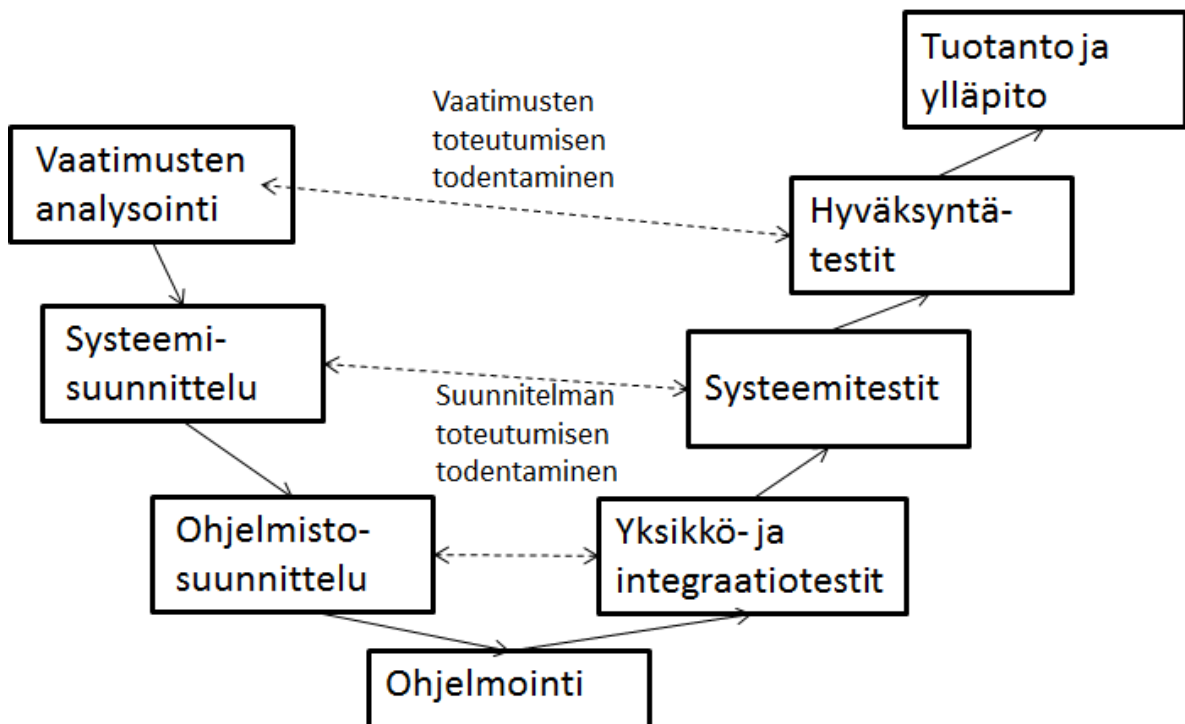


Kuva 1. Vesiputousmalli (Royce 1970).

2.2 V-malli

Eräs vesiputousmallin kehitysversio on V-malli, joka kuvaa testausvaiheiden linkittymistä vaatimusten määrittelyyn, analysointiin ja suunnitteluun (kuva 2). Sen avulla voidaan vesiputousmallia selkeämmin visualisoida kehitysiteraatiota, sillä jos oikean laidan testausosiossa löydetään virheitä, palataan silloin vasemman laidan vaiheisiin virheiden korjaamiseksi. Testauksen suunnittelu tehdään vähintään alustavalla tasolla osana ohjelmiston määrittelyä ja vastaavasti testauksessa verrataan toteutusta määrittelydokumentaatioon. Eli yksikkö- ja integraatiotestillä varmistetaan, että koodi vastaa suunniteltua toteutusta (design), systeemitestissä varmistetaan että ohjelma on systeemisuunnit-

lussa kuvatun mukainen ja hyväksyntätesteissä varmistetaan, että toteutus on vaatimusmäärittelyn mukainen. Yksikkö-, integraatio- ja systeemitestit tekee yleensä koodaaja itse tai ohjelmoinnin tekevän yrityksen erityisesti nimeämät testaajat. Hyväksyntätestin tekee useimmiten asiakas itse tai asiakas vähintäänkin hyväksyy katselmoinnin kautta testien tulokset, koska hyväksyntätestien läpäisy merkitsee käytännössä laskutuslupaa ohjelmointiyritykselle: on toimitettu se, mitä tilattiin. Kuva havainnollistaa myös virheen löytämisen ja korjauksen kustannuksia – mitä korkeammalla V:n oikealla laidalla ollaan, sitä kalliimmaksi löytyneen virheen kustannukset tulevat. Eli yksikkötestissä löytyneen virheen korjaaminen on edullisempaa kuin hyväksyntätestauksessa löydetyn, koska integraatiota esim. muihin järjestelmiin on jo tehty ja korjaus saattaa vaikuttaa hyvinkin laajalle alueelle. (Haikala ym. 2004: 40; Haikala ym. 2011: 208, Pfleeger *et al.* 2010: 78–79.)



Kuva 2. V- malli (Pfleeger *et al.* 2010)

3. OHJELMISTOKEHITYSPROSESSIN KEHITTÄMINEN

Prosessikehityksen kenties perinteisin tapa on nykytilan prosessin kuvaaminen, tavoitetilan määrittely ja kehityskohteiden tunnistaminen analysoimalla nykytilan ja tavoitetilan erot sekä etsimällä ratkaisut, joilla ero saadaan poistettua. Kuten Mäkilä (2012:14) toteaa, yleisenä ”totuutena” etenkin perinteisten ohjelmistokehitysprosessimallien (mm. vesiputousmalli) osalta pidetään sitä, että hyvä ja toimiva prosessi tuottaa parhaimman laadun myös itse prosessin kohteessa eli ohjelmointityössä. Eli prosessia kehittämällä parannetaan myös valmistuvan ohjelmistotyön laatua. Erityisesti huolellista suunnittelua ja vaatimusten hallintaa pidetään tärkeinä tekijöinä paremman laadun aikaansaamiseksi. Projektin tai kehitystyön kohteen (engl. scope) muuttuminen kehitysprosessin aikana on projektin onnistumisen kannalta todettu kriittisimmäksi epäonnistumisen riskiä kasvattavaksi tekijäksi (Hashim, Abbas & Hashim 2013).

Prosessin analysoinnissa voidaan käyttää hyväksi mm. tilastollisia menetelmiä. Jos prosessia sovelletaan hyvin samankaltaisiin projekteihin, voidaan analysoida mm. eri vaiheiden ajallista kestoa, kestoajoissa esiintyvää vaihtelua ja vaihtelun syitä. Six Sigma laadunkehitysmallin perusajatuksiin kuuluu mm. se, että laadullisesti parasta tulosta tuottaa sellainen prosessi, jossa prosessin sisäinen vaihtelu on mahdollisimman vähäistä. Tätä näkökulmaa korostaa myös Deming (1981). Pyritään siis tasaiseen laatuun ja siihen, että prosessin toistaminen kerrasta toiseen tuottaa ennalta määritellyn ja yllätyksettömän lopputuloksen (Watson 2005).

CMMI –malli eli kehityksen maturiteettimalli on Software Engineering Institutun (SEI) kehittämä ja ylläpitämä malli, jonka avulla voidaan arvioida prosessia joko jatkuvana tai vaiheittaisena mallina. Vaiheittaisessa mallissa jokaiselle prosessivaiheelle on määritetty maturiteetti- eli kypsyystasot (5 tasoa) sekä kuvattu, mitä kunkin kypsyystason saavuttaminen edellyttää. Jatkuvassa mallissa maturiteettitasojen (4 tasoa) avulla arvioidaan prosessin kyvykkyyttä. Arvioinnin jälkeen prosessivaiheet pisteytetään toteutuneen kypsyystason mukaan ja tulosten perusteella voidaan arvioida eri prosessivaiheiden kehitystarvetta. Mikäli jokin osa-alue on maturiteetiltaan tai suorituskyvyltään muita al-

haisempi, kehityskohteeksi voidaan ottaa sen nostaminen seuraavalle tasolle. Tavoitteeksi voidaan myös asettaa kokonaisprosessin maturiteetin nostaminen nykytasoa ylemmäs tai tietylle määritellylle kypsyydentalle (Software Engineering Institute 2010). Koska tässä tutkielmassa pyritään löytämään prosessin jatkuvaan parannukseen soveltuvia työkaluja, CMMI –mallista hyödynnetään ensisijaisesti jatkuvan kehityksen mallin maturiteettitasoja. Nelitasoisen mallin vaiheet ovat SEI:n määritelmän mukaan seuraavat:

Taso 0: Epätäydellinen. Prosessi ei ole toimiva tai toimii vain osittain.

Taso 1: Toimiva. Prosessi toteuttaa lopputuotteensa valmiiksi saamisen edellyttämät työvaiheet ja tehtävät.

Taso 2: Johdettu. Täyttää tason 1 kriteerit. Sen lisäksi kaikki prosessialueeseen liittyvä työ noudattaa organisaatiossa määritellyjä käytäntöjä. Tekemisessä mukana olevilla henkilöillä on kaikki työn valmiiksi tekemiseen tarvittavat työkalut ja resurssit käytössään. Asian omistajat (engl. stakeholders) ovat aktiivisesti osallisina prosessissa, kaikki tekemiset ovat valvottuja, kontrolloituja ja katselmoituja ja niiden prosessinmukaisuus on arvioitu.

Taso 3: Määritelty. Kaikki 2 tason kriteerit täyttyvät. Sen lisäksi prosessi noudattaa organisaation yleisiä prosessikäytäntöjä, prosessi on dokumentoitu ja se tuottaa tuloksia, mittareita ja muuta prosessikehityksessä tarvittavaa tietoa organisaatiolle.

Kun prosessialue on saavuttanut kyvykkyydessään tason 3, voidaan sitä edelleen kehittää vaiheittaisen mallin maturiteettitasoille 4 ja 5, jotka on määritelty seuraavasti:

Taso 4: Määrällisesti johdettu. Kaikki alempien tasojen kriteerit täyttyvät. Sen lisäksi prosessialue on kontrolloitu ja sitä kehitetään mittauksen ja määrällisten arvioiden perusteella. Laadun ja prosessin suorituskyvyn määrälliset mittarit on asetettu asiakastarpeen mukaisesti ja niitä käytetään prosessin johtamiseen.

Taso 5: Optimoitu. Kaikki 4 tason kriteerit täyttyvät. Sen lisäksi prosessin tehokkuutta kehitetään jatkuvasti ja prosessi on aina sopeutettu ja optimoitu määrällisten arvojen mukaan vastaamaan asiakastarpeita.

Usein prosessia yritetään kehittää niin, että opitaan aiemmista projekteista ja kehitetään niissä tehtyjen havaintojen perusteella projektin toteuttamisessa käytettyä prosessia. Tavoitteena on, että parannettua prosessia noudattamalla lopputulos olisi aiempaa parempi tulevissa projekteissa. Näiden kokemuksesta ja aiemmista projekteista opittujen asioiden vieminen prosessissa taaksepäin ja jalkauttaminen opituksi siten, että seuraavissa projekteissa asiat sujuisivat paremmin, ei kuitenkaan näytä olevan kovin tehokasta. Standish Groupin tutkimusten mukaan onnistuneiden projektien prosenttiosuus pysyy vuodesta toiseen varsin alhaisella tasolla.

