

Hyväksymistestaus ohjelmistokehityksessä – Prosessi, tehtävät ja roolit

Lasse Keskitalo

Tietojenkäsittelyn koulutusohjelma

Tekijä tai tekijät Lasse Keskitalo	Ryhmä HETI
Raportin nimi Hyväksymistestaus ohjelmistokehityksessä – Prosessi, tehtävät ja roolit	Sivu- ja liitesivumäärä 59 + 10
Opettajat tai ohjaajat Kai Kivimäki	
<p>Tämän opinnäytetyön aiheena oli tutkia tietojärjestelmän hyväksymistestauksen johtamista ja prosesseja tietojärjestelmää käyttävän asiakkaan näkökulmasta. Toimeksiantaja tässä opinnäytetyössä oli Sofigate Oy. Opinnäytetyön aiheeksi valittiin hyväksymistestaus ja sen prosessit, koska Sofigaten kehittämä Tietohallintomalli ei sisällä ohjeita hyväksymistestaukseen. Myös valtaosa ohjelmistotestauksen kirjallisuudesta ja malleista on suunnattu toimittajan ohjelmistokehittäjille.</p> <p>Opinnäytetyön tavoitteena oli luoda käsitys siitä, miten yrityksen tulisi johtaa hyväksymistestauksia, jotta se voisi turvata liiketoiminnan jatkumisen ottaessaan käyttöön uusia järjestelmiä ja järjestelmäpäivityksiä. Tämän työn tavoitteena oli luoda hyväksymistestauksen prosessi ja tehtäväkuvaukset. Lisäksi oli selvitettävä mitä rooleja ja kompetensseja prosessin ja tehtävien suorittamiseen tarvitaan. Menetelmänä käytettiin teemahaastattelua ja kyselylomaketta. Opinnäytetyö tehtiin kevään ja kesän 2012 aikana.</p> <p>Tutkimuksen tuloksena syntyivät hyväksymistestauksen prosessikuvaukset, tarkemmat tehtäväkuvaukset, sekä testauspäällikön että testaajan roolikuvaukset. Teoriaosuudessa käytetyn kirjallisuuden, sekä haastattelujen vastauksien perusteella voidaan päätellä, että luodut prosessi-, tehtävä- ja roolikuvaukset pystytään soveltamaan käytäntöön.</p>	
Asiasanat Hyväksyminen, testaus, prosessit, roolit, ohjelmistokehitys	

Degree Programme in Information Technology

Authors Lasse Keskitalo	Group HETI
The title of thesis Acceptance testing in software development - Process, tasks and roles	Number of pages and appendices 59+10
Supervisor(s) Kai Kivimäki	
<p>The purpose of this thesis was to study the management of information system acceptance testing and its processes in software development from the customer's perspective. This thesis was commissioned by Sofigate Oy. The subject of this thesis was chosen because the ICT standard for Management book published by Sofigate Oy does not cover how the acceptance testing in a software development project should be executed and managed. Also literature about software testing and its processes is mainly focused on testing from the perspective of development, not from the customer's perspective. This despite the fact that the customer is responsible for executing and managing the acceptance testing.</p> <p>The aim of this thesis was to build understanding about how companies, using information systems, should manage acceptance testing in software development projects to ensure business continuity. The purpose was to define acceptance testing processes and task lists. In addition, the purpose was to clarify the roles and competences which are required to execute the defined processes and tasks. The method used was theme interview. The thesis was written during the spring of 2012.</p> <p>User acceptance testing process and detailed task lists were created as a result of the thesis. Also the role descriptions for the test manager and tester were created. Based on the conclusions of the study, the interviews and the theory, we can conclude that the processes, task lists and roles can be brought into practice.</p>	
Key words Acceptance, testing, processes, roles, software development	

Sisällys

1	Johdanto	1
1.1	Tausta.....	1
1.2	Tavoitteet.....	2
2	Testaus osana järjestelmäkehitystä	5
2.1	Yleistä testauksesta ja testauksen tasot	6
2.2	Hyväksymistestauksen sijoittuminen V-Modell® XT mallissa.....	11
2.3	Hyväksymistestauksen sijoittuminen Scrum-mallissa.....	17
3	Hyväksymistestaus	23
3.1	Hyväksymistestausprosessi ja siihen kuuluvat tehtävät	24
3.2	Keskeiset roolit, tehtävät ja vaatimukset	32
4	Toteutettu haastattelututkimus.....	36
4.1	Tausta ja perusteet.....	36
4.2	Tutkimuskysymykset ja perusteet.....	38
4.3	Haastattelujen toteutus	40
4.4	Tutkimusanalyysi	40
5	Yhteenveto ja pohdinta	51
5.1	Tulosten yhteenveto.....	53
5.2	Jatkotutkimuskohteet.....	54
	Lähteet.....	57
	Liitteet.....	60
	Liite 1. Suunnitteluvaiheen tehtävät.....	60
	Liite 2. Testiympäristön valmistelu ja ylläpito vaiheen tehtävät.....	61
	Liite 3. Teemahaastattelujen kysymysrunko	62
	Liite 4. Hyväksymistestauksen prosessi.....	63
	Liite 5. Hyväksymistestauksen pääprosessin tehtävät	64
	Liite 6. Hyväksymistestauksen testiympäristön hallinnan tehtävät.....	65
	Liite 7. Testaajan roolikuvaus	66
	Liite 8. Testauspäällikön roolikuvaus	67
	Liite 9. Käsitteet	68

1 Johdanto

Voiko kukaan enää kuvitella yritystä, jossa liiketoimintaprosesseja ei tukisi jokin toiminnanohjausjärjestelmä? Varmasti lähes jokaisella yrityksellä on jonkinlainen toiminnanohjausjärjestelmä jota käyttävät yrityksen omat työntekijät, että yrityksen asiakkaat. Esimerkiksi VR:n lipunmyyntiautomaatteja ja internetin kauppapaikkaa käyttävät päivittäin tuhannet asiakkaat ympäri Suomen ja tämän järjestelmän rikkouduttua kaaos on valmis. Vuonna 2011 VR:n järjestelmiin tehtiin muutoksia, mutta käyttöönottovaiheessa tuli suuria ongelmia. Liiketoiminta kärsi tuntuja tappioita, joiden suuruudesta VR ei halua puhua. VR on saanut tapauksen johdosta paljon risuja asiakkailtaan, mutta syyt järjestelmän toimimattomuudesta ja ongelmista on vieritetty järjestelmätoimittajille Tiedolle ja Accenturelle. Tuntematta yllämainitun esimerkin taustoja herää kysymys, miksi kaikki vastuu vieritetään toimittajille? Onhan hyväksymistestaus kuitenkin aina tilaajan vastuulla.

1.1 Tausta

Opinnäytetyön toimeksiantaja Sofigate Oy on tietohallinnon johtamisen asiantuntijapalveluja tarjoava, vuonna 2003 perustettu yritys, jonka liikevaihto vuonna 2010 oli 9,6 M€. Yrityksessä työskentelee n. 100 tietohallinnon ammattilaista, jotka toimivat pääkaupunkiseudun lisäksi Tampereella ja Oulussa. Sofigate on kehittänyt tietohallintomallin, jossa on kuvattuna koko tietohallinnon toimintakenttä. Tietohallintomallissa testaus ja tarkemmin hyväksymistestaus mainitaan osana projektijohtamista. Tietohallintomallin (Huovinen 2009, 84-87.) mukaan hyväksymistestauksen järjestämis- ja toteutusvastuu ovat asiakkaan ja tarkemmin käyttäjäorganisaation vastuulla, mutta mitään konkreettisia työkaluja sen johtamiseen mallissa ei anneta. Selvitän tässä opinnäytetyössä miten asiakasyrityksen hyväksymistestausta tulee johtaa.

Tämän opinnäytetyön tarkoituksena oli tutkia tietojärjestelmän hyväksymistestauksen johtamista ja prosesseja tietojärjestelmää käyttävän asiakkaan näkökulmasta. Tutkimuksen tarkoituksena oli luoda käsitys siitä miten yrityksen tulisi johtaa hyväksymistestausta ja miten se varmistaa liiketoiminnan jatkumisen ottaessaan käyttöön uusia järjestelmiä ja järjestelmäpäivityksiä. Tarkoitus oli tutkia mistä kokonaisuuksista asiakkaan vastuulla

oleva hyväksymistestaus koostuu ja minkälaisia voimavaroja ja kompetensseja tämän prosessin johtamiseen vaaditaan.

Tämän työn lopputuloksena syntyi kuvaus hyväksymistestauksen prosessista, sen tehtävistä, sekä prosessin suorittamiseen tarvittavista rooleista ja vastuista. Tutkimuksen tulokset on kuvattu työn tutkimusanalyysiosuudessa. Lisäksi opinnäytetyön tilaajan toiveesta on luotu erilliset mallipohjat seuraavasti:

1. Hyväksymistestauksen prosessikuvaus ja tehtävät (Liite 4.)
2. Hyväksymistestauksen prosessin tarkemmat tehtäväkuvaukset (Liite 5 & 6.)
3. Hyväksymistestauksen suorittamiseen tarvittavat roolit, vastuut ja vaadittu osaaminen (Liite 7.)

Näitä kuvauksia ja mallipohjia tullaan hyödyntämään Sofigate Oy:n Tietohallintomalli v2.0 kehittämisessä.

Tutkimustulosten toivotaan myös auttavan tietojärjestelmiä hyödyntävien yritysten tietohallinnon johtohenkilöitä ymmärtämään mihin kokonaisuuteen heidän tulee tietojärjestelmien hyväksymistestauksessa panostaa ja mitä testausprosessin johtaminen edellyttää.

Opinnäytetyössä käytetyt käsitteet on listattu liitteessä 9.

1.2 Tavoitteet

Tutkimuksen tavoitteena oli selvittää mikä on hyväksymistestauksen prosessi ja sen tehtävät, sekä missä ohjelmistokehitysprojektin vaiheessa asiakasorganisaation tulisi työt hyväksymistestausta silmällä pitäen aloittaa. Näiden lisäksi tuli selvittää mitä rooleja hyväksymistestauksen suorittamiseen ja johtamiseen tarvitaan, sekä mitä pätevyys- ja kyvykkyyksiä töiden suorittamiseen tarvitaan.

Yllämainituista lähtökohdista tämän opinnäytetyön asiakkaan kontaktihenkilön kanssa on laadittu seuraavat tarkentavat tutkimuskysymykset, joihin halutaan saada vastaukset.

- Mikä on hyväksymistestauksen prosessi ja mitä tehtäviä sen suorittamiseen kuuluu?
- Milloin asiakasorganisaation tulee aloittaa työt hyväksymistestausta silmällä pitäen?

- Onko hyväksymistestaus ainoa testaustaso, jota tilaajan tulee johtaa ja voiko eri testaustasojen testausta johtaa yksi ja sama henkilö?
- Onko valitulla ohjelmistokehityksen viitekehityksellä vaikutusta hyväksymistestauksen prosesseihin (ketterän kehityksen malli vs. V-malli)?
- Mitä rooleja hyväksymistestauksen tekemiseen ja johtamiseen tarvitaan, sekä mitä pätevyys- ja kyvykkyyksiä tähän vaaditaan?
- Mitä tehtäviä hyväksymistestauksessa asiakasyrityksen liiketoiminnan tulee tehdä ja mitä tehtäviä vastaavasti kuuluu tietohallinnolle?
- Mitä muita asioita hyväksymistestauksen organisoinnissa tulee ottaa huomioon?

Opinnäytetyön tutkimusosuudessa haastattelin testauksen ja tietojärjestelmäkehityksen ammattilaisia ja sain tarkennuksia ja kokemusperäistä tietoa teoriaosuuden löydöksiin. Haastattelujen perimmäisenä tavoitteena oli tuoda käytännön näkökulma tutkimukseen ja varmistaa, että laadittavat prosessikuvaukset eivät ole pelkkää teoriaa vaan että ne toimisivat myös käytännössä. Haastattelujen toivottiin myös tuovan ideoita ja vinkkejä tietojärjestelmän hyväksymistestauksen prosessin ja tehtävien tekemiseen ja johtamiseen.

Työn teoriaosuuden tarkoituksena on perehdyttää lukija yleisellä tasolla tietojärjestelmätestaukseen ja sen tasoihin, sekä antaa tarkempaa tietoa hyväksymistestauksen prosessista, tehtävistä ja siitä, miten hyväksymistestaus sijoittuu eri ohjelmistokehitysmalleissa. Lukijalla on kuitenkin hyvä olla jonkinlainen käsitys tietohallinnosta ja ohjelmistokehityksestä ennen tämän työn lukemista. Hyväksymistestauksen sijoittumista on tarkasteltu V-Modell XT järjestelmäkehityksen viitekehityksessä, sekä ketterän kehityksen Scrum -viitekehityksessä.

Lisäksi teoriaosuudessa haettiin vastauksia alla oleviin kysymyksiin.

- Miksi ylipäättensä tietojärjestelmiä pitää testata?
- Mitä tietojärjestelmätestaukseen kuuluu ja mitkä testauksen tasot ovat toimittajan ja mitkä asiakkaan vastuulla?
- Millainen on V-mallinen ohjelmistokehitysmalli ja miten hyväksymistestaus sopii kuvaan?

- Millainen on ketterän kehityksen ohjelmistokehitysmalli, kuten Scrum, ja miten hyväksymistestaus sopii kuvaan?
- Mitä on hyväksymistestaus ja mitkä ovat hyväksymistestauksen prosessit ja tehtävät?
- Mitä rooleja hyväksymistestauksen prosessien suorittamiseen tarvitaan?

2 Testaus osana järjestelmäkehitystä

Ohjelmistojen ja järjestelmien kehittäminen tehdään yleensä projekteina, joissa toimittaja toteuttaa asiakkaalta saadut vaatimukset. Ohjelmistotoimittajan kannalta projektin lähtökohtana ovat asiakkaan vaatimukset, kun taas asiakkaan vaatimukset kehitettävälle järjestelmälle tulevat yleensä liiketoimintatarpeista. Asiakas näkee projektin tuotokset paljon laajemmin kuin järjestelmätoimittaja ja projektit saattavatkin olla osa suurempaa kokonaisuutta kuten hanketta tai ohjelmaa. Asiakkaan kannalta on helpompaa jakaa suurempi hankekokonaisuus pienempiin palasiin ja perustaa niistä projekteja, jotta yletömän pitkiltä projekteilta välttyään. Esimerkkinä osaprojektit voivat olla esitutkimusprojekti, määrittelyprojekti tai toteutusprojekti. (Haikala & Mikkonen, 17-20.)

Ohjelmistotuotantoa varten on olemassa monia erilaisia lähestymistapoja. Näistä kaikkein yksinkertaisimmassa ns. ”code-and-hack” lähestymistavassa, ohjelmistoa kasvatetaan, laajennetaan, muutetaan ja korjataan kunnes asiakas on tyytyväinen (Haikala & Mikkonen, 17-20). Tutkimuksessa keskitytään tutkimaan hyväksymistestausta perinteisempään vesiputousmalliin pohjautuvan V-mallin, sekä iteratiivisen ketterän kehityksen Scrum-mallin kautta. Näiden kahden mallin valintaan vaikuttivat seuraavat seikat: Perättäisiä tehtäviä soljuvasti eteenpäin vievää vesiputousmallia pidetään ideaalisena kehitysmallina, koska vaatimukset on dokumentoitu jo alkumetreillä mikä antaa selkeän kuvan testauksen ja vaatimusten kytköksistä (Hambling & Samaroo, 21-22). Tähän vesiputousmalliin myös V-malli pohjautuu. Scrum on taas yleisimmin käyttöön otettu ketterän kehityksen menetelmä, joka on saanut kiitosta ohjelmistojen kehittäjiltä (Haikala & Mikkonen, 46-47).

Selvitän tässä työssä miten yllämainitut mallit vaikuttavat asiakkaan vastuulla olevaan hyväksymistestaukseen ja onko toimittajan valitsemalla kehitysmallilla vaikutuksia siihen.

Tarkastelu aloitetaan kertomalla yleisesti testauksesta ja testauksen eri tasoista. Sen jälkeen perehdytään V-Modell XT, sekä Scrum kehitysprosesseihin ja tarkastellaan hyväksymistestauksen sijoittumista kyseisiin prosesseihin. Viimeiseksi teoriaosuudessa pereh-

dytään itse tutkimuksen kohteeseen eli hyväksymistestaukseen, sen prosesseihin, tehtäviin ja tarvittaviin resursseihin.

2.1 Yleistä testauksesta ja testauksen tasot

Laadunvarmistus on ja tulee olemaan haaste tietojärjestelmiä ja ohjelmistoja kehitettäessä. Haikalan ja Mikkosen (2011, 205.) mukaan testauksen työvaiheisiin ja niihin liittyviin virheiden jäljitykseen ja korjaukseen (debugging) kuluu yli puolet projektin resursseista, joten testauksen vaiheet tulee suunnitella huolellisesti. Pitää kuitenkin muistaa, että testaus on vain yksi väline millä laatuun voidaan konkreettisesti vaikuttaa, käyttämällä testauksen tuloksia tietojärjestelmän ja ratkaisun parantamiseen. Tämä ei kuitenkaan laadukkaaseen lopputulokseen pääsemiseen yksin riitä, vaan laatu tulee rakentaa tietojärjestelmiin sisään. (Broekman, Koomen, van der Aalst & Vroon 2006, 40.)

ISO standardi määrittelee laadun hyvin vaikeasti kuvailtavaksi. Mitä laatu tarkoittaa toiselle, voi se toiselle tarkoittaa aivan päinvastaista. Laadunhallinnan kannalta on tärkeää, että tietojärjestelmää kehitettäessä yritetään minimoida epäsuorat vaatimukset ja muutetaan ne määritellyiksi vaatimuksiksi, jotta myöhemmin pystytään näyttämään kuinka hyvin lopputulos vastaa määriteltyjä vaatimuksia. Tärkeää on myös ettei testaus ole ainoa tapa laadun varmistamiseen, vaan laadunhallinnassa testaus on yksi hammasratas koko koneistossa. (Broekman ym. 2006, 40-41.)

Ohjelmistojen testauksessa testaus mielletään yleisesti järjestelmälliseksi virheiden etsimiseksi, joka tapahtuu käyttämällä ohjelmistoa tai sen tiettyä osaa. Valitettavan usein testaus tapahtuu kokeilemalla ohjelmaa satunnaisesti, vaikka testauksen lähtökohtana tulisi olla suunnitelmallinen virheiden etsiminen. Kun testaajana on ohjelmiston tekijä, tavoitteena on ennemmin ohjelmiston toimivuuden todistaminen kuin virheiden etsiminen. (Haikala & Mikkonen 2011, 205.)

Testausprosessissa on mukana monia eri tahoja kuten testaajat, ohjelmistojen rakentajat, projektin johto ja asiakkaat. Heillä kaikilla on omat tavoitteensa testausprosessissa. Naik ja Tripathy (2008, 10-11) ovat kuvailleet tavoitteet seuraavasti:

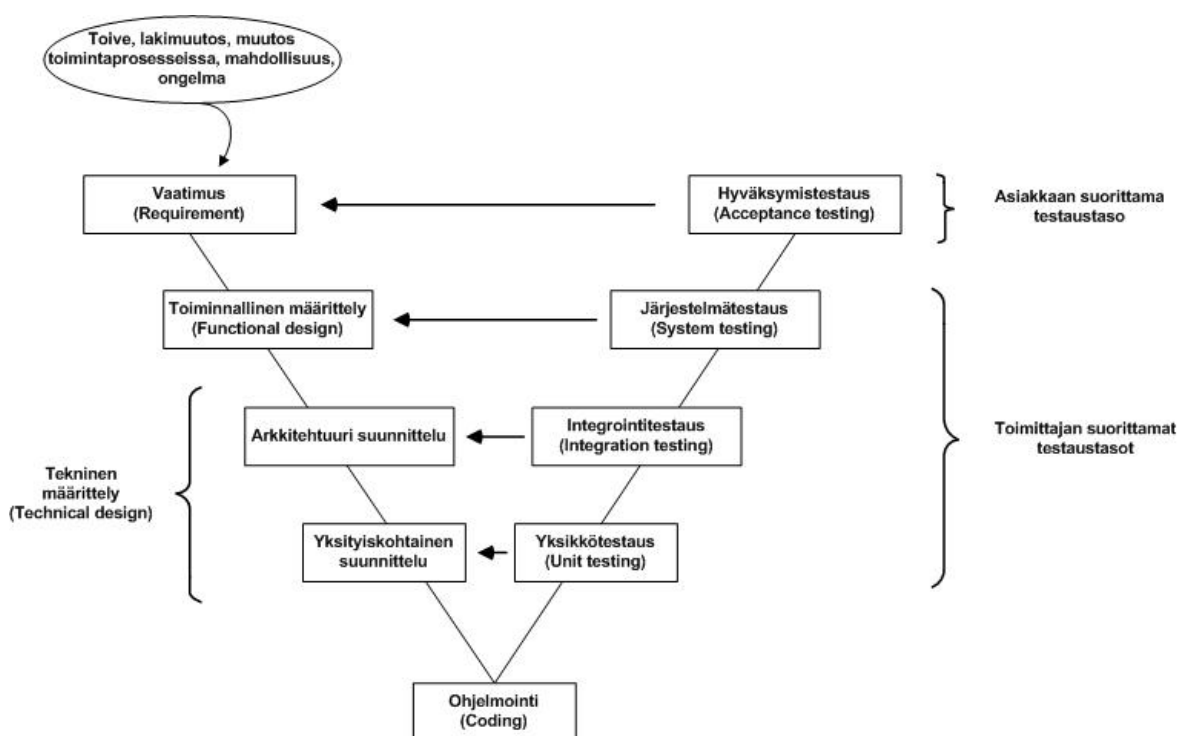
- **Se toimii:** Järjestelmän yksikön (unit) toteutusvaiheessa järjestelmän kehittäjä haluaa testata, että järjestelmän osa toimii normaaleissa olosuhteissa. Hän haluaa saada varmuuden siitä, että osa toimii tyydyttävällä tavalla. Samanlainen tarve tulee esille järjestelmän integrointi (integration) vaiheessa, jolloin halutaan kokeilla järjestelmän perustoimintojen toimivuus. Kehittäjän kannalta testauksen tavoite on enemmänkin osoittaa järjestelmän toimivuus, kuin sen toimimattomuus.
- **Se ei toimi:** Järjestelmän kehittäjien ollessa sovellukseen tai sen osaan tyytyväisiä, aloitetaan sovelluksen tai sen osan virheiden metsästy. Tavoitteena on saada sovel- lus tai sen osa rikkoontumaan.
- **Vähennetään riskiä häiriöistä:** Suurin osa monimutkaisista järjestelmistä sisältää vikoja, jotka saavat sovelluksen rikkoontumaan silloin tällöin. Tämä termi ”hajoaa silloin tällöin” (failing from time to time) johtaa käsitteeseen **häiriöaste** (failure ra- te). Kun testausvaiheiden aikana häiriöitä korjataan ja testejä jatketaan, ohjelmiston häiriöaste pienenee. Näin ollen ylemmän tason tavoitteena testauksen suorittamisel- le on häiriöasteen pienentäminen hyväksyttävälle tasolle.
- **Vähennetään testauksesta aiheutuvia kuluja:** Testausprosessin kulut muodos- tuvat seuraavista tehtäväkokonaisuuksista ja niihin liittyvistä kuluista:
 - testitapausten suunnittelu, ylläpito ja suorittamiseen liittyvät kulut,
 - testitapausten tulosten analysointiin liittyvät kulut,
 - testitapausten dokumentointiin liittyvät kulut,
 - sekä järjestelmän käyttäminen ja sen dokumentointi.