Tutkimuksessaan projektin suorituskyvyn ja onnistumisen esteistä Paul Bannerman (2013) totesi projektin onnistumisen yleisesti määritellyjä faktoreita olevan mm. johdon sitoutuminen, liiketoiminnan tiivis osallistuminen, tavoitteiden selkeys, vaatimusten pysyvyys ja muuttumattomuus, projektin tai toteutuksen sisällön minimoiminen, projektijohdon kokemus, muutoshallinnan ja käytettyjen työkalujen sopivuus. Toisaalta onnistumisen tekijöitä on Bannermanin mukaan luokiteltu kyvykkyyksien perusteella sekä yksilön että organisaation näkökulmasta, jolloin näkökulmana on, että oikeat ihmiset oikein ohjattuna johtavat projektissa todennäköisimmin onnistuneeseen lopputulokseen. Vastaavasti prosessinäkökulmasta oikean toimintatavan oikea noudattaminen on projektin lopputuloksen laadun mahdollistaja. Koska perinteisesti projektin onnistumista on arvioitu näistä em. näkökulmista, on johtopäätös projektin epäonnistuessa ollut se, että ei tehty oikeita asioita oikein ja päätetään ”ottaa opiksi ja tehdä ensi kerralla paremmin”. Tämä ei kuitenkaan ole kovin tehokasta, koska opit ja havainnot saattavat usein olla niin tilannesidonnaisia tai tiettyihin olosuhteisiin liittyviä, ettei niistä ole mitään opiksi otettavaa. Toisaalta opittavaa tai kehitettävää voi olla niin paljon, vaikeaa tai jopa mahdotonta tunnistaa, mitä pitäisi ”tehdä paremmin”. Deming (1981) toteaa: ”best efforts are not sufficient” eli parhaansa tekeminen on välttämätöntä, mutta se ei vielä riitä pyrittävässä parantamaan toiminnan laatua ja prosessin tehokkuutta ja tuottavuutta. Mikäli te-

kemistä ei määrätietoisesti johdeta kohti parempaa laatua, yksittäisten henkilöiden ja ryhmien toimiminen ”paremmin” johtaa helposti siihen, että kukin toimija kehittää tekemistä omaan suuntaansa eikä lopputulos parane.

Pressmanin (2005:13–14) mukaan eräs ohjelmistokehitysprojekteihin liittyvä myytti on, se että myöhässä olevan projektin aikataulua voisi kuroa kiinni lisäämällä projektiin uusia ohjelmoijia tai muita resursseja. Todellisuudessa uusien henkilöiden tuominen mukaan kesken projektin on ennemminkin omiaan hidastamaan prosessia entisestään, koska mukana aiemmin olleiden asiantuntijoiden ajasta osa kuluu uusien henkilöiden perehdyttämiseen ja tiedon jakamiseen. Tällöin varsinaiseen projektityöhön käytettävissä oleva aika vähenee ja todennäköisempää siis on, että uusien tekijöiden lisääminen projektiin saa aikataulun myöhästymään entisestään. Ohjelmistokehityksen kehittymisen perustana Pressman (2005:22) painottaa pitkäjänteistä laatuun keskittymistä ja laadukkaaseen tekemiseen sitoutumista koko organisaatiolta. Total Quality Management, Six Sigma ja muut vastaavat laatuideologiat luovat kulttuurin jatkuvalla laadun ja prosessien kehitykselle, joka johtaa hänen mukaansa edelleen entistä paremman ohjelmistokehitysmallin kehittymiseen.

4. CASE: KANSAINVÄLISEN YRITYKSEN OHJELMISTOKEHITYS- PROSESSIN KEHITTÄMINEN - TUTKIMUKSEN SUUNNITTELU

Tutkimuksessa perehdytään erään kansainvälisen yrityksen yhden liiketoimintayksikön ohjelmistokehitysprosessiin. Lähtötilanteessa kaikki prosessin osapuolet totesivat, että sen hetkinen toiminnan laatu ja tehokkuus eivät vastaa sen paremmin IT:n kuin liiketoiminnankaan tarpeita. Töiden eteneminen kestää kauan, ne tuntuvat jumittuvan kehitysprosessiin ja kun ne viimein saadaan prosessista ulos, löydetään hyväksyntätestausvaiheessa todella paljon virheitä, jotka palauttavat tekemisen prosessissa takaisinpäin. Koska yrityksen tarvitsema ohjelmiston koodaustyö ostetaan kokonaan ulkoisilta toimitajilta, on kehityksen kannalta kriittisintä keskittyä siihen, mitä tilataan ja miten teke mistä ohjataan. Eli tutkimuksessa keskitytään yrityksen sisäisen IT-yksikön ja liiketoiminnan väliseen yhteistyöhön ja kehitysprosessin ohjausmalliin.

Tutkimuksessa on analysoitu projektien nykytilaa perehtymällä kehitystöiden etenemisen seurantaan ja eri vaiheiden ajallisiin kestoihin. Tiedot on kerätty kehitystöistä ylläpidetystä viikkoseurantaraportista, pääasiassa vuodelta 2013. Viikoittainen seurantaraportti on toteutettu MS excel taulukolla, jossa ylläpidettävät tiedot ovat:

- työn yksilöivä numero
- työn nimi
- toteutusvastaava IT organisaatiossa
- työn vaihe ja tilanne (etenee, odottaa jne.)
- tuotantoon viennin suunniteltu päivä
- hyväksyntätestauksen aloittamisen tavoitepäivä, jotta toteutus voidaan viedä tuotantoon suunniteltuna päivänä
- vastuutaho, jonka käsittelyssä työ on raportointiviikolla (liiketoiminta, IT, toteutusalihankkija, hyväksyntätestaus)
- toteutuksesta vastaavan IT-alihankkijan nimi
- hyväksyntätestauksen arvioitu kesto aika eli aikavaraus, joka testaukselle minimissään on jätettävä testin aloittamisen ja tuotantoon viennin väliin

Data-aineistossa ovat mukana kaikki ne tutkielman kohteena olleen liiketoimintayksikön kehitystyöt, joita on edistetty vuoden 2013 aikana eli vuoden 2013 aikana valmistuneet, aloitetut ja työn alla olleet projektit. Niiden projektien osalta, jotka oli aloitettu vuoden 2012 puolella, kerättiin etenemistiedot ko. vuoden dokumenteista. Tietoja kerättiin viikkoon 6/2014 saakka. Eli aineistossa on mukana projektit, jotka ovat valmistuneet viimeistään helmikuun alussa (viikolla 6) vuonna 2014. Tästä seurantataulukosta on tutkimusta varten poimittu erityisesti kehitystöiden tilannetietoa viikkotasolla. Kerättyjen tietojen perusteella pyritään löytämään riippuvuuksia sekä tunnistamaan kehityskohteita, joilla prosessin kyvykkyyttä (mm. työn läpimenoaikaa) saadaan parannettua. Kun prosessin kyvykkyys paranee, paranee samalla myös sen laatu ja kustannustehokkuus (vrt. Deming 1981).

Tutkimusta on täydennetty keskusteluin, joissa kartoitettiin prosessin osapuolten näkemyksiä siitä, millainen prosessin pitäisi tavoitetilassaan olla ja millaisena kukin toimija prosessin nykytilan näkee. Lisäksi selvitettiin, mitä tavoitetilan toteutuminen tarkoittaa eri osapuolten näkökulmasta eli mitä kukin prosessiin osallistuva tarvitsee ja edellyttää prosessin muilta tekijöiltä, jotta tavoitetilaan päästään. Samoin keskusteluissa koottiin lisätietoja yksittäisten kehitystöiden osalta, kun todettiin, ettei dokumentoiduista tiedoista saanut kaikilta osin riittävästi tietoa. Keskusteluja käytiin yhteensä kahdeksan (8) henkilön kanssa.

Käytyjen keskustelujen avulla luotiin myös pohjaa tulevalle kehitystyölle, jossa ihmisten motivaatio oman toiminnan muuttamiseen on ensiarvoisen tärkeää. Keskustelujen yksi keskeinen tavoite oli havahduttaa henkilöt huomaamaan, miten jokaisen tekeminen ja tekemisen oikea-aikaisuus on ensiarvoisen tärkeää kokonaisuuden onnistumiselle. Tavoitetilan määrittämisen yhteydessä haastateltavilta kysyttiin myös nykytilasta ja heidän näkemyksistään sen parantamiseksi, mitkä ovat keskeisimmät ongelmat ja mihin pitäisi erityisesti kiinnittää huomiota.

5. TUTKIMUKSEN TOTEUTTAMINEN

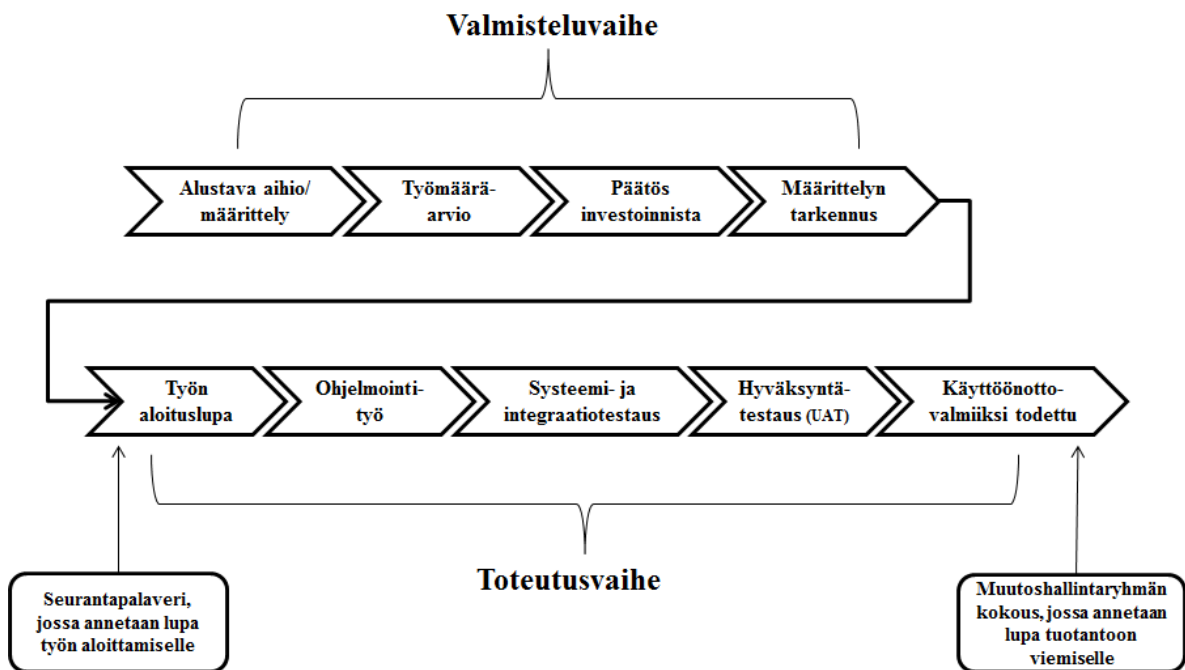
Tutkimus käynnistettiin keskustelulla liiketoiminnan keskeisimpien sidosryhmien pääliiköiden kanssa. Keskustelussa määriteltiin ylätason tavoitetilä, mitä prosessilta halutaan ihannetilanteessa ja tunnistettiin nykyisen prosessin todennäköisiä pullonkauloja ja ongelmakohtia sekä nimettiin henkilöitä, joiden kanssa keskustelemalla saa täydentävää tietoa ja näkemyksiä tutkimustyön tueksi.

5.1 Prosessin kuvaus

Liiketoimintayksikön nykyisen ohjelmistokehitysprosessimallin karkean tason kuvaus on esitetty kuvassa 3. Varsinainen kehitysprosessi alkaa siitä, kun määrittelydokumentaation perusteella pyydetään IT toteutuksen tekevältä yhteistyökumppanilta arviot työmääristä investointipäätöksen tekemistä varten. Ennen työmääräarviota liiketoiminnan ja IT:n edustajat sekä tarvittava joukko asiantuntijoita eri organisaation osista ovat kokoontuneet useita kertoja. Näissä tapaamisissa he ovat työstäneet määrittelydokumentit eli kuvanneet sen, mitä on tarkoitus tehdä ja miten toteutettavan kokonaisuuden pitää toimia.