Yllämainittuja kuluja silmälläpitäen sitä vähemmän testaus tulee maksamaan, mitä vä- hemmän yksittäisiä testitapauksia suunnitellaan. Silti testitapausten minimointi ei ole hyvä tapa vähentää kuluja. Ylimmän tason testauksen tavoitteena on tuottaa pienen riskitason ohjelmisto jota testataan vähemmällä määrällä testitapauksia. Tällä haetaan **testitapausten tehokkuutta** eli toisin sanoen testaajat ymmärtävät valita vähemmän, mutta tehokkaampia testitapauksia. (Naik & Tripathy 2008, 10-11.)

Ohjelmistokehityksen vaiheet ja testaus ovat läheisesti kytköksissä toisiinsa. Kuviossa 1 on kuvattu näiden kahden suhdetta V-mallia hyväksikäyttäen. Kuviossa näkyy selkeästi testauksen neljä tasoa ja niiden kytkökset kuhunkin kehityksen vaiheeseen. V-mallissa kuvion vasemmalla puolella kuvataan kehittämisen vaiheet vaatimuksista kokonaiseksi

tuotokseksi. Kuvion oikea puoli kuvaa testauksen tasoa ja keskellä oleva viiva osoittaa sitä mitä kehittämisen osiota vasten testejä tulee tehdä. Näistä testitasoista kolmesta ensimmäisestä testitasosta (alhaalta lukien) vastaa ohjelmistotoimittaja ja viimeisestä tasosta järjestelmän vastaanottava asiakas. Yleisesti luullaan, että V-malli soveltuu vain vesiputousmalliseen kehitykseen, mutta tämä ei pidä paikkaansa. V-mallia voidaan soveltaa sekä iteratiivisissa että inkrementaalisissa ohjelmistokehitysmalleissa. (Broekman ym. 2006, 43-45.)

Kuvio 1 on rakennettu hyödyntäen neljää eri lähdettä, koska yksikään kirjallisuudesta löydetystä V-mallin kuvioista ei kuvannut V-mallia toivotusti. Esimerkiksi Haikalan & Mikkosen (207.) kuviosta puuttui kokonaan hyväksymistestauksen taso.



Kuvio 1. V-mallin tasot. (Broekman ym. 2006, 48; Naik & Tripathy 2008, 16; Hambling & Samaroo 2009, 22; Haikala & Mikkonen 2011, 207.)

Testaustasojä on Naik & Tripathyn (2008, 16-17) mukaan neljä erilaista. Tasot ovat yksikkötestaus (unit testing), integroititestausta (integration testing), järjestelmätestaus (system testing) sekä hyväksymistestausta (acceptance testing). Hambling ja Samaroo (2009, 22) ovat myös samoilla linjoilla, mutta heidän yksikkötestausta kutsutaan moduulitestaukseksi (module test). Haikalan & Mikkosen (2011, 208) hyväksymistestausta

käytäntö poikkeaa myös hieman jo edellä mainituista. He kuvaavat asiakkaan suorittamaa testausta alfa- ja beta-testauksella. Alfa-testauksen asiakas tekee toimittajan tiloissa ja beta-testauksella tarkoitetaan asiakkaan tekemiä testauksia asiakkaan omissa tiloissa.

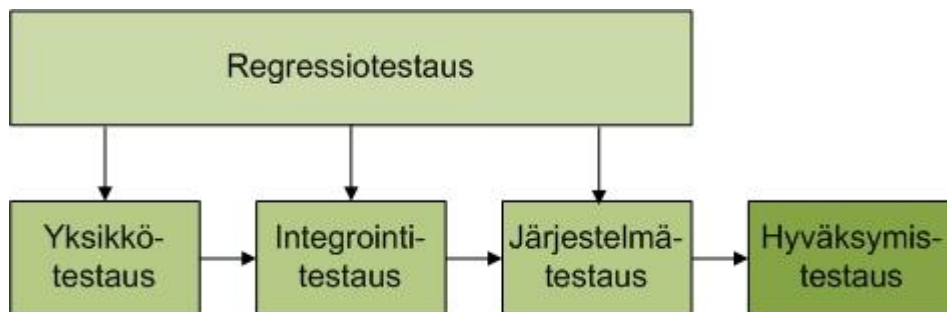
Yksikkötestauksessa testataan yksittäinen luokka tai moduuli ja testauksen suorittajana toimii ohjelmoija. Testijärjestelmänä käytetään kehitysjärjestelmää. Testin tavoitteena on varmistaa, että yksikkö tai moduuli täyttää sille teknisessä määrittelyssä kuvatut vaatimukset. (Naik & Tripathy 2008, 16; Haikala & Mikkonen 2011, 207.)

Integroititestauksessa yhdistetään yhteen luokkia tai moduuleja ja tämän testaustason painopiste on moduulien välisten rajapintojen toimivuuden tutkiminen. Testaus suoritetaan kehitysjärjestelmässä ja testaajina toimivat yhteistyössä ohjelmoija ja toimittajan integroititestausinsinöörit. Testin tavoitteena on varmistaa, että ohjelmisto toimii teknisessä määrittelyssä kuvatun mukaisesti. (Naik & Tripathy 2008, 16-17; Haikala & Mikkonen 2011, 207-208.)

Järjestelmätestauksessa tarkastelukohteena on koko järjestelmä, mukaan lukien laitteistot sekä ohjelmistot. Testauksen tuloksia verrataan määrittelyvaiheessa tehtyihin toiminnallisiin määrittelyihin ja testaajina pitää olla testaajia, jotka ovat mahdollisimman riippumattomia itse kehitystyöstä. Tämä sen vuoksi, että toteuttajat keskittyvät usein ohjelman osiin, jotka tiedetään toimiviksi. He eivät tämän vuoksi löydä virheitä yhtä hyvin kuin ulkopuolinen testaaja. Ulkopuoliset testaajat tulevat siis mukaan järjestelmätestausvaiheessa kun taas yksikkö- ja integroititestauksen tekevät ohjelmoijat. (Haikala & Mikkonen 2011, 207-211; Naik & Tripathy 2008, 192.)

Järjestelmätestauksen vaihe on testausvaiheista kaikkein kriittisin, koska testattavaa on paljon ja on kuitenkin pystyttävä pitämään projektille asetettu aikataulu. Jos ja kun virheitä huomataan ja niitä korjataan, voi se tarkoittaa muutoksen tekemistä myös muihin moduuleihin. Tämä saattaa tarkoittaa järjestelmän uudelleen testaamista. Järjestelmätestauksessa suoritetaan toiminnallisuustestien lisäksi myös erilaisia ei-toiminnallisia testejä esimerkiksi asennus-, toiminnallisuus-, kuormitus-, luotettavuus- ja stressitestit. (Naik & Tripathy 2008, 16-17 & 192-194.)

Regressiotestaus on myös tärkeä testauksen taso, mutta sitä ei luokitella normaaliksi testauksen tasoksi. Regressiotestausta suoritetaan yksikkö-, integrointi-, ja järjestelmätestausvaiheissa. Kuviossa 2 on kuvattu regressiotestauksen suhdetta muihin testauksen tasoihin. Regressiotestaus suoritetaan käyttämällä testikirjastossa valmiina olevia testitapauksia ja se toteutetaan aina kun järjestelmän komponentti muutetaan. Sen tarkoitus on varmistaa, ettei muutos aiheuta uusia vikoja sellaisilla alueilla, jotka eivät suoraan ole muutoksen kohteena. Tämä testaustaso on kallis, koska sitä saatetaan toistaa useita kertoja kehitysvaiheen aikana. Sen vuoksi on tarkkaan harkittava mitkä testitapaukset valitaan ja kuinka kattava regressiotestaus tullaan suorittamaan. (Naik & Tripathy 2008, 17; Broekman ym. 2006, 501-502.)



Kuvio 2. Regressiotestaus verrattuna testauksen tasoihin. (Naik & Tripathy 2008, 17.)

Kun järjestelmätestaus on onnistuneesti tehty, ohjelmisto toimitetaan asiakkaalle, jolloin aloitetaan hyväksymistestaus. Hyväksymistestauksen tekee asiakas ja sen tarkoituksena on pääasiassa ohjelmiston laadun varmistaminen ei niinkään systemaattinen virheiden etsiminen, mikä on järjestelmätestauksen tarkoitus. Hyväksymistestauksen käynnistyessä asiakkaalla tulee olla hyväksymiskriteerit, jotka perustuvat asiakkaan omiin odotuksiin järjestelmästä. Hyväksymistestauksesta on olemassa myös toinenkin kategoria nimeltään liiketoiminnan hyväksymistestaus (Business Acceptance Testing), jonka tekee toimittaja. BAT:in tarkoituksena on varmistaa, että järjestelmä aikanaan tulee läpäisemään asiakkaan tekemän hyväksymistestauksen. (Naik & Tripathy 2008, 450.)

Broekman ym. (2006, 81-83) sen sijaan jakavat testaustasot ryhmiin, jotka ovat kehitystestaus (development tests), järjestelmätestaus ja hyväksymistestaus. Kehitystestaus pitää sisällään yksikkötestauksen ja integrointitestauksen. Järjestelmätestaus on samankal-

tainen kuin aiemmin esitetty. Suurin ero aiemmin esittelystä on se, että hyväksymistestaus tasoja on neljä.

- Järjestelmäintegroititestaus (System integration test), jonka tarkoituksena on varmistaa, että järjestelmäintegraatioita koskevat vaatimukset on täytetty.
- Toiminnallinen hyväksymistestaus (Functional acceptance test), jonka tarkoituksena on varmistaa, että järjestelmä täyttää toiminnalliset vaatimukset.
- Käyttäjän hyväksymistestaus (User acceptance testing), jonka tarkoituksena on varmistaa, että järjestelmä täyttää käyttäjien asettamat vaatimukset.
- Tuotannollinen hyväksymistestaus (Production acceptance testing), jonka tarkoituksena on varmistaa, että järjestelmä täyttää järjestelmän hallinnoinnista vastaavien vaatimukset.

Yllämainituista kolme ensimmäistä testaustasoa suorittaa järjestelmän tulevat käyttäjät ja neljännen tason testaa järjestelmän ylläpitäjät. Testit tehdään kehitysympäristössä, joka on mahdollisimman hyvin tuotantojärjestelmää simuloiva ympäristö. (Broekman ym. 2006, 82.)

Julkisen hallinnon tietohallinnon neuvottelukunnan JHS 182 suosituksen (2011, 9) mukaan tärkein testauksen taso on hyväksymistestaus, joka on ICT-palvelun tilaajan vastuulla. Tämä kyseinen tapahtuma on pakollinen, jolla hyväksytään toimittajan toteuttama tuotos. Kyseisen dokumentaation liitteessä on kuvattu laadunvarmistuksen tehtäviä, jotka pitävät sisällään palvelun laadunvarmistuksen. Testaus tulee suunnitella niin, että sillä todennetaan ohjelmiston toiminnallisten ja ei-toiminnallisten vaatimusten oikeanlainen toiminta. Testauksella tulee siis todeta toteutuksen laatu ja havainnot ja mitatut tulokset ja poikkeamat pitää dokumentoida. (Julkisen hallinnon tietohallinnon neuvottelukunta 2011, liite 1.)

2.2 Hyväksymistestauksen sijoittuminen V-Modell® XT mallissa

V-Modell® XT on ohjelmistokehityksen viitekehys, jolla ohjataan sekä projektin suunnittelu, että toteutusvaihetta. V-Modell® XT on viimeisin versio V-Modell sarjasta, joka on toiminut vuodesta 1997 lähtien ohjelmistokehitysprojektien viitekehyyksenä Saksan valtion siviili- ja armeijan virastoille. Tämä viimeisin versio kehitettiin mallista V-Modell

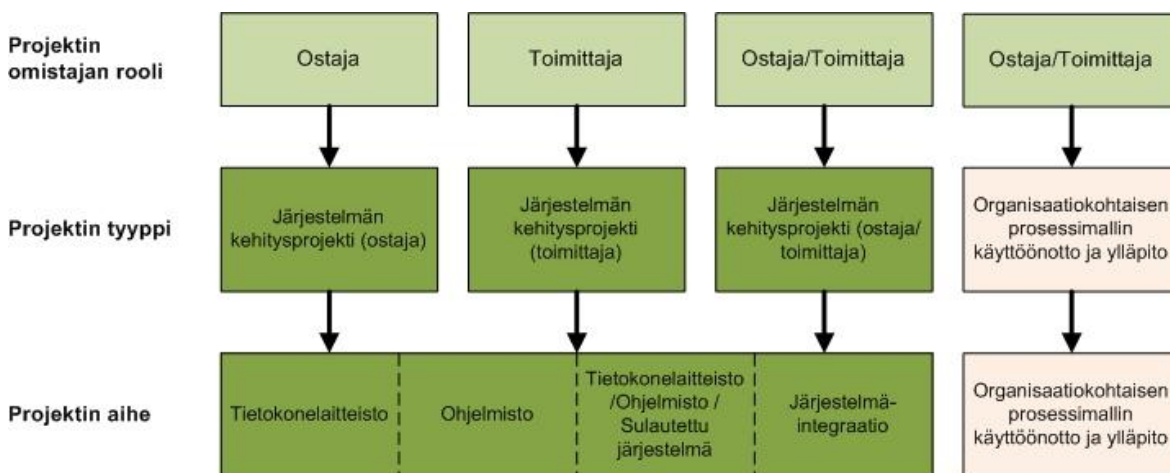
97 vastaamaan 2000 luvun IT järjestelmäkehityksien tarpeita. (IABG.; IABG 2006a, 5-6.)

V-Modell® XT viitekehyksellä on neljä päätavoitetta. Tavoitteet ovat projektiin liittyvien riskien minimoiminen, projektin tulosten laadun varmistaminen ja kehittäminen, projektien ja järjestelmänelinkaaren kustannusten alentaminen sekä kommunikaation parantaminen kaikkien sidosryhmien välillä. (IABG 2006a, 6.)

Kuviossa 3 on kuvattu mallin sisältä löytyvät kolme erillistä ohjelmiston kehitysprojektityyppiä. Kaikki nämä tyypit käyvät tietokonelaitteisto-, ohjelmisto-, sulautettujen järjestelmien, sekä integraatioprojektien toteuttamiseen. Projektityypit määräytyvät sen mukaan millainen projektirooli projektin omistajalla on. Projektityypit ovat

- **järjestelmän kehitysprojekti (ostaja),**
- **järjestelmän kehitysprojekti (toimittaja),**
- **sekä järjestelmän kehitysprojekti (ostaja/toimittaja).**

Näiden lisäksi viitekehys pitää sisällään projektityypin **organisaatiokohtaisen prosessimallin käyttöönotto ja ylläpito**, joka on tarkoitettu projekteille joiden tarkoituksena on vakiinnuttaa tietty prosessimalli organisaation käyttöön. (IABG 2006a, 11-12.)

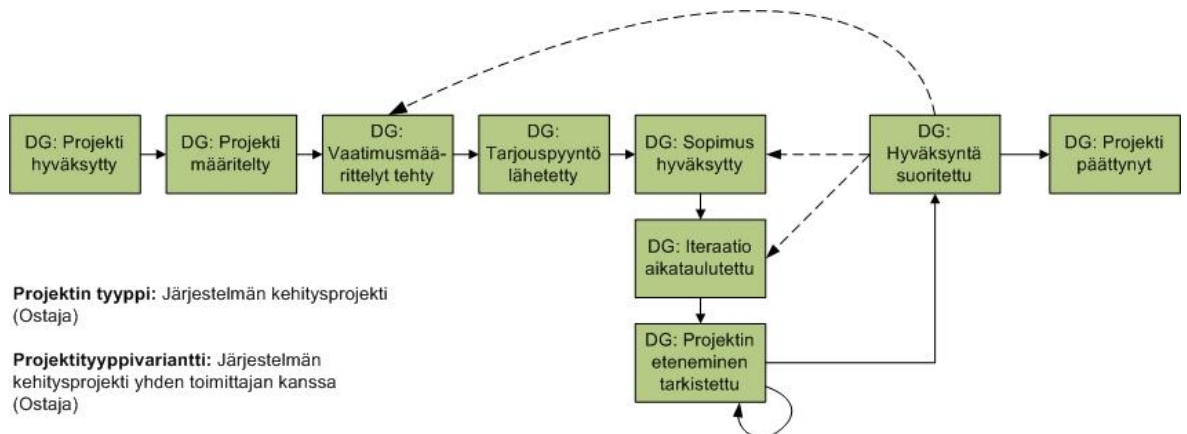


Kuvio 3. V-Modell® XT projektin roolit, tyypit ja mahdolliset aiheet (IABG 2006a, 11-12.)

Projektityypin ohella viitekehys tarjoaa jokaiselle tyypille ainakin yhden projektityypin variantin, joilla projektin luonne ja mahdolliset prosessipolut tarkentuvat. Näiden ohella

olennainen osa viitekehystä ovat prosessimoduulit (esimerkiksi vaatimusmäärittelyt), jotka kuvaavat mitä moduulissa tulee tehdä. (IABG 2006a, 13-14, IABG 2006c, 3.)

Näitä hyväksyntäportteja hyväksikäyttäen projektin eteneminen voidaan kuvata. Kuviossa 4 on kuvattu järjestelmän kehitysprojektin (ostaja) prosessikuvaus projektissa, jossa on mukana vain yksi toimittaja. Tämä projekti ei pidä itsessään sisällään kehitystä, koska tässä projektityypissä asiakas ostaa kehityksen toimittajalta.



Kuvio 4. Järjestelmän kehitysprojektin prosessikuvaus yhden toimittajan kanssa (IABG 2006b, 30-31.)

Projektin ensimmäisenä vaiheena on saada hyväksyntä projektin aloittamiselle ja näin läpäistä hyväksyntäportti **DG: Projekti hyväksytty**. Tässä vaiheessa projektin sponso- rin nimeämä henkilö kerää päätöksenteon kannalta olennaiset tiedot ja tekee niistä esi- tyksen projektin muodossa, joita hyväksikäyttäen päätös projektin aloittamisesta tai sen hylkäämisestä voidaan tehdä. (IABG 2006b, 31.)

Hyväksynnän jälkeen valmistellaan projektin- ja laadunvarmistuksen käsikirjat, jotka määrittävät projektin rungon. Näitä dokumentteja arvioidaan **DG: Projektin määritte- ly** hyväksyntäportissa, jolloin niiden täytyy olla sopivat projektin toteuttamiseen ostajan näkökulmasta. (IABG 2006b, 31 & 135.)

Projektin määrittelyvaiheen hyväksynnän jälkeen, aloitetaan vaatimusmäärittelyjen do- kumentointi. Vaatimusmäärittelyjen valmistuttua ne asetetaan tutkittavaksi joko projek- tin ohjausryhmälle tai ohjausryhmän nimeämälle käyttäjälle. He varmistavat vaatimus-

määrittelyn täydellisyyden sekä vaatimusten priorisoinnin oikeellisuuden. Positiivisen tutkimustuloksen jälkeen vaatimusmäärittelyt kootaan vaatimusmäärittelydokumentiksi, jonka perusteella toimittaja voi rakentaa halutun järjestelmän. Vaatimusmäärittelydokumentin lisäksi hyväksymisportti **DG: Vaatimusmäärittelyt tehty** varten valmistellaan tarjouspyyntökonsepti (RFP concept). (IABG 2006b, 32.)

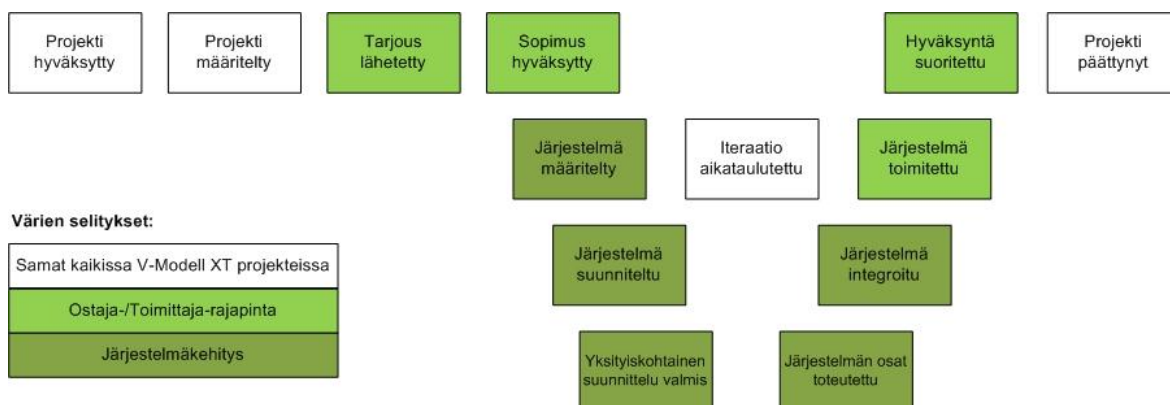
Yllämainitulla tarjouspyyntökonseptillä varmistetaan, että tarjouspyyntöprosessissa otetaan huomioon lait ja säännökset. Tämä on tärkeää julkista sektoria koskevissa tarjouspyynnöissä ja joissakin tapauksissa myös yksityisen sektorin tarjouspyynnöissä. (IABG 2006d, 78).

Vaatimusmäärittelyjen valmistuttua projektipäällikkö tekee tarjouspyyntödokumentit tarjouspyyntökonseptin ja vaatimusmäärittelydokumenttien pohjalta. Tähän vaiheeseen kuuluu myös tarjousten arviointikriteerien määrittely, jota käytetään seuraavassa vaiheessa tarjousten arviointiin ja vertailuun. Tähtäimessä on saavuttaa hyväksymisportti **DG: Tarjouspyyntö lähetetty**, jonka läpäisemisen jälkeen tarjoukset lähetetään mahdollisille toimittajille. (IABG 2006b, 32.)

Tarjouksien saapumisen jälkeen ne arvioidaan kriteeriluettelon avulla. Tarjouksien ja toimittajien kanssa käytyjen neuvottelujen perusteella ostaja valitsee toimittajan, jonka kanssa sopimus tehdään. Sopimus velvoittaa toimittajaa toimittamaan projektin sopimuksen mukaisesti sen tilaajalle eli ostajalle. Nämä toimenpiteet johtavat hyväksyntäporttiin **DG: Sopimus hyväksytty**, jonka positiivisen arvioinnin jälkeen voidaan aloittaa järjestelmäkehityksen prosessi. (IABG 2006b, 32-33.)

Projektin seuraava vaihe on iteraation suunnittelu, jossa järjestelmäkehityksen aikataulu, hyväksyntäportit tuotosten hyväksynnälle ja järjestelmänkehityksen laajuus suunnitellaan yhdessä toimittajan kanssa. Tässä vaiheessa tarkastellaan myös projektin- ja laadunvarmistuksen käsikirjat, jotta ne vielä tässä vaiheessa vastaavat edelleen projektin tarpeita. Nämä toimenpiteet johtavat **DG: Iteraatio aikataulutettu** hyväksymisporttiin. Tässä vaiheessa myös toimittajan järjestelmän kehitystyö alkaa. Kuviossa 5 on kuvattu kehitysprojekti toimittajan näkökulmasta ja siinä erottuu selvästi järjestelmäkehi-

tyksen hyväksymisportit ja järjestelmäkehityksen V-mallin mukainen kehitys. (IABG 2006b, 33 & 45.)



Kuvio 5. Kehitysprojekti toimittajan näkökulmasta sisältäen hyväksymisportit (IABG 2006a, 30.)

Kappaleessa 2.1 ja kuviossa 1 on kerrottu tarkemmin V-mallin vasemman ja oikean puolen linkityksistä toisiinsa.