Tarkastelussa prosessi on jaettu kahteen osa-alueeseen: valmistelemaan työhön ja varsinaiseen toteutukseen. Vaikka määrittelyjen tekeminen on oleellinen osa kokonaisprosessia, se on tässä analyysissä rajattu tutkimuksen ulkopuolelle. Kehityshankkeen toteutuksen kestoajassa ei siis ole huomioitu määrittelyvaiheen kestoä eikä sitä edeltävän kaupallisen suunnitteluvaiheen kestoä. Määrittelytyöstä on laskettu kokonaiskestoaikaan ainoastaan se osa, joka on tehty investointipäätöksen jälkeen (käytännössä sisältää määrittelyjen tarkennustyön). Tähän ratkaisuun päädyin siksi, koska ennen investointipäätöstä tehdyn määrittelytyön kesto ei ole dokumenteista todennettävissä. Yhtäläillä on mahdoton todentaa, milloin kehityshankkeen kaupallinen suunnittelu on alkanut. Kehityshankkeiden ideoita saatetaan kehitellä eri työryhmissä pitkiäkin aikoja, ennen kuin ne etenevät yksityiskohtaisempaan suunnitteluun. Näiden suunnittelu- ja ideointivaiheiden

tietojen selvittäminen muista lähteistä olisi vaatinut yksittäisten henkilöiden omien muistiinpanojen ja kalenterimerkintöjen perusteella tehtävää selvitystä. Totesin, että ohjelmistokehitysprosessin tehokkuuden näkökulmasta oleellisempaa on keskittää analysointi prosessiin, joka käynnistyy siitä, kun liiketoiminta luovuttaa tilauksen edistämisvastuun IT:lle (=valmisteluvaiheen alku). Tästä käynnistyvästä prosessista tutkitaan, miten tilaus etenee siihen pisteeseen, jossa IT luovuttaa tuotantoon vientiin valmiin työn liiketoiminnan käyttöön (=toteutuksen valmistuminen).

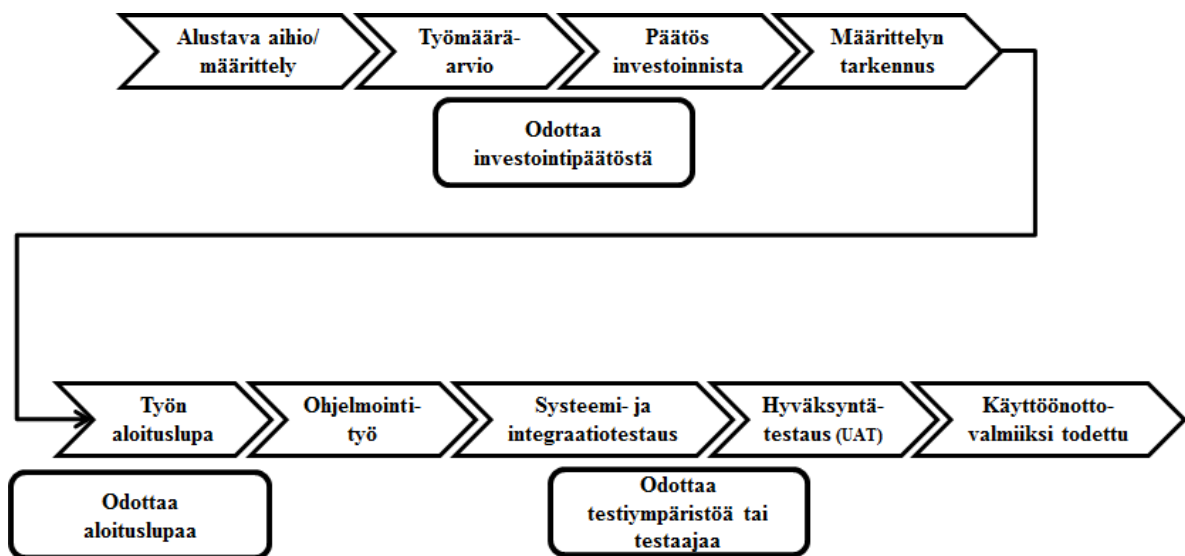


Kuva 3. Karkean tason prosessikuva työn etenemisestä.

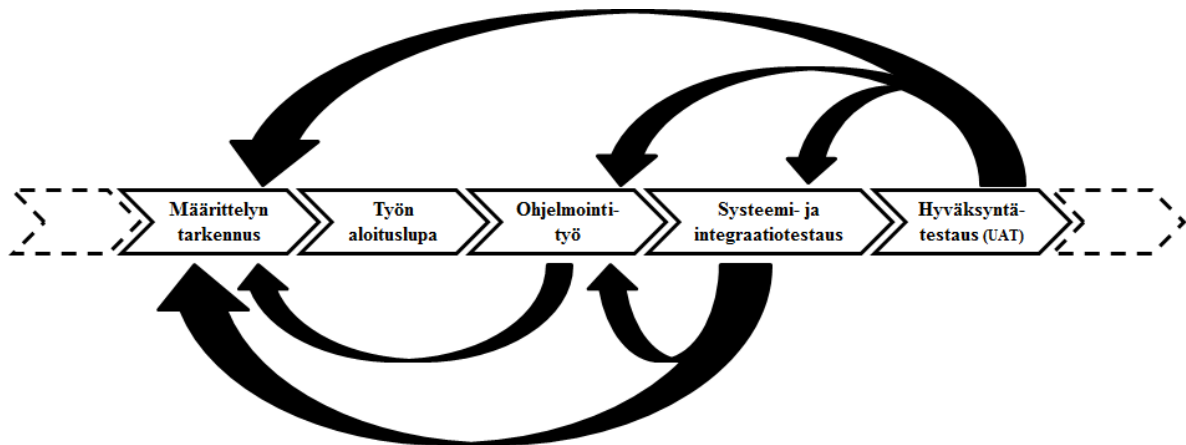
Prosessin tehokkuuden ja toiminnan analysoinnissa on huomioitu myös mahdolliset joutovaiheet, joissa työ ei etene, vaan odottaa pääsyä seuraavaan käsittelypisteeseen. Tässä prosessissa odottamaan voi joutua lähes mitä tahansa vaihetta, mutta seurannan kannalta oleellisimpia odottamisen paikkoja ovat rahoituksen odottaminen (=investointipäätös), työn aloitusluvan odottaminen, testiympäristön vapautumisen odottaminen, hyväksyntätestausajan ja testausresurssien odottaminen sekä tietojärjes-

telmien muutostöiden jäädytysajasta kesällä (n. 4 viikkoa) ja jouluna (n. 2 viikkoa) mahdollisesti aiheutuva odotusaika. Muutostöiden jäädytysaika vaikuttaa lähinnä käyttöönottoihin ja testaukseen. Meneillään oleva ohjelmointityö alihankkijalla jatkuu yleensä normaalisti etenkin kesätauon ajan, joulun pyhäpäivinä myös ohjelmointityö on tauolla.

Jonotuksen lisäksi prosessissa voi esiintyä uudelleen tekemistä (rework), jossa joudutaan palaamaan prosessiketjussa takaisinpäin johonkin aiempaan vaiheeseen. Tyypillisimmin kyse on toteutuksen aikana havaitusta määrittelytarkennusten tarpeesta tai testauksessa löytyneistä merkittävistä virheistä, joiden vuoksi työ palautuu kokonaan tai osittain takaisin toteutukseen. Kuvassa 4 on havainnollistettu jonotuksen kohdat prosessissa (pois lukien kalenteriin sidotut jäädytysajat, jotka voivat osua mihin tahansa prosessivaiheeseen) ja kuvassa 5 vaiheet, joissa esiintyy uudelleen tekemistä.



Kuva 4. Odottaminen kehitysprosessissa.



Kuva 5. Uudelleen tekeminen kehitysprosessissa.

5.2 Tutkimusaineisto

Tutkimusaineistoksi koottiin tiedot liiketoimintayksikön ohjelmistokehitystöistä, jotka ovat olleet työn alla vuoden 2013 aikana. Kehitystyöt tällä liiketoiminta-alueella ovat tyypillisesti pitkiä, joten osa töistä oli käynnistynyt vähintään määrittelytarkennusten ja työmääräarvioiden selvittämisen osalta vuoden 2012 puolella. Töiden etenemistä seurattiin vuoden 2014 viikon 6 loppuun saakka. Seuranta päätettiin tuohon viikkoon, koska sen jälkeen seurantadokumentointiaineisto hajautui ja keskitetyn seurannan sijasta ryhdyttiin seuraamaan pienempiä osakokonaisuuksia. Samalla myös valmistauduttiin isoon organisaatiomuutokseen, jonka vaikutuksesta seurantaan tuli muutamien viikkojen epäjatkuvuus.

Aineiston kerääminen tehtiin manuaalisesti. Tiedonlähteenä käytettiin viikoittaisen priorisointi- ja etenemisseurantapalaverien muistioita ja v. 2013 käyttöönotettuja erillisiä seurantataulukoita, joista selvisi projektin vaihe kullakin viikolla. Näistä aineistoista voitiin mm. laskea eri prosessivaiheiden summittainen kesto. Kestoajat eri vaiheille on laskettu viikon tarkkuudella. Tiedot perustuvat muistioiden ja seurantataulukoiden kirjauksiin.

Vuonna 2013 työn alla olleiden kehitystöiden kokonaismäärä oli 38. Niistä 18 valmistui vuoden 2014 viikon 6 loppuun mennessä. Lisäksi 4 työtä valmistui osittain (vaiheittainen käyttöönotto osin valmis), 2 peruttiin ja 14 oli tarkastelujakson lopussa kesken. Analysoinnissa on käytetty valmistuneita töitä, joten datarivejä on 18. Yksi datarivi sisältää kehitystyön yksilöintitiedon (työn numero) sekä kunkin prosessivaiheen keston viikkoina. Data on laadultaan osin epätäydellistä ja todennäköisesti osin vinoutunutta. Kunkin kehitystyön kesto-aika oli tarkasti todennettavissa, mutta prosessivaiheiden keston osalta data on laadultaan heikompaa. Seurantataulukoihin ja muistioihin kirjattujen tietojen laatu vaihteli kirjaajan mukaan, koska kirjaukselle ei ollut olemassa yhtenäistä käytäntöä.

Lähteenä käytettyjen dokumenttien laadintatapa kehittyi tarkastelujakson aikana, joten seurannan tarkkuus parani oleellisesti vuoden 2013 keväästä eteenpäin. Silti on todettava, että merkinnöissä oli tulkinnanvaraa ja joissain tilanteissa asioiden etenemisen historiaa täytyi tarkistaa kehitystyössä mukana olleilta, heidän omiin muistiinpanoihinsa perustuvia tietoja hyödyntäen. Manuaalinen keräystapa oli todella hidas ja sitä mutkisti edelleen se, että tieto oli useassa eri dokumentissa. Asioiden kirjauskäytäntö oli epäyhtenäinen, mm. oli dokumentoitu työn odottavan hyväksyntätietien alkamista, vaikka testaus ei ollut mahdollista työn keskeneräisyyden tai testiympäristöongelmien vuoksi. Samoin löytyi esimerkkejä, joissa työn tilaksi oli merkitty työmääräarvion selvittämissä vaihe, vaikka työ oli jo toteutuksessa. Kaikissa tilanteissa ei ollut yksiselitteisen selvää, etenikö työ suunnitellusti vai vaadittiinko määrittelyjen tarkennuksia. Kaikkea uudelleen tekemistä vaatinutta työtä ei todennäköisesti dokumentoitu erikseen eli seurantataulukko oli merkitty, että ”työ on toteutuksessa”, mutta ei ollut merkintää siitä, että toteutus oli todellisuudessa viivästynyt määrittelytarkennustarpeen vuoksi. Samoin työn oli merkitty olevan hyväksyntätestauksessa, mutta siinä yhteydessä ei dokumentoitu, että testauksessa oli saman tien löytynyt virheitä, jotka käytännössä palauttivat työn takaisin ohjelmointiin. Nämä seikat ilmenivät prosessin osapuolten kanssa käydyissä keskusteluissa. Ne voitiin todentaa henkilöiden omista muistiinpanoista ja alihankkijan kanssa käydyistä kirjeenvaihdosta, mutta ne eivät näy kaikilta osin datassa.

Tutkimusaineiston graafinen yhteenveto, prosessin kehitystyökohtaisten kestoajkojen osalta, on esitetty taulukossa 1. Laskelmat on tehty Minitab ® ohjelmalla, joka laskee aineistosta tavanomaisimmat tilastolliset suureet eli otoksen keskiarvon, mediaanin ja keskihajonnan sekä testaa, onko aineiston jakauma normaali.

Normaalijakauman testaamiseen ohjelma käyttää Anderson-Darling normaalisuustestiä, joka on T.W. Andersonin ja D.A. Darlingin vuonna 1952 kehittämä menetelmä jakauman arviointiin. Menetelmä tutkii, onko otos peräisin painottuneesta jakaumasta (=nollahypoteesi). Testin luottamusväliksi on valittu 95 %, jolloin $\alpha=0,05$. Nollahypoteesi on, että jakauma ei ole normaalisti jakautunut, ja nollahypoteesin hyväksymisen kriteerinä on se, että p -arvo on pienempi kuin α . Kuten taulukosta 1 nähdään, p -arvo on 0,519, joten nollahypoteesi hylätään ja voidaan todeta, että aineisto on todennäköisimmin normaalisti jakautunut. Normaalilla jakaumalla on merkitystä tulosten analysoinnissa, koska mm. regressioanalyysi edellyttää analysoitavalta aineistolta riittävää normaalijakautuneisuutta.

Otoksen koko on aiemmin mainittu 18 kehitystyötä. Niiden keskimääräinen kesto aika viikkoina oli noin 37 viikkoa (36,889) ja keskihajonta on 18,4. Lyhin kehitystyö kesti 8 viikkoa ja kestoltaan pisimpään työhön kului aikaa 66 viikkoa.

Taulukko 1. Keskeisimpiä tilastollisia suureita kehitystöiden kestoajoista.

Datarivien määrä (kpl)	N = 18
Keskiarvo kokonaiskestolle (viikkoja)	36,889
Keskihajonta	18,410
p -arvo normaalisuustestille (Anderson-Darling)	0,519
Minimiarvo kokonaiskestolle (viikkoja)	8,000
1. kvartiili	16,500
Mediaaniarvo	38,500
3. kvartiili	54,500
Maksimiarvo kokonaiskestolle	66,000

5.3 Tutkimuksen suorittaminen

Tulosten tutkimisessa hyödynnettiin tilastollisia menetelmiä, joita sovellettiin data-aineistossa eri prosessivaiheille laskettuihin kestoaikoihin sekä muihin muuttujiin (mm. IT tiimi). Tilastolliset analyysit tehtiin Minitab® ohjelmistoa hyödyntäen. Lisäksi koottiin tietoa nykyisestä prosessista, dokumentointikäytännöistä ja ohjausmalleista. Tietoa koottiin sekä tutkimuksen aikaisella havainnoinnilla että prosessin osapuolia haastatteleamalla. Tätä haastatteluissa saatua, organisaation kokemukseen perustuvaa, tietoa hyödynnettiin myös tulosten tulkinnassa ja johtopäätösten teossa. Näiden havaintojen ja analyysien lopputuloksena syntyy kehityssuunnitelma, jolla prosessia voidaan tulevaisuudessa kehittää. Kehityssuunnitelman suositusten viitekehyksenä on hyödynnetty CMMI –mallin jatkuvan kehityksen tasojen määrittelyä ja pyritty löytämään sellaisia suosituksia, joita toteuttamalla prosessin seuraava kypsyystaso olisi saavutettavissa.

6. TUTKIMUSTULOKSET

Tutkimuksen tuloksena saatiin tietoa sekä nykyisestä prosessista ja sen kestoajasta että nykyisen prosessin käytännöistä mm. dokumentointiin, päätöksentekoon ja toimintatapoihin liittyen. Tulokset perustuvat kehitystöiden etenemisen seurantalavereissa ylläpidetyistä seurantataulukoista koottuihin tietoihin sekä prosessin avainhenkilöiden kanssa käytyihin keskusteluihin. Havaintoja on tehty sekä prosessin käytännöistä että töiden tiloista ja etenemisestä.

6.1 Havainnot etenemisen seurannan käytännöistä prosessissa

Tutkimuksessa käytetty aineisto kerättiin manuaalisesti. Tiedon lähteenä käytettiin prosessin ohjaukseen liittyvien viikoittaisten palaverien dokumentointia. Projektidokumentation osalta havaittiin, että dokumentoinnin taso vaihteli tarkastelujakson aikana. Etenkin v. 2012 materiaaleissa oli havaittavissa epä johdonmukaisuuksia, inhimillisiä virheitä ja tulkinnanvaraisia sisältöjä. Yksittäinen kehitystyö saattoi muistiossa olla merkittynä sekä työmääräarviovaiheeseen että toteutuksen alla olevaksi. Toteutuslistalle merkittiin työt jo siinä vaiheessa, kun ne oli päätetty toteuttaa eikä kaikissa tilanteissa voitu todentaa, milloin toteutustyö todellisuudessa pääsi alkamaan. Lisäksi kehitystyö saattoi puuttua jonkin viikon listalta kokonaan eli se oli edellisellä viikolla merkitty toteutuksessa olevaksi ja seuraavalta viikolta tieto puuttui kokonaan. Tässä yhteydessä on syytä todeta, että osa virheistä ja epätäsmällisyyksistä oli seurausta siitä, että toimintamalli oli juuri käynnistetty. Systemaattinen seuranta töiden etenemisestä aloitettiin vuoden 2012 aikana. Selvää on, että oppimisen ja seurannasta kertyneen kokemuksen myötä dokumentointitapakin kehittyi. Vuoden 2012 lopussa dokumentointiin otettiin erillisen muistion rinnalle MS Excel –taulukko, jossa ”työmääräarviota odottavat” ja ”toteutukseen siirretyt” työt olivat aiempaa selkeämmin esillä. Itse taulukon sisältö oli alkuvaiheessa hyvin pelkistetty ja sisälsi yksilöivän numerotunnisteen ja työn nimen lisäksi suppean merkinnän työn vaiheesta. Tavallista oli, että vaihetta ei päivitetty joka viikolla,

vaan esimerkiksi ”pyydetty lisäselvityksiä” –tieto saattoi olla kehitystyön tilatietona useamman viikon ajan.

Alkukevään 2013 aikana dokumentoinnissa panostettiin aiempaa enemmän taulukko-muotoisen yhteenvedon laatimiseen. Tilatietojen päivityksessä oli havaittavissa selkeää kehitystä, mutta edelleen yksittäiset työt saattoivat jäädä useiksi viikoiksi epämääräisiin välitiloihin, joista ei voinut yksiselitteisesti tulkita, etenikö työ normaalisti vai ei. Jokaisesta käsittelykerrasta tehtiin oma taulukko, ts. edellisen viikon statustiedot korvattiin uusilla, ja tallennettiin uusi viikkotaulukko työryhmätallennustilaan (MS Sharepoint työtila). Yksittäisen työn etenemisen seuraaminen tai todentaminen jälkikäteen onnistui ainoastaan avaamalla kaikki yksittäiset taulukot ja tarkistamalla tilanne erikseen jokaisen viikon osalta.

Loppuvuoden 2013 aikana seuranta kehittyi edelleen niin, että samaan taulukkoon kerrottiin pidempää historiaa töiden etenemisestä, jolloin uusi tilatieto lisättiin taulukkoon edellisten tilatietojen perään. Jokaisesta käsittelykerrasta tallennettiin edelleen oma tiedostonsa, joten tietojen muuttumisen pystyi tarvittaessa varmentamaan kahden peräkkäisen viikon tiedostoja vertaamalla. Nyt kuitenkin syntyi jokaiselle työlle aiempaa havainnollisempi näkymä ja mahdolliset töiden etenemisen pysähtymiset tulivat taulukosta esille selkeämmin kuin vuoden alkupuoliskolla.

Tietojen päivittämiseen liittyvät käytännöt olivat vaihtelevat. Yleisesti voidaan todeta, että tiedon sisällön laatu ja määrä kasvoivat vuoden 2013 loppua kohden mentäessä. Sisällön laadun kasvaminen tarkoitti sitä, että kirjatusta tilatiedoista pystyi yhä useammin selvittämään kunkin työn tarkan tilan. Jokaisen viikon kohdalle oli merkitty, missä vaiheessa työ oli. Myös tilatietojen määrä kasvoi ja yksisanaisten ”toteutuksessa” – ilmaisujen rinnalle kirjattiin täydentävää tietoa toteutuksen tilasta. Tässä tutkimuksessa eräänä tutkimuskohteena olleen uudelleen tehtävän työn (engl. rework) analysoinnissa kirjausten laatu vaikuttaa todennäköisesti havaintoihin ja tuloksiin. Uudelleentehtävää työtä oli raportoitu 56 prosentissa valmistuneista töistä. Maininta hyväksyntätesteissä havaittujen virheiden aiheuttamasta uudelleen tehtävästä kehitystyöstä oli kirjattu vain 28 prosentille töistä ja määrittelyyn palaamisesta ja tarkennuksista aiheutuvaa uudelleen

tehtävää työtä oli 39 prosentissa töistä. Molempien osa-alueiden (määrittely ja testaus) uudelleen tehtävää työtä oli hieman yli 11 prosentissa kehitystöissä. Prosessin osapuolten kanssa käytyjen keskustelujen perusteella on kuitenkin ilmeistä, että uudelleen tehtävää työtä oli sekä testausten yhteydessä että määrittelyssä ollut merkittävästi enemmän. Lähes jokaisen työn yhteydessä testauksessa löytyi jotain virheitä, jotka piti korjata ja testata uudelleen ennen tuotantoon vientiä. Samoin keskusteltavien omien muistiinpanojen mukaan määrittelyiden tarkennuksia tehtiin useampaan työhön kuin mitä dokumentoidun tiedon perusteella oli luettavissa. Koska etenkin uudelleen tehtävän työn dokumentoinnissa on ollut suurta kehitystöiden välistä vaihtelua tarkkuudessa, jätetään sen osalta analyysi ylätasolle. Samoin kaikkia uudelleen tehtävään työhön liittyviä havaintoja voidaan datan laadun vuoksi käsitellä vain viitteellisinä.

Kirjaustarkkuuteen ja informaatioisisältöön oli prosessin osapuolten mukaan vaikuttanut myös kehitystyön tärkeys ja kiinnostavuus liiketoiminnan puolella. Niissä kehitystöissä, joihin liittyi liiketoiminnan erityinen intressi, oli työn etenemisestä kirjattu yksityiskohdaiset kuvaukset mm. siksi, että liiketoiminnan henkilöt voivat saada mahdollisesti tarvitsemansa tarkemmat tiedot suoraan seurantataulukosta. Toisaalta ne ovat myös olleet töitä, joiden etenemisestä oli keskusteltu seurantapalaverien yhteydessä laajimmin, joten tietoa tilanteesta (ja siten myös dokumentoitavaa tietoa) on ollut käytettävissä keskimääräistä enemmän. Jos taas jokin kehitystyö oli sellainen, joka pitää tehdä, mutta ei erityisesti kiinnosta liiketoiminnassa tai jonka tilaaja ei osallistunut töiden etenemistä koskeviin keskusteluihin, se oli vaarassa jäädä vähemmälle huomiolle. Silloin sen työn etenemistä ei välttämättä seurattu yhtä tarkasti kuin kiinnostavampana pidetyn työn etenemistä. Sen, että etenemistä ei aktiivisesti seurata ei välttämättä tarvitse merkitä työn toteutuksen hidastumista, mutta aineiston perusteella työn viivästymisen riski kasvoi, koska vähäiseksi jääneen seurannan myötä myöskään työn osapuolten välinen kommunikointi ei ollut aktiivista. Työ saattoi odottaa jotain tarkennusta tai lisämäärittelyitä, mutta puutteellisen viestinnän vuoksi lisätietojen saanti viivästy, mikä pidensi työn kokonaiskestoa.