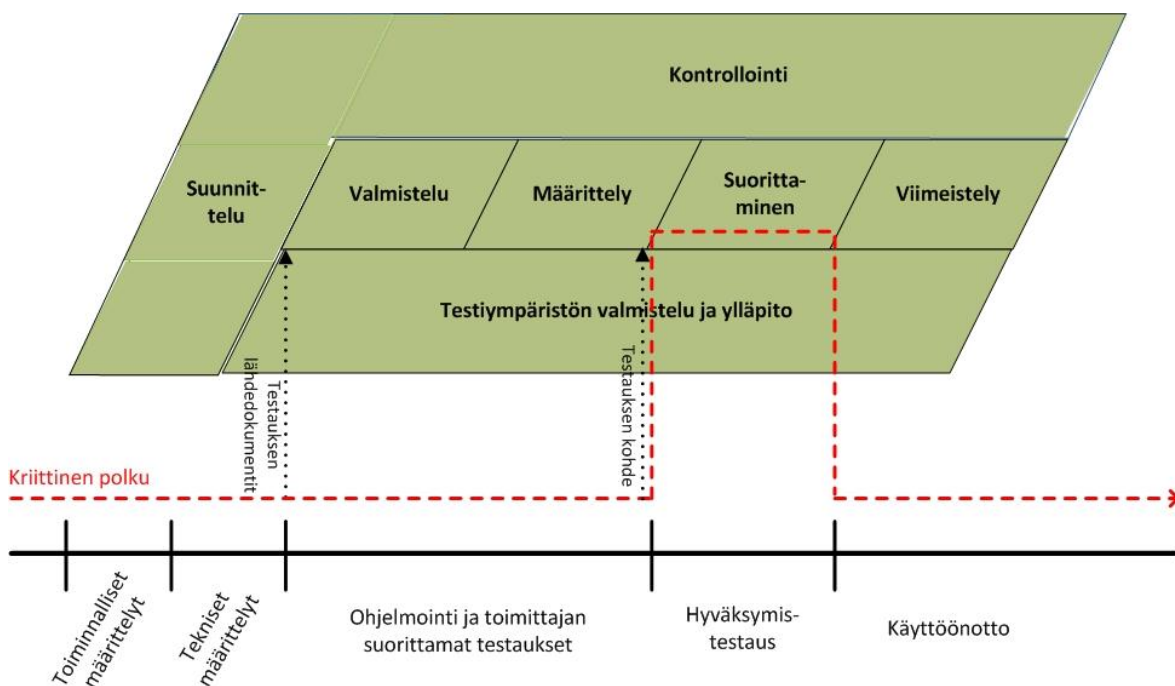
Huomioitavaa on, että V-Modell XT viitekehys ei rajoita toimittajaa tiettyyn kehitysmalliin. Valittavana on inkrementaalinen (Incremental), komponenttipohjainen (component-based) tai prototyyppi (Prototypic) kehitysmallit. Oli kehitysmalli mikä tahansa, on hyväksymistestaus suoritettava osana projektia. (IABG 2006b, 43, 49 & 61.)

Kehitystyön aikana toimittaja raportoi sovitun mukaisesti ostajalle, jossa toimittaja kertoo saavutetuista tuloksista sovittuihin merkkipaaluihin (milestone) nähden. Tämän prosessin ajan ostajan projektipäällikkö raportoi edistymisestä ohjausryhmään omalla raportillaan. Hänen on myös tuettava toimittajaa heidän projektissaan. Yllämainituilla toimenpiteillä pyritään läpäisemään hyväksymisportti **DG: Projektin eteneminen tarkistettu**. Tämä portti saatetaan ohittaa useita kertoja, kunnes on kehityksessä saavuttanut sovitun merkkipaalun ja voidaan siirtyä hyväksymisvaiheeseen. (IABG 2006b, 33 & 49.)

Toimittajan toimitettua sopimuksessa sovitut tuotokset ostajalle aloitetaan hyväksymistestaus, jonka ostaja suorittaa. Toimitettua tuotosta peilataan vaatimusmäärittelyssä dokumentoituihin hyväksymiskriteereihin (Acceptance Criteria) (IABG 2006d, 95.). Jos

toimitettu tuotos vastaa vaatimusmäärittelyä saavutetaan **DG: Hyväksyntä suoritettu** hyväksymisportti. (IABG 2006b, 34.)

V-Modell XT ei kuvaa tarkasti sitä, missä vaiheessa hyväksymistestauksen valmistelevat toimenpiteet tulee aloittaa ja mitä siihen kuuluu. Soveltamalla TMAP testauksen elinkaarimallia voidaan hyväksymistestauksen prosessin ja tehtävien liittyminen projektiin havainnollistaa. Kuviossa 6 käy ilmi, että testien valmisteluvaihe voidaan aloittaa kun testauksen kohteen lähdedokumentit ovat valmiina. Ohjelmoinnin ja toimittajan suorittamien testien jälkeen testauksen kohde toimitetaan hyväksymistestaukseen. Kuten kuvioista käy ilmi, on hyväksymistestauksen suorittaminen järjestelmäkehitysprojektin kannalta kriittisellä polulla. Sen vuoksi on tärkeää, että valmistelut ja määrittelyt on tehty mahdollisimman huolellisesti. (Broekman ym. 2006, 153.)



Kuvio 6. Hyväksymistestauksen prosessien liittyminen järjestelmäkehitysprojektiin (Broekman ym. 2006, 153.)

Hyväksymisvaiheesta voidaan siirtyä takaisin iteroinnin suunnitteluvaiheeseen esimerkiksi tapauksissa, joissa tuotokset toimitetaan osissa ja sopimukseen kuuluu monta iterointivaihetta. Toimittaja ja ostaja ovat voineet myös sopia, että yhden iteraation hyväksymisen jälkeen palataan neuvottelupöytään ja tehdään uusi sopimus. Tämä johtaa hyväksymistestauksen läpäisyn jälkeen siihen, että siirrytään takaisin sopimuksen tekemi-

sen vaiheeseen. Tapauksissa joissa on huomattu, että vaatimuksia joudutaan muokkaamaan ja muutokset eivät mahdu projektin laajuuteen, joudutaan palaamaan vaatimusmäärittely vaiheeseen. (IABG 2006b, 34.)

Kun kaikki vaatimukset on täytetty ja muutoslistalla olevat muutokset on suljettu, voidaan päättää projektin sulkemisesta. Projektille luodaan projektin lopetusraportti. Loppuraportti pitää sisällään saavutettujen tulosten lisäksi projektin opetukset (lessons learned), joita voidaan hyödyntää tulevaisuuden projekteissa. Tällä dokumentilla haetaan ohjausryhmästä hyväksyntä projektin sulkemiselle ja saavutetaan hyväksymisportti **DG: Projekti päättynyt.** (IABG 2006b, 74; IABG 2006d, 55.)

2.3 Hyväksymistestauksen sijoittuminen Scrum-mallissa

Scrum on ketterän kehityksen projektihallinnan viitekehys, joka on yleistynyt viime vuosina jopa niin, että termistä Scrum on tullut sanan ”ketterä” synonyymi (Haikala & Mikkonen 2011, 46).

Scrum on lähtöisin Japanista ja sillä kuvataan Ikujiro Nonakan ja Hirotaka Takeuchin kehittämää tuotekehitysprosessia. Nimi itsessään juontaa juurensa Rugby pelistä, jossa Scrum tarkoittaa rykelmäaloitusta jossa pallo pannaan takaisin peliin. Kumpaakin näistä yhdistetään mukautuvuuteen, nopeuteen, itsestään ohjautuvuuteen sekä kummassakin on vähän taukoja. Kyseisen pelin ja hyper-produktiivisen tuotantoprosessin yhtäläisyyksien takia nimi tuli jäädäkseen. (Beedle & Schwaber 2002, 1.)

Scrumissa on vain kolme roolia, jotka ovat tuoteomistaja (Product Owner), Scrum-mestari ja Scrum-tiimi (Scrum Team). Perinteisillä nimillä tuoteomistajan tehtävät ovat lähellä tuotepäällikön tehtäviä, Scrum-mestari roolin tehtävät ovat lähellä projektipäällikön tehtäviä ja Scrum-tiimi tarkoittaa projektiryhmää. (Beedle & Schwaber 2002, 31-37; Haikala & Mikkonen 2011, 48-49; Lindström 2011.)

Tuoteomistajan vastuulla on hallita ja ylläpitää tuotteen työlistaa asiakkaan tarpeiden ja liiketoiminnan vision pohjalta. Lindström (2011.) painottaa roolia kriittiseksi, jotta projektilla käytettävissä oleva aika ja resurssit käytetään oikeiden asioiden tekemiseen. Tuo-

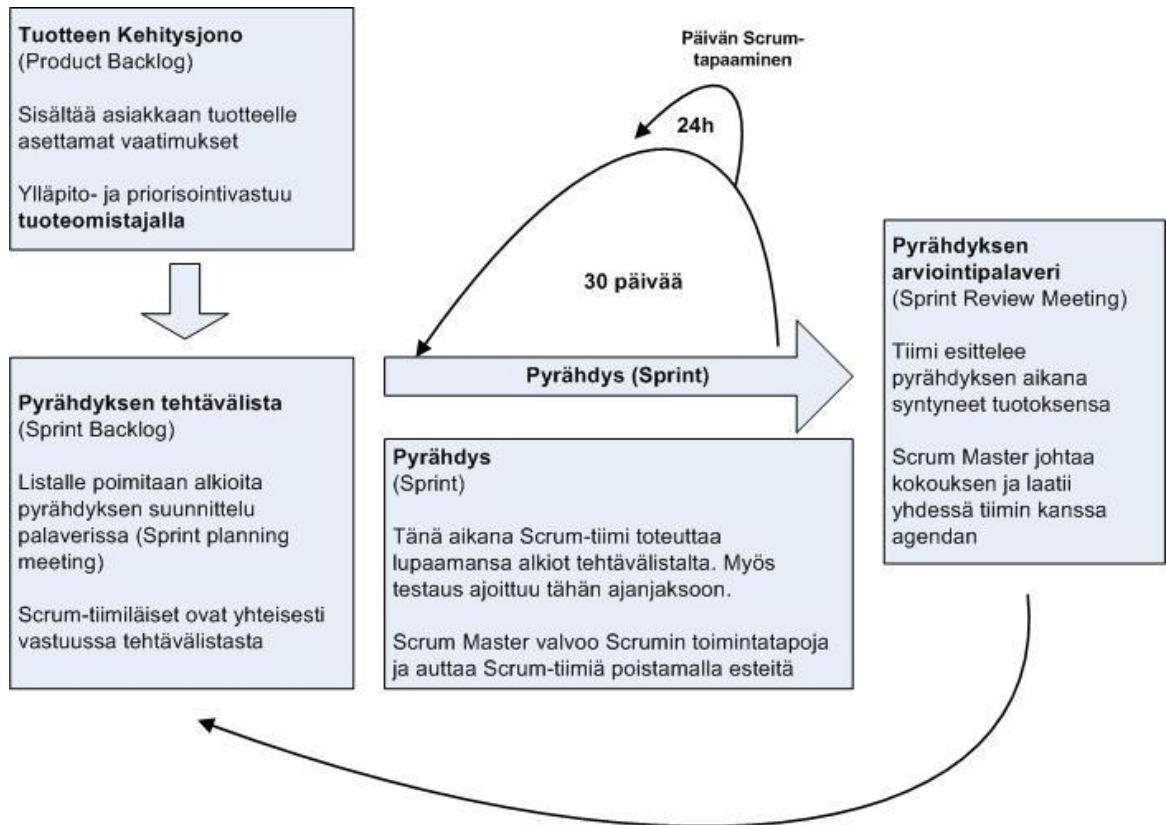
teomistaja ei voi olla komitea, vaan se on yksi henkilö. Tällä vältetään mahdolliset päällekkäiset tehtävälistat, jolloin Scrum-tiimi ei enää tiedä mikä lista on voimassaoleva. Hän toimii käyttäjien ja asiakkaan edustajana, sekä hänen vastuullaan on ohjata tuotteen suunta ja antaa prioriteetteja tuotteen työlistalla oleville alkioille. (Beedle & Schwaber 2002, 34.)

Scrum-mestari on henkilö, joka vastaa projektin sujuvasta etenemisestä. Hän auttaa asiakasta löytämään tuotteen omistajan, sekä auttaa muodostamaan oikeanlaisen Scrum-tiimin. Hänen vastuullaan on varmistaa, että projektin aikana noudatetaan Scrumin arvoja, käytäntöjä ja sääntöjä. Hänen on myös varmistettava, että tiimin jäsenillä on oikeat työkalut työn tekemiseen ja auttaa heitä ongelmien ratkomisessa. Oikea-aikaisia päätöksiä tekemällä ja esteitä poistamalla, Scrum-mestari pystyy ottamaan kaiken irti Scrum-tiimistä. (Beedle & Schwaber 2002, 31-32; Haikala & Mikkonen 2011, 49.)

Itseohjautuvan Scrum-tiimiin vastuulla on toimittaa pyrähdykselle (Sprint) asetetut tavoitteet ja heillä on valtuudet tehdä kaikki vaadittava, jotta asetettuihin tavoitteisiin päästään. Tiimin optimaalinen koko on 7 henkilöä ja siihen voi kuulua esimerkiksi ohjelmistosuunnittelijoita, testaajia ja ohjelmoijia. Schwaber ja Beedle eivät tarkalleen määrittele mitä rooleja Scrum-tiimissä on pakko olla, mutta käytännössä tiimi tulee rakentaa sellaiseksi, että pyrähdykselle asetetut tavoitteet voidaan saavuttaa. Tiimin vastuulla on aina analyysin tekeminen, suunnittelun laadinta, koodaus, testaus ja dokumentaation tekeminen. Tiimin on myös testattava tuotoksensa pyrähdyksen aikana ja joissakin projekteissa tiimiin on otettu oma laadunvarmistuksesta vastaavia testaajia. (Beedle & Schwaber 2002, 35-37.)

Scrum-prosessin eteneminen on kuvattu kuviossa 7 ja seuraavissa kappaleissa kerrotaan tarkemmin kuviossa kuvatun prosessin eteneminen. Scrum-prosessin aluksi asiakkaan ohjelmistolle asettamat vaatimukset kerätään tuotteen kehitysjonoon (Product Backlog). Tälle listalle kuvataan tuotteesta sekä toiminnalliset, että teknologiset vaatimukset. Listalla olevat vaatimukset voivat käytännössä olla mitä tahansa ja ne voivat olla lähtöisin keneltä tahansa esimerkiksi käyttäjiltä, myynnin edustajilta, arkkitehdeiltä ja niin edelleen. Vaikka lähteitä on monia, vain yksi henkilö, tuoteomistaja, saa priorisoida ke-

hitysjonoa. Yhteistyössä muiden Scrum-projektissa mukana olevien kanssa, tuoteomistaja arvioi kehitysjonossa olevien vaatimusten valmistamiseen tarvittavan ajan. Tämä aika pitää sisällään kaiken alkion tekemiseen tarvittavan ajan, sisältäen esimerkiksi suunnittelun, ohjelmoinnin ja testauksen. (Beedle & Schwaber 2002, 7 & 35; Lindström 2011.)



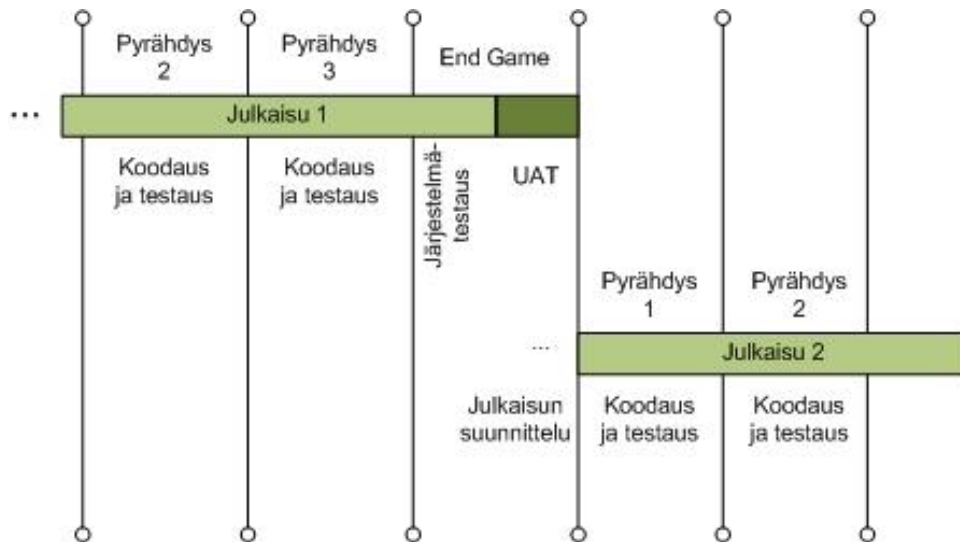
Kuvio 7. Scrum-prosessikuvaus (Beedle & Schwaber 2002, 8; Lindström 2011; Haikala & Mikkonen 2011, 46-51.)

Tuotteen kehitysjonon valmistuttua Scrum-tiimi valitsee tuotteen kehitysjonosta ne vaatimukset, jotka he haluavat rakentaa julkaisun aikana. Valituista vaatimuksista syntyy julkaisun kehitysjono (Release Backlog). Scrum-tiimi priorisoi kehitysjonon vaatimukset ja arvioi niiden toteuttamiseen tarvittavan ajan. Suuremmat vaatimukset pilkotaan pienemmiksi, helpommin hallittaviksi tehtäviksi. Työmäärien ja prioriteettien ollessa tiedossa, tiimin on helppoa jakaa julkaisun kehitysjonolla olevat vaatimukset pyrähdyksiin, joita yhdessä julkaisussa voi olla useita. (Shoajee 2012.)

Julkaisun kehitysjonon valmistuttua Scrum-tiimi kokoontuvat pyrähdysen suunnittelupalaveriin kokoamaan pyrähdysen työlistan (Sprint Backlog). Tiimi pitää kokouksen myös tuoteomistajan kanssa, jossa mietitään mitkä ominaisuudet valitaan seuraavan pyrähdysen työlistalle. Pyrähdysen työlistan valmistuttua aloitetaan itse pyrähdys (Sprint), mikä kestää maksimissaan 30 päivää. Pyrähdysen aikana toteutetaan kaikki toimenpiteet vaatimuksille, jotka on valittu pyrähdysen työlistalle. Tiimiläiset kokoontuvat päivittäin 15 minuutin mittaiseen tilanne katsaukseen, jota kutsutaan päivän Scrum-tapaamiseksi (Daily Scrum meetings). Raportointi intervalli on 24 tuntia ja tapaamisessa tiimi esittää Scrum-mestarille edellisen kokouksen jälkeen aikaan saadut asiat ja kertoo mitä tullaan tekemään ennen seuraavaa tapaamista. He myös tiedottavat Scrum-mestarille esteistä, joita he ovat kohdanneet. (Beedle & Schwaber 2002, 7, 40 & 47.)

Pyrähdysen lopuksi pidetään tapaaminen, jossa Scrum-tiimi esittelee pyrähdysen aikana tehdyt tuotoksensa yleisölle. Tätä tapaamista kutsutaan pyrähdysen arviointipalaveriksi (Sprint Review Meeting) ja tähän tapaamiseen osallistuu Scrum-mestari ja tuoteomistaja, mutta myös yrityksen johto ja käyttäjät ovat tervetulleita. Tuotoksien arvioinnin jälkeen tuoteomistaja saattaa tehdä muutoksia tuotteen kehitysjonoon, jonka jälkeen Scrum-tiimi valitsee seuraavan pyrähdysen työlistalle tulevat vaatimukset. (Beedle & Schwaber 2002, 10 & 54.)

Kun julkaisun kaikki pyrähdykset on valmiina ja järjestelmä on vakaa, Crispin ja Gregory (2009, 456-466.) suosittaa ns. viimeistelyvaiheen (The End Game) aloittamista. Kuviossa 8 on kuvattu diagrammina julkaisun aikajana, josta nähdään viimeistelyvaiheen sijoittuminen projektin loppupäähän. Viimeistelyvaiheen aikana suoritetaan järjestelmätestaus ja asiakkaan suorittama hyväksymistestaus. Hyväksymistestauksessa tulisi olla edustettuina kaikki liiketoiminta-alueet, joihin kyseinen julkaisu vaikuttaa.



Kuvio 8. Julkaisun aikajana sisältäen hyväksymistestauksen (Crispin & Gregory 2009, 466.)

Kappaleesta 2.2 löytyvässä kuviossa (kuvio 6) on kerrottu järjestelmäkehityshankkeen ja hyväksymistestauksen prosessien linkittymisestä. Tätä kyseistä lähestymistapaa voidaan soveltaa myös silloin, kun käytetään Scrum mallia. Tämä sen vuoksi, että oli järjestelmäkehityksen elinkaarimalli sitten mikä tahansa, on järjestelmäkehityksen tehtävät kaikissa samat. Esimerkiksi V-mallissa toiminnalliset ja tekniset määrittelyt syntyvät jo ennen koodauksen aloittamista, mutta iteratiivisissa malleissa kuten Scrum nämä dokumentit valmistuvat vasta pyrähdysten lopussa. (Broekman ym. 2006, 153-154.)

Kyseisten dokumentaatioiden myöhäinen valmistuminen asettaa tiettyjä haasteita testauksen lähdedokumentaatiota kerätessä. Tämä ei kuitenkaan saa olla este hyväksymistestauksen prosessien etenemiselle, vaan testauspäällikön ja testaajien on valittava vaihtoehtoinen lähestymistapa, kun järjestelmäkehitys toteutetaan iteratiivisesti. Tällaisessa tapauksessa testauksen lähdedokumentiksi hyväksytään 80% valmiina oleva dokumentti ja luodaan sen pohjalta testitapaukset. Kun testaus voidaan aloittaa ja dokumentaatio on viimeistelty, testaaja korjaa testitapausta. Yleensä tässä vaiheessa korjaukset tulevat olemaan pieniä. (Broekman ym. 2006, 275-278.)

Ketterän kehityksen (Agile) malleja käytettäessä on toimittavan organisaation myös lisättävä yhteistyötä ja kommunikaatiota asiakkaan kanssa. Kaikkien projektissa mukana olevien tahojen on ymmärrettävä ketterän kehityksen pelisäännöt. Ketteräkehitys vaatii asiakkaalta erilaista lähestymistä projektiin, koska vaatimukset ja vaatimusmäärittelyt

saattavat muuttua kehitysprojektin aikana ja kehityksen tuotokset toimitetaan asiakkaalle katselmoitavaksi ja testattavaksi osissa. Tämä asettaa myös testaukselle uusia vaatimuksia verrattuna perinteisiin järjestelmäkehitysmalleihin. (Kasurinen, Kettunen, Smolander & Taipale 2010, 237.)

Asiakasorganisaation on arvioitava omia testaustapoja ja käytäntöjä ketterän kehityksen vaatimuksia vasten. Kun kehitystuotokset toimitetaan asiakkaalle osissa, tarkoittaa se asiakkaan mukanaoloa koko projektin kehitysvaiheen ajan. Tällä on positiivinen vaikutus hyväksymistestaukseen, koska on helpompaa ja nopeampaa testata järjestelmä osissa, ennemmin kuin koko järjestelmä yhdellä kertaa. Myös jatkuva yhteistyö ja kommunikatio toimittajan ja asiakkaan välillä, sekä nopea mielipiteiden ja kommenttien vaihto, auttaa järjestelmänkehittäjää ymmärtämään asiakkaan vaatimuksia selkeämmin. (Kasurinen ym. 2010, 237-238.)

3 Hyväksymistestaus

Hyväksymistestaus on testaustasoista viimeinen yksikkötestauksen, integrointitestauksen ja järjestelmätestauksen jälkeen. Tarkempaa tietoa testauksen tasoista löytyy kappaleesta 2.1. Kehitysoikeuden saavutettua vaiheen, jossa toimittava taho on suorittanut järjestelmätestauksen onnistuneesti, siirtyy testausvastuu asiakkaalle. Asiakas tekee hyväksymistestauksen joka perustuu aiemmin määriteltyihin hyväksymiskriteereihin. Hyväksymiskriteereillä ja niiden läpäisyllä halutaan varmistaa, että järjestelmä täyttää sille asetetut liiketoimintavaatimukset. Yleensä asiakkaalla on oikeus hylätä järjestelmä, jos se ei läpäise hyväksymistestaukselle asetettuja testitapauksia, jotka pohjautuvat hyväksymiskriteereihin. (Naik & Tripathy 2008, 450; Broekman ym. 2006, 151.)