6.2 Havainnot töiden tiloista ja etenemisestä prosessissa

Tutkimuksen tuloksena todettiin, että vuonna 2013 työn alla olleista kehitystöistä valmistui viikon 06/2014 loppuun mennessä 53 % eli 18 työtä yhteensä 38 meneillään olleesta kehitystyöstä. Näistä valmistuneista töistä valtaosa (89 %) oli yksittäisiä kehitystöitä ja loput (11 %) kahden tai useamman kehitysaihion yhdistelmiä. Kehitystöiden yhdistäminen oli tehty tyypillisesti tilanteessa, jossa kahdella kehitystyöllä oli voimakas keskinäinen riippuvuus kehitystyön sisällön suhteen. Tämä tarkoitti sitä, että työt piti tehdä joko samanaikaisesti tai tietyssä järjestyksessä ja siksi ne oli toteutuksen kannalta mielekästä yhdistää yhteen tarkastelukokonaisuuteen. Toinen syy töiden niputtamiseen oli nk. ”kakkosvaiheilmio”. Tällä tarkoitetaan tilannetta, jossa toteutuksen käynnissä ollessa huomataan, että on tarve kehittää jotain lisää tai laajentaa käynnissä olevaa muutostyötä merkittävästi laajemmaksi kuin, mitä on alun perin määritelty. Vaikka tämä havaittu muutostarve on käytännössä uusi kehitysaihio, on organisaatiolle tyypillistä, että uudet ominaisuudet yritetään saada toteutukseen alkuperäisen kehitysaihion vaiheena 2. Vaikka kakkosvaiheen toteuttaminen vaatii uuden rahoituspäätöksen ja käsittelyn, eli käytännössä ihan samat valmistelevat vaiheet ja etenemisen kuin ihan erillinen työkin, se halutaan yhdistää aiemmin aloitetun työn kanssa. Todennäköinen motiivi tälle on se, että ajatellaan työn saavan korkean prioriteetin tai etenevän nopeammin, kun se liitetään olemassa olevan työn yhteyteen tai jatko-osaksi. Jatko-osien liittäminen tarkoittaa sitä, että työt liitetään seurannassa yhteen ja kokonaiskestoajaa mitataan ensimmäisen osan alusta alkaen, mikä vähintäänkin tilastointi- ja seurantamielessä pidentää töiden kokonaiskestoja ja luo vaikutelman, että asioita valmistuu vähän. Toisaalta tutkimuksissa (mm. Fenech *et al.* 2013; Hashim *et al.* 2013) on todettu, että parhaat onnistumisen mahdollisuudet ovat silloin, kun työ on kohtuullisen kokoinen ja mahdollisimman tarkasti määritelty.

Suurin osa (61 %) valmistuneista töistä oli aloitettu vuonna 2012 ja suurin osa niistä myös valmistui 2013 (89 % kaikista tarkastelujaksolla valmistuneista töistä). Kaikkiaan 39 % töistä sekä aloitettiin että saatiin valmiiksi vuonna 2013 ja 11 % valmistuneista vaati yli vuoden työstöajan sillä ne aloitettiin v. 2012 ja valmistuivat vasta 2014 alku-

vuonna. Valmistuneiden töiden jakautuminen aloitus- ja valmistumisvuosien mukaan on havainnollistettu taulukossa 2.

Taulukko 2. Vuonna 2013 työn alla olleiden, ja tarkastelujakson kuluessa valmistuneiden, töiden jakautuminen aloitus- ja valmistumisvuoden mukaan.

Tuotantoon viedyt työt (18 kpl)	Aloitusvuosi		Yhteensä
	Valmistumisvuosi	2012	
2013	50 %	39 %	89 %
2014	11 %	0	11 %
Yhteensä	61 %	39 %	100 %

Taulukossa 3 on esitetty valmistuneiden töiden kokonaiskestoajat viikkoina, jonossa olon aika viikkoina sekä jonossa olon prosenttiosuus työn kokonaiskestosta. Maksimijonotusaika 62,5 % oli lyhytkestoisella muutostyöllä, joka valmistui juuri ennen kesäaikana olevaa tietojärjestelmien muutosten jäädytysjakson alkua. Se joutui odottamaan tuotantoon vientiä kesätauon ajan. Lähes yhtä suuri jonotuksen osuus (59 %) oli työllä, joka jätettiin odottamaan hyväksyntätestausta yhteensä n. 7 kuukauden (28 viikkoa) ajaksi. Tämä odotusaika koostuu useammasta lyhyemmästä odotusjaksosta. Prosessin osapuolten haastattelussa selvisi, että kyseinen työ oli siirretty hyväksyntätestauksen jonoon vaiheessa, jossa useita kriittisemmäksi luokiteltuja töitä piti saada edistettyä ensin, joten se jäi testausjonoon odottamaan vuoroaan. Kun työ tuli jonotuksen jälkeen viimein testausvuoroon, selvisi, ettei toteutus olekaan testausvalmis, vaan työ piti palauttaa takaisin toteutukseen, josta se palautui takaisin testausjonoon odottamaan jälleen vuoroaan kiireellisemmiksi luokiteltujen töiden keskelle. Tämä toistui vuoden 2013 aikana useamman kerran, mistä kertyi hyväksyntätestauksen jonoajaksi laskettavaa jonotusta tuo edellä mainittu 7 kuukautta.

Taulukko 3. Valmistuneiden töiden kestoaja, jonossa olon aika ja jonotuksen osuus kokonaiskestoajasta.

Valmistuneet työt (18 kpl)	minimi	maksimi	keskimäärin
Kokonaiskestoaja (viikkoa)	8	66	36,7
Jonossa oloaika (viikkoa)	1	39	12,2
Jonossa olon osuus (% kestopäivästä)	7,7 %	62,5 %	30,5 %

Töiden tarkempi jakautuminen vaiheisiin on esitetty taulukossa 4. Siinä on kuvattu valmistuneiden töiden kokonaiskestoajan prosentuaalinen jakautuminen valmisteluun, toteutukseen, testaukseen ja käyttöönottoon sekä odottamiseen. Taulukosta voidaan todeta aiemmin mainittu kirjauskäytännön epätasaisuus, joka näkyy datassa siten, että osassa töistä ei ole tehty datan perusteella mitään valmistelutöitä, vaan työ on mennyt suoraan toteutukseen. Yleensä työlle tehdään kuitenkin vähintään määrittelyjen tarkistus ja katselmointi sen jälkeen, kun se on tullut IT:n työlistalle ja ennen toteutuksen aloitusta. Prosessin osapuolten kanssa käytyjen keskustelujen perusteella valmistelutyötä pitäisi olla jokaisella työllä jonkin verran.

Taulukko 4. Työn jakautumisen vaihtelu valmistuneissa töissä (18 kpl).

Vaihe	min	max	keskiarvo
Valmistelutyö	0,0 %	25,0 %	7,2 %
Toteutus	18,3 %	74,1 %	48,2 %
Testaus ja käyttöönotto	1,5 %	46,4 %	14,0 %
Jonotus	7,7 %	62,5 %	30,5 %

Testaus- ja käyttöönotto on yhden työn kohdalla vienyt melkein puolet kokonaiskestoajasta. Kyse on varsinaisen ohjelmointityön toteutuksen keston mukaiselta kokoluokaltaan melko pienestä työstä, jonka osalta testaus jouduttiin suorittamaan useita kertoja uudelleen ja määrittelyjä tarkentamaan monta kertaa työn aikana. Tällä työllä myös uudelleen tehtävän työn (rework) osuus on suhteellisesti mitattuna korkea. Peräti 75 %

työn kokonaiskestoajasta oli tuplatyötä. Prosessin osapuolten kanssa käydyissä keskusteluissa ilmeni, että tämän työn ongelmien todennäköinen aiheuttaja oli määrittelytyössä. Työtä aloitettaessa arvioitiin, että kyseessä on pieni muutos, joten määrittely tehtiin nopeasti, ja jälkikäteen arvioiden liian epätarkasti. Tämä johti lukuisiin tarkennustarpeisiin toteutuksen aikana ja siihen, että hyväksyntätestaus keskeytyi useita kertoja testauksessa ilmenneiden isojen virheiden vuoksi. Kuten Hashim *et al.* (2013) tutkimuksessaan ohjelmistokehitysprojektien onnistumisen kriittisistä tekijöistä toteavat, merkittävimmät riskitekijät projektin onnistumiselle ovat projektin sisällön muuttuminen ja huono kommunikointi osapuolten välillä.

Jonotuksen osuus prosessin kestosta on valmistuneilla töillä ollut vähintään 7,7 % kokonaisajasta. Lisäksi jonotusaika on keskiarvolla mitaten suuri, joten sitä tarkastellaan tarkemmin jäljempänä. Jonossa olon ajan ja kokonaiskeston välillä on voimakas riippuvuus, mikä on jo loogisestikin ajateltuna ilmeistä. Pearsonin korrelaatiokerroin näiden tekijöiden välillä on 0,801, mikä tarkoittaa varsin voimakasta positiivista lineaarista riippuvuutta. Kahden satunnaismuuttujan (X,Y) välinen Pearsonin korrelaatiokerroin kuvaa muuttujien välistä lineaarista riippuvuutta. Kerroin lasketaan jakamalla näiden muuttujien välinen kovarianssi muuttujien keskihajontojen tulolla eli

$$\rho(X,Y) = \text{cov}(X,Y) / \sigma_X \sigma_Y. \quad (1)$$

Korrelaatiokerroin saa arvonsa välillä +1 ja -1. Jos kertoimen arvo on 0, muuttujien välillä ei ole korrelaatiota. Arvo 1 kertoo voimakkaasta positiivisesta (+1) tai negatiivisesta (-1) korrelaatiosta muuttujien välillä. Lisäksi p-arvo korrelaatiokertoimelle on 0,000 joka on pienempi kuin 0,05 (α arvo 95 % luottamusvälillä). Eli voidaan todeta, että näiden tekijöiden välillä on tilastollinen riippuvuus. Sitä vastoin uudelleen tehdyn työn määrä (rework) ei korreloi lineaarisesti kokonaiskeston kanssa. Keston ja uudelleen tehdyn työn välinen korrelaatio on vain 0,086 ja p-arvo reilusti suurempi kuin 0,05, joten tilastollisesti merkittävää korrelaatiota ei niiden välillä ole. Tässä yhteydessä on hyvä muistaa aiemmin tässä luvussa todettu epäily uudelleen tehdyn työn seurannan luotettavuudesta, millä saattaa olla vaikutusta tuloksiin.

Muuttujien keskinäinen suhde voidaan todeta myös regressioanalyysin avulla. Regressioanalyysissä pyritään aineistolle löytämään pienimmän neliösumman menetelmällä paras sovite ts. sovittamaan suora pistejoukkoon (lineaarinen regressio). Käytännössä lasketaan estimaatit regressioyhtälön

$$y_i = \alpha + \beta x_i + \varepsilon_i \quad (2)$$

kertoimille α ja β siten, että residuaalivirheen ε_i neliöiden summa minimoidaan. Tutkimusaineistosta analysointiin työn kestoajan (kesto) ja työn jonossa oloajan lineaarista regressiota eli kuinka hyvin jonoaika selittää työn kokonaiskestoaikaa. Regressioanalyysin tulokset on esitetty taulukossa 5.

Taulukko 5. Regressioanalyysi jonoajan vaikutuksesta kokonaiskesto aikaan.

	kerroin	kertoimen keskivirhe	p-arvo
Vakiotermi	19,434	4,219	0,000
Jonoaika	1,4346	0,268	0,000

$$R^2 = 64,2 \%$$

Taulukosta voidaan todeta, että työn kestoajan ja jonoajan välinen regressioyhtälö on:

$$\text{Kesto aika} = 19,434 + 1,4346 * \text{Jono aika}.$$

Jonoaika siis selittää aineistossa 64,2 % kehitystyön kokonaiskestosta (R^2) ja koska p -arvo jonoaika –kertoimelle on pienempi kuin 0,05, saatu tulos on tilastollisesti merkittävä. Eli regressioyhtälön perusteella tämä dataotos viittaa siihen, että yksi viikko jonossa pidentäisi kehitystyön kokonaisaikaan tässä prosessissa lähes 1,5 viikolla (jonoajan kerroin regressioyhtälössä = 1,43).