Asiakas voi myös ulkoistaa hyväksymistestauksen suunnittelun ja toteutuksen kolmannelle osapuolelle. Moni alan konsultointiyrityksistä tarjoaa tällaista palvelua. Suomessa tällaista palvelua tarjoaa esimerkiksi Cybercom (Cybercom Finland, 2011). Vaikka asiakas ulkoistaisi kyseisen prosessin, on asiakkaan laadittava tai oltava mukana laatimassa hyväksymiskriteerit järjestelmälle. Hyväksymiskriteerit on sovittavat asiakkaan ja toimittajan välillä jo sopimusvaiheessa, jotta vältetään riitatilanteita. (Naik & Tripathy 2008, 450-451.)

Hyväksymistestauksen prosessi on hyvin samankaltainen kuin järjestelmätestauksen prosessi, mutta niistä vastaavat eri toimijat. Järjestelmätestauksen tekee toimittaja ja hyväksymistestauksen asiakas. Järjestelmäkehitysoikeudessa on hyvä huomioida tämä seikka ja näille kahdelle testaustasolle tulee harkita itsenäisiä prosesseja. Toisin sanoen molemmille tasoille voidaan laatia omat suunnitelmat, budjetit ja rakentaa erilliset testijärjestelmät testien tekemistä varten. Hyväksymistestauksen prosessin valmistelevat työvaiheet ajoittuvat samanaikaisesti kehityksen kanssa, joten prosessi kannattaa käynnistää samanaikaisesti, kun toiminnallisia määrittelyjä tehdään. (Broekman ym. 2006, 151.)

Kun testauksen kohteena oleva järjestelmämuutos on laaja on hyvin todennäköistä, että järjestelmä ei läpäise kaikkia hyväksymiskriteereitä yhdellä kertaa. Sen vuoksi testauk-

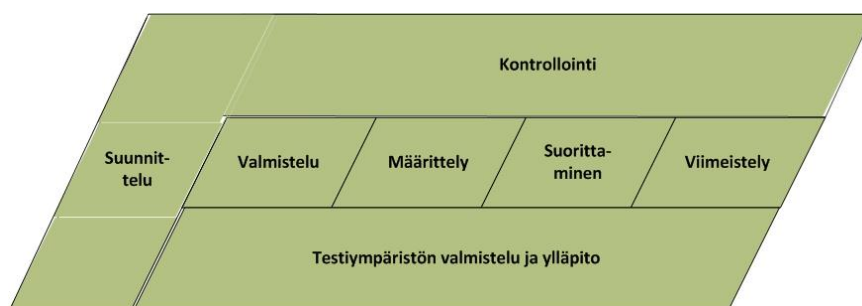
sessä on käytännön syistä hyvä keskittyä kolmeen hyväksymistestauksen päätavoitteeseen, jotka ovat:

- varmistaa, että järjestelmä vastaa sille asetettuja vaatimuksia ja täyttää hyväksymiskriteerit,
- keskityä tunnistamaan ja korjaamaan järjestelmävirheitä,
- arvioi järjestelmän valmius tuotantoon siirtoon. (Naik & Tripahty 2008, 451.)

3.1 Hyväksymistestausprosessi ja siihen kuuluvat tehtävät

Hyväksymistestauksen suorittamiseen Broekmanin ym. (2006, 151) mukaan voidaan käyttää TMap elinkaarimallia. Se sisältää prosessin aktiviteetit, niiden sidonnaisuudet ja suoritusjärjestyksen. Prosessi on jaettu 7 vaiheeseen jotka ovat suunnittelu (Planning), kontrollointi (Control), testiympäristön valmistelu ja ylläpito (Setting up and maintaining infrastructure), valmistelu (Preparation), määrittely (Specification), suorittaminen (Execution) ja viimeistely (Completion).

Kuviossa 9 on kuvattu TMap elinkaarimalli ja siinä on silmäänpestävää prosessin suunnikas muoto. Tähän muotoon on päädytty sen vuoksi, että prosessin aktiviteetteja ei välttämättä suoriteta peräkkäin. Esimerkkinä tapaus jolloin toisen muutoksen testaus on jo aloitettu ja toisen testauksen määrittelyt on vielä kesken. On myös suositeltavaa, että testauksen viimeistelyvaiheen tehtävien suunnittelu aloitetaan jo määrittelyvaiheessa. (Broekman ym. 2006, 152.)



Kuvio 9. TMap elinkaarimalli (Broekman ym. 2006, 152.)

Suunnitteluvaiheen tarkoituksena on luoda laadukas testaus suunnitelma, jolla toimeksianto voidaan onnistuneesti suorittaa. Suunnitteluvaiheessa rakennetaan testaus suunnitelma, joka pitää sisällään testauksen lähestymistavan, aikataulun, budjetin, suoritettavat

aktiviteetit ja listan tuotoksista, jotka tullaan tuottamaan testauksen aikana. Suunnitteluvaiheen tuotokset kuvataan yleensä erillisellä testaus suunnitelmalla, jos testaus suoritetaan itsenäisenä aktiviteettinä. Järjestelmäkehityksen lähestymistavan ollessa iteratiivinen tai jos käytetään ketterän kehityksen malleja, saattaa hyväksymistestaus olla integroitu kokonaisprosessiin ja näin mukana projektisuunnitelmassa. (Broekman ym. 2006, 155.)

Ennen suunnitteluvaiheen käynnistymistä on tiettyjen asioiden oltava testauspäällikön tietoisuudessa. Jos esimerkiksi järjestelmäkehitykselle valittu lähestymistapa ei ole tiedossa, voi tällä olla negatiivinen vaikutus suunnitteluvaiheen pituuteen. Jotta suunnittelun vaihe voidaan aloittaa ja suorittaa järkevästi ja aikataulussa, tulisi seuraavien asioiden olla selvillä:

- hyväksymistestausvaiheen asiakas,
- järjestelmän tai testattavan paketin tavoite ja tärkeys asiakkaalle,
- yleiset vaatimukset,
- kehitys-, ylläpito- ja käyttöönottoprosessien organisaatiot,
- kehitystyön toimitusaikataulu,
- järjestelmäkehityksen lähestymistapa (esim. ketterä kehittäminen),
- jos on olemassa ns. päätestisuunnitelma (Master Test Plan), tulee se korjata ja hyväksyttää,
- ymmärrys käytettävistä testaus- ja tuotantojärjestelmistä, jotta käytettävä testijärjestelmä saadaan määriteltä. (Broekman ym. 2006, 155-156.)

Hyväksymistestauksen suunnitteluvaiheen tärkein resurssi on testauspäällikkö tai testauskoordinaattori, joka vastaa testisuunnitelmasta. Jos projektille ei ole määriteltä päätestisuunnitelmaa, täytyy testauspäällikön ottaa kantaa suunnitteluvaiheessa seuraaviin näkökulmiin: tulokset (Results), riskit (Risks), aika (Time) ja kulut (Costs). Tämän testausvaiheen tehtävät, jotka Broekman ym. ovat määritelleet, on listattu liitteessä 1 (liite 1.). (Broekman ym. 2006, 156.)

Hyväksymistestauksen **kontrollointivaiheella** halutaan taata asiakkaalle ja sidosryhmille ymmärrys testien edistymisestä, testattavan järjestelmän tai objektin laadusta ja ris-

keistä, sekä testiprosessin laadusta. Näitä tietoja raportoimalla annetaan asiakkaalle mahdollisuus vaikuttaa testauksen lopputulokseen. Tämä prosessivaihe aloitetaan kun suunnitteluvaihe on onnistuneesti saatettu päätökseen. Tätä vaihetta johtaa testauspäälikkö ja vaiheen tehtävät ovat johtaminen, monitorointi, raportointi ja mukauttaminen. (Broekman ym. 2006, 224-225.)

Tehtävällä ”johtaminen” tarkoitetaan testausprosessin, havaittujen vikojen, sekä testattavan järjestelmän johtamista siten, että muilla projektin sidosryhmillä ja asiakkaalla on ajantasaista tietoa niiden edistymisestä ja laadusta. Monitoroinnissa testauspäälikkö valvoo testaussuunnitelman toteutumista. Tässä tehtävässä testauspäälikön aika menee suurelta osin testiryhmän valvomiseen ja ohjaamiseen. Tähän tehtävään kuuluu myös tärkeänä osana kommunikointi testiryhmän ulkoisten sidosryhmien kanssa. Tehtävän ollessa hyvin haastava on sanomattakin selvää, että testauspääliköltä edellytetään hyviä sosiaalisia- ja kommunikaatiotaitoja. (Broekman ym. 2006, 224-226.)

Raportointi pitää sisällään sekä testattavan objektin tilan ja laadun, että testausprosessin etenemisen ja laadun raportoinnin. Raportoinnilla halutaan varmistaa, että asiakas ja muut sidosryhmät pystyvät ohjaamaan testauksen kulkua tarvittaessa. Raportoinnin syklistä ja raportoinnin laajuudesta päätetään suunnitteluvaiheessa ja ne on kirjattu testaussuunnitelmaan. Raportilla raportoidut korjaavat tai parantavat toimenpiteet, joille saadaan asiakkaan suostumus, toteutetaan ”mukauttaminen” tehtävässä. Mukauttamistehtävään kuuluu myös testausprosessin tuotosten muokkaaminen sallittujen rajojen sisällä. (Broekman ym. 2006, 237 & 249.)

Testiympäristön valmistelu ja ylläpito vaiheen tarkoituksena on toimittaa vaaditut testiympäristöt testausaktiviteettejä varten. Testiympäristö pitää sisällään testauksen suorittamiseen tarvittavat testijärjestelmät, testausta tukevat testityökalut, sekä työasemat ja tilat testaajille. Vaiheeseen kuuluu yhteensä kuusi tehtävää, jotka on listattu liitteessä 2. Suositeltavaa on, että näitä tehtäviä ei suorita testauspäälikkö, vaan tehtävään nimetään eri henkilö, jota kutsutaan testiympäristön koordinaattoriksi. Ennen tämän vaiheen aloittamista tulisi testaussuunnitelmasta käydä ilmi yleisellä tasolla testiympäristön vaatimukset. Jos testaus työkaluja käytetään, pitäisi olla tiedossa miten muut testausprosessin aktiviteetit tullaan suorittamaan. (Broekman ym. 2006, 253-255.)

Valmisteluvaiheessa on tarkoitus varmistaa, että testauksen lähdedokumentit (test basis) ovat laadullisesti riittävällä tasolla, jotta testitapausten suunnittelu voidaan toteuttaa. Testauksen lähdedokumenteilla tarkoitetaan kaikkea sitä tietoa millä kuvataan järjestelmän haluttua toiminnallisuutta. Tämän vaiheen tehtäviä on yhteensä neljä ja ne ovat testauksen lähdedokumenttien kerääminen, tarkastuslistan laatiminen, testauksen lähdedokumenttien arviointi ja testattavuuden arvioinnin raportointi. Tarkemmat kuvaukset testausvaiheen tehtävistä löytyy taulukosta 1. (Broekman ym. 2006, 271-274.)

Testauspäällikkö on avainroolissa valmisteluvaiheessa, mutta kaikkia vaiheita hänen ei tarvitse suorittaa itse. Hän laatii testattavuuden arviointiraportin, mutta muut tehtävät voi suorittaa kuka tahansa testaustiimistä. Valmisteluvaihe voidaan käynnistää, kun hyväksymistestauksen testisuunnitelma on saatu valmiiksi ja kun testauksen lähdedokumentaatio on valmis. (Broekman ym. 2006, 271-274.)

Taulukko 1. Valmisteluvaiheen tehtävät ja niiden tarkoitus (Broekman ym. 2006, 275-283.)

Tehtävä	Tehtävän tarkoitus
Testauksen lähdedokumenttien kerääminen	Kerätään täsmälliset testauksen lähdedokumentit yhdessä liiketoiminnan / asiakkaan kanssa, sisältäen esim. Toiminnalliset ja tekniset määrittelyt, haastattelu raportit, vaatimusmäärittelyt, käyttöjäohjeet ja yms..
Tarkastuslistan laatiminen	Laaditaan tarkastuslista perustuen testistrategiaan, joka on taltioitu testausuunnitelmaan. Tätä tarkastuslistaa käytetään arvioimaan testauskohteen testattavuutta.
Testauksen lähdedokumenttien arviointi	Arvioidaan testauksen lähdedokumentit. Arvioidaan miten hyvin vaatimukset on käännettävissä testitapauksiksi, kuinka täydellisiä ja johdonmukaisia dokumentit ovat ja kuinka hyvä saatavuus testauksen lähdedokumenteilla on.
Testattavuuden arvioinnin raportointi	Tällä raportilla tähdätään raportoimaan testauksen lähdedokumenttien laadusta ja sen vaikutuksista testeihin, informoidaan järjestelmäsuunnittelun heikoista kohdista varhaisessa vaiheessa ja informoidaan mahdollisista projektiin kohdistuvista riskeistä.

Määrittelyvaiheen tarkoitus on määritellä testausjaksolle tarvittavat testit ja niiden aloitusajankohdat. Tarkoituksena on valmistella niin paljon kuin mahdollista etukäteen,

jotta testaukset voidaan suorittaa mahdollisimman nopeasti siitä kun toimittaja toimittaa testauksen kohteen. Tämä vaihe voidaan aloittaa kun testattavuuden arviointi on testausperustalle suoritettu ja arvioinneissa mahdollisesti löydetty epäkohdat on korjattu. Myös testauksen lähdedokumentaation on oltava saatavilla. (Broekman ym. 2006, 285-286.)

Tämän vaiheen sisältävät aktiviteetit suorittavat testaajat ja vaiheen aktiviteetit ovat testauksen määrittely, keskeisten lähtöpisteiden määrittely ja testiobjektin vastaanoton määrittely. Tehtävien tarkemmat kuvaukset on kerrottu taulukossa 2. Määrittelyvaiheessa testaajat määrittelevät tarvittavat testit jokaiselle testiyksikölle (test unit). Käytännössä tämä tarkoittaa testitapausten tai tarkastuslistan laatimista valitulla testien suunnittelumenetelmällä. Testiyksikkö pitää sisällään kokoelman prosesseista, tapahtumista ja toiminnoista, jotka testataan yhdessä. (Broekman ym. 2006, 285-286.)

Taulukko 2. Määrittelyvaiheen tehtävät ja niiden tarkoitus. (Broekman ym. 2006, 287-306.)

Tehtävä	Tehtävän tarkoitus
Testauksen määrittely	Testaaja määrittelee jokaiselle testiyksikölle vaadittavat testit. Testien tyypistä ja tekniikoista riippuen, tämä aktiviteetti saattaa sisältää mitä tahansa lähtien suunnittelun tarkistuslistoista testitapausten kirjoittamiseen.
Keskeisten lähtöpisteiden määrittely	Määritellään yksi tai useampi lähtöpiste testien aloittamiselle, jolloin käyttäjä voi hankkia dataa testien suorittamista varten. Tämä pitää sisällään sekä datan, että järjestelmän tilan missä sen pitää olla testien aloituksen hetkellä.
Testiobjektin vastaanoton määrittely	Määritellään testiobjektin vastaanotto tekemällä testiobjektille tarkistuslista, jolla voidaan todeta testiobjektin täydellisyys. Tämän lisäksi laaditaan esitestiskripti, jolla varmistetaan, että testiobjekti on laadultaan tarpeeksi hyvä testien aloittamiselle.

Testien **suorittamisvaiheessa** tarkoituksena on luoda näkemys testiobjektin laadusta. Tämä toteutetaan ajamalla määrittelyvaiheessa määritellyt ja sovitut testit. Kaikissa aiemmissa vaiheissa tehdyt alustavat toimenpiteet takaavat sen, että testausajat pystytään pitämään mahdollisimman lyhyinä. Testaaminen voidaan aloittaa kun testiobjekti on toimitettu, testiskriptit ovat valmiina kyseiselle testiobjektille ja testaukseen vaadittu testiympäristö on paikallaan. (Broekman ym. 2006, 307-308.)

Suorittamisvaiheen tehtävät ovat testiobjektin vastaanotto, lähtöpisteiden valmistelut, testien suorittaminen ja testitulosten tarkistus ja arviointi. Tarkemmat kuvauksen tehtävistä on listattu taulukossa 3. Lähes kaikki tehtävät suorittamisvaiheessa voi tehdä kuka tahansa testaustiimistä, ainoastaan testiobjektin vastaanotto on testauspäällikön vastuulla. Hän saa tarvittaessa apua testiympäristö koordinaattorilta. (Broekman ym. 2006, 309-320.)

Taulukko 3. Suorittamisvaiheen tehtävät ja niiden tarkoitus. (Broekman ym. 2006, 309-320.)

Tehtävä	Tehtävän tarkoitus
Testiobjektin vastaanotto	Määrittelyvaiheessa tehtyjen tarkistuslistojen ja esitesti skriptin avulla, varmistetaan toimitettujen testiobjektien toimivuus. Tällä varmistetaan, että testit voidaan viedä läpi asianmukaisesti.
Lähtöpisteiden valmistelut	Ennen testitapausten suorittamista testiobjekti tuodaan sellaiseen tilaan, että testaus voidaan suorittaa. Käytännössä suoritetaan ne toimenpiteet, jotka määrittelyvaiheessa tunnistettiin.
Testien suorittaminen	Suoritetaan testit ja laaditaan testien tulokset, jotta testiobjektin arviointi voidaan suorittaa.
Testitulosten tarkistus ja arviointi	Testituloksia verrataan ennustettuihin tuloksiin ja sen pohjalta analysoidaan niiden eroavaisuudet.

Viimeistelyvaiheen tehtävänä on ottaa oppia kokemuksista, jotka kerättiin testijakson aikana ja taltioida testiympäristö tulevaisuutta varten. Viimeistelyvaihe voidaan aloittaa suorittamisvaiheen loppupuolella. Viimeistelyvaiheen tehtäviä ovat testiprosessin arviointi ja testiympäristön taltiointi. (Broekman ym. 2006, 322.)

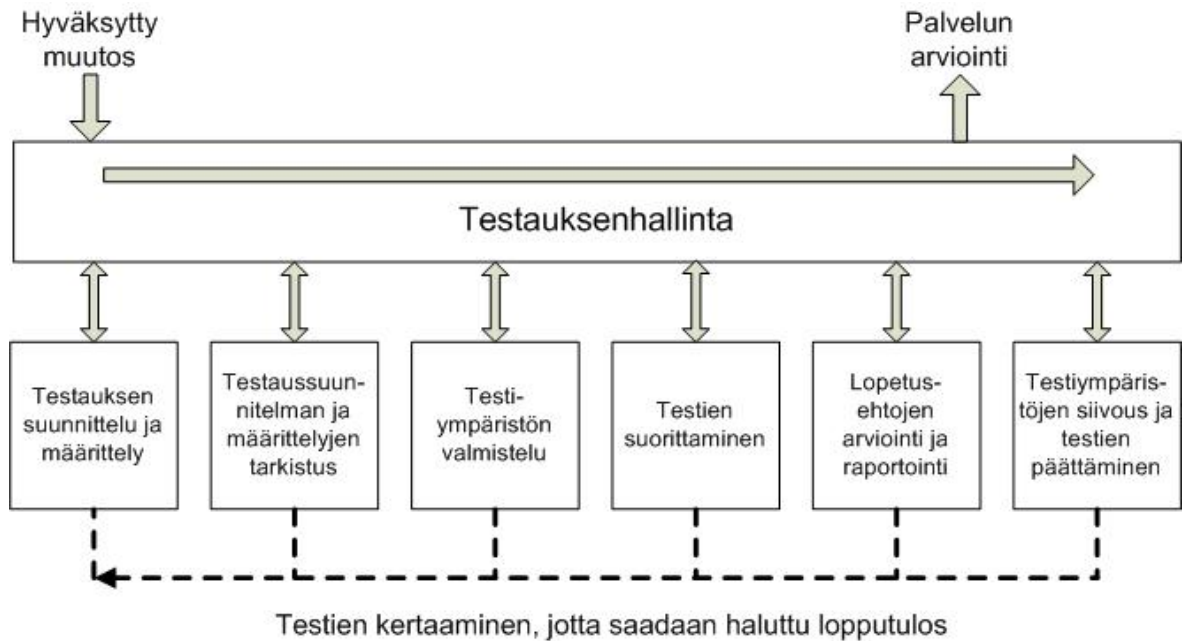
Testiprosessin arviointiin voi osallistua kaikki tiimin jäsenet ja tämän tarkoituksena on kerätä kaikki oleellinen testien aikana kertynyt tieto ja oppi, josta on mahdollisesti hyötyä tulevaisuudessa. Tieto on paras kerätä arviointipalaverissa ja nämä sessiot toimivat erinomaisena palautteena testauspäällikön toteuttamaan testiraporttiin, joka laaditaan kontrollointivaiheessa. Testiympäristön taltioinnilla on läheinen linkki testiympäristön valmistelu- ja ylläpitotehtävään. Tarkoituksena on valita ja päivittää testiympäristö siten,

että sitä voidaan optimaalisesti hyödyntää tulevaisuuden testeissä. (Broekman ym. 2006, 322-326.)

ITIL (The Information Technology Infrastructure Library), joka on globaalisti tunnustettu prosessikehys IT-palveluiden hallintaan ja johtamiseen, sisältää myös kuvauksen testauksen prosesseista. ITIL:issä testaus on osa palvelun siirtoa (Service Transition). ITIL:in kuvaamaan testausprosessiin kuuluu 7 vaihetta, sama määrä kuin TMAP elinkaarimallissa. Prosessin vaiheet ovat:

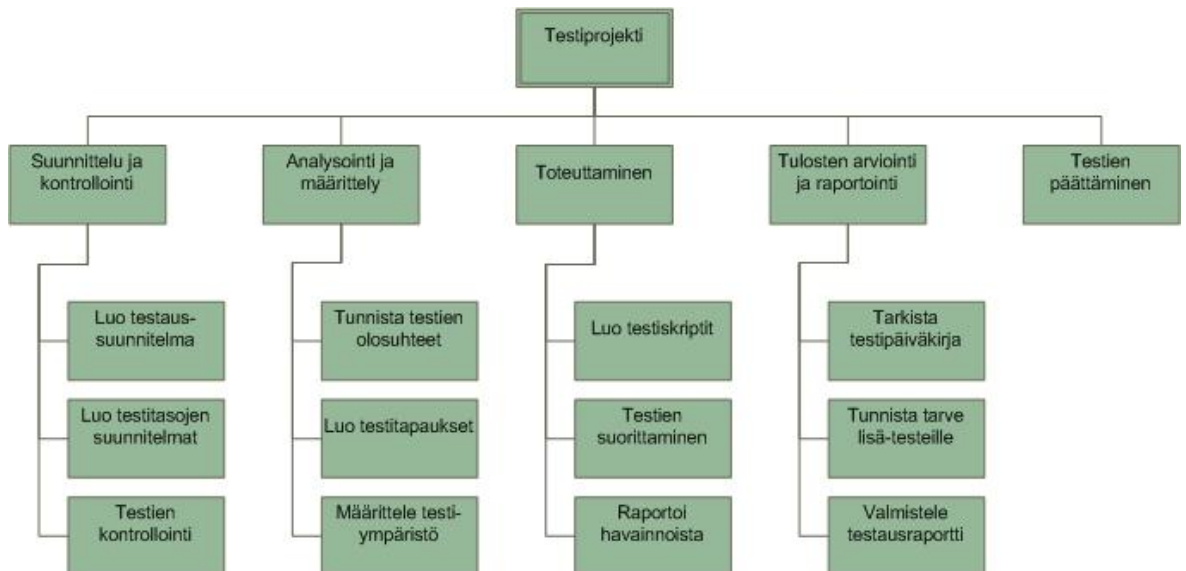
- kelpuuttaminen ja testauksenhallinta (Validation and test management),
- testauksen suunnittelu ja määrittely (Plan and design tests),
- testaussuunnitelman ja määrittelyjen tarkistus (Verify test plan and test designs),
- testiympäristön valmistelu (Prepare test environment),
- testien suorittaminen (Perform tests),
- lopetusehtojen arviointi ja raportointi (Evaluate exit criteria and report),
- sekä testiympäristöjen siivous ja testien päättäminen. (Office of Government Commerce 2007, 133-134.)