Taulukosta 6 nähdään, että kun regressioanalyysiin lisätään jonossa olon ajan lisäksi myös uudelleen tehty työ, nousee yhtälön selitysaste (R^2) vain reilulla kahdella prosenttiyksiköllä. Tämä todentaa saman, joka jo korrelaatiokertoimen laskennassa todettiin eli uudelleen tehdyllä työllä ei ole data-aineiston perusteella tilastollisesti merkittävää vaikutusta työn kokonaiskestoajaan.

Taulukko 6. Regressioanalyysi jonoajan ja uudelleen tehtävän työn (rework) vaikutuksesta työn kokonaiskestoajaan.

	kerroin	kertoimen keskivirhe	p-arvo
Vakiotermi	18,806	4,636	0,001
Jonoaika	1,431	0,2756	0,000
"Rework"	0,1983	0,5181	0,707

$$R^2 = 64,5 \%$$

Tutkielman kohteena olleet kehitystyöt keskittyvät suurelta osin yhteen järjestelmään, sillä valmistuneista 18 työstä 14 kpl oli tehty tähän järjestelmään. Näitä töitä koodaa kaksi eri yhteistyökumppanin tiimiä, joten voidaan tarkastella myös, miten koodaava tiimi vaikuttaa kestoajaan ja uudelleen tehtävän työn määrään. Tätä on tutkittu varianssianalyysillä eli selvitetty poikkeavatko kokonaiskestoajojen keskiarvot tilastollisesti merkittävästi toisistaan tiimien välillä (taulukko 7). Tiimit on merkitty tunnuksilla A ja B.

Taulukko 7. Ohjelmointitiimien välinen ero työn keskimääräisessä kokonaiskestoajassa.

Tiimi	N (kpl)	keskiarvo	keskihajonta
A	8	38,63	20,31
B	6	39,83	15,07

$$R^2 = 0,12 \%$$

$$p\text{-arvo} = 0,905$$

Taulukosta 7 voidaan todeta, että selitysaste (R^2) on pieni ja p –arvo suuri. Varianssianalyysin nollahypoteesi on, että kahden ryhmän keskiarvot ovat samat eikä niiden välillä ole tilastollista merkittävyyttä. Nollahypoteesi hylätään, jos p –arvo on pieni (alle 0,05), joten tässä analyysissä nollahypoteesi hyväksytään ja todetaan, ettei ohjelmointitiimillä ole tilastollisesti merkittävää vaikutusta kehitystyön kokonaiskestoan tutkimuksen kohteena olleissa kehitystöissä.

Taulukkoon 8 on koottu tulokset samasta testistä IT-tiimin ja uudelleen tehtävän työn (rework eli prosessin aikainen työn palauttaminen aiempaan työvaiheeseen) kokonaismäärän välillä. Tulosten perusteella näyttää siltä, että tiimin B töillä uudelleen tehtävän työn määrä on suurempi kuin tiimissä A. Selitysaste on 22,66 % ja p -arvo on vain 0,085, joka on suhteellisen lähellä 95 % luottamusvälin α arvoa (0,05), jota pienemmällä p -arvolla nollahypoteesi voitaisiin hylätä. Tutkimusaineiston pieni otos (vain 14 työtä tässä analyysissä) johtaa kuitenkin siihen, että yksittäinen poikkeama ja yksittäisen työn suuri uudelleen tehtävän työn määrä saattaa merkittävästi vaikuttaa tuloksiin. Johtopäätösten tekemiseen tarvitaan isompi aineisto, etenkin kun aiemmin on jo todettu, ettei uudelleen tehtävän työn kirjauskäytäntö ole aukoton. Inhimillisistä syistä on todennäköistä, että uudelleen tehtävää työtä on raportoitu systemaattisemmin ja täsmällisemmin sellaiselle työlle, jonka aikataulussa pysymisen seurantaan on ollut erityinen intressi liiketoiminnan puolelta. Eli uudelleen tehtävän työn osalta ei voida luotettavasti osoittaa olevan eroa eri ohjelmistotoimitusalihankkijoiden välillä, mutta laatu ja kustannustehokkuusmielessä tämä voisi olla mielenkiintoinen seuranta- ja tutkimuskohde jatkossa.

Taulukko 8. Uudelleen tehtävän työn määrän keskiarvo ohjelmointitiimeittäin.

Tiimi	N (kpl)	keskiarvo	keskihajonta
A	8	1,500	2,330
B	6	7,000	7,950

$$R^2 = 22,66 \%$$

$$p \text{ -arvo} = 0,085$$

6.2.1 Jonossa oloajan analysointia

Kuten edellä taulukossa 3 esitettiin, jonossa oloajan osuus prosessin kokonaiskestosta on merkittävä. Koska tutkielman tavoitteena on löytää kehityskohteita, jotka parantavat prosessin tehokkuutta myös läpimenoajan osalta, tarkastellaan seuraavaksi lähemmin jonossa oloon liittyviä havaintoja.

Jonottamisen syyt voidaan jakaa viiteen pääryhmään. Työ odotti:

- 1) investointipäätöstä
- 2) toteutuksen aloittamista
- 3) testausympäristön vapautumista
- 4) hyväksymistestauksen aloittamista
tai
- 5) työ oli pysähdyksissä liiketoimintapäätöksellä (sisältää tietojärjestelmien jäädytysajan, jolloin muutokset eivät ole sallittuja).

Jonotuksen syiden jakauma ja kunkin jonotustyyppin keskimääräinen vaikutus kokonaiskestoon on kuvattu taulukossa 9. Siitä voidaan todeta, että yleisimmin työ on pysähdyksissä tietojärjestelmien jäädytysajasta johtuen, mikä on odotettavissa oleva havainto, koska työt kestävät tyypillisesti yli puoli vuotta. Eli joko kesäajan tai joulun jäädytysaika vaikuttaa lähes jokaiseen työhön. Tässä kohden on syytä jälleen todeta myös datan manuaalisen kirjauskäytännön ja vaiheiden sisällön kirjauksen tarkkuus. Mikäli seuranta-aulukoista ei voitu todeta työn edenneen jäädytystauon aikana, tulkittiin, että se on ollut pysähdyksissä. Tämä saattaa joidenkin töiden kohdalla olla virheellinen tulkinta. Myöskään muulla liiketoimintapäätöksellä keskeytettyjä töitä, joissa keskeytys johtui muusta syystä kuin järjestelmien jäädytysajasta, ei pystytä kaikilta osin erittelemään datasta. Vähintään 17 % töistä oli sellaisia, joissa liiketoiminnan päättämisen jonotuksen syynä oli jokin muu päätös kuin tuotantoon vientiä rajoittava järjestelmien jäädytysaika.

Rahoituksen, toteutuksen aloituksen ja hyväksyntätestauksen aloituksen jonottaminen oli lähes yhtä yleistä, mutta mikäli työ joutui odottamaan hyväksyntätestauksen aloittamista, jonotuksen vaikutus kokonaiskestoon oli merkittävästi suurempi kuin toteutuksen

aloituksen tai investointipäätöksen odottamisen vaikutus. Tämä selittyy loogisesti odotettavan työvaiheen kestolla. Investointipäätöksen tekeminen on yksittäinen tapahtuma siinä, missä hyväksyntätestaus saattaa kestää useita työpäiviä ja jopa viikkoja, joten testauksen odottamisesta aiheutuva viive työn kokonaiskesto on siitä syystä pidempi. Testiympäristön vapautumista tai tuoreuttamista odotti data-aineiston mukaan vain 11 % töistä ja jonotusajan keskimääräinen vaikutus oli vähäinen. Täydentävissä keskusteluissa saadut tiedot ovat tämän havainnon kanssa ristiriidassa, koska testiympäristöstä johtuvia vähintään viikon viiveitä oli todettu v. 2013 töissä lukuisia kertoja. Todennäköistä on, että testiympäristön vapautumista tai tuoreuttamista odottava työ on tilaseurannassa kirjattu hyväksyntätestauksen aloitusta odottavaksi ilman, että on eritelty tarkempaa syytä, miksi odottaa testausta. Tämän seurauksena hyväksyntätestauksen jonossa on siis sekä testausresurssia että testiympäristöä odottavia töitä, joita ei pystytä käyttävissä olevien tietojen perusteella täysin erittelemään.

Taulukko 9. Jonotuksen vaiheet, jakauma ja vaiheiden keskimääräinen vaikutus kesto-aikaan.

Vaihe, jota jonotettiin	Investointipäätös	Toteutuksen aloitus	Testiympäristön vapautuminen	Hyväksyntätestauksen aloitus	Liiketoimintapäätös tai jäädytysaika
Kuinka suuri osa (%) kaikista valmistuneista jonotti ko. vaiheessa	39%	39%	11%	44%	83%
Jonotusajan keskimääräinen %-osuus kokonaiskestoajasta	3,70%	3,30%	1,20%	10%	12,30%

6.2.2 Pitkät ja pitkittyneet työt

Töiden kokonaiskestoajoja tarkastellessa todettiin, että valmistuneiden töiden kokonaiskestoajassa on suurta vaihtelua (minimi 8 viikkoa maksimi 66 viikkoa). Vaihtelu selittyy osin töiden erilaisella kokoluokalla, on luonnollista että isompi muutostyö vaatii

enemmän aikaa kuin pienet muutokset. Kestoajan eroja syntyy myös keskenään samaa kokoluokkaa olevien töiden välille ja silloin merkittävänä kestoajan pidentymiseen vaikuttavana tekijänä on jonotusajan määrä. Koska tavoitteena on löytää keinoja, joilla voidaan lyhentää prosessin kokonaiskestoa, on pisimpään kestäneitä töitä havainnoitu erikseen.

Valmistuneista töistä 5 työtä (n. 28 % valmistuneista) oli kestoajaltaan yli vuoden mittaisia. Kesto aika näissä töissä vaihteli 54 viikosta 66 viikkoon. Taulukossa 10 on esitetty prosessin päävaiheiden keston jakautuminen näiden töiden osalta. Tästä havaitaan, että kahdessa työssä jonotusaika on ollut merkittävin syy valmistumisen viivästymiseen. Yhdessä työssä sekä jonotus että testauksen venyminen ovat pidentäneet kokonaiskesto-aikaa, ja kahdessa työssä kyse on ollut laajasta toteutuksesta eli varsinainen toteutusaika muodostaa valtaosan (yli 74 %) työn kestoajasta. Lisätietoa näille havainnoille saatiin prosessiosapuolten kanssa käydyistä keskusteluista. Työ numero 3 odotti pitkään testausresursseja, kun kiireellisemmiksi ja tärkeämmiksi arvioituja töitä ohitti sen testausvaiheeseen siirryttäessä. Kun testaamaan viimein päästiin, löytyi toteutuksesta useita korjausta vaativia asioita ja testausta jouduttiin tekemään useita kertoja. Dokumentaation mukaan tähän työhön ei ole käytetty valmisteluaikaa ollenkaan sen jälkeen, kun työ on saanut rahoituspäätöksen ja toteutusluvan. Kyse voi olla puutteellisesta tilatietojen dokumentoinnista, koska mahdollinen valmistelutyö on ajoittunut v. 2012 puolelle, jolloin töiden tilatietojen kirjauskäytäntöä vasta harjoiteltiin eikä se kaikilta osin ollut yhtä kattava kuin vuoden 2013 aikana. Tämän työn kohdalla myös organisaatiomuutokset ja vastuuhenkilöiden vaihdokset tunnistettiin jälkikäteen työn etenemistä hidastaneiksi tekijöiksi. Kaksi työtä (työt 1 ja 4), joissa jonotusaika oli pitkä, päätyivät jonotuslistalle prosessin eri vaiheissa liiketoiminnan päätöksellä eli tärkeämpiä ja nopeammin eteneviä töitä laitettiin tietoisesti päättäen näiden kahden työn edelle.