ITIL:issä kuvattu testausprosessi (kuvio 10.) on samankaltainen kuin edellä mainittu TMAP prosessi, mutta ITIL-testausprosessissa ei puhuta pelkästään hyväksymistestauksesta vaan palvelujen hyväksymistestauksesta. Palvelun hyväksymistestaus alkaa palvelun vaatimusten todentamisella. Esimerkiksi liiketoiminnan henkilöt, jotka ovat muutosten takana, hyväksyvät palvelun hyväksymiskriteerit ja hyväksymistestaussuunnitelman. Testauksen suorittamisen jälkeen suoritetaan arviointi ja raportointi, jonka jälkeen järjestelmät siivotaan. (Office of Government Commerce 2007, 122-134.)



Kuvio 10. ITIL:in testausprosessi (Office of Government Commerce 2007, 133.)

Kuviossa 11 on esitetty Hamblingin ja Samaroon (2009, 32-33 & 100-101.) testausprosessi, jonka he ovat jakaneet viiteen eri aktiviteettiin. Testaus aloitetaan suunnittelulla, jossa pystytetään valvonta ja kontrollointimekanismit. Tämä varmistaa kokonaisprosessin toimivuuden ja tehokkuuden. Tärkeimpänä aktiviteettinä pidetään testauksen tavoitteiden ja hyväksymiskriteereiden asettamista, jotka määräävät testien lopettamisen asetettavat rajat. Seuraavaksi tunnistetaan testattava tilanne ja testitapaukset, jonka jälkeen suunnitellaan testiskriptit ja suoritetaan ne. Testien suorittamisen jälkeen tulokset arvioidaan testien läpäisy (pass/fail), jonka jälkeen tuloksia vielä verrataan hyväksymiskriteereihin. Kun hyväksymiskriteerit täyttyvät siirrytään testauksen lopetusaktiviteetteihin.



Kuvio 11. Testausprosessi joka sisältää yltäason aktiviteetit (Hambling & Samaroo 2009, 101.)

3.2 Keskeiset roolit, tehtävät ja vaatimukset

Testaaja on toinen keskeisistä hyväksymistestaukseen kuuluvista rooleista. Kun verrataan Broekman ym. (2006, 461.) ja The British Computer Societyn (2012b) testaajien toimenkuvia, on BCS:n testauksen urapolun taso 3 ja Broekman ym. testaajan tehtävät ja vaadittavat pätevyudet hyvin lähellä toisiaan. Testaajalla tulisi olla mainittujen ominaisuuksien lisäksi olla suoritettuna ISTQB-BCS Certified Tester Foundation tason sertifikaatti (BCS 2012a).

Testaus on osana riskien vähentämistä, jossa testaaja on itsessään riski. Jos testaaja ei testaa oikein ja asiankuuluvalla tarkkuudella, ei riskejä voida eliminoida ja seuraukset voivat olla katastrofaaliset. Sen vuoksi testaajan täytyy osata testata ja hänen täytyy tuntea testauksen kohteena oleva järjestelmä hyvin. Testaajan tulee tuntea myös liiketoimintaprosessit, joihin kyseinen muutos vaikuttaa, sekä teknologia minkä päälle kyseinen järjestelmä on rakennettu. (Broekman ym. 2006, 453-454.)

Testaaja kommunikoi testien valmistelujen ja suorittamisen aikana useiden eri tahojen kanssa kuten projektipäällikkö, kehittäjät ja toiset testaajat. Sen vuoksi testaajan yksi tärkeimmistä luonteenpiirteistä on omata hyvät suulliset ja kirjalliset kommunikaatio-aidot. Tietyissä tilanteissa hänen täytyy olla myös hienotunteinen, jotta hän osaa käsi-

tellä kireätkin tilanteet oikein. Toisin sanoen hänen tulee olla hyvä tiimityöskentelyssä. Testaajan täytyy olla tarkka ja analyyttinen, ja hänen on osattava olla tekemättä oletuksia. Hänen täytyy pystyä olemaan vakuuttava ja hellittämätön, mutta myös puolueeton ja positiivisesti kriittinen. Testaaja tarvitsee luovuutta, jotta hän osaa rakentaa mahdollisimman hyviä testitapauksia. (Broekman ym. 2006, 455-456; BCS 2012b.)

Testaaja suorittaa ensisijaisesti testaukseen liittyviä tehtäviä ja nämä tehtävät ovat luonteeltaan luovuutta vaativia, mutta myös hyvin aikaa vieviä. Testaajan suorittamia tehtäviä hyväksymistestauksen suorittamista edeltävänä aikana ovat esimerkiksi testauksen lähtedokumenttien testattavuuden arviointi sekä loogisten ja fyysisten testitapausten, aloitustilanteiden ja testiskriptien määrittely. Hyväksymistestauksen aikana testaajan tehtäviin kuuluu aloitustilanteiden valmistelu, testien suorittaminen (dynaaminen testaus), testitulosten tarkistus ja arviointi, tarkistusten ja tutkimusten suorittaminen (staattinen testaus) sekä havaittujen vikojen kirjaaminen. Testaajan vastuulla on myös testauksen materiaalien taltiointi. (Broekman ym. 2006, 461; BCS 2012b.)

Testaajalta vaaditaan paljon erilaista osaamista, jotta hän osaa toimia hyvin testauksen eri osa-alueilla. Testaajan täytyy ymmärtää testauksen prosessien lisäksi liiketoiminnan prosessit ja yrityksen linjaorganisaation rakenne ja toiminta. Alhaalla on listattu tärkeimpiä pätevyys- ja taitoja, jota hyvältä testaajalta edellytetään:

- taitoa osata arvioida testattavuutta ja hallita testien suunnittelutekniikat,
- taitoa käyttää hyödyksi tarkistuslistoja,
- testausautomaation tietämystä ja tuntemusta,
- tietämystä yleisistä järjestelmäkehityksen malleista,
- taitoa tulkita testauksen lähtedokumentaatioita (vaatimukset, toiminnalliset ja teknilliset määrittelyt)
- kykyä omaksua uusia testaustyökaluja nopeasti,
- tietämystä projektityöstä ja projektiorganisaatioista,
- taitoa osata kirjoittaa dokumentteja ja osata käyttää taulukkolaskentaohjelmia,
- luovuutta ja tarkkuutta. (Broekman ym. 2006, 461; BCS 2012b.)

Yllä mainittujen taitojen lisäksi The British Computer Society (BCS 2012b.) mukaan testaajan tulisi olla sinnikäs ja hänen tulisi pystyä hallitsemaan oman ajan käyttöä.

Testauspäälikkö on toinen keskeinen rooli hyväksymistestauksessa. Testauspäälikön tehtävä on johtaa, koordinoida ja toteuttaa hyväksymistestaus. Hän on vastuussa siitä, että testausvaihe suoritetaan ajallaan, budjetissa ja oikealla laadulla. Hän raportoi testausvaiheen prosessin etenemisestä ja testauksen kohteena olevan järjestelmän laadusta. Testien aikana hän myös ylläpitää asiakaskontakteja liittyen testauksen aihepiireihin. (BCS 2012c; Broekman ym. 2006, 466.)

Testauspäälikön tehtävät ajoittuvat koko testijakson ajalle ja tehtävät sisältävät paljon erilaisia aktiviteetteja. Prosessin alussa laaditaan testaus suunnitelma ja hankitaan sille hyväksyntä. Testauspäälikön tulee myös ylläpitää kyseistä suunnitelmaa. Testauspäälikkö vastaa hyväksymistestauksen suorittamisesta ajallaan ja budjetissa laatuvaatimusten mukaisesti ja raportoi siitä ohjausryhmälle sovitusti. Hän myös koordinoi ja arvioi kaikkia testausprosessin aktiviteetteja ja testaajia testien aikana. Hän johtaa sisäistä konsultointia sekä laadunvarmistusta ja antaa neuvoja ja opastusta työyhteisölle testauksen suunnittelun ja läpi viennin aikana. Hän edustaa testausta projektiryhmän kokouksissa ja yhtenä tärkeänä tehtävänä on juuri projektin riskien tunnistaminen, ennakointi ja raportointi. Myös testauspolitiikan luonti on hänen tehtävälustallaan. (BCS 2012c; Broekman ym. 2006, 464-467.)

Testauspääliköltä vaaditaan laaja-alaista osaamista. Hänen tulee pystyä hahmottamaan vaikeat ja monimutkaiset ongelmat kokonaisuuksina ja pilkkomaan ne pienemmiksi ja paremmin ymmärrettäviksi konsepteiksi tai malleiksi. Hänellä täytyy olla hyvä suunnittelu- ja organisointikyky ja hänen täytyy pystyä tekemään päätöksiä aikailematta, jotka perustuvat annettuihin faktoihin, tilanteeseen ja prioriteetteihin. Hänen on myös osattava katsoa asioita yli eri toimintojen rajojen ja ymmärrettävä muiden toimintojen funktiot. (BCS 2012c.)

Muita tärkeitä taitoja ja osaamisia ovat:

- järjestelmäkehityksen erilaiset elinkaarimallit ja niihin liittyvät aktiviteetit,
- testausprosessin ja siihen liittyvien aktiviteettien tunteminen, sekä erinomaiset testien johtamisen taidot,
- ymmärtää konfiguraationhallinta ja sen merkitys IT palvelujen hallintaan,

- laaja ymmärrys automatisoinnista ja sen hallinnasta,
- ymmärrys arkkitehtuuriin ja järjestelmäkehityksen työkaluista,
- hyvät arviointi ja raportointi taidot liittyen IT aktiviteetteihin,
- erinomaiset projektinhallinta- ja kontrollointitaidot, sekä kokemusta projektien johtamisesta,
- erinomaiset tiimityöskentely- ja motivointitaidot,
- erinomaiset kommunikointitaidot, sekä suullinen, että kirjallinen kommunikointi,
- hyvä ymmärrys liiketoiminnoista ja niiden linjaorganisaatioista. (BCS 2012c; Broekman ym. 2006, 467.)

Yllämainittujen taitojen ja osaamisten lisäksi BCS:n (2012a.) mukaan testauspäälliköllä tulisi olla sekä Intermediate Certificate in Software Testing sertifikaatti ja 6-10 vuotta työkokemusta kyseiseltä alalta.

Testauspäällikön ja testaajien lisäksi on testaustiimissä hyvä olla nimetty testiympäristön koordinaattori. Hänen tehtävänä on koordinoida testiympäristöjä ja varmistaa niiden, sekä testaustyökalujen ja työasemien oikea-aikainen ja hyvä saatavuus. Hän toimii linkkinä testaustiimin ja testiympäristön toimittajan välillä. Testien eri vaiheissa koordinaattori auttaa testiympäristöön kohdistuvien häiriöiden ja vikojen selvittämisessä, avustaa testiympäristöön liittyvien kulujen ja vaatimusten kartoituksessa, sekä avustaa testiympäristöön kohdistuvien muutosten vaikutusten analysoinnissa. (Broekman ym. 2006, 724-725.)

4 Toteutettu haastattelututkimus

Seuraavissa kappaleissa kerrotaan tehdystä tutkimuksesta. Kappaleessa 4.1 kerrotaan tutkimuksen taustasta, perusteista ja tavoitteista, sekä haastateltujen ammattilaisten taustoista ja syyt miksi juuri heidät on valittu tähän kvalitatiiviseen tutkimukseen haastateltavaksi. Kappaleessa 4.2 käydään läpi kaikki tutkimuskysymykset ja valotetaan taustoja miksi nämä kysymykset on valittu. Kappaleessa 4.3 on kuvattu miten tutkimus suoritettiin ja kappaleesta 4.4 löytyy tutkimusanalyysi eli tutkimuksen tulokset.

4.1 Tausta ja perusteet

Tutkimus suoritettiin kvalitatiivisena eli laadullisena tutkimuksena ja haastateltavien haastattelut tehtiin teemahaastattelun avulla. Tutkimuksessa tärkeimpänä asiana oli saada laadukkaita vastauksia, vastauksien määrän sijasta. Tällä lähestymistavalla haluttiin varmistaa, että tutkimuksen asiakkaan pyytämät lopputuotokset eivät ole vain teoriasta poimittua tietoa, vaan niihin on saatu käytännönläheinen näkökulma. Haastatteluilla haettiin myös ideoita ja vinkkejä hyväksymistestauksen johtamiseen ja prosessin suorittamiseen.

Tutkimuksen kohteena on tietojärjestelmätestaus ja ennen kaikkea asiakkaan vastuulla oleva hyväksymistestausprosessi ja siihen kuuluvat tehtävät, sekä niihin liittyvät roolit. Teemahaastatteluja hyväksikäyttäen haluttiin varmistaa se, että seuraaviin asiakkaan asettamiin tutkimuskysymyksiin saadaan käytännönläheinen näkökulma. Tutkimuskysymykset ovat seuraavat:

- Mikä on hyväksymistestauksen prosessi ja mitä tehtäviä sen suorittamiseen kuuluu?
- Milloin asiakasorganisaation tulee aloittaa työt hyväksymistestausta silmällä pitäen?
- Onko hyväksymistestaus ainoa testauksen taso, jota tilaajan tulee johtaa ja voiko eri testaustasojen testausta johtaa yksi ja sama henkilö?
- Onko valitulla ohjelmistokehityksen viitekehityksellä vaikutusta hyväksymistestauksen prosesseihin (ketterän kehityksen malli vs. V-malli)?
- Mitä rooleja hyväksymistestauksen tekemiseen ja johtamiseen tarvitaan, sekä mitä pätevyyyksiä ja kyvykkyyksiä tähän vaaditaan?

- Mitä tehtäviä hyväksymistestauksessa asiakasyrityksen liiketoiminnan tulee tehdä ja mitä tehtäviä vastaavasti kuuluu tietohallinnolle?
- Mitä muita asioita hyväksymistestauksen organisoinnissa tulee ottaa huomioon?

Varmistaakseni tutkimustulosten laadun valitsin haastateltaviksi kolme oman alansa ammattilaista, joilla on erilainen tausta. Haastateltavat olivat testausjärjestelmiä kehittävän yrityksen toimitusjohtaja, erään suomalaisen suuryrityksen ohjelmistokehityksestä vastaava henkilö, sekä kokenut projektihallinnan ja testauksen ammattilainen. Olen itse työskennellyt tietohallinnon alueella viimeiset kahdeksan vuotta ja sen vuoksi haastateltavat oli helppo tunnistaa, koska olen tutustunut heihin aiempien toimeksiantojeni kautta.

Tietojärjestelmätestauksen ammattilaisen (H1) valintaan vaikutti hänen taustansa ja omistautumisensa tietojärjestelmätestaukseen. Työskentelin hänen kanssaan aiemmin eräässä ohjelmistokehitysprojektissa ja hänen ammattitaitonsa oli vakuuttavaa. Hän on tällä hetkellä erään testauksen hallintaan ohjelmistoja kehittävän yrityksen toimitusjohtaja. Hänellä on vuosien kokemus vaativien ohjelmistokehitysprojektien testauspäällikkönä toimimisesta. Häntä haastatteleamalla halusin saada tutkimukseen testauspäällikön perspektiivin.

Ohjelmistokehityksen ammattilaisen (H2) valinta oli myös helppo, koska tunnen hänet jo pitkän ajan takaa ja tunnen hänen taustansa hyvin. Hän toimii tällä hetkellä eräässä suuressa tietotekniikka- ja tuotekehityspalveluja tarjoavassa yhtiössä ja vastaa terveydenhuoltoon suunnattujen ohjelmistojen kehittämisestä. Näiden ohjelmistojen toimivuus on sanomattakin tärkeää ja he panostavat ohjelmistokehitysvaiheen testaukseen paljon aikaa ja resursseja. Häntä haastatteleamalla halusin saada tutkimukseen ohjelmistoja rakentavan toimittajan näkökulman.

Projektihallinnan ja testauksen ammattilaisen (H3) valintaan päädyin esimieheni vinkistä. Hän toimii tällä hetkellä testauspäällikkönä suuressa vähittäiskauppaa harjoittavassa yrityksessä ja hänellä on pitkä tausta tietojärjestelmätestauksesta ja projektien johtamisesta. Hänellä on International Software Testing Qualifications Boardin myöntämä

ISTQB foundation sertifikaatti. Häntä haastatteleamalla halusin saada tutkimukseen projektipäällikön ja testauspäällikön näkökulman.

4.2 Tutkimuskysymykset ja perusteet

Tutkimuskysymykset laadittiin teoriaosuuden valmistumisen jälkeen ja kysymykset voidaan jakaa kolmeen eri kategoriaan. Nämä kategoriat ovat hyväksymistestauksen suorittamiseen liittyvät kysymykset, hyväksymistestauksen organisointiin, rooleihin ja vastuisiin liittyvät kysymykset, sekä hyväksymistestauksen apuvälineisiin liittyvät kysymykset. Kyselylomake sisältää ainoastaan avoimia kysymyksiä ja haastatelluilta odotetaan laadukkaita ja hyvin perusteltuja vastauksia.

Hyväksymistestauksen suorittaminen

Teoriaosuutta kirjoittaessani huomasin hyväksymistestausprosessien olevan hyvin samankaltaisia kaikissa kolmessa löytämässäni esimerkissä (katso luku 3.1). Tämän vuoksi päätin, että kysymysasettelu täytyy tehdä ohjelmistokehityksen näkökulmasta eli pyritään löytämään V-mallisen ohjelmistokehityksen ja ketterän kehityksen mallikohtaisia eroavaisuuksia hyväksymistestaukseen nähden. Tällä lähestymistavalla uskoin saavani paremmin selville hyviä ideoita prosessiin ja sen kehittämiseen.

Teemahaastattelun kysymyksillä 1-3 (liite 3.) haetaan vastauksia V-mallisen järjestelmäkehityksen, sekä ketterän kehityksen eroavaisuuksista asiakkaan näkökulmasta. Kysymyksellä 1 haetaan hieman provosoiden hyvää aloitusta haastattelulle ja yritetään löytää esimerkkejä mallien eroavaisuuksista, joilla saattaa olla vaikutusta syntyvään hyväksymistestauksen prosessikuvaan. Kysymykset 2-3 ovat enemmän täydentäviä kysymyksiä ja näillä yritetään selvittää, onko valitulla ohjelmistokehitysprosessilla vaikutusta hyväksymistestausprosessiin. Tuleeko hyväksymistestausprosessista laatia kaksi erillistä prosessikuvausta vai selvittääkö yhdellä prosessikuvauksella?

Kysymyksillä 4 ja 7 (liite 3.) haetaan ideoita ja vinkkejä hyväksymistestauksen suorittamiseen tutkimuksen tavoitteiden mukaisesti. Haastateltavien kokemuksen perusteella

odotettavissa on erittäin tärkeitä vastauksia, joilla saattaa olla vaikutusta sekä hyväksymistestauksen prosessiin, että sen johtamiseen.

Kysymykset 5-6 (liite 3.) auttavat löytämään vastauksen tutkimuskysymyksiin, jotka käsittelevät asiakkaan ohjelmistokehitysprojektin aikaisia vastuita. Kysymyksellä 5 haetaan vastausta siihen, kuinka järkeä organisointi hyväksymistestauksen suorittamiseen tarvitaan. Kysymyksessä 6 keskitytään löytämään käytännön näkökulmaa luvussa 2.1 kuvattuihin testautasoihin ja niiden omistajuuteen.

Hyväksymistestauksen organisointi, roolit ja vastuut

Teoriaosuutta kirjoittaessani tunnistin kolme roolia, jotka hyväksymistestauksen prosessin suorittamiseen tarvitaan. Tämän osa-alueen kysymykset on suunnattu löytämään mahdollisia uusia rooleja ja tarkentamaan löydettyjen roolien vastuita ja ominaisuuksia.

Kysymyksien 8-10 (liite 3.) avulla halutaan selvittää ammattilaisten mielipide hyväksymistestauksen prosessin suorittamiseen tarvittavista henkilöistä. Onko käytännön testijaksoilla joku rooli erityisen tärkeä ja onko havaittavissa jokin uusi rooli, jota teoriassa ei ole kuvattu. Aikomuksena on selvittää myös heidän kantansa liiketoiminnan ja tietohallinnon roolien ja vastuiden eroista.

Kysymyksien 11-12 (liite 3.) avulla haetaan vastauksia testauspäällikön ja testaajan tärkeimmistä ominaisuuksista ja luonteenpiirteistä. Teoriaosuudessa tätä on jo käsitelty, mutta tarkoituksena on tunnistaa roolin kaikkein tärkeimmät vaaditut ominaisuudet.

Kysymys 13 (liite 3.) on tarkka kysymys liittyen hyväksymistestauksen prosessissa huomioitaviin asioihin. Tämän kysymyksen toivotaan antavan varmennusta ja tarkennusta teoriassa kirjattuihin asioihin hyväksymistestauksen tavoitteista. Kysymyksellä toivotaan saavan myös vastauksia testauspäällikön ja testaajan tehtävien priorisointiin.

Hyväksymistestauksen apuvälineet

Kysymyksillä 14 ja 15 ei suoranaisesti ole mitään tekemistä tutkimuksen tavoitteiden kanssa, mutta teoriaosuuden kokoamisen yhteydessä sekä automaatiotestaus, että tes-

taushallinnan työvälineiden tärkeys nousi useassa paikassa esille. Näiden kysymysten tarkoituksena on ymmärtää näiden työkalujen tärkeys käytännön hyväksymistestauksen johtamisessa. Vastauksien avulla voidaan löytää mahdollisia jatkotutkimuskohteita. Koska nämä kysymykset eivät suoranaisesti liity tutkimukseen, käsitellään näiden kysymysten vastauksia yhteenveto ja pohdintaosuudessa.

4.3 Haastattelujen toteutus

Haastattelut toteutettiin teemahaastatteluna 9.5.2012 – 30.5.2012 välisenä aikana. Teemahaastattelun kysymysrunko (liite 3.) toimitettiin haastateltaville sähköpostitse noin 2 viikkoa ennen haastattelua. Haastattelut oli alun perin tarkoitettu tehtäväksi kasvotusten, joka myös toteutui haastateltavien H1 ja H2 osalta. Haastateltavan H3 haastattelu pidettiin Microsoft Live Meeting -neuvotteluohjelman avulla.

Haastattelujen laadun varmistamiseksi kaikki haastatellut nauhoitettiin ja tärkeimmät muistiinpanot kirjattiin Microsoft One Note -ohjelman avulla haastattelun yhteydessä. Haastattelujen jälkeen jokainen nauhoitus litteroitiin eli äänitallenne puhtaaksikirjoitettiin tekstimuotoon.

4.4 Tutkimusanalyysi

Tässä osiossa on kuvattu tämän tutkimuksen tulokset. Ensin käsitellään hyväksymistestauksen prosessiin ja sen suorittamiseen liittyviä tutkimuskysymyksiä, jonka jälkeen käsitellään hyväksymistestauksen organisointiin, rooleihin ja vastuisiin liittyviä kysymyksiä.

Hyväksymistestauksen suorittaminen

Tutkimuksen yhtenä tärkeimpänä asiana oli selvittää mikä on hyväksymistestauksen prosessi ja mitä tehtäviä siihen kuuluu. Tutkimuksen tuloksena syntyi liitteessä 4 kuvattu hyväksymistestauksen prosessi, sekä liitteissä 5 ja 6 kuvatut prosessin tehtävät. Prosessikuvauksen ja tehtävälistojen pohjana on käytetty kaikkia kolmea teoriaosuudessa kuvatuista malleista. Teoriassa kuvatut mallit olivat niin kattavia, ettei asiantuntijahaas-

tatteluissa löytynyt kuin yksi lisäys, joka mielestäni oli tärkeä lisätä prosessikuvaan ja tehtävälistoille.

Tämä lisäys oli H1:sen antama vastaus, jossa hän suositteli hyväksymistestaukseen siirtyvän järjestelmän käyttöönoton simulointia testausympäristön luovutuksen yhteydessä. Haastattelun yhteydessä hän kertoi, että yksi suurimmista ongelmista hyväksymistestausvaiheessa on se, että testausympäristö ei ole vakaa tai sieltä puuttuu testauksen kannalta olennaista tietoa. Hän suosittelee, että ennen hyväksymistestauksen suoritusvaiheen aloittamista, järjestelmän käyttöönottosuunnitelma tehtävineen olisi valmis. Tätä suunnitelmaa testataan testiympäristön päivityksessä ja näin ollen simuloidaan tulevaa käyttöönottovaihetta. Käyttöönottosuunnitelmassa tulisi olla mukana teknisten päivitysten lisäksi myös Master Datan päivitykset.

Muuta huomion arvoista hyväksymistestauksen prosessin ja tehtävien suorittamiseen liittyen tuli haastatellulta H3. Hän suositteli hyvien esimerkkien kautta hyväksymistestauksen määrittelyvaiheen testidokumentteihin panostamista. Hänen mielestään testidokumenttien ja skenaarioiden suunnitteluun täytyy käyttää tarpeeksi aikaa ja niihin tulee panostaa kunnolla. Hän on törmännyt urallaan tilanteisiin, joissa hyväksymistestauksessa on käytetty samoja testidokumentteja kuin mitä kehitysvaiheen testeissä on käytetty ja tämä on johtanut negatiiviseen lopputulokseen.

Tutkimuksen toinen tavoite oli selvittää **milloin asiakasorganisaation tulee aloittaa työt hyväksymistestausta silmällä pitäen**. Teoriaosuuden ja haastattelujen perusteella työt voidaan aloittaa heti kun sopimus toimittajan kanssa on allekirjoitettu. Hyväksymistestauksen valmistelut aloitetaan siis samaan aikaan kun toimittaja aloittaa toiminnallisten määrittelyjen työstämisen. Huomioitavaa on, että vaikka vaatimusmäärittelyiden tekemistä ei lasketa hyväksymistestauksen prosessiin tai sen tehtäviin, on niiden laadulla suuri merkitys hyväksymistestauksen suorittamiseen. Jokainen haastateltava oli sitä mieltä, että vaatimusmäärittelyjen ja järjestelmäkehitysprojektin suunnitteluvaiheen tehtävät pitää suorittaa ammattimaisesti, oli valittu kehitysmalli sitten mikä tahansa. Muutoin hyväksymistestauksen suorittamisvaiheessa voi tulla ikäviä yllätyksiä ja aikataulut voivat venyä. Haastateltava H1 painotti muutoksen vaikutusanalyysin laadukasta

toteuttamista, jotta testien suunnittelussa osataan kattaa kaikki olennaisimmat testattavat asiat.

Tutkimuskysymykseen ”**Onko hyväksymistestaus ainoa testauksen taso, jota tilaajan tulee johtaa ja voiko eri testaustasojen testausta johtaa yksi ja sama henkilö?**” vastaus on seuraavanlainen: käytännössä kokonaisuus ratkaisee. Haastateltava H2 kertoi, että hänen kokemuksen mukaan jo järjestelmätestausvaihe toteutetaan yhdessä asiakkaan kanssa, sitä tosin johtaa toimittajan edustaja. Teoriassa kuvatussa V-mallissa toimittajan vastuulla on johtaa ja toteuttaa yksikkö-, integrointi- ja järjestelmätestauksen tasot. Asiakkaan vastuulle jää hyväksymistestauksen johtaminen ja suorittaminen.

Yksikkö- ja integraatiotestauksen osalta haastateltava H2 kommentoi, että asiakkaan mielenkiinto kohdistuu yleensä hänen päivittäiseen työhönsä. Silloin yksikkö- ja integrointitestaus menevät liian yksityiskohtaiselle tasolle. Haastateltava ei myöskään näe hyvänä sitä, että asiakas tulee neuvomaan toimittajaa miten kehitysaikaiset testit tulisi suorittaa. Harvemmin asiakkaalta löytyy tarvittavaa kompetenssia testauksen johtamisen alueelta. Asiakkaan on hyvä antaa vaatimuksia toimittajan suorittamille testeille, mutta työtapoihin ja käytettäviin menetelmiin on turha ottaa kantaa. Haastateltavalta H3 saatiin samankaltainen vastaus ja hän oli sitä mieltä, että kun ammattilaiset ovat järjestelmää rakentamassa on heillä käytettävissä myös ammattitaitoiset testaajat ja testauksen metodit.

ERP ohjelmistokehityksestä puhuttaessa haastateltava H1 näkee asian toisin. Hänen mielestään jo integrointitestausvaihe toteutetaan siten, että työpanoksesta 70% tulee asiakkaalta ja 30% toimittajalta. Hän perustelee vastausta sillä, että asiakas tietää huomattavasti paljon paremmin miten ohjelmiston tulisi toimia. Sen vuoksi jo integrointi- ja järjestelmätestausvaiheen voi johtaa asiakkaan nimeämä testauspäällikkö ja suorittaa asiakkaan nimeämät testaajat. Hän myös suosittelee, että asiakas tekisi yksikötason testausdokumentit toimittajalle, joita sitten toimittaja täydentää.

Ohjelmistokehitysprojektin alkuvaiheissa tulee laatia projektin testaussuunnitelma ja samalla päättää testauksen tasoista, resursseista ja siitä miten testausta johdetaan. Testa-

uksen tasoista ja organisoinnista täytyy toimittajan ja asiakkaan sopia yhteisesti. Näiden vastausten perusteella testausvaiheet voi johtaa yksi ja sama henkilö.

Tutkimuskysymykseen ”**Onko valitulla ohjelmistokehityksen viitekehyksellä vaikutusta hyväksymistestauksen prosesseihin (ketterän kehityksen malli vs. V-malli)?**” tutkimuksessa selvisi, että prosessi ja tehtävät ovat samat, käytettiin sitten V-mallista tai ketterän kehityksen malleja. Testaustasot ovat myös samat. Tätä tukee sekä teoria, että haastateltavat H1 ja H2. Haastatellulla H3 ei ollut kokemusta ketterän kehityksen mallista. Toisaalta haastatteluissa kävi ilmi, että ohjelmistokehityksen valitulla lähestymistavalla on suuri merkitys hyväksymistestaukseen.

Suurin ero V-mallisen ja ketterän kehityksen lähestymistavan välillä hyväksymistestauksen näkökulmasta on se, että hyväksymistestausta suoritetaan ketterässä kehityksessä jo pyrähdysten sisällä. Tämä tarkoittaa sitä, että julkaisun tullessa hyväksymistestattavaksi, on osia siitä jo hyväksymistestattu. Hyväksymistestauksen suorittamisen työmäärä jakaantuu siis koko projektin ajalle eikä kasaannu projekti loppuvaiheeseen. Tällä on positiivinen vaikutus julkaisuvaiheen hyväksymistestauksen suorittamisen pituuteen ja siinä vaiheessa löydettyjen virheiden määrään. Haastateltavien H1 ja H2 vastaukset puoltavat tätä väittämää.

Kysyttäessä V-mallisen kehityksen eduista ja haitoista H1 ja H2 kertoivat, että kyseisessä mallissa asiakas pääsee käsiksi järjestelmään hyvin myöhäisessä vaiheessa. Vasta tässä viimeisessä testausvaiheessa löydettyjen virheiden korjaamisen hinta nousee. Yllätyksiä voi tulla ja ohjelmistoon voidaan joutua tekemään isojaakin muutoksia, jotka pitkittävät projektia entisestään. Hyväksymistestausvaiheen kesto on vaikea määritellä ja testit saattavat kuormittaa asiakkaan resursseja pitkiäkin aikoja. Ketterän kehityksen mallilla hyväksymistestausta suoritetaan jo pyrähdysten aikana ja näin pienennetään loppuvaiheen työkuormaa.

H2:sen mielestä V-mallisessa lähestymisessä riskit kasaantuvat loppuvaiheeseen ja pahimmassa tapauksessa saatetaan huomata, että on sijoitettu suuria summia rahaa kehitykseen, mutta lopputulos ei palvele liiketoimintaa halutusti. Ketterässä kehityksessä tämä olisi ymmärretty jo aikaisemmassa vaiheessa ja asiaan olisi voitu puuttua.

Haastatellut kertoivat myös haasteista vaatimusmäärittelyjen osalta. Vaatimusmäärittelyt eivät nimittäin ole yleensä tarpeeksi kattavia. Kehityksen aikana tulee aina eteen asioita, joita ei ole osattu projektin alkuvaiheissa ottaa huomioon. V-mallisessa lähestymisessä tämä on iso ongelma, koska vaatimusmäärittelyt ovat ”pyhiä” ja niitä ei muuteta. Ketterät menetelmät lähtevät siitä lähtökohdasta, että alussa luodut dokumentaatiot eivät ole täydellisiä ja niitä voidaan pyrähdysten aikana täydentää. Ketterässä kehityksessä mahdolliset ongelmat dokumenteissa havaitaan aikaisemmin ja muutoksia määrityksiin ja tuotteeseen voidaan tehdä pyrähdysten sisällä. Asiakkaalta saadaan välittömästi palautetta kun kehitys on valmis ja toimittaja voi välittömästi tehdä vaadittavat toimenpiteet asiakkaan tarpeen tyydyttämiseksi. Huomioitavaa kuitenkin on, että julkaisuvaiheen hyväksymistestaus tulee peilata alkuvaiheissa määriteltyihin hyväksymiskriteereihin! Jos kehityksen aikana löydetään uusia hyväksymiskriteerejä, on niiden lisäämisestä listalle sovittava toimittajan kanssa.

Haastateltu H3 taas sanoo V-mallisen lähestymistavan olevan hyväksyttävissä, mutta sitä pitää asiakaskohtaisesti soveltaa. Kyseinen lähestymistapa vaatii resursseja ja edellyttää resursseilta paljon. Hän oli sitä mieltä, että jos kehityksen alkupään tehtävät eivät ole kunnossa (kuten vaatimusmäärittelyt), tulee se vaikuttamaan suoraan hyväksymistestauksen suorittamiseen.

Ketterän kehityksen malli sai V-malliin verrattuna haastatelluilta paljon enemmän positiivista palautetta. Silti siitä löytyi myös negatiivisia asioita. H2 kertoi, että toimittajan näkökulmasta Scrum-malli vaatii enemmän yhteistyötä asiakkaan kanssa. Heille täytyy järjestää enemmän erilaisia tilaisuuksia kuten järjestelmäkatselmuksia ym. vastaavaa. Haastateltava H1 totesi, että ketterien kehityksen mallien ollessa hyvin intensiivisiä, voi toimituksen hinta olla asiakkaan kannalta kalliimpi verrattuna V-malliseen lähestymistapaan.

Hieman yllättävänä positiivisena piirteenä ketterien kehityksen mallissa haastateltava H2 kertoi, että heidän siirryttyä toteuttamaan ohjelmistokehitystä Scrum-mallilla on dokumenttien laatu parantunut huomattavasti. Syitä tähän on esimerkiksi dokumentaatio keskittyminen pienempiin alueisiin. Silloin dokumentaation kirjoittaminen on oh-

jelmoijan kannalta mielekästä kun kirjoitusurakka ei tunnu mammuttimaisen suurelta. Dokumentointi myös tapahtuu pienemmällä aikajänteellä ja sitä katselmoidaan paljon tiiviimmin. Dokumentaation laadun parannuksella on suora vaikutus asiakkaan hyväksymistestauksen laatuun, koska näitä dokumentteja tarvitaan testauksen määrittelyvaiheessa. Dokumentaation parantuminen auttaa asiakasta myös tulevaisuudessa, koska hyvät järjestelmädokumentit auttavat järjestelmän tukitoimintoja ratkaisemaan ongelmatilanteet nopeammin ja laadukkaammin.

Hyväksymistestauksen organisointi, roolit ja vastuut

Prosessien ja tehtävien lisäksi tutkimuksen tavoitteena oli selvittää **mitä rooleja hyväksymistestauksen tekemiseen ja johtamiseen tarvitaan, sekä mitä pätevyksiä ja kyvykkyksiä tähän vaaditaan**. Tutkimuksessa selvisi, että hyväksymistestauksessa on kaksi keskeistä roolia, jotka ovat testauspäällikkö ja testaaja. Heitä tukee testausympäristön koordinaattori, jolle on myös kuvattu tehtäviä hyväksymistestausprosessissa (liite 4). Testauspäällikölle ja testaajalle syntyi roolikuvaukset, jotka ovat löydettävissä liitteet osiossa. Testaajan roolikuvaus löytyy liitteestä 7 ja testauspäällikön roolikuvaus löytyy liitteestä 8. Roolikuvaukset syntyivät teorian ja haastattelujen pohjalta ja ne kuvaavat mielestäni hyvin kummankin keskeisen roolin tehtäviä ja vaadittavia kyvykkyys-

Haastattelujen ja teorian perusteella testauspäällikkö on erittäin tärkeässä roolissa hyväksymistestauksen onnistumisen kannalta. Jokainen kolmesta haastatellusta nimesi testauspäällikön tai testauksen vastuuhenkilön ensimmäisenä, kysyttäessä hyväksymistestauksessa tarvittavista rooleista. Teoriaosuudessa löytyvät vaatimukset puhuvat myös sen puolesta, että testauspäällikkö on erittäin kriittinen rooli hyväksymistestauksen onnistumiselle. Testauspäälliköltä vaaditaan vahvaa henkilöjohtajuutta ja paljon kokemusta myös muista IT toiminnoista sekä liiketoiminnan osa-alueista.

Haastateltava H1 painotti, että varsinkin isoissa hankkeissa testauspäällikkö ei voi olla osa-aikainen oman työn ohella tehtävä rooli. Roolissa toimivalla henkilöllä täytyy olla aikaa hyväksymistestauksen johtamiseen ja hänen täytyy pystyä keskittymään siihen sataprosenttisesti. Hän painotti myös testauspäällikön objektiivisuutta eli testauspäällik-

kö ei saa ajaa omia etujaan vaan hänen täytyy katsoa kokonaisuutta ja pyrkiä saavuttamaan yrityksen kannalta paras mahdollinen lopputulos. Haastateltava on urallaan törmännyt tilanteisiin, jossa testauspäällikkö on tullut esimerkiksi talousosastolta ja kyseinen testauspäällikkö oli ajanut vain talouden intressejä testien aikana.

Haastateltava H2 näki testauspäällikön roolin hieman erilaisena kuin muut haastatellut ja hän kuvaili kyseistä hyväksymistestauksen roolia testauksen vastuuhenkilöksi, jonka tehtävänä on huolehtia projektille laadukkaat resurssit liiketoiminnasta suorittamaan hyväksymistestausta. Hänen täytyy myös ymmärtää testauksen arvokkuus ja hyöty. Haastatellun mukaan asiakkaan testauskoordinaattorin ei tarvitse olla testauksen ammattilainen, mutta hänen tulee omata projektipäällikön ja testauspäällikön kaltaisia ominaisuuksia, joita on kuvattu tarkemmin liitteessä 8. Haastateltavan mielestä toimittajalta täytyy tulla projektiin testauksen johtamisen ammattilaisen kompetenssit, koska ei voida olettaa, että asiakkaalta löytyy kyseisiä kompetensseja. Toimittajalla tällaisia kompetensseja on pakko olla.

Haastateltava H3 oli samoilla linjoilla muiden haastateltujen kanssa testauspäällikön vaatimuksista ja hän vielä painotti kokonaisuuksien hallintaa, mutta myös yksityiskohtien ymmärtämisen tärkeyttä. Haastateltavalla oli itsellään kehittäjän tausta ja hänen mielestään kyseisestä taustasta on testauspäällikön toimenkuvassa paljon hyötyä.

Mielestäni teorian ja haastattelujen perusteella voidaan sanoa, että testauspäälliköllä tulisi olla ISTQB Advanced / Test manager sertifiointi. Kyseisiä kursseja Suomessa järjestää ainakin Sovelto ja Tieturi.

Hyväksymistestauksen toinen tärkeä rooli on testaaja. Hyväksymistestauksen testaajien tulee tuntea liiketoimintaprosessit hyvin. Mieluiten he tulevat liiketoiminnasta. Liiketoiminnan henkilöt tuntevat prosessinsa kaikkein parhaiten ja voivat arvioida tehdyn ratkaisun soveltuvuutta liiketoiminnan tarpeisiin ja vaatimuksiin. Teoriaosuudessa kerrotaan testaajan olevan itsessään riski, koska testaus on riskien vähentämisen toimenpide. Jos testaajalla ei ole kunnollista pätevyyttä ja hän ei testaa asiankuuluvalla tarkkuudella, ei riskejä pystytä eliminoimaan. Haastateltavan H1 mukaan testaajalla täytyy olla kyky löytää virheitä ja halu auttaa niiden korjaamisessa.

Haastateltu H3 kertoi, että hyvien testaajien löytäminen liiketoiminnasta ei ole vaikeaa, mutta heidän aikansa on yleensä rajallista. Hyvien testaajien ajankäytöstä joutuu useasti taistelemaan ja jos testaajille ei anneta liiketoimintajohdosta aikaa testata, eivät he pysty siihen myöskään panostamaan. Suositeltavaa on, että testiresurssit kartoitetaan hyväksymistestausprosessin varhaisessa vaiheessa ja ajankäytöstä sovitaan sekä testaajan, että hänen linjaorganisaatiossa olevan esimiehensä kanssa.

Liiketoimintaprosessien hyvän tuntemuksen lisäksi täytyy testaajan pystyä laatimaan hyviä testidokumentteja ja testiskenaarioita. Hänen täytyy olla luova ja pystyä tekemään viekkaita skenaarioita, joilla mahdolliset viat pystytään paikallistamaan. Haastateltu H3 kuvaili testaajaa viekkaaksi ketuksi, joka pystyy laittamaan ohjelmiston sen äärirajoille ja varmistamaan, että se toimii odotetusti. Kaikki kolme haastateltua olivat sitä mieltä, että hyvä ja motivoitunut testaaja on kullan arvoinen projektin onnistumisen kannalta. Tarkeimmat testaajan osaamisen vaatimukset on koottu liitteessä 7, jotka on laadittu sekä teorian, että haastattelujen perusteella.

Teorian ja haastattelujen perusteella hyväksymistestauksen testaajalla ei tarvitse olla testauksen sertifikaattia. Jos testaaja testaa paljon ja on kiinnostunut kehittämään itseään ja laajentamaan osaamistaan testauksen osalta, olisi hänelle suotavaa järjestää ISTQB:n Foundation eli perustason koulutus.

Teoriaosuudessa mainitaan testausympäristön koordinaattorin rooli, jonka tehtävänä hyväksymistestauksessa on testausympäristön pystyttäminen ja ylläpito. Tämän rooli on tärkeä, mutta ei todennäköisesti vaadi kokopäiväistä resurssia. Haastatelluista kukaan ei tuonut esille juuri samankaltaista roolia kuin teoriaosuudessa on kuvattu, mutta uskon tämän roolin olevan tärkeä ainakin ERP- hankkeissa. Jonkun täytyy huolehtia testausympäristöstä ja sen valmisteluista ja ylläpidosta. Testauspäällikön aika ei välttämättä riitä tämän kaltaisten aktiviteettien suorittamiseen. Haastateltavalta H1 tuli kaikkein lähin kommentti liittyen tähän rooliin. Hän totesi, että prosessissa tulisi olla henkilö, joka vastaa käyttöönottosuunnitelmasta ja Master Datasta.

Muita haastatteluissa ilmitulleita rooleja ja tukiryhmiä olivat muun muassa ohjelmoijat, jotka auttavat ratkomaan löytyneitä virheitä ja vastaavat toiminnallisiin kysymyksiin. Erikseen mainittakoon, että haastateltava H1 painotti myös järjestelmäintegraatioiden asiantuntijakonsulttia, mikä on varsinkin ERP- hankkeissa hyvin tärkeässä roolissa. Haastateltava H3 kertoi, että testauksien aikana myös liiketoimintaprosessien omistajan tulisi olla tavoitettavissa. Testauksen eri vaiheissa voidaan joutua ottamaan kantaa esimerkiksi prosessien suunnitteluun.

Hyväksymistestauksella täytyy olla projektin tapaan päätöksentekoeelin, joka monessa tapauksessa on ohjausryhmä. Ohjausryhmä hyväksyy loppukädessä hyväksymistestauksen tulokset ja päättää miten edetään. H3 kertoi, että joissakin tekemissään projekteissa liiketoimintaprosessien omistajalla on ollut valtuutus antaa hyväksyntäpäätös jatkolle.

Tutkimuskysymykseen ”**Mitä tehtäviä hyväksymistestauksessa asiakasyrityksen liiketoiminnan tulee tehdä ja mitä tehtäviä vastaavasti kuuluu tietohallinnolle?**” vastauksia kysyttiin haastatelluilta asiantuntijoilta. H1 ja H2 mielestä testauspäällikkö on tietohallinnon täyttämä rooli. Tätä perusteltiin esimerkiksi sillä, että yleensä testattavana on paljon muitakin uudistuksia, kuten tietokanta- tai infrastruktuuripäivityksiä, joista liiketoiminnalla ei välttämättä ole mitään käsitystä.

Hyväksymistestaajan rooli oli kaikkien haastateltujen mielestä liiketoiminnan edustajien täytettävä. Tietohallinto ei voi tuntea liiketoimintaprosesseja niin hyvin kuin liiketoiminta itse. Tietohallinto voi olla mukana tukemassa toimittajaa toimittajan testeissä. Haastateltava H3 kertoi, että valitettavasti muutamissa tapauksissa liiketoiminnalla ei ole ollut antaa resursseja hyväksymistestausvaiheeseen. Tällaisissa tapauksissa tietohallinto on suorittanut hyväksymistestauksen liiketoiminnan puolesta. Tämä voi johtaa tilanteeseen, jossa tietohallinto hyväksyy muutoksen, joka myöhemmin käyttöönoton jälkeen huomataan liiketoiminnan kannalta käyttökelvottomaksi. Haastatellulla H2 oli kokemusta tällaisista tilanteista.

Liiketoiminnan tulee olla edustettuna ohjausryhmässä. Ohjausryhmässä tulee olla jokaisen liiketoimintayksikön edustaja jota muutos koskee. Henkilöillä täytyy olla tarpeeksi äänivaltaa saada tehdä päätöksiä liiketoimintayksikön puolesta. Liiketoiminnan tulee

myös täyttää liiketoimintaprosessien omistajan rooli. Testausympäristön koordinoijan rooli kuuluu oman kokemuksen mukaan tietohallinnolle.

Kysymykseen ”**Mitä muita asioita hyväksymistestauksen organisoinnissa tulee ottaa huomioon?**” olen nostanut kaksi mielestäni tärkeintä asiaa, joihin olen törmännyt teoriaa kootessani ja asiantuntijahaastattelujen yhteydessä. Nämä kaksi asiaa eivät suoranaisesti ole hyväksymistestausprosessin tehtäviä, mutta niillä on suuri vaikutus hyväksymistestauksen onnistumiseen.