Taulukko 10. Yli vuoden kestäneiden töiden kokonaiskestoajan jakautuminen prosessivaiheisiin.

	Valmistelu	Toteutus	Testaus/käyttöönotto	Jonotus
1	6,1%	33,3%	1,5%	59,1%
2	6,5%	74,2%	1,6%	17,7%
3	0,0%	18,3%	40,0%	41,7%
4	0,0%	42,9%	7,1%	50,0%
5	3,7%	74,1%	3,7%	18,5%

6.2.3 Toteutuneet kestoajat ja alkuperäiset työmääräarviot

Kun työstä on tehty alustava määrittely ja kuvattu, mitä on tarkoitus toteuttaa, tehdään toteutuksesta työmääräarvio. Varsinaisen ohjelmointityön tekijäyritys arvioi saamansa määrittelyn perusteella ohjelmointityön laajuuden eli antaa työmääräarvion ja hinnan varsinaiselle toteutustyölle. Sen jälkeen päätetään työn toteuttamisesta tai toteuttamatta jättämisestä. Päätöksen teossa työmääräarvion suuruudella on keskeinen rooli, koska nämä kustannukset kehitystyössä ovat ulkoisia kuluja eli talosta ulos maksettavaa rahaa. Kehitystöiden yhteydessä seurataan myös sisäisten resurssien käyttöä ja työlle kohdistuvia kustannuksia, jotta toteutuksen todellinen hinta olisi laskettu mahdollisimman realistisesti, mutta ulkoisen työn kustannukset ovat painavin peruste toteutuksesta päätettäessä: arvioidaan onko ohjelmointityöhön satsattu euromäärä muutoksen arvoinen.

Data-aineistossa olevien töiden alkuperäisiä työmääräarvioita verrattiin toteutuneisiin työmääriin. Tiedot poimittiin käsin ja yksitellen ohjelmointialihankkijayritysten antamista kirjallisista toteutusarvioista. Näitä tietoja verrattiin viikkoraportoinnin yhteydessä kirjattuihin toteutuneisiin ohjelmointityön kestoaikoihin. Vertailua väärästi hieman se, että työmääräarviot annetaan työpäivinä ja töiden etenemisen seurannan tarkkuuden taso oli viikkotasolla. Mahdollinen töiden aloitus ja lopetus keskellä viikkoa saattaa siis johtaa siihen, että viikkoseurantareportoinnissa työn kesto aikaan tulee ylimääräinen viikko. Tätä virhettä pyrittiin eliminoimaan pyöristämällä työmääräarviot ylöspäin täysiksi viikoiksi (työpäivien arvioitu määrä jaettiin luvulla 5 ja tulos pyöristettiin ylöspäin).

Kun työmääräarvioita verrattiin raporttien perusteella tilastoituun toteutuneeseen ohjelmointiaikaan, todettiin jokaisen työn kohdalla kuluneen enemmän aikaa kuin työmääräarviossa oli ennakoitu. Lähimmäksikin osuneessa arviossa todellinen työmäärä oli 20 % ennakoitua suurempi. Kun toteutuneesta ohjelmointityön kestoajasta vähennettiin uudelleen tehdyn työn osuus, näyttivät arviot osuvan hieman paremmin kohdalleen ja noin 38 % töistä työmääräarvio ja toteutuma olivat ± 10 % vaihteluvälin sisällä. Uudelleen tehdyn työn osuus sisälsi siis työn käynnistymisen jälkeisiin määrittelytarkennuksiin kulu- neen ajan (jolloin ohjelmointi oli käynnissä, mutta kenties hidastunut tai viivästynyt tar- kennustarpeiden takia) sekä hyväksyntätestien aikaisten virheiden korjauksen ajan. Kun nämä kaksi ”häiriötekijää” poistettiin, arviot olivat osassa töistä jo merkittävästi pa- remmin kohdalleen osuvia. Mutta silti aineistossa oli paljon töitä, joissa toteutunut työ- määrä oli moninkertainen arvioon verrattuna. Kuten aiemmin todettiin, uudelleen tehtä- vän työn kirjauskäytännöt ovat olleet hyvin vaihtelevat ja niitä ei useiden töiden kohdal- la ole erikseen seurattu, minkä vuoksi työmääräarvioiden ja toteumien laajempi vertailu ei ole mielekästä tulosten satunnaisuuden vuoksi.

Työmääräarvion ja työn toteutuneen kokonaiskeston välillä ei ole todettavissa riippu- vuutta, koska työn kokonaiskeston näyttävät vaikuttavan merkittävimmin muut tekijät kuin varsinaisen ohjelmointityön kesto, kuten esimerkiksi toteutusten priorisointi ja hy- väksyntätestauksen resurssit. Pearsonin korrelaatiokerroin työmääräarvion ja kehitys- työn kokonaiskeston välillä saa arvon 0,150 ja P-arvo on todella suuri (0,723). Erikseen tarkasteltuna edellisessä kappaleessa esillä olleet pitkittyneet työt (kokonaiskesto yli 52 viikkoa) eivät tässä työmääräarvioiden ja toteutumien vertailussa poikenneet muista. Tau- lukossa 9 esitetyistä kehitystöistä numeron 4 kohdalla työmääräarvio ylittyi vain 15 %, joten viivästyminen aiheutui selkeästi muista syistä. Toisaalta samassa taulukossa työ numero 2 ylitti alkuperäisen työmääräarvionsa 23 -kertaisesti. Tämä esimerkkityö osal- taan vahvistaa käsitystä kirjausten epätasmuällisyydestä, koska uudelleen tehtävää työtä ei tällä työllä ole raporttien mukaan ollut ollenkaan (ei siis esim. määrittelytarkennuksia työn alkamisen jälkeen), ja silti muutaman viikon kestäväksi arvioitu ohjelmointityöko- naisuus oli seurantadokumentoinnin perusteella yli 40 viikkoa toteutuksessa. Toden- näköinen juurisyy tälle arviointivirheelle on lähtötilanteen määrittelyjen epätasmuällisyys

tai työn aikana tapahtunut merkittävä määrittelyiden muutos (joka ei ole dokumentoinnista tosin todennettavissa).

7. JOHTOPÄÄTÖKSET

Tutkimuksen kohteena olevassa it-kehitystöiden prosessissa on suurta vaihtelua, mikä todennäköisesti aiheuttaa sekä ylimääräisiä kustannuksia että vaihtelevaa laatua (vrt. Deming 1981). Vaihtelua oli sekä kehitystöiden kokonaiskestoajoissa että jonotuksen määrässä eli ajassa, jolloin työ odotti jotain eikä edennyt. Kokonaiskesto aika oli keskimäärin 36 viikkoa: lyhimmillään 8 viikkoa ja enimmillään 66 viikkoa. Jonottamiseen kului keskimäärin 30 % kokonaiskestoajasta: vähimmillään noin 7 % ja enimmillään yli 62 %. Jonottaminen pidensi työn kokonaiskestoaa keskimäärin hieman yli 12 viikkoa eli noin yhden vuosineljänneksen verran.

Prosessi ei myöskään ole kovin tehokas tai tuottava, koska yli vuoden kestäneen tarkastelujakson aikana valmistui vain 18 työtä, mikä oli hieman yli puolet v. 2013 työn alla olleista kehitystöistä (53 %). Valmistuneista töistä 11 % eli yhteensä viisi (5) työtä vaati yli vuoden työstöajan. Näistä töistä kahdessa jonotuksen osuus oli vähintään puolet työn kokonaiskestoajasta ja vähimmilläänkin jonotusta oli melkein viidennes (18,5 %) työn kestoasta.

Hyväksyntätestauksen prosessin parantamisella voidaan vaikuttaa prosessin läpimenoajan lyhentämiseen. Testausresurssia joutui odottamaan 44 % töistä ja tästä odotuksesta tuli työlle keskimäärin 3-4 viikon lisäviive, koska testauksen odottamisen viive oli n. 10 % keskimääräisestä työn kokonaiskestoajasta. Lisäresursoinnilla ei välttämättä olisi voitu ratkaista kaikkia testausresursseihin liittyviä ongelmia. Kuten mm. Pressmann (2005) toteaa, lisäresurssien tuominen käynnissä olevaan projektiin ei nopeuta työn etenemistä vaan pikemminkin vaikuttaa hidastavasti, kun projektissa mukana olevien aikaa kuluu uusien tekijöiden perehdyttämiseen. Mutta joustavuuden kasvattaminen siinä mielessä, että testattavia kokonaisuuksia voidaan jakaa tehtäväksi aiempaa laajemmalle tekijäjoukolle, tehostaisi töiden valmistumista. Myös testausympäristöstä johdettavista ongelmista tuli ylimääräistä viivästystä työn etenemiseen. Tähän ongelmaan ratkaisun etsiminen on jo käynnistetty tutkimuksen aikana. Työn alla ovat erillisen hyväksyntätestausympäristön rakentaminen, edellä mainittu testaajaverkoston laajentaminen

ja hyväksyntätestien keventäminen. Aiemmassa toimintamallissa hyväksyntätestauksen yhteydessä on tehty myös merkittävässä määrin regressiotestejä, jotka ovat tehneet testauksesta melko raskasta sekä sitoneet vähäisiä hyväksyntätestausresursseja ja siten hidastaneet töiden valmistumista.

Kustannustehokas ja laadukas prosessi edellyttää prosessin johtamista, säännöllistä seuranta- ja työskentely- ja seurantamenetelmien kehittämistä. Tutkimuksessa havaittiin, että kehitystyö jää helposti pysähdyksiin ja viivästyy, mikäli sen etenemistä ei systemaattisesti ohjata. Tämä näkyi mm. data-aineistosta, jossa työ saattoi olla samassa vaiheessa useita viikkoja peräkkäin. Prosessiosapuolten kanssa käydyissä keskusteluissa selvisi, että siinä vaiheessa, kun tällaisen pysähtyneen työn perään lähdettiin erikseen kyselemään, oli tyypillisesti kyse siitä, että odotettiin määrittelytarkennusta liiketoiminnan edustajalta tai toteutettavuuteen liittyvää kommenttia tekniikan asiantuntijalta. Tiedustelu oli tehty sähköpostitse ja siihen oli vastaanottaja unohtanut vastata eikä kysyjä ollut asiasta muistuttanut. Asioita tulisi edistää ennalta sovituisissa palavereissa eikä sähköpostikeskusteluihin, jotta työskentely ja eteneminen olisivat tehokasta.

Selkeänä parannuksena ja prosessia nopeuttavana toimenpiteenä on jo vuoden 2014 alussa muutettu investointipäätöskäytäntöä siten, että pieniä ja keskikokoisia muutostöitä varten tehdään vuosineljänneskohtaiset investointipäätökset, jolloin jokaiselle yksittäiselle kehitystyölle ei tarvitse hakea rahoituspäätöstä erikseen. Tutkimusaineistossa havaituista rahoituspäätöksen odotusajoista ja viiveistä (39 % töistä odotti rahoituspäätöstä) on tästä johtuen päästy tähän päivään mennessä suurelta osin eroon.

Työmääräarviot näyttävät pysyvän parhaiten arvioidun mukaisena silloin, kun määrittelytyö on tehty hyvin. Mahdollista uudelleen tehtävän työn tarvetta ei työmääräarvioissa pystytä ennakoimaan, joten kesken toteutuksen tehtävät määrittelyiden tarkennukset ja muutokset sekä havaittujen virheiden korjaaminen hyväksyntätestien jälkeen nostavat jatkossakin työmäärät arvioitua suuremmiksi. Kustannustehokkuuden näkökulmasta on tavoiteltavaa, että työmääräarviot olisivat mahdollisimman realistiset ja oikeaan osuvat. Tästä syystä määrittelyn laatu ja täsmällisyys ovat olennainen kehityskohde kustannustehokkuuden parantamiseksi.

Mikäli tutkielman kohteena olevaa prosessia arvioidaan CMMI –mallin kriteerien perusteella (ks. luku 3), voidaan sen todeta nykytilassaan olevan tasolla 1. Prosessi tuottaa kyllä ne tuotokset, joita siltä odotetaan ja pääpiirteissään sen voidaan todeta olevan toimiva. Mutta kuten edellä on todettu, toiminnan laadussa ja johtamisessa on kehitettävää. Toimintamallit ovat kyllä organisaation määritellyn prosessimallin mukaiset, kuten tasolla 2 edellytetään, mutta tekemisten valvonta ja seuranta suhteessa määriteltyyn prosessiin eivät vielä ole tason 2 edellyttämällä tasolla. Myös asianomistajien aktiivisempaan mukanaoloon ja kommunikoinnin kehittämiseen eri osapuolten välillä tulee kiinnittää huomiota, jotta taso 2 saavutetaan. Kehityskohteista ja suosituksesta on laadittu erillinen yhteenveto, joka on tämän tutkielman liitteenä (liite 1). Siinä on huomioitu myös ne kehityshankkeet, joiden valmistelu on käynnistetty ennen tämän tutkielman valmistumista ja jotka ovat jo työn alla.

LÄHTEET

- Bannerman, P. L. (2013). *Barriers to project performance*. Hawaii International Conference on System Sciences (HICSS-46), p. 4324-4333. ISBN 978-1-4673-5933-7.
- Deming, W. E. (1981). Improvement of quality and productivity through action by management. *National Productivity Review* 1:1, 12-22. Wiley Online Library. ISSN 1520-6734.
- DSDM Consortium. *What is DSDM?* [online]. [Viitattu 27.3.2014]. Voittoa tavoittelemattoman järjestön DSDM yhteenliittymän julkaisema verkkosivusto. Saatavissa: <http://www.dsdm.org>.
- Eveleens, L. & Verhoef, C. (2010). The rise and fall of the chaos report figures. *IEEE Software* Jan/Feb 2010, 27:1, 30-36. ISSN 0740-7459.
- Fenech K. & De Raffaele, C. (2013). *Overcoming ICT project failures – a practical perspective*. World Congress on Computing and Information Technology (WCIT), 2013, 1-6. ISBN 978-1-4799-0460-0.
- Haikala I. & Märijärvi, J. (2004). *Ohjelmistotuotanto*. 10. painos. Helsinki: Talentum. ISBN 952-14-0850-2.
- Haikala I. & Mikkonen T. (2011). *Ohjelmistotuotannon käytännöt*. 12. Painos. Helsinki. Talentum. ISBN 978-952-14-1754-2.
- Hashim R., Abbas, M. & Hashim, M. (2013). *Critical success factors assessment in software projects*. London: The Science and Information Organization (SAI, <http://thesai.org>). Science and Information Conference, 2013, p. 282-287.
- IEEE Std 610.12-1990. *IEEE Standard Glossary of Software Engineering Terminology*. Institute of Electrical and Electronics Engineers. Saatavissa <http://ieeexplore.ieee.org.proxy.tritonia.fi/stamp/stamp.jsp?tp=&arnumber=15934>

- Mäkilä, T. (2012). *Software development process modeling. Developers perspective to contemporary modeling techniques*. TUCS Dissertations No 148. Turku: Centre of Computer Science. ISBN 978-952-12-2790-5
- Pfleeger, S. L. & Atlee, J. M. (2010). *Software Engineering*. 4th Edition. Boston, London: Pearson Education Inc. ISBN 978-0-13-814181-3.
- Pressman, R. S. (2005). *Software Engineering: A Practitioner's Approach*. 6th ed. New York: Mc Graw-Hill Companies Inc. ISBN 0-07-285318-2.
- A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*. (2008). 4th Edition. Newton Square (Pa.): Project Management Institute. ISBN 978-1-933890-51-7.
- Royce, W. (1970). *Managing the development of large software systems*. Technical Papers of Western Electronic Show and Convention (WesCon) August 25–28, 1970, Los Angeles, USA. Saatavissa myös <http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf>.
- Scrum.org. *About Scrum.org*. [online]. [Viitattu 27.03.2014]. Scrum –mallin kehitykseen osallistuneen Ken Schwaberin perustama yhteisö, joka edistää Scrum menetelmien jatkokehitystä ja käyttöä ohjelmistokehityksessä. Saatavissa <http://www.scrum.org>.
- Software Engineering Institute (2010). *CMMI® for Development, Version 1.3. Improving Process for Developing Better Products and Services*. [Verkkodokumentti]. Carnegie Mellon University. [Viitattu 10.7.2014]. Saatavissa: <http://cmmiinstitute.com/cmmi-solutions/cmmi-for-development>
- Standish Group (2014). *Definition of Project Success*. [Viitattu 10.7.2014]. Saatavissa <http://blog.standishgroup.com/news>.
- Watson, G. H. (2005). *Design for Six Sigma*. 1st edition, Salem: Goal / QPC. ISBN 1-57681-078-X.

LIITTEET

LIITE 1: Suositukset ohjelmistokehitysprosessin laadun ja kustannustehokkuuden parantamiseen

Suosituksissa tähdätään CMMI:n jatkuvan kehittämisen mallin 2 tasolle, jossa kaikki tekemiset ovat valvottuja, kontrolloituja ja katselmoituja ja niiden prosessinmukaisuus on arvioitu (ks. tarkemmin luku 3).

Kehitystöiden etenemisseurannan kehittäminen:

- Seurannan jakaminen pienempiin osa-alueisiin (alihankkijakohtainen töiden etenemisten seuranta) – toteutettu osin alkuvuoden 2014 aikana, laajentunut syyskuu 2014 koko IT alihankintaan
- Etenemisen dokumentoinnin kehittäminen siten, että:
 - o Meneillään olevien töiden kokonaisuuden seuranta on yksinkertaista, mutta antaa kuitenkin riittävän kuvan kokonaistilanteesta – toteutettu osin v. 2014 aikana siirtämällä seuranta Jira –järjestelmään, jossa käsitellään myös kehitysehdotukset
 - o Yksittäisen työn jumittuminen jonoon tulee selkeästi esille. Prosessivaiheen kestoajaksi on jo nyt nähtävillä työ kerrallaan, mutta kokonaisuus ei piirry kovin selkeäksi. Tämä vaatii jatkokehitystä, nyt on yksittäisten henkilöiden havainnointikyvyyn ja muistin varassa, seurantajärjestelmässä (Jira) ei vielä riittävällä tasolla erotu
 - o Alihankkijakohtaisten seurantapalavereiden dokumentoinnin sisällön yhtenäistämistä siten, että jokainen seurantapalaveri tuottaa aidosti lisäarvoa prosessille. Dokumentointi tulisi tehdä suoraan Jira -järjestelmään, jolloin ajantasainen tieto olisi loogisesti kunkin työn yhteydessä.
- Työmääräarviokäytännön laajentaminen kokonaiskestoajan ja valmistumisajan kohdan arviointiin siten, että alihankkijan kanssa yhteistyössä arvioidaan mahdollinen aloitusajankohta ja lasketaan sen perusteella testausajat ja tuotantoon

vientien huoltoikkuna-ajankohdat huomioiden todennäköinen valmistumisajankohta, jolloin ominaisuus on tuotannossa. Eli hyödynnetään projekteissa käytössä olevaa aikataulun muodostamisen mallia kevennettynä versiona myös pienkehitystöissä. Mahdolliset määrittelymuutokset kehitystyön aikana pitää ottaa aikatauluarvioinnissa huomioon ja valmistumisajankohta-arviota tulee päivittää vastaavasti.

- Pidetään työt kohtuullisen kokoisina, jotta saadaan valmista ulos riittävällä tahdilla. Mikäli kehitystyön aikana tunnistetaan tarve nk. kakkosvaiheelle, tehdään se, mutta tuodaan prosessiin uutena kehitysaihiona eikä liitetä väkisin alkuperäisen työn kylkeen ja viivästyttämään sen työn etenemistä ja loppuun saamista.

Testauksen osalta:

- Rakennetaan erillinen testiympäristö, joka varataan vain testauksen käyttöön (käynnissä)
- Hyväksyntätestaus palautetaan oikeaan laajuuteensa: IT:n puolella ja alihankkijoiden toimesta tapahtuvassa yksikkö-, systeemi- ja integraatiotestauksessa tulee riittävällä tasolla varmentaa, että toteutuksen toimivuus. Nämä testitulokset katselmoidaan ja vasta hyväksytyin katselmoinnin jälkeen työ voi edetä hyväksyntätesteihin. Koska hyväksyntätestaus on niukkojen resurssien vuoksi ollut pullonkaula, on oltava erityisen tarkka siitä, ettei testausjonoon viedä suotta odottamaan sellaisia töitä, jotka eivät ole aidosti valmiita hyväksyntätestaukseen.
- Hyväksyntätestauksen testausresurssien määrää kasvatetaan mm. hyödyntämällä myynnin ja tilausten käsittelyn ammattilaisia osa-aikaisina testaajina, isompiin projektikonaisuuksiin palkataan erillinen testaaja tai testaajaryhmä. Suurempi testaajien määrä antaa myös enemmän joustovaraa tilanteisiin, joissa testattavaa valmistuu määrällisesti paljon. Resurssivaraukset tehdään mahdollisimman tarkasti ennakkoon, jotta henkilöiden työpanos käytetään optimaalisesti.

Viestinnän ja tiedonkulun tehokkuuden kannalta

- Säännölliset palaverit osapuolten kesken (liiketoiminta, business IT, IT), joissa varmistetaan riittävän tiedon saaminen määrittelyihin ja toisaalta riittävä viestintä liiketoiminnan suuntaan työn etenemisestä. Vähintäänkin tuotealue / ohjel-

mointialihankkijakohtaisesti, isommissa kokonaisuuksissa tarvittaessa projekti-kohtaisesti.

- Kehitystyön sidosryhmien ja ryhmien luonteen tunnistaminen: kuka päättää, kenen pitää olla tietoinen. Päätöksentekoon osallistuvat kehitystyössä vain ne, joilla on valtuudet päättää, muita henkilöitä tiedotetaan. Jo päätettyä asiaa ei nosteta uudelleen keskusteluun ja päätetä uudelleen projektin aikana ilman erityisen painavaa syytä. Jos näin toimitaan, seurauksena on aina hyväksyttävä se, että kehitystyö tulee viivästyämään eikä viivettä saada kurottua kiinni projektin aikana.

Muut, yleiset huomiot ja suositukset:

- Myöhässä olevaa projektia ei saada vauhditettua projektin resursseja lisäämällä kesken projektin. Lisähenkilöiden tuominen projektiin todennäköisemmin vain hidastaa etenemistä, kun uusien henkilöiden opastamiseen kuluu aikaa. Mikäli halutaan tukea projektin sujuvaa etenemistä, keskitetään toimenpiteet projektin ulkopuolisten hidasteiden poistamiseen tai niiden vaikutusten minimointiin (mm. samoista tekijöistä kilpailevat kehitystyöt, ristiriitaiset vaatimukset muilta kehitysprojekteilta).
- Kehitystyön tavoitteet on pidettävä kohtuullisen kokoisena ja selkeästi rajattuna. Jos halutaan ketteryyttä kehitystöiden valmistumiseen, pitää muuttaa nykyistä pienempiä kokonaisuuksia kerralla. Etenkin määrittelytyön loppuvaiheessa, jossa usein kertyy runsaasti lisämäärittelyjä (ja usein myös ne vaikeimmin toteutettavissa olevat vaatimukset), on noudatettava erityistä tarkkuutta kehityskohteen rajaamiseen järkevän kokoiseksi.