Ohjelmistokehitysmallin valinta vaikuttaa hyvin paljon hyväksymistestauksen organisointiin. Teorian ja haastattelujen perusteella Scrum ketterän kehityksen lähestymistapana on parempi lähestymistapa verrattuna V-malliin. Scrum vaikuttaa varsinkin hyväksymistestauksen organisoinnin ja resursoinnin kannalta paremmalta vaihtoehdolta. Scrum-mallissa koko julkaisun hyväksymistestausaikataulu tulee olemaan tiiviimpi, koska pyrähdyksien aikana on tehty jo hyväksymistestaukseksi luokiteltavaa testausta. Näin ollen hyväksymistestauksen toimenpiteet jakaantuvat projektin eri vaiheille eikä pelkästään viimeisille viikoille.

Asiaan liittyen H1 mainitsi hänen mielestään suurimman syyn V-mallilla tehtävän ohjelmistokehityksen hyväksymistestauksen epäonnistumiselle. Se on asiakkaan hyväksymistestauksen resurssien puute. Laadukkaita resursseja testauksen johtamiseen ja suorittamiseen ei ole saatavilla sovitusti. Suositeltavaa on, että tarkka resurssisuunnitelma ja organisoituminen tapahtuvat vasta sen jälkeen kun on toimittajan kanssa sovittu, mitä ohjelmistokehitysmallia hyödynnetään ohjelmistokehitysprojehtin aikana. Jos päätetään edetä V-mallin mukaisesti, on asiakkaan ymmärrettävä resursoinnin kuormittuminen projektin loppuun.

Vaatusmääritys ja järjestelmän suunnitteluvaiheeseen panostaminen on toinen tärkeä elementti. Kaikki haastatellut olivat sitä mieltä, että jos vaatimusmääritysihin ei ole asiakkaan puolelta satsattu tarpeeksi, on sillä suora vaikutus hyväksymistestauksen onnistumiselle. Vaatusmääritysten täytyy olla tarpeeksi kattavia ja vaatimuksiin on liitettävä hyväksymiskriteerit. Olen urallani itsekkin törmännyt vastaaviin tilanteisiin, jossa hyväksymiskriteerejä ei ole vaatimusmäärityksissä ollenkaan huomioitu. Tällöin on hy-

vin vaikea suorittaa hyväksymistestausta, kun ei voida olla täysin varmoja mitä vaatimuksia vasten hyväksymistestausta tulee suorittaa.

Myös haastateltava H2 painotti hyväksymiskriteerien linkittämistä vaatimusmäärittelyihin. Jos siis hyväksymistestauksen suunnitteluvaiheessa huomataan, että dokumentaatioissa on puutteita, kannattaa tarkkaan harkita hyväksymistestauksen määrittelyvaiheen resursointia. Voi olla, että vaadittava määrittelyvaiheen työmäärän on normaalia korkeampi.

5 Yhteenveto ja pohdinta

Tutkin opinnäytetyössäni hyväksymistestauksen prosesseja, tehtäviä ja prosessin toteuttamiseen vaadittavia rooleja. Opinnäytetyö toteutettiin kvalitatiivisena eli laadullisena tutkimuksena ja haastattelut suoritettiin teemahaastattelua hyväksikäyttäen. Tutkimuksen toimeksiantajana toimi Sofigate Oy, joka on tietohallinnon asiantuntijapalveluita tarjoava yritys.

Tutkimuksen aiheen valintaprosessi alkoi jo syksyllä 2011, kun keskustelimme esimieheni kanssa laadunvarmistuksesta ja hyväksymistestauksesta ohjelmistokehitysprojekteissa. Olimme kumpikin urallamme törmänneet hankkeisiin, jossa hyväksymistestaus oli hoidettu huolimattomasti ja projektin tuotantoon siirron jälkeen liiketoiminta ei ollut ollut tyytyväinen järjestelmään ja sen liiketoimintahyötyihin. Tämä keskustelu päättyi siihen, että tehtäväkseni jäi suunnitella tutkimuksen tarkentavat tutkimuskysymykset ja ehdottaa lähestymistapaa tutkimukselle.

Tutkimusaiheen pureskeluun ja rajaamiseen meni yllättävän kauan aikaa. Aloitin syksyllä 2011 aiheen pohdinnan laadunvarmistuksen lähtökohdista ja aluksi luulin tekeväni tutkimuksen suuremmasta laadunvarmistuksen kokonaisuudesta hyväksymistestauksen sijasta. En kuitenkaan saanut minkäänlaista otetta aiheesta. Laadunvarmistus, ISO standardeineen, tuntui olevan liian iso kakku pureksittavaksi ja en saanut aihetta rajattua järkeväksi kokonaisuudeksi.

Tammikuussa 2012 pyysin kollegaltani apua aiheeseen liittyen, koska tiesin hänellä olevan paljon kokemusta laadunvarmistuksesta ja monista ohjelmistokehitysprojekteista. Ensin keskustelimme aiheesta puhelimitse, jonka jälkeen tapasimme Sofigaten Espoon konttorilla. Tämä kyseinen tapaaminen oli ns. jäänsärkijä kokous. Tässä kokouksessa hahmottui tarkka tutkimuksen kohde ja tarkemmat tutkimuskysymykset.

Tutkimustyöskentelyn aloitin helmikuun alussa. Aloitin työn tekemisen etsimällä testauksen lähdekirjallisuutta, sekä lukemalla internetistä ja eri foorumeilta aiheeseen liittyviä

artikkeleja. Lähteiden löytäminen oli aiheen tarkentumisen selkeydyttyä yllättävänkin helppoa. Löysin viisi hyvää kirjaa, jotka toimivat perustana teoriaosuudelle.

Teoriaosuuden kirjoittaminen oli työläs vaihe, joka valmistui huhtikuussa 2012. Välillä teoriaosuutta kirjoittaessani luulin hukanneeni punaisen langan. Silloin varasin istunnon ohjaavalta opettajaltani, sekä osallistuin Haaga-Helian järjestämille opinnäytetyöseminaaripäiville, jolloin sain sisällysluettelon rungon hyvään malliin. Tämän jälkeen kirjoitus ja tutkimuksen edistyminen jatkui verkkaisesti. Teoriaosuuden valmistuttua haastattelujen tutkimuskysymysten laatiminen oli helppoa, koska tiesin heti mihin asioihin halusin vastaukset asiantuntijoilta.

Asiantuntijahaastattelut sujuivat hyvin ja sain vastaukset haluamiini kysymyksiin. Eräs teemahaastattelun kysymys osoittautui tutkimuksen kannalta turhaksi, mutta tämä ei heikennä millään tavalla tutkimuksen lopputulosten laatua. Tutkimustulosten analysoinnin jälkeen luovutin tutkimuksen tulokset opinnäytetyön toimeksiantajalle kesäkuussa 2012. Tämä on muutamaa kuukautta aiemmin, mitä tutkimus virallisesti julkaisiin. Tämä sen vuoksi, että tutkimuksen tuotokset tarvittiin toimeksiantajan asiantuntijoiden käyttöön mahdollisimman pian niiden valmistuttua.

Kun mietitään asioita, mitä tekisin toisin seuraavalla kerralla, keskittyisin rajaamaan tutkimuskysymyksiä heti alussa todella tarkasti. Käytännössä menetin kaksi kuukautta aikaa, kun yritin tutkia hyväksymistestausta laadunvarmistuksen näkökulmasta. Ensin olisi pitänyt tutkia ja perehtyä itse hyväksymistestaukseen ja sitä kautta myös kokonaisvaltainen laadunvarmistus olisi avautunut. Kokonaisuutena opinnäytetyöprosessi oli opettavainen ja ensi kerralla tällaisen tutkimuksen tekeminen on minulle huomattavasti helpompaa.

Opinnäytetyöprosessin varrella olen mielestäni kehittynyt valtavasti tietojärjestelmätestauksen saralla ja tunnen pystyväni keskustelemaan asiasta alan ammattilaisten kanssa. Olen toiminut testauspäällikkönä opinnäytetyöprosessin aikana ja olen ollut erittäin onnekas, kun olen pystynyt yhdistämään työn ja tutkimuksen. Ne ovat tukeneet toinen toisiaan erinomaisesti. Kaiken kaikkiaan olen erittäin tyytyväinen aiheen valintaan, koska nyt tiedän mihin suuntaan ammatilliset tavoitteeni seuraavan kolmen vuoden aikana.

5.1 Tulosten yhteenveto

Tutkimuksen tavoitteena oli laatia hyväksymistestaukselle prosessikuvaus sekä tarkemat prosessin tehtävät, jotka toimisivat myös käytännössä. Tuotokset, jotka ovat löydettävissä liitteistä 4-6, perustuvat pitkälti teoriaosuudesta löytyvään TMAP malliin. Muista kahdesta mallista eli ISEB ja ITIL kuvauksiin löytyi yksittäisiä tehtäviä. Prosessikuvaukset on siis luotu suurimmalta osin teorian pohjalta, koska asiantuntijahaastattelusta ei paljastunut uusia merkityksellisiä tehtäviä, joita prosessi ja tehtäväkuvauksiin olisi pitänyt lisätä. Tämä mielestäni todistaa sen, että kirjallisuudesta löytyneet mallit ovat luotettavia ja niitä pystytään soveltamaan käytännössä.

Olin hieman yllättynyt kuinka hyvin testauksen prosesseja on jo kuvattu. Tosin valtaosa kirjallisuudesta ja malleista on suunnattu toimittajan ohjelmistokehittäjille ja näistä teoksista ei ollut apua tässä tutkimuksessa. Toinen yllättävä piirre hyväksymistestauksen prosesseihin ja suorittamiseen liittyen on se, että hyväksymistestauksen johtaminen asiakasyrityksissä saattaa olla suurempikin ongelma. Olen käynyt useita keskusteluja eri ihmisten ja asiantuntijoiden, kuten projektipäälliköiden, ohjelmistokehittäjien ja tietohallinnon asiantuntijoiden kanssa aiheen tiimoilta. Olen kuullut monen sanovan, että asiakkaiden hyväksymistestaus on hoidettu huonosti tai sen ammattimaiseen suorittamiseen ei kiinnitetä tarpeeksi huomiota. Toisaalta olen törmännyt myös tapauksiin, jossa hyväksymistestauksen prosesseja on johdettu ammattimaisesti.

Toivon, että saamme Sofigaten asiantuntijoiden kanssa jalostettua tutkimuksen tuloksena syntyneistä prosesseista sellaiset, että ne voidaan liittää osaksi Tietohallintomallia. Silloin prosessikuvaukset, tehtävät ja roolikuvaukset olisivat kaikkien yritysten saatavilla ilmaiseksi. Näin voisin olla omalta osaltani vaikuttamassa suomalaisten yritysten testauksen laatuun ja parantamassa käyttäjätyytyväisyyttä.

Tutkimuksen toisena tavoitteena oli tutkia hyväksymistestauksen keskeisimpiä rooleja ja tehdä niiden roolikuvaukset. Roolikuvauksia syntyi kaksi kappaletta, toinen testaajalle ja toinen testauspäällikölle. Kyseiset roolikuvaukset löytyvät liitteistä 7 ja 8. Kuvaukset ovat luotu sekä teorian, että haastattelujen pohjalta. Mielestäni teoria ja haastattelut tu-

kivat toinen toisiaan ja roolikuvaukset kuvaavat hyvin henkilöltä vaadittavia vaatimuksia ja henkilön vastuita.

Tiesin jo tutkimuksen alussa, että testauspäällikön rooli on tärkeä, mutta en osannut kuvitellakaan kuinka tärkeä se oikeasti on. Käytännössä testauspäällikön rooli on kriittinen koko ohjelmistokehitysprojektin kannalta. Jos testausta ei ole koordinoitu asianmukaisesti ja oikealla ammattitaidolla voi lopputulos olla katastrofaalinen. Jos testauspäällikkö ei ole hyvä henkilöjohtaja ja hänellä ei ole koordinoitukykyä, testaaminen ei etene aikataulun mukaisesti ja laatu ei vastaa vaatimuksia. Tämän tyyppisiä vastauksia sain myös haastatelluilta.

Testaajan rooli on myös tärkeä, mutta yksittäisen testaajan rooli ei ole niin kriittinen kuin testauspäällikön rooli. Tärkeimpänä asiana testaajaa valitessa pidän testaajan halua ja motivaatiota testauksen suorittamiseen. Vaikka henkilö olisi hyvä testaaja joka tuntee liiketoiminnan prosessit, mutta motivaatio puuttuisi, ei lopputulos voi olla hyvä. Testaajan roolikuvaus muodostui samaan tapaan kuin testauspäällikön rooli eli teorian ja haastattelujen perusteella. Mielestäni roolikuvaus on onnistunut ja kuvaa hyvin testaajan roolin vaativuutta.

Näiden tuotosten lisäksi tutkimuksen tilaaja antoi seitsemän tutkimuskysymystä joihin kaikkiin löytyi vastaus joko teoriasta, haastatteluista tai molemmista. Jokaiseen tutkimuskysymykseen löydettiin vastaus ja vastaukset on luettavissa luvussa 4.4. Teoriaosuudelle asetettiin kuusi kysymystä, joihin mielestäni löytyy vastaukset teoriaosuudessa.

5.2 Jatkotutkimuskohteet

Tutkimuksen aikana löytyi useita jatkotutkimuskohteita. Yksi mielenkiintoisimmista kohteista on selvittää, miten suomalaisissa yrityksissä hyväksymistestausta johdetaan ja kuinka paljon siihen panostetaan. Tämä olisi syytä tutkia siksi, että tutkimuksen aikana vastaukset hyväksymistestauksen johtamisen tilasta olivat ristiriitaisia.

Toinen jatkotutkimuskohde liittyy vaatimusmäärittelyihin ja niiden rakentamiseen siten, että ne tukevat riittävästi hyväksymistestausta ja sen suorittamista. Haastattelujen yhteydessä asiantuntijat kommentoivat, että yksi suurimmista ongelmista hyväksymistestauksen suorittamisessa on se, että vaatimusmäärittelyt eivät ole lainkaan sillä tasolla millä niiden pitäisi olla. Vaatimusmäärittelyistä saattaa esimerkiksi puuttua hyväksymiskriteerit, joita vastaan hyväksymistestaus tulee suorittaa. Tämä johtaa tilanteeseen, että toimittaja ei tiedä mitä vastaan hyväksyminen tullaan suorittamaan ja asiakkaan testaajat eivät tiedä milloin he voivat sanoa, että kyseinen muutos voidaan hyväksyä.

Suosittelen jatkokehityskohteeksi vaatimusmäärittelyprosessin kuvaamista ja siihen liittyvien tehtävien listaamista. Tutkimuksen tulisi keskittyä kuvaamaan yllä mainittu prosessi ja luoda esimerkki laadukkaasta vaatimusmäärittelydokumentista. Se voisi listata asiat, jotka vaatimusmäärittelyssä tulee olla ja mitkä asiat oltava ”nice to have” tasolla.

Kolmantena jatkotutkimuskohteena suosittelen tutkimaan mitä erilaisia testidokumenttien malleja on olemassa ja mikä olisi suositeltava malli. Tutkimuksen tavoitteena voisi olla hyvä testidokumenttimallin luominen ja testidokumenttiluomisprosessin kuvaaminen. Tämä tukisi hyvin myös tätä opinnäytetyötä ja myöhemmin voisi olla myös osa Tietohallintomallia.

Teoriaosuuden koostamisen aikana huomasin jatkuvasti törmääväni automaatiotestaukseen ja sen hyödyntämisen kysymyksiin. Päätin tutkimuskysymyksiä laatiessani kysyä ammattilaisilta mitä mieltä he ovat automaatiotestauksen käyttämisestä hyväksymistestausprosessissa.

Haastateltavat olivat sitä mieltä, että nykyisillä työkaluilla on vaikea saada toivottua kattavuutta, jotta testaaja voitaisiin korvata automaatiolla. Automaatiotestaukseen käytettävien työkalujen hintaa pidettiin myös kalliina ja ainoastaan hyväksymistestauksen tarpeisiin sille ei tulla saamaan positiivista kustannuslaskelmaa.

Mielestäni tämä on kuitenkin hyvä jatkotutkimuskohde ja tutkimuksessa suosittelen automaatiotestauksen työkalujen lähempää tarkastelua ja niiden hyödyntämistä hyväk-

symistestauksessa. En usko, että kolmen ihmisen haastatteleminen automaatiotestauksesta ilman teoriaosuutta ja tarkempaa tutkimusta on kyllin laadukas tulos.

Toinen samoista lähtökohdista tullut kysymys asiantuntijoille liittyi testaushallinnan työkaluihin ja niiden käyttöön hyväksymistestauksen aikana. Jokainen haastateltu oli sitä mieltä, että ilman testauksen hallinnan työkalua testauksen johtaminen on todella haastavaa. Toivoisinkin, että tulevaisuudessa tutkittaisiin markkinoilla olevia testausenhallinnan sovelluksia ja vertailtaisiin näitä keskenään.

Olen törmännyt tämän tutkimuksen aikana lukuisiin testausenhallinnan järjestelmiin, mutta en ole tutkinut niitä tarkemmin. Yritykset hyötyisivät tästäkin tutkimuksesta paljon. Heillä olisi käytettävissään hyväksymistestauksen prosessit, tehtävät ja roolikuvaukset, jotka tulivat tämän tutkimuksen tuloksena ja sen lisäksi suositus järjestelmästä, joka tukee tätä prosessia.

Lähteet

BCS 2012a. Software Testing Career path. Luettavissa:

<http://certifications.bcs.org/content/ConMediaFile/18088>. Luettu: 28.04.2012.

BCS 2012b. Testing skill level 3. Extranet. SFIAPlus. Browse SFIAPlus. Testing Task

3. Luettavissa: <https://apps.bcs.org/wam/sfiaplus.aspx>. Luettu: 28.04.2012

BCS 2012c. Testing skill level 5. Extranet. SFIAPlus. Browse SFIAPlus. Testing Task 5.

Luettavissa: <https://apps.bcs.org/wam/sfiaplus.aspx>. Luettu: 28.04.2012

Beedle, M. & Schwaber, K. 2002. Agile Software Development with Scrum. Prentice-Hall, Inc. Upper Saddle River, New Jersey, Yhdysvallat.

Broekman, B., Koomen, T., van der Aalst, L. & Vroon, M. 2006. TMap Next for result-driven testing. UTN Publishers. CN 's-Hertogenbosch, Alankomaat.

Crispin, L. & Gregory, J. 2009. Agile testing: a practical guide for testers and agile teams. Pearson Education, Inc. Boston, MA, Yhdysvallat.

Cybercom Finland 2011. Hyväksymistestaus palvelukuvaus. Luettavissa:

<http://www.cybercom.com/fi/Ratkaisut/Digitaaliset-palvelut/Testaus/Hyvaksymistestaus>. Luettu: 07.04.2012.

Haikala, I. & Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt. Talentum. Helsinki.

Hambling, B. & Samaroo, A. 2009. Software Testing: An ISEB Intermediate Certificate. CAPDM Limited. Edinburgh, Skotlanti.

Huovinen, J. 2009. Tietohallintomalli. Luettavissa:

<https://www.tietohallintomalli.fi/tietohallintomalli>. Luettu: 13.05.2012.

IABG. Willkommen auf den V-Modell-Seiten der IABG. Luettavissa: <http://v-modell.iabg.de>. Luettu: 07.03.2012.

IABG 2006a. V-Modell® XT Part 1: Fundamentals of the V-Modell. Luettavissa: http://v-modell.iabg.de/index.php?option=com_docman&task=doc_details&gid=47&Itemid=30. Luettu: 07.03.2012

IABG 2006b. V-Modell® XT Part 3: V-Modell Reference Tailoring. Luettavissa: http://v-modell.iabg.de/index.php?option=com_docman&task=doc_details&gid=47&Itemid=30. Luettu: 07.03.2012

IABG 2006c. V-Modell® XT Part 4: V-Modell Reference Roles. Luettavissa: http://v-modell.iabg.de/index.php?option=com_docman&task=doc_details&gid=47&Itemid=30. Luettu: 07.03.2012

IABG 2006d. V-Modell® XT Part 5: V-Modell Reference Work Products. Luettavissa: http://v-modell.iabg.de/index.php?option=com_docman&task=doc_details&gid=47&Itemid=30. Luettu: 07.03.2012

Julkisen hallinnon tietohallinnon neuvottelukunta 2011. JHS 182 ICT-palvelujen kehittäminen: Laadunvarmistus. Luettavissa: <http://www.jhs-suositukset.fi/suomi/jhs182>. Luettu: 05.02.2012.

Kasurinen, J., Kettunen, V., Smolander, K. & Taipale, O. 2011. A study on agility and testing processes in software organizations. Teoksessa International Symposium on Software Testing and Analysis (toim.). ISSSTA '10 Proceedings of the 19th international symposium on Software testing and analysis, s. 231-240. ACM. New York, Yhdysvallat.

Lindström, J. 2011. Scrum. Luettavissa: <http://reaktor.fi/osaaminen/scrum>. Luettu: 23.02.2012.

Naik, K. & Tripathy, P. 2008. Software testing and quality assurance: Theory and practice. John Wiley & Sons, Inc. New Jersey, Yhdysvallat.

Office of Government Commerce 2007. ITIL – Service Transition. TSO. Norwich, Iso-Britannia.

Shojaee, H. 2012. Scrum in 10 minutes. Katsottavissa:
<http://www.youtube.com/watch?v=XU0llRltyFM>. Katsottu: 28.04.2012.

Liitteet

Liite 1. Suunnitteluvaiheen tehtävät

Suunnitteluvaiheen tehtävät ja niiden tarkoitus. (Broekman ym. 2006, 158-223.)

Tehtävä	Tehtävän tarkoitus
Vahvistetaan toimeksianto	Muotoillaan toimeksianto sellaiseksi, että kaikilla osapuolilla on selkeä käsitys mitkä tämän testitason tavoitteet, tehtävät ja vastuut ovat
Ymmärretään toimeksianto	Haalitaan ymmärrys projektin organisaatiosta, järjestelmään kohdistuvista muutoksista ja niiden tavoitteista, ja vaatimuksista jotka sen tulee täyttää. Tunnistetaan hyväksyjät hyväksymiskriteerejä ja dokumentaatioita hyväksikäyttäen.
Määritellään testauksen perusta	Määritellään ja tunnustetaan testauksen lähdedokumentit siten, että mahdollisimman varhaisessa vaiheessa tiedetään mitä vaatimuksia vasten testattavana olevaa kohdetta testataan. Testauksen perusta on sisältää kaiken dokumentoidut ja ei dokumentoidut vaatimukset.
Analysoidaan tuotteen riskit	Analysoidaan muutoksen alla olevan järjestelmään kohdistuvat riskit ja riskien vaikutukset organisaatiolle. Käytännössä vastataan kysymykseen: mikä on riski organisaatiolle, jos järjestelmässä ei ole odotettua laatua.
Määritellään testaus strategia	Päätetään riskianalyysin perusteella tarvittavat testityypit, sekä testauksen laajuus jokaiselle muutoksen kohteena olevalle järjestelmän ominaisuudelle / objektille.
Työmäärän arviointi	Arvioidaan tarvittava työmäärä testauksen suorittamiselle perustuen testauksen strategiaan, jotta asiakas voi hyväksyä tai vaatia muutoksia testisuunnitelmiin ja siihen liittyviin kuluihin.
Suunnitelman laatiminen	Laaditaan testaustasolle mahdollisimman luotettava suunnitelma, joka pitää sisällään aloitus- ja lopetus-päivämäärät ja testauksen tuotokset. Suunnittelun perustana on, että keskitytään aluksi löytämään ne olennaisimmat viat, joiden löytäminen on kyseisen testaustason tehtävä.
Jaetaan testiyksiköt ja testi teknikat	Jaetaan testauksen kohteena olevat järjestelmän ominaisuudet / objektit pienempiin helpommin hallittaviin testiyksiköihin. Edellisen lisäksi testiyksikölle ja valitun testitason mukaan (esim. toiminnallinen hyväksymistestaus) valitaan yksi tai useampi sopiva testaustekniikka minkä perusteella testaus tulee suunnitella ja toteuttaa.
Määritellään testauksen tuotokset	Määritellään ne tuotokset joita testauksen aika tullaan tuottamaan. Näitä ovat esimerkiksi testausuunnitelma, testausraportit ja jne.
Määritellään organisaatio	Määritellään roolit, tehtävät, valtuudet ja vastuut joita testauksen toteuttamiseen tarvitaan. Tämä pitää sisällään myös henkilöiden allokoinnin rooleihin ja mahdollisten koulutusten järjestämisen.
Määritellään testiympäristö	Määritellään tarvittava testiympäristö, mahdolliset muut testaustyökalut, mahdolliset työtilat ja muodostetaan suunnitelma ympäristön tarpeille.
Johtamisen organisointi	Määritellään miten testausprosessia, ympäristöä, testin tuotoksia ja testauksen aikana löytyviä virheiden organisointia johdetaan.
Selvitetään testausprosessiin kohdistuvat riskit	Selvitetään ja tunnustetaan riskit ja niiden vastatoimet. Tämä harjoitus tehdään pelkätään testausprosessiin.
Testisuunnitelman toteutus ja palaute	Kootaan edellisissä tehtävissä tehdyt tuotokset yhdeksi testisuunnitelmaksi, otetaan vastaan palaute ja tehdään viimeistelevät toimenpiteet. Tämän jälkeen julkaistaan virallinen testisuunnitelma.

Liite 2. Testiympäristön valmistelu ja ylläpito vaiheen tehtävät

Taulukko. Testiympäristön valmistelu ja ylläpito vaiheen tehtävät ja niiden tarkoitus.
(Broekman ym. 2006, 258-270.)

Tehtävä	Tehtävän tarkoitus
Tarkennetaan testiympäristövaatimukset	Suunnitteluvaiheessa testisuunnitelmalle määriteltyjen testiympäristövaatimusten tarkentaminen ja suunnitelman laatiminen.
Suunnitelman toteuttaminen	Toteutetaan aiemmin laadittu suunnitelma. Esimerkiksi tilataan / ostetaan tarvittavat laitteisto ja asennetaan ne.
Määritellään testiympäristön luovuttaminen	Määritellään se lähestymistapa miten testiympäristö luovutetaan testaajille ja tarkistetaan (tarkistuslistaa hyödyntäen), että aikaisemmin määritellyt tarpeet tulee tyydytettyä.
Suoritetaan testiympäristön luovuttaminen	Käydään läpi tarkistuslista, joka luotiin edellisessä vaiheessa. Tässä tehtävässä varmistetaan, että testijärjestelmät, testityökalut ja työtilat ovat kuten on vaadittu.
Testiympäristön ylläpito	Laadukkaalla ylläpidolla varmistetaan testiympäristön saatavuus testaajille testien aikana.
Taltioidaan testiympäristö (valinnainen tehtävä)	Tunnistetaan, päivitetään ja siirretään ylläpidetty testiympäristö siten, että sitä voidaan käyttää myöhemmin uudestaan.

Teemahaastattelujen kysymysrunko:

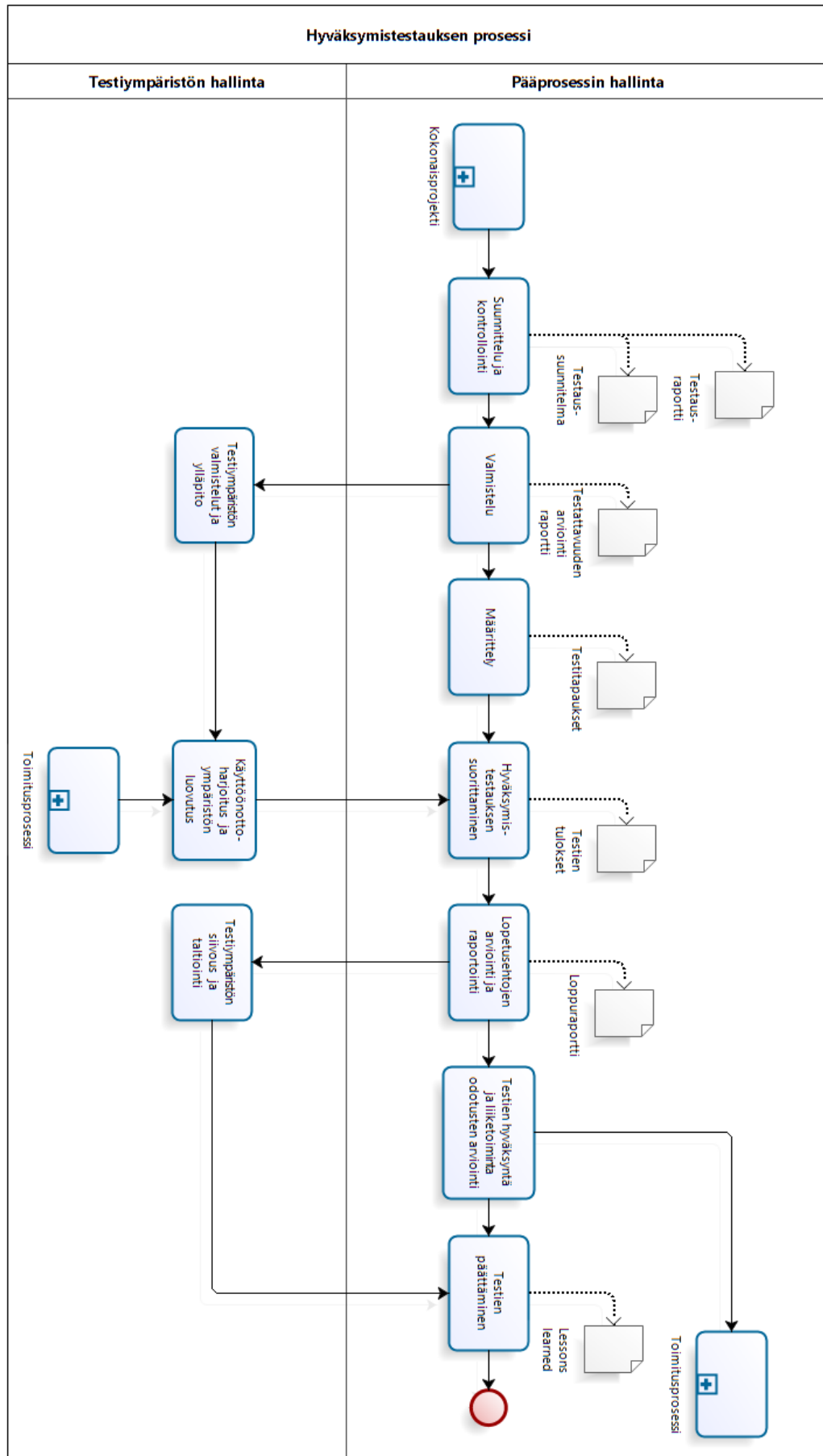
Hyväksymistestaus osana järjestelmäkehitystä

1. V-Malli / vesiputousmallia pidetään optimaalisena järjestelmäkehitysmallina asiakkaan näkökulmasta, onko se mielestäsi sitä? Perustele vastauksesi.
2. Eroaako Scrum tai muu Agile järjestelmäkehitysmalli positiivisesti / negatiivisesti asiakkaan suorittaman hyväksymistestauksen kannalta kun sitä verrataan vesiputous / V-malliin? Perustele vastauksesi.
3. Eroavatko mielestäsi asiakkaan suorittaman hyväksymistestauksen prosessit, roolit ja vastuut V-mallisen ja Scrum mallin välillä? Perustele vastauksesi.
4. Minkälaisiin ongelmiin olet törmännyt järjestelmäkehitysprojekteissa hyväksymistestaus vaiheessa?
5. Tulisiko hyväksymistestaus mielestäsi projektisoida siten, että sillä on omat resurssit, budjetti ja aikataulu? Perustele vastauksesi.
6. Tulisiko asiakkaan johtaa tai olla mukana muissa testaustasoissa kuin hyväksymistestausta? (Yksikkötestaus, Integroititestaus tai järjestelmättestaus) Perustele vastauksesi.
7. Mitkä ovat mielestäsi "Key Success Factors" hyväksymistestauksen onnistumiselle? Perustele vastauksesi.

Hyväksymistestauksen organisointi, roolit ja vastuut

8. Mitä rooleja mielestäsi hyväksymistestauksen prosessin toteuttamiseen kuuluu? Miksi valitsit juuri nämä roolit?
9. Millainen rooli liiketoiminnalla on hyväksymistestauksella? Millaisia rooleja liiketoiminnan edustaja tulisi mielestäsi täyttää? (vs tietohallinnon roolit) Perustele vastauksesi.
10. Jos saisit vapaat kädet suunnitella ja organisoida testaustiimin asiakkaan organisaatiossa, millaisen tiimin kokoaisit? Perustele vastauksesi.
11. Mitkä ominaisuudet/luonteenpiirteet ovat tärkeimmät testauspääliköllä? Perustele vastauksesi.
12. Mitkä ominaisuudet/luonteenpiirteet ovat tärkeimmät testaajalla? Perustele vastauksesi.
13. Mihin asioihin hyväksymistestauksessa tulisi mielestäsi keskittyä? Perustele vastauksesi.
14. Mitä hyötyjä automaatiotestauksella voi saavuttaa hyväksymistestauksessa? Perustele vastauksesi.
15. Mitä hyötyjä näet testaushallinnan työkalun (Esim. HP Quality Center tai Project Top) käyttämisestä hyväksymistestauksen johtamisen tukena?

Liite 4. Hyväksymistestauksen prosessi



Liite 5. Hyväksymistestauksen pääprosessin tehtävät

Hyväksymistestauksen pääprosessin tehtävät			
Prosessin vaihe	Tarkemmat tehtävät	Tuotokset	Vastuullinen
Suunnittelu ja kontrollointi	<ol style="list-style-type: none"> 1. Haalitaan ymmärrys toimeksiannosta ja kirkastetaan toimeksianto kaikille osapuolille. 2. Määritellään testauksen perusta eli määritellään ja tunnistetaan testauksen lähdedokumentit 3. Analysoidaan testauksen kohteeseen kohdistuvat riskit 4. Määritellään testaus strategia 5. Arvioidaan tarvittava työmäärä 6. Laaditaan testaussuunnitelma 7. Jaetaan testiyksiköt ja valitaan sopivat testaustekniikat 8. Määritellään testauksen tuotokset 9. Määritellään testaus organisaatio 10. Määritellään testiympäristö (Järjestelmä, tilat & työkalut) 11. Organisoidaan testauksen johtaminen 12. Servitetaan testausprosessiin kohdistuvat riskit 13. Testaussuunnitelman koostaminen ja palaute 14. Testausraportoinnin suunnittelu ja raportoinnin käynnistäminen 	Testaussuunnitelma Testausraportti (Testauksen eteneminen ja riskit)	Testauspäällikkö
Valmistelu	<ol style="list-style-type: none"> 1. Kerätään testauksen lähdedokumentit (esim. FS / TS dokumentit, haastattelu raportit, vaatimusmäärittelyt, käyttöohjeet jne.) 2. Testiobjektin tarkistuslistan laatiminen, jolla arvioidaan testauskohteen testattavuus 3. Testauksen lähdedokumenttien arviointi 4. Laaditaan testattavuuden arviointi raportti 	Testattavuuden arviointi raportti	Testauspäällikkö
Määrittely	<ol style="list-style-type: none"> 1. Testiyksiköiden testauksen määrittely (sisältää kaiken suunnittelun tarkistuslistoista testitapausten kirjoittamiseen asti) 2. Keskeisten lähtöpisteiden määrittely (Testien aloittamisen lähtöpiste eli määrittelee missä tilassa järjestelmän ja datan täytyy olla kun testaus alkaa) 3. Tarkistuslistan laatiminen testiobjektin vastaanotolle 	Testitapaukset	Testaaja
Hyväksymistestauksen suorittaminen	<ol style="list-style-type: none"> 1. Testiobjektin vastaanotto 2. Lähtöpisteiden valmistelut 3. (Uudelleen-)testauksen suorittaminen 4. Testitulosten tarkitus ja arviointi 	Testien tulokset	Testauspäällikkö (Testiobjektin vastaanotto) Testaaja
Lopetusehtojen arviointi ja raportointi	<ol style="list-style-type: none"> 1. Testitulosten vertaaminen hyväksymiskriteereihin ja arviointi 2. Uudelleentestaus tarpeen arviointi 3. Loppuraportin laatiminen 	Loppuraportti (sisältää suosituksen julkaisun jatkolle)	Testauspäällikkö
Testien hyväksyntä ja liiketoiminta odotusten arviointi	<ol style="list-style-type: none"> 1. Testitulosten esittely liiketoiminta- ja projektijohdolle 2. Hyväksymispäätöksen tekeminen 	Hyväksymispäätös (Go - No Go päätös)	Testauspäällikkö Ohjausryhmä / Liiketoiminta vastaava (hyväksymispäätöksen teko)
Testien päättäminen	<ol style="list-style-type: none"> 1. Testausprosessin arviointi ja opitujen asioiden taltiointi 2. Laaditaan lessons learned dokumentaatio 	Lessons learned - dokumentti	Testauspäällikkö

Liite 6. Hyväksymistestauksen testiympäristön hallinnan tehtävät

Testiympäristön hallinnan tehtävät			
Prosessin vaihe	Tarkemmat tehtävät	Tuotokset	Vastuullinen
Testiympäristön valmistelu ja ylläpito	1. Tarkennetaan testiympäristön vaatimukset ja laaditaan tarkempi suunnitelma 2. Suunnitelman toteuttaminen 3. Määritellään testiympäristön luovuttaminen ja luodaan käyttöönottosuunnitelma	Käyttöönottosuunnitelma <i>Raportointi testauspäällikön testausraportilla(testiprosessin eteneminen)</i>	Testausympäristön koordinaattori
Käyttöönotto harjoitus ja ympäristön luovutus	1. Suoritetaan käyttöönottoharjoitus (simuloidaan Go-Live tilanne) 2. Suoritetaan testiympäristön luovuttaminen 3. Testiympäristön ylläpito koko testijakson ajan	<i>Raportointi testauspäällikön testausraportilla(testiprosessin eteneminen)</i>	Testausympäristön koordinaattori
Testiympäristön siivous ja taltiointi	1. Suoritetaan testiympäristön taltiointi (valinnainen) 2. Testiympäristön siivous		Testausympäristön koordinaattori

Liite 7. Testaajan roolikuvaus

Roolin nimi Testaaja	<i>"Testaaja on kuin viekas kettu, joka yrittää löytää järjestelmästä virheitä mitä viekkaimmilla tavoilla"</i>
Toimenkuva Testaajan työ on laadun varmistaminen liiketoiminnan näkökulmasta. Testaajan tehtävänä hyväksymistestauksessa on todentaa, että järjestelmä täyttää sille annetut vaatimukset. Hänen tulee varmistaa, että vaatimusmäärittelyissä kirjatut hyväksymiskriteerit täyttyvät.	
Keskeisimmät työtehtävät hyväksymistestausprosessissa <ul style="list-style-type: none">• Testauksen lähdedokumenttien testattavuuden arviointi• Testitapausten ja aloitustilanteiden määrittely• Testien aloitustilanteiden valmistelu• Testien suorittaminen (dynaaminen ja staattinen testaus)• Testitulosten sekä hyväksymiskriteerien täyttymisen tarkistus ja arviointi• Vikojen kirjaaminen• Testaus materiaalien taltiointi	
Vaatimukset henkilön luonteenpiirteille <ul style="list-style-type: none">• Henkilöllä on halua toimia testaajana• On luonteeltaan huolellinen ja systemaattinen• On luonteeltaan tarkka ja analyyttinen (osaa olla tekemättä olettamuksia!)• On utelias, kokeilunhaluinen ja omaa luovan ajattelun kyvyn• Omaa hyvät suullisen ja kirjallisen kommunikoinnin taidot• On hyvä tiimipelaaja	
Vaatimukset henkilön osaamiselle <ul style="list-style-type: none">• Liiketoimintaprosessien ja organisaation hyvä tuntemus• Testauksen kohteena olevan ohjelmiston/järjestelmän hyvä tuntemus• Taitoa osata arvioida testattavuutta ja hallita testien suunnittelu tekniikat• Taitoa osata kirjoittaa dokumentteja ja osata käyttää taulukkolaskentaohjelmaa• Taitoa käyttää hyväksi tarkistuslistoja• Taitoa tulkita testauksen lähdedokumentaatioita• Testausautomaation tietämystä ja tuntemusta• Tietämystä yleisistä järjestelmäkehityksen ja testauksen malleista• Tietämystä projektityöstä ja projektiorganisaatioista• Kyky omaksua uusia testaustyökaluja nopeasti• Kyky löytää ohjelmisto/järjestelmä virheitä ja halua ratkaista ne	

Liite 8. Testauspäällikön roolikuvaus

<p>Roolin nimi</p> <p>Testauspäällikkö</p>	<p><i>"Testauspäällikkö on kuin projektipäällikkö ja hänen tehtävänä on johtaa testauksen aktiviteetit maaliin oikealla laadulla, aikataulussa ja budjetissa."</i></p>
<p>Toimenkuva</p> <p>Testauspäällikön työ on johtaa, koordinoita ja varmistaa laadukkaasti toteutettu hyväksymistestaus ohjelmistokehitysprojekteissa. Hän toimii yhdessä projektipäällikön, toimittajan edustajien ja testaajien kanssa ja varmistaa, että kaikki osapuolet ovat tietoisia testausprosessin tilanteesta, riskeistä sekä raportoi testauksen kohteena olevan järjestelmän laadusta.</p>	
<p>Keskeisimmät työtehtävät hyväksymistestausprosessissa</p> <ul style="list-style-type: none"> • Testaussuunnitelman laatiminen, ylläpito ja hyväksynnän hankkiminen • Vastaa hyväksymistestauksen suorittamisesta ajallaan, budjetissa ja oikealla laadulla • Raportoi testausprosessin kulusta koko testausjakson ajan • Koordinoi ja arvioi kaikkia testausprosessin aktiviteetteja ja testaajia testien aikana • Johtaa sisäistä konsultointia ja antaa neuvoja ja opastusta liittyen testauksen suunnitteluun ja suorittamiseen • Johtaa sisäistä laadunvarmistusta • Tunnistaa, ennakoii ja raportoi testauksen ja projektin riskeistä • Laatii testauksen loppuraportin • Kommunikoi aktiivisesti eri tahojen kuten toimittajan, projektipäällikön, testaajien, asiakkaan ja muiden sidosryhmien kanssa testaukseen liittyvissä asioissa 	
<p>Vaatimukset henkilön luonteenpiirteille</p> <ul style="list-style-type: none"> • Henkilö on luonteeltaan auktoriteettinen johtaja • Omaa hyvät henkilöjohtamisen taidot ja osaa motivoida ihmisiä • On luonteeltaan järjestelmällinen, suunnitelmallinen ja on hyvä organisoimaan asioita • Uskaltaa ja osaa tehdä päätöksiä aikailematta perustuen faktoihin, tilanteeseen ja prioriteetteihin • Osaa katsoa asioita yli eri toimintojen rajojen ja on ymmärtää eri toimintojen funktiot • Omaa erinomaiset suullisen ja kirjallisen kommunikaation taidot • Omaa kyvyn hahmottaa ja ymmärtää vaikeita ja monimutkaisia kokonaisuuksia ja osaa pilkkoa ne pienemmiksi ja paremmin ymmärrettäviksi malleiksi / konsepteiksi 	
<p>Vaatimukset henkilön osaamiselle</p> <ul style="list-style-type: none"> • Taitoa osata tehdä realistisia suunnitelmia ja varmistaa niiden toteutuminen • Tietämystä järjestelmäkehityksen erilaisista elinkaarimalleista ja niihin liittyvistä tehtävistä • Testausprosessin ja siihen liittyvien aktiviteettien tunteminen • Testauksen kohteena olevan ohjelmiston/järjestelmän hyvä tunteminen • <u>Kyky osata katselmoida testitapauksia ja testaajien laatimia testaussuunnitelmia</u> • Ymmärtää <u>konfiguraationhallinnan</u> ja sen merkityksen IT palvelujen hallintaan • Omaa ymmärryksen arkkitehtuurin ja järjestelmäkehityksen työkaluihin • Laaja ymmärrys automatisoinnista ja sen hallinnasta • <u>Taitoa osata arvioida IT aktiviteetteja ja raportoida niistä</u> 	

Liite 9. Käsitteet

Häiriö (failure) = Ohjelmiston poikkeama odotetusta toimituksesta, palvelusta tai tuloksesta.

Iteratiivinen kehitysmalli (iterative) = Kehityksen linkaarimalli, jossa projekti jaetaan iteraatioihin. Iteraatio on kokonainen kehitysjakso, joka päättyy lopullisen tuotteen yksittäisen suoritettavan osan julkaisuun.

Järjestelmä = Termiä käytetään kun kyse on ohjelmiston ja laitteiston muodostamasta kokonaisuudesta

Ketterä ohjelmistokehitys (agile software development) = Joukko ohjelmistotuotantoprojekteissa käytettäviä menetelmistöjä, joille on yhteistä toimivan ohjelmiston ensisijaisuus, suora viestintä ja nopea muutoksiin reagointi.

Ohjelmisto = Termi kattaa sekä tietokoneohjelma sekä siihen liittyvät dokumentaatiot

Ohjelmistotuotanto = tietokoneohjelmistojen rakentamiseen yleisesti käytettyjä työkaluja, tekniikoita, menettelytapoja ja periaatteita.

Pyrähdyksen tehtävälista (sprint backlog) = Scrumissa käytetty tehtävälista, joka pitää sisällään pyrähdyksen aikana toteutettavat tehtävät tarkalla tasolla.

Pyrähdys (Sprint) = Scrumin 1-4 viikkoa kestävä kehityssykli, jonka päätteeksi esitellään toteutetut toiminnallisuudet.

Scrum = Scrum on ketterä kehityksen prosessikehys.

Testauksen lähdedokumentit (test basis) = Kaikki dokumentit, joista voi päätellä testattavan kohteen vaatimuksia ja johon testaus perustuu.

Testiskripti (test script) = Testiskripti sitoo yhteen useita testitapauksia, jotta ne voidaan suorittaa yksinkertaisesti ja tehokkaasti.

Testitapaus (test case) = Syötearvojen, suoritusten esiehtojen, odotettujen tulosten ja suoritusten jälkiehtojen muodostama kokonaisuus, joka on muodostettu tiettyä tavoitetta tai testauksen kohdetta varten

Testiyksikkö (test unit) = Kokoelma prosesseja, tapahtumista ja toiminnoista, jotka testataan yhdessä.

Tuotteen kehitysjono (product backlog) = Scrumissa käytetty priorisoitu lista kaikesta, mitä saatetaan kehittää.

Vaiheittainen kehittämismalli (incremental) = Kehityksen elinkaarimalli, jossa projekti on jaettu vaiheisiin, joista jokainen toteuttaa osan projektin koko toiminnallisuuden vaatimuksista.

Vika (fault/defect) = Komponentissa tai järjestelmässä oleva virhe, joka voi aiheuttaa sen, että komponentti tai järjestelmä ei pysty suorittamaan siltä edellytettävää toimintoa.

Virhe (error, mistake, bug) = Testauksen yhteydessä poikkeama spesifikaatiosta.

Virheiden jäljitys (debugging) = Ohjelmistohäiriöiden syiden etsimisen, analysoinnin ja poistamisen prosessi

V-malli = Kehysmalli, joka kuvaa ohjelmistokehityksen elinkaaren toiminnot määrittelystä hyväksymistestaukseen